TURBOSENS: AN INTERACTIVE TURBOFAN ENVIRONMENT FOR PROBING WORLD MODEL REPRESENTATIONS AGAINST HIDDEN ENGINE STATE

Anonymous authors
Paper under double-blind review

ABSTRACT

World models are expected to learn compact representations that capture the underlying state of dynamical systems, but evaluating whether these representations preserve meaningful latent state remains difficult under partial observability. We introduce TurboSens, a controlled environment for measuring latent state preservation in world models. TurboSens provides a configurable turbofan simulation with paired latent state and sensor observations, allowing representation quality to be evaluated against known underlying system state rather than only through prediction or reconstruction performance. The environment is designed to simulate issues that could arise when relevant state is only partially observable and must be inferred from temporal context, actions, and changing operating conditions. We evaluate representations from several world model families using frozen encoders and inverse probes that measure latent state recovery, recovery as a function of observability, robustness to distribution shift, and responses to maintenance interventions. Across controlled changes in observability and degradation context, we find that self supervised representations preserve far less of the state than the observations demonstrably contain, with recoverability depending on the pretraining objective and on memory aligned with maintenance events rather than on context length. These results highlight a gap between what the observations contain and what learned representations preserve, and demonstrate how controlled simulation can provide a direct measurement of whether world model representations retain the underlying state required for prediction and decision making under partial observability.

1 INTRODUCTION

World models (Balestriero & LeCun, 2025; Assran et al., 2025; Okada & Taniguchi, 2021; 2022; Maes et al., 2026) are increasingly proposed as general purpose representations for complex dynamical systems, trained in a self-supervised manner to predict future state and spanning joint embedding, recurrent latent dynamics, and autoregressive architectures. A central open question is whether such representations recover the *underlying generative* state of the system or merely fit statistical regularities of the observation stream. Answering this requires paired access to sensor observations *and* the true latent state, which existing physics and prognostics benchmarks (Bear et al.; Riochet et al., 2018; Bordes et al., 2025; Saxena et al., 2008) replace by a low dimensional readout (a plausibility verdict, a scalar remaining useful life) that cannot fully identify whether a representation recovered the underlying generative structure.

We introduce **TurboSens**, an interactive turbofan simulator and accompanying dataset designed as an *instrument* for evaluating world-model representations. TurboSens is generated using OpenDeckSMR (Psaropoulos et al., 2025), a component-level thermodynamic aircraft engine simulator that exposes, at every timestep, both a sensor stream and the full ten-dimensional ground-truth health state $\mathbf{s}_t$. This paired access enables an *inverse evaluation protocol*: train a world model self supervised on sensor sequences and maintenance actions, freeze the encoder, then probe the latent space against the true $\mathbf{s}_t$ through state regression, observability stratified diagnostics, and two distribution shifts. Because the true state is available but never used as a training signal, TurboSens asks whether a

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

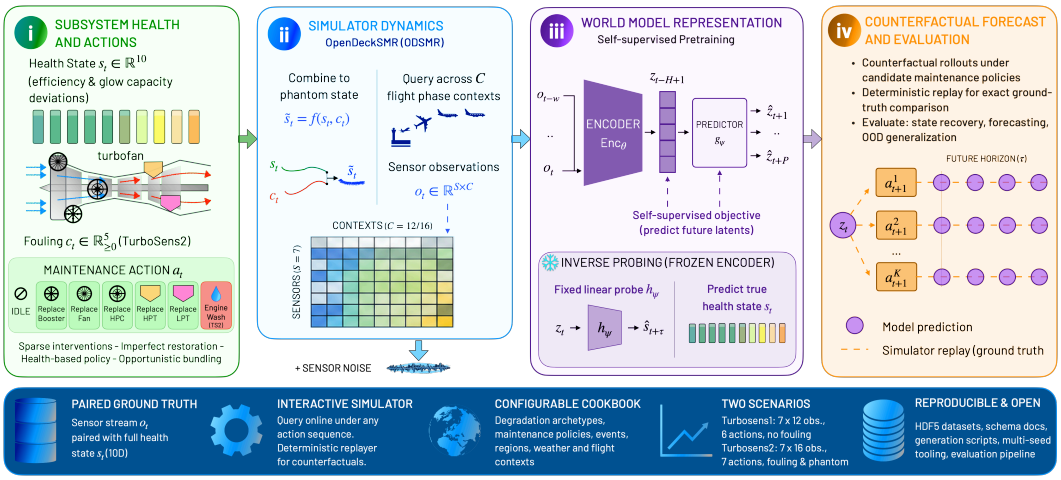


Figure 1: **The TurboSens evaluation framework. (i)** A turbofan engine is decomposed into subsystems whose health $\mathbf{s}_t \in \mathbb{R}^{10}$ degrades over time and is partially restored by discrete maintenance actions a_t; a fouling channel $\mathbf{c}_t$ accumulates between washes. **(ii)** The two channels combine into a phantom state $\tilde{\mathbf{s}}_t = f(\mathbf{s}_t, \mathbf{c}_t)$ that OpenDeckSMR (ODSMR) maps to observations $\mathbf{o}_t \in \mathbb{R}^{11 \times P}$ (seven sensors and four operating point parameters) across P flight operating points. **(iii)** A world model is pretrained self supervised (encoder Enc_θ, predictor g_ψ); at evaluation the encoder is frozen and a linear supervised head trained on its embeddings, which we call a *probe* (h_ψ), decodes them against the ground truth $\mathbf{s}_t$. **(iv)** Counterfactual forecasting forks an episode under alternative actions $\{a^k_{t+1}\}$ and compares the model's rollout against a deterministic simulator replay over a future horizon τ.

learned representation recovers the underlying generative state of engine health from observations alone, going beyond scalar RUL regression and binary violation-of-expectation discrimination.

TurboSens is *interactive* rather than a static dump of trajectories: the simulator can be queried online under arbitrary action sequences, which enables counterfactual replay and on demand generation of shifted regimes, with degradation archetypes, maintenance policies, event rates, and operating regions exposed as independent knobs. Its central difficulty is that observability depends on actions. Mechanical wear is masked by a fouling channel that accumulates between engine washes, so a single observation is consistent with many combinations of wear and fouling, and the sensors expose the true wear only shortly after a wash. A representation must therefore preserve information that becomes informative at specific maintenance events. Every modeling element is loosely inspired by industrial operating practice, but the environment is not calibrated to a real serviced engine; its assumptions are documented and configurable so that users can audit them independently of the protocol. Code at `https://anonymous.4open.science/r/turbosens_ICLR-EBBC`

Contributions.

1. An **interactive turbofan environment** built on OpenDeckSMR that exposes paired ground truth state and sensor observations under discrete maintenance actions, with a two channel hidden state (wear and fouling) whose observability depends on the action history, released with an in distribution split and two statistically validated shift splits.
2. An **inverse probing protocol** with a reference ladder (supervised and self supervised).
3. An **empirical study** of three world model families showing that self supervised pretraining preserves far less state than the observations contain.

2 Related Work

Why turbofan engines. Turbofan degradation is only partially observable through sensors, yet the system carries a physically meaningful latent state (component level efficiency and flow capacity

deviations) (Lacaille, 2014; Karchmer, 1977) that a simulator can emit at every step, so a probe against the ground truth is well posed. The observations are scalar sensor channels rather than pixels, which removes the visual decoding bottleneck of video benchmarks, and a single engine combines noise, multiple operating contexts, action interventions, regional dynamics, and rare events (Riegler & Bichlmaier, 2007; Dunn et al., 1987; Thoma et al., 2020; Aygun et al., 2024). The substrate connects to predictive maintenance (Zonta et al., 2020; Lee et al., 2014; Tsui et al., 2015; Jaw, 2005) and digital twins (Mihai et al., 2022; Belfadel et al., 2025), but our goal is the evaluation of world model representations, not prognostics.

World models and physical understanding. A growing family of world models, spanning joint-embedding predictive architectures (Balestriero & LeCun, 2025; Assran et al., 2025; Maes et al., 2026), recurrent latent-dynamics models (Okada & Taniguchi, 2021; 2022), and autoregressive sequence models (Wang et al., 2020; Krishnan et al., 2022; Deng et al., 2024; Calabrese et al., 2021), train encoders and predictors on high-dimensional observation streams (pixels, sensors) to recover latent dynamics without supervision. A parallel body of work asks whether such representations encode *physical* structure: Physion (Bear et al.) scores rigid-body outcome prediction, CLEVRER (Yi et al.) probes causal reasoning over collision videos, and IntPhys/IntPhys2 (Riochet et al., 2018; Bordes et al., 2025) adopt a violation-of-expectation paradigm. Two structural limitations recur across the field. First, these evaluations reduce physical understanding to binary discrimination, effectively an out-of-distribution detection task that flags anomalies rather than recovering the underlying state. Second, ground-truth access to the latent state is typically unavailable, so evaluation collapses onto downstream task return or reconstruction quality and cannot ensure whether the representation captures physical structure or merely statistical regularities of the observation stream.

Interactive simulators and agentic environments. Gym-style interactive environments are the standard testbed for reinforcement learning and model predictive control (Towers et al., 2024; Todorov et al., 2012), but industrial degradation settings have received comparatively little attention in this form. Maulana et al. (2023) argue for maintenance policies that reason over longer action horizons, a setting that requires an online simulator rather than a static dataset. No existing turbofan benchmark exposes this interface: an agent cannot apply a maintenance action and observe the consequence.

Turbofan degradation datasets. Turbofan prognostics research has long been dominated by C-MAPSS (Saxena et al., 2008) and N-CMAPSS (Arias Chao et al., 2021), which pair sensor streams with a scalar remaining-useful-life label as the sole evaluation target (Ramasso & Saxena, 2015). These datasets established the field but expose neither the full latent health state for inverse problems Vu et al. (2024) nor an interactive interface.

3 Problem Formulation: Turbofan Degradation and Simulator

Evaluating a world model on TurboSens means asking whether its representation captures the latent state that generates the data: the state itself, its dynamics, the effect of actions, and the processes that obscure it. We build the formulation in three steps; Figure 1 summarises it.

Observation, state $\mathbf{s}_t$, and the OpenDeckSMR engine simulator. At every flight cycle t the engine has a latent mechanical wear state $\mathbf{s}_t \in \mathbb{R}^K$ with $K{=}10$, encoding efficiency and flow capacity deviations of the five sub systems (booster, fan, high pressure compressor, high pressure turbine, low pressure turbine). At each flight cycle the engine is evaluated at $P{=}16$ flight operating points $\phi_p \in \mathbb{R}^4$ (ambient temperature deviation, altitude, Mach number, power command). At each of them, the thermodynamic simulator OpenDeckSMR returns seven sensor readings, and the observation stacks these readings with the operating point parameters, which an operator always knows:

$$\mathbf{o}_{t,p} \;=\; \big[\, g_{\mathrm{ODS}}(\tilde{\mathbf{s}}_t,\, \phi_p) + \boldsymbol{\varepsilon}_{t,p}\,;\; \phi_p \,\big] \;\in\; \mathbb{R}^{7+4}, \qquad \mathbf{o}_t = (\mathbf{o}_{t,1}, \ldots, \mathbf{o}_{t,P}) \in \mathbb{R}^{11\times 16}, \tag{1}$$

so a flattened observation has 176 entries. Here $\boldsymbol{\varepsilon}$ is sensor noise and $\tilde{\mathbf{s}}_t$ is the phantom state defined below, which equals $\mathbf{s}_t$ when the engine is clean. The probing target throughout the paper is $\mathbf{s}_t$; everything else either drives its dynamics or shapes how it is observed.

Wear dynamics and maintenance. Each episode is assigned a *degradation archetype* $\eta \in \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$ that acts as a per sub system drift multiplier (compressor heavy, fan and booster

heavy, turbine heavy, balanced), and a *weather* channel w_t modulates the operating envelope and the per step drift. A discrete *maintenance action* $a_t \in \mathcal{A}$ ($|\mathcal{A}|$=7: idle, replacement of one of five module assemblies, engine wash) restores selected components of $\mathbf{s}_t$ through a diagonal restoration mask $R(a_t) \in \{0,1\}^{K\times K}$ with imperfect efficacy $\boldsymbol{\rho}_{\text{mech}}$:

$$\mathbf{s}_{t+1} \;=\; \mathbf{s}_t \;+\; \boldsymbol{\Delta}_s(\mathbf{s}_t,\, \eta,\, w_t) \;+\; R(a_t)\, \boldsymbol{\rho}_{\text{mech}}(a_t,\, \eta) \;+\; \boldsymbol{\eta}_t, \tag{2}$$

where $\boldsymbol{\Delta}_s$ collects the archetype and weather conditional drift and $\boldsymbol{\eta}_t$ is zero mean process noise. Actions are chosen by a heuristic, health correlated policy (life limit ceilings, health triggered visits, opportunistic bundling, imperfect restoration, periodic washes) that is not calibrated to operator records (Appendix B.1.4). Stochastic events (permanent step changes, transient anomalies, sensor perturbations) add sparse discontinuities; they are parameterised abstractions inspired by real phenomena rather than calibrated rates (Appendix B.1.5).

Fouling, phantom observation, and wash actions. A transient *fouling* channel $\mathbf{c}_t \in \mathbb{R}^{K_c}_{\geq 0}$ (K_c=5, one per module) fouls module efficiency and biases corrected flow upward. The engine wash decays $\mathbf{c}_t$ toward zero without touching $\mathbf{s}_t$:

$$\mathbf{c}_{t+1} \;=\; \gamma(a_t)\, \mathbf{c}_t \;+\; \mathbf{u}_t(\phi_t,\, \eta_{\text{reg}}) \;+\; \boldsymbol{\zeta}_t, \tag{3}$$

with $\gamma(a_t)$ the decay factor (close to one for non cleaning actions, small for a wash) and $\mathbf{u}_t$ a region and phase dependent fouling source. Episodes are drawn from one of six *region archetypes* η_{reg} (operating environments characterised by their dominant atmospheric stressor, e.g. Arid, Littoral, Industrial) that shift event rates, archetype priors, and fouling dynamics. The two channels do not reach the sensors independently: they are first combined into a *phantom state* $\tilde{\mathbf{s}}_t = \text{clip}(\mathbf{s}_t + M\max(\mathbf{0}, -\mathbf{s}_t) + C\mathbf{c}_t)$ (full form in Appendix A, eq. 10), with M a sparse non negative coupling that lets local wear inflate downstream flow capacity and C a sparse non negative coupling that lets fouling inflate corrected flow and depress efficiency. The simulator is then queried on $\tilde{\mathbf{s}}_t$ rather than $\mathbf{s}_t$, so a single $\mathbf{o}_t = g_{\text{ODS}}(\tilde{\mathbf{s}}_t, \phi_t) + \boldsymbol{\varepsilon}_t$ is consistent with many $(\mathbf{s}_t, \mathbf{c}_t)$ pairs and the observability of $\mathbf{s}_t$ is action dependent: immediately after a wash, $\mathbf{c}_t \approx 0$ so $\tilde{\mathbf{s}}_t \approx \mathbf{s}_t$ and the sensors expose the true wear; between washes fouling rebuilds and the two diverge. A representation that simply regresses $\tilde{\mathbf{s}}_t$ from $\mathbf{o}_t$ is therefore mis specified, and only a world model that integrates the action history can disentangle the two channels.

POMDP form and evaluation task. Collecting $(\mathbf{s}_t, \mathbf{c}_t, w_t, \phi_t, \eta_{\text{reg}}, e_t)$ into a single latent $\omega_t \in \Omega$ recovers the standard partially observable Markov decision process $(\Omega, \mathcal{A}, \mathcal{O}, f, g, \rho_0)$, which we use in Appendix A for the channel wise transitions in their general form. We ask whether a world model trained self supervised on observation and action histories learns representations that recover $\mathbf{s}_t$. Maintenance actions condition only the predictor during pretraining; the encoder never receives actions and maps a short window of observations to an embedding

$$\mathbf{z}_t \;=\; \text{Enc}_\theta(\mathbf{o}_{t-w+1:t}) \;\in\; \mathbb{R}^d, \tag{4}$$

with w=1 for every configuration except one (w=4), so probed embeddings cannot read the action stream. For recurrent world models we also probe the belief state $\mathbf{b}_t$, the predictor's hidden state rolled over embeddings and actions, and report it separately since actions do enter it. The probed representation $\mathbf{r}_t \in \{\mathbf{z}_t, \mathbf{b}_t\}$ is frozen and a linear probe h_ψ is fit to predict

$$\hat{\mathbf{s}}_t \;=\; h_\psi(\mathbf{r}_t) \tag{5}$$

on held out episodes (Section 5); forecasting at horizons $\tau > 0$ is an auxiliary task (Appendix F).

4 The TurboSens Environment and Dataset

TurboSens is generated with OpenDeckSMR (Psaropoulos et al., 2025), a component level thermodynamic simulator, and exposes *both* the sensor stream and the ground truth state $\mathbf{s}_t$ (together with the fouling state $\mathbf{c}_t$), the property that enables direct evaluation of learned representations. Different operating points excite different parts of the thermodynamic cycle, so the sixteen slices of $\mathbf{o}_t$ carry more information about $\mathbf{s}_t$ together than in isolation. The released data (Table 1) comprise 500 training episodes and four evaluation splits of 60 episodes each: `test` (in distribution), and two shift

Table 1: Core notation and scale of the released data. The full symbol table is Table 3 in Appendix A; a simpler base scenario without fouling is described in Appendix C.1.

