FROM LATENT REPRESENTATIONS TO CAUSAL HEALTH STATES: DEGRADATION-AWARE GEOMETRY FOR REMAINING USEFUL LIFE PREDICTION

Anonymous authors

Paper under double-blind review

ABSTRACT

Remaining Useful Life (RUL) prediction is commonly formulated as direct regression from condition-monitoring signals to a scalar lifetime estimate. While increasingly powerful models achieve strong predictive performance, this formulation does not explicitly ensure that their learned representations reflect the underlying degradation process. We introduce Health-Aware Manifold Learning (HAML), a new representation-backend-portable framework that constructs an explicit degradation geometry over latent representations before RUL prediction. HAML first encodes causal sensor windows into a low-dimensional representation without RUL or lifecycle-progress supervision. Using training data only, lifecycle progress guides the construction of a sparse degradation-aware landmark graph that penalises cross-stage shortcuts through a progress-aware graph metric. Distributed healthy and failure regions define a graph-geodesic health coordinate, which a persistent level-velocity filter transforms into a bounded, non-decreasing Health Index (HI) using only observations available up to the current cycle. The resulting health dynamics and complementary manifold geometry support downstream RUL estimation. Experiments on all four C-MAPSS aircraft-engine subsets and the PHM12 bearing dataset demonstrate competitive RUL prediction, meaningful lifecycle alignment, and favourable late-life error behaviour, while controlled C-MAPSS sensor perturbations demonstrate robustness to test-time noise. Controlled ablations clarify the roles of individual components, while alternative representation backends demonstrate HAML's portability beyond UMAP. These results show that explicitly modelling degradation geometry and its temporal evolution offers a principled alternative to unconstrained sensor-to-RUL regression.

1 INTRODUCTION

Remaining Useful Life (RUL) prediction is central to Prognostics and Health Management (PHM), supporting condition-based maintenance and safer operation of engineering assets (Zio, 2022; Lei et al., 2018). Data-driven models based on recurrent, convolutional, Transformer, and graph architectures have achieved strong predictive performance (Lei et al., 2018; Wang et al., 2025b). However, direct sensor-to-RUL optimisation does not require the learned representation to form a coherent, lifecycle-consistent degradation state, particularly under heterogeneous operating conditions and sensor noise. Moreover, Euclidean proximity in a learned representation may not faithfully reflect intrinsic nonlinear relationships (Bengio et al., 2013; Tenenbaum et al., 2000), allowing lifecycle-distant states to remain close in the latent space. This motivates the explicit construction of degradation geometry rather than relying solely on representations optimised for predictive performance.

We therefore propose Health-Aware Manifold Learning (HAML), which separates representation learning from degradation-aware metric construction. A replaceable backend is fitted without lifecycle-progress or RUL supervision; training-only lifecycle progress then shapes a sparse landmark graph whose distributed healthy/failure regions define a graph-geodesic health coordinate. A persistent level–velocity filter converts this coordinate into a non-decreasing HI, from which RUL

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.

is inferred together with complementary geometric information. Throughout this paper, causal denotes temporal/online causality: prediction at cycle *t* uses no observations after *t*, and does not refer to causal-effect identification.

Unlike previous manifold- or representation-based prognostic methods, in which degradation information determines the representation or directly supervises an HI (Benkedjouh et al., 2013; Wang et al., 2021; Yu et al., 2019; Wei et al., 2021), HAML introduces lifecycle supervision after representation fitting. The resulting health coordinate is therefore neither a particular latent dimension nor a directly regressed HI, but a relative graph-geodesic position between healthy and failure regions under a degradation-aware metric. Thus, HAML acts as a post-representation geometric layer that converts heterogeneous latent representations into explicit causal degradation states.

Our contributions are threefold: (i) **Post-representation degradation-aware geometry.** We separate representation fitting from degradation supervision and introduce a progress-aware landmark graph in which distributed healthy/failure regions define a graph-geodesic health coordinate, (ii) **Geometric-to-causal degradation-state estimation.** We introduce a persistent level–velocity filter that converts the geometric health observation into a temporally causal, bounded, non-decreasing HI while retaining degradation momentum. We establish structural and stability properties covering cross-stage shortcut suppression, out-of-sample coordinate stability, reference-distance approximation, and causal monotonic state evolution, and (iii) **Component-level empirical validation and backend portability.** Across C-MAPSS and PHM12, we evaluate predictive performance and degradation-state quality; controlled C-MAPSS experiments further examine graph geometry, reference construction, causal filtering, decoder information, noise robustness, and alternative representation backends.

2 Related work

RUL approaches broadly include physics-based models (PBMs) and data-driven methods (DDMs). PBMs provide physically interpretable degradation models but require detailed system knowledge and can be difficult to generalise (Zheng et al., 2024; Yu et al., 2021). DDMs learn directly from monitoring signals using CNNs (Li et al., 2018), recurrent models (Xiang et al., 2020; Shang et al., 2022), GNNs (Liu et al., 2023; Wang et al., 2023; 2024), attention models (Vu et al., 2024), and recent time-series architectures (Wu et al., 2023; Wang et al., 2025a; Nie et al., 2023; Liu et al., 2024). Unlike sensor-level GNNs (Liu et al., 2023; Wang et al., 2023; 2024), HAML uses a graph to define a degradation-aware metric over latent asset states rather than for sensor-level message passing.

Alternative representations expose temporal, spectral, or geometric degradation structure, including phase-space image representations (Vu et al., 2026) and transfer from pretrained vision models (Chen et al., 2025; Siru et al., 2025). These studies motivate exploiting degradation geometry beyond direct correlations among observations.

Most closely related are manifold-based degradation representations (Benkedjouh et al., 2013; Wang et al., 2021) and learned HIs based on autoencoding or latent-variable models (Yu et al., 2019; Wei et al., 2021). HAML differs in where degradation supervision enters: the representation is fitted independently, after which lifecycle progress modifies graph traversal costs and defines endpoint regions. Health is therefore measured over a retained representation before temporal regularisation.

3 Method

3.1 Temporal Descriptor Encoder

In this study, we consider run-to-failure training trajectories from a population of assets. For asset i, let $\mathbf{x}_{i,t} \in \mathbb{R}^F$ denote the multivariate sensor observation at cycle t. A causal sensor window of length W ending at t is $\mathbf{X}_{i,t} = [\mathbf{x}_{i,t-W+1}, \ldots, \mathbf{x}_{i,t}] \in \mathbb{R}^{W\times F}$, where training samples are constructed only from fully observed windows of length W. For C-MAPSS, the capped training target is $y_{i,t} = \min\{T_i - t, R_{\max}\}$, where T_i is the failure cycle and $R_{\max}$ is the RUL cap (Wang et al., 2025b; Saxena et al., 2008); PHM12 uses the bearing-normalised target defined in Appendix

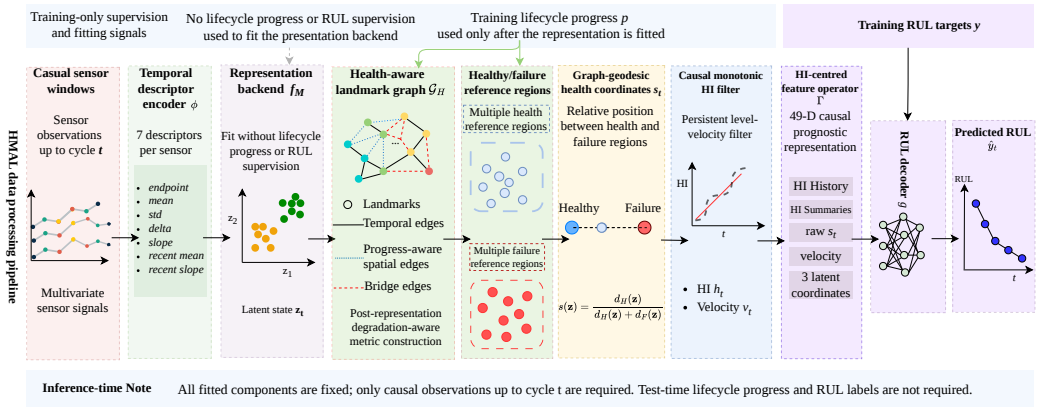

Figure 1: **General HAML framework.** The representation backend is fitted without lifecycle-progress or RUL supervision; training lifecycle progress subsequently shapes the health graph and reference regions, and RUL targets train the final decoder. At inference, all fitted components are fixed and only causal sensor observations are required.

D (Nectoux et al., 2012; Wei et al., 2023). For training trajectories, we define normalised degradation progress at valid window endpoints as:

$$p_{i,t} = (t - t_i^{(0)})/(T_i - t_i^{(0)}) \tag{1}$$

where $t_i^{(0)}$ is the first endpoint admitting a complete training window. The progress variable is training-only supervision for the health-geometry stage: it is used to shape the health-aware graph and define healthy/failure reference regions, but it is not required for test-time prediction. Rather than flattening a raw window, HAML converts each sensor channel into seven interpretable temporal descriptors: endpoint value, mean, standard deviation, endpoint-to-start change, full-window least-squares slope, recent mean, and recent slope. For sensor k, an input window $\mathbf{X}$ is encoded into a vector of seven descriptors $[q_k^{\text{last}}, \mu_k, \sigma_k, \Delta_k, a_k, \mu_k^{(R_{\text{desc}})}, a_k^{(R_{\text{desc}})}]$. Concatenating all channels yields $\mathbf{q}_{i,t} = \phi(\mathbf{X}_{i,t}) \in \mathbb{R}^{7F}$. The descriptor standardizer is fitted on training windows and reused unchanged at validation and test time. Details are in Appendix B.

3.2 Health-Aware Landmark Graph

A representation backend $f_{\mathcal{M}}$ maps the standardised descriptors into a low-dimensional latent state, $\mathbf{z}_{i,t} = f_{\mathcal{M}}(\mathbf{q}_{i,t}) \in \mathbb{R}^d$, where $d \ll 7F$. Let denote the selected descriptor-space landmark set as $\mathcal{L} = \{\ell_1, \ldots, \ell_{M_L}\}$, while $\mathcal{Z}_L = \{\mathbf{z}_j = f_{\mathcal{M}}(\ell_j)\}$ denotes the latent landmark set. To separate latent representation fitting from degradation supervision, the representation backend is fitted without using lifecycle progress or RUL targets in its learning objective (McInnes et al., 2018). The descriptor-space landmarks are sampled in a round-robin manner across engines and, for multi-condition datasets, across engine–condition groups to improve population balance; lifecycle progress and RUL targets are not used for landmark selection. The fitted representation backend is then used to transform all training and test windows. Thus, degradation supervision enters only after the latent representation has been formed, during health-graph construction and reference-region definition.

Given the latent landmarks, HAML constructs an undirected weighted graph $\mathcal{G}_H = (V, E_s \cup E_t \cup E_b)$, where E_s contains mutual-k_GNN spatial edges, E_t contains temporal edges between consecutive selected states of the same training asset, and E_b contains bridge edges inserted only if needed to obtain a connected graph. For a mutual spatial neighbour pair (u, v), the edge cost is:

$$w_{uv}^{(s)} = \|\mathbf{z}_u - \mathbf{z}_v\|_2 \left(1 + \lambda_G \left|p_u - p_v\right|\right) \tag{2}$$

where $\lambda_G \geq 0$ is the health-progress penalty. This weighting increases the cost of spatial connections between lifecycle-distant states without modifying the representation backend (Theorem

1, Appendix C). The result characterises the progress-aware spatial metric and does not assert an analogous lower bound for arbitrary paths containing temporal edges.

Based on this cost, latent neighbours that are far apart in degradation progress remain connected only at a larger traversal cost, attenuating cross-stage spatial shortcuts without altering the backend itself. For consecutive selected states u, v from the same training asset, HAML adds a temporal edge $w_{uv}^{(t)} = \eta_T \tilde{d}_{uv} \sqrt{\Delta c_{uv}}$, where $\eta_T > 0$ is a temporal factor, $\Delta c_{uv} = |t_u - t_v|$ is the cycle gap, and $\tilde{d}_{uv}$ is the latent transition distance defined by:

$$\tilde{d}_{uv} = \begin{cases} d_{uv}, & \text{if } \mathbf{C}_u = \mathbf{C}_v, \\ \min\left(d_{uv},\, Q_{0.95}(\mathrm{Ord})\right), & \text{otherwise.} \end{cases} \tag{3}$$

where $d_{uv} = \|\mathbf{z}_u - \mathbf{z}_v\|_2$, c_u is the operating-condition label of landmark u, $\mathrm{Ord} = \{d_{uv} : (u,v) \in E_t,\ \mathbf{C}_u = \mathbf{C}_v\}$ is the unchanged-condition set for (u,v) in the temporal edge set, and $Q_{0.95}(\mathrm{Ord})$ is the 95th percentile of latent transition distances with the unchanged condition. The cap $\tilde{d}_{uv}$ prevents operating-condition-induced latent displacements from disproportionately inflating temporal edge costs.

If several edge mechanisms connect the same unordered pair, the minimum admissible cost is retained. Besides, if the graph is not yet connected, each smaller component will be connected to the largest component using its nearest admissible bridge, whose cost $w_{uv}^{(b)}$ follows the same progress-aware form as Eq. (2). Shortest-path distances on the connected graph define the HAML graph metric d_G.

$$d_G(u,v) = \min_{\Pi \in \mathcal{P}(u,v)} \sum_{(a,b)\in\Pi} w_{ab} \tag{4}$$

where $\mathcal{P}(u,v)$ is the set of all paths in the graph from u to v, and the cost of a path is the sum of its retained edge costs, with $w_{ab} = \min\{w_{ab}^{(s)}, w_{ab}^{(b)}, w_{ab}^{(t)}\}$.

3.3 Multi-Source Graph-Geodesic Health Coordinate

Healthy and failure reference regions are formed from the training landmarks as $\mathcal{H} = \{j : p_j \le \tau_H\}$ and $\mathcal{F} = \{j : p_j \ge \tau_F\}$, with $0 \le \tau_H < \tau_F \le 1$ are two threshold progresses determining healthy and failure regions, respectively. Multiple reference landmarks are retained rather than collapsing each region into a single prototype. For landmark j, define $D_H(j) = \min_{h\in\mathcal{H}} d_G(j,h)$ and $D_F(j) = \min_{f\in\mathcal{F}} d_G(j,f)$. For a query latent state $\mathbf{z}$, let $\mathcal{N}_{K_{\mathrm{OOS}}}(\mathbf{z})$ denote its K_{OOS} nearest fitted landmarks used for query out-of-sample (OOS) graph distance. The OOS distances are:

$$d_H(\mathbf{z}) = \min_{j\in\mathcal{N}_{K_{\mathrm{OOS}}}(\mathbf{z})} \left[\|\mathbf{z} - \mathbf{z}_j\|_2 + D_H(j)\right] \tag{5}$$

$$d_F(\mathbf{z}) = \min_{j\in\mathcal{N}_{K_{\mathrm{OOS}}}(\mathbf{z})} \left[\|\mathbf{z} - \mathbf{z}_j\|_2 + D_F(j)\right] \tag{6}$$

The raw graph-geodesic health coordinate is defined as:

$$s(\mathbf{z}) = \frac{d_H(\mathbf{z})}{d_H(\mathbf{z}) + d_F(\mathbf{z})} \in [0,1] \tag{7}$$

Values near 0 and 1 indicate proximity to healthy and failure regions, respectively. Conditional on an unchanged OOS K_{OOS}-nearest-landmark assignment, the coordinate is locally Lipschitz with respect to the latent state Theorem (2, Appendix C). Moreover, conditional on bounded errors in the healthy/failure reference distances and a denominator bounded away from zero, Theorem 3 quantifies the corresponding perturbation of the health coordinate (Appendix C). We reserve the notation s_t for this raw geometric coordinate and h_t for the final filtered Health Index (HI).

3.4 CAUSAL MONOTONIC HEALTH INDEX

Although the graph-geodesic coordinate s_t provides a degradation-aware geometric representation, it may still exhibit local fluctuations caused by sensor noise, latent-space variability, or changes in out-of-sample landmark attachment. We therefore convert s_t into a causal HI h_t under the assumption that degradation is irreversible at the considered sampling scale. This is a structural modelling assumption for the run-to-failure settings considered here rather than a mechanistic physical degradation law; accordingly, the resulting constraints are intended to enforce degradation-consistent trajectory behaviour rather than physical-model fidelity. A cumulative-maximum filter can enforce this constraint, but it transfers the full magnitude of any new upward excursion in s_t to the HI, after which the increase cannot be reversed. HAML instead uses a persistent level–velocity filter that attenuates transient observations while retaining evidence of sustained degradation.

For notational simplicity, we omit the asset index. Given the previous HI h_{t-1} and the persistent degradation-velocity state v_{t-1}, HAML first propagates the velocity and constructs a one-step health prediction:

$$\hat{v}_t = \rho v_{t-1}, \qquad \hat{h}_t = h_{t-1} + \hat{v}_t, \qquad e_t = s_t - \hat{h}_t \tag{8}$$

where $0 \leq \rho \leq 1$ controls the persistence of previously accumulated degradation momentum, and e_t is the innovation between the geometric observation and the predicted health state. Small positive innovations need not represent genuine degradation. We therefore introduce a one-sided dead zone:

$$u_t = [e_t - \tau_n]_+ = \max(e_t - \tau_n,\, 0) \tag{9}$$

and update the persistent velocity as

$$v_t = \min\left(\max\left(\hat{v}_t + \beta u_t,\, 0\right),\, v_{\max}\right) \tag{10}$$

where β controls velocity reinforcement and $v_{\max}$ limits the maximum degradation momentum. Consequently, innovations below the learned threshold τ_n cannot reinforce the velocity state, whereas repeated positive evidence above this threshold can accelerate degradation tracking.

The threshold is estimated exclusively from the training data using positive early-life variations of the raw geometric coordinate:

$$\tau_n = \min\left(\max\left(Q_{q_n}\left(\{s_t - s_{t-1} > 0 \mid p_{t-1},\, p_t \leq p_E\}\right),\, \tau_{\min}\right),\, \tau_{\max}\right) \tag{11}$$

where $p_E = 0.10$ defines the early-life region. Thus, τ_n acts as an empirical noise floor learned before inference; test-time lifecycle progress or RUL information is not required.

The geometric observation then corrects the predicted level as $\tilde{h}_t = \hat{h}_t + \alpha_t e_t$ using an asymmetric gain, where $\alpha_t = \alpha^+$ if $e_t \geq 0$, and $\alpha_t = \alpha^-$ otherwise. When $u_t > 0$, the updated velocity can contribute within the same cycle through

$$c_t = \begin{cases} \max\left\{\tilde{h}_t,\; h_{t-1} + v_t\right\}, & u_t > 0, \\ \tilde{h}_t, & u_t = 0. \end{cases} \tag{12}$$

Finally, c_t is projected onto the admissible interval defined by the non-decreasing and bounded-step degradation constraints:

$$h_t = \Pi_{\mathcal{I}_t}(c_t), \qquad \mathcal{I}_t = [h_{t-1},\; \min\{h_{t-1} + \Delta_{\max},\, 1\}] \tag{13}$$

or equivalently,

$$h_t = \min\left(\max\left(c_t, h_{t-1}\right),\; \min\{h_{t-1} + \Delta_{\max},\, 1\}\right).$$

This construction guarantees $0 \leq h_t - h_{t-1} \leq \Delta_{\max}$ and $0 \leq h_t \leq 1$. Thus, negative innovations can reduce the proposed increase relative to the velocity-guided prediction, but they cannot produce a health reversal that violates the assumed irreversible-degradation constraint; accumulated velocity instead decays gradually through ρ. Appendix C establishes the monotonicity and boundedness of the filter, its constrained convex-projection interpretation (Parikh & Boyd, 2014), and the effect of the dead zone on degradation momentum.

Table 1: C-MAPSS RUL prediction performance across ten seeds. RMSE, S-score, and P95 Late Error are reported as mean ± sample standard deviation over ten fixed seeds. **Red bold** denotes the best result, while ***blue bold italic*** denotes the second-best result.

Dataset	Metric	HAML	FAT	LOGO	LSTM	GRU	FC-STGNN	STGNN	TimesNet	TimeMixer++	PatchTST	iTransformer
FD001	RMSE ↓	**13.13 ± 0.50**	13.37 ± 0.45	13.43 ± 0.40	14.88 ± 1.22	14.84 ± 1.20	15.35 ± 1.07	14.34 ± 0.41	13.88 ± 0.65	15.94 ± 3.37	22.21 ± 8.08	***13.23 ± 0.26***
	S-score ↓	**274.3 ± 39.0**	315.6 ± 33.1	345.1 ± 38.0	461.9 ± 202.3	416.1 ± 95.1	409.4 ± 147.2	***311.8 ± 30.1***	315.1 ± 34.7	442.3 ± 200.5	2070.5 ± 3381.4	318.8 ± 55.8
	P95 Late Error ↓	**25.03 ± 3.04**	26.32 ± 2.83	27.34 ± 2.87	29.08 ± 3.37	26.50 ± 2.04	29.66 ± 5.19	***25.93 ± 3.54***	26.74 ± 4.24	26.70 ± 4.47	31.32 ± 16.40	27.84 ± 2.68
FD002	RMSE ↓	14.81 ± 0.70	15.32 ± 0.95	16.04 ± 9.48	17.19 ± 1.48	20.25 ± 1.23	14.59 ± 1.07	14.08 ± 0.71	**13.48 ± 0.60**	14.14 ± 0.72	21.20 ± 2.01	***13.96 ± 0.58***
	S-score ↓	884.3 ± 166.6	1149.6 ± 248.1	4864.3 ± 12956.7	1869.3 ± 606.5	2928.4 ± 890.2	1025.4 ± 206.9	**767.9 ± 65.1**	***831.6 ± 177.6***	889.3 ± 131.0	4012.4 ± 2307.2	905.3 ± 107.4
	P95 Late Error ↓	***30.69 ± 3.15***	36.48 ± 2.30	36.26 ± 11.62	38.56 ± 3.78	40.27 ± 3.47	33.61 ± 2.67	**28.88 ± 2.14**	32.73 ± 3.14	33.17 ± 3.13	43.65 ± 6.32	36.59 ± 2.32
FD003	RMSE ↓	14.31 ± 0.43	13.59 ± 0.87	14.87 ± 0.89	***12.01 ± 0.91***	**11.63 ± 0.69**	13.16 ± 0.84	12.89 ± 0.86	12.90 ± 0.79	12.77 ± 0.64	24.40 ± 15.93	13.11 ± 0.68
	S-score ↓	1069.1 ± 231.7	419.4 ± 160.9	762.4 ± 308.6	**247.0 ± 52.9**	***257.9 ± 84.8***	426.3 ± 143.1	331.7 ± 105.9	355.3 ± 82.5	339.6 ± 122.7	56993.6 ± 167429.5	392.2 ± 128.8
	P95 Late Error ↓	31.37 ± 1.72	33.75 ± 2.21	36.55 ± 1.95	**28.94 ± 2.37**	***30.31 ± 2.61***	33.71 ± 2.92	31.60 ± 2.80	32.73 ± 3.20	31.77 ± 3.40	44.77 ± 24.14	33.40 ± 2.75
FD004	RMSE ↓	15.79 ± 0.63	17.71 ± 1.46	***14.66 ± 0.41***	18.31 ± 2.17	23.03 ± 1.45	**14.42 ± 0.53**	15.19 ± 0.29	14.88 ± 0.56	15.80 ± 0.65	23.54 ± 4.19	15.32 ± 0.71
	S-score ↓	**1050.1 ± 141.5**	1866.1 ± 407.6	1156.7 ± 137.4	3744.5 ± 3412.6	7315.8 ± 2486.7	1145.3 ± 148.2	***1075.8 ± 131.3***	1080.6 ± 191.3	1382.5 ± 269.4	6589.6 ± 6445.9	1159.7 ± 129.2
	P95 Late Error ↓	**27.34 ± 1.49**	35.93 ± 3.07	34.40 ± 2.95	37.88 ± 4.86	48.14 ± 4.18	33.85 ± 3.28	29.70 ± 2.36	***28.93 ± 3.17***	31.65 ± 2.42	40.23 ± 11.20	31.82 ± 2.84

3.5 HI-Centred RUL Decoder and RUL Prediction

At each endpoint t, the decoder is centred on the causal HI trajectory rather than on the instantaneous scalar HI alone. The causal feature operator Γ constructs a 49-dimensional prognostic representation $\boldsymbol{\xi}_t$ comprising 44 features derived from the HI history and temporal dynamics, together with the raw graph coordinate s_t, the normalised persistent degradation-velocity state v_t^{norm}, and three standardised latent coordinates $\tilde{\mathbf{z}}_t$ (Appendix B). We train a decoder, i.e., a regressor, g, defining $y_{\max} = R_{\max}$ for C-MAPSS and $y_{\max} = 1$ for PHM12, and use the trained model to predict RUL:

$$\hat{y}_t = \min\left(\max\left(g(\boldsymbol{\xi}_t),\, 0\right),\, y_{\max}\right) \tag{14}$$

4 Experiments

We evaluate four questions: whether health-aware graph geometry produces a lifecycle-consistent degradation coordinate; whether causal filtering improves degradation-state consistency under the irreversible-degradation assumption; whether the complete HAML pipeline remains robust under controlled sensor perturbations; and whether the proposed health construction remains applicable across heterogeneous representation backends.

4.1 Experimental Setup

Baselines. We compare HAML with LSTM, GRU, FAT (Vu et al., 2024), LOGO (Wang et al., 2023), STGNN (Liu et al., 2023), FC-STGNN (Wang et al., 2024), TimesNet (Wu et al., 2023), TimeMixer++ (Wang et al., 2025a), PatchTST (Nie et al., 2023), and iTransformer (Liu et al., 2024); detailed configurations are provided in Appendix D.

Datasets. We use all C-MAPSS subsets (Saxena et al., 2008) and PHM12 (Nectoux et al., 2012). C-MAPSS uses 14 sensors, 50-cycle windows, and a 125-cycle RUL cap (Wang et al., 2025b). PHM12 follows the predefined bearing split using horizontal vibration (Wang et al., 2025b; Wei et al., 2023); bearing-specific lifetime scaling is used only for target construction and evaluation and is never provided as model input (Wang et al., 2025b; Wei et al., 2023). Dataset statistics and complete preprocessing protocols are provided in Appendix D.

Configuration. The primary HAML backend uses 3-D UMAP with 1,500 landmarks, healthy and failure thresholds of 0.05 and 0.97, respectively, a progress penalty of 4.5, and Balanced Regression Forest (BRF) decoding; complete settings are provided in Appendix E.

Evaluation. We report ten-seed mean ± sample standard deviation for RMSE, S-score (S_{v1} for C-MAPSS tasks and S_{v2} for PHM tasks (Wang et al., 2025b)), P95 Late Error, and HI diagnostics (Coble & Hines, 2009). Metric definitions and additional analyses are provided in Appendices D and G.

4.2 COMPARING TO BASELINES

HAML ranks within the top three in eight of twelve C-MAPSS dataset–metric comparisons, achieving the best three metrics on FD001 and the best S-score and P95 Late Error on FD004 (Table 1). FD003 is the principal challenging case, where several baselines obtain lower RMSE/S-score; Appendix G shows that its elevated S-score is concentrated in a small number of extreme late predictions. On PHM12, HAML ranks first in 12 of 33 bearing–metric comparisons and within the top two in 20 of 33 (Appendix G).

4.3 NOISE SENSITIVITY ANALYSIS

Models are trained on clean data and evaluated without retraining after adding Gaussian noise $\mathcal{N}\left(0, \ell^2 \sigma^2_{k,\mathbf{C}}\right)$ to the raw test sensor values, where $\sigma_{k,\mathbf{C}}$ is estimated from the training data for sensor k under operating condition $\mathbf{C}$, and $\ell \in \{0.05, 0.1, 0.2\}$. Identical perturbations are used across models; details are in Appendix G. To illustrate the impact of noise on methods' performance, we quantify degradation by $\Delta\mathcal{Q}(\ell)(\%) = 100 \times [\mathcal{Q}(\ell)/\mathcal{Q}(0) - 1]$, for each metric $\mathcal{Q}$ where $\mathcal{Q}(0)$ denotes performance on the clean test data and $\mathcal{Q}(\ell)$ denotes performance at noise level ℓ.