Symbol	Description	Value
$\mathbf{s}_t \in \mathbb{R}^K$	Mechanical wear, probing target	K=10
$\mathbf{c}_t \in \mathbb{R}^{K_c}_{\geq 0}$	Fouling channel	K_c=5
$\tilde{\mathbf{s}}_t$	Phantom state queried by the simulator	eq. 12
$\mathbf{o}_t$	7 sensors and 4 operating point parameters, P contexts	$11 \times 16 = 176$
$a_t \in \mathcal{A}$	Idle, five module replacements, engine wash	$\lvert\mathcal{A}\rvert$=7
$\mathbf{z}_t$, $\mathbf{b}_t$	Encoder embedding, recurrent belief state	
h_ψ	Linear ridge probe on the frozen representation	
	Episodes (train / test / test_shift / test_region)	500 / 60 / 60 / 60
	Flights in train; mean episode length	7.0M; 14,022
	Repair rate; wash rate (per step)	0.09%; 1.05%
	Region archetypes; degradation archetypes; event types	6; 4; 12 (+2 regional)

splits, `test_shift` and `test_region`, validated statistically as described in Section 5. Interventions are sparse (component overhauls on about 0.09% of steps, engine washes on about 1%), so trajectories unfold as a sawtooth of partial resets punctuated by rare stochastic events. Wear follows a fast then slowing profile per module (Appendix B.1.3), and regions, weather, events, and maintenance are specified in Appendix B.1. The generator is released with the data, so new splits, policies, and regimes can be produced on demand, and the deterministic `EpisodeReplayer` re executes any episode from its seed, optionally forked under a counterfactual action sequence (Appendix F.2). The released splits are HDF5 files with a documented schema.

5 Evaluation Framework

The environment supports an inverse evaluation protocol: a world model is trained on the sensor stream and maintenance actions of the train split; we then freeze the encoder and *probe* its embeddings against the ground truth mechanical state $\mathbf{s}_t$ that the simulator exposes. By a probe we mean a small head trained on the frozen embeddings to predict the ground truth state; the rest of the world model is held fixed and no probe gradient ever reaches the encoder. The protocol is architecture agnostic: any encoder that produces a per frame embedding $\mathbf{z}_t$ admits the same probe without modification.

We use two distinct words throughout. A *rollout* is what the world model does: starting from a context window of embeddings, the predictor autoregressively produces $\hat{\mathbf{z}}_{t+\tau}$ for $\tau = 1, \ldots, \tau_{\max}$, which the probe decodes to $\hat{\mathbf{s}}_{t+\tau}$. A *replay* is what the simulator does: a deterministic re execution of an episode from its parent seed, optionally forked at a chosen branch point under a counterfactual action sequence (the `EpisodeReplayer` of Appendix F.2). Counterfactual evaluation pairs the two: we compare a model rollout against a simulator replay under the same intervention.

Two evaluation tasks are central to the present paper, and counterfactual replay complements them; two auxiliary tasks (degradation velocity estimation and latent forecasting with an action divergence gap) are shipped and described in Appendix F.

Task 1. Mechanical state estimation. A linear ridge probe h_ψ is fit on the training episodes to map frozen embeddings to $\mathbf{s}_t \in \mathbb{R}^{10}$; its single regularization hyperparameter is chosen by cross validation across training episodes only, and the fit is closed form and deterministic. The probe is then evaluated on held out splits never seen by the world model or the probe. We report the pooled mean Pearson correlation between predictions and the true state, averaged over the ten components (higher is better, 1 is perfect recovery), with per component Pearson, RMSE and R^2 in the appendix. Uncertainty is reported at two levels. Per run: every individual model carries a 95 percent confidence interval from 2000 bootstrap rounds that each redraw the evaluation episodes with replacement and recompute the score; timesteps are never resampled since they are correlated within an episode. Per configuration: each configuration is three independently pretrained models, and tables report mean $\pm$ sd across them.

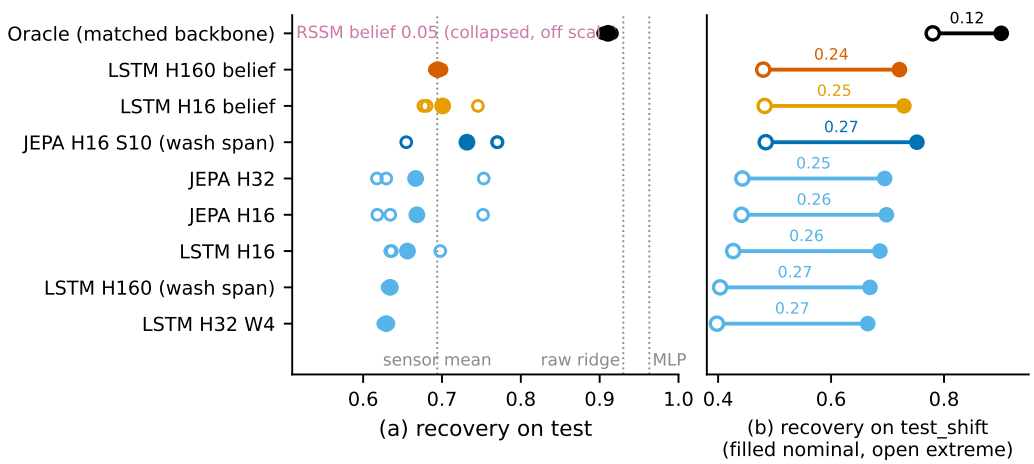


Figure 2: **(a)** State recovery on `test` across the ladder: open circles are pretraining seeds, filled markers seed means, dotted lines the direct references; the RSSM collapses off scale. **(b)** Composition shift on `test_shift`: recovery on nominal (filled) versus extreme (open) episodes, with the gap annotated; every self supervised gap is about twice the oracle's.

Task 2. OOD probing on two validated axes. Two shift splits of 60 episodes each probe orthogonal axes. `test_shift` raises the extreme degradation fraction from 5 to 35 percent while an episode level two sample analysis finds the sensor marginals statistically indistinguishable from `test`, isolating a pure fleet composition shift. `test_region` holds out the two harshest region archetypes; four to six of the ten state components shift significantly at the episode level (BH adjusted $p < 0.05$, statistics in Appendix E.6). We report Task 1 metrics per axis, together with two stratified diagnostics: recovery as a function of flights since the last engine wash (the observability axis of the phantom map), and nominal versus extreme episode strata on the composition axis.

5.1 Results

We apply the protocol to nine trained configurations: seven self supervised world models (JEPA and LSTM predictors trained with the LeWM objective, latent prediction plus SIGReg (Maes et al., 2026), that differ in predictor history H, stride S and encoder window W, and a reconstruction based RSSM) and two supervised references (an MLP on the raw observation and an oracle that trains the self supervised backbone with state supervision). Each is pretrained with three seeds (two for the RSSM, whose belief state is evaluated on one). From these we also probe belief states and *wash anchors*, which concatenate the current embedding with the embedding stored at the most recent wash and need no training. Table 2 and Figure 2 report the reference ladder; run inventories and per run confidence intervals are in Appendix E.

Models. All learned representations share one encoder: a two layer transformer that reads the 176 observation values as separate tokens, mean pools them, and projects them to a 64 dimensional embedding. The LeWM world models place either a causal transformer (JEPA, depth 4) or a two layer LSTM (hidden size 256) on top of this encoder, predict the next embedding from H past embeddings and the maintenance actions, and add the SIGReg term against collapse. The RSSM follows the DreamerV3 design (512 dimensional recurrent state, 32 categorical codes of 32 classes) and reconstructs the observation. Every model is trained for 10 epochs on the 500 training episodes, with three seeds; the oracle trains the same encoder with state supervision. Full specifications are in Appendix D.

State is available but not preserved. The MLP on the raw observation reaches 0.963 ± 0.002, so the state is almost fully extractable from a single observation, and a linear ridge on the raw observation already reaches 0.930. The oracle, which uses the same 1.14M parameter backbone as the self supervised encoders but is trained with state supervision, reaches 0.911 ± 0.006, so encoder

Table 2: The reference ladder. Pooled mean Pearson between linear probe predictions and the true ten dimensional state (higher is better), mean ± sd over pretraining seeds; bold marks the best self supervised entry per block. H is the predictor history in embeddings, S the stride between them (H16 S10 spans 160 flights, more than one wash interval of about 95 flights), W the encoder window. The masking slope is the recovery drop from the freshest to the oldest wash stratum on `test` (Figure 3). Direct probes are deterministic; their episode bootstrap intervals are in Appendix E. ‡One seed.

Representation	test	test_shift	test_region	masking slope
Supervised references				
MLP on raw observation	0.963 ± 0.002	0.933 ± 0.003	0.946 ± 0.005	0.027
Oracle (matched backbone)	0.911 ± 0.006	0.842 ± 0.006	0.882 ± 0.003	0.038
Oracle + wash anchor	0.919 ± 0.006	0.853 ± 0.006	0.891 ± 0.004	0.025
Self supervised snapshot encoders				
JEPA H16	0.668 ± 0.073	0.579 ± 0.046	0.649 ± 0.051	0.062
JEPA H32	0.667 ± 0.075	0.581 ± 0.056	0.652 ± 0.049	0.055
JEPA H16 S10 (wash span)	$\mathbf{0.732 \pm 0.067}$	$\mathbf{0.627 \pm 0.044}$	$\mathbf{0.699 \pm 0.053}$	0.059
LSTM H16	0.656 ± 0.036	0.572 ± 0.023	0.644 ± 0.013	0.058
LSTM H32 W4	0.629 ± 0.003	0.548 ± 0.002	0.631 ± 0.004	0.052
LSTM H160 (wash span)	0.634 ± 0.003	0.553 ± 0.003	0.639 ± 0.006	0.050
Belief states (recurrent, actions integrated)				
LSTM H16 belief	0.701 ± 0.039	0.620 ± 0.017	0.686 ± 0.016	0.046
LSTM H160 belief	0.695 ± 0.004	0.615 ± 0.006	0.686 ± 0.006	0.036
RSSM belief‡	0.045	0.042	0.048	0.003
Zero training anchors				
LSTM H16 + wash anchor	0.683 ± 0.037	0.590 ± 0.021	0.677 ± 0.013	0.029
JEPA H16 S10 + wash anchor	$\mathbf{0.749 \pm 0.057}$	$\mathbf{0.637 \pm 0.042}$	$\mathbf{0.718 \pm 0.042}$	0.031
No learning references (ridge on the observation)				
Raw observation	0.930	0.909	0.919	—
Sensor mean	0.694	0.610	0.702	—
Action history only	0.432	0.431	0.442	—

capacity does not explain the gap at this budget. The best self supervised representation reaches 0.732 ± 0.067, close to the sensor mean baseline (0.694), while maintenance history alone reaches 0.432, so the probe cannot score well by reading the policy. Self supervised pretraining therefore preserves only part of the state that the observations demonstrably contain, and the gap is attributable to the pretext task rather than to the backbone or the data. A stronger probe does not change this picture: a two layer MLP trained on the same frozen embeddings raises the self supervised scores by only 0.04 to 0.08, to at most 0.78, while the oracle stays at 0.912. The gains come mostly just after a wash, so the masking slope grows rather than shrinks (JEPA H16 S10: 0.059 to 0.077), whereas an MLP on the raw observation has a slope of only 0.027 (Appendix E.4). The missing state is not just hard to read out; the embeddings do not hold it in a form either probe can recover. That no learned representation beats the direct references is what the ladder is built to measure.

Seed variability exceeds configuration differences. Within the best JEPA configuration, individual seeds range from 0.655 to 0.770, and seed spreads across JEPA configurations reach 0.13, larger than any difference between JEPA configuration means (at most 0.065). The LSTM encoders with longer history are far more stable (sd 0.003). Single seed comparisons are therefore unreliable on this environment, which is why every configuration is reported over seeds with per run confidence intervals.

Partial observability and event aligned memory. Figure 3 stratifies recovery by flights since the last engine wash. Recovery declines for every frozen representation as the wash recedes, exposing an observability gradient rather than a uniform error. A nonlinear readout of the raw observation nearly removes it, so the information is still present in the instantaneous observation but is not linearly recoverable where fouling is heaviest. Extending the context barely changes the gradient (JEPA H16 to H16 S10: slope 0.062 to 0.059; LSTM H16 to H160: 0.058 to 0.050), whereas storing the embedding from the most recent wash halves it for every model (oracle 0.038 to 0.025, LSTM H16

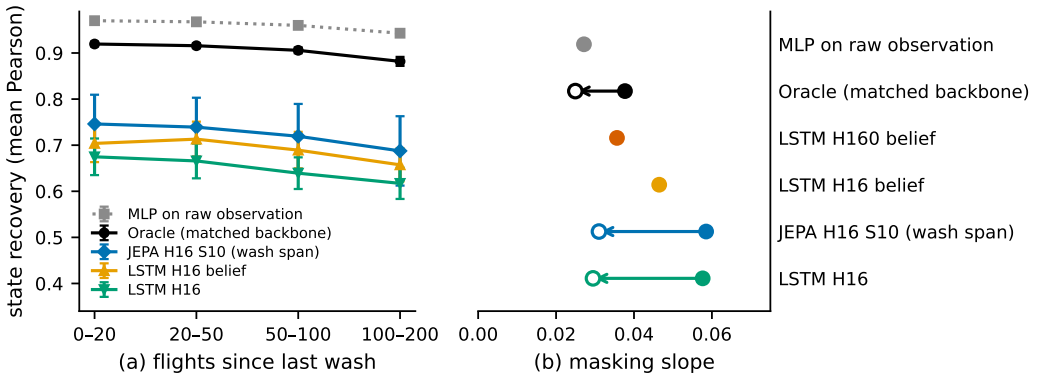


Figure 3: **(a)** State recovery on `test` as a function of flights since the last engine wash. **(b)** Masking slope per representation, with arrows to the same representation plus a wash anchor. Recovery declines as the wash recedes for every frozen linear readout; a stored wash embedding halves the slope at zero training cost, and a nonlinear readout of the raw observation nearly removes it. The stratum before an episode's first wash has one third of the label variance and no anchor, and is reported separately in Appendix E.3.

0.058 to 0.029, JEPA H16 S10 0.059 to 0.031). A belief state that integrates embeddings and actions across a wash cycle (LSTM H160 belief) reaches 0.036, between the snapshot encoders (0.050 to 0.062) and the anchors, at 27 hours of training per seed.

What action integration adds. Because the encoder never sees actions, the protocol can isolate their contribution. Explicit clock features (time since each repair, repair and wash counts; Appendix E.5) raise pooled recovery for every model but lower it before an episode's first wash, where no maintenance has yet occurred (oracle 0.584 to 0.535, self supervised encoders from about 0.24 to about 0.09), so aggregate scores can reward shortcuts that fail in the regimes they do not cover. The direct test holds one trained LSTM fixed and probes two of its representations that differ only in action integration: moving from the encoder embedding to the belief state raises fouling recovery from 0.31 to 0.56 (the oracle reaches 0.55) and state recovery from 0.63 to 0.70. JEPA takes a different route, encoding little fouling (0.35) while recovering the mechanical state better. Partial observability is therefore not addressed by context length alone: what matters is whether the representation preserves information that becomes informative at specific maintenance events.

The two shift axes discriminate differently. On the composition shift (Figure 2b), the oracle loses 0.12 between nominal and extreme episodes while self supervised representations lose 0.24 to 0.27. The region shift is milder for every representation: the direct probes lose 0.011 (ridge) and 0.017 (MLP), the oracle and the JEPA encoders lose 0.02 to 0.03, and the LSTM encoders are essentially unchanged. The composition axis thus separates supervised from self supervised representations, while the region axis mainly probes sensitivity to the operating environment.

RSSM collapse. The RSSM reaches only 0.045, flat across strata and shift axes, although its reconstruction loss converges normally. Yet, it still tracks some level of fouling (0.37). Reconstruction quality alone is therefore not evidence that a world model preserves the state it needs; the LSTM belief states trained with the LeWM objective keep far more (0.70).

Counterfactual replay. Forking held out episodes with the `EpisodeReplayer` and comparing the simulator's state change with the model rollout under the same forced action (Appendix E.8) shows that predictors capture the direction of action effects more reliably than their magnitude (sign agreement 0.60 to 0.74). The clearest failure is the engine wash: the simulator leaves $\mathbf{s}$ unchanged while predictors produce large state changes, reproducing the observable sawtooth without its null mechanical effect.

5.2 Discussion and Limitations

The central finding is that the state a self supervised representation preserves can be far below what its observations contain, and that the loss concentrates where the state is least observable. Pooled scores hide this and single seeds blur it, which is why the protocol stratifies by observability, reports seeds, and reads every reference with the same probe. The results also indicate where to look next: not a longer context but a memory aligned with the actions that reveal the state.

Limitations. TurboSens is synthetic. Sensor generation relies on the OpenDeckSMR thermodynamic model, but degradation, fouling, maintenance, and event rates are shape matched rather than calibrated to operator records (fidelity scope in Appendix E.7), so no operational conclusion about real engines may be drawn from it. Model coverage is limited to matched cost JEPA, LSTM, and RSSM implementations of about one million encoder parameters, and our findings are stated for these models; larger or diffusion based world models are natural subjects for the same protocol. Finally, our probes are a linear model and a small MLP; a larger probe could recover somewhat more, so the scores we report are lower bounds on the state the embeddings contain. However, the field of prognostics favours representations in an accessible state Wang et al. (2018).

6 Conclusion

TurboSens pairs an interactive turbofan environment with the ground truth state that generates its sensors, so that what a world model representation preserves can be measured directly rather than inferred from its training loss. On this environment, self supervised representations preserve far less state than their observations contain, lose it where fouling masks the wear, and recover part of it through memory aligned with maintenance events rather than through longer context. We release the environment, data, and protocol as a common setting for testing whether future world model objectives preserve the state required for prediction and decision making.

Ethics statement

TurboSens is a synthetic environment and contains no personal or operator data. It is not calibrated to any fleet and must not be used to support maintenance or airworthiness decisions on real engines; the paper states this scope explicitly.

AI use statement

We used large language model assistants in three ways. For writing, they helped polish the text and draft parts of sections, which the authors then rewrote and checked. For research execution, they helped write and review parts of the evaluation code (probes, bootstrap statistics, figure and table generation), monitor experiments, and check every reported number against the result files. For ideation, they took part in discussions of experimental controls, such as the nonlinear probe and the single frame baselines. The authors designed the environment and the protocol, verified all results, and take full responsibility for the content of the paper.

Reproducibility statement

We release the environment code (simulator, episode generator, deterministic replayer), the TurboSens splits (HDF5 with a documented schema and Croissant metadata), and the evaluation toolkit (probes, bootstrap statistics, stratified diagnostics). Architectures, training budgets, probe definitions, and split validation statistics are given in Appendices D and E; training code for the reference models will be released with the final version. The data are hosted under an anonymous account for review.