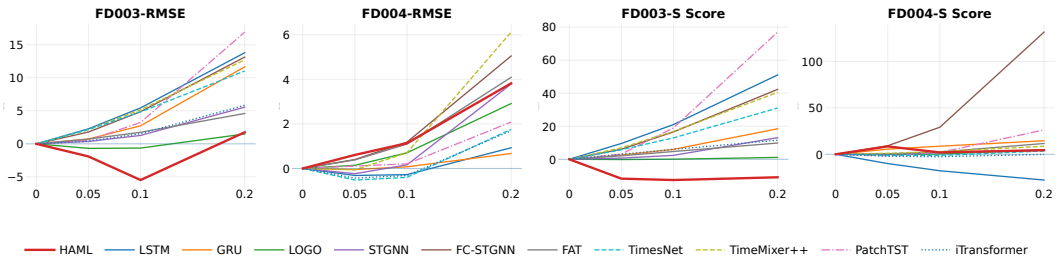

Figure 2: **Test-time Gaussian-noise sensitivity on FD003 and FD004.** Lower values indicate greater robustness; negative values denote improvement relative to the clean reference.

HAML exhibits controlled degradation under increasing test-time noise (Figure 2). On FD003 (single condition), its RMSE changes by -1.8%, -5.4% and $+1.8\%$ across the three noise levels, while its S-score remains below the corresponding clean score at all levels; several sequence baselines degrade substantially more. On heterogeneous FD004, HAML's advantage is smaller, but degradation remains bounded, with RMSE increasing by at most 3.8% and S-score by at most 8.0%. These results show that the complete HAML pipeline remains comparatively stable under controlled sensor perturbations, although the degree of robustness is dataset-dependent.

4.4 FURTHER ANALYSIS

Effect of causal filtering. Filtering improves Pearson and Spearman correlations with held-out lifecycle progress across all four subsets, with gains of 0.016 - 0.029 and 0.044 - 0.072, respectively (Figure 3). The larger rank-correlation gains observed on FD003–FD004 indicate a greater benefit under stronger local ordering ambiguity. Because monotonicity is imposed by construction, the important observation is that lifecycle alignment improves while the filtered state remains strongly associated with the underlying graph coordinate.

Latent geometry and health progression. Figure 4 shows that the unsupervised FD001 representation contains broad healthy-to-failure organisation but substantial local mixing, and therefore does not itself define a unique degradation axis. HAML extracts a graph-geodesic health coordinate from this multidimensional structure and temporally regularises it into the HI, illustrating the intended separation between representation geometry and health-state construction.

Twelve of 17 sensitivity sweeps produce $< 2.5\%$ mean composite change; partial progress supervision and alternative landmark sampling are analysed in Appendix G.

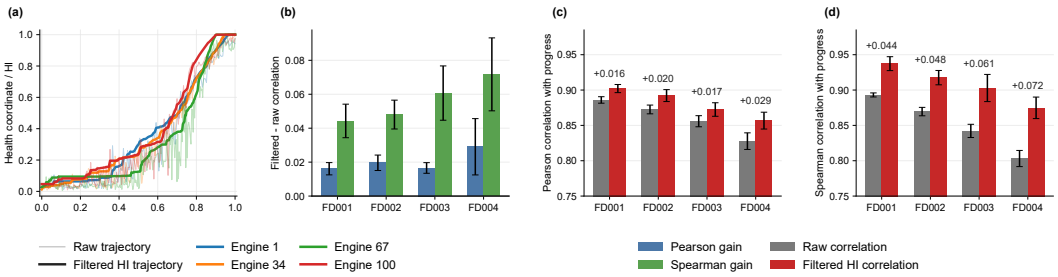

Figure 3: **Causal filtering improves lifecycle alignment while preserving graph-derived health information.** (a) Representative raw graph-geodesic coordinates s_t and corresponding filtered HIs h_t. (b) Pearson and Spearman correlations between the s_t and h_t. (c, d) Pearson and Spearman correlations between normalised degradation progress and s_t and h_t, averaged over ten seeds. Error bars indicate standard deviation.

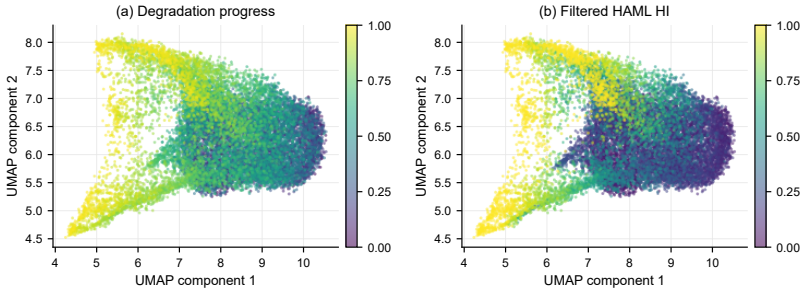

Figure 4: **From latent degradation geometry to HI-centred prognostic decoding.** (a) FD001 test latent states coloured retrospectively by normalised degradation progress and (b) the same states coloured by the filtered HAML HI. The progress colouring is used only for visualisation and is not supplied to the inference pipeline.

5 Ablation Study

We evaluate eight controlled ablations covering graph geometry (A1–A4), causal filtering (A5–A6), decoder information (A7), and representation-backend choice (A8) (Table 2). All other applicable settings are fixed; full definitions and HI diagnostics are provided in Appendix G.

Ablation findings. A1 and A2 produce relatively modest predictive changes but consistently reduce HI–progress correlation, indicating that graph-geodesic and progress-aware weighting primarily improve lifecycle-consistent health geometry rather than providing a large standalone accuracy gain. The strongest geometric effect occurs in A3: collapsing distributed endpoint regions to single medoids markedly reduces progress correlation, prognosability, and healthy–failure separation, particularly on FD003–FD004. Removing temporal trajectory edges (A4) has a smaller, dataset-dependent effect, suggesting that most lifecycle structure is already captured by the spatial health graph. A5 exposes an explicit accuracy-constraint trade-off. Removing causal filtering slightly improves clean-data prediction but eliminates the imposed non-decreasing trajectory behaviour: monotonicity falls from 1.0 to approximately 0.52–0.54 and cycle-to-cycle HI variation increases substantially. In contrast, A6 retains monotonicity but generally performs worse than the proposed persistent level–velocity construction, showing that degradation memory contributes beyond the monotonic constraint itself. A7 further shows that scalar HI alone is insufficient for accurate RUL estimation: restoring HI history, temporal dynamics, and complementary geometric state information progressively improves predictive performance. Finally, A8 demonstrates representation-backend portability rather than representation invariance. PCA, Isomap, PHATE, and Diffusion Maps can all support the downstream HAML construction, but their predictive and HI characteristics differ substantially, confirming that HAML is not defined by UMAP while remaining dependent on the quality of the supplied representation.

Table 2: Predictive effects of HAML ablations on C-MAPSS. **Red bold** denotes a result numerically better than A0.

Dataset	Ablation	Configuration	RMSE ↓	S-score ↓	P95 Late Error ↓
FD001	A0	HAML	13.13 ± 0.50	274.3 ± 39.0	25.03 ± 3.04
	A1	Graph-geodesic → Euclidean health coordinate	13.38 ± 0.51	283.4 ± 20.9	25.16 ± 2.37
	A2	Progress-aware edge penalty removed	13.24 ± 0.70	277.5 ± 41.7	25.40 ± 3.57
	A3	Multi-source reference regions → single medoids	14.06 ± 0.91	292.7 ± 58.1	**24.82 ± 3.41**
	A4	Temporal trajectory edges removed	13.39 ± 0.52	283.8 ± 32.0	25.61 ± 2.33
	A5	Causal monotonic HI filter removed	**12.77 ± 0.55**	**248.3 ± 39.8**	**24.17 ± 2.45**
	A6	Persistent level–velocity HI → cumulative-max HI	14.01 ± 0.74	316.1 ± 49.6	25.72 ± 2.51
	A7a	Current HI only (1 feature)	16.08 ± 1.10	412.6 ± 98.7	27.63 ± 4.44
	A7b	30-step HI history (30 features)	14.91 ± 1.04	360.6 ± 78.9	26.52 ± 4.47
	A7c	HI temporal decoder (44 features)	13.69 ± 0.84	312.3 ± 60.8	28.23 ± 3.87
	A7d	HI decoder + s_t + velocity (46 features)	13.66 ± 0.88	308.3 ± 61.8	27.20 ± 5.24
	A8-PCA	Representation-backend substitution: PCA	13.25 ± 0.55	279.1 ± 36.0	25.63 ± 2.04
	A8-Isomap	Representation-backend substitution: Isomap	13.15 ± 0.69	**261.7 ± 39.9**	**24.91 ± 2.75**
	A8-PHATE	Representation-backend substitution: PHATE	**12.97 ± 0.73**	**261.9 ± 51.1**	**24.24 ± 3.69**
	A8-Diffusion	Representation-backend substitution: Diffusion Maps	14.23 ± 0.73	323.6 ± 85.2	25.34 ± 3.80
FD002	A0	HAML	14.81 ± 0.70	884.30 ± 166.6	30.69 ± 3.15
	A1	Graph-geodesic → Euclidean health coordinate	14.82 ± 0.73	887.80 ± 125.9	30.76 ± 2.48
	A2	Progress-aware edge penalty removed	14.86 ± 0.60	907.80 ± 127.1	30.74 ± 3.37
	A3	Multi-source reference regions → single medoids	14.81 ± 0.63	889.90 ± 107.9	**29.26 ± 2.10**
	A4	Temporal trajectory edges removed	14.84 ± 0.64	885.60 ± 125.5	30.77 ± 2.36
	A5	Causal monotonic HI filter removed	**14.73 ± 0.49**	**798.40 ± 77.50**	**26.27 ± 2.59**
	A6	Persistent level–velocity HI → cumulative-max HI	15.10 ± 0.52	898.20 ± 117.9	**28.95 ± 1.25**
	A7a	Current HI only (1 feature)	18.81 ± 0.49	1425.7 ± 141.4	32.20 ± 4.26
	A7b	30-step HI history (30 features)	17.19 ± 0.83	1264.3 ± 332.7	31.11 ± 3.79
	A7c	HI temporal decoder (44 features)	15.67 ± 0.51	997.60 ± 145.9	32.14 ± 3.12
	A7d	HI decoder + s_t + velocity (46 features)	15.55 ± 0.55	981.30 ± 154.4	31.11 ± 2.66
	A8-PCA	Representation-backend substitution: PCA	**13.55 ± 0.38**	**727.90 ± 75.30**	**27.43 ± 2.16**
	A8-Isomap	Representation-backend substitution: Isomap	**14.32 ± 0.55**	**817.80 ± 109.2**	**30.07 ± 2.89**
	A8-PHATE	Representation-backend substitution: PHATE	15.35 ± 0.67	932.60 ± 114.8	30.79 ± 2.88
	A8-Diffusion	Representation-backend substitution: Diffusion Maps	**14.73 ± 0.24**	**869.80 ± 86.70**	**30.56 ± 1.45**
FD003	A0	HAML	14.31 ± 0.43	1069.1 ± 231.7	31.37 ± 1.72
	A1	Graph-geodesic → Euclidean health coordinate	**14.25 ± 1.06**	**855.30 ± 253.2**	31.55 ± 2.68
	A2	Progress-aware edge penalty removed	14.37 ± 0.67	**968.70 ± 279.4**	31.52 ± 2.42
	A3	Multi-source reference regions → single medoids	14.39 ± 0.47	**568.40 ± 147.7**	33.16 ± 3.36
	A4	Temporal trajectory edges removed	**14.14 ± 0.69**	**965.60 ± 341.5**	**30.73 ± 3.55**
	A5	Causal monotonic HI filter removed	**13.73 ± 0.50**	**1000.7 ± 126.0**	**30.23 ± 2.88**
	A6	Persistent level–velocity HI → cumulative-max HI	14.46 ± 0.70	**1023.9 ± 212.8**	**31.00 ± 2.71**
	A7a	Current HI only (1 feature)	17.88 ± 0.71	**1046.7 ± 244.4**	36.60 ± 2.91
	A7b	30-step HI history (30 features)	17.08 ± 0.50	**1041.3 ± 328.7**	36.97 ± 1.77
	A7c	HI temporal decoder (44 features)	14.97 ± 0.53	**985.10 ± 296.8**	34.66 ± 2.63
	A7d	HI decoder + s_t + velocity (46 features)	14.80 ± 0.51	**991.30 ± 304.1**	35.23 ± 2.37
	A8-PCA	Representation-backend substitution: PCA	**12.91 ± 0.58**	**322.30 ± 57.00**	**30.72 ± 4.83**
	A8-Isomap	Representation-backend substitution: Isomap	14.62 ± 1.11	**640.60 ± 335.4**	33.49 ± 2.64
	A8-PHATE	Representation-backend substitution: PHATE	14.81 ± 1.31	**635.30 ± 231.5**	32.11 ± 3.25
	A8-Diffusion	Representation-backend substitution: Diffusion Maps	14.93 ± 1.05	**786.70 ± 240.2**	33.07 ± 4.21
FD004	A0	HAML	15.79 ± 0.63	1050.1 ± 141.5	27.34 ± 1.49
	A1	Graph-geodesic → Euclidean health coordinate	15.80 ± 0.81	1057.8 ± 115.4	28.08 ± 1.95
	A2	Progress-aware edge penalty removed	15.94 ± 0.72	1101.4 ± 112.7	29.15 ± 1.74
	A3	Multi-source reference regions → single medoids	15.88 ± 0.74	1171.3 ± 204.4	28.62 ± 3.51
	A4	Temporal trajectory edges removed	15.83 ± 0.61	1068.2 ± 143.5	27.68 ± 2.28
	A5	Causal monotonic HI filter removed	**15.55 ± 0.47**	**1014.2 ± 117.6**	**26.82 ± 2.50**
	A6	Persistent level–velocity HI → cumulative-max HI	16.18 ± 0.39	1175.2 ± 96.80	28.75 ± 1.35
	A7a	Current HI only (1 feature)	22.11 ± 1.43	4258.2 ± 1355	38.22 ± 5.07
	A7b	30-step HI history (30 features)	19.78 ± 1.09	2205.7 ± 726.0	35.52 ± 3.83
	A7c	HI temporal decoder (44 features)	17.29 ± 1.02	1361.4 ± 329.5	30.52 ± 3.07
	A7d	HI decoder + s_t + velocity (46 features)	16.96 ± 0.89	1252.7 ± 244.7	30.21 ± 3.02
	A8-PCA	Representation-backend substitution: PCA	16.04 ± 0.41	1104.2 ± 130.5	28.01 ± 1.74
	A8-Isomap	Representation-backend substitution: Isomap	16.55 ± 0.65	1201.3 ± 179.3	29.80 ± 4.61
	A8-PHATE	Representation-backend substitution: PHATE	16.19 ± 0.49	1195.9 ± 151.2	28.67 ± 3.41
	A8-Diffusion	Representation-backend substitution: Diffusion Maps	16.79 ± 0.75	1389.8 ± 271.2	30.18 ± 2.18

6 Conclusion

We introduced Health-Aware Manifold Learning (HAML), a framework that separates latent representation learning from degradation-aware health construction for causal RUL prediction. Lifecycle progress is introduced only after representation fitting to shape a progress-aware landmark graph and distributed healthy/failure reference regions. The resulting graph-geodesic health coordinate is converted into a causal, bounded, non-decreasing HI through persistent level–velocity filtering, while selected temporal and geometric state information supports downstream RUL estimation.

Across C-MAPSS and PHM12, HAML provides competitive prediction, favourable late-error behaviour, and meaningful lifecycle alignment; controlled C-MAPSS perturbations further demonstrate stable performance under test-time noise. Ablations identify complementary roles for graph geometry, endpoint references, causal filtering, degradation memory, and HI-centred decoding, while alternative backends demonstrate portability beyond UMAP. HAML assumes irreversible run-to-failure degradation and relies on lifecycle-progress supervision, representation quality, and landmark configuration, which are discussed in Appendix H.

Reproducibility Statement. Implementation details and hyperparameters are provided in Appendices D– E, theoretical assumptions and proofs in Appendix C, and additional experiments in Appendix G. Code will be released upon publication.

AI Usage Statement. An OpenAI language model was used only for language refinement, including grammar, wording and presentation. All scientific ideas, methods, experiments, analyses and conclusions were developed and verified by the authors.

REFERENCES

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Tarak Benkedjouh, Kamal Medjaher, Noureddine Zerhouni, and Saïd Rechak. Remaining useful life estimation based on nonlinear feature reduction and support vector regression. *Engineering Applications of Artificial Intelligence*, 26(7):1751–1760, 2013.

Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. Visionts: Visual masked autoencoders are free-lunch zero-shot time series forecasters. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Jamie Coble and J Wesley Hines. Identifying optimal prognostic parameters from data: a genetic algorithms approach. In *Annual conference of the PHM society*, volume 1, 2009.

Yaguo Lei, Naipeng Li, Liang Guo, Ningbo Li, Tao Yan, and Jing Lin. Machinery health prognostics: A systematic review from data acquisition to rul prediction. *Mechanical systems and signal processing*, 104:799–834, 2018.

Xiang Li, Qian Ding, and Jian-Qiao Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172:1–11, 2018.

Jiayang Liu, Xiaosun Wang, Fuqi Xie, Shijing Wu, and Deng Li. Condition monitoring of wind turbines with the implementation of spatio-temporal graph neural network. *Engineering Applications of Artificial Intelligence*, 121:106000, 2023.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. doi: 10.21105/joss.00861.

P Nectoux, R GOURIVEAU, K MEDJAHER, et al. An experimentalplatform forbearingsaccelerated degradation tests. In *IEEE. International Conference on Prognostics and Health Management. Beijing:[sn] 2012*, pp. 23–25, 2012.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3): 127–239, 2014.

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. IEEE, 2008.

Yajun Shang, Xinglu Tang, Guangqian Zhao, Peigang Jiang, and Tian Ran Lin. A remaining life prediction of rolling element bearings based on a bidirectional gate recurrent unit and convolution neural network. *Measurement*, 202:111893, 2022.

Zhong Siru, Ruan Weilin, Ming Jin, Li Huan, Wen Qingsong, and Liang Yuxuan. Time-vlm: Exploring multimodal vision-language models for augmented time series forecasting. In *Forty-Second International Conference on Machine Learning (ICML 2025)*. Proceedings of Machine Learning Research, 2025.

Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

Hieu Vu, Thanh Nguyen, and Eyad Elyan. Geometry-aware visual representation for remaining useful life prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

Trung Hieu Vu, Eyad Elyan, Will Vorley, Joe Goodlad, Truong Dang, and Tien Thanh Nguyen. Fat: Fusion-attention transformer for remaining useful life prediction. In *International Conference on Pattern Recognition*, pp. 286–301. Springer, 2024.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. In *International Conference on Learning Representations*, volume 2025, pp. 16980–17016, 2025a.

Xin Wang, Lingli Cui, Huaqing Wang, and Hong Jiang. A generalized health indicator for performance degradation assessment of rolling element bearings based on graph spectrum reconstruction and spectrum characterization. *Measurement*, 176:109165, 2021.

Yucheng Wang, Min Wu, Ruibing Jin, Xiaoli Li, Lihua Xie, and Zhenghua Chen. Local–global correlation fusion-based graph neural network for remaining useful life prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):753–766, 2023.

Yucheng Wang, Yuecong Xu, Jianfei Yang, Min Wu, Xiaoli Li, Lihua Xie, and Zhenghua Chen. Fully-connected spatial-temporal graph for multivariate time-series data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 15715–15724, 2024.

Yucheng Wang, Min Wu, Xiaoli Li, Lihua Xie, and Zhenghua Chen. A survey on graph neural networks for remaining useful life prediction: Methodologies, evaluation and future trends. *Mechanical Systems and Signal Processing*, 229:112449, 2025b.

Yupeng Wei, Dazhong Wu, and Janis Terpenny. Learning the health index of complex systems using dynamic conditional variational autoencoders. *Reliability Engineering & System Safety*, 216:108004, 2021.

Yupeng Wei, Dazhong Wu, and Janis Terpenny. Bearing remaining useful life prediction using self-adaptive graph convolutional networks with self-attention mechanism. *Mechanical Systems and Signal Processing*, 188:110010, 2023.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023.

Sheng Xiang, Yi Qin, Caichao Zhu, Yangyang Wang, and Haizhou Chen. Lstm networks based on attention ordered neurons for gear remaining life prediction. *ISA transactions*, 106:343–354, 2020.

Wen Yu, Dechang Pi, Lingqiang Xie, and Yi Luo. Multiscale attentional residual neural network framework for remaining useful life prediction of bearings. *Measurement*, 177:109310, 2021.

Wennian Yu, II Yong Kim, and Chris Mechefske. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. *Mechanical Systems and Signal Processing*, 129:764–780, 2019.

Guokang Zheng, Yasong Li, Zheng Zhou, and Ruqiang Yan. A remaining useful life prediction method of rolling bearings based on deep reinforcement learning. *IEEE Internet of Things Journal*, 11(13):22938–22949, 2024.

Enrico Zio. Prognostics and health management (phm): Where are we and where do we (need to) go in theory and practice. *Reliability Engineering & System Safety*, 218:108119, 2022.

APPENDIX

Contents

A. Mathematical Notations **13**

B. Temporal Descriptor Encoder and Decoder Features **15**

B1. TEMPORAL DESCRIPTOR ENCODER 15

B2. DECODER FEATURES 16

C. Structural and Stability Properties **18**

D. Evaluation Details **32**

D1. DATASETS AND PROCESSING 32

D2. BASELINES AND PARAMETER SETTINGS 34

E. Algorithm and Parameter Settings **42**

F. Computational Complexity and Empirical Efficiency **44**

G. Additional Results and Analysis **46**

G1. EFFECT OF REPRESENTATION AND DECODER CHOICE 46

G2. COMPARING TO BASELINES 47

G3. FURTHER OBSERVATIONS ON HAML BEHAVIOUR 52

G4. SENSITIVITY ANALYSIS 55

G5. FURTHER ANALYSIS 62

G6. ABLATION STUDY 65

H. Limitations and Future Work **70**

A. MATHEMATICAL NOTATIONS

Data, window, and target | **Manifold, landmark, and graph**

Notations	Meaning	Notations	Meaning
i	Asset/engine index	$f_{\mathcal{M}}$	Manifold backend mapping.
t	Step/endpoint index	$\mathbf{z}_{i,t}$	Latent state.
T_i	Failure step/lifespan of asset i	d	Latent dimension.
F	Number of sensors	$\mathcal{L} = \{\ell_1, \ldots, \ell_{M_L}\}$	Landmark set.
W	Causal sensor window length	M_L	Number of implemented landmarks.
$\mathbf{X}_{i,t}$	Causal sensor window ending at step t	$\mathcal{G}_H$	HAML health graph.
$\mathbf{x}_{i,t}$	Sensor observation at step t of asset i	E_s, E_t, E_b	Spatial, temporal, and bridge edge sets.
$R_{\max}$	RUL cap	$w_{uv}^{(s)}$	Spatial edge cost.
$y_{i,t}$	Capped training RUL target	$w_{uv}^{(t)}$	Temporal edge cost.
$\hat{y}_{i,t}$	Predicted RUL	λ_G	Health-progress penalty.
$t_i^{(0)}$	First valid endpoint with complete window	η_T	Temporal-edge factor.
$p_{i,t}$	Normalised degradation progress (train-only)	Δc_{uv}	Step gap $\lvert t_u - t_v \rvert$.
R_{desc}	Recent descriptor length	d_{uv}	Euclidean latent transition distance between landmarks u, v.
$\mathbf{q}_{i,t}$	Concatenated temporal descriptor vector	$\widetilde{d}_{uv}$	Condition-capped transition distance.
$\phi(\mathbf{X}_{i,t})$	Full multi-sensor descriptor encoder, with $\phi(\mathbf{X}_{i,t}) = \mathbf{q}_{i,t} \in \mathbb{R}^{7F}$	$Q_{0.95}$	95th percentile operator for ordinary transitions.
$q_k^{\text{last}}, \mu_k, \sigma_k, \Delta_k, a_k$	Endpoint value, mean, standard deviation, endpoint delta, and LS slope	$d_G(u, v)$	Shortest-path graph geodesic distance.
$\mu_k^{(R_{\text{desc}})}, a_k^{(R_{\text{desc}})}$	Recent mean and recent slope	$\mathcal{P}(u, v)$	Set of graph paths between u and v.
k	Sensor channel	$\mathcal{H}, \mathcal{F}$	Healthy/failure landmark reference sets.
u, v	Graph/landmark vertices	τ_H / τ_F	Healthy/failure progress thresholds.
j	Generic landmark index used in reference/OOS minimisations	$D_H(j), D_F(j)$	Landmark distance-to-reference precomputes.
$\phi_k(\mathbf{X})$	Seven-descriptor encoder for sensor channel k	$\mathcal{N}_{K_{\text{OOS}}}(\mathbf{z})$	K_{OOS} nearest fitted landmarks for query.
		$d_H(\mathbf{z}), d_F(\mathbf{z})$	Query-to-healthy/failure graph-geodesic distances.
		$s(\mathbf{z}), s_t$	Raw graph-geodesic health coordinate.
		$\mathcal{Z}_L = \{\mathbf{z}_j = f_{\mathcal{M}}(\ell_j)\}$	Latent landmark set.
		p_j	Training degradation progress associated with the corresponding landmark.
		$w_{uv}^{(b)}$	Bridge-edge cost.
		k_G	Mutual-nearest-neighbour graph degree.
		V	Landmark-graph vertex set.
		w_{uv}	Final retained graph-edge cost after resolving applicable spatial, temporal, and bridge costs.

Causal HI filter

Notations	Meaning
h_t	Filtered Health Index.
v_t	Persistent degradation velocity state.
$\hat{v}_t = \rho v_{t-1}$	Predicted velocity prior $= \rho v_{t-1}$.
$\widetilde{h}_t$	Asymmetrically corrected level before monotone projection.
e_t	Innovation/residual.
τ_n	Early-life noise threshold.
u_t	Positive innovation $\max(e_t - \tau_n, 0)$.
α^+, α^-	Positive and negative residual gain.
β	Velocity reinforcement gain.
ρ	Velocity persistence.
$v_{\max}$	Maximum persistent velocity.
$\Delta_{\max}$	Maximum one-step HI step.
$\tau_{\min}$, $\tau_{\max}$	Clip bounds of the learned noise threshold.
$\hat{h}_t$	One-step level prior.
q_n	Quantile level used to estimate the early-life noise threshold.
$Q_{q_n}(\cdot)$	Quantile operator at level q_n.
α_t	Step-dependent level-correction gain.
$\mathcal{F}_{\text{HI}}$	Causal monotonic level–velocity filter.

Decoder and metrics

Notations	Meaning
$\boldsymbol{\xi}_{i,t}$	Final decoder feature vector.
$\Gamma(\cdot)$	Causal decoder-feature operator.
$g(\cdot)$	Stratified balanced regression forest decoder.
H_{hist}	HI history length.
$w \in \{10, 20, 40\}$	Trend windows for slope/delta.
$\varepsilon_{\text{rate}}$	Rate denominator epsilon.
v_t^{norm}	Normalised velocity feature.
$\widetilde{\mathbf{z}}_t$	Standardised latent coordinates.
N_{eval}	Number of evaluation samples.
N_{eng}	Number of engines/assets for HI metrics.
ε_g	Denominator numerical epsilon.
n_t	Number of HI observations available up to the current endpoint.
$\mathbf{f}_t^{(\text{HI},44)}$	44-dimensional causal HI-history and temporal-summary feature vector.

Theorem-specific

Notations	Meaning
$\mathcal{G}_s = (V, E_s)$	Spatial subgraph of the HAML health graph used in Theorem 1.
Δp_{uv}	Degradation-progress discrepancy across a spatial edge $\lvert p_u - p_v \rvert$.
$\Pi = (u_0, \ldots, u_m)$	Purely spatial path between two landmarks.
$L_{\lambda_G}(\Pi)$	Progress-aware length of spatial path Π.
$d_{s,0}(u, v)$	Spatial shortest-path distance without progress penalty.
Ω	Latent region in which the K_{OOS}-nearest-landmark set is unchanged.
$f_j^H(\mathbf{z})$	Healthy-reference landmark attachment function in Theorem 2.
$d_H^*(\mathbf{z})$, $d_F^*(\mathbf{z})$	Target/ideal healthy- and failure-reference distances in Theorems 3–4.
$s^*(\mathbf{z})$	Health coordinate defined by the target reference distances.
ε_M	Uniform bound on healthy/failure reference-distance approximation error.
$L_0(\Pi)$	Unpenalized length of spatial path Π.
$d_{\min}$	Minimum positive spatial edge length in E_s.
$d_{s,\lambda_G}(u, v)$	Spatial shortest-path distance with progress penalty.
δ_Ω	Lower bound $\inf_{\mathbf{z}\in\Omega} [d_H(\mathbf{z}) + d_F(\mathbf{z})]$.
$f_j^F(\mathbf{z})$	Failure-reference landmark attachment function in Theorem 2.
$d_{H,M}(\mathbf{z})$, $d_{F,M}(\mathbf{z})$	Finite-M landmark/graph/OOS approximations of the target distances.
$s_M(\mathbf{z})$	Health coordinate defined by the finite-M approximated distances.
$\delta_0 > 0$	Uniform lower bound on $d_H^\star + d_F^\star$ used in Theorems 3–4.
$d_{\mathcal{M}}$	Intrinsic dimension of the theoretical latent manifold/support.
M	Number of i.i.d. theoretical landmarks in Theorem 4.