References

Manuel Arias Chao, Chetan Kulkarni, Kai Goebel, and Olga Fink. Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data*, 6(1):5, January 2021. ISSN 2306-5729. doi: 10.3390/data6010005. URL https://www.mdpi.com/2306-5729/6/1/5.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba, Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. (arXiv:2506.09985), 2025. doi: 10.48550/arXiv.2506.09985. URL http://arxiv.org/abs/2506.09985. arXiv:2506.09985 [cs].

Hakan Aygun, Mohammad Rauf Sheikhi, and Hakan Caliskan. Thermodynamics, environmental and sustainability impacts of a turbofan engine under different design conditions considering variable needs in the aviation industry. *Global Challenges*, 8(2):2300205, 2024.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. (arXiv:2511.08544), November 2025. doi: 10.48550/arXiv.2511.08544. URL http://arxiv.org/abs/2511.08544. arXiv:2511.08544 [cs].

Daniel Bear, Elias Wang, Damian Mrowca, Felix Jedidja Binder, Hsiao-Yu Tung, RT Pramod, Cameron Holdaway, Sirui Tao, Kevin A Smith, Fan-Yun Sun, et al. Physion: Evaluating physical prediction from vision in humans and machines. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Abdelhadi Belfadel, Stephen Creff, and Amira Ben Hamida. Advancing industrial digital twins: Towards an open platform aligned with standards. In Pradorn Sureephong, Christophe Danjou, and Abdelaziz Bouras (eds.), *Product Lifecycle Management. Leveraging AI, Digital Twins, and Smart Technologies*, pp. 110–124, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-93323-3.

Florian Bordes, Quentin Garrido, Justine T. Kao, Adina Williams, Michael Rabbat, and Emmanuel Dupoux. Intphys 2: Benchmarking intuitive physics understanding in complex synthetic environments. (arXiv:2506.09849), 2025. doi: 10.48550/arXiv.2506.09849. URL http://arxiv.org/abs/2506.09849. arXiv:2506.09849 [cs].

Francesca Calabrese, Alberto Regattieri, Marco Bortolini, Mauro Gamberi, and Francesco Pilati. Predictive maintenance: A novel framework for a data-driven, semi-supervised, and partially online prognostic health management application in industries. *Applied Sciences*, 11(8):3380, 2021.

Weikun Deng, Khanh TP Nguyen, Christian Gogu, Kamal Medjaher, and Jérôme Morio. Enhancing prognostics for sparse labeled data using advanced contrastive self-supervised learning with downstream integration. *Engineering Applications of Artificial Intelligence*, 138:109268, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Michael G Dunn, Corso Padova, JE Moller, and RM Adams. Performance deterioration of a turbofan and a turbojet engine upon exposure to a dust environment. 1987.

Link C Jaw. Recent advancements in aircraft engine health management (ehm) technologies and recommendations for the next step. In *turbo expo: power for land, sea, and air*, volume 46997, pp. 683–695, 2005.

Allen Martin Karchmer. Identification and measurement of combustion noise from a turbofan engine using correlation and coherence techniques. Technical report, 1977.

Rayan Krishnan, Pranav Rajpurkar, and Eric J Topol. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, 6(12):1346–1352, 2022.

Jérôme Lacaille. Robust monitoring of turbofan sensors. In *2014 IEEE aerospace conference*, pp. 1–8. IEEE, 2014.

Jay Lee, Fangji Wu, Wenyu Zhao, Masoud Ghaffari, Linxia Liao, and David Siegel. Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. *Mechanical systems and signal processing*, 42(1-2):314–334, 2014.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworldmodel: Stable end-to-end joint-embedding predictive architecture from pixels. (arXiv:2603.19312), March 2026. doi: 10.48550/arXiv.2603.19312. URL http://arxiv.org/abs/2603.19312. arXiv:2603.19312 [cs].

Faisal Maulana, Andrew Starr, and Agusmian Partogi Ompusunggu. Explainable data-driven method combined with bayesian filtering for remaining useful lifetime prediction of aircraft engines using nasa cmapss datasets. *Machines*, 11(2):163, January 2023. ISSN 2075-1702. doi: 10.3390/machines11020163. URL https://www.mdpi.com/2075-1702/11/2/163.

Stefan Mihai, Mahnoor Yaqoob, Dang V Hung, William Davis, Praveer Towakel, Mohsin Raza, Mehmet Karamanoglu, Balbir Barn, Dattaprasad Shetve, Raja V Prasad, et al. Digital twins: A survey on enabling technologies, challenges, trends and future prospects. *IEEE Communications Surveys & Tutorials*, 24(4):2255–2291, 2022.

Masashi Okada and Tadahiro Taniguchi. Dreaming: Model-based reinforcement learning by latent imagination without reconstruction. In *2021 ieee international conference on robotics and automation (icra)*, pp. 4209–4215. IEEE, 2021.

Masashi Okada and Tadahiro Taniguchi. Dreamingv2: Reinforcement learning with discrete world models without reconstruction. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 985–991. IEEE, 2022.

Michail Psaropoulos, Vasilis Gkoutzamanis, Anestis I. Kalfas, Panagiotis Giannakakis, Sebastien Razakarivony, Solene Thepaut, and Don Quan Vu. Opendecksmr, December 2025. URL https://doi.org/10.5281/zenodo.17804895.

Emmanuel Ramasso and Abhinav Saxena. *Review and Analysis of Algorithmic Approaches Developed for Prognostics on CMAPSS Dataset*. 2015. URL https://ntrs.nasa.gov/api/citations/20150007677/downloads/20150007677.pdf.

C Riegler and C Bichlmaier. The geared turbofan technology—opportunities, challenges and readiness status. *Porceedings CEAS. Sep*, pp. 10–13, 2007.

Ronan Riochet, Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard, and Emmanuel Dupoux. Intphys: A framework and benchmark for visual intuitive physics reasoning. *arXiv preprint arXiv:1803.07616*, 2018.

Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management*, pp. 1–9, Denver, CO, USA, October 2008. IEEE. ISBN 978-1-4244-1935-7. doi: 10.1109/PHM.2008.4711414. URL http://ieeexplore.ieee.org/document/4711414/.

Evangelia Maria Thoma, Tomas Grönstedt, and Xin Zhao. Quantifying the environmental design trades for a state-of-the-art turbofan engine. *Aerospace*, 7(10):148, 2020.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Kwok L Tsui, Nan Chen, Qiang Zhou, Yizhen Hai, and Wenbin Wang. Prognostics and health management: A review on data driven approaches. *Mathematical Problems in Engineering*, 2015 (1):793161, 2015.

Dong Quan Vu, Sebastien Razakarivony, Yosra Marnissi, and Michel Nocture. A comprehensive literature review on the resolution of turbine engine performances' inverse problems. *Journal of Engineering for Gas Turbines and Power*, 147(5):050801, November 2024. ISSN 0742-4795. doi: 10.1115/1.4066778. URL https://doi.org/10.1115/1.4066778.

Dong Wang, Kwok-Leung Tsui, and Qiang Miao. Prognostics and health management: A review of vibration based bearing and gear health indicators. *IEEE Access*, 6:665–676, 2018. doi: 10.1109/ACCESS.2017.2774261.

Tian Wang, Meina Qiao, Mengyi Zhang, Yi Yang, and Hichem Snoussi. Data-driven prognostic method based on self-supervised learning approaches for fault detection. *Journal of Intelligent Manufacturing*, 31(7):1611–1619, 2020.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. In *International Conference on Learning Representations*.

Tiago Zonta, Cristiano André Da Costa, Rodrigo da Rosa Righi, Miromar José De Lima, Eduardo Silveira Da Trindade, and Guann Pyng Li. Predictive maintenance in the industry 4.0: A systematic literature review. *Computers & industrial engineering*, 150:106889, 2020.

APPENDIX CONTENTS

A POMDP and Observation Model **13**

B The TurboSens Environment **14**

B.1 TurboSens: Regional Operating Environments and Fouling Dynamics 14

C Base Scenario Without Fouling **23**

C.1 Overview 23

D Architecture and Implementation Details **28**

D.1 Sensor Encoder 28

D.2 Temporal Aggregator 28

D.3 World Model and Loss 30

D.4 Predictor and AR LSTM Baseline 30

D.5 Training Setup 30

D.6 Downstream Evaluation Probes 31

E Evaluation Details: Ladder, Split Validation, and Fidelity Scope **31**

E.1 Run inventory 32

E.2 Reference ladder definitions 32

E.3 Wash strata and the pre wash stratum 33

E.4 Nonlinear probe 33

E.5 Action history baseline features 34

E.6 Split validation 34

E.7 Fidelity scope 34

E.8 Counterfactual fidelity curves 34

F Auxiliary Evaluation Tasks **35**

F.1 Degradation velocity estimation 35

F.2 Deterministic replay and counterfactual API 35

F.3 Latent forecasting with action divergence gap 36

A POMDP AND OBSERVATION MODEL

Latent state. An episode of TurboSens is a trajectory of the partially observable Markov decision process

$$\mathcal{M} \;=\; (\Omega,\, \mathcal{A},\, \mathcal{O},\, f,\, g,\, \rho_0),$$

with latent state

$$\omega_t \;=\; \big(\mathbf{s}_t,\, \mathbf{c}_t,\, w_t,\, \phi_t,\, \eta_{\mathrm{reg}},\, e_t\big) \;\in\; \Omega,$$

where $\mathbf{s}_t \in \mathbb{R}^K$ is the persistent mechanical wear (K=10), $\mathbf{c}_t \in \mathbb{R}^{K_c}_{\geq 0}$ is the transient fouling channel (zero in the base scenario), w_t is the weather state (ambient temperature deviation, precipitation gate), $\phi_t \in \mathcal{C}$ is the current flight phase context with $|\mathcal{C}|$=C, η_{reg} is the episode constant region and

archetype label, and $e_t \in \{0,1\}^{n_{\text{evt}}}$ is the current stochastic event flag vector. The engine mechanical state $\mathbf{s}_t$ is the probing target; the other channels are carried because they shape its dynamics or the observation map.

Transitions. The channels of ω_t evolve under a deterministic conditional drift f (conditional on the region and archetype label η_{reg} and the current weather and phase) which is then perturbed by process noise ξ_t and modulated by the maintenance action a_t. The action enters multiplicatively through a per channel factor $\Lambda_k(a_t)$ and additively through a restoration source $\boldsymbol{\rho}_k(a_t, \omega_t)$. The transition is therefore deterministic in f but stochastic overall through ξ_t and the discrete event mask e_t:

$$\omega_{t+1}^{(k)} \;=\; \Lambda_k(a_t)\, f_k(\omega_t) \;+\; \boldsymbol{\rho}_k(a_t, \omega_t) \;+\; \xi_t^{(k)}, \qquad k = 1, \ldots, n_{\text{ch}}. \tag{6}$$

When several independent modulations act on the same channel (for instance the fouling decay factor of a cleaning action combined with a context dependent deposition gate), the per channel factor is itself a product series

$$\Lambda_k(a_t) \;=\; \prod_{j=1}^{J_k} \lambda_{k,j}(a_t), \tag{7}$$

so the joint effect of the action across channels is captured by the product $\prod_{k,j} \lambda_{k,j}(a_t)$ rather than by a single transition matrix. Concretely, for the two channels that carry most of the paper's signal,

$$\mathbf{s}_{t+1} \;=\; \mathbf{s}_t \;+\; \boldsymbol{\Delta}_s(\mathbf{s}_t, \eta_{\text{reg}}) \;+\; R(a_t)\, \boldsymbol{\rho}_{\text{mech}}(a_t, \eta_{\text{reg}}) \;+\; \boldsymbol{\eta}_t, \tag{8}$$

$$\mathbf{c}_{t+1} \;=\; \gamma(a_t)\, \mathbf{c}_t \;+\; \mathbf{u}_t(\phi_t, \eta_{\text{reg}}) \;+\; \boldsymbol{\zeta}_t, \tag{9}$$

where $\boldsymbol{\Delta}_s$ is the archetype specific degradation drift (Appendix C.1.2), $R(a_t) \in \{0,1\}^{K \times K}$ is the diagonal restoration mask of action a_t applied to the components it targets with per component magnitude $\boldsymbol{\rho}_{\text{mech}}$, $\gamma(a_t) \in [0,1]$ is the fouling decay factor of the action (equal to 1 for non cleaning actions and to a small value for cleans), $\mathbf{u}_t$ is the region and phase dependent fouling source, and $\boldsymbol{\eta}_t, \boldsymbol{\zeta}_t$ are zero mean Gaussian noise terms that collect into the single noise symbol ξ_t of equation 6.

Phantom observation map. Sensors do not observe ω_t directly. In TurboSens, the observation map g first collapses the two wear channels through a phantom state

$$\tilde{\mathbf{s}}_t \;=\; \text{clip}\big(\mathbf{s}_t \;+\; M\, \max(\mathbf{0},\, -\mathbf{s}_t) \;+\; C\, \mathbf{c}_t,\; s_{\min},\, s_{\max}\big), \tag{10}$$

and then evaluates the OpenDeckSMR thermodynamic simulator at the current flight phase:

$$\mathbf{o}_t \;=\; g(\omega_t) \;+\; \boldsymbol{\varepsilon}_t \;=\; g_{\text{ODS}}\big(\tilde{\mathbf{s}}_t,\, \phi_t\big) \;+\; \boldsymbol{\varepsilon}_t, \qquad \mathbf{o}_t \in \mathbb{R}^{S \times C}, \tag{11}$$

with $M \in \mathbb{R}^{K \times K}$ a fixed sign reversal mask allowing specific failure modes to manifest as apparent improvement, $C \in \mathbb{R}^{K \times K_c}$ a fouling to wear coupling matrix, and g_{ODS} the thermodynamic observation map evaluated across C flight phase contexts simultaneously. In In the base scenario, there is no fouling channel and $\tilde{\mathbf{s}}_t = \mathbf{s}_t$, so the observation reduces to $\mathbf{o}_t = g_{\text{ODS}}(\mathbf{s}_t, \phi_t) + \boldsymbol{\varepsilon}_t$.

Action dependent observability. The equations above imply that the observability of $\mathbf{s}_t$ is itself a function of the action history. Immediately after a cleaning action, $\mathbf{c}_{t+1} \approx 0$ so $\tilde{\mathbf{s}}_{t+1} \approx \mathbf{s}_{t+1}$ and the sensor stream reveals the true mechanical state; between cleans, fouling rebuilds and $\tilde{\mathbf{s}}_t$ diverges from $\mathbf{s}_t$, obscuring it. A representation that regresses directly on the phantom is mis specified; a world model that conditions on the action history can in principle identify both channels by comparing observations before and after clean events.

B The TurboSens Environment

B.1 TurboSens: Regional Operating Environments and Fouling Dynamics

TurboSens extends the simpler base scenario (Appendix C.1) along three physical axes that are absent from existing turbofan benchmarks: the operating environment in which an engine spends its service life, the separation between true mechanical wear and reversible fouling of the gas path, and the cross-module aerodynamic coupling that causes a locally damaged component to make its neighbours appear healthier than they are. The scenario retains the core ingredients of the base

Symbol	Description
POMDP latent state	
$\omega_t \in \Omega$	Full POMDP latent state at time t
$\mathbf{s}_t \in \mathbb{R}^K$	Engine mechanical wear (probing target)
$s_t^k \in [-b_k, 0+\epsilon]$	Wear of component k
b_k	Per component degradation bound
$\mathbf{c}_t \in \mathbb{R}^{K_c}_{\geq 0}$	Fouling channel
$\tilde{\mathbf{s}}_t$	Phantom state, $\tilde{\mathbf{s}}_t = g_\phi(\mathbf{s}_t, \mathbf{c}_t)$
w_t	Weather state (ambient ΔT, precipitation gate)
$\phi_t \in \mathcal{C}$	Flight phase context
η_{reg}	Episode constant region and archetype label
$e_t \in \{0,1\}^{n_{\text{evt}}}$	Stochastic event flag vector
K, K_c	Mechanical and fouling dimensions (K=10)
S	Number of sensor types (7)
C	Number of flight phase contexts (4 / 12 / 16)
n_{ch}	Number of latent channels in ω_t
Actions and dynamics	
$a_t \in \mathcal{A}$	Maintenance action at time t
$\Lambda_k(a_t)$	Per channel multiplicative action factor
$\lambda_{k,j}(a_t)$	Independent modulation j on channel k, $\Lambda_k = \prod_j \lambda_{k,j}$
$R(a_t) \in \{0,1\}^{K\times K}$	Diagonal restoration mask of action a_t
$\boldsymbol{\rho}_{\text{mech}}$	Per component restoration magnitude
$\gamma(a_t) \in [0,1]$	Fouling decay factor
κ	Restoration quality coefficient
ξ_t	Process noise collecting $(\boldsymbol{\eta}_t, \boldsymbol{\zeta}_t, \ldots)$
T	Episode length (variable)
Observations and probing	
$\mathbf{o}_t \in \mathbb{R}^{S\times P}$	Observation tensor at time t (P operating contexts)
g, g_{ODS}	Observation map; OpenDeckSMR thermodynamic map
$M \in \mathbb{R}^{K\times K}$	Sign reversal mask in the phantom state
$C \in \mathbb{R}^{K\times K_c}$	Fouling to wear coupling matrix
$\boldsymbol{\varepsilon}_t$	Observation noise
$\mathbf{z}_t \in \mathbb{R}^d$	Encoder embedding, $\text{Enc}_\theta(\mathbf{o}_{\leq t}, a_{\leq t})$
$\mathbf{z}_t^{\text{agg}} \in \mathbb{R}^d$	Temporally aggregated embedding
$\hat{\mathbf{s}}_{t+\tau}$	Probe prediction at horizon τ
h_ψ	Probe head, $\hat{\mathbf{s}}_{t+\tau} = h_\psi(\mathbf{z}_t)$
d	Embedding dimension
w	Encoder temporal window size
H	Predictor history (context) length
seq_len	Context length of the downstream target probe
λ	SIGReg regularisation weight

Table 3: Full notation used throughout the paper. The phantom state $\tilde{\mathbf{s}}_t$ is defined by equation 10.

scenario (heterogeneous degradation archetypes, life-limiting-part regulatory constraints, an imperfect expert maintenance policy, a library of stochastic events, multi-context sensor observations and a fully exposed hidden state) and layers on top of them a regional operating model, a two-channel hidden state, a seasonal weather process with shop-visit downtimes, and an additional maintenance action capturing periodic engine water washing.

The dataset is organised around six region archetypes, each representing a distinct operating environment characterised by its dominant atmospheric stressor (sand, salt, humidity, pollution, thermal swing). A region archetype simultaneously biases the prior over degradation archetypes, scales the mechanical drift rates of the most exposed sub-systems, governs the fouling accumulation rate, restricts the library of admissible stochastic events, and modulates the seasonal weather process and the flight cadence. The generator supports between 80 and 400 episodes per region archetype; the released TurboSens splits comprise 500 training episodes plus 60 episodes for each of `test`, `replication`, `test_shift`, and `test_region` with a mean length of 14,000 timesteps and a hard ceiling of 20,000 timesteps.

B.1.1 TWO-CHANNEL HIDDEN STATE

Mechanical and fouling channels. The base scenario hidden health state $\mathbf{s}_t \in \mathbb{R}^{10}$ is preserved and interpreted strictly as the *mechanical* deviation of efficiency and corrected flow from nominal for the five sub-systems (Booster, Fan, HPC, HPT, LPT). Its evolution is driven solely by slow stochastic wear (archetype-specific drift, burn-in transient, and speed-regime noise); it is irreversible except through partial restoration at shop visits. To capture the fast, reversible fouling of the gas path we introduce a second hidden channel $\mathbf{c}_t \in [0,1]^5$ encoding a normalised fouling level per sub-system. Fouling accumulates each flight at a rate conditioned on region, flight phase and active stochastic event set, and resets to zero after any engine-wash action. Crucially, $\mathbf{c}_t$ does *not* enter the mechanical state update: $\mathbf{s}_t$ and $\mathbf{c}_t$ evolve on independent dynamics, representing two physically distinct phenomena (material wear vs. surface deposition).

Observation model: the phantom state. The thermodynamic simulator produces sensor readings from an engine state vector, so every hidden effect that modifies the gas-path flow must enter the simulator through that vector. We therefore define a *phantom state* $\tilde{\mathbf{s}}_t \in \mathbb{R}^{10}$ that aggregates, on top of $\mathbf{s}_t$, the two observable-but-hidden effects that the simulator cannot model intrinsically:

1. *Phantom flow coupling*: a breach or severe local wear redirects airflow through neighbouring modules, producing corrected-flow readings that are anomalously high for the healthy-looking neighbour while the true cause lies elsewhere.
2. *Fouling coupling*: surface fouling perturbs module efficiency and apparent flow even when the underlying mechanical state is pristine.

Both effects are implemented as additive, non-negative couplings clipped to the validated thermodynamic operating range:

$$\tilde{s}_t^k \;=\; \mathrm{clip}\Big(s_t^k \;+\; \sum_{j=1}^{10} M_{kj}\,\max\big(0,\,-s_t^j\big) \;+\; \sum_{m=1}^{5} C_{km}\,c_t^m,\;\; \underline{b}_k,\,\overline{b}_k\Big), \quad k = 1,\dots,10, \tag{12}$$

where $M \in \mathbb{R}^{10\times 10}$ is the sparse phantom flow-coupling matrix of the base scenario (strictly positive entries only on flow dimensions immediately downstream of each degrading efficiency component), and $C \in \mathbb{R}^{10\times 5}$ is a new sparse, non-negative fouling-to-phantom matrix whose entry C_{km} is non-zero only when fouling of module m physically perturbs state dimension k (for example, HPC fouling degrades HPC efficiency and biases HPC flow upward). The mechanical-state bounds $[\underline{b}_k, \overline{b}_k]$ are those used by the underlying simulator: strictly non-positive on efficiency channels, and admitting positive flow deviations up to $+0.03$ for Fan, Booster and HPC flow and $+0.05$ for HPT and LPT flow. Coupling weights in M and C are calibrated empirically to keep $\tilde{\mathbf{s}}_t$ within validated thermodynamic ranges under all archetype and region combinations and are exposed as named constants to support ablation studies; M is fixed across the dataset and C is drawn once per engine, while $\mathbf{s}_t$ and $\mathbf{c}_t$ vary per episode and timestep.

Final observation map. In the base scenario the fouling channel is absent, $\tilde{\mathbf{s}}_t$ collapses to $\mathbf{s}_t$, and the simulator is queried directly on the mechanical state. In TurboSens, the simulator is queried on $\tilde{\mathbf{s}}_t$ rather than $\mathbf{s}_t$, so fouling and phantom coupling both reach the sensors through the same physically grounded pathway, the state vector that the thermodynamic model actually consumes:

$$\mathbf{o}_t \;=\; g\big(\tilde{\mathbf{s}}_t, \phi_t\big) \;+\; \varepsilon_t, \tag{13}$$

where g is the OpenDeckSMR thermodynamic model, ϕ_t collects the ambient and command conditions of the flight context, and ε_t is residual sensor noise. Because $\mathbf{c}_t$ enters the observation exclusively through $\tilde{\mathbf{s}}_t$ and is reset by the engine-wash action while $\mathbf{s}_t$ is not, a wash produces the characteristic sawtooth pattern in the sensor stream while the underlying mechanical trajectory $\mathbf{s}_t$ is unaffected. Both $\mathbf{c}_t$ and the phantom contribution $\tilde{\mathbf{s}}_t - \mathbf{s}_t$ are exposed in the dataset metadata as per-step ground-truth labels, enabling supervised evaluation of disentangling methods.

B.1.2 REGION ARCHETYPES

Each episode is assigned one of six region archetypes at generation time (Table 4). A region archetype determines four generation-time parameters: (i) a prior over the four base scenario degradation

archetypes, (ii) a per-subsystem mechanical drift scaling factor, (iii) a baseline fouling accumulation rate, and (iv) a mean flight cadence controlling how many simulated flights map to one calendar day.

Table 4: Region archetypes in TurboSens. Severity is a qualitative summary of combined mechanical and fouling load.

ID	Name	Dominant stressor	Severity
R1	Continental	None (baseline)	Benign
R2	Littoral	Salt spray, humidity	Moderate
R3	Arid	Sand, dust, thermal swing	Harsh
R4	Equatorial	Heat, humidity, bio-fouling	Moderate
R5	Highland	Cold ambient, thin air	Benign-Moderate
R6	Industrial	Chemical pollution, soot	Moderate-Harsh

Region-conditional archetype prior. The four base scenario degradation archetypes (A compressor-driven, B fan/booster-driven, C turbine-driven, D balanced) are retained unchanged. Their prior probability is now conditioned on the region (Table 5), so that for instance Arid over-represents A (sand erosion of the HPC) and Littoral over-represents B (salt corrosion of the fan and booster).

Table 5: Conditional prior $\mathbb{P}(\text{archetype} \mid \text{region})$ used at episode generation.

Region	A	B	C	D
Continental	0.30	0.20	0.25	0.25
Littoral	0.20	0.40	0.20	0.20
Arid	0.45	0.30	0.15	0.10
Equatorial	0.30	0.30	0.20	0.20
Highland	0.20	0.15	0.35	0.30
Industrial	0.35	0.25	0.20	0.20

Per-subsystem mechanical scaling. Independently of the archetype, the region applies a multiplicative scaling $\rho_k^{(r)} > 0$ to the drift rate α_k of equation (18), modelling the physical acceleration of wear by the regional stressor (Table 6). Littoral scales fan and booster drift, Arid scales HPC drift, Highland shifts stress toward the thermal section.

Table 6: Region-induced mechanical drift multipliers on α_k and baseline fouling accumulation multiplier.

Region	HPC	Fan / Booster	HPT / LPT	Fouling rate
Continental	1.0	1.0	1.0	1.0×
Littoral	1.0	1.5	1.1	1.4×
Arid	1.6	1.3	1.0	1.8×
Equatorial	1.1	1.2	1.0	1.5×
Highland	0.9	0.9	1.2	0.7×
Industrial	1.2	1.2	1.1	1.6×

B.1.3 Degradation and Fouling Dynamics

Extended mechanical increment. The base scenario degradation increment (18) is extended with two region-conditional terms: the region scaling $\rho_k^{(r)}$ and a slow corrosion term driven by the accumulated fouling,

$$\Delta s_t^k \;=\; -\beta_k e^{-t_k/\tau_k} \;-\; \rho_k^{(r)}\,\alpha_k^{(m)} \;-\; \lambda_k\, c_t^{\pi(k)} \;+\; \varepsilon_t, \tag{14}$$

where $\pi(k) \in \{1, \ldots, 5\}$ maps a state component to its parent sub-system and λ_k is a small corrosion coefficient set so that a sub-system held at $c = 1$ for an entire episode would contribute an additional

drift of the same order as 15 to 20 percent of the base archetype drift. Salt and sand therefore accelerate mechanical wear only to the extent that cleaning is deferred.

Fouling increment. Fouling accumulates between maintenance visits at a rate modulated by the region archetype and the flight phase of each of the sixteen snapshot contexts seen within a step:

$$\Delta c_t^{\ell} \;=\; \eta_{\ell}^{(r)} \sum_{i=1}^{16} w_{\phi_i} \;-\; \delta\, c_t^{\ell}, \tag{15}$$

where $\eta_{\ell}^{(r)}$ is the region-conditional deposition rate (listed last column of Table 6 after normalisation by sub-system exposure), w_{ϕ_i} weights the contribution of each snapshot context by its flight phase (takeoff and climb deposit more than cruise), and δ is a small self-cleaning term accounting for aerodynamic blow-off. Fouling is clipped to $[0, 1]$ per sub-system.

Events (Section B.1.5) can additionally apply a large step increment to $\mathbf{c}_t$ (a `trans_drift_1` or `regional_perm_1` event deposits a bulk increment on $\mathbf{c}_t$ in a single step) or, for events such as `perm_step_1`, a joint step change to both $\mathbf{s}_t$ and $\mathbf{c}_t$ that is visible in the snapshot at the event step. Abstract labels are used everywhere in the paper; the released simulator currently exposes legacy code labels that will be aligned with the abstract scheme in a subsequent release.

Phantom-coupling interpretation. Because the phantom map (12) depends on the true state $\mathbf{s}_t$, an episode in which a single efficiency dimension degrades aggressively will exhibit a non-physical improvement in the downstream flow dimension of a neighbouring module, followed by a discontinuous return to nominal after maintenance restores the efficiency. This provides a genuine counter-example to monotonic degradation heuristics and serves as a latent-structure probe for representation learning: a world model that has learned the true mechanical state should detect the upstream efficiency loss as the cause of the downstream flow excursion, whereas a purely local model will mis-attribute the event.

B.1.4 Maintenance Policy and Extended Action Set

Engine water wash. A seventh maintenance action, `engine_wash`, is added to the original six base scenario actions (Table 7). A wash resets the fouling vector $\mathbf{c}_t$ to zero (up to a small residual noise), does not restore mechanical state, and incurs only a short downtime (no multi-month calendar jump). Wash timings are not triggered by health; they follow a semi-periodic schedule per episode, with the mean wash interval drawn from $[50, 150]$ steps for benign regions and $[30, 90]$ steps for harsh regions, jittered by a per-wash noise term. This aims at modeling the scheduled maintenance practice of most commercial operators and produces the sawtooth pattern in the observed sensor stream that our dataset is designed to expose.

Table 7: TurboSens maintenance actions. Actions 0 through 5 are inherited from the base scenario; action 6 is the new engine wash.

ID	Name	Target
0	`do_nothing`	None
1	`fan_overhaul`	Fan Eff, Fan Wc
2	`hpc_overhaul`	Booster Eff/Wc, HPC Eff/Wc
3	`turbine_overhaul`	HPT Eff/Wc, LPT Eff/Wc
4	`full_overhaul`	All ten mechanical components
5	`targeted_patch`	Most-degraded component (oracle)
6	`engine_wash`	Reset $\mathbf{c}_t$ (fouling only)

Shop-visit downtime and calendar jumps. Unlike the base scenario, TurboSens represents the wall-clock cost of a shop visit in the weather process rather than abstracting it away. Each non-wash maintenance visit advances the episode calendar clock by a duration drawn from a severity-dependent range: light targeted patches add 0.5 to 1 month, mid-weight module overhauls add 2 to 3 months, and heavy or extreme-degradation overhauls add 5 months. The seasonal weather state (Section B.1.7)

jumps forward accordingly at the next step, producing abrupt ambient conditions transitions at the return from a shop visit that are a distinctive feature of fleet operations. Fouling is reset to zero at every maintenance visit, whether or not a wash action is formally issued.

Seasonal modulation of proactive scheduling. TurboSens makes the calendar trigger from the base scenario explicit through the seasonal weather process. The proactive scheduler now scales its borderline window by a seasonal factor $\kappa_{\text{season}}(t) \in [0.6, 1.3]$ that peaks in local summer (stronger deferral pressure) and is lowest in local winter (stronger anticipation pressure). Life-limiting-part ceilings and emergency thresholds are not subject to this seasonal modulation: airworthiness-mandated replacement remains inflexible whatever the season.

LLP and base policy structure. Life-limiting-part cycle limits are identical across all regions and archetypes, reflecting their regulatory origin rather than any physical susceptibility. Health-correlated scheduling, min-built bundling, deferred repairs and imperfect restoration behave exactly as in the base scenario, operating on the mechanical state $\mathbf{s}_t$ and ignoring the fouling channel $\mathbf{c}_t$. The wash action is handled as a separate track that never interacts with the LLP counter and never causes min-built bundling.

B.1.5 Stochastic Event Library

The base scenario event library is retained and extended with region based gates on admissible events. `trans_drift_2` is excluded in Arid and Equatorial regions regardless of instantaneous weather, since the ambient temperature gate that conditions it is never satisfied in those regions. `regional_perm_1` is strongly over represented in Arid and to a lesser extent Industrial regions. `regional_drift_1`, introduced in TurboSens as a new event type, is gated to Littoral and occasionally Equatorial. Events that jointly affect $\mathbf{s}_t$ and $\mathbf{c}_t$ (`perm_step_1` and `regional_perm_1`) apply their joint step change at the event timestep, so that the snapshot observation at that step already reflects the perturbed engine.

The event rate per region is calibrated to keep the global stochastic event rate below one percent of timesteps, preserving the sparsity property that makes events challenging to detect. All event metadata (type, timestep, affected components, magnitude) is exposed as ground-truth labels for anomaly-detection and representation-probing tasks.

B.1.6 Flight Contexts

Each timestep is observed under $C = 16$ fixed flight contexts spanning three physically supported phase types in OpenDeckSMR: cruise (CR), max-climb (MCL) and take-off (MTO). The breakdown is eight nominal cruise contexts spanning altitudes from 33,000 to 39,000 ft and Mach numbers from 0.76 to 0.84, one low-power cruise context at 15,000 ft approximating an idle approach regime, four max-climb contexts covering nominal, stepped, hot and cold variants, and three take-off contexts covering nominal, hot-day and heavy-weight variants. The low-power context is labelled as a low-thrust proxy rather than a true descent phase because OpenDeckSMR does not validate thermodynamic behaviour in the windmilling descent regime; its command value is reduced to the lower bound of the CR envelope.

Each context remains internally consistent: take-off contexts combine sea-level altitude with high ambient temperature and zero Mach, cruise contexts combine high altitude with low ambient temperature and high Mach, climb contexts span intermediate values. Cross-contaminating context values (for example a cruise DTAMB applied at take-off altitude) are explicitly forbidden by the sampler to keep every snapshot inside the validated operating envelope of the physics simulator.

B.1.7 Seasonal Weather Process

TurboSens replaces the single-level Ornstein-Uhlenbeck weather process of the base scenario with a three-level hierarchical generator that captures seasonality, flight-to-flight variability and within-snapshot noise.

Seasonal baseline. A slow sinusoidal drift $\mu^{(r)}_{\text{season}}(d_t)$ indexed by the simulated calendar day d_t models the region-specific annual cycle of ambient temperature. Its amplitude is largest for

Continental (approximately ± 25 K peak-to-peak around the regional mean), intermediate for Highland and Industrial, and smallest for Equatorial (near-flat). The calendar day advances by $1/\nu^{(r)}$ per timestep, where $\nu^{(r)}$ is the region-conditional mean flight cadence: Continental and Industrial at 4 to 5 flights per day, Littoral and Equatorial at 2 to 3, Arid and Highland at 1 to 2, with per-region jitter. Maintenance downtimes advance the calendar by the severity dependent amount described in Section B.1.4.

Flight-level OU. Around the seasonal baseline, a faster OU process governs the weather realised for one flight,

$$d(\Delta T_{\text{flight}}) = \theta_{\text{f}}\big(\mu^{(r)}_{\text{season}}(d_t) - \Delta T_{\text{flight}}\big)\, dt + \sigma_{\text{f}}\, dW_t, \tag{16}$$

with θ_{f} chosen so the mean-reversion half-life covers several flights and σ_{f} scaled to produce a realistic day-to-day ambient dispersion. A fresh realisation is drawn at the start of each simulated flight so that consecutive snapshots within a flight share correlated weather, while consecutive flights can differ substantially.

Within-snapshot jitter. A third small-amplitude Gaussian perturbation is finally added to each of the sixteen snapshot contexts within a step, in addition to the context-specific offsets documented in Section B.1.6. This captures turbulence and small-scale atmospheric variability that differentiates the snapshots even under an otherwise stable flight-level OU.

B.1.8 Ground-Truth Metadata

Every TurboSens episode carries a dense set of ground-truth labels exposed alongside the sensor stream, making the dataset usable for self-supervised pretraining, probing, anomaly detection and fine grained diagnostic studies:

- Region archetype label $r \in \{\text{R1}, \ldots, \text{R6}\}$ and degradation archetype label $\in \{\text{A,B,C,D}\}$.
- Mechanical state trajectory $\mathbf{s}_t \in \mathbb{R}^{10}$ and fouling trajectory $\mathbf{c}_t \in \mathbb{R}^5$ at every step.
- Phantom-coupled state $\tilde{\mathbf{s}}_t$ and the coupling matrix M used to generate it.
- Action label $a_t \in \{0, \ldots, 6\}$ including the new wash action, together with The base scenario extended action metadata (component-service mask, visit-type flag, deferred repair count).
- Stochastic event mask and per-event parameters.
- Simulated calendar day d_t, season label, and flag indicating whether the step follows a shop-visit calendar jump.
- Flight cadence $\nu^{(r)}$ realised for the episode, wash-schedule timings, and an extreme-degradation boolean flag.