Notations	Meaning	Notations	Meaning
$(\mathcal{M}, D_{\mathcal{M}})$	Compact theoretical Riemannian latent support and its geodesic metric.	μ	Probability distribution from which theoretical landmarks are sampled.
$\mathcal{Z}_M = \{\mathbf{z}_1, \ldots, \mathbf{z}_M\}$	I.i.d. latent landmark set used only in Theorem 4.	$B_{\mathcal{M}}(\mathbf{z}, r)$	Geodesic ball centred at $\mathbf{z}$ with radius r.
r_M	Covering radius of $\mathcal{Z}_M$ under $D_{\mathcal{M}}$.	r_0	Radius below which the local volume lower bound is assumed.
$\pi(\mathbf{z})$, $\pi_{\min}$	Sampling density with respect to Riemannian volume and its positive lower bound.	N_r	Number of centres in an $r/2$-net of $\mathcal{M}$.
$\mathrm{Vol}(\cdot)$	Riemannian volume measure.	C_{dist}	Constant relating reference-distance error to landmark covering radius.
c_{vol}	Constant in $\mathrm{Vol}(B_{\mathcal{M}}(\mathbf{z}, r)) \geq c_{\mathrm{vol}} r^{d_{\mathcal{M}}}$.	η_{prob}	Failure probability/confidence parameter in Theorem 4.
C_{cov}	Constant in the covering-number bound $N_r \leq C_{\mathrm{cov}} r^{-d_{\mathcal{M}}}$.	$\kappa = c_* A^{d_{\mathcal{M}}}$	Probability-decay exponent parameter in the covering bound.
$c_* = \frac{\pi_{\min} c_{\mathrm{vol}}}{2^{d_{\mathcal{M}}}}$	Lower probability-mass constant used in Theorem 4.	r	Generic geodesic radius used in the covering argument.
A	Positive constant used to choose the covering radius r.	δ_G	Positive lower bound of degradation-progress discrepancy used in Theorem 1, Property 2.
C	Final constant in the Theorem 4 health-coordinate error bound.		

For notational simplicity, we suppress the asset index i throughout this subsection and write s_t, h_t, and v_t for the state variables of a single asset trajectory.

B. Temporal Descriptor Encoder and Decoder Features

B1. Temporal descriptor encoder

In the temporal descriptor encoder module, HAML converts each sensor channel into seven interpretable temporal descriptors: endpoint value, mean, standard deviation, endpoint-to-start change, full-window least-squares slope, recent mean, and recent slope. For sensor k, an input window $\mathbf{X}$ (length W) is encoded into a vector of seven descriptors $\left[q_k^{last}, \mu_k, \sigma_k, \Delta_k, a_k, \mu_k^{(R_{desc})}, a_k^{(R_{desc})} \right]$ to describe the raw data inside, where the endpoint value $q_k^{last} = x_{W,k}$, mean $\mu_k = \frac{1}{W} \sum_{l=1}^{W} x_{l,k}$, standard deviation $\sigma_k = \sqrt{\frac{1}{W} \sum_{l=1}^{W} \left(x_{l,k} - \bar{x}_k \right)^2}$, endpoint-to-start change $\Delta_k = x_{W,k} - x_{1,k}$, full-window least-squares slope $a_k = \frac{\sum_{l=1}^{W} \left(l - \bar{l} \right) \left(x_{l,k} - \bar{x}_k \right)}{\sum_{l=1}^{W} \left(l - \bar{l} \right)^2}$, recent mean over the most recent R_{desc} steps $\mu_k^{(R_{desc})} = \frac{1}{R_{desc}} \sum_{l=W-R_{desc}+1}^{W} x_{l,k}$, and recent slope $a_k^{(R_{desc})} = \frac{\sum_{l=W-R_{desc}+1}^{W} \left(l - \bar{l} \right) \left(x_{l,k} - \bar{x}_k \right)}{\sum_{l=W-R_{desc}+1}^{W} \left(l - \bar{l} \right)^2}$, R_{desc} set to 5 in this study. Each descriptor is normalised using training-fitted z-score standardisation.

For C-MAPSS, each causal window $\mathbf{X}_{i,t}$ contains $W = 50$ cycles from the 14 retained prognostic sensor channels. Using descriptor encoder ϕ yields the concatenated descriptor vector $\mathbf{q}_{i,t} = \phi(\mathbf{X}_{i,t}) \in \mathbb{R}^{98}$. After training-fitted descriptor standardisation, $\mathbf{q}_{i,t}$ is passed to the replaceable representation backend f_M, which produces the latent state $\mathbf{z}_{i,t} = f_M(\mathbf{q}_{i,t}) \in \mathbb{R}^d$.

Meanwhile, for PHM12, each prediction uses only the current horizontal-vibration acquisition $\mathbf{x}_{i,t} \in \mathbb{R}^{2560}$. Following Min–Max preprocessing fitted using training data only, each acquisition is partitioned into $W = 50$ contiguous, non-overlapping intra-acquisition patches that jointly cover all 2,560 samples exactly once. The first 10 patches contain 52 samples each, and the remaining 40 patches contain 51 samples each. A bearing feature adapter converts each patch into 24 time- and frequency-domain bearing feature channels, yielding a tensor $\in \mathbb{R}^{50\times 24}$. The 24 feature channels follow the time- and frequency-domain bearing-feature definitions in (Lei et al., 2008; Juodelyte et al., 2022). Stacking the corresponding feature values across the 50 patches yields $\in \mathbb{R}^{50\times 24}$, where each column represents the evolution of one bearing feature across the 50 intra-acquisition patches. The standard HAML temporal descriptor encoder ϕ is then applied independently to each of the 24 feature-channel sequences, producing seven temporal descriptors per channel and hence $\mathbf{q}_{i,t} \in \mathbb{R}^{168}$. After training-fitted descriptor standardisation, $\mathbf{q}_{i,t}$ is supplied to the replaceable representation backend f_M.

The 50 patch positions describe within-acquisition vibration structure and are not lifecycle steps. Each resulting latent state corresponds to one acquisition along the bearing trajectory; HAML temporal graph edges connect consecutive acquisitions of the same training bearing, not consecutive intra-acquisition patches. Thus, temporal causality on PHM12 is defined at the acquisition level: prediction at acquisition t uses the current acquisition and previously estimated HAML states, but no future acquisition.

B2. DECODER FEATURES

At each endpoint, HAML constructs an HI-centred 49-dimensional causal prognostic representation. The first 44 features $\mathbf{f}_t^{(\mathrm{HI},44)}$ are derived from the filtered HI trajectory: 30 recent HI values, three multi-scale slopes, three multi-scale deltas, five global trajectory summaries, two degradation-activity fractions, and one HI-rate-based RUL estimate. Five additional state variables (the raw graph coordinate, persistent degradation-velocity state, and three standardised latent coordinates) retain selected dynamic and geometric information that is not represented explicitly by the scalar HI trajectory.

Accordingly, the final decoder should be interpreted as HI-centred rather than HI-only: the filtered HI and its causal history provide the principal degradation-state representation, while the five augmentation variables compensate for information lost during scalar health-coordinate compression.

For notational simplicity, we suppress the asset index i on the state variables throughout this subsection and write s_t, h_t, and v_t for a single asset trajectory; asset-specific indexing is retained where required, e.g., in $t_i^{(0)}$.

These features are divided into seven groups:

Group 1 (30 HI history features): This is the largest part of the representation, comprising the 30 most recent HI: $[[h_{t-29}, h_{t-28}, \ldots, h_{t-1}, h_t]$

Group 2 (3 HI slope features): The slope quantifies the average rate of change of the HI over a temporal scale w. Since the HI is designed to be approximately monotonically increasing with degradation, a larger positive slope generally indicates a faster degradation rate. Three temporal scales: 10, 20, and 40 steps were used, helping distinguish short-term increases from persistent degradation trends.

$$\mathrm{Slope}_w(t) = \frac{[h_t - h_{t-s_w}]}{s_w}, \qquad s_w = \min(w, n_t - 1) \tag{B1}$$

n_t is the number of HI observations available up to the current endpoint.

Group 3 (3 HI delta features): The delta feature quantifies the increase of HI over a temporal scale w. In this study, three temporal scales: 10, 20, and 40 steps were used.

$$\Delta_w(t) = h_t - h_{t-s_w} \tag{B2}$$

Group 4 (5 Global HI summaries): This group includes *(i) Global delta:* measures the total accumulated change in the HI from the first observed HI value of the asset within the decoder sequence to the current endpoint, *(ii) Global endpoint slope:* measures the average rate of change of the HI by directly connecting the first observed point and the current endpoint of the trajectory, *(iii) Global least- squares slope:* This feature fits a least-squares linear regression over all observed HI values rather than connecting only the two endpoints. Compared with the endpoint slope, the least-squares slope is more robust to local fluctuations and noise because it uses the entire observed HI trajectory. *(iv) Normalised AUC:* measures the normalised area under the observed HI trajectory, with the integral approximated using the trapezoidal rule. For uniformly sampled steps, the normalised AUC is approximately equivalent to the time-averaged HI. It captures how long the asset has remained at a high degradation level, rather than considering only the HI value at the current endpoint, and *(v) Time since last increase (TSLI):* The code identifies HI increases satisfying $\Delta h > 10^{-6}$. Let stp_t denote the number of steps since the most recent HI increase; if no increase has occurred $stp_t = n_t - 1$. The resulting feature is therefore bounded in [0,1]. A value close to 0 indicates a recent HI increase, whereas a larger value indicates that the HI trajectory has remained on a plateau for a longer period.

$$\mathrm{GlobalDelta}(t) = h_t - h_{t_i^{(0)}} \tag{B3}$$

$$\mathrm{EndpointSlope}(t) = \frac{[h_t - h_{t_i^{(0)}}]}{\max(n_t - 1, 1)} \tag{B4}$$

$$b_{\mathrm{LS}}(t) = \frac{\sum_{\vartheta=t_i^{(0)}}^{t} (\vartheta - \bar{\vartheta})(h_\vartheta - \bar{h})}{\sum_{\vartheta=t_i^{(0)}}^{t} (\vartheta - \bar{\vartheta})^2} \tag{B5}$$

$$\mathrm{AUC}_{\mathrm{norm}}(t) = \frac{[\int_{t_i^{(0)}}^{t} h(\tau)\, d\tau]}{n_t - 1} \tag{B6}$$

$$\mathrm{TSLI}(t) = \frac{\min(\mathrm{stp}_t, 125)}{125} \tag{B7}$$

Group 5 (2 Positive-step features): These two features do not measure the magnitude of the HI increase; rather, they capture the frequency of recent positive HI changes. For example, two assets may have the same h_t and the same $Delta_{w(t)}$, yet exhibit different degradation dynamics: one may increase steadily over many steps, while the other may increase only once and then remain on a plateau. The positive-step fraction enables the decoder to distinguish between these two types of temporal behaviour. In this study, two recent steps, 20 and 40, were used to demonstrate the frequency of HI increases over the recent/medium-term and longer-term window.

$$\mathrm{PSF}_w = \frac{\#\{\Delta h > 10^{-6} \text{ in the most recent } w \text{ steps}\}}{\#\text{of steps}} \tag{B8}$$

Group 6 (R UL rate): This is a simple rate-based feature: if the current HI is h_t and the recent degradation rate is r, the estimated time required to reach the failure level HI=1 can be approximated by the remaining health margin divided by the degradation rate. Using up to the w most recent steps and $h_{recent} = \left[h_{\{t-s_w\}}, \ldots, h_t\right]$, estimate the speed by using the least-squares slope $r_t^{HI} = \max\left(LS_{slope(h_{recent})}, 0\right)$. With ε_{rate} set to 0.0005, the RUL rate is defined as:

$$\mathrm{RUL}_{\mathrm{rate}} = \min\left\{\max\left(\frac{1 - h_t}{\max(r_t^{\mathrm{HI}}, \varepsilon_{\mathrm{rate}})}, 0\right), 125\right\} \tag{B9}$$

Group 7 (Dynamic/geometric augmentation features): Five dynamic/geometric state variables augment the 44-dimensional HI-derived temporal representation in the final RUL decoder. The HI trajectory remains the organising core of the decoder, while these augmentation variables retain selected degradation momentum and latent-state information that may be lost when multidimensional

degradation structure is compressed into a scalar HI. In detail, the first feature is the raw HAML graph coordinate, i.e., the current geometric position between the healthy and failure reference regions:

$$s_t = \min\left\{\max\left(\frac{d_H(\mathbf{z}_t)}{\max[d_H(\mathbf{z}_t) + d_F(\mathbf{z}_t), \varepsilon_g]}, 0\right), 1\right\} \tag{B10}$$

where $d_H(\mathbf{z}_t)$ and $d_F(\mathbf{z}_t)$ are the graph-geodesic distances from the current state to the healthy reference set and the failure reference set, respectively, and ε_g prevents numerical division by zero, which is set to 10^{-8}. Eq. (B10) is the numerically safeguarded implementation of Eq. (7) in the main paper, using $\varepsilon_g = 10^{-8}$; under the positive-denominator assumptions used in Theorems 2–4, this safeguard is inactive.

The second feature is the normalised persistent degradation-velocity state maintained by the causal HI filter. It is an internal state variable of the causal persistent level-velocity filter. Because this state has memory, it represents degradation momentum carried from one step to the next. A single downward noisy observation does not immediately erase it.

$$v_t^{\text{norm}} = \min\left\{\max\left(\frac{v_t}{v_{\max}}, 0\right), 1\right\} \tag{B11}$$

The final three features are the standardised latent coordinates. After each sensor window is converted to temporal descriptors and standardised with training-fitted preprocessing, UMAP maps the descriptor vector $\mathbf{q}_t$ into a three-dimensional latent state: $\mathbf{z}_t = UMAP(\mathbf{q}_t) = [z_{t,1},\ z_{t,2}, z_{t,3}]^\top \in \mathbb{R}^3$. In this study, the dimension of the latent space is fixed to three before fitting UMAP; thus, the number three is the requested embedding dimension, not a dimensionality automatically inferred by UMAP. The coordinates are then standardised as $\widetilde{\mathbf{z}}_t$ by using Z-Score normalisation.

The final feature vector can be summarised as:

$$\boldsymbol{\xi}_t = \left[\underbrace{\mathbf{f}_t^{(\text{HI},44)}}_{\text{filtered HI history}},\ \underbrace{s_t}_{\text{currnent graph geometry}},\ \underbrace{v_t^{\text{norm}}}_{\text{filter dynamics}},\ \underbrace{\widetilde{\mathbf{z}}_t}_{\text{3 manifold coordinates}}\right] \in \mathbb{R}^{49} \tag{B12}$$

C. Structural and Stability Properties

This appendix establishes structural and stability properties of specific components of HAML. The results characterise the behaviour of the progress-aware spatial metric, the landmark-based health coordinate, and the causal HI filter under their stated assumptions. They are not intended to prove end-to-end statistical consistency, predictive optimality, or convergence of the complete sensor-to-RUL pipeline. In particular, results involving finite landmark approximation are conditional on the reference- distance approximation assumptions stated below.

From Eq. (2), we have:

$$w_{uv}^{(s)} = d_{uv}\left(1 + \lambda_G \mid p_u - p_v \mid\right), d_{uv} = \parallel \mathbf{z}_u - \mathbf{z}_v \parallel_2 \tag{C1}$$

where $p_u, p_v \in [0,1]$ are normalised training degradation progress values and $\lambda_G \geq 0$ is the progress-aware penalty strength. The edge-level properties below also apply to bridge edges whenever they use the same progress-aware cost. The path-level and shortest-path results are stated for the spatial subgraph.

Theorem 1: Let $\mathcal{G}_s = (V, E_s)$ denote the spatial subgraph of $\mathcal{G}_H = (V, E_s \cup E_t \cup E_b)$. For every spatial edge $(u,v) \in E_s$, let $d_{uv} = \parallel \mathbf{z}_u - \mathbf{z}_v \parallel_2 > 0$ be its unpenalized latent-space length and let $\Delta p_{uv} = \mid p_u - p_v \mid$ be its degradation-progress discrepancy. HAML assigns the multiplicative progress-aware cost

$$w_{uv}^{(s)} = d_{uv}\,(1 + \lambda_G \Delta p_{uv}) \tag{C2}$$

Then the following properties hold.

1. Spatial edges are never shortened

For every $(u, v) \in E_s$,

$$w_{uv}^{(s)} \geq d_{uv} \tag{C3}$$

Equality holds if either $\lambda_G = 0$ or $p_u = p_v$.

Thus, HAML leaves same-stage spatial relationships unchanged while increasing the traversal cost of spatial edges connecting different degradation stages.

2. Cross-stage edges are inflated proportionally to degradation discrepancy

If $\Delta p_{uv} \geq \delta_G > 0$, then

$$w_{uv}^{(s)} \geq (1 + \lambda_G \delta_G)\, d_{uv} \tag{C4}$$

Therefore, spatial neighbours separated by a larger lifecycle discrepancy receive a correspondingly larger multiplicative traversal penalty.

3. The accumulated cost of a spatial path includes a progress-dependent excess term

For any spatial path $\Pi = (u_0, u_1, \ldots, u_m)$ between landmarks u_0 and u_m, define its unpenalized length as

$$L_0(\Pi) = \sum_{r=0}^{m-1} d_{u_r u_{r+1}} \tag{C5}$$

and its HAML progress-aware length as

$$L_{\lambda_G}(\Pi) = \sum_{r=0}^{m-1} d_{u_r u_{r+1}} \left(1 + \lambda_G \mid p_{u_{r+1}} - p_{u_r} \mid\right) \tag{C6}$$

Then

$$L_{\lambda_G}(\Pi) \geq L_0(\Pi) \tag{C7}$$

If the finite spatial graph contains no zero-length edges and

$$d_{\min} = \min_{(u,v)\in E_s} d_{uv} > 0 \tag{C8}$$

then

$$L_{\lambda_G}(\Pi) \geq L_0(\Pi) + \lambda_G d_{\min} \mid p_{u_0} - p_{u_m} \mid \tag{C9}$$

Accordingly, any purely spatial path between lifecycle-distant states incurs an additional cost that grows with their endpoint degradation discrepancy.

Proof of Property 1

From Eq. C2, $w_{uv}^{(s)} = d_{uv}\,(1 + \lambda_G \Delta p_{uv})$. Since $\lambda_G \geq 0, \Delta p_{uv} \geq 0$, we have $1 + \lambda_G \Delta p_{uv} \geq 1$. Because $d_{uv} > 0$, $w_{uv}^{(s)} = d_{uv}\,(1 + \lambda_G \Delta p_{uv}) \geq d_{uv}$. Therefore, $w_{uv}^{(s)} \geq d_{uv}$

Equality requires $\lambda_G \Delta p_{uv} = 0$ which occurs when either $\lambda_G = 0$ or $p_u = p_v$ $\square$

Thus, the progress-aware weighting leaves same-progress spatial edges unchanged while increasing the traversal cost of edges connecting different degradation stages.

Proof of Property 2

Divide Eq. (C2) by $d_{uv} > 0$: $\frac{w_{uv}^{(s)}}{d_{uv}} = 1 + \lambda_G \mid p_u - p_v \mid$

Suppose $\mid p_u - p_v \mid \geq \delta_G$ then $1 + \lambda_G \mid p_u - p_v \mid \geq 1 + \lambda_G \delta_G$. Multiplying by d_{uv}, $w_{uv}^{(s)} \geq (1 + \lambda_G \delta_G)\, d_{uv}$

Thus, the penalty does not merely identify whether two stages differ; its strength increases continuously with their degradation-progress discrepancy.

Practical interpretation: Since HAML normalises progress to $p \in [0, 1]$, the multiplicative factor satisfies $1 \leq 1 + \lambda_G \mid p_u - p_v \mid \leq 1 + \lambda_G$. With the default experimental setting $\lambda_G = 4.5$, the multiplicative spatial-edge factor lies between 1 and 5.5, so:

- same-progress neighbours: factor 1;
- progress difference 0.2: factor 1.9;
- progress difference 0.5: factor 3.25;
- maximally separated progress 1.0: factor 5.5.

These values provide a direct interpretation of the multiplicative effect of the selected progress-penalty strength on spatial edge costs.

Proof of Property 3: Path-level result

Consider a path $\Pi = (u_0, \ldots, u_m)$. Its progress-aware length is : $L_{\lambda_G}(\Pi) = \sum_{r=0}^{m-1} w_{u_r u_{r+1}}^{(s)}$ Substituting Eq. (C2), $L_{\lambda_G}(\Pi) = \sum_{r=0}^{m-1} d_{u_r u_{r+1}} \left[1 + \lambda_G \mid p_{u_{r+1}} - p_{u_r} \mid\right]$

Expanding,

$$L_{\lambda_G}(\Pi) = \sum_{r=0}^{m-1} d_{u_r u_{r+1}} + \lambda_G \sum_{r=0}^{m-1} d_{u_r u_{r+1}} \mid p_{u_{r+1}} - p_{u_r} \mid$$

The first term is exactly the unpenalized path length, $L_0(\Pi) = \sum_{r=0}^{m-1} d_{u_r u_{r+1}}$

Therefore,

$$L_{\lambda_G}(\Pi) = L_0(\Pi) + \lambda_G \sum_{r=0}^{m-1} d_{u_r u_{r+1}} \mid p_{u_{r+1}} - p_{u_r} \mid \tag{C10}$$

Since every term in the second summation is non-negative, $L_{\lambda_G}(\Pi) \geq L_0(\Pi)$. This already establishes that progress-aware weighting cannot make a spatial route shorter.

Stronger endpoint-progress bound

Because HAML's fitted landmark graph is finite, suppose there are no duplicate zero-length spatial edges and define $d_{\min} = \min_{(u,v) \in E_s} d_{uv} > 0$. Then every spatial edge on Π satisfies: $d_{u_r u_{r+1}} \geq d_{\min}$

Hence,

$$\sum_{r=0}^{m-1} d_{u_r u_{r+1}} \mid p_{u_{r+1}} - p_{u_r} \mid \quad \geq d_{\min} \sum_{r=0}^{m-1} \mid p_{u_{r+1}} - p_{u_r} \mid \tag{C11}$$

Now apply the triangle inequality to the scalar progress sequence

$$\sum_{r=0}^{m-1} |p_{u_{r+1}} - p_{u_r}| \geq |\sum_{r=0}^{m-1} (p_{u_{r+1}} - p_{u_r})|$$

The inner sum telescopes: $\sum_{r=0}^{m-1} (p_{u_{r+1}} - p_{u_r}) = p_{u_m} - p_{u_0}$

Thus,

$$\sum_{r=0}^{m-1} |p_{u_{r+1}} - p_{u_r}| \geq |p_{u_m} - p_{u_0}| \tag{C12}$$

Combining Eqs. (C11) and (C12),

$$\sum_{r=0}^{m-1} d_{u_r u_{r+1}} |p_{u_{r+1}} - p_{u_r}| \geq d_{\min} |p_{u_m} - p_{u_0}|$$

Substituting into Eq. (C10), $L_{\lambda_G}(\Pi) \geq L_0(\Pi) + \lambda_G d_{\min} |p_{u_m} - p_{u_0}| \square$

Corollary for shortest-path distances on the spatial graph

For any (u, v) connected by at least one path $\Pi : u \to v$ in the spatial graph $\mathcal{G}_s$, define

$$d_{s,\lambda_G}(u, v) = \min_{\Pi:\, u \to v} L_{\lambda_G}(\Pi)$$

and $d_{s,0}(u, v) = \min_{\Pi: u \to v} L_0(\Pi)$

Since every spatial path satisfies

$$L_{\lambda_G}(\Pi) \geq L_0(\Pi) + \lambda_G d_{\min} |p_u - p_v|$$

Taking the minimum over all paths $\Pi : u \to v$ gives

$$d_{s,\lambda_G}(u, v) = \min_{\Pi} L_{\lambda_G}(\Pi) \geq \min_{\Pi} [L_0(\Pi) + \lambda_G d_{\min} |p_u - p_v|]$$

The second term is independent of the particular path Π, so

$$d_{s,\lambda_G}(u, v) \geq d_{s,0}(u, v) + \lambda_G d_{\min} |p_u - p_v| \tag{C13}$$

Hence, within the spatial subgraph and under the stated assumptions, the progress-aware shortest-path distance includes an additional lower-bound term that increases with endpoint progress discrepancy.

It is noted that Theorem 1 applies to spatial edges and to bridge edges when they use the same progress-aware cost. Temporal edges are excluded because they explicitly encode observed consecutive transitions of a training trajectory; movement in degradation progress along such edges is therefore admissible rather than a topological shortcut. Consequently, the theorem establishes penalisation of cross-stage spatial shortcuts, not a universal lower bound on every path in the complete heterogeneous HAML graph.

We next establish the local stability of the implemented OOS coordinate under a fixed landmark-neighbourhood assignment. This theorem does not claim global Lipschitz continuity across kNN-boundary changes, nor does it establish stability of the preceding mapping from raw sensors through the descriptor encoder and representation backend. Instead, it guarantees that within a fixed OOS landmark neighbourhood, small latent perturbations cannot produce arbitrarily large changes in the geometric health coordinate.

Theorem 2: Let $D_H(j)$ and $D_F(j)$ be the fixed landmark-to-reference graph distances defined during training, and $d_H(\mathbf{z})$, $d_F(\mathbf{z})$ denote the OOS query distances in Eqs. (5) and (6), respectively, with regions induced by the K_{OOS}-nearest-landmark structure. Consider a latent region Ω on which $\mathcal{N}_{K_{OOS}}(\mathbf{z})$ is unchanged. For Euclidean query-to-landmark attachment costs, both d_H and d_F are 1-Lipschitz on Ω. Let

$$\delta_\Omega = \inf_{\mathbf{z}\in\Omega} [d_H(\mathbf{z}) + d_F(\mathbf{z})] > 0$$

Then the raw health coordinate in Eq. (7),

$$s(\mathbf{z}) = \frac{d_H(\mathbf{z})}{d_H(\mathbf{z}) + d_F(\mathbf{z})}$$

satisfies

$$| \, s(\mathbf{z}) - s\,(\mathbf{z}') \, | \leq \frac{\| \, \mathbf{z} - \mathbf{z}' \, \|_2}{\delta_\Omega}, \mathbf{z}, \mathbf{z}' \in \Omega.$$

Hence HAML's landmark-based OOS health mapping is piecewise Lipschitz over the latent space, with regions induced by the K_{OOS}-nearest-landmark structure.

Proof: Suppose Eq. (5) in the main paper is of the form

$$d_H(\mathbf{z}) = \min_{j\in\mathcal{N}_{K_{OOS}}(\mathbf{z})} [\| \, \mathbf{z} - \mathbf{z}_j \, \|_2 + D_H(j)] \tag{C14}$$

and Eq. (6) is

$$d_F(\mathbf{z}) = \min_{j\in\mathcal{N}_{K_{OOS}}(\mathbf{z})} [\| \, \mathbf{z} - \mathbf{z}_j \, \|_2 + D_F(j)] \tag{C15}$$

where $\mathbf{z}_j$ is landmark j, $D_H(j), D_F(j)$ are fixed after training. The proof consists of three steps.