A minority of episodes is flagged as extreme degradation episodes at generation time (2.8 percent in `train`, 5 percent in `test`, 35 percent in `test_shift`). In these episodes the proactive policy's planning window and emergency threshold are temporarily relaxed, so one or more components traverse a deep sag toward the degradation bound before a heavy overhaul is dispatched; the overhaul restores the affected components close to the pristine bound, resets fouling, and advances the calendar clock. Flagged episodes carry metadata describing the sag window, the components involved, and the restoration quality.

B.1.9 Dataset Statistics

Table 8 reports the observed statistics of the released TurboSens splits. The train split contains 500 episodes and each evaluation split 60 episodes (drawn with held out seeds, identical generator); `test` uses the training priors, `test_shift` raises the fraction of extreme degradation episodes to 35 percent, and `test_region` draws only from the two held out regions.

All splits exhibit a healthy 100% valid-flight rate, reflecting the context-envelope hardening introduced in TurboSens (see §B.1.6); no OpenDeckSMR convergence failures occur in the released data. Repair

Table 8: TurboSens dataset statistics, measured on the released HDF5 files.

Property	Train	Test	Test_shift	Test_region
Episodes	500	60	60	60
Total flights	7,010,872	830,756	847,654	842,000
Mean episode length (steps)	14,022	13,846	14,128	14,033
Min / max episode length	8,000 / 20,000	8,000 / 20,000	8,000 / 20,000	8,015 / 20,000
Extreme-degradation episodes	2.8%	5.0%	35.0%	1.7%
Engine-wash fraction	1.052%	1.043%	1.044%	1.173%
Repair fraction (non-wash)	0.088%	0.089%	0.091%	0.087%
of which proactive	0.085%	0.084%	0.085%	0.082%
of which re-repair	0.004%	0.005%	0.004%	0.004%

actions are dominated by `proactive` visits, consistent with the health-correlated maintenance policy: emergency visits are essentially absent because the maintenance scheduler intervenes before components exceed the airworthiness threshold. Engine-wash events fire roughly once per 100 flights, which amounts to about ten wash cycles per average-length episode and gives the world model meaningful opportunities to observe the phantom-to-true transition.

Region and archetype coverage. Table 9 shows the joint region $\times$ archetype episode distribution for the train split. The region prior is skewed toward Industrial (97) and Highland (91) and away from Arid (70) and Littoral (71); the archetype prior is dominated by A_compressor (179) and B_fan_booster (128). The region-conditional archetype prior described in §B.1.2 produces the expected concentrations (Arid is the only region where A_compressor reaches 40 of 70, reflecting HPC-fouling bias; Highland reaches 42 turbine episodes out of 91). Figure 4 visualises the full heatmap for all three splits.

Table 9: Region and archetype marginal counts for the TurboSens train split.

Region	# Episodes	Archetype	# Episodes
Continental	84	A_compressor	179
Littoral	71	B_fan_booster	128
Arid	70	C_turbine	121
Equatorial	87	D_balanced	72
Highland	91		
Industrial	97		

Event rates by region. Table 10 reports the rising-edge event rates per ten thousand flights on the train split. The two *regional* events (`regional_perm_1`, `regional_drift_1`) fire orders of magnitude more often than the global event library and carry strong regional gating: `regional_perm_1` reaches 898 firings per ten thousand flights in Arid and 339 in Industrial while dropping to 31 in Highland, and `regional_drift_1` is concentrated in Littoral, Equatorial and Industrial while being fully gated off in Arid and Highland. The global events retain order-of-magnitude rates of 0.1 to 2.7 per ten thousand flights across all regions, matching the base scenario rate envelope.

B.1.10 Sensor, Hidden-State and Fouling Marginals

The dataset ships with three empirical artefacts that together characterise the observation surface of TurboSens. Figure 5 overlays the marginal distribution of each of the seven DeckSMR sensors across the sixteen flight-phase contexts, illustrating both the context separation in the observation space and the inter-episode spread induced by the context-jitter mechanism described in §B.1.6. Figure 6 contrasts the marginals of the true mechanical state $\mathbf{s}$ (blue) with those of the phantom state $\tilde{\mathbf{s}}$ (orange) on a per-component basis: the five Efficiency channels at even indices coincide almost perfectly, but the five flow capacity (Wc) channels at odd indices are systematically shifted upward in $\tilde{\mathbf{s}}$, a direct empirical footprint of the phantom coupling $C\mathbf{c}_t$ in equation 12. Figure 7 shows the per-module fouling marginals, bounded in $[0, 0.04]$ and exponentially decreasing due to the wash

Table 10: Event rising-edge rate per 10^4 flights, TurboSens train split. Two rightmost columns show the two regional-only events; all other rows are the base scenario event library. Rates are sampling artefacts of the generator and not calibrated to real failure statistics.

Event	All	Cont.	Litt.	Arid	Equa.	High.	Ind.
`perm_step_1`	0.11	0.14	0.18	0.07	0.11	0.08	0.09
`trans_drift_1`	1.80	1.50	1.04	3.93	1.13	0.78	2.65
`trans_anomaly_1`	1.05	1.05	1.09	0.91	1.18	1.01	1.03
`perm_step_2`	0.33	0.31	0.23	0.37	0.40	0.37	0.29
`trans_fouling_1`	0.19	0.24	0.16	0.17	0.17	0.16	0.21
`trans_anomaly_2`	0.44	0.46	0.40	0.34	0.47	0.44	0.50
`perm_drift_1`	0.11	0.14	0.15	0.08	0.08	0.10	0.10
`perm_drift_2`	0.30	0.37	0.30	0.26	0.34	0.30	0.25
`sensor_pulse_1`	1.00	0.92	0.98	1.12	1.05	0.98	0.96
`trans_drift_2`	0.23	0.25	0.17	0.00	0.00	0.65	0.21
`regional_perm_1`	245.91	49.38	48.54	897.67	197.38	30.76	338.53
`regional_drift_1`	23.70	33.93	1.51	0.00	47.92	0.00	48.96

schedule. Figure 8 shows the episode-length distribution, which is capped at 20 000 steps and floored at 8 000 steps.

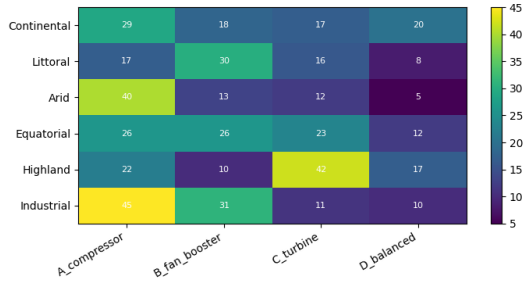


Figure 4: Region × archetype episode counts for the TurboSens train split.

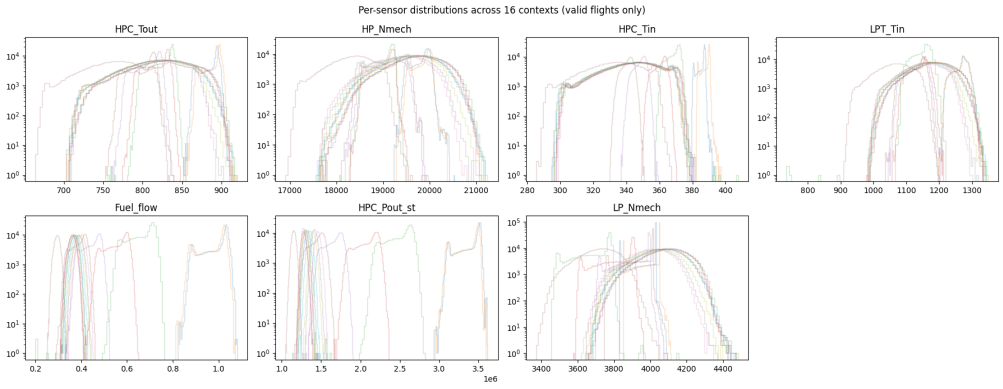


Figure 5: Marginal distribution of the seven DeckSMR sensors across the sixteen flight-phase contexts, TurboSens train split. Each subplot overlays the sixteen contexts for one sensor on a shared log-scale y-axis; the separation between curves reflects the context-dependent operating envelope exposed to the world model via ϕ_t.

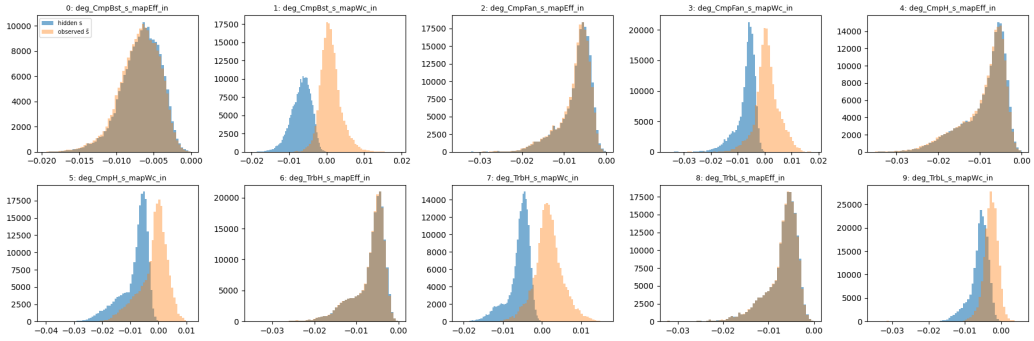


Figure 6: Per-component marginals of the true mechanical state $\mathbf{s}_t$ (blue) and the phantom state $\tilde{\mathbf{s}}_t$ (orange), TurboSens train split. Efficiency channels at even indices overlap almost perfectly; flow capacity channels at odd indices are displaced upward in $\tilde{\mathbf{s}}_t$ by the fouling-driven coupling $C\mathbf{c}_t$. The gap between the two histograms is exactly the non-identifiability signal the world model must learn to close by integrating the action history.

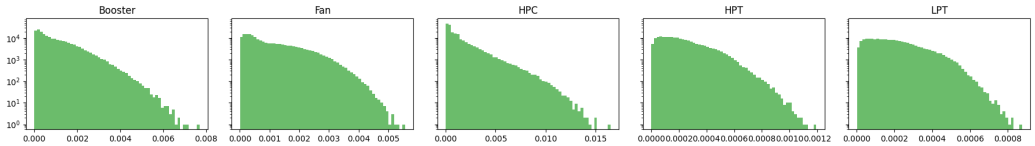


Figure 7: Marginal distribution of the five fouling channels (Booster, Fan, HPC, HPT, LPT), TurboSens train split. Log-scale y-axis. The HPC channel concentrates higher mass at large values, reflecting the combination of (i) higher base fouling rate and (ii) a sulfur-driven multiplier in the Industrial region.

C Base Scenario Without Fouling

The base scenario is the first scenario of the environment: it shares the wear, maintenance, and event machinery of TurboSens but has no fouling channel, no region archetypes, and twelve operating contexts. It is released as a simpler setting and is not used in the main text.

C.1 Overview

We introduce a synthetic turbofan engine degradation dataset designed to benchmark world-model and representation-learning methods under conditions of partial observability, heterogeneous failure modes, and maintenance interventions. Unlike existing turbofan benchmarks such as CMAPSS (Saxena et al., 2008), which expose only a single degradation trajectory with scalar health indicators, the base scenario provides a *full ground-truth hidden state* alongside structured sensor observations, enabling the direct evaluation of latent-state recovery rather than remaining-useful-life regression alone.

The released base scenario dataset comprises 400 episodes of 10,000 timesteps each, totalling 4×10^6 timesteps; the generator's maximum episode length cap is reached by all released episodes. Each timestep represents one flight cycle. At every step the simulator records a ten dimensional hidden degradation state $\mathbf{s}_t \in \mathbb{R}^{10}$, a sensor observation tensor $\mathbf{o}_t \in \mathbb{R}^{7\times 12}$, a discrete maintenance action $a_t \in \{0, \ldots, 5\}$, a stochastic event mask, and weather covariates. The hidden state is available at training time as a supervised signal and at evaluation time for probing learned representations; it is never used as an input to the agent.

C.1.1 State and Observation Spaces

Hidden health state. The state $\mathbf{s}_t = (s_t^1, \ldots, s_t^{10})^\top$ encodes the degradation of five turbofan sub-systems, each represented by an efficiency deviation and a corrected flow deviation:

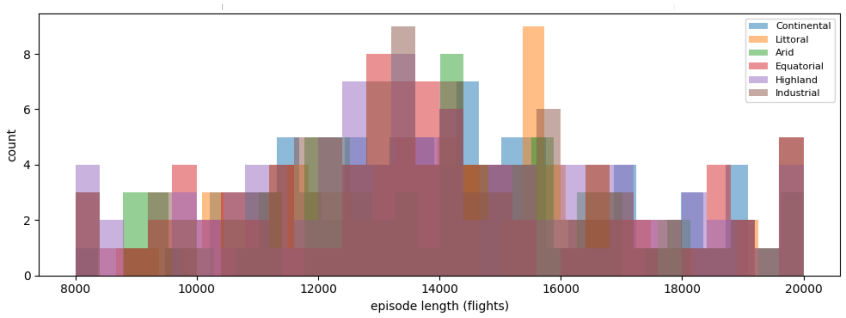


Figure 8: Episode length distribution, TurboSens train split. Lengths are bounded in $[8\,000, 20\,000]$ steps and centred at roughly $14\,000$ steps; episodes are terminated when the maintenance scheduler cannot keep the mechanical state above the proactive threshold.

$$s_t^k \in [-b_k,\ 0], \quad k = 1, \ldots, 10, \tag{17}$$

so that $s_t^k = 0$ denotes a pristine component and $s_t^k = -b_k$ corresponds to the OpenDeckSMR airworthiness bound for that dimension, where b_k is component-specific. The five sub-systems are: Booster (indices 1, 2), Fan (3, 4), High-Pressure Compressor (HPC; 5, 6), High-Pressure Turbine (HPT; 7, 8), and Low-Pressure Turbine (LPT; 9, 10).

Initial state and manufacturing variability. Each episode initialises the health state as $\mathbf{s}_0 \sim \mathcal{N}(\mathbf{0},\ \sigma_0^2\mathbf{I})$, truncated to $[-b_k/4,\ 0]$ per component to avoid initialising engines already near their airworthiness limits. This reflects the empirical observation that turbofan engines exhibit measurable component-level discrepancies in efficiency and flow capacity directly out of the factory, before any degradation has accumulated. The dispersion σ_0 is calibrated so that the initial spread accounts for roughly 5–10% of the per-component degradation budget, consistent with manufacturer acceptance test data.

Sensor observations. At each timestep the physics simulator is queried at twelve fixed flight contexts (see Section C.1.5), yielding an observation matrix $\mathbf{o}_t \in \mathbb{R}^{7\times 12}$. The seven sensor channels are listed in Table 11. The multi-context structure is deliberate: different flight conditions excite different parts of the thermodynamic cycle, so the twelve slices of $\mathbf{o}_t$ together carry more information about $\mathbf{s}_t$ than any single context in isolation. A world model that learns to aggregate evidence across contexts is therefore rewarded.

Table 11: Sensor channels in $\mathbf{o}_t \in \mathbb{R}^{7\times 12}$.

Index	Name	Description
0	`HPC_Tout`	HPC outlet total temperature
1	`HP_Nmech`	HP shaft rotational speed
2	`HPC_Tin`	HPC inlet total temperature
3	`LPT_Tin`	LPT inlet total temperature
4	`Fuel_flow`	Fuel mass flow rate
5	`HPC_Pout_st`	HPC outlet static pressure
6	`LP_Nmech`	LP shaft rotational speed

C.1.2 Degradation Dynamics

Burn-in and steady drift. Component degradation follows a two-regime model. In the early life of each component, a rapid non-linear wear phase (burn-in) is superimposed on a persistent linear drift. The incremental update at each step is

$$\Delta s_t^k = -\beta_k \exp\left(-\frac{t_k}{\tau_k}\right) - \alpha_k^{(m)} + \varepsilon_t, \tag{18}$$

where t_k is the time since the last repair of component k, $\beta_k > 0$ is the burn-in amplitude, τ_k is the burn-in time constant sampled from $\mathcal{N}(500, 80^2)$ steps, $\alpha_k^{(m)}$ is the drift rate under speed regime $m \in \{\text{slow, normal, fast}\}$, and $\varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ is per-step Gaussian noise. Integrating (18) from a pristine component gives

$$\Delta s^k(T) = \beta_k \tau_k \Big(1 - e^{-T/\tau_k}\Big) + \alpha_k T, \tag{19}$$

which is calibrated so that a typical engine consumes roughly 70 to 80 percent of its degradation budget over a 10,000-step episode. Speed regimes are resampled independently per component every 500 steps according to per-component Categorical priors, introducing temporally correlated but heterogeneous degradation rates across the state vector.

Degradation archetypes. To model fleet heterogeneity, each episode is assigned one of four failure-mode archetypes at generation time (Table 12). An archetype modulates α_k and β_k via per-component multipliers, concentrating wear in the sub-system that characterises the failure mode. Archetypes A, B and C may have a delayed onset: with probability 0.30 the failure mode is active from the first step; otherwise, onset is drawn uniformly from the interval $[5\%, 35\%]$ of the episode length, before which the engine degrades as a D-balanced engine.

Table 12: Degradation archetypes and their dominant sub-systems.