Step 1. Each landmark attachment function is 1-Lipschitz

For a fixed landmark j, define

$$f_j^H(\mathbf{z}) = \| \, \mathbf{z} - \mathbf{z}_j \, \|_2 + D_H(j) \tag{C16}$$

Because $D_H(j)$ is a constant, $| \, f_j^H(\mathbf{z}) - f_j^H\,(\mathbf{z}') \, | = | \| \, \mathbf{z} - \mathbf{z}_j \, \|_2 - \| \, \mathbf{z}' - \mathbf{z}_j \, \|_2 | \, .$

By the reverse triangle inequality, $| \| \, \mathbf{z} - \mathbf{z}_j \, \|_2 - \| \, \mathbf{z}' - \mathbf{z}_j \, \|_2 | \leq \| \, \mathbf{z} - \mathbf{z}' \, \|_2 \, .$

Therefore,

$$| \, f_j^H(\mathbf{z}) - f_j^H\,(\mathbf{z}') \, | \leq \| \, \mathbf{z} - \mathbf{z}' \, \|_2 \tag{C17}$$

and $f_j^H$is 1-Lipschitz. The same argument gives

$$f_j^F(\mathbf{z}) = \| \, \mathbf{z} - \mathbf{z}_j \, \|_2 + D_F(j)$$

and

$$| \, f_j^F(\mathbf{z}) - f_j^F\,(\mathbf{z}') \, | \leq \| \, \mathbf{z} - \mathbf{z}' \, \|_2 \tag{C18}$$

Step 2. The minimum over a fixed landmark set is also 1-Lipschitz

Within a region Ω in which $\mathcal{N}_{K_{OOS}}(\mathbf{z}) = \mathcal{N}$ does not change, Eq. (5) becomes $d_H(\mathbf{z}) = \min_{j\in\mathcal{N}} f_j^H(\mathbf{z})$

Let $j^* \in \arg\min_{j\in\mathcal{N}} f_j^H(\mathbf{z})$ then $d_H(\mathbf{z}) = f_{j^*}^H(\mathbf{z})$. At $\mathbf{z}'$, $d_H\,(\mathbf{z}') = \min_{j\in\mathcal{N}} f_j^H\,(\mathbf{z}') \leq f_{j^*}^H\,(\mathbf{z}')$

Hence

$$d_H(\mathbf{z}') - d_H(\mathbf{z}) \leq f_{j^*}^H(\mathbf{z}') - f_{j^*}^H(\mathbf{z})$$

From Eq. (C17),

$$d_H(\mathbf{z}') - d_H(\mathbf{z}) \leq \| \mathbf{z} - \mathbf{z}' \|_2$$

Swapping $\mathbf{z}$ and $\mathbf{z}'$gives

$$d_H(\mathbf{z}) - d_H(\mathbf{z}') \leq \| \mathbf{z} - \mathbf{z}' \|_2$$

Therefore,

$$| d_H(\mathbf{z}) - d_H(\mathbf{z}') | \leq \| \mathbf{z} - \mathbf{z}' \|_2 \tag{C19}$$

Thus d_H is 1-Lipschitz within a stable-neighbour region. Identically,

$$|d_F(\mathbf{z}) - d_F(\mathbf{z}')| \leq \| \mathbf{z} - \mathbf{z}' \|_2 \tag{C20}$$

Step 3. Propagate the OOS distance stability to Eq. (7)

For compactness, define $a = d_H(\mathbf{z}), b = d_F(\mathbf{z})$ and $a' = d_H(\mathbf{z}'), b' = d_F(\mathbf{z}')$. Let $Q = a + b, Q' = a' + b'$. Equation (7) gives $s(\mathbf{z}) = \frac{a}{Q}, s(\mathbf{z}') = \frac{a'}{Q'}$

Therefore,

$$| s(\mathbf{z}) - s(\mathbf{z}') | = | \tfrac{a}{Q} - \tfrac{a'}{Q'} | = \tfrac{|aQ' - a'Q|}{QQ'} \tag{C21}$$

Expanding the numerator,

$$\begin{aligned} aQ' - a'Q &= a(a' + b') - a'(a + b) \\ &= ab' - a'b \end{aligned} \tag{C22}$$

Add and subtract ab:

$$ab' - a'b = a(b' - b) + b(a - a')$$

Hence (C22) becomes,

$$aQ' - a'Q = a(b' - b) + b(a - a')$$

Applying the triangle inequality,

$$| aQ' - a'Q | \leq a | b' - b | + b | a - a' | \tag{C23}$$

From Eqs. (C19)–(C20),

$$| a - a' | \leq \| \mathbf{z} - \mathbf{z}' \|_2$$

and

$$| b - b' | \leq \| \mathbf{z} - \mathbf{z}' \|_2$$

Therefore, (C23) becomes,

$$\begin{aligned} \mid aQ' - a'Q \mid \quad &\leq (a+b) \parallel \mathbf{z} - \mathbf{z}' \parallel_2 \\ &= Q \parallel \mathbf{z} - \mathbf{z}' \parallel_2 \end{aligned} \tag{C24}$$

Substituting into Eq. (C21),

$$\mid s(\mathbf{z}) - s(\mathbf{z}') \mid \leq \frac{Q \parallel \mathbf{z} - \mathbf{z}' \parallel_2}{QQ'} = \frac{\parallel \mathbf{z} - \mathbf{z}' \parallel_2}{Q'} \tag{C25}$$

By assumption,

$$d_H(\mathbf{z}') + d_F(\mathbf{z}') \geq \delta_\Omega,$$

so

$$Q' \geq \delta_\Omega$$

Therefore,

$$\mid s(\mathbf{z}) - s(\mathbf{z}') \mid \leq \tfrac{1}{\delta_\Omega} \parallel \mathbf{z} - \mathbf{z}' \parallel_2 \tag{C26}$$

Thus S is locally Lipschitz continuous within every region in which the K_{OOS}-nearest-landmark set remains unchanged □.

We next quantify how reference-distance approximation error propagates to the graph-geodesic health coordinate. Let $a = d_H^*(\mathbf{z})$, $b = d_F^*(\mathbf{z})$, denote target healthy- and failure-reference distances under a chosen reference metric, and let $\hat{a} = d_{H,M}(\mathbf{z}), \hat{b} = d_{F,M}(\mathbf{z})$ denote their landmark approximations with $\mid \hat{a} - a \mid \leq \varepsilon_M, \mid \hat{b} - b \mid \leq \varepsilon_M$. The following result quantifies how bounded approximation error in the two reference distances propagates to the graph-geodesic health coordinate.

Theorem 3: Let $s^*(\mathbf{z}) = \frac{a}{a+b}, s_M(\mathbf{z}) = \frac{\hat{a}}{\hat{a}+\hat{b}}$. Suppose $a + b \geq \delta_0 > 0$, and $\mid \hat{a} - a \mid \leq \varepsilon_M, \mid \hat{b} - b \mid \leq \varepsilon_M$ with $2\varepsilon_M < \delta_0$. Then

$$\mid s_M(\mathbf{z}) - s^*(\mathbf{z}) \mid \leq \frac{\varepsilon_M}{\delta_0 - 2\varepsilon_M} \tag{C27}$$

Proof:

$$\mid s_M(\mathbf{z}) - s^*(\mathbf{z}) \mid = \mid \frac{\hat{a}}{\hat{a}+\hat{b}} - \frac{a}{a+b} \mid = \frac{\mid \hat{a}b - a\hat{b} \mid}{\left(\hat{a}+\hat{b}\right)(a+b)}$$

Because $\mid \hat{a} - a \mid \leq \varepsilon_M, \mid \hat{b} - b \mid \leq \varepsilon_M$

$$\mid \hat{a}b - a\hat{b} \mid = \mid \hat{a}b - ab + ab - a\hat{b} \mid = \mid b(\hat{a} - a) + a\left(b - \hat{b}\right) \mid \leq \mid b(\hat{a} - a) \mid + \mid a\left(b - \hat{b}\right) \mid \leq \leq \varepsilon_M(a+b).$$

Also, $\hat{a} + \hat{b} \geq a - \varepsilon_M + b - \varepsilon_M \geq \delta_0 - 2\varepsilon_M$.

Therefore, $\mid s_M(\mathbf{z}) - s^*(\mathbf{z}) \mid \leq \frac{\varepsilon_M}{\delta_0 - 2\varepsilon_M}$ □

The error term ε_M abstracts the approximation arising from finite landmark coverage, sparse graph construction, and restricted OOS landmark attachment. Theorem 3 is therefore an error-propagation result conditional on the reference-distance approximation bound; it does not establish that the representation backend recovers a ground-truth degradation geometry.

Theorem 4: Conditional random-landmark approximation of the health coordinate

Let $(\mathcal{M}, D_{\mathcal{M}})$ be a compact $d_{\mathcal{M}}$-dimensional Riemannian manifold with its geodesic metric, and let $\mathcal{Z}_M = \{\mathbf{z}_1, \ldots, \mathbf{z}_M\}$ consist of M landmarks drawn independently from a probability distribution μ on $\mathcal{M}$. Assume that μ admits a density π with respect to the Riemannian volume measure satisfying

$$\pi(\mathbf{z}) \geq \pi_{\min} > 0, \forall \mathbf{z} \in \mathcal{M} \tag{C28}$$

Define the landmark covering radius with distance $D_{\mathcal{M}}$

$$r_M = \sup_{\mathbf{z}\in\mathcal{M}} \min_{1\leq j\leq M} D_{\mathcal{M}}(\mathbf{z}, \mathbf{z}_j) \tag{C29}$$

Let $d_H^*(\mathbf{z})$ and $d_F^*(\mathbf{z})$ denote target healthy and failure-reference distances, and let $d_{H,M}(\mathbf{z})$ and $d_{F,M}(\mathbf{z})$ denote their landmark/graph/OOS approximations. Suppose that, for some constant $C_{dist} > 0$,

$$\sup_{\mathbf{z}\in\mathcal{M}} \mid d_{H,M}(\mathbf{z}) - d_H^*(\mathbf{z}) \mid \leq C_{dist} r_M \tag{C30}$$

and

$$\sup_{\mathbf{z}\in\mathcal{M}} \mid d_{F,M}(\mathbf{z}) - d_F^*(\mathbf{z}) \mid \leq C_{dist} r_M \tag{C31}$$

Further assume that the two target reference distances are uniformly separated from the singular case

$$\inf_{\mathbf{z}\in\mathcal{M}} [d_H^*(\mathbf{z}) + d_F^*(\mathbf{z})] \geq \delta_0 > 0 \tag{C32}$$

Define

$$s^*(\mathbf{z}) = \frac{d_H^*(\mathbf{z})}{d_H^*(\mathbf{z}) + d_F^*(\mathbf{z})} \qquad \text{and} \qquad s_M(\mathbf{z}) = \frac{d_{H,M}(\mathbf{z})}{d_{H,M}(\mathbf{z}) + d_{F,M}(\mathbf{z})} \tag{C33}$$

Then, for every failure probability $\eta_{prob} \in (0, 1)$, there exists a constant $C > 0$, independent of M, such that for sufficiently large M,

$$\Pr\left[\sup_{\mathbf{z}\in\mathcal{M}} \mid s_M(\mathbf{z}) - s^*(\mathbf{z}) \mid \leq C\left(\frac{\log(M/\eta_{prob})}{M}\right)^{1/d_{\mathcal{M}}}\right] \geq 1 - \eta_{prob} \tag{C34}$$

Proof:

Step 1. Local volume lower bound

Because $\mathcal{M}$ is compact and $d_{\mathcal{M}}$-dimensional, there exist constants $r_0 > 0, c_{vol} > 0$, such that for every $\mathbf{z} \in \mathcal{M}$ and every $0 < r \leq r_0$,

$$Vol\left(B_{\mathcal{M}}(\mathbf{z}, r)\right) \geq c_{vol} r^{d_{\mathcal{M}}} \tag{C35}$$

where $B_{\mathcal{M}}(\mathbf{z}, r) = \{\mathbf{y} \in \mathcal{M} : D_{\mathcal{M}}(\mathbf{z}, \mathbf{y}) < r\}$. This is the geometric ingredient that gives the $r^{d_{\mathcal{M}}}$ scaling. Because the sampling density satisfies $\pi \geq \pi_{\min}$,

$$\begin{aligned} \mu\left(B_{\mathcal{M}}(\mathbf{z}, r)\right) &= \textstyle\int_{B_{\mathcal{M}}(\mathbf{z},r)} \pi(\mathbf{y})\, d\left(Vol(\mathbf{y})\right) \\ &\geq \pi_{\min} Vol\left(B_{\mathcal{M}}(\mathbf{z}, r)\right) \geq \pi_{\min} c_{vol} r^{d_{\mathcal{M}}} \end{aligned} \tag{C36}$$

Step 2. Construct a finite $\mathbf{r/2}$-net

For $0 < r \leq r_0$, let $\{\{\mathbf{u}_1, \ldots, \mathbf{u}_{N_r}\}$ be an $r/2$-net of $\mathcal{M}$. Thus, for every $\mathbf{z} \in \mathcal{M}$, there exists some $\mathbf{u}_i$ such that

$$D_{\mathcal{M}}(\mathbf{z}, \mathbf{u}_i) \leq \tfrac{r}{2} \tag{C37}$$

Because $\mathcal{M}$ is compact and $d_{\mathcal{M}}$-dimensional, its covering number satisfies

$$N_r \leq C_{cov} r^{-d_{\mathcal{M}}} \tag{C38}$$

for a constant $C_{cov} > 0$. We prove this inequality by taking a maximal $r/2$-separated set. Then the balls $B_{\mathcal{M}}(\mathbf{u}_i, r/4)$ are pairwise disjoint. By Eq. (C35), $Vol\left(B_{\mathcal{M}}(\mathbf{u}_i, r/4)\right) \geq c_{vol}\left(\frac{r}{4}\right)^{d_{\mathcal{M}}}$. Therefore, $N_r c_{vol}\left(\frac{r}{4}\right)^{d_{\mathcal{M}}} \leq Vol(\mathcal{M})$, which implies

$$N_r \leq \frac{4^{d_{\mathcal{M}}} Vol(\mathcal{M})}{c_{vol}} r^{-d_{\mathcal{M}}} \tag{C39}$$

Thus, Eq. (C38) holds with an appropriate constant C_{cov}.

Step 3. Probability that a net ball contains no landmark

Consider one of the balls $B_{\mathcal{M}}(\mathbf{u}_i, r/2)$. From Eq. (C36), $\mu\left(B_{\mathcal{M}}(\mathbf{u}_i, r/2)\right) \geq \pi_{\min} c_{vol}\left(\frac{r}{2}\right)^{d_{\mathcal{M}}}$ Define

$$c_* = \frac{\pi_{\min} c_{vol}}{2^{d_{\mathcal{M}}}} \tag{C40}$$

Then

$$\mu\left(B_{\mathcal{M}}(\mathbf{u}_i, r/2)\right) \geq c_* r^{d_{\mathcal{M}}} \tag{C41}$$

Since the M landmarks are sampled independently, the probability that none falls inside this ball is $\left[1 - \mu\left(B_{\mathcal{M}}(\mathbf{u}_i, r/2)\right)\right]^M$. Using $1 - x \leq e^{-x}$, we obtain

$$\Pr\left[\mathcal{Z}_M \cap B_{\mathcal{M}}(\mathbf{u}_i, r/2) = \varnothing\right] \leq e^{-c_* M r^{d_{\mathcal{M}}}} \tag{C42}$$

Step 4. Uniform landmark coverage

Apply the union bound across the N_r net balls: $\Pr\left[\exists i : \mathcal{Z}_M \cap B_{\mathcal{M}}(\mathbf{u}_i, r/2) = \varnothing\right] \leq N_r e^{-c_* M r^{d_{\mathcal{M}}}}$ Using Eq. (C38),

$$\Pr\left[\exists i : B_{\mathcal{M}}(\mathbf{u}_i, r/2) \text{ is empty}\right] \leq C_{cov} r^{-d_{\mathcal{M}}} e^{-c_* M r^{d_{\mathcal{M}}}} \tag{C43}$$

Now suppose every $r/2$-net ball contains at least one landmark. For arbitrary $\mathbf{z} \in \mathcal{M}$, choose a net centre $\mathbf{u}_i$ satisfying $D_{\mathcal{M}}(\mathbf{z}, \mathbf{u}_i) \leq \frac{r}{2}$. Because its ball contains a landmark, there exists $\mathbf{z}_j \in \mathcal{Z}_M$ satisfying $D_{\mathcal{M}}(\mathbf{u}_i, \mathbf{z}_j) \leq \frac{r}{2}$. By the triangle inequality on the distance $D_{\mathcal{M}}$,

$$\begin{aligned} D_{\mathcal{M}}(\mathbf{z}, \mathbf{z}_j) &\leq D_{\mathcal{M}}(\mathbf{z}, \mathbf{u}_i) + D_{\mathcal{M}}(\mathbf{u}_i, \mathbf{z}_j) \\ &\leq \tfrac{r}{2} + \tfrac{r}{2} \\ &= r \end{aligned}$$

Since this is true for every $\mathbf{z}$,

$$r_M \leq r \tag{C44}$$

Therefore,

$$\Pr\left(r_M > r\right) \leq C_{cov} r^{-d_{\mathcal{M}}} e^{-c_* M r^{d_{\mathcal{M}}}} \tag{C45}$$

Step 5. Choose r as a function of M

Set

$$r = A\left(\frac{\log(M/\eta_{prob})}{M}\right)^{1/d_{\mathcal{M}}} \tag{C46}$$

where $A > 0$ is a sufficiently large constant. Then

$$r^{d_{\mathcal{M}}} = A^{d_{\mathcal{M}}} \frac{\log\left(M/\eta_{prob}\right)}{M}$$

and therefore

$$e^{-c_* M r^{d_{\mathcal{M}}}} = e^{-c_* A^{d_{\mathcal{M}}} \log(M/\eta_{prob})} = \left(\frac{M}{\eta_{prob}}\right)^{-c_* A^{d_{\mathcal{M}}}} \tag{C47}$$

Also,

$$r^{-d_{\mathcal{M}}} = \frac{M}{A^{d_{\mathcal{M}}} \log(M/\eta_{prob})} \tag{C48}$$

Substitution into Eq. (C45) gives

$$\Pr\left(r_M > r\right) \leq \frac{C_{cov} M}{A^{d_{\mathcal{M}}} \log(M/\eta_{prob})} \left(\frac{M}{\eta_{prob}}\right)^{-c_* A^{d_{\mathcal{M}}}} \tag{C49}$$

Let $\kappa = c_* A^{d_{\mathcal{M}}}$. Choose A sufficiently large that

$$\kappa \geq 2 \tag{C50}$$

Then $M^{1-\kappa} \leq M^{-1}$, and, since $0 < \eta_{prob} < 1$, we have $\eta_{prob}^{\kappa} \leq \eta_{prob}$

Hence, for sufficiently large M,

$$\Pr\left(r_M > r\right) \leq \eta_{prob} \tag{C51}$$

Consequently, with probability at least $1 - \eta_{prob}$,

$$r_M \leq A\left(\frac{\log(M/\eta_{prob})}{M}\right)^{1/d_{\mathcal{M}}} \tag{C52}$$

This is the landmark covering-radius rate.

Step 6. Convert landmark coverage into reference-distance error

On the high-probability event in Eq. (C52),

$$\begin{aligned} \sup_{\mathbf{z}} \mid d_{H,M}(\mathbf{z}) - d_H^*(\mathbf{z}) \mid \quad &\leq C_{dist} r_M \\ &\leq C_{dist} A\left(\frac{\log(M/\eta_{prob})}{M}\right)^{1/d_{\mathcal{M}}} \end{aligned}$$

and identically,

$$\sup_{\mathbf{z}} \mid d_{F,M}(\mathbf{z}) - d_F^*(\mathbf{z}) \mid \leq C_{dist} A \left(\frac{\log(M/\eta_{prob})}{M} \right)^{1/d_{\mathcal{M}}} \tag{C53}$$

Define

$$\varepsilon_M = C_{dist} A \left(\frac{\log(M/\eta_{prob})}{M} \right)^{1/d_{\mathcal{M}}} \tag{C54}$$

Thus

$$\mid d_{H,M}(\mathbf{z}) - d_H^*(\mathbf{z}) \mid \leq \varepsilon_M, \mid d_{F,M}(\mathbf{z}) - d_F^*(\mathbf{z}) \mid \leq \varepsilon_M$$

uniformly over $\mathcal{M}$.

Step 7. Apply Theorem 3

From Eq. (C32), $d_H^*(\mathbf{z}) + d_F^*(\mathbf{z}) \geq \delta_0$ uniformly. Because $\varepsilon_M = O\left[\left(\frac{\log(M/\eta_{prob})}{M} \right)^{1/d_{\mathcal{M}}} \right] \to 0$, there exists M_0 such that, for every $M \geq M_0$,

$$\varepsilon_M \leq \tfrac{\delta_0}{4}. \tag{C55}$$

Hence $2\varepsilon_M < \delta_0$, so the conditions of Theorem 3 are satisfied. Theorem 3 gives

$$\mid s_M(\mathbf{z}) - s^*(\mathbf{z}) \mid \leq \tfrac{\varepsilon_M}{\delta_0 - 2\varepsilon_M} \tag{C56}$$

Using Eq. (C55), $\delta_0 - 2\varepsilon_M \geq \delta_0 - \frac{\delta_0}{2} = \frac{\delta_0}{2}$ Therefore,

$$\mid s_M(\mathbf{z}) - s^*(\mathbf{z}) \mid \leq \tfrac{2\varepsilon_M}{\delta_0} \tag{C57}$$

Substituting Eq. (C54),

$$\mid s_M(\mathbf{z}) - s^*(\mathbf{z}) \mid \leq \frac{2C_{dist} A}{\delta_0} \left(\frac{\log\left(M/\eta_{prob}\right)}{M} \right)^{1/d_{\mathcal{M}}}$$

Because all preceding bounds are uniform over $\mathbf{z}$,

$$\sup_{\mathbf{z} \in \mathcal{M}} \mid s_M(\mathbf{z}) - s^*(\mathbf{z}) \mid \leq \tfrac{2C_{dist} A}{\delta_0} \left(\tfrac{\log(M/\eta_{prob})}{M} \right)^{1/d_{\mathcal{M}}} \tag{C58}$$

Finally, define

$$C = \tfrac{2C_{dist} A}{\delta_0} \tag{C59}$$

Therefore, with probability at least $1 - \eta_{prob}$, $\sup_{\mathbf{z} \in \mathcal{M}} \mid s_M(\mathbf{z}) - s^*(\mathbf{z}) \mid \leq C \left(\frac{\log(M/\eta_{prob})}{M} \right)^{1/d_{\mathcal{M}}}$

Theorem 4 separates two sources of approximation. The probabilistic argument establishes that i.i.d. landmarks cover a compact $d_{\mathcal{M}}$-dimensional latent support at rate $O\left(\left(\log\left(M/\eta_{prob}\right)/M \right)^{1/d_{\mathcal{M}}} \right)$. The theorem then assumes that the healthy- and failure-reference distances produced by the graph/OOS construction approximate their target counterparts at the same order; under this condition, Theorem 3 shows that the graph-geodesic health coordinate inherits the landmark-coverage rate. Thus, Theorem 4 is a conditional coordinate-stability result, rather than a proof that finite landmark coverage alone guarantees convergence of the complete HAML graph metric.

Theorem 4 assumes i.i.d. random landmark sampling and therefore provides a reference finite-sampling result rather than a convergence guarantee for HAML's implemented asset-/condition-balanced landmark selection. The practical effects of landmark budget and sampling strategy are evaluated empirically.

Structural Properties of the Causal Monotonic HI Filter

For a single asset trajectory, define the filter state by

$$\hat{v}_t = \rho v_{t-1}, \hat{h}_t = h_{t-1} + \hat{v}_t, e_t = s_t - \hat{h}_t \tag{C60}$$

$$u_t = [e_t - \tau_n]_+, \qquad v_t = \Pi_{[0,v_{\max}]}(\hat{v}_t + \beta u_t) \tag{C61}$$

and

$$\widetilde{h}_t = \hat{h}_t + \alpha_t e_t, \alpha_t = \begin{cases} \alpha^+, & e_t \geq 0 \\ \alpha^-, & e_t < 0 \end{cases} \tag{C62}$$

Let the velocity-aware candidate be

$$c_t = \begin{cases} \max\left\{\widetilde{h}_t, h_{t-1} + v_t\right\}, & u_t > 0 \\ \widetilde{h}_t, & u_t = 0 \end{cases} \tag{C63}$$

and define

$$U_t = \min\{h_{t-1} + \Delta_{\max}, 1\}, \mathcal{I}_t = [h_{t-1}, U_t]$$

The final HI update is

$$h_t = \Pi_{\mathcal{I}_t}(c_t) \tag{C64}$$

Assume throughout that

$$0 \leq \rho \leq 1, \beta \geq 0, \tau_n \geq 0, 0 \leq \alpha^-, \alpha^+ \leq 1,$$

with $v_{\max} \geq 0$, $\Delta_{\max} \geq 0$, $h_0 \in [0,1]$, and $v_0 \in [0, v_{\max}]$.

Theorem 5: Causality, feasibility, and irreversible bounded evolution For any sequence of raw geometric coordinates $\{s_t\}_{t\geq 1}$, the HAML filter defined by Eqs. (C60)– (C64) satisfies, for every t,

$$0 \leq v_t \leq v_{\max} \tag{C65}$$

and

$$0 \leq h_t \leq 1 \; and \; 0 \leq h_t - h_{t-1} \leq \min\{\Delta_{\max}, 1 - h_{t-1}\} \tag{C66}$$

Consequently,

$$0 \leq h_t - h_{t-1} \leq \Delta_{\max} \tag{C67}$$

so $\{h_t\}$ is non-decreasing and bounded. The sequence therefore converges to a limit $h_\infty \in [0,1]$. Furthermore, the state at step t depends only on $s_{1:t}$, the previous filter states, and parameters fixed from training; hence the filter is temporally causal.

Proof:

From Eq. (C61), $v_t = \Pi_{[0,v_{\max}]}(\hat{v}_t + \beta u_t)$. Projection onto $[0, v_{\max}]$ immediately gives $0 \leq v_t \leq v_{\max}$. For the level state, Eq. (C64) gives $h_t = \Pi_{\mathcal{I}_t}(c_t)$, where $\mathcal{I}_t = [h_{t-1}, U_t], U_t = \min\{h_{t-1} + \Delta_{\max}, 1\}$.

Because $h_{t-1} \in [0,1]$ and $\Delta_{\max} \geq 0$, $h_{t-1} \leq U_t \leq 1$, so $\mathcal{I}_t$ is non-empty. Projection therefore yields $h_{t-1} \leq h_t \leq U_t = \min\{h_{t-1} + \Delta_{\max}, 1\}$.

Subtracting h_{t-1} gives $0 \leq h_t - h_{t-1} \leq \min\{\Delta_{\max}, 1 - h_{t-1}\}$, which proves Eqs. (C66)-(C67). Starting from $h_0 \in [0,1]$, induction establishes $h_t \in [0,1]$ for every t.

Because $h_t \geq h_{t-1}$, the sequence is non-decreasing, while $h_t \leq 1$ gives an upper bound. The monotone convergence theorem for real sequences therefore implies the existence of $h_\infty \in [0,1]$. Finally, Eqs. (C60)–(C64) use only the current observation s_t, the previous states h_{t-1}, v_{t-1}, and fixed parameters. No future observation enters the update, establishing causality.

Theorem 5 verifies that the implemented filter satisfies its designed temporal causality and trajectory constraints; it does not establish that the resulting HI is statistically optimal or uniquely corresponds to the underlying physical degradation process.

Corollary 5.1: Absorbing failure boundary

If $h_{t-1} = 1$, then $\mathcal{I}_t = [1,1]$, and hence $h_t = 1$. Thus, once the HI reaches its normalised failure boundary, subsequent observations cannot move it back toward a healthier state.

Corollary 5.2: Bounded total variation

Since h_t is non-decreasing,

$$\sum_{t=1}^{T} |h_t - h_{t-1}| = h_T - h_0 \leq 1 \tag{C68}$$

Thus, in addition to the local step bound $\Delta_{\max}$, the complete filtered trajectory has bounded total variation. This provides a direct mathematical explanation for why low step-to-step variation is expected under the HAML filter; consequently, empirical mean-HI-step values should be interpreted as behaviour within the imposed bounded-step constraint rather than as an unconstrained measure of representation smoothness.

Proposition 6: Convex optimisation interpretation of the level update

For fixed $\hat{h}_t$, s_t, and $\alpha_t \in [0,1]$, the asymmetric correction

$$\widetilde{h}_t = (1 - \alpha_t)\,\hat{h}_t + \alpha_t s_t \tag{C69}$$

is the unique minimizer of

$$\widetilde{h}_t = \arg\min_{h \in \mathbb{R}} \left\{ \tfrac{1-\alpha_t}{2} \left(h - \hat{h}_t\right)^2 + \tfrac{\alpha_t}{2} (h - s_t)^2 \right\} \tag{C70}$$

Furthermore, conditional on the velocity-aware candidate c_t, the final HAML update is the unique solution of

$$h_t = \arg\min_{h \in \mathcal{I}_t} \tfrac{1}{2} (h - c_t)^2 \tag{C71}$$

Hence, the final update is the Euclidean projection of the corrected state onto the monotone and bounded-step feasible interval defined by the HAML degradation-state constraints.

Proof:

Define $J_t(h) = \tfrac{1-\alpha_t}{2} \left(h - \hat{h}_t\right)^2 + \tfrac{\alpha_t}{2} (h - s_t)^2$. Differentiating,

$$J_t'(h) = (1 - \alpha_t) \left(h - \hat{h}_t\right) + \alpha_t (h - s_t)$$

Setting $J_t'(h) = 0$ gives

$$h = (1 - \alpha_t)\,\hat{h}_t + \alpha_t s_t = \hat{h}_t + \alpha_t \left(s_t - \hat{h}_t\right) = \hat{h}_t + \alpha_t e_t$$

which is exactly Eq. (C62). Moreover,

$$J_t''(h) = 1 > 0$$

so the solution is unique.

For the final level update, consider the feasible interval $\mathcal{I}_t = [h_{t-1}, U_t], U_t = \min\{h_{t-1} + \Delta_{\max}, 1\}$. Conditional on the candidate level c_t, consider the constrained optimization problem

$$h_t^\star = \arg\min_{h \in \mathcal{I}_t} \tfrac{1}{2}(h - c_t)^2 \tag{C71}$$

The objective is strictly convex since its second derivative with respect to $h$is 1, so the minimizer is unique. Without the interval constraint, the minimizer is $h = c_t$. If $c_t \in \mathcal{I}_t$, it remains the constrained minimizer. If $c_t < h_{t-1}$, the closest feasible point is the lower boundary h_{t-1}; if $c_t > U_t$, the closest feasible point is the upper boundary U_t. Therefore,

$$h_t^\star = \begin{cases} h_{t-1}, & c_t < h_{t-1} \\ c_t, & h_{t-1} \le c_t \le U_t \\ U_t, & c_t > U_t \end{cases}$$

or equivalently,

$$h_t^\star = \min\{\max\{c_t, h_{t-1}\}, U_t\} = \Pi_{\mathcal{I}_t}(c_t)$$

By Eq. (C64), this is exactly the implemented HAML level update.

Hence, conditional on the candidate c_t, the final HI update is the unique Euclidean projection onto the constraint-defined feasible set $\mathcal{I}_t$ □.

Importantly, the filter is not claimed to solve or approximate a mechanistic physical degradation model. Rather, each level update has an exact local convex-projection interpretation in which the geometric observation is combined with a persistent state prediction and projected onto a feasible set encoding the assumed irreversible, bounded-step degradation behaviour.

Theorem 7: Dead-zone suppression and persistence of degradation momentum

Suppose $v_{t-1} \in [0, v_{\max}]$. Then the velocity update satisfies

$$0 \le v_t - \rho v_{t-1} \le \beta [e_t - \tau_n]_+ \tag{C72}$$

In particular, if $e_t \le \tau_n$, then

$$v_t = \rho v_{t-1} \tag{C73}$$

If the condition $e_{t+j} \le \tau_n$ holds for m consecutive steps, then

$$v_{t+m} = \rho^m v_t \tag{C74}$$

If instead $e_t > \tau_n$and the upper velocity bound is inactive, then

$$v_t = \rho v_{t-1} + \beta (e_t - \tau_n) \tag{C75}$$

Proof:

Since $u_t = [e_t - \tau_n]_+ \ge 0$, the unprojected velocity update satisfies $\hat{v}_t + \beta u_t \ge \hat{v}_t = \rho v_{t-1}$. Because $0 \le \rho \le 1$ and $v_{t-1} \le v_{\max}$, we have $0 \le \rho v_{t-1} \le v_{\max}$.

Therefore, upper clipping cannot reduce v_t below ρv_{t-1}, giving $v_t - \rho v_{t-1} \ge 0$.

Since $u_t \geq 0$, $\beta \geq 0$, and $0 \leq \rho v_{t-1} \leq v_{\max}$, the argument of the projection is non-negative. Hence, the lower clipping boundary is inactive and:

$$v_t = \min\left\{\rho v_{t-1} + \beta u_t, v_{\max}\right\}$$

Therefore,

$$v_t - \rho v_{t-1} = \min\left\{\beta u_t, v_{\max} - \rho v_{t-1}\right\}$$

Both terms in the minimum are non-negative, which gives

$$v_t - \rho v_{t-1} \geq 0$$

Moreover, the minimum cannot exceed its first argument, so

$$v_t - \rho v_{t-1} \leq \beta u_t = \beta\left[e_t - \tau_n\right]_+$$

Consequently,

$$0 \leq v_t - \rho v_{t-1} \leq \beta\left[e_t - \tau_n\right]_+$$

Thus, the velocity cap can only reduce the amount of newly introduced reinforcement; it can never amplify it.

When $e_t \leq \tau_n$, $u_t = 0$, and hence

$$v_t = \Pi_{[0,v_{\max}]}\left(\rho v_{t-1}\right) = \rho v_{t-1}$$

which proves Eq. (C73). Repeated application over m consecutive sub-threshold steps gives Eq. (C74). If $e_t > \tau_n$ and the velocity cap is inactive, projection does not modify the update, yielding Eq. (C75). □

D. Evaluation Details

D1. Datasets and Processing

The C-MAPSS aircraft engines dataset characterises the degradation behaviour of aircraft turbofan engines under varying operating conditions (Saxena et al., 2008). It comprises multivariate time-series data collected from 21 sensors that capture key physical measurements, including temperature, pressure, and rotational speeds. Consistent with prior work Wang et al. (2025b), sensors exhibiting constant readings, specifically sensors 1, 5, 6, 10, 16, 18, and 19, are excluded in HAML and GNN-based methods as they do not contribute useful degradation information. The four subsets combine one or six operating conditions with one or two fault modes: FD001 contains one condition and one fault mode, FD002 six conditions and one fault mode, FD003 one condition and two fault modes, and FD004 six conditions and two fault modes. As each trajectory spans the full operational life of an engine from healthy state to failure, training samples are generated using a sliding window approach to cover the full duration of the lifecycle.

For an asset i with a lifespan T_i, a fixed-length window of W steps slides along the sensor trajectory with stride one. Each sample corresponds to a window ending at time t, with its RUL defined as $y_{i,t} = \min\left\{T_i - t, R_{\max}\right\}$. To better reflect practical degradation patterns and stabilise learning, a piecewise linear RUL formulation is adopted (Wang et al., 2025b), where RUL values exceeding a predefined threshold are capped. In this work, the maximum RUL is set to 125 cycles (Wang et al., 2023). Finally, all sensor measurements are scaled using min-max normalisation to the range [0, 1], ensuring consistent feature magnitudes across operating conditions (Wang et al., 2025b). This preprocessing pipeline enables robust learning of degradation trends and facilitates benchmarking across diverse prognostics models.

The PHM12 bearing dataset records accelerated run-to-failure experiments conducted on the PRONOSTIA platform developed by the FEMTO-ST Institute (Nectoux et al., 2012). The dataset

comprises three operating conditions with different rotational speeds and radial loads. Conditions 1 and 2 contain seven bearings each, while Condition 3 contains three bearings. For each bearing, horizontal and vertical vibration signals, together with temperature measurements, were recorded throughout the degradation process.

Following prior RUL studies (Wei et al., 2023; Wang et al., 2025b), we retain only the horizontal vibration signal, as it provides the primary degradation information used for RUL prediction. The vibration signals were sampled at 25.6 kHz over 0.1-s acquisition intervals, yielding 2,560 points per vibration segment. Min–max normalisation fitted on the training bearings was applied to the input signals and reused unchanged for validation and test bearings. Following the benchmarking protocol of Wang et al. (Wang et al., 2025b) and the prior PHM12 studies on which it is based (Wei et al., 2023), RUL targets are normalised independently for each bearing because bearing lifetimes vary substantially across runs. For bearing b, let $R_{b,t}$ denote the ground-truth RUL at sample t and $R_b^{\max}$ its maximal RUL over the benchmark trajectory. The normalised target is $y_{b,t} = R_{b,t}/R_b^{\max}$. Thus, all bearings are evaluated on a common normalised degradation scale despite their different absolute lifetimes.

Table D1. Statistics of the C-MAPSS and PHM12 datasets. W denotes the input sequence length for C-MAPSS, while for PHM12, $2{,}560 \times 1$ denotes the raw acquisition; the model-side representation is 50×24, and k is the number of input channels. For C-MAPSS, # Test denotes the number of testing engine units; for PHM12, it denotes the number of testing samples for the specified bearing. For PHM12, Max RUL is the bearing-specific benchmark scaling value used to normalise ground-truth RUL targets and, where required, convert normalised errors to absolute benchmark RUL units during evaluation; it is not an input to the predictive model.

Dataset	Subset / Condition	Test ID	# Train	# Test	Max RUL	**W**	k
C-MAPSS	FD001	–	15,731	100	125	50	14
	FD002	–	41,019	259	125	50	14
	FD003	–	19,820	100	125	50	14
	FD004	–	49,048	248	125	50	14
PHM12	Cond. 1	B3	3,674	1,802	2,375	2,560	1
		B4	–	1,139	1,173	2,560	1
		B5	–	2,302	2,463	2,560	1
		B6	–	2,302	2,448	2,560	1
		B7	–	1,502	2,259	2,560	1
	Cond. 2	B3	1,708	1,202	1,955	2,560	1
		B4	–	612	751	2,560	1
		B5	–	2,002	2,311	2,560	1
		B6	–	572	701	2,560	1
		B7	–	172	230	2,560	1
	Cond. 3	B3	2,152	352	434	2,560	1

Importantly, $R_b^{\max}$ is part of the benchmark ground-truth construction and evaluation protocol and is not supplied to HAML or the baseline models as an input feature during test-time prediction. Models generate predictions from the observed vibration segment alone on the normalised target scale. Bearing-specific $R_b^{\max}$ is used afterwards only when normalised prediction errors are expressed in absolute RUL units for benchmark scoring.

Specifically, the first two bearings under each operating condition are used for training, while the remaining bearings are used for testing. Consequently, Bearings 3-7 are evaluated under Conditions 1 and 2, and Bearing 3 is evaluated under Condition 3. The numbers of training and testing samples and the bearing-specific maximal RUL values used by the benchmark for target normalisation and step-scale scoring are summarised in Table D1.

D2. BASELINES AND PARAMETER SETTINGS

We compare HAML against ten representative benchmark models spanning diverse methodological categories, including LSTM, GRU, a Transformer-based approach (FAT) (Vu et al., 2024), graph neural network methods (LOGO (Wang et al., 2023), FC-STGNN (Wang et al., 2024), and STGNN (Liu et al., 2023), and time series forecast methods (TimesNet (Wu et al., 2023), TimeMixer ++ (Wang et al., 2025a), PatchTST (Nie et al., 2022), and iTransformer (Liu et al., 2024). For all baselines, hyperparameters were set following the configurations reported in the original publications and the survey study (Wang et al., 2025b) to ensure fair comparison. In particular, LOGO and FC-STGNN require dataset-specific tuning; their hyperparameters were therefore optimized for each experimental dataset in accordance with the procedures described in their respective papers. The detailed configurations of all benchmark models are provided in the tables below.

Raw PHM12 input is one 2,560-point horizontal-vibration acquisition. After train-fitted preprocessing, each acquisition is transformed into 50 intra-acquisition patches with 24 feature channels per patch, yielding a $[50, 24]$ model input. No previous or future acquisition is included in the raw model input. Thus, 50 means causal cycles for C-MAPSS but intra-acquisition patches for PHM12. The same training-fitted $[50, 24]$ PHM12 adapter output is supplied to all applicable baseline models; HAML subsequently applies its seven-descriptor encoder across the 24 feature channels.

For each PHM12 operating condition, Bearings 1 and 2 form the predefined training set. For models requiring validation-based model selection or early stopping, Bearing 1 is used for fitting and Bearing 2 for validation; validation data are excluded from preprocessing fitting. After model selection, preprocessing is refitted using Bearings 1 and 2, and the model is reinitialised and refitted on the same full training set before evaluation on the held-out test bearings. Models using fixed training schedules are trained directly on both training bearings.

Table D2. Key parameters of LSTM and GRU used in the experiments

Parameter	LSTM	GRU	Description
Recurrent cell	LSTM	GRU	Only architectural difference between the two baselines.
RNN layer 1 hidden size	32	32	1 recurrent layer.
RNN layer 2 hidden size	64	64	1 recurrent layer.
Dropout	0.1	0.1	Applied after recurrent stack.
Dense head	$64 \to 8 \to 8 \to 1$	$64 \to 8 \to 8 \to 1$	ReLU after the first two dense layers.
Epochs	60	60	No early stopping in the supplied reproduction runner.
Batch size	128	128	Training batch size.
Learning rate	1×10^{-3}	1×10^{-3}	Adam.
Optimizer	Adam	Adam	Default Adam settings.
Loss	MSE	MSE	Uses the dataset-specific target normalisation specified below
Gradient clipping	1.0	1.0	Clip gradient norm; ≤ 0 would disable.
Target normalisation	True	True	C-MAPSS: capped RUL/125; PHM12: bearing-normalised RUL defined in Appendix D above.

Table D3. Key parameters of FAT used in the experiments

Parameter	Value	Description
Attention layer types	[log, auto]	Alternating/combined attention layer types from supplied baseline.
Dimension of model	128	Model hidden dimension.
#heads	2	Attention heads.
#layers	2	Transformer layers.
dropout	0.1	Network dropout.
Learning rate	0.001	Adam learning rate.
Batch size	128	Training batch size.
Max epochs	100	Validation selects epoch count; final model refit on full training for selected epochs.
Early stopping patience	40	Validation patience.
LR scheduler patience	10	Reduce-on-plateau patience.
LR factor	0.3	Learning-rate reduction factor.
Causal kernel size	3	Causal temporal convolution kernel.
Value kernel size	1	Value projection kernel.
Auto moving avg	15	Moving-average size in auto component.
L	3	Baseline-specific transform parameter.
Modes	32	Number of modes.
Mode select	random	Mode selection method.
Optimizer / Loss	Adam / MSE	No explicit weight decay in fair runner.

Table D4. Key parameters of LOGO used in the experiments

Common parameter	Value			
Epochs	81			
Batch size	100			
Learning rate	1×10^{-3}			
Weight decay	1×10^{-4}			
Optimizer	Adam			
Loss	MSE + $\theta \times$ Graph_Loss			
Window length	50			
#nodes	14 (for C-MAPSS), 24 (for PHM12)			
Dataset	θ (Graph Loss)	Patch size	#patch	Hidden Dimension
FD001	0.001	10	5	8
FD002	0.01	2	25	6
FD003	0.01	10	5	32
FD004	0.001	10	5	10
PHM12-C1	0.001	10	5	8
PHM12-C2	0.01	10	5	32
PHM12-C3	0.001	10	5	10

Table D5. Key parameters of STGNN used in the experiments

Parameter	Value
Epochs	81
Batch size	100
Learning rate	1×10^{-3}
Weight decay	1×10^{-4}
Optimizer / Loss	Adam / MSE
Patch size	50
#patch	1
#nodes	14 (for C-MAPSS), 24 (for PHM12)
Hidden dimension	64
K	3
Top k	10

Table D6. Key parameters of FC-STGNN used in the experiments

Common parameter	Value
Epochs	81
Batch size	100
Learning rate	1×10^{-3}
Weight decay	1×10^{-4}
Optimizer / Loss	Adam / MSE
Pooling choice	Mean value
Decay	0.7
Moving window	[2,2]
MPNN stride	[1,2]

Dataset	Patch	#patch	Enc_time_out	Enc_hidden	Enc_out	Kernel	Hidden	#Seq	#Windows
FD001	25	2	27	8	32	2	8	6	2
FD002	1	50	3	8	12	2	8	10	74
FD003	1	50	3	8	6	2	24	25	74
FD004	2	25	4	8	6	2	8	10	36
PHM12-C1	25	2	27	8	32	2	8	6	2
PHM12-C2	2	25	4	8	6	2	8	10	36
PHM12-C3	25	2	27	8	32	2	8	6	2

Enc_time_out: temporal resolution after patch encoder; Enc_hidden: width inside encoder.

Enc_out: dimension of representation from patch encoder; Kernel: temporal convolution receptive field at encoder.

Hidden: graph hidden representation width; #Seq: Configure the number of sequential graph units processed in the downstream graph block.

#Windows: Number of moving graph windows that fully-connected graph processing must process; Decay: The decay rate of cross-time connection strength with increasing temporal distance.

Moving window and MPNN stride: receptive field / downsampling schedule of moving-pooling graph convolution; Pooling choice: the way to aggregate node features.

The following tables report the specific configurations used for time series forecast models including TimesNet, TimeMixer++, PatchTST, iTransformer.

Table D7. Common optimisation settings used in modern time-series baselines

Parameter	Value	Description
Input	50x14 for C-MAPSS, 50x24 for PHM12	Sequence length $\times$ features.
Loss	MSE	Applied to normalised RUL.
Optimizer	Adam	Common optimiser for all four additional neural baselines.
Learning rate	1×10^{-3}	Initial learning rate.
Weight decay	0	No explicit weight decay.
Batch size	128	Training and evaluation batch size.
Maximum epochs	100	Upper bound during asset-disjoint model selection.
Validation split	20% TRAIN engines for C-MAPSS; Bearing 1 for training, Bearing 2 for validation for PHM12	Asset-disjoint; validation assets do not fit preprocessing.
Early-stopping patience	40	Selection stops after 40 stale epochs.
Selection min delta	1×10^{-4}	Required RMSE improvement for resetting patience.
LR scheduler	ReduceLROnPlateau	Patience = 10; factor = 0.3.
Gradient clipping	1.0	Global gradient-norm clipping.
Final training	Full-TRAIN refit	Model is reinitialised and trained for the validation-selected epoch count; the validation-derived LR schedule is replayed.

Table D8. The key parameters of TimesNet used in the experiments

Parameter	Value	Description
Implementation	THUML Time-Series-Library	TimesNet model implementation.
Output dimension	1	One-output head used for scalar RUL regression.
d_model	64	Model hidden dimension.
d_ff	128	Feed-forward dimension.
# encoder layers	2	Number of TimesNet encoder blocks.
# attention heads	4	Shared configuration field used by the library.
Top-k periods	3	Top periodic components retained.
# kernels	4	Number of Inception-style kernels.
Dropout	0.1	Network dropout.
Activation	GELU	Nonlinear activation.
Factor	3	Library attention/configuration factor.
Moving average	25	Decomposition moving-average parameter in the shared config.
Normalisation	Enabled	use_norm = 1.

Table D9. The key parameters of TimeMixer++ used in the experiments

Parameter	Value	Description
Implementation	PyPOTS BackboneTimeMixerPP	Maintained TimeMixer++ implementation.
Output dimension	1	One-output head used for scalar RUL regression.
# layers	2	TimeMixer++ backbone depth.
d_model	64	Hidden representation dimension.
d_ffn	128	Feed-forward network dimension.
# heads	4	Attention heads.
Top-k	3	Top periodic components.
# kernels	4	Number of kernels.
Dropout	0.1	Network dropout.
Channel mixing	False	Disabled in the supplied configuration.
Channel independence	False	Disabled in the supplied configuration.
Downsampling layers	1	Uses model-side scales $50 \rightarrow 25$; the 50 positions are C-MAPSS cycles or PHM12 intra-acquisition patch positions.
Downsampling window/method	2 / average	Multiscale downsampling configuration.
Normalisation	False	use_norm = False.
Embedding/frequency	fixed / h	PyPOTS temporal embedding configuration.

Table D10. The key parameters of PatchTST used in the experiments

Parameter	Value	Description
Implementation	THUML Time-Series-Library	PatchTST model implementation.
Output dimension	1	One-output head used for scalar RUL regression.
Patch length	16	Temporal patch size.
Patch stride	8	Stride between adjacent patches.
d_model	128	Model hidden dimension.
d_ff	256	Feed-forward dimension.
# encoder layers	3	Transformer encoder depth.
# heads	4	Attention heads.
Dropout	0.1	Network dropout.
Activation	GELU	Nonlinear activation.
Factor	3	Library attention/configuration factor.

Table D11. The key parameters of iTransformer used in the experiments

Parameter	Value	Description
Implementation	THUML Time-Series-Library	iTransformer model implementation.
Output dimension	1	One-output head used for scalar RUL regression.
d_model	128	Model hidden dimension.
d_ff	256	Feed-forward dimension.
# encoder layers	3	Transformer encoder depth.
# heads	4	Attention heads.
Dropout	0.1	Network dropout.
Activation	GELU	Nonlinear activation.
Factor	3	Library attention/configuration factor.

The UMAP3 method is a representation control using exact HAML descriptor/UMAP stages rather than external SOTA model families.

Table D12. The key parameters of UMAP3 used in the experiments

Parameter	Value	Description
Forest implementation	HAML Stratified Balanced Forest Regressor	Locked HAML forest implementation copied into the benchmark package.
#Estimators	300	Number of regression trees.
Minimum samples per leaf	4	Minimum leaf size.
Max features	0.75	Fraction of available features considered at each split.
Landmark budget	1500	Maximum number of landmarks used to fit UMAP.
Landmark sampling	asset-balanced; asset-condition-balanced on FD002/FD004	Uses the locked HAML round-robin landmark sampler.
UMAP dimensions	3	Latent representation dimension.
UMAP n_neighbours	14	Requested neighbourhood size.
UMAP min_dist	0.10	Minimum embedding distance.
UMAP metric	Euclidean	Latent-space metric.
Downstream predictor	Balanced RF	Uses the common forest settings above.

Table D13. Fixed parameters of the four downstream decoders

Decoder	Parameter	Value	Description
BRF	Type	StratifiedBalanced ForestRegressor	Asset–RUL stratified regression forest used by HAML.
	n_estimators	300	Number of trees.
	samples_per_asset	64	Bootstrap samples drawn per asset for each tree.
	min_samples_leaf	4	Minimum samples per leaf.
	max_features	0.75	Fraction of input features considered at each split.
	aggregation	mean	Mean aggregation across tree predictions.
	random_state	Run seed	Fixed experimental seed.
	Input standardisation	None added	Tree decoder uses the supplied representation directly.
Ridge	Type	sklearn.linear_model Ridge	Weighted linear ridge regression.
	alpha	1.0	L2 regularization strength.
	fit_intercept	True	Intercept is fitted.
	Input standardisation	TRAIN-only StandardScaler	Scaler is fitted on the selected representation using TRAIN only.
SVR	Type	sklearn.svm.SVR	Exact radial-basis-function support-vector regression.
	kernel	rbf	Radial-basis-function kernel.
	C	100.0	Penalty parameter.
	epsilon	0.1	Epsilon-insensitive tube width.
	gamma	scale	scikit-learn scale rule.
	cache_size	4096 MB	Kernel cache size.
	Input standardisation	TRAIN-only StandardScaler	Scaler is fitted on the selected representation using TRAIN only.
XGBoost	Type	XGBRegressor	Gradient-boosted tree regression.
	n_estimators	300	Number of boosting trees.

Decoder	Parameter	Value	Description
	max_depth	6	Maximum tree depth.
	learning_rate	0.05	Boosting learning rate.
	subsample	0.8	Training-row subsampling fraction.
	colsample_bytree	0.8	Feature subsampling fraction per tree.
	min_child_weight	1.0	Minimum child weight.
	reg_lambda	1.0	L2 regularization.
	objective	reg:squarederror	Squared-error regression objective.
	tree_method	hist	Histogram-based tree construction.
	n_jobs	-1	Use available CPU threads.
	random_state	Run seed	Fixed experimental seed.
	Input standardisation	None added	Tree decoder receives the representation directly.

D2. METRICS

We evaluate model performance using a combination of accuracy-and safety-oriented metrics. There are two accuracy-oriented metrics: The **Root Mean Squared Error (RMSE),**

$$\text{RMSE} = \sqrt{\frac{1}{N_{eval}} \sum_{n=1}^{N_{eval}} (\widehat{y}_n - y_n)^2} \tag{D1}$$

where y_n is the ground-truth RUL, $\widehat{y}_n$ is the predicted RUL, and N_{eval} is the total number of evaluation samples RMSE measures the overall prediction accuracy and penalizes large errors more heavily; lower values are better. Notably, RMSE treats early and late predictions equally; however, in prognostics, late predictions of RUL can be more harmful (Wang et al., 2025b). To address this, the **S-score v1** is proposed,

$$S_{v1} = \sum_{n=1}^{N_{eval}} score_n, \text{ where } score_n = \begin{cases} \exp\left(-\frac{\widehat{y}_n - y_n}{13}\right) - 1, & \text{if } \widehat{y}_n - y_n < 0 \\ \exp\left(\frac{\widehat{y}_n - y_n}{10}\right) - 1, & \text{if } \widehat{y}_n - y_n \geq 0 \end{cases} \tag{D2}$$

S_{v1} has primarily been used in aeroengine tasks, while another version S_{v2} has mainly been applied to bearing tasks (Wang et al., 2025b).

$$S_{v2} = \sum_{n=1}^{N_{eval}} score_n, \text{ where } score_n = \begin{cases} \exp\left(-\ln(0.5)\left(\frac{y_n - \widehat{y}_n}{5y_n}\right) \times 100\right) & \text{if } \widehat{y}_n - y_n \geq 0 \\ \exp\left(\ln(0.5)\left(\frac{y_n - \widehat{y}_n}{20y_n}\right) \times 100\right) & \text{if } \widehat{y}_n - y_n < 0 \end{cases} \tag{D3}$$

S-score encodes the asymmetric operational risk inherent in industrial prognostics. Late predictions ($\widehat{y}_n > y_n$) overestimate remaining life and risk unscheduled catastrophic failure, thus triggering a severe exponential penalty (e.g. scaling with divisor 10). Early predictions ($\widehat{y}_n < y_n$) merely cause premature maintenance, triggering a softer exponential penalty (e.g. scaling with divisor 13).

For PHM12, S-score v2 is evaluated from the relative prediction error on the normalised RUL scale, following (Wang et al., 2025b). RMSE and step-based late-error statistics are reported in absolute RUL units after applying the bearing-specific benchmark scaling described above.

To explicitly assess safety, we report the P95 Late Error, defined as the 95th percentile of positive errors,

$$P95\ Late\ Error = \mathrm{Q}_{0.95}\left(\{\widehat{y}_n - y_n \mid \widehat{y}_n > y_n\}\right) \tag{D4}$$

P95 Late Error characterises a high upper-tail quantile of dangerous RUL overestimation while being less sensitive than the maximum error to isolated extremes. Smaller values therefore indicate better control of severe but non-maximal late-prediction errors. If a trajectory contains no positive errors, P95 Late Error is defined as zero.

We used five Health Index (HI) diagnostics to characterise complementary properties of the resulting degradation trajectories. Monotonicity and mean HI step quantify trajectory behaviour that is directly influenced by the non-decreasing and bounded-step constraints of the HAML filter, whereas HI-progress correlation, prognosability, and healthy-failure separation provide empirical measures of lifecycle alignment and endpoint consistency that are not guaranteed by those constraints.

Monotonicity Progress. This metric measures the proportion of HI evolution consistent with a non-decreasing degradation trajectory. Higher values indicate fewer apparent health reversals. For the full HAML filter, monotonicity is expected to equal 1 by construction because the final state update projects onto a non-decreasing feasible interval (Theorem 5); we therefore use this metric primarily to verify constraint satisfaction and to contrast HAML with ablations in which the monotonic constraint is removed.

$$\text{Monotonicity} = \frac{1}{N_{eng}} \sum_{i=1}^{N_{eng}} \left| \frac{\sum_{t=t_i^{(0)}}^{T_i-1} \text{sgn}\left(h_{i,t+1} - h_{i,t}\right)}{T_i - t_i^{(0)}} \right| \tag{D5}$$

where N_{eng} is the total number of assets, $h_{i,t}$ is the HI of asset i at step t, and T_i is its total lifespan. For the full HAML filter, monotonicity is enforced by construction and is therefore interpreted primarily as a constraint-satisfaction diagnostic.