ID	Name	Global scale	Dominant degradation
A	Compressor-driven	$1.0\times$	HPC and Booster (α up to $2.8\times$ base); models fouling and erosion in the hot section
B	Fan/Booster-driven	$1.0\times$	Fan and Booster (α up to $2.5\times$); models leading-edge erosion
C	Turbine-driven	$1.0\times$	HPT and LPT (α up to $2.8\times$); models thermal fatigue
D	Balanced	$0.65\times$	All components degrade at 65% of base rate; long-lived engine

The training split uses prior probabilities $\mathbb{P}(\text{A}) = 0.35$, $\mathbb{P}(\text{B}) = \mathbb{P}(\text{C}) = 0.25$, $\mathbb{P}(\text{D}) = 0.15$, reflecting the empirical distribution of failure modes in commercial turbofan fleets. An episode terminates early if any component reaches 97% of its degradation bound, modelling an airworthiness limit. In the generated dataset this end-of-life criterion is triggered in 91.7% of A-type episodes, 60.2% of C-type, 38.4% of B-type, and only 0.7% of D-type episodes, confirming that D engines typically exhaust the maximum episode length rather than failing catastrophically.

C.1.3 Maintenance Policy and Actions

Six discrete maintenance actions are available (Table 13). The maintenance policy is a heuristic, health correlated scheduler with imperfect execution, inspired by condition based practice but not calibrated to operator records; it is designed to produce action sequences that are strongly correlated with the hidden health state while still leaving room for improvement by a learned agent.

Life-Limiting-Part (LLP) ceilings. Each of the ten health-state components is associated with a fleet-wide, regulation-level cycle limit L_k that is identical across archetypes, modelling international airworthiness requirements. A per-component counter c_k is incremented each step and reset to zero when the component is replaced. Whenever $c_k = L_k$ for any component, a maintenance visit is forced before the next step, irrespective of the current health state. LLP ceilings therefore act as a hard upper bound on the inter-maintenance interval and are state-independent by design.

Health-correlated proactive scheduling. On top of the LLP ceiling, the policy estimates for each component the predicted time until either the LLP cycle limit or an internal health threshold (a fraction of the degradation bound, kept conservative so that airworthiness margins are never approached) is reached, based on the current instantaneous degradation rate. The trigger is then modulated by the operational calendar: commercial operators try to minimise scheduled visits during

peak traffic windows so that engines are not removed from service while the airline most needs them, and prefer to bring forward a borderline visit into the surrounding shoulder season when it is operationally safe to do so. Concretely, a flight that lands inside a high-demand window is more likely to defer a borderline-but-safe visit and a flight that lands outside it is more likely to bring one forward, conditional on the engine condition permitting it; the airworthiness margin remains intact in either case. Because the predicted time depends on the degradation rate, which is itself governed by archetype, speed regime and recent event history, the timing of health-triggered visits is correlated with the latent state, with the calendar acting as a soft secondary modulator.

Min-built opportunistic bundling. Whenever a maintenance visit is triggered for any reason (LLP, health, or deferred repair-driven, see below), the policy scans all remaining components and applies min-built logic: if component k is predicted to need replacement within a min-built window $W_{\mathrm{mb}} \in [2000, 3000]$ steps under its current degradation trajectory, it is serviced during the same visit. Because an engine shop visit has a large fixed downtime cost in practice (on the order of months), bundling is strongly preferred over revisiting. If the bundle spans two of the three engine modules, the recorded action is promoted to `full_overhaul` rather than leaving one module idle during the downtime.

Imperfect restoration. The restoration applied at a maintenance visit is imperfect even for LLP-mandated replacements. A newly installed component is mounted inside an ageing engine environment whose bearings, seals, casings and neighbouring modules continue to wear, so no action restores the engine to a pristine overall state. Concretely, an LLP replacement restores the targeted component by a fraction drawn from a tight distribution centred around 0.80 to 0.90 of its degradation budget; a repair action restores by a left-skewed fraction in the ranges given in Table 13, reflecting the variability of human intervention quality. A manufacturing-quality coefficient $\kappa \sim \mathcal{U}(0.70, 1.00)$ multiplies the restoration uniformly across all visits of an episode, capturing engine-to-engine variability in shop quality.

Interval asymmetry. The combination of LLP ceilings, health-correlated scheduling, min-built bundling and deferred repairs produces an inter-maintenance interval distribution that is strongly asymmetric in the dataset. The LLP ceiling imposes a sharp upper cutoff, the health trigger concentrates most visits in a band that depends on the archetype-specific degradation rate, and deferred repairs populate a short-interval tail via forced emergency visits. This is intentionally different from the earlier symmetric uniform scheduling used in our prior datasets, in which action timing was statistically independent of the hidden state and therefore provided no signal for horizon prediction.

Table 13: Maintenance actions and the components each one services.

ID	Name	Components serviced
0	`do_nothing`	None
1	`fan_overhaul`	Fan Eff, Fan Wc
2	`hpc_overhaul`	Booster Eff/Wc, HPC Eff/Wc
3	`turbine_overhaul`	HPT Eff/Wc, LPT Eff/Wc
4	`full_overhaul`	All ten components
5	`targeted_patch`	Most-degraded component (oracle)

Extended action metadata. To carry the richer information produced by the LLP and min-built logic, each recorded action is accompanied by a metadata vector containing: a ten-dimensional boolean mask of components actually serviced during the visit (LLP and min-built inclusions expand this mask beyond the nominal action target); a categorical flag distinguishing LLP-forced, health-triggered, and deferred repair-forced visits; and the per-visit deferred repair count. This preserves the discrete six-action label used by the RL environment while exposing the full per-component servicing information needed by diagnostics and representation-learning probes.

C.1.4 Stochastic Event Library

The probabilities and magnitudes of these events are not calibrated to real-world aircraft failure rates and do not represent specific failure modes observed in operational fleets. They are included to

make the latent state recovery problem harder and the encoder representations more challenging to learn, by injecting sparse, structured discontinuities into the trajectories of $\mathbf{s}_t$, $\mathbf{c}_t$ or $\mathbf{o}_t$. Calibrating the events to operational data, or implementing physically grounded event models more broadly, is left for future work. We accordingly use abstract effect typed labels in the paper (`perm_step_*`, `trans_anomaly_*`, etc.).

Ten categories of stochastic events enrich the dataset (Table 14). Events are drawn from independent Poisson processes. Permanent events (`perm_step_1`, `perm_step_2`, `perm_drift_1`, `perm_drift_2`) apply an immediate step change or sustained drift increment to one or more components of $\mathbf{s}_t$. Transient events (`trans_anomaly_1`, `trans_anomaly_2`, `trans_drift_1`, `trans_drift_2`, `trans_fouling_1`) inject a short-lived perturbation that decays back toward baseline. Sensor-level events (`sensor_pulse_1`) perturb the observation tensor $\mathbf{o}_t$ without modifying the latent state, deliberately introducing observation noise that is not explained by $(\mathbf{s}_t, \mathbf{c}_t)$ and thus testing a world model's ability to distinguish structural degradation from transient sensor artefacts. The overall stochastic event rate across the training set is 0.77% of timesteps. Some events are phase-gated to low-altitude contexts and others additionally require a cold ambient temperature below a threshold derived from the ISA standard lapse rate.

Table 14: Stochastic event library: label, type, and mean Poisson rate per $10{,}000$ steps. Rates and effect magnitudes are not calibrated to any real-world statistics.

Event	Type	Rate / 10k	Effect
`perm_step_1`	Permanent	0.1	Single-component step decrement on $\mathbf{s}$
`perm_step_2`	Permanent	0.3	Step decrement on a different component group
`perm_drift_1`	Permanent	0.1	Slow multi-component drift increment
`perm_drift_2`	Permanent	0.3	Single-channel sustained drift
`trans_anomaly_1`	Transient	1.0	Short pulse on shaft-speed channels
`trans_anomaly_2`	Transient	0.5	Short pulse with elevated drift
`trans_drift_1`	Transient	1.5	Multi-step drift-rate boost
`trans_drift_2`	Transient	0.3	Phase-gated drift modulator (cold contexts)
`trans_fouling_1`	Transient	0.2	Multi-step bias on fouling channel
`sensor_pulse_1`	Sensor	1.0	Instantaneous outlier on one sensor channel

C.1.5 Flight Contexts and Weather Process

Each step is observed under $C = 12$ fixed flight contexts spanning three operating phases: six cruise conditions, three climb conditions, and three takeoff conditions. The contexts vary along four axes: altitude (0 to 37,000 ft), Mach number (0 to 0.82), ambient temperature deviation ΔT_{amb} (from -5 K to $+20$ K above ISA), and thrust command. This design ensures that sensors associated with the compressor path (HPC_Tout, HP_Nmech) are strongly excited during takeoff, while cruise contexts provide the most sensitive signal for turbine-path sensors, making context selection a non-trivial component of inference.

The weather process models temporal correlations in ΔT_{amb} via an Ornstein-Uhlenbeck (OU) process,

$$d(\Delta T) = \theta\big(\mu - \Delta T\big)\, dt + \sigma\, dW_t, \tag{20}$$

generating slowly varying ambient conditions with a characteristic mean-reversion timescale of roughly 25 steps (confirmed by the empirical autocorrelation function). Rain is modelled as a Bernoulli gate that opens only when ΔT_{amb} is below a threshold, introducing correlation between precipitation and cold-ambient contexts. The episode-level rain rate averages approximately 27%.

C.1.6 Dataset Statistics and Health Assessment

Table 15 summarises the key quantitative properties of the training split.

All released episodes run to the $10{,}000$ step cap; archetype D (balanced) under represents the failure modes that drive earlier termination, but with the maintenance policy in place, even the heavy

Table 15: Base scenario released training split: summary statistics.

Property	Value
Episodes	400
Total timesteps	4×10^6
Episode length	10,000 steps (all episodes at the cap)
Archetype counts (A / B / C / D)	121 / 104 / 102 / 73
NaN or Inf values in states	0
NaN or Inf values in sensors	0
Non zero maintenance rate	0.04% of timesteps
Stochastic event rate	$<1\%$ of timesteps
Sensor observation shape	$(N, 7, 12)$

archetypes A–C are kept within airworthiness limits and reach the maximum length. Users who want longer trajectories for forecasting can produce them with the same configuration.

To characterise the separation between health regimes in the hidden state space, we assign a binary health label to each timestep: a timestep is *healthy* if it falls within the first 300 steps of an episode or within 300 steps of a non-zero maintenance action, and *degraded* if it lies beyond 55% of the episode lifetime and is not covered by a healthy window. Figure 9 shows the resulting overlapping marginal distributions for all ten state dimensions, with Cohen's d reported per dimension. Separation is pronounced for HPC and turbine efficiency and flow dimensions ($d \approx 1.5$ to 2.0), confirming that the hidden state carries a strong and consistent signal distinguishing healthy from worn engines. The bleed_error and nac_drag dimensions exhibit moderate overlap, reflecting their weaker coupling to the dominant failure modes in archetypes A through C.

D Architecture and Implementation Details

This section documents the encoder, predictor, optimisation, and probe configurations used to produce every JEPA and AR LSTM number reported in Section 5.1. The hyperparameters listed here are the defaults; per run deviations are recorded in Section E.1.

D.1 Sensor Encoder

A turbofan observation $\mathbf{o}_t \in \mathbb{R}^{n_s}$ with $n_s = S \cdot P$ is the concatenation of S sensor channels read across P flight phase contexts, all reflecting the same degradation state. The base scenario has S=7, P=12, n_s=84; TurboSens enlarges the context vector to include four operating point parameters per context, so $n_s = (7 + 4) \cdot 16 = 176$. Earlier iterations of this benchmark encoded $\mathbf{o}_t$ as a grayscale image and applied a Vision Transformer (Dosovitskiy et al., 2020); we replace that pipeline with a sensor native transformer that treats each of the n_s scalars as an independent token, removing the spatial inductive bias mismatch of the image based encoder.

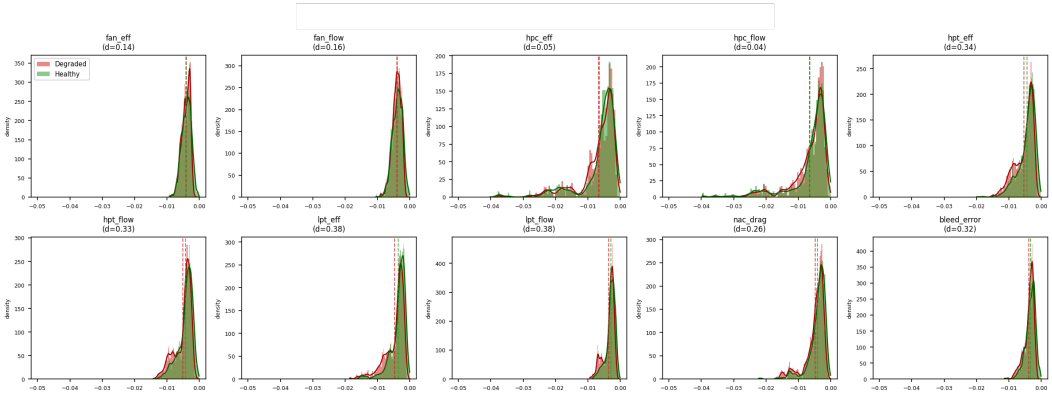
The encoder $\mathrm{Enc}_\theta : \mathbb{R}^{n_s} \to \mathbb{R}^d$ projects each scalar x_i to a d dimensional token via a shared linear layer, adds a learnable positional embedding $\mathbf{p}_i$, runs the sequence through an L layer Pre LayerNorm Transformer encoder, and mean pools across tokens:

$$\mathbf{z}_t = \mathrm{LayerNorm}\left(\frac{1}{n_s} \sum_{i=1}^{n_s} \mathrm{TF}_\theta(\mathbf{W}_v \, x_{t,i} + \mathbf{p}_i) \right), \quad \mathbf{z}_t \in \mathbb{R}^d. \tag{21}$$

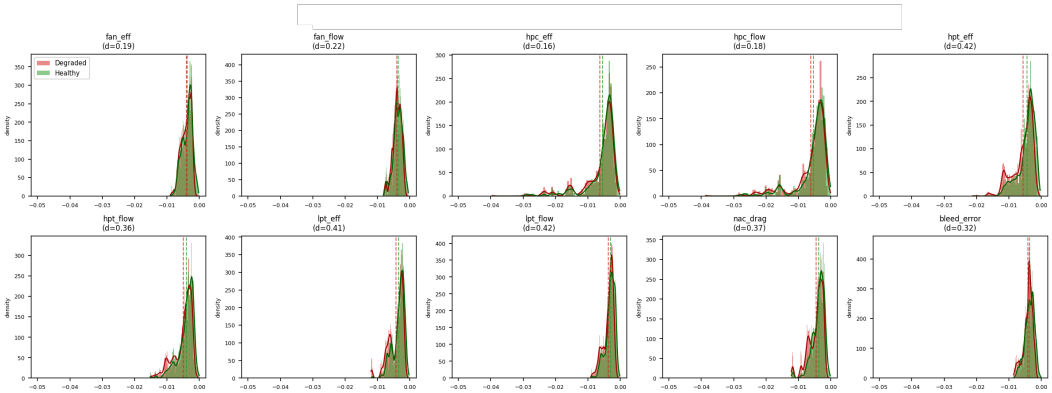
The encoder output dimension d=64 is matched to the world model latent dimension so the embedding is fed to the predictor without an extra projection bottleneck. Table 16 summarises the configuration.

D.2 Temporal Aggregator

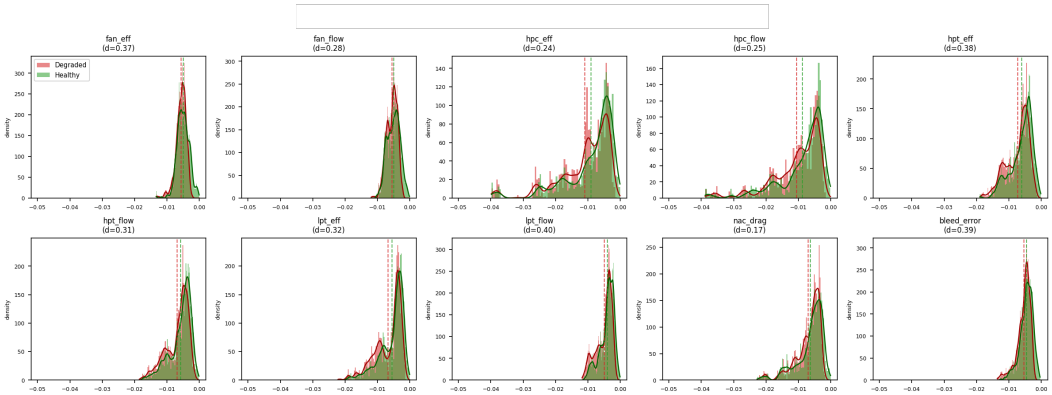
The encoder in (21) reads a single snapshot, so it cannot directly express degradation rate or trend direction. When the encoder is configured with a temporal window $w > 1$, each of the w consecutive frames $\mathbf{o}_{t-w+1}, \ldots, \mathbf{o}_t$ is encoded independently with shared weights, and a lightweight Temporal Aggregator fuses the per frame embeddings into one trajectory aware embedding $\mathbf{z}_t^{\mathrm{agg}} \in \mathbb{R}^d$. The



(a) Train set underlying state distribution between healthy (green) and degraded (red).



(b) Test set in-distribution with train, between healthy (green) and degraded (red).



(c) Test-hard set out-of-distribution, between healthy (green) and degraded (red). The distribution shift is visible, but remains within an operable envelope.

Figure 9: Distribution for the base scenario of the ten health-state components under healthy (blue) and degraded (orange) regimes. Train and test share the same distribution; test-hard introduces stronger degradation, varied drift rates, and modified maintenance to assess generalization beyond the training manifold.

Table 16: Sensor encoder configuration.

Hyperparameter	Value
Input tokens n_s	84 (base scenario) or 176 (TurboSens)
Embedding dimension d	64
Attention heads	4
Transformer layers L	2
Feed forward multiplier	4×
Normalisation	Pre LayerNorm
Dropout	0.1
Positional encoding	Learnable
Pooling	Mean over tokens

aggregator is a single layer transformer encoder (d=64, 4 heads) with a learnable positional code per slot; the output at the last position attends over the full window and is used as the aggregated embedding:

$$\mathbf{z}_t^{\text{agg}} = \text{TempAgg}(\mathbf{z}_{t-w+1}, \ldots, \mathbf{z}_t)\,. \tag{22}$$

Setting w=1 recovers the snapshot baseline; the parameter overhead of the aggregator is $\mathcal{O}(d^2)$, negligible relative to the predictor.