HI Progress Correlation. This metric measures the temporal alignment between the learned HI sequence $h_{i,t}$ and the true normalised degradation progress $p_{i,t} = \frac{t - t_i^{(0)}}{T_i - t_i^{(0)}} \in [0, 1]$. In this study, high progress correlation indicates that the HI is well aligned with independently defined normalised lifecycle progress. Unlike monotonicity, this alignment is not guaranteed by the causal filter: a non-decreasing trajectory can still exhibit weak correspondence with the true ordering or rate of degradation. Higher values are therefore interpreted as stronger empirical lifecycle alignment. Higher values are better (ideal target is 1.0).

$$\text{Progress Correlation} = \frac{1}{N_{eng}} \sum_{i=1}^{N_{eng}} \frac{\sum_{t=t_i^{(0)}}^{T_i} \left(h_{i,t} - \bar{h}_i\right)\left(p_{i,t} - \bar{p}_i\right)}{\sqrt{\sum_{t=t_i^{(0)}}^{T_i} \left(h_{i,t} - \bar{h}_i\right)^2} \sqrt{\sum_{t=t_i^{(0)}}^{T_i} \left(p_{i,t} - \bar{p}_i\right)^2}} \tag{D6}$$

where N_{eng} is the total number of assets, T_i is the operational lifespan of asset i, $h_{i,t}$ is the HI value at step t, and $\bar{h}_i, \bar{p}_i$ denote their respective temporal means.

Prognosability. This metric measures the variability of HI values across different assets precisely at their end-of-life point ($t = T_i$). High prognosability indicates that all units fail at a consistent failure threshold (e.g., $h \approx 1.0$), establishing reliable failure criteria. Higher values are better (closer to 1.0).

$$\text{Prognosability} = \exp\left(-\frac{\text{std}_i\left(h_{i,T_i}\right)}{\text{mean}_i\left(\left|h_{i,T_i} - h_{i,t_i^{(0)}}\right|\right)}\right) \tag{D7}$$

Mean HI Step. This metric calculates the average absolute single-step HI increment and summarises step-to-step trajectory variation. In HAML, the maximum possible one-step increase is

explicitly bounded by $\Delta_{\max}$; therefore, the mean HI step is partly controlled by construction and should not be interpreted as an independent measure of representation quality. Its empirical value nevertheless reflects how frequently and by how much the filtered state moves within that admissible range. Lower values indicate smoother evolution, but a trivially constant HI would also attain a small value; we therefore interpret this metric jointly with HI-progress correlation and healthy–failure separation.

$$\text{Mean HI Step} = \frac{1}{N_{eng}} \sum_{i=1}^{N_{eng}} \frac{1}{T_i - t_i^{(0)}} \sum_{t=t_i^{(0)}}^{T_i-1} \left| h_{i,t+1} - h_{i,t} \right| \tag{D8}$$

Healthy-Failure Separation. This metric measures the gap between the mean HI in the initial healthy region ($p \le \tau_H$) and the final failure region ($p \ge \tau_F$). A wider separation creates a distinct, easily discriminable boundary between operational life stages. Higher values are better.

$$\text{Separation} = \frac{1}{N_{eng}} \sum_{i=1}^{N_{eng}} \left[\frac{1}{\left|\mathcal{T}_i^{\mathcal{F}}\right|} \sum_{t \in \mathcal{T}_i^{\mathcal{F}}} h_{i,t} - \frac{1}{\left|\mathcal{T}_i^{\mathcal{H}}\right|} \sum_{t \in \mathcal{T}_i^{\mathcal{H}}} h_{i,t} \right] \tag{D9}$$

where $\mathcal{T}_i^{\mathcal{H}} = \{t \mid p_{i,t} \le \tau_H\}$ and $\mathcal{T}_i^{\mathcal{F}} = \{t \mid p_{i,t} \ge \tau_F\}$ represent the sets of healthy and failure steps, respectively.

Interpretation of HI metrics. The five reported HI metrics do not play equivalent evidential roles. Monotonicity is directly enforced by the causal projection, and the maximum local step is bounded by $\Delta_{\max}$; these metrics therefore primarily verify that the implemented trajectory obeys its intended constraints. By contrast, HI-progress correlation, prognosability, and healthy–failure separation are not guaranteed by the filter and are used to assess whether the constrained trajectory remains meaningfully aligned with degradation progression and endpoint structure.

E. Algorithm and Parameter Settings

The HAML pseudocode is presented in Algorithm 1.

Algorithm 1 Health-Aware Manifold Learning (HAML)

Training:

1. $\mathbf{X} \leftarrow$ causal sensor windows; $\mathbf{q} \leftarrow \phi(\mathbf{X})$.
2. $\mathcal{L} \leftarrow$ selected descriptor-space landmark set.
3. Fit a representation backend $f_{\mathcal{M}}$ on $\mathcal{L}$ and obtain latent states $\mathbf{z}$.
4. Fit a health graph $\mathcal{G}_H$ using mutual k_G-NN edges, progress-aware costs (2), temporal edges (3), and multi-source healthy/failure references with corresponding distances in (5) and (6).
5. Compute
$$s = \frac{d_H}{d_H + d_F}$$
(7), and estimate the training-only noise threshold τ_n by (11).
6. Compute causal monotonic HI by (12).
7. Fit g on training data.

Inference at step t:

8. $\mathbf{q}_t \leftarrow \phi(\mathbf{X}_t)$; $\mathbf{z}_t \leftarrow f_{\mathcal{M}}(\mathbf{q}_t)$.
9. $$s_t \leftarrow \frac{d_H(\mathbf{z}_t)}{d_H(\mathbf{z}_t) + d_F(\mathbf{z}_t)}.$$
10. $$h_t, v_t \leftarrow \mathcal{F}_{\text{HI}}\left(s_t \mid h_{<t}, v_{t-1}\right).$$
11. Compute $\hat{y}_t$ using (13).

The selected HAML hyperparameters used across all primary experiments, including the benchmark comparisons, noise sensitivity analysis (Section 4), and ablation study (Section 5), are reported in Table E1. The primary configuration was fixed before test evaluation; sensitivity analysis is used post hoc to assess robustness to reasonable perturbations of these settings rather than to select dataset-specific test-optimal values.

Table E1. Selected values of hyperparameters for HAML in experiments

Parameter	Final value	Description
W	50	Causal lifecycle-window length in cycles for C-MAPSS
PHM12 input shape	50x24	50 intra-acquisition patches $\times$ 24 bearing-feature channels
$R_{\max}$	125	Piecewise-linear RUL cap in cycles for C-MAPSS only
R_{desc}	5	Recent segment used by the temporal descriptor: recent mean/slope.
d	3	UMAP embedding dimension.
M_L	1500	Number of asset-/condition-balanced landmarks selected without lifecycle-progress or RUL supervision.
τ_H	0.05	Healthy landmark reference threshold.
τ_F	0.97	Failure landmark reference threshold.
k_G	18	Mutual nearest-neighbour graph degree target.
Max neighbours	50	Upper bound used by graph connectivity/neighborhood construction.
K_{OOS}	9	Nearest fitted landmarks used for query out-of-sample graph distance.
λ_G	4.5	Progress-aware spatial edge penalty strength.
η_T	0.45	Scale factor for temporal edge costs.
α^+	0.35	Level correction gain for non-negative residuals.
α^-	0.05	Level correction gain for negative residuals.
β	0.02	Positive-innovation velocity reinforcement gain.
ρ	0.995	Persistent velocity carry-over.
$v_{\max}$	0.02	Maximum persistent velocity.
$\Delta_{\max}$	0.04	Maximum one-step filtered HI increment.
τ_{min}	0.002	Lower clip bound for learned noise threshold.
τ_{max}	0.03	Upper clip bound for learned noise threshold.
H_{hist}	30	Number of recent filtered HI values in the decoder.
ε_{rate}	0.0005	Minimum rate denominator for rate-based RUL estimate.
q_n	0.25	Quantile level used for positive early-life raw-coordinate differences.
UMAP n_neighbours	14	Number of neighbours controlling the local embedding scale.
UMAP min_dist	0.1	Minimum distance controlling local packing in the UMAP embedding.

We next present a sensitivity analysis of key hyperparameters and their impact on model performance. The selection of these hyperparameters is guided by a principled criterion that emphasizes components most critical to the health-aware geometric construction and the causal health index (HI). Based on this rationale, we design and evaluate 13 sensitivity cases, as summarised in Table E2. In addition to the results reported in the main paper, further discussion and extended analyses of hyperparameter sensitivity are provided in Appendix G.

Table E2. Key hyperparameters used in the sensitivity analysis

	Parameter	Default	Sweep	Reason for sensitivity analysis
S1	λ_G	4.5	$\lambda_G \in \{1.5, 2.5, 3, 3.5, 4.5, 5\}$	Directly contributes to the shortcut cross-stage penalty.
S2	k_G	18	$k_G \in \{10, 14, 18, 24, 30\}$	Influence on topology/connectivity and geodesic stability.
S3	M_L	1500	$M_L \in \{500, 1000, 1500, 2000, 3000\}$	Evaluate the predictive-quality/runtime trade-off as the landmark budget increases.
S4	τ_H, τ_F	0.05 / 0.97	$\tau_H \in \{0.03, 0.05, 0.08, 0.1\}$ $\tau_F \in \{0.9, 0.95, 0.97, 0.99\}$	Reference regions to determine the health coordinate.
S5	η_T	0.45	$\eta_T \in \{0.2, 0.35, 0.45, 0.6, 0.8\}$	Evaluate the sensitivity of temporal geometry.
S6	K_{OOS}	9	$K_{OOS} \in \{3, 5, 9, 15, 25\}$	Influence on out-of-sample distance interpolation and inference stability.
S7	ρ	0.995	$\rho \in \{0.96, 0.98, 0.995\}$	Main parameter of persistent degradation memory.
S8	α^+, α^-	0.35 / 0.05	$\alpha^+ \in \{0.2, 0.35, 0.5\}$; $\alpha^- \in \{0, 0.05, 0.10\}$	Evaluate the claim of asymmetric response/suppression of self-healing.
S9	β	0.02	$\beta \in \{0.005, 0.01, 0.02, 0.04, 0.08\}$	Control velocity that learns from positive innovation.
S10	$\Delta_{\max}$, $v_{\max}$	0.04 / 0.02	$\Delta_{\max} \in \{0.02, 0.04, 0.08\}$; $v_{\max} \in \{0.01, 0.02, 0.04\}$	Hard bounds on degradation-state velocity and one-step HI increment; may influence the late-error tail.
S11	q_n	0.25	$\{0.10, 0.25, 0.50, 0.75\}$	Evaluate the robustness of the learned early-life noise floor.
S12	d	3	$\{3, 4, 5, 6\}$	Relate to geometric capacity.
S13	UMAP n_neighbours / min_dist	14 / 0.10	n_neighbours $\in \{8, 14, 24, 40\}$; min_dist $\in \{0, 0.1, 0.3, 0.5\}$	Evaluate manifold backend sensitivity since backend geometry can change graph neighborhood structure.

F. Computational Complexity and Empirical Efficiency

Theoretical complexity of HAML: Let N_{tr} denote the number of causal training windows. Because HAML extracts a fixed set of seven descriptors from every channel, descriptor construction is linear in the number of samples, temporal context, and channels $T_{desc} = O(N_{tr}\, W\, F)$. The representation backend $f_{\mathcal{M}}$ and $\mathcal{G}_H$ are fitted only on M_L landmarks. After neighbourhood search $O\left(M_L^2 D_q\right),\ D_q = 7F$, the mutual-$k_G$-NN spatial graph contains $O(M_L k_G)$ edges, while temporal and connectivity-bridge edges add only sparse connections. The exact cost of fitting $f_{\mathcal{M}}$ and performing neighbourhood search is representation backend-dependent. In the present implemen-

tation, the dominant HAML-specific geometric operation is the all-pairs shortest-path computation over M_L landmark graph used to obtain the graph metric d_G in Eq. (4). Running sparse Dijkstra from each landmark requires: $T_{geo} = O\left(M_L\left(e_H + M_L\, log\, M_L\right)\right)$ and $S_{geo} = O\left(M_L^2\right)$ where S_{geo} is the storage required for the landmark-to-landmark geodesic matrix and $e_H = |E_s \cup E_t \cup E_b|$. Importantly, this graph-geodesic term is controlled by M_L rather than directly on the total number of training windows N_{tr}. With M_L fixed, increasing N_{tr} still increases descriptor construction, representation transformation, landmark-selection/neighbourhood processing, and downstream decoder fitting, while the size of the precomputed landmark-to-landmark geodesic matrix remains fixed. Finally, for out-of-sample transform and HI filter, the time complexities are $T_{oos} = O\left(N_{tr} M_L D_q\right)$ and $T_{HI_filter} = O\left(N_{tr}\right)$, respectively.

Empirical runtime comparison. All methods were evaluated on the same hardware platform (Intel Core i5-10300H CPU, 8 GB RAM, and NVIDIA GeForce RTX 1650 4 GB GPU). Wall-clock measurements depend on implementation details, software libraries, device utilisation, parallelism, and hardware characteristics; the results below should therefore be interpreted as empirical comparisons under this common evaluation environment rather than architecture-independent computational speed guarantees.

Table F1 reports the total runtime and training time across C-MAPSS and PHM12. Under this evaluation environment, HAML exhibits lower measured training times than several of the more computationally intensive neural baselines. On C-MAPSS, the measured HAML training time is 81.86–224.21 s, compared with 968.34–6556.12 s for FAT, corresponding to 11.2–31.8× lower observed training time under the tested environment. Larger wall-clock differences are observed relative to several modern time-series baselines: under the same evaluation environment, HAML's measured training time is approximately 21.2-35.8× lower than TimesNet, 40.5-57.7× lower than TimeMixer++, and 5.8-7.4× lower than PatchTST. HAML also shows lower measured training time than FC-STGNN on all four C-MAPSS subsets, with observed ratios of approximately 1.4-10.2×. A similar pattern is observed on PHM12: HAML requires 15.34-29.49 s for training across the three operating conditions, corresponding under the tested environment to approximately 7.0-10.6×, 17.0-45.1×, and 24.9-46.6× lower measured training times than FAT, TimesNet, and TimeMixer++, respectively. These measurements show that, in the evaluated implementations, HAML incurs substantially lower training-time overhead than several heavier deep temporal baselines despite introducing additional representation and graph-processing stages. The comparison should be interpreted jointly with the theoretical complexity analysis above rather than as an architecture-independent ranking of computational efficiency.

HAML is, however, not the fastest method in the wall-clock comparison. Lightweight recurrent models and STGNN generally show lower measured runtimes, particularly on C-MAPSS. For example, HAML's measured total runtime is 107.20–311.80 s across FD001–FD004, compared with 35.01–103.30 s for STGNN, corresponding to approximately 3.0–3.4× higher wall-clock time for HAML under the tested environment. Similar behaviour is observed on PHM12-C1 and C2, where HAML requires 71.58 and 36.86s compared with 25.39 and 12.20s for STGNN; on the smaller PHM12-C3 condition, however, HAML is faster overall (17.07 versus 20.54s). HAML additionally performs operations absent from direct sequence-to-RUL baselines, including representation transformation, out-of-sample landmark attachment, graph-geodesic health-coordinate evaluation, and causal level-velocity filtering; these stages contribute additional computational work, although their precise contribution to wall-clock runtime depends on implementation and hardware utilisation. Thus, HAML trades some computational simplicity relative to lightweight recurrent or sparse graph models for an explicit degradation-aware health representation with causal non-decreasing trajectory constraints.

Under the tested hardware and benchmark workloads, HAML's measured test-time latency remains below one second per evaluated engine. On C-MAPSS, HAML's total runtime exceeds its measured training time by approximately 25–88 s depending on dataset size, while direct measurements on the stated hardware give an inference latency of 253.60–352.97 ms per engine. The stored model checkpoints range from 29.80 to 50.27 MB . Therefore, although HAML is not the lowest-latency method, its measured inference latency remains below one second per engine on the stated hardware, while its observed training time is lower than that of several Transformer and modern time-series baselines in the common evaluation environment. Overall, the theoretical and empirical results indicate a trade-off between the additional geometric processing introduced by HAML and its observed

runtime profile: the method incurs greater wall-clock cost than some lightweight baselines, while requiring substantially less measured training time than several heavier deep architectures under the tested environment. The absolute timings and cross-method ratios should not be interpreted as hardware-or implementation-independent properties.

Table F1. Empirical wall-clock runtime and training-time comparison of HAML and baseline implementations on C-MAPSS and PHM12 under the common hardware environment. Times are reported as mean ± sample standard deviation over ten fixed seeds; absolute values and cross-model ratios are implementation- and hardware-dependent. **Red bold**: best, ***blue bold italic***: second best.

Dataset	Model	Runtime	Train time	Dataset	Runtime	Train time
FD001	HAML	107.20 ± 7.85	81.86 ± 7.95	PHM12-C1	71.58 ± 15.73	29.49 ± 16.51
	FAT	971.09 ± 376.9	968.34 ± 350.12		227.55 ± 3.70	225.31 ± 3.59
	LOGO	81.08 ± 1.40	79.24 ± 1.51		50.04 ± 6.47	49.31 ± 6.27
	LSTM	62.00 ± 9.67	62.00 ± 9.67		***18.28 ± 10.52***	***17.92 ± 10.33***
	GRU	***58.75 ± 0.47***	***58.75 ± 0.47***		**16.65 ± 0.62**	**16.34 ± 0.61**
	FC-STGNN	114.93 ± 1.10	114.09 ± 1.13		64.11 ± 5.50	63.47 ± 5.42
	STGNN	**35.01 ± 0.19**	**34.85 ± 0.18**		25.39 ± 0.15	24.85 ± 0.15
	TimesNet	2933.7 ± 610.06	2931.49 ± 602.06		514.64 ± 126.29	502.75 ± 118.12
	TimeMixer++	3322.3 ± 292.63	3319.04 ± 287.63		748.33 ± 480.45	734.32 ± 480.58
	PatchTST	547.95 ± 80.11	545.87 ± 78.51		277.26 ± 107.97	274.79 ± 107.94
	iTransformer	144.36 ± 24.35	144.36 ± 24.35		78.45 ± 47.61	77.95 ± 47.39
FD002	HAML	272.00 ± 2.70	206.14 ± 1.10	PHM12-C2	36.86 ± 0.44	15.34 ± 0.24
	FAT	6560.28 ± 984.27	6556.12 ± 979.32		108.19 ± 0.50	107.02 ± 0.42
	LOGO	248.68 ± 1.85	246.75 ± 1.79		175.10 ± 0.87	171.31 ± 0.93
	LSTM	159.97 ± 2.11	159.97 ± 2.11		**7.73 ± 0.35**	**7.55 ± 0.35**
	GRU	***155.23 ± 1.10***	***155.23 ± 1.10***		***8.26 ± 0.22***	***8.08 ± 0.22***
	FC-STGNN	720.46 ± 11.39	720.07 ± 11.62		102.93 ± 1.09	102.10 ± 1.03
	STGNN	**90.65 ± 0.41**	**90.44 ± 0.40**		12.20 ± 0.21	11.91 ± 0.22
	TimesNet	4381.44 ± 220.08	4377.81 ± 219.89		367.48 ± 151.70	364.31 ± 151.70
	TimeMixer++	10148.58 ± 3537.56	10147.66 ± 3535.95		477.69 ± 191.74	470.70 ± 191.70
	PatchTST	1205.74 ± 152.49	1203.52 ± 152.44		126.57 ± 22.18	125.35 ± 22.17
	iTransformer	279.59 ± 23.22	279.53 ± 23.22		48.32 ± 2.64	48.11 ± 2.64
FD003	HAML	126.30 ± 8.00	91.80 ± 5.33	PHM12-C3	17.07 ± 3.04	15.45 ± 2.83
	FAT	1035.49 ± 187.01	1032.13 ± 186.65		164.43 ± 15.89	164.27 ± 15.87
	LOGO	398.45 ± 2.60	396.55 ± 2.69		30.25 ± 0.43	30.18 ± 0.50
	LSTM	***75.87 ± 0.29***	***75.87 ± 0.29***		**9.10 ± 0.13**	**9.07 ± 0.13**
	GRU	76.45 ± 6.04	76.45 ± 6.04		***9.71 ± 0.12***	***9.68 ± 0.12***
	FC-STGNN	940.54 ± 172.54	937.52 ± 173.62		36.94 ± 0.55	36.89 ± 0.55
	STGNN	**36.89 ± 1.16**	**36.55 ± 0.77**		20.54 ± 0.58	20.48 ± 0.57
	TimesNet	3159.66 ± 732.56	3157.65 ± 732.56		696.77 ± 124.03	696.51 ± 124.02
	TimeMixer++	4173.35 ± 871.33	4173.33 ± 869.51		721.01 ± 202.22	720.47 ± 202.22
	PatchTST	676.61 ± 123.79	675.55 ± 123.77		238.87 ± 105.67	238.77 ± 105.67
	iTransformer	155.62 ± 32.51	155.60 ± 32.51		53.93 ± 8.28	53.92 ± 8.28
FD004	HAML	311.80 ± 6.42	224.21 ± 0.91			
	FAT	3977.11 ± 908.10	3975.53 ± 902.89			
	LOGO	249.57 ± 3.34	247.60 ± 3.27			
	LSTM	190.23 ± 0.42	190.23 ± 0.42			
	GRU	***185.78 ± 0.96***	***185.78 ± 0.96***			
	FC-STGNN	456.48 ± 0.67	456.08 ± 0.35			
	STGNN	**103.30 ± 0.41**	**103.07 ± 0.38**			
	TimesNet	5981.97 ± 1582.49	5980.96 ± 1580.40			
	TimeMixer++	12935.66 ± 2390.63	12933.93 ± 2388.23			
	PatchTST	1489.37 ± 227.08	1488.36 ± 227.08			
	iTransformer	320.84 ± 11.69	320.84 ± 11.69			

G. Additional Results and Analysis

G1. Effect of Representation and Decoder Choice

To assess whether HAML depends on the downstream regressor, we evaluate Balanced Regression Forest (BRF), Ridge Regression, SVR, and XGBoost using three representations: the original

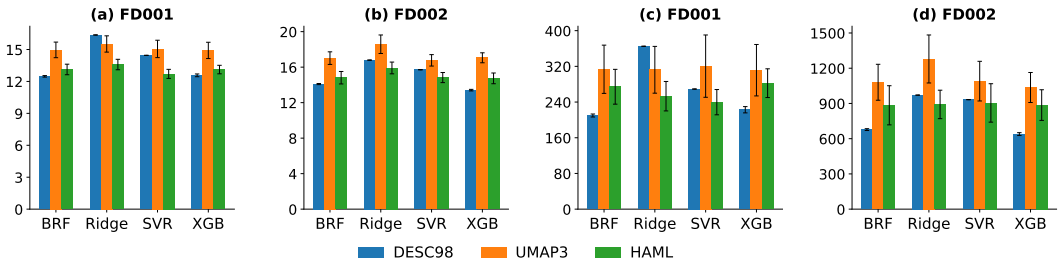

Figure G1. Effect of representation and downstream RUL decoder on FD001 and FD002. RMSE (a–b) and S-score (c–d) for four decoders operating on the original 98-dimensional descriptors (DESC98), the three-dimensional UMAP representation (UMAP3), and the proposed HAML prognostic representation (HAML). Bars show mean performance over ten seeds and error bars denote sample standard deviation. Lower is better

98-dimensional descriptors (DESC98), the 3-dimensional UMAP embedding (UMAP3), and the proposed HAML representation (HAML). Figure G1 shows that HAML consistently outperforms UMAP3 across all decoders and both FD001 and FD002, indicating that the gain arises from the health-aware graph and causal degradation-state construction rather than dimensionality reduction alone. Compared with DESC98, the results are decoder-dependent: DESC98 performs better with BRF and XGBoost, whereas HAML outperforms DESC98 with Ridge and SVR on both datasets. This suggests that unrestricted high-dimensional descriptors can benefit flexible tree ensembles, while HAML provides a more compact and structured representation that is easier for simpler decoders to exploit. Importantly, HAML additionally provides an explicit graph-geodesic health coordinate, causal non-decreasing degradation trajectory, and interpretable healthy-to-failure progression, trading some predictive flexibility for greater degradation-state structure.

We use BRF as the default decoder because it provides stable performance across datasets and risk-sensitive metrics, captures nonlinear interactions among HI-temporal and geometric features, and supports the feature-importance analysis used in this study. The cross-decoder results further show that HAML's behaviour is not specific to BRF.

G2. COMPARING TO BASELINES

Descriptor-matched baseline comparison. To examine whether HAML's performance is primarily attributable to the seven temporal descriptors, we further adapt all baseline architectures to the same HAML descriptor representation, using 50×98 inputs rather than their original sensor representation (Table G1). Under this descriptor-matched setting, HAML remains competitive, achieving the best RMSE and S-score on FD001, the lowest P95 Late Error on FD003 and FD004, and the second-best S-score on FD004. Performance on FD002 is weaker, where several adapted baselines benefit substantially from the shared descriptor representation. Overall, these results indicate that HAML's competitive performance cannot be explained solely by the handcrafted descriptors: even when the same descriptor information is supplied to competing architectures, the proposed health-aware geometry and causal degradation-state construction retain advantages on several accuracy-and risk-sensitive metrics. At the same time, the gains are dataset-dependent, showing that the descriptors themselves are informative and can also strengthen alternative models.

Engine-level concentration of asymmetric risk on FD003. To better understand the comparatively high S-score obtained on FD003, we decomposed the aggregate score into engine-level contributions for each of the ten seeds. Figure G2(a) reports the proportion of the total S-score contributed by the single largest contributor and by the five largest contributors. The concentration is particularly pronounced on FD003: the largest-contributing engine alone accounts for $72.1 \pm 8.7\%$ of the total score, while the five largest contributors account for $83.8 \pm 6.3\%$. In comparison, the corresponding top-one contributions are only approximately 20.1%, 8.8%, and 14.1% for FD001, FD002, and FD004, respectively. Thus, the poor aggregate FD003 S-score is not distributed broadly across the test population, but is dominated by a small number of high-penalty engines.

Table G1. Comparison of HAML and temporal descriptors-based models on C-MAPSS. HAML is reproduced from Table 1 of the main manuscript as a reference; all other rows use the standardized HAML 98 representation (14 sensors × 7 temporal descriptors). RMSE, S-score, and P95 Late Error are reported as mean ± sample standard deviation over ten fixed seeds; **Red bold**: best, ***blue bold italic***: second best.

Dataset	Model	RMSE ↓	S-score ↓	P95 Late Error ↓
FD001	**HAML**	**13.13 ± 0.50**	**274.3 ± 39.0**	***25.03 ± 3.04***
	FAT	13.81 ± 0.36	328.8 ± 52.4	29.84 ± 2.34
	LOGO	13.52 ± 0.57	299.2 ± 43.7	26.29 ± 3.20
	LSTM	13.93 ± 0.31	305.1 ± 28.7	27.24 ± 1.26
	GRU	13.93 ± 0.25	***296.7 ± 29.1***	26.68 ± 2.22
	FC-STGNN	14.22 ± 0.89	297.4 ± 38.3	**22.26 ± 2.73**
	STGNN	14.41 ± 0.88	300.1 ± 26.9	25.61 ± 3.68
	TimesNet	13.51 ± 0.64	305.5 ± 63.1	28.43 ± 3.32
	TimeMixer++	13.67 ± 0.43	310.4 ± 38.7	29.03 ± 2.53
	PatchTST	20.25 ± 5.69	1945.3 ± 3451.8	44.20 ± 7.76
	iTransformer	***13.45 ± 0.56***	318.1 ± 76.9	27.66 ± 3.63
FD002	**HAML**	14.81 ± 0.70	884.3 ± 166.6	30.69 ± 3.15
	FAT	12.88 ± 0.70	816.9 ± 202.3	32.55 ± 3.31
	LOGO	***12.82 ± 0.37***	730.1 ± 86.2	29.90 ± 2.67
	LSTM	**12.77 ± 0.31**	696.6 ± 50.2	31.14 ± 1.95
	GRU	12.84 ± 0.43	***682.1 ± 46.8***	30.13 ± 3.67
	FC-STGNN	13.39 ± 0.84	**647.9 ± 93.0**	**26.72 ± 2.84**
	STGNN	14.07 ± 0.59	815.1 ± 82.0	30.34 ± 2.94
	TimesNet	13.04 ± 0.33	686.1 ± 70.3	***28.75 ± 1.58***
	TimeMixer++	13.68 ± 0.35	827.6 ± 94.4	31.90 ± 2.27
	PatchTST	17.61 ± 0.42	1488.8 ± 87.9	37.99 ± 1.83
	iTransformer	13.51 ± 0.65	820.6 ± 91.5	31.59 ± 2.23
FD003	**HAML**	14.31 ± 0.43	1069.1 ± 231.7	**31.37 ± 1.72**
	FAT	14.29 ± 0.78	599.4 ± 157.7	38.94 ± 3.24
	LOGO	16.24 ± 8.76	3075.0 ± 8468.2	38.88 ± 13.82
	LSTM	***13.28 ± 0.93***	409.3 ± 143.3	35.92 ± 2.43
	GRU	13.28 ± 0.50	431.3 ± 99.8	37.54 ± 1.72
	FC-STGNN	**13.08 ± 0.60**	**301.7 ± 65.3**	***32.51 ± 3.01***
	STGNN	14.00 ± 1.10	***398.6 ± 89.6***	35.45 ± 2.91
	TimesNet	13.43 ± 0.71	464.0 ± 139.7	36.98 ± 2.84
	TimeMixer++	13.92 ± 1.04	548.8 ± 214.1	39.18 ± 2.95
	PatchTST	18.80 ± 2.10	972.3 ± 321.3	38.10 ± 5.18
	iTransformer	13.72 ± 0.78	456.7 ± 141.4	36.73 ± 1.92
FD004	**HAML**	15.79 ± 0.63	***1050.1 ± 141.5***	**27.34 ± 1.49**
	FAT	***13.74 ± 0.25***	1181.8 ± 110.4	31.88 ± 2.13
	LOGO	14.10 ± 0.41	**1040.0 ± 87.1**	***29.14 ± 2.03***
	LSTM	13.96 ± 0.19	1191.1 ± 95.0	32.74 ± 2.09
	GRU	**13.70 ± 0.18**	1105.9 ± 99.5	32.25 ± 1.57
	FC-STGNN	14.45 ± 0.52	1072.7 ± 172.0	29.92 ± 2.44
	STGNN	15.02 ± 0.27	1161.6 ± 96.3	30.49 ± 2.48
	TimesNet	14.16 ± 0.56	1113.3 ± 192.2	31.86 ± 4.23
	TimeMixer++	14.34 ± 0.42	1175.9 ± 194.8	31.20 ± 2.62
	PatchTST	19.14 ± 0.21	2074.5 ± 185.0	37.43 ± 2.13
	iTransformer	14.49 ± 0.50	1098.3 ± 201.6	29.47 ± 2.81

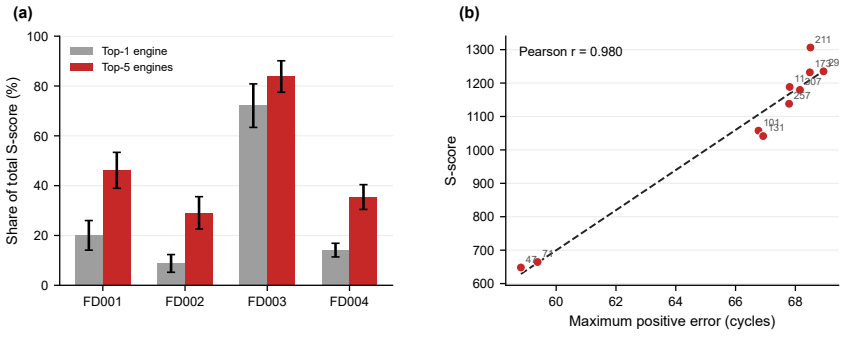

Figure G2. FD003 S-score concentration and extreme late-error association. (a) Mean share of the total S-score contributed by the largest single engine and the five largest engine-level contributors across ten seeds; error bars denote sample standard deviation. (b) Relationship between total S-score and maximum positive RUL prediction error across the ten FD003 seeds.

Table G2. Health-index diagnostics for HAML ablations on C-MAPSS. A7a–A7d are collapsed because they alter only the downstream readout and share the same upstream HI trajectory. Higher is better for monotonicity, HI-progress correlation, prognosability, and healthy-failure separation; lower is better for mean HI step. Monotonicity and bounded step behaviour are directly influenced by the causal-filter constraints and are therefore interpreted as constraint-satisfaction diagnostics, whereas progress correlation, prognosability, and healthy-failure separation provide complementary measures of HI quality. **Red bold**: numerically better than A0.

Dataset	Ablation	Monotonicity ↑	HI progress corr. ↑	Prognosability ↑	Mean HI step ↓	Healthy–failure sep. ↑
FD001	A0	1.000 ± 0.000	0.902 ± 0.006	0.997 ± 0.004	0.0061 ± 0.0000	0.912 ± 0.010
	A1	1.000 ± 0.000	0.883 ± 0.008	0.997 ± 0.003	0.0061 ± 0.0000	**0.918 ± 0.007**
	A2	1.000 ± 0.000	0.886 ± 0.008	0.997 ± 0.004	0.0061 ± 0.0000	0.910 ± 0.010
	A3	1.000 ± 0.000	0.816 ± 0.025	0.810 ± 0.056	**0.0041 ± 0.0001**	0.585 ± 0.022
	A4	1.000 ± 0.000	0.899 ± 0.007	**0.999 ± 0.002**	0.0062 ± 0.0000	**0.931 ± 0.009**
	A5	0.540 ± 0.003	0.886 ± 0.005	0.975 ± 0.006	0.0437 ± 0.0010	0.897 ± 0.008
	A6	1.000 ± 0.000	**0.906 ± 0.006**	0.988 ± 0.003	**0.0060 ± 0.0000**	0.879 ± 0.010
	A7a–A7d	1.000 ± 0.000	0.902 ± 0.006	0.997 ± 0.004	0.0061 ± 0.0000	**0.913 ± 0.010**
	A8/PCA	1.000 ± 0.000	0.886 ± 0.004	0.976 ± 0.005	0.0061 ± 0.0000	0.842 ± 0.011
	A8/Isomap	1.000 ± 0.000	0.884 ± 0.003	0.988 ± 0.007	0.0062 ± 0.0000	0.880 ± 0.011
	A8/PHATE	1.000 ± 0.000	0.840 ± 0.010	0.988 ± 0.008	0.0063 ± 0.0000	0.897 ± 0.019
	A8/Diffusion	1.000 ± 0.000	0.818 ± 0.010	0.987 ± 0.010	0.0063 ± 0.0000	**0.941 ± 0.016**
FD002	A0	1.000 ± 0.000	0.892 ± 0.008	0.997 ± 0.002	0.0060 ± 0.0000	0.897 ± 0.009
	A1	1.000 ± 0.000	0.873 ± 0.007	0.997 ± 0.003	0.0060 ± 0.0000	**0.900 ± 0.009**
	A2	1.000 ± 0.000	0.875 ± 0.009	0.996 ± 0.003	**0.0059 ± 0.0000**	0.894 ± 0.008
	A3	1.000 ± 0.000	0.851 ± 0.014	0.899 ± 0.039	**0.0045 ± 0.0001**	0.654 ± 0.025
	A4	1.000 ± 0.000	0.888 ± 0.011	**0.999 ± 0.002**	0.0060 ± 0.0000	**0.911 ± 0.008**
	A5	0.527 ± 0.002	0.873 ± 0.006	0.964 ± 0.008	0.0562 ± 0.0019	0.864 ± 0.009
	A6	1.000 ± 0.000	**0.898 ± 0.008**	0.984 ± 0.005	**0.0059 ± 0.0000**	0.854 ± 0.010
	A7a–A7d	1.000 ± 0.000	0.892 ± 0.008	0.997 ± 0.002	0.0060 ± 0.0000	**0.898 ± 0.009**
	A8/PCA	1.000 ± 0.000	0.879 ± 0.006	0.966 ± 0.009	**0.0059 ± 0.0001**	0.829 ± 0.021
	A8/Isomap	1.000 ± 0.000	0.874 ± 0.007	0.977 ± 0.010	0.0060 ± 0.0001	0.857 ± 0.020
	A8/PHATE	1.000 ± 0.000	0.848 ± 0.007	0.985 ± 0.009	0.0062 ± 0.0000	0.897 ± 0.019
	A8/Diffusion	1.000 ± 0.000	0.836 ± 0.006	0.982 ± 0.016	0.0063 ± 0.0001	**0.925 ± 0.022**
FD003	A0	1.000 ± 0.000	0.872 ± 0.010	0.998 ± 0.003	0.0048 ± 0.0000	0.889 ± 0.007
	A1	1.000 ± 0.000	0.847 ± 0.010	0.998 ± 0.003	0.0048 ± 0.0000	**0.900 ± 0.008**
	A2	1.000 ± 0.000	0.852 ± 0.010	0.998 ± 0.004	0.0048 ± 0.0000	0.889 ± 0.009
	A3	1.000 ± 0.000	0.531 ± 0.101	0.472 ± 0.035	**0.0021 ± 0.0001**	0.359 ± 0.066
	A4	1.000 ± 0.000	0.870 ± 0.009	**0.999 ± 0.001**	0.0048 ± 0.0000	**0.906 ± 0.006**
	A5	0.531 ± 0.002	0.856 ± 0.008	0.972 ± 0.006	0.0449 ± 0.0009	0.886 ± 0.008
	A6	1.000 ± 0.000	**0.873 ± 0.010**	0.988 ± 0.004	**0.0047 ± 0.0000**	0.858 ± 0.006
	A7a–A7d	1.000 ± 0.000	0.872 ± 0.010	0.998 ± 0.003	0.0048 ± 0.0000	**0.891 ± 0.007**
	A8/PCA	1.000 ± 0.000	0.834 ± 0.005	0.981 ± 0.009	0.0048 ± 0.0000	0.851 ± 0.012
	A8/Isomap	1.000 ± 0.000	0.821 ± 0.014	0.992 ± 0.006	0.0048 ± 0.0000	0.867 ± 0.009
	A8/PHATE	1.000 ± 0.000	0.796 ± 0.008	0.990 ± 0.008	0.0049 ± 0.0000	**0.901 ± 0.017**
	A8/Diffusion	1.000 ± 0.000	0.799 ± 0.017	**0.999 ± 0.002**	0.0049 ± 0.0000	**0.910 ± 0.022**
FD004	A0	1.000 ± 0.000	0.857 ± 0.012	0.982 ± 0.022	0.0044 ± 0.0004	0.800 ± 0.110
	A1	1.000 ± 0.000	0.834 ± 0.007	**0.987 ± 0.007**	0.0047 ± 0.0001	**0.864 ± 0.015**
	A2	1.000 ± 0.000	0.838 ± 0.012	0.977 ± 0.026	0.0044 ± 0.0004	**0.801 ± 0.109**
	A3	1.000 ± 0.000	0.482 ± 0.079	0.465 ± 0.071	**0.0018 ± 0.0004**	0.291 ± 0.060
	A4	1.000 ± 0.000	0.853 ± 0.005	**0.994 ± 0.004**	0.0047 ± 0.0001	**0.877 ± 0.016**
	A5	0.520 ± 0.003	0.828 ± 0.012	0.937 ± 0.030	0.0649 ± 0.0101	0.786 ± 0.096
	A6	1.000 ± 0.000	0.857 ± 0.008	0.973 ± 0.014	0.0044 ± 0.0004	0.765 ± 0.102
	A7a–A7d	1.000 ± 0.000	0.857 ± 0.012	0.982 ± 0.022	0.0045 ± 0.0004	**0.802 ± 0.109**
	A8/PCA	1.000 ± 0.000	0.840 ± 0.021	0.967 ± 0.015	0.0046 ± 0.0003	0.797 ± 0.071
	A8/Isomap	1.000 ± 0.000	0.825 ± 0.019	0.967 ± 0.017	0.0046 ± 0.0002	**0.819 ± 0.056**
	A8/PHATE	1.000 ± 0.000	0.821 ± 0.020	0.968 ± 0.024	0.0047 ± 0.0003	**0.844 ± 0.081**
	A8/Diffusion	1.000 ± 0.000	0.806 ± 0.010	**0.992 ± 0.016**	0.0049 ± 0.0002	**0.891 ± 0.079**

This behaviour is consistent with the exponential asymmetry of the S-score, under which positive errors, i.e., RUL overestimation, receive rapidly increasing penalties. Figure G2(b) further shows that across the ten FD003 runs, the total S-score is strongly correlated with the maximum positive error ($r = 0.980$). Seeds containing a larger extreme overestimation consequently incur substantially larger aggregate scores, even when the majority of engine predictions remain relatively accurate. This interpretation is also consistent with the broader error statistics: FD003 retains a large proportion of predictions close to the ground truth, while exhibiting occasional positive-error extremes that disproportionately affect the asymmetric metric. The result therefore distinguishes tail-risk concentration from a general degradation of predictive accuracy.

Importantly, this analysis does not imply that the extreme late predictions are operationally unimportant; rather, it identifies where the residual safety risk is concentrated. The result motivates the simultaneous reporting of RMSE, S-score, and P95 Late Error in our evaluation: RMSE summarises overall prediction accuracy, P95 Late Error characterises the broader upper tail of dangerous overestimation, whereas the exponential S-score can be dominated by a small number of extreme events. Taken together, these metrics provide a more complete assessment of prognostic performance than any single measure in isolation.

Table G3. Experimental results of HAML and benchmarks on PHM12 condition 1 (test bearings 3–7). RMSE, S-score v2 (higher is better), and P95 Late Error are reported as mean ± sample standard deviation over ten fixed seeds; **Red bold**: best, ***blue bold italic***: second best.

Models	C1 · Test bearing 3			C1 · Test bearing 4			C1 · Test bearing 5		
	RMSE ↓	S-score v2 ↑	P95 Late Error ↓	RMSE ↓	S-score v2 ↑	P95 Late Error ↓	RMSE ↓	S-score v2 ↑	P95 Late Error ↓
HAML	**277.80 ± 69.44**	***0.375 ± 0.115***	**454.08 ± 160.74**	303.93 ± 102.37	0.258 ± 0.110	566.77 ± 171.33	860.50 ± 186.97	0.200 ± 0.076	1521.28 ± 334.22
FAT	401.70 ± 19.74	0.323 ± 0.024	706.73 ± 94.45	***237.58 ± 10.08***	***0.314 ± 0.015***	***441.05 ± 52.03***	851.78 ± 58.66	0.227 ± 0.014	1608.10 ± 85.71
LOGO	597.26 ± 18.23	0.299 ± 0.002	578.50 ± 36.20	329.87 ± 1.26	0.234 ± 0.002	522.63 ± 17.64	**671.14 ± 5.00**	0.243 ± 0.002	**1010.33 ± 38.19**
LSTM	529.80 ± 157.02	0.349 ± 0.092	609.55 ± 334.84	304.94 ± 86.16	0.263 ± 0.063	466.06 ± 192.60	776.23 ± 202.43	0.230 ± 0.054	1382.95 ± 355.89
GRU	441.04 ± 67.13	0.359 ± 0.082	650.93 ± 235.39	268.14 ± 37.23	0.290 ± 0.030	479.39 ± 113.22	836.06 ± 180.72	0.221 ± 0.047	1537.58 ± 275.42
FC-STGNN	***318.04 ± 33.42***	**0.400 ± 0.056**	***497.59 ± 109.77***	**210.51 ± 23.40**	**0.354 ± 0.021**	**324.68 ± 58.96**	***685.88 ± 95.90***	**0.277 ± 0.038**	1377.32 ± 136.58
STGNN	367.12 ± 15.46	0.373 ± 0.031	565.43 ± 65.95	251.60 ± 7.12	0.308 ± 0.007	485.68 ± 35.04	705.34 ± 43.38	***0.266 ± 0.009***	1372.81 ± 78.19
TimesNet	472.90 ± 63.53	0.321 ± 0.042	596.13 ± 203.42	342.46 ± 38.41	0.235 ± 0.015	608.62 ± 111.48	786.19 ± 105.97	0.230 ± 0.019	1428.74 ± 230.82
TimeMixer++	580.41 ± 140.11	0.274 ± 0.076	768.46 ± 365.58	401.66 ± 84.63	0.205 ± 0.040	688.08 ± 186.31	915.11 ± 213.04	0.200 ± 0.049	1586.04 ± 398.56
PatchTST	1294.09 ± 1142.03	0.156 ± 0.118	1211.01 ± 1386.26	673.77 ± 574.07	0.139 ± 0.099	796.95 ± 730.91	1504.02 ± 1253.78	0.135 ± 0.103	1715.74 ± 1650.57
iTransformer	471.92 ± 124.02	0.326 ± 0.056	642.15 ± 189.49	278.98 ± 41.61	0.274 ± 0.038	491.75 ± 92.81	743.92 ± 93.21	0.241 ± 0.038	***1333.56 ± 294.87***

Models	C1 · Test bearing 6			C1 · Test bearing 7		
	RMSE ↓	S-score v2 ↑	P95 Late Error ↓	RMSE ↓	S-score v2 ↑	P95 Late Error ↓
HAML	**592.32 ± 138.59**	**0.288 ± 0.103**	***1121.75 ± 285.33***	**258.86 ± 60.68**	0.381 ± 0.084	453.10 ± 95.15
FAT	828.89 ± 56.57	0.231 ± 0.014	1583.17 ± 80.94	290.32 ± 37.93	***0.470 ± 0.035***	415.65 ± 86.28
LOGO	***670.22 ± 4.56***	0.241 ± 0.002	**1017.33 ± 37.94**	579.40 ± 23.61	0.341 ± 0.002	***349.29 ± 34.44***
LSTM	763.07 ± 200.35	0.230 ± 0.053	1362.91 ± 360.61	439.10 ± 155.65	0.394 ± 0.129	374.30 ± 292.27
GRU	806.55 ± 169.42	0.224 ± 0.043	1504.32 ± 273.39	346.17 ± 69.44	0.437 ± 0.104	399.60 ± 195.49
FC-STGNN	699.27 ± 82.22	***0.271 ± 0.027***	1387.43 ± 122.55	366.36 ± 55.13	0.455 ± 0.042	359.24 ± 69.05
STGNN	688.26 ± 39.13	0.267 ± 0.005	1334.72 ± 82.34	***275.00 ± 40.64***	**0.542 ± 0.022**	**346.41 ± 46.27**
TimesNet	760.23 ± 91.60	0.230 ± 0.014	1395.99 ± 226.30	453.94 ± 67.20	0.355 ± 0.051	607.56 ± 212.19
TimeMixer++	886.65 ± 195.98	0.202 ± 0.043	1552.94 ± 395.85	551.15 ± 131.13	0.293 ± 0.085	779.86 ± 358.99
PatchTST	1471.71 ± 1212.79	0.136 ± 0.098	1733.81 ± 1569.18	1203.73 ± 1104.84	0.173 ± 0.147	975.40 ± 1373.43
iTransformer	753.21 ± 91.32	0.238 ± 0.031	1351.83 ± 298.49	433.34 ± 156.50	0.398 ± 0.068	427.47 ± 169.02

Experiments on PHM data. Tables G3-G4 extend the evaluation to PHM12, providing a more challenging bearing-level assessment across three operating conditions. HAML demonstrates competitive performance across substantially different degradation trajectories, ranking first in 12 of the 33 metric-bearing comparisons and within the top two in 20 of 33 comparisons. Under Condition 1, HAML performs particularly well on test bearings 3 and 6. For bearing 3, it achieves the lowest RMSE (277.80) and P95 Late Error (454.08), representing reductions of 12.7% and 8.7%, respectively, relative to the strongest competing results, while obtaining the second-highest S-score v2 (0.375). Similarly, on bearing 6, HAML achieves the best RMSE (592.32) and S-score (0.288) and the second-best P95 Late Error. HAML also yields the lowest RMSE on bearing 7, although the graph-based methods achieve stronger S-score and late-error performance. Conversely, bearings 4 and 5 are comparatively difficult for HAML, demonstrating that its advantage varies with the degradation trajectory rather than being uniform across bearings.

The advantage becomes more pronounced under Condition 2, where HAML ranks within the top two on 11 of the 15 metric-bearing comparisons. On bearings 4 and 6, HAML simultaneously obtains the best RMSE and S-score and the second-best P95 Late Error. In particular, relative to the strongest competing method, RMSE is reduced by approximately 9.1% and 9.3% on bearings 4 and 6, while S-score increases by 21.7% and 23.4%, respectively. HAML also ranks second across all three metrics on bearing 7. Notably, the P95 Late Error is zero for bearings 3 and 5, although the corresponding RMSE values are not optimal. This suggests a trade-off between overall prediction accuracy and avoidance of late predictions on these trajectories, rather than uniformly improved accuracy. Under Condition 3, HAML again achieves the best RMSE (102.19) and second-best S-score (0.329) on test bearing 3, although its P95 Late Error is less competitive. Collectively, the PHM12 results show that HAML generalises beyond C-MAPSS and remains competitive across heterogeneous bearing degradation patterns, while also revealing trajectory-dependent cases in which different accuracy and late-error objectives can favour different models.

Table G4. Experimental results of HAML and benchmarks on PHM12 conditions 2 and 3 (C2 test bearings 3–7; C3 test bearing 3). RMSE, S-score v2 (higher is better), and P95 Late Error are reported as mean ± sample standard deviation over ten fixed seeds; **Red bold**: best, ***blue bold italic***: second best.

Models	C2 · Test bearing 3			C2 · Test bearing 4			C2 · Test bearing 5		
	RMSE ↓	S-score v2 ↑	P95 Late Error ↓	RMSE ↓	S-score v2 ↑	P95 Late Error ↓	RMSE ↓	S-score v2 ↑	P95 Late Error ↓
HAML	662.54 ± 19.25	0.266 ± 0.030	**0.00 ± 0.00**	**187.17 ± 17.46**	**0.314 ± 0.038**	***364.32 ± 29.44***	841.04 ± 24.31	0.153 ± 0.012	**0.00 ± 0.00**
FAT	490.35 ± 38.29	0.208 ± 0.047	878.12 ± 45.66	283.53 ± 15.11	0.163 ± 0.022	495.81 ± 18.15	890.84 ± 46.03	0.172 ± 0.035	1581.52 ± 51.96
LOGO	***421.02 ± 89.25***	**0.357 ± 0.053**	***489.74 ± 253.13***	***205.91 ± 33.77***	***0.258 ± 0.041***	**333.26 ± 102.79**	**653.94 ± 103.41**	**0.255 ± 0.035**	***1108.65 ± 286.85***
LSTM	**392.93 ± 61.11**	***0.303 ± 0.065***	751.32 ± 82.28	248.59 ± 18.52	0.201 ± 0.022	446.52 ± 27.00	778.18 ± 73.52	***0.236 ± 0.041***	1434.53 ± 93.68
GRU	434.68 ± 42.15	0.286 ± 0.037	818.79 ± 64.65	259.89 ± 17.55	0.194 ± 0.020	464.76 ± 23.44	832.32 ± 59.11	0.219 ± 0.027	1515.16 ± 77.14
FC-STGNN	425.32 ± 68.47	0.273 ± 0.089	718.01 ± 194.74	249.38 ± 31.83	0.188 ± 0.050	430.89 ± 78.49	***755.18 ± 91.20***	0.214 ± 0.071	1358.98 ± 212.16
STGNN	509.28 ± 64.43	0.172 ± 0.066	802.04 ± 94.92	274.97 ± 30.95	0.150 ± 0.047	484.12 ± 38.04	832.06 ± 93.53	0.135 ± 0.058	1483.80 ± 113.31
TimesNet	530.37 ± 167.94	0.210 ± 0.094	946.95 ± 210.15	302.12 ± 67.01	0.147 ± 0.074	522.46 ± 81.44	943.61 ± 197.88	0.166 ± 0.056	1651.37 ± 246.80
TimeMixer++	483.19 ± 81.86	0.240 ± 0.059	892.64 ± 123.24	280.07 ± 34.78	0.169 ± 0.051	497.48 ± 42.35	887.15 ± 100.44	0.178 ± 0.043	1590.54 ± 137.28
PatchTST	821.39 ± 342.45	0.150 ± 0.131	1158.59 ± 579.54	386.25 ± 159.54	0.106 ± 0.119	581.76 ± 233.51	1217.56 ± 453.10	0.122 ± 0.084	1862.39 ± 723.94
iTransformer	476.51 ± 64.39	0.210 ± 0.073	863.15 ± 83.13	284.98 ± 28.20	0.154 ± 0.044	492.51 ± 34.43	871.83 ± 77.08	0.165 ± 0.046	1553.37 ± 95.56

Models	C2 · Test bearing 6			C2 · Test bearing 7			C3 · Test bearing 3		
	RMSE ↓	S-score v2 ↑	P95 Late Error ↓	RMSE ↓	S-score v2 ↑	P95 Late Error ↓	RMSE ↓	S-score v2 ↑	P95 Late Error ↓
HAML	**169.55 ± 13.07**	**0.327 ± 0.037**	***330.49 ± 28.83***	***61.43 ± 5.74***	***0.239 ± 0.041***	***108.37 ± 7.31***	**102.19 ± 28.63**	***0.329 ± 0.125***	75.84 ± 82.55
FAT	245.89 ± 14.32	0.184 ± 0.022	439.63 ± 18.22	77.70 ± 4.75	0.188 ± 0.024	137.39 ± 5.35	122.13 ± 15.21	0.293 ± 0.038	52.66 ± 13.38
LOGO	***187.03 ± 28.38***	***0.265 ± 0.035***	**298.92 ± 87.37**	**57.98 ± 9.03**	**0.289 ± 0.037**	**89.01 ± 32.15**	110.79 ± 2.38	0.278 ± 0.001	124.02 ± 6.18
LSTM	211.65 ± 22.90	0.235 ± 0.032	394.31 ± 33.13	67.85 ± 5.56	0.233 ± 0.027	122.74 ± 8.26	110.30 ± 30.29	0.304 ± 0.073	56.72 ± 26.42
GRU	233.54 ± 17.86	0.210 ± 0.022	424.17 ± 23.18	70.66 ± 5.32	0.221 ± 0.024	126.50 ± 7.28	***108.54 ± 37.83***	0.305 ± 0.090	54.83 ± 23.77
FC-STGNN	213.59 ± 24.77	0.221 ± 0.042	383.57 ± 67.24	69.76 ± 8.73	0.215 ± 0.061	121.14 ± 25.98	129.10 ± 15.86	0.276 ± 0.048	50.55 ± 23.68
STGNN	215.18 ± 27.52	0.192 ± 0.055	402.94 ± 32.08	78.54 ± 9.41	0.141 ± 0.059	134.52 ± 11.11	130.57 ± 12.74	0.305 ± 0.037	51.29 ± 14.56
TimesNet	267.94 ± 67.06	0.160 ± 0.077	467.82 ± 80.15	82.88 ± 20.07	0.164 ± 0.075	141.14 ± 24.35	128.59 ± 25.84	0.286 ± 0.077	**41.09 ± 21.27**
TimeMixer++	248.55 ± 36.74	0.186 ± 0.051	450.80 ± 39.73	76.58 ± 10.39	0.187 ± 0.047	135.40 ± 13.92	147.68 ± 29.61	0.309 ± 0.066	62.97 ± 18.00
PatchTST	361.65 ± 151.88	0.103 ± 0.118	539.27 ± 219.75	111.94 ± 46.99	0.115 ± 0.127	167.10 ± 70.68	189.09 ± 85.78	0.186 ± 0.127	***46.22 ± 45.20***
iTransformer	246.65 ± 23.75	0.176 ± 0.040	438.60 ± 29.43	78.39 ± 8.53	0.175 ± 0.050	136.14 ± 9.93	109.97 ± 12.34	**0.336 ± 0.036**	71.40 ± 17.28

Table G5. Health-index diagnostics of HAML on PHM12 across the three operating conditions. Higher is better for HI-progress correlation, prognosability, and healthy–failure separation; monotonicity verifies satisfaction of the imposed non-decreasing constraint, while mean HI step summarises acquisition-to-acquisition variation under the bounded-step filter.

Dataset	Condition	Monotonicity ↑	HI progress corr. ↑	Prognosability ↑	Mean HI step ↓	Healthy–failure sep. ↑
PHM12	C1	1.000 ± 0.000	0.883 ± 0.045	0.985 ± 0.005	0.0005 ± 0.0000	0.560 ± 0.029
	C2	1.000 ± 0.000	0.763 ± 0.012	0.999 ± 0.003	0.0011 ± 0.0000	0.895 ± 0.022
	C3	1.000 ± 0.000	0.589 ± 0.048	0.996 ± 0.003	0.0008 ± 0.0001	0.538 ± 0.050

Across all three PHM12 operating conditions, HAML satisfies the imposed non-decreasing HI constraint (monotonicity = 1.000±0.000) and achieves consistently high prognosability (0.985 - 0.999). The alignment between HI and normalised degradation progress is more condition-dependent, decreasing from 0.883 on C1 to 0.763 on C2 and 0.589 on C3. Healthy-failure separation is strongest on C2 (0.895 ± 0.022), while C1 and C3 show more moderate separation (0.560 ± 0.029 and 0.538 ± 0.050, respectively). The mean HI step remains small in every condition, consistent with gradual acquisition-to-acquisition evolution under the bounded-step causal filter; because step size is explicitly constrained, this statistic is interpreted as a trajectory diagnostic rather than independent evidence of learned degradation quality. Overall, the PHM12 results confirm that the implemented filter satisfies its intended irreversible trajectory constraint, while the non-trivial empirical evidence lies in the high prognosability, healthy-failure separation, and condition-dependent alignment with normalised degradation progress. The lower progress correlation on C3 further shows that satisfying monotonicity alone does not guarantee close correspondence with lifecycle progression.