D.3 World Model and Loss

The world model follows the Joint Embedding Predictive Architecture (JEPA, LeWM) (Balestriero & LeCun, 2025; Maes et al., 2026). Given a context window of H consecutive embeddings $\mathbf{z}_{t-H+1}^{\text{agg}}, \ldots, \mathbf{z}_t^{\text{agg}}$ and the corresponding action embeddings, an autoregressive predictor g_ψ outputs the next latent $\hat{\mathbf{z}}_{t+1}^{\text{agg}}$. Training combines a latent prediction loss with a SIGReg term (Balestriero & LeCun, 2025) that encourages an isotropic Gaussian latent and prevents collapse:

$$\mathcal{L} = \underbrace{\left\| g_\psi(\mathbf{z}_{t-H:t}^{\text{agg}},\, \mathbf{a}_{t-H:t}) - \mathbf{z}_{t+1}^{\text{agg}} \right\|_2^2}_{\text{prediction loss}} \;+\; \lambda \cdot \mathcal{L}_{\text{SIGReg}}. \tag{23}$$

D.4 Predictor and AR LSTM Baseline

The JEPA predictor g_ψ is a causal transformer (depth 4, 8 heads, MLP dimension 512, head dimension 32, dropout 0.1) that ingests the history of H context embeddings and their action embeddings and emits one next step latent. A two layer MLP projector (BatchNorm) maps the predictor output back to the shared d=64 latent.

The AR LSTM baseline shares the sensor encoder and the temporal aggregator and replaces only the latent dynamics: the transformer predictor is swapped for a two layer LSTM (hidden size 256, dropout 0.1). All other components, including the SIGReg term and the projector, are identical, so any difference at evaluation time isolates the predictor family. The LSTM predictor has 1.14M parameters, comparable to the JEPA predictor.

D.5 Training Setup

All models train with AdamW (lr=7×10^{-5}, weight decay 10^{-3}) under a linear warmup followed by cosine annealing, gradient clipping at global norm 0.3, and an effective batch size of 512 (achieved either as a native batch of 512 or as a smaller native batch with gradient accumulation; the effective batch is held constant across runs). The SIGReg regulariser uses λ=0.15, 17 knots, 1024 random projections.

Epoch budget. JEPA configurations are trained for 10 epochs, sufficient for the latent prediction loss to plateau on TurboSens at the effective batch size of 512; longer runs do not change the probe ranking. The AR LSTM baseline does not have to share the same epoch budget because its predictor family converges differently; we report it under a comparable wall budget for fairness, with per run details in Section E.1.

Hardware. Each run uses a single GPU, either an RTX 5090 or an H100. The hardware choice does not affect the reported numbers; differences between runs are explained by the hyperparameters listed in Section E.1.

Table 17: Default training hyperparameters used across JEPA and AR LSTM runs.

Hyperparameter	Value
Optimiser	AdamW
Learning rate	7×10^{-5}
Weight decay	10^{-3}
LR schedule	Linear warmup + cosine annealing
Gradient clip (global norm)	0.3
Effective batch size	512
Epochs (JEPA)	10
SIGReg weight λ	0.15
SIGReg knots / projections	17 / 1024
Predictor history H	$\{8, 16, 32\}$
Encoder window w	$\{1, 5, 10\}$
Predictor stride S	$\{1, 5, 10, 20\}$

D.6 Downstream Evaluation Probes

After training, the encoder and the temporal aggregator (when $w > 1$) are frozen, and small heads are trained on the frozen embeddings to expose what the world model representation actually captures. The probes use the same train and test splits as the world model, but they see the ground truth target as a supervised signal that the world model itself never accesses during pretraining.

Health state probe. The main probe is a linear ridge regression from the frozen embedding to the ten dimensional ground truth state $\mathbf{s}_t$: closed form, deterministic, with the regularization constant chosen by cross validation grouped by training episodes and per component standardized targets. We report pooled and per component Pearson, RMSE, and R^2. Every individual run carries a 95 percent confidence interval from 2000 bootstrap rounds that each redraw the evaluation episodes with replacement and recompute the score; timesteps are never resampled, as they are correlated within an episode. The default probe input is a single frozen embedding, which isolates what the encoder alone has captured; for recurrent world models the belief state $\mathbf{b}_t$ (the predictor's hidden state rolled over embeddings and actions) is probed with the identical pipeline and reported separately.

Action timing probe. A second, optional probe predicts the number of steps to the next maintenance action, $\tau_t = \min(\text{steps to next maintenance}, \tau_{\text{max}})$, with τ_{max}=300. Targets are $\log(1 + \tau_t)$ transformed before standardisation; metrics are computed on uncensored timesteps only. Across all reported encoders the linear probe Pearson is at or below 0.03, indicating that 10 epoch self supervised training does not expose maintenance timing structure to a linear readout. We report this probe as a placeholder for future work that should explicitly handle event sparsity.

What each temporal hyperparameter controls. The encoder window w, the predictor history H, and the probe input length `seq_len` are independent design knobs whose effects are summarised in Table 18.

E Evaluation Details: Ladder, Split Validation, and Fidelity Scope

This appendix records the run inventory, the reference ladder definitions, the split validation, and the fidelity scope of the evaluation reported in Section 5.1.

Parameter	Affects encoder?	Affects predictor?	Affects probe?
w (encoder window)	Yes, depth of temporal fusion inside $\mathbf{z}_t$	Indirectly, through richer $\mathbf{z}_t$	Indirectly
H (predictor history)	No	Yes, number of past embeddings the predictor sees	No
`seq_len`	No	No	Yes, window fed to the probe head

Table 18: Effect of each temporal hyperparameter.

E.1 Run inventory

Each configuration is trained for ten epochs with three independent seeds on a single H100 GPU. Contexts are given in raw flight cycles; the engine wash interval is roughly 95 cycles, so the wash spanning configurations integrate across at least one full wash cycle.

Configuration	Family	Context (cycles)	Seeds	Wall clock / seed
JEPA H16	JEPA (latent loss)	16	3	$\sim$10 h
JEPA H32	JEPA (latent loss)	32	3	$\sim$12 h
JEPA H16 S10	JEPA (latent loss)	160 (wash spanning)	3	$\sim$30 h
LSTM H16	AR LSTM (latent loss)	16	3	$\sim$3.5 h
LSTM H32 W4	AR LSTM (latent loss)	32, window 4	3	$\sim$7 h
LSTM H160	AR LSTM (latent loss)	164 (wash spanning)	3	$\sim$27 h
RSSM T64	RSSM (reconstruction)	64	2	$\sim$6.5 h
Supervised oracle	shared backbone, supervised	1	3	$<$1 h
Direct supervised MLP	MLP on raw obs, supervised	1	3	$<$0.5 h

Table 19: Run inventory of the evaluation campaign. One RSSM seed exceeded its walltime allocation, so the RSSM has two trained seeds; its belief state readout in Table 2 is reported for one of them. Direct probes (raw ridge, sensor mean, action history) require no pretraining and are deterministic.

E.2 Reference ladder definitions

The probe W is always the ridge of Appendix D.6.

- **Direct supervised MLP**: $\hat{\mathbf{s}}_t = g(\mathbf{o}_t)$ with g an MLP (under half a million parameters) trained by supervised regression on the training episodes. Measures how much state a single observation contains at all.
- **Supervised oracle**: $\hat{\mathbf{s}}_t = W f_{\text{sup}}(\mathbf{o}_t)$ where f_{sup} is the identical encoder architecture the SSL models use, trained from scratch with supervision (no SSL weights). The SSL to oracle gap is attributable to the pretext task.
- **SSL world models**: $\hat{\mathbf{s}}_t = W f_{\text{ssl}}(\mathbf{o}_t)$ with f_{ssl} frozen after pretraining; for recurrent models also $\hat{\mathbf{s}}_t = W \mathbf{b}_t$ on the belief state.
- **Raw sensor ridge**: $\hat{\mathbf{s}}_t = W \mathbf{o}_t$. Linear extractability with no pretrained model.
- **Sensor mean**: $\hat{\mathbf{s}}_t = W m(\mathbf{o}_t)$, where m averages each of the 7 sensors over the 16 operating points. The practitioner floor.
- **Action history only**: $\hat{\mathbf{s}}_t = W h(a_{1:t})$, defined below. Measures how much the maintenance policy alone reveals about the state.

Per seed pooled Pearson on `test` (transfer probe): JEPA H16 $0.634/0.752/0.618$; JEPA H32 $0.753/0.629/0.617$; JEPA H16 S10 $0.770/0.770/0.655$; LSTM H16 $0.636/0.634/0.698$; LSTM H32 W4 $0.625/0.632/0.630$; LSTM H160 $0.636/0.636/0.631$; LSTM H160 belief $0.691/0.699/0.693$; RSSM belief 0.045 (one seed); oracle $0.904/0.916/0.912$; direct MLP $0.965/0.963/0.961$. Every individual run additionally carries a 95 percent episode bootstrap confidence interval (Appendix D.6); as one example, JEPA H16 seed 0 scores 0.634 $[0.573, 0.688]$

while seed 1 scores 0.752 $[0.704, 0.795]$, non overlapping intervals that establish the seed instability reported in Section 5.1 as real run to run variation.

E.3 WASH STRATA AND THE PRE WASH STRATUM

Table 20 reports state recovery on `test` in the freshest wash stratum (0 to 20 flights since the last wash), the oldest anchored stratum (100 to 200 flights), and the stratum before an episode's first wash. The masking slope of Table 2 is the difference between the first two columns. The pre wash stratum is not a masking diagnostic: its label variance is one third of the other strata, its fouling sits in the middle of the range, and no wash anchor exists there, so its lower scores largely reflect variance geometry.

Table 20: State recovery by wash stratum on `test` (pooled mean Pearson, mean $\pm$ sd over seeds). †Two seeds.

Representation	0 to 20 flights	100 to 200 flights	before first wash
MLP on raw observation	0.970 ± 0.001	0.943 ± 0.002	0.834 ± 0.001
Oracle (matched backbone)	0.919 ± 0.005	0.882 ± 0.010	0.584 ± 0.013
Oracle + wash anchor	0.922 ± 0.005	0.897 ± 0.008	0.561 ± 0.006
JEPA H16	0.687 ± 0.070	0.624 ± 0.068	0.245 ± 0.036
JEPA H32	0.685 ± 0.074	0.630 ± 0.065	0.262 ± 0.041
JEPA H16 S10 (wash span)	0.746 ± 0.063	0.688 ± 0.075	0.295 ± 0.071
LSTM H16	0.675 ± 0.040	0.617 ± 0.034	0.238 ± 0.016
LSTM H32 W4	0.647 ± 0.003	0.596 ± 0.005	0.183 ± 0.032
LSTM H160 (wash span)	0.650 ± 0.002	0.600 ± 0.002	0.242 ± 0.010
LSTM H16 belief	0.704 ± 0.041	0.657 ± 0.036	0.263 ± 0.014
LSTM H160 belief	0.688 ± 0.003	0.652 ± 0.003	0.279 ± 0.012
LSTM H16 + wash anchor	0.682 ± 0.039	0.653 ± 0.035	0.300 ± 0.016
JEPA H16 S10 + wash anchor	0.747 ± 0.059	0.716 ± 0.065	0.341 ± 0.073

E.4 NONLINEAR PROBE

To check whether the gap in Table 2 comes from the linear readout, we refit the probe on the same frozen embeddings with a two layer MLP (256 units per layer). Table 21 compares the two probes. The MLP barely moves the oracle and adds 0.04 to 0.08 on `test` to every self supervised representation, with smaller gains under the composition shift. The masking slope grows for every self supervised representation: the MLP recovers more of the state where the sensors expose the wear, not where fouling masks it. An MLP on the raw observation keeps a slope of 0.027, so the masked state is present in the observation but not in these embeddings.

Table 21: Linear (ridge) versus two layer MLP probe on the same frozen embeddings. Pooled mean Pearson, mean $\pm$ sd over three pretraining seeds; masking slope as in Table 2.

Representation	Probe	test	test_shift	test_region	masking slope
Oracle (matched backbone)	ridge	0.911 ± 0.006	0.842 ± 0.006	0.882 ± 0.003	0.038
	MLP	0.912 ± 0.007	0.842 ± 0.009	0.884 ± 0.003	0.037
JEPA H16 S10 (wash span)	ridge	0.732 ± 0.067	0.627 ± 0.044	0.699 ± 0.053	0.059
	MLP	0.775 ± 0.030	0.631 ± 0.028	0.723 ± 0.028	0.077
JEPA H16	ridge	0.668 ± 0.073	0.579 ± 0.046	0.649 ± 0.051	0.062
	MLP	0.744 ± 0.027	0.619 ± 0.023	0.707 ± 0.013	0.079
LSTM H16	ridge	0.656 ± 0.036	0.572 ± 0.023	0.644 ± 0.013	0.058
	MLP	0.733 ± 0.033	0.624 ± 0.009	0.699 ± 0.025	0.083
LSTM H160 (wash span)	ridge	0.634 ± 0.003	0.553 ± 0.003	0.639 ± 0.006	0.050
	MLP	0.689 ± 0.041	0.590 ± 0.026	0.660 ± 0.030	0.068
LSTM H160 belief	ridge	0.695 ± 0.004	0.615 ± 0.006	0.686 ± 0.006	0.036
	MLP	0.778 ± 0.006	0.660 ± 0.014	0.734 ± 0.008	0.057

E.5 ACTION HISTORY BASELINE FEATURES

The action history probe receives fourteen strictly causal features per timestep: the number of cycles since each of the ten components was last serviced (derived from the per component service masks), cycles since the last engine wash, the cumulative wash count, the cumulative repair count, and the elapsed cycles in the episode. All counters reset at episode start and are $\log(1 + x)$ transformed: maintenance recency effects are fast then slowing, so log time linearizes them for the ridge probe, compresses the heavy tailed counters, and handles the zero of a same cycle service. The baseline reaches pooled Pearson 0.432 on `test`, 0.431 on `test_shift`, and 0.442 on `test_region`, far below the sensor mean baseline (0.694): the maintenance policy alone is a weak predictor of the state.

E.6 SPLIT VALIDATION

All split level claims rest on episode level two sample analyses (episodes are the independent sampling unit). A second in distribution draw is released as a `replication` split; it is statistically indistinguishable from `test` on most state components and carries no OOD claim. `test_region` (Arid and Industrial archetypes held out) shifts four to six of the ten state components significantly at the episode level (Mann–Whitney with Benjamini–Hochberg correction at $p < 0.05$, Wasserstein distances up to 0.87 reference standard deviations). `test_shift` raises the extreme degradation fraction from about 5 percent to 35 percent while its sensor marginals remain statistically indistinguishable from `test`; the shift manifests as stratified degradation (supervised reference 0.901 on nominal versus 0.780 on extreme episodes) rather than as a marginal shift, isolating a pure fleet composition axis. The two axes behave differently in an informative way: direct probes on the raw observation barely move under the region shift (0.930 to 0.919), the oracle and the JEPA encoders lose 0.02 to 0.03, and the LSTM encoders are essentially unchanged, so that axis stresses what models learned rather than the observation mapping itself.

E.7 FIDELITY SCOPE

Table 22 states the fidelity level of each simulator layer relative to its reference. Fidelity bounds how far conclusions may be carried; the validity of the probing measurement itself is exact by construction, because the ground truth state generates every observation. No operational conclusion about real engines may be drawn from TurboSens.

Layer	What the simulator does	Fidelity level	Reference
Observation map	Sensor responses computed at 16 operating points per cycle	Model faithful	OpenDeckSMR station thermodynamics
Degradation profile	Fast then slowing wear per module	Shape matched, rates uncalibrated	Qualitative shape reported in practice
Fouling and wash	Accumulation cleared by washes, per engine coupling	Shape matched, rates uncalibrated	Compressor fouling asymmetry
Maintenance policy	Health correlated heuristic scheduler, six visit types	Heuristic structure	Condition based scheduling
Stochastic events	Parameterized distributions, modular library	Statistical abstraction	By design, extensible
Fleet priors	Region and archetype priors, shift splits	Statistical abstraction	Designed for validated shift axes

Table 22: Fidelity scope of the simulator, stated as a hierarchy relative to each layer's reference rather than as a binary.

E.8 COUNTERFACTUAL FIDELITY CURVES

Figure 10 reports the per checkpoint and per action curves underlying the counterfactual fidelity summary in Section 5.1. The top row plots absolute mean HI RMSE between the world model and the simulator under the same overridden action; the bottom row plots the differential $|\Delta_{\text{model}}(a, 0) -$

$\Delta_{\text{sim}}(a, 0)|$, which subtracts the *do nothing* baseline from both rollouts and isolates the action specific response. Curves are means over thirty episodes and three branch positions per checkpoint, with a one standard deviation band around the mean.

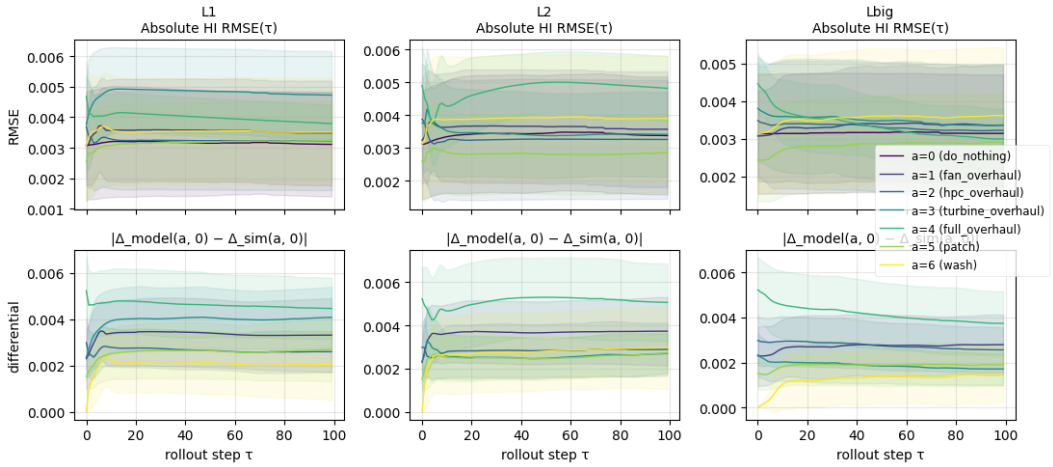


Figure 10: Counterfactual fidelity per checkpoint and per action. Top row: absolute HI RMSE between world model and simulator under the same overridden action a. Bottom row: differential $|\Delta_{\text{model}}(a, 0) - \Delta_{\text{sim}}(a, 0)|$, isolating the response to the action change against the *do nothing* baseline.

F Auxiliary Evaluation Tasks

The benchmark ships with two further tasks that are not the focus of the present paper but are useful for stress testing world model representations in future work. The data and metric infrastructure for both tasks is included in the released evaluation code.

F.1 Degradation velocity estimation

Rather than predicting the absolute health state, a velocity probe asks whether the encoder captures the local rate of change of degradation. The target is $\Delta \mathbf{s}_t = \mathbf{s}_t - \mathbf{s}_{t-1}$ within episode boundaries and restricted to non maintenance timesteps. Predicting $\Delta \mathbf{s}_t$ from a single snapshot is strictly harder than predicting $\mathbf{s}_t$: it requires the encoder to capture local curvature of the degradation manifold. A maintenance alarm score is defined by thresholding $\|\widehat{\Delta \mathbf{s}}_t\|_2$ and evaluated at horizons $K \in \{10, 20, 50, 100\}$ as the AUC of the binary event "maintenance occurs within K steps".

F.2 Deterministic replay and counterfactual API

The `scenarios.scenario4.replay.EpisodeReplayer` class is a pure python entry point around the simulator. It takes a generated HDF5 file, reads the `ep_meta/seed`, region and `is_extreme` flag of any episode, and either replays the full trajectory bit identically or forks it at a chosen branch point. The data class `CounterfactualSpec` exposes four levers: `branch_t` (where to fork), `override_action` (closed loop, the expert maintenance policy resumes after the forced action), `forced_actions` (open loop, a fixed sequence of actions is applied after `branch_t`), and `weather_seed` + `weather_overrides` (re seed the seasonal OU). Bit identity is enforced by a regression test (`test/scenario4/test_replay_bit_identity.py`) and the schema requirement is documented in the file's docstring; the source tree currently lives on the `nodegpu` branch.

F.3 LATENT FORECASTING WITH ACTION DIVERGENCE GAP

The world model's latent predictor is evaluated directly by rolling out τ steps autoregressively from an initial context window and decoding each predicted embedding $\hat{\mathbf{z}}_{t+\tau}$ to HI space using the Task 1 probe (no additional training). Rollouts use zero actions; the world model propagates the current degradation trajectory without any maintenance intervention. To quantify the cost of ignoring maintenance, each τ step trajectory is split into clean segments (no maintenance event in $(t, t+\tau]$) and event segments (at least one maintenance event); we report mean HI RMSE on both and their difference $\delta(\tau) = \mathrm{RMSE}_{\mathrm{event}}(\tau) - \mathrm{RMSE}_{\mathrm{clean}}(\tau)$, the action divergence gap. A large gap at short horizons indicates that maintenance transitions are the primary source of forecast error and motivates action conditioned world modelling. The action conditioned counterfactual rollout that closes this gap is the camera ready experimental target referenced in Section 5.1.

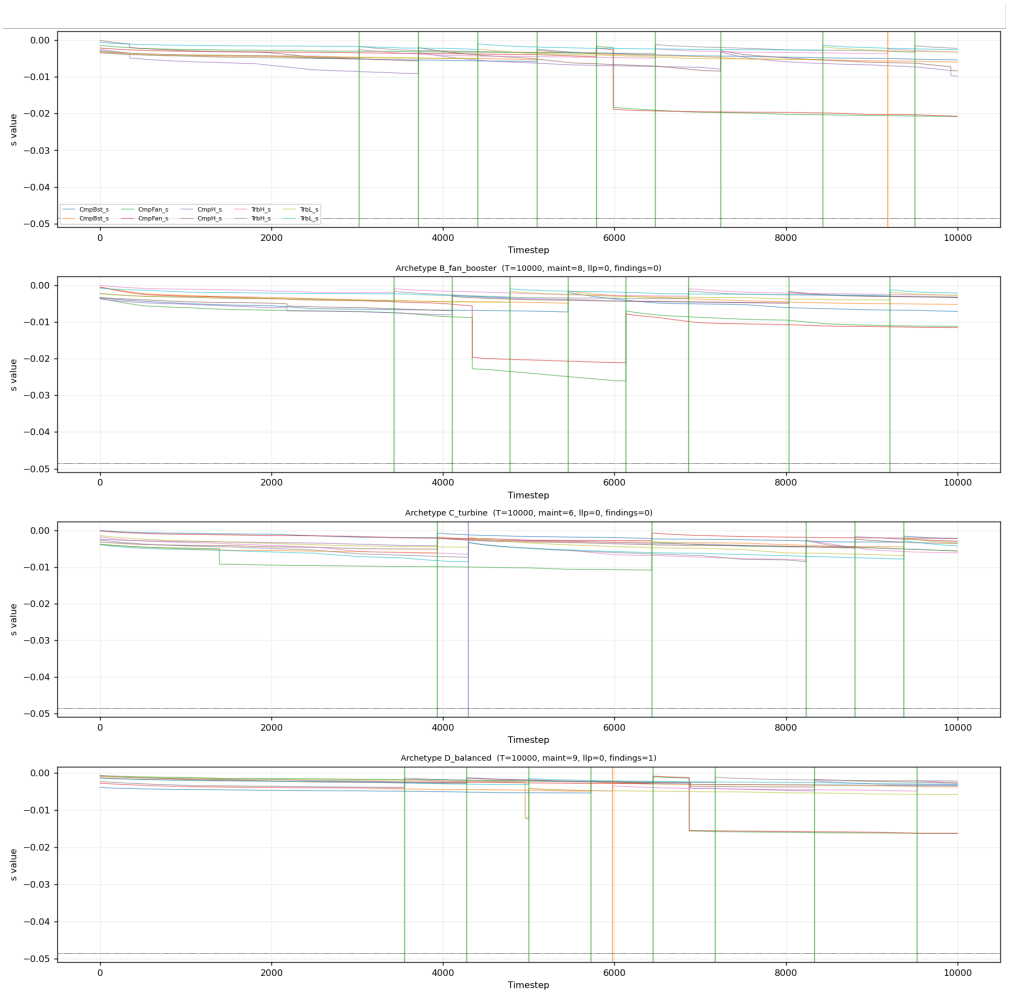


Figure 11: Base scenario hidden state trajectories under the four degradation archetypes (A compressor, B fan/booster, C turbine, D balanced). Each panel overlays the ten state components for a single representative episode; vertical green and orange lines mark maintenance interventions. Archetype D produces nearly flat trajectories that survive the full $10{,}000$ step horizon without intervention, while A and C show the steeper early drift that drives recurring maintenance.

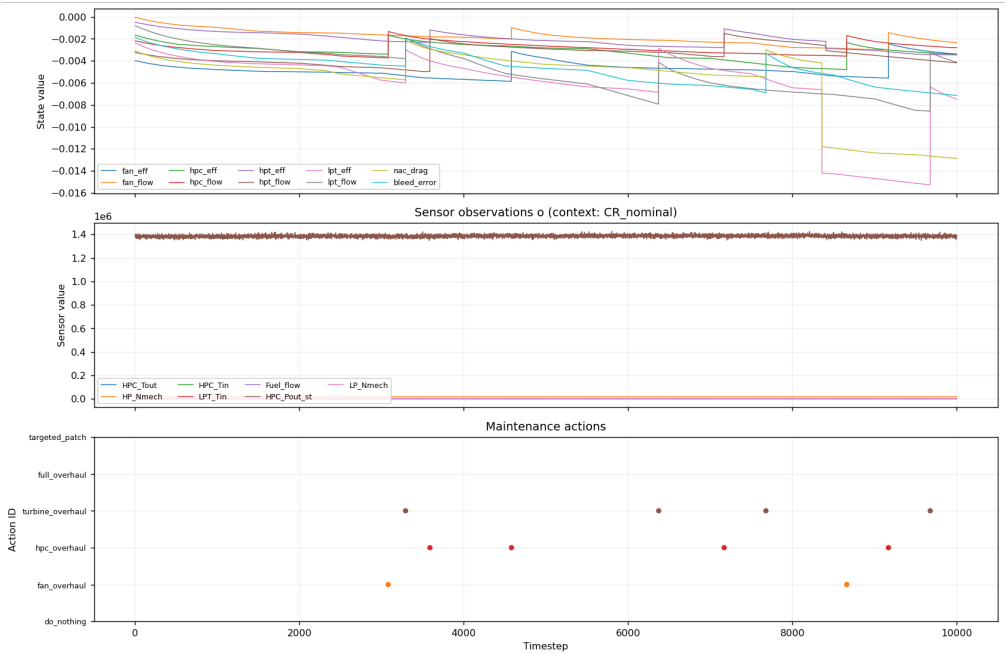


Figure 12: Single base scenario episode (episode 7, archetype C, observed under the `CR_nominal` flight context). Top: the ten dimensional hidden state $\mathbf{s}_t$ evolving as a sawtooth of partial resets. Middle: the seven dimensional sensor stream $\mathbf{o}_t$ at the chosen context, showing how observation noise and contextual variation obscure the underlying state. Bottom: the discrete maintenance action sequence; coloured markers indicate the action type at each maintenance event.

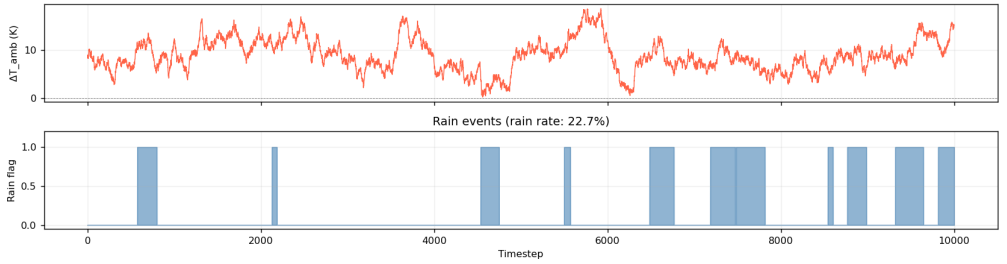


Figure 13: Weather process for a Single base scenario episode: the ambient temperature deviation ΔT_{amb} follows a slow Ornstein-Uhlenbeck trajectory (top) and rain events (bottom) are gated by the seasonal phase, here at a 22.7% episode wide rain rate.

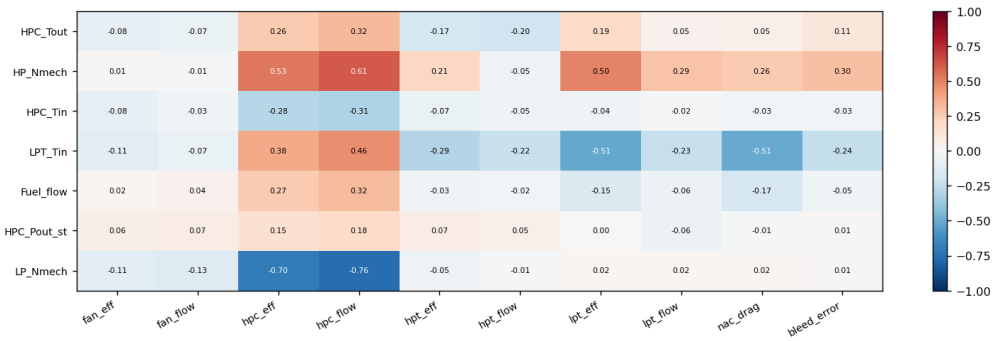


Figure 14: Base scenario sensor / state correlation matrix. Context averaged Pearson correlations between the seven sensor channels and the ten hidden state dimensions, computed over the train split. Most cells are near zero, but a handful of strong positive (HP shaft speed vs HPC efficiency / flow, $r \in [0.5, 0.6]$) and strong negative (LP shaft speed vs HPC flow, $r = -0.76$) couplings establish that the observation map is informative about the latent state but far from a one-to-one decoder; this is the structure the inverse evaluation protocol asks the encoder to recover.

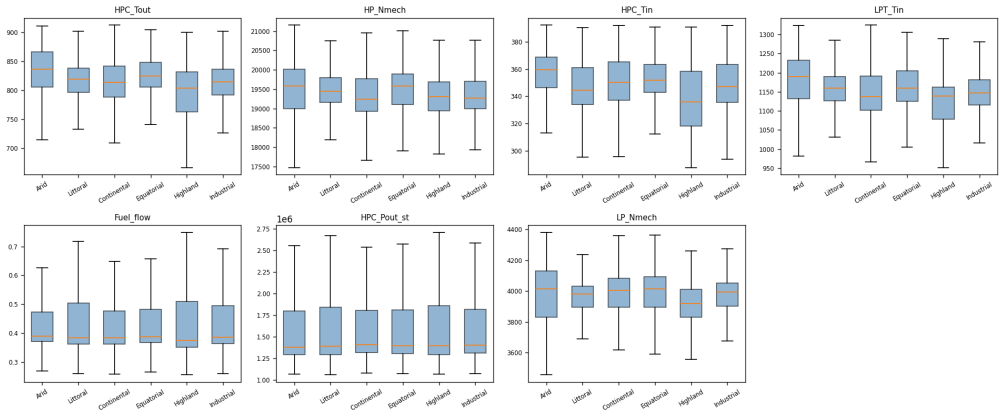


Figure 15: TurboSens per region sensor distributions, pooled over all timesteps and contexts. Each panel shows a box plot of one of the seven sensor channels across the six geographic regions (Arid, Littoral, Continental, Equatorial, Highland, Industrial). Region affects both the median and the spread of every channel, most noticeably on HPC inlet and outlet temperatures (HPC_Tin, HPC_Tout) where the Highland and Arid extremes are visibly separated. The OOD evaluation in Section 5 relies on held out region / archetype draws sitting in these distributional tails.

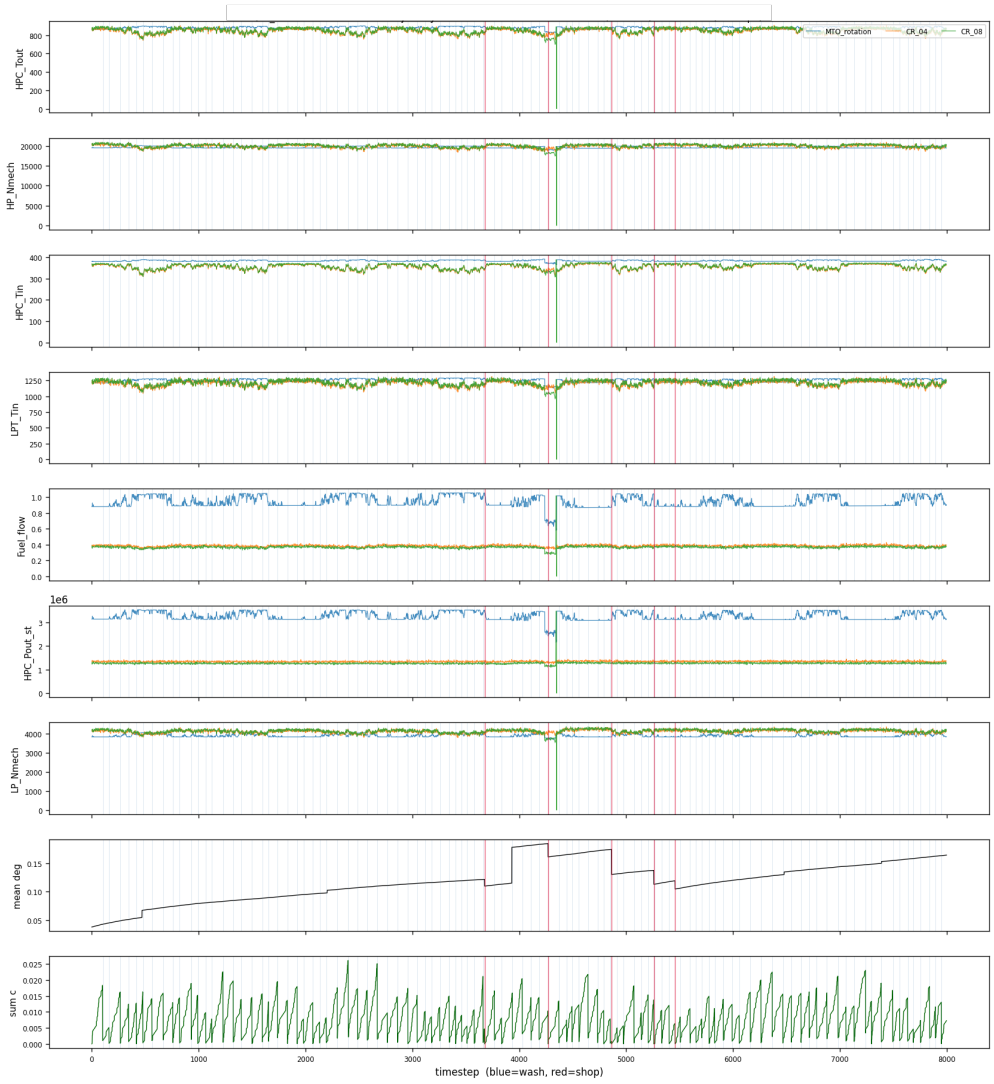


Figure 16: Full trajectory of a TurboSens episode in the Arid region (archetype C, $T \approx 8000$ steps, ~ 100 engine washes, 5 shop visits). Each row shows one of the seven sensors over the full episode. Vertical red and orange markers indicate shop visits and engine washes respectively; the dramatic dips in HP_Nmech and HPC_Pout_st around the shop visits are visible directly in the sensor stream. The bottom two rows show the calendar month and a fouling proxy, illustrating how often the wash action is exercised in a high fouling region.