Noise robustness under increasing engine coverage. To further assess robustness to sensor corruption, we fixed the noise level at $\ell = 0.2$ and progressively increased the proportion of noisy

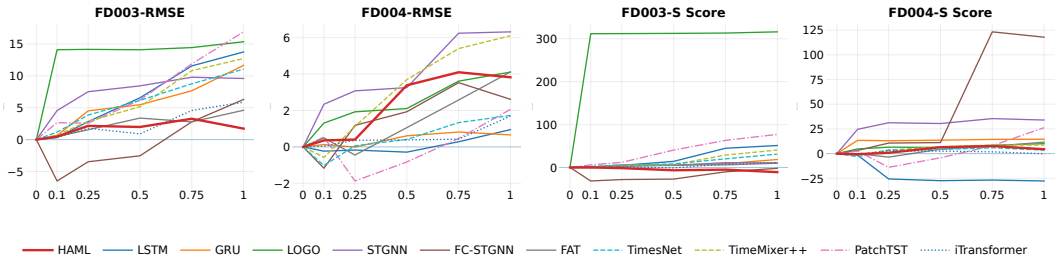

Figure G3. Robustness to increasing corrupted-engine coverage at fixed noise intensity. The Gaussian noise level is fixed at $\ell = 0.2$, while the proportion of engines corrupted over their complete observed trajectories increases from 10% to 100%. Lower values indicate greater robustness; negative values denote improvement relative to the clean reference.

test engines, $\omega_e \in \{10\%, 25\%, 50\%, 75\%, 100\%\}$. Each model is trained on clean data and frozen at evaluation. For a selected engine, zero-mean Gaussian noise with sensor-specific standard deviation $\sigma_{noise,k} = \ell\sigma_{train,k}$ is added to the raw sensor channels over its complete observed trajectory, while unselected engines remain clean. The selected engine sets are deterministic and nested, $U_{10\%} \subset U_{25\%} \subset U_{50\%} \subset U_{75\%} \subset U_{100\%}$, and the same perturbations and seeds are used across models. Performance degradation is reported relative to the clean reference as $\Delta Q = 100\,[Q\,(\ell, \omega_e)\,/Q(0) - 1]$.

Figure G3 shows that HAML remains comparatively stable as corrupted-engine coverage increases, particularly on FD003. Across 10-100% coverage, its RMSE change remains within $0.5\% - 3.3\%$; at 100% coverage, RMSE increases by only 1.7%, compared with substantially larger increases for most baselines. Its FD003 S-score is also stable and decreases by 10.9% at full coverage, indicating that the perturbation does not amplify the late-prediction penalty. On the more heterogeneous FD004, HAML exhibits a moderate but controlled response: RMSE increases by at most 4.1% and S-score by at most 7.7%, with full-coverage changes of $+3.8\%$ and $+4.4\%$, respectively. HAML is therefore not uniformly the least sensitive method on FD004, but avoids the large S-score instabilities observed for several baselines. The non-monotonic changes across coverage levels are expected because progressively increasing the nested subset introduces engines with different degradation characteristics and noise sensitivities. Negative ΔQ values should likewise not be interpreted as noise being beneficial; they indicate perturbation-specific shifts toward better-calibrated predictions for particular engines.

G3. FURTHER OBSERVATIONS ON HAML BEHAVIOUR

Table G6. Effect of causal filtering on lifecycle alignment. Pearson and Spearman correlations of normalised degradation progress with the raw graph coordinate s_t and filtered HI h_t, averaged over ten seeds.

Dataset	**Raw s_t: Pearson**	**Filtered h_t: Pearson**	**Gain**
FD001	0.886	0.902	+0.016
FD002	0.873	0.892	+0.020
FD003	0.856	0.873	+0.017
FD004	0.828	0.857	**+0.029**
Dataset	**Raw s_t: Spearman**	**Filtered h_t: Spearman**	**Gain**
FD001	0.893	0.937	+0.044
FD002	0.870	0.918	+0.048
FD003	0.842	0.903	+0.061
FD004	0.803	0.875	**+0.072**

Filtering gain under increasingly challenging degradation structure. Table G6 summarises the improvement in progress correlation introduced by causal filtering. The gain is positive for every dataset but is generally larger for the more challenging FD003-FD004 settings. In particular, the Spearman improvement increases from $+0.044$ on FD001 and $+0.048$ on FD002 to $+0.061$ on FD003 and $+0.072$ on FD004. The Pearson improvement is more modest but shows the largest gain on FD004 ($+0.029$).

These results reveal an inverse relationship between the initial quality of the raw graph coordinate and the potential benefit of temporal filtering. FD001 begins with the strongest raw lifecycle ordering (Spearman $= 0.893$) and receives the smallest improvement, whereas FD004 has the weakest raw ordering (Spearman $= 0.803$) and obtains the largest improvement. This empirical pattern suggests that the causal filter becomes particularly valuable when operating-condition variation, multiple fault modes, or their combination produces stronger local ambiguity in the latent graph. Under cleaner latent organisation, as in FD001, less correction is required; when the raw coordinate contains more local reversals or ordering ambiguity, the persistent monotonic state provides greater regularisation.

We emphasise that this result should not be interpreted as evidence that the filter eliminates all effects of heterogeneous degradation. FD004 retains lower absolute progress correlation than FD001 after filtering. Rather, the filter partially compensates for increasing geometric disorder, with its largest improvement occurring where the unfiltered coordinate is least lifecycle-consistent.

Table G7. HAML decoder feature-family importance across C-MAPSS subsets. The 49 features are grouped into five functional families: HI history, HI summaries, raw graph coordinates, velocity, and latent coordinates.

Dataset	HI history	HI dynamics / summaries	Raw graph coordinate	Velocity	Latent coordinates
FD001	45.4%	33.9%	10.5%	6.9%	3.4%
FD002	20.3%	27.8%	21.7%	12.0%	18.2%
FD003	54.7%	15.9%	15.8%	11.3%	2.2%
FD004	46.4%	17.1%	26.5%	4.1%	5.9%

Feature-family importance across seeds. Individual decoder features are highly correlated by construction: for example, the current HI, recent HI history, multi-scale changes, global HI summaries, raw graph coordinate, and latent coordinates all encode related aspects of degradation state. We therefore aggregate the 49 random-forest features into five mechanistically interpretable families and average their importance over ten seeds. The resulting distributions in Table G7 show that HI-centred temporal state information provides the largest contribution to the fitted decoder across all four datasets. HI history contributes approximately 45.4%, 20.3%, 54.7%, and 46.4% on FD001–FD004, respectively, while HI dynamics and trajectory summaries contribute a further 33.9%, 27.8%, 15.9%, and 17.1%. Persistent velocity provides an additional 6.9%, 12.0%, 11.3%, and 4.1%.

A complementary pattern emerges for geometric features. The raw graph coordinate contributes approximately 10.5%, 21.7%, 15.8%, and 26.5%, while the latent coordinates contribute 3.4%, 18.2%, 2.2%, and 5.9% on FD001–FD004, respectively. Hence, direct geometric information accounts for almost 40% of the importance on FD002 and more than 32% on FD004, considerably more than on the single-condition FD001 and FD003 subsets. This pattern is consistent with the interpretation that scalar health-coordinate compression does not preserve every aspect of system state under heterogeneous operating regimes; the decoder therefore benefits from retaining selected latent-geometric information alongside the HI-centred temporal representation.

The relatively large error bars for several feature families also indicate seed-dependent redistribution of importance among correlated predictors. We therefore interpret these values as evidence of information usage within the fitted forest, rather than as causal effects of individual feature families. The controlled A7 ablation provides stronger evidence that these feature groups contribute complementary predictive information.

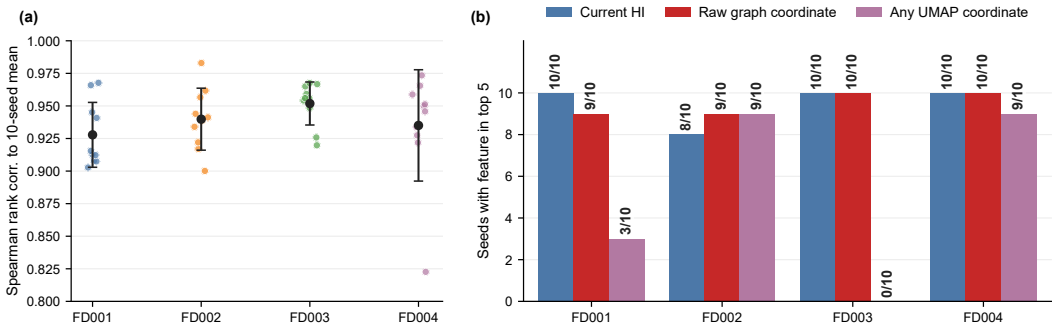

Figure G4. Stability of HAML decoder-feature importance across ten seeds. (a) Spearman rank correlation between each seed-specific 49-feature importance ranking and the corresponding ten-seed mean ranking; points denote individual seeds and black markers/error bars denote mean $\pm$ standard deviation. (b) Number of seeds in which the current filtered HI, raw graph coordinate, or at least one latent coordinate appears among the five most important decoder features.

Cross-seed stability of decoder importance. Although the exact magnitude assigned to individual features varies across fitted forests, their overall importance ordering is highly reproducible. Figure G4(a) compares each seed's 49-feature ranking with the corresponding ten-seed mean ranking. Mean Spearman rank correlations are 0.928, 0.940, 0.952, and 0.935 for FD001–FD004, respectively, demonstrating that the broad decoder information hierarchy is stable despite random variation in model fitting.

Figure G4(b) provides a complementary top-ranking analysis. The current filtered HI appears among the five most important individual features in $10/10$, $8/10$, $10/10$, and $10/10$ seeds on FD001–FD004, respectively. The raw graph coordinate is similarly persistent, appearing in the top five for $9/10$, $9/10$, $10/10$, and $10/10$seeds. These results reinforce the complementary roles of the filtered degradation state and its underlying graph-geodesic position.

The latent coordinates exhibit a more dataset-dependent pattern. At least one UMAP coordinate appears among the top five features in only $3/10$ FD001 seeds and none of the FD003 seeds, but in $9/10$ seeds for both FD002 and FD004. This agrees with the feature-family analysis: explicit latent-state information becomes more useful in the multi-condition datasets, where different operating regimes can generate state variation that is not completely represented by the scalar HI alone.

Importantly, high ranking stability coexists with noticeable variation in absolute feature importance. This is expected because many HAML variables are correlated representations of degradation state, allowing a tree ensemble to redistribute importance among substitute predictors across seeds. Consequently, the most defensible conclusion is not that a particular individual feature has a fixed universal importance, but that the same feature families and key state variables repeatedly occupy the dominant part of the decoder hierarchy.

Latent degradation organisation versus inferred HI across C-MAPSS. To examine whether the final HI reflects structure already present in the latent representation rather than forming an arbitrary monotone trajectory, Figure G5 visualises the latent representations of FD001–FD004 using two complementary colour variables. For each dataset, the representation in (a) is coloured retrospectively by normalised degradation progress, whereas the corresponding representation in (b) shows the same latent states coloured by the filtered HAML HI. Across all four subsets, the progress-coloured representations exhibit broad healthy-to-failure organisation, indicating that the representation fitted without lifecycle-progress or RUL supervision contains degradation-relevant structure. However, the degree of local stage mixing and geometric complexity varies substantially across datasets, particularly in the more heterogeneous FD002 and FD004 subsets.

The corresponding HI-coloured representations preserve the broad global organisation observed under lifecycle-progress colouring while providing a more regularised health ordering over the same latent states. Low-HI and high-HI regions generally occupy locations consistent with earlier and later lifecycle stages, although the correspondence is not uniform across all datasets. In particular, FD002 and FD004 retain greater geometric complexity than FD001, consistent with their multi-

ple operating conditions and fault-mode heterogeneity. The agreement between the two colourings is therefore qualitative rather than exact and is consistent with the dataset-dependent HI–progress correlations reported in Table G6.

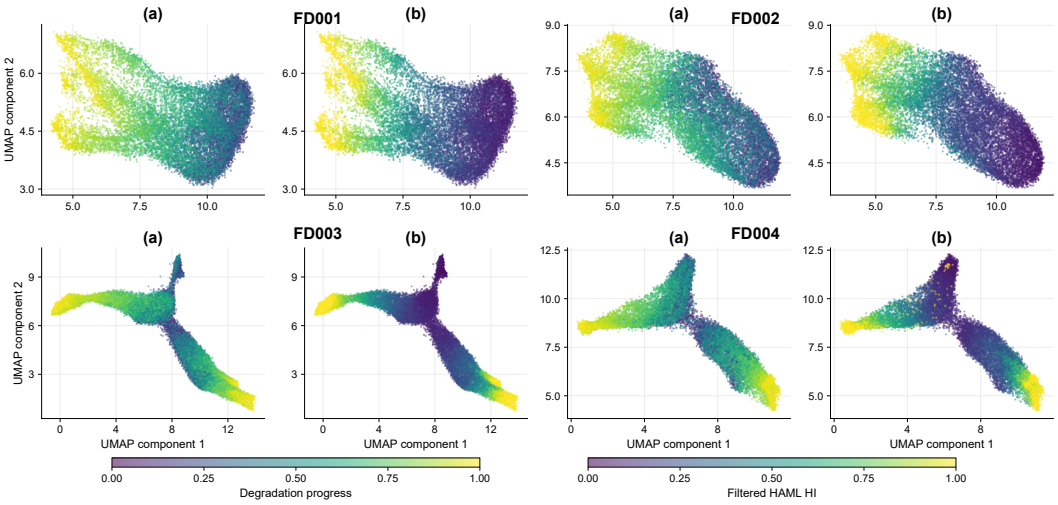

Figure G5. From latent degradation geometry to the causal health representation across C-MAPSS. For FD001–FD004, (a) shows the latent representation coloured retrospectively by normalised degradation progress, while (b) shows the same latent states coloured by the corresponding filtered HAML HI. Degradation-progress colouring is used only for retrospective evaluation and visualisation and is unavailable to the test-time inference pipeline.

Taken together, these cross-dataset visualisations illustrate the intended distinction between representation geometry and health-state construction in HAML. The representation backend preserves multidimensional latent structure in which degradation progression is present but locally ambiguous; the progress-aware graph and multi-source healthy/failure references then extract a graph-geodesic health coordinate from this structure; and the causal filter regularises its temporal evolution into the final HI. Importantly, normalised degradation progress in representation in (a) of Figure G5 is used only retrospectively for evaluation and visualisation. It is not provided to the test-time inference pipeline.

G4. SENSITIVITY ANALYSIS

We conduct a systematic one-at-a-time sensitivity analysis of the principal parameters governing HAML's latent geometry, landmark graph, graph-geodesic health coordinate, and causal level–velocity filter. The examined parameters include the progress-aware edge penalty, graph connectivity and landmark budget, healthy/failure reference thresholds, temporal and OOS graph settings, latent representation parameters, and the gains and constraint bounds of the causal HI filter. For each experiment, only the parameter under investigation is varied over the range specified in Table E2, while all remaining preprocessing, model, decoder, and evaluation settings are held fixed. Results are reported over the same ten fixed seeds using RMSE, S-score, and P95 Late Error, allowing sensitivity in both predictive accuracy and risk-sensitive performance to be assessed. It is noted that the primary HAML configuration is treated as a fixed reference, and lower test errors observed at individual sweep values are interpreted as evidence of parameter sensitivity rather than used retrospectively to retune the reported model.

The overall results show that 12 of the 17 individual parameter sweeps produce less than a 2.5% mean composite change across normalised RMSE, S-score, and P95 Late Error, indicating that HAML is generally not highly sensitive to precise hyperparameter choices. Here, composite impact denotes the mean absolute percentage deviation from the default configuration, averaged across the non-default sweep values, four C-MAPSS subsets, and three prediction metrics. The largest sensitivities are associated with the landmark budget, negative HI correction gain, two UMAP parameters:

the number of approximate nearest neighbours used to construct the initial high-dimensional graph (n_neighbours) and the minimum distance between points in low-dimensional space (min_dist), and OOS- neighbourhood size. In contrast, several parameters governing the causal HI dynamics and noise suppression show substantially flatter responses.

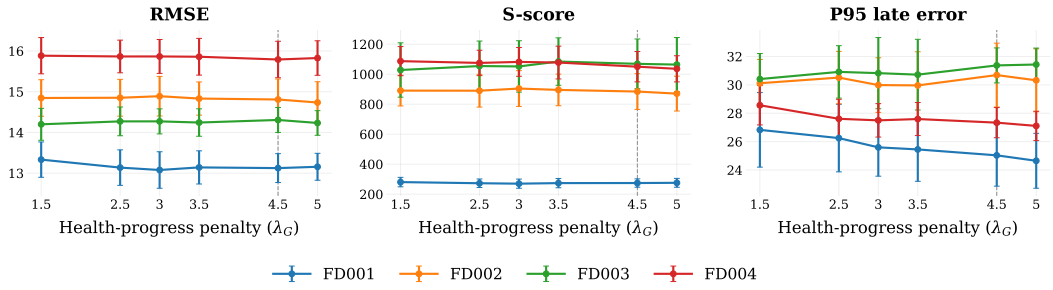

Figure G6. Sensitivity to the health-progress penalty. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

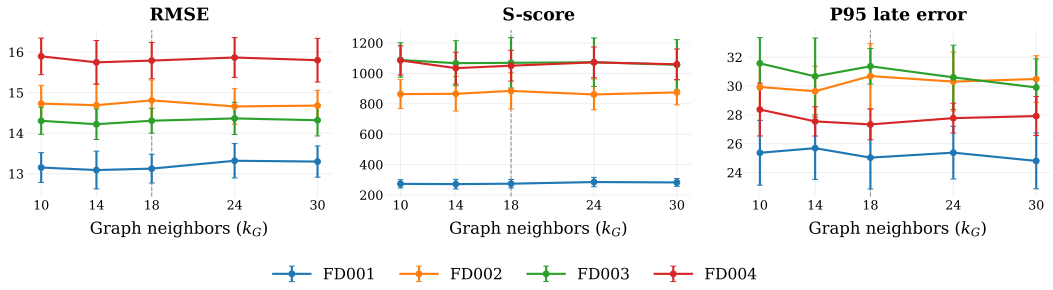

Figure G7. Sensitivity to the graph neighbours. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

S1 - Health-progress penalty λ_G. Performance is relatively stable over the broad range $\lambda_G \in [1.5, 5]$, with no common optimum across datasets (Figure G6). FD001 favours approximately 3 for RMSE/S-score, FD002 favours larger values near 5, while FD003 obtains lower RMSE and S-score at the smallest tested value. FD004 improves noticeably as the penalty increases, with $\lambda_G = 4.5 - 5$ giving the strongest overall behaviour. Importantly, the variation around the default is modest: the composite sensitivity is only about 1.40%, despite varying the penalty by more than a factor of three. This indicates that progress-aware graph weighting is not critically tuned to a single value.

S2 - Graph neighbours k_G. Varying the mutual-neighbour graph degree from 10 to 30 produces only modest changes, with a composite impact of about 1.44% (Figure G7). The best value is dataset and metric-dependent: $k_G = 14$ gives strong RMSE across FD001, FD003, and FD004, $k_G = 24$ gives the lowest FD002 RMSE and S-score, while $k_G = 30$ produces the lowest P95 Late Error on FD001 and FD003. There is therefore no systematic benefit from either very sparse or dense graphs.

S3 - Number of landmarks M_L. This is the most influential parameter in the sensitivity study, with a composite impact of about 4.22% (Figure G8). Very small landmark sets ($M_L = 500$) generally degrade RMSE and/or tail-risk performance, consistent with insufficient coverage of the latent degradation structure. Increasing the budget can improve prediction substantially; for example, on FD001, $M_L = 2000$ reduces RMSE from 13.13 to 12.48, S-score from 274.3 to 216.4, and P95 Late Error from 25.03 to 23.12. However, the effect is not monotonic across datasets: FD002 often favours 1000, while FD003 and FD004 favour different larger budgets depending on the metric. Under the same evaluation environment, increasing the landmark budget above the default $M_L = 1500$ produces a measurable increase in total wall-clock runtime, with the largest tested configuration increasing runtime by approximately 60–83%. This empirical trend is consistent with the theoretical dependence of shortest-path computation and $O\left(M_L^2\right)$ geodesic storage on the landmark budget.

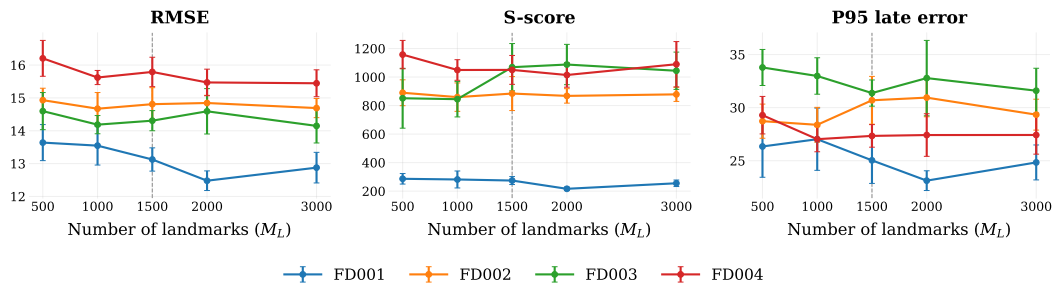

Figure G8. Sensitivity to the number of landmarks. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

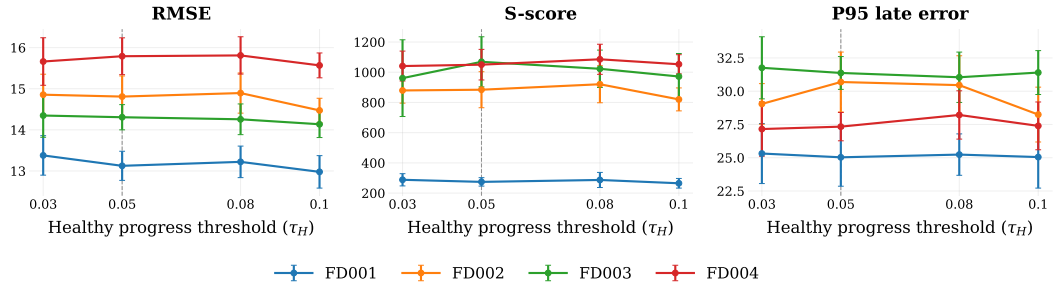

Figure G9. Sensitivity to the healthy-reference threshold. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

S4a - Healthy-reference threshold τ_H. The healthy threshold has a moderate effect (composite impact about 2.41%) and shows stronger variation than the failure threshold (Figure G9). Increasing τ_H to 0.10 improves RMSE on all four datasets, including 13.13 to 12.98 on FD001, 14.81 to 14.47 on FD002, and 15.79 to 15.57 on FD004. S-score behaviour is less uniform: $\tau_H = 0.03$ performs especially well on FD003, whereas 0.10 is strongest on FD001/FD002. This indicates a genuine trade-off in how broadly the healthy reference region is defined. A wider region supplies more healthy anchors but may also include states further along the lifecycle.

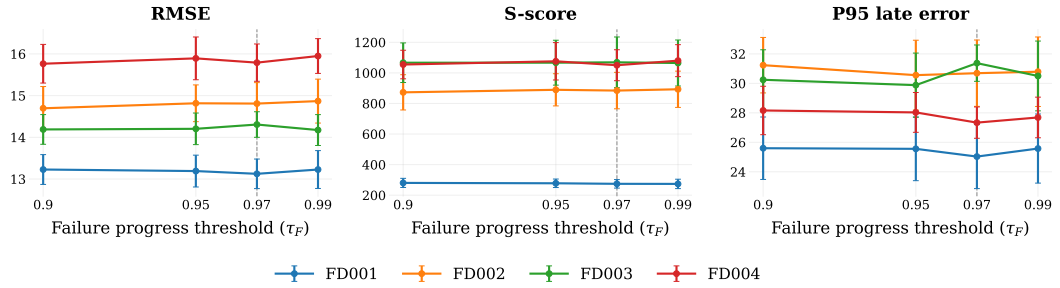

Figure G10. Sensitivity to the failure-reference threshold. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

S4b - Failure-reference threshold τ_F. HAML is considerably less sensitive to the failure threshold than to the healthy threshold (Figure G10). Across $\tau_F \in \{0.90, 0.95, 0.97, 0.99\}$, RMSE and S-score curves are nearly flat relative to their seed variability, with a composite impact of only 1.32%. Different datasets favour different thresholds; for example, FD002 RMSE favours 0.90, while FD003 RMSE favours 0.99 and there is no consistent direction of improvement. These results suggest that retaining multiple failure references makes the health coordinate relatively insensitive to the precise boundary used to define the terminal reference set. This provides a useful empirical

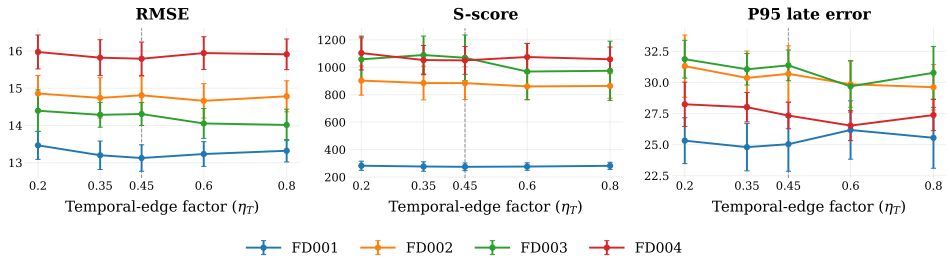

Figure G11. Sensitivity to the temporal-edge factor. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

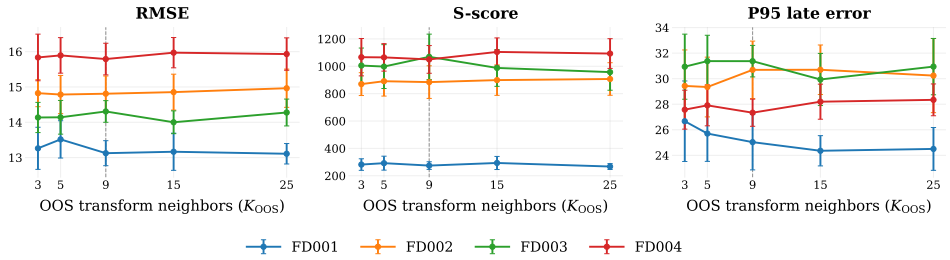

Figure G12. Sensitivity to the out-of-sample transform neighbours. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

complement to the A3 result showing the importance of multi-source, rather than single-medoid, endpoint references.

S5 - Temporal-edge factor η_T. The effect of temporal-edge weighting is non-monotonic and dataset-dependent, with a composite impact of about 1.99% (Figure G11). Increasing η_T to 0.6 substantially improves FD003 S-score (1069.1 to 968.7) and P95 Late Error (31.37 to 29.68), and also improves FD002 and FD004 P95. However, the same setting worsens FD001 P95 Late Error and FD004 RMSE relative to the default. Very small $\eta_T = 0.2$ is generally weaker. The broad stability around $0.35 - 0.8$ supports the interpretation from ablation A4: temporal edges provide a useful temporal inductive bias, but they are not the dominant determinant of HAML performance.

S6 - OOS transform neighbours K_{OOS}. OOS neighbourhood size has a somewhat larger influence than the graph degree, with a composite impact of about 2.56%, but again there is no common optimum (Figure G12). Larger neighbourhoods improve some metrics, for example, $K_{OOS} = 15$ gives FD003 RMSE 14.00 and P95 Late Error 29.94, while 25 gives FD001 S-score 267.3. Conversely, the default $K_{OOS} = 9$ gives the strongest overall FD004 results. This reflects the expected interpolation trade-off: too few neighbours may make OOS attachment locally unstable, whereas too many can blend geometrically distinct regions.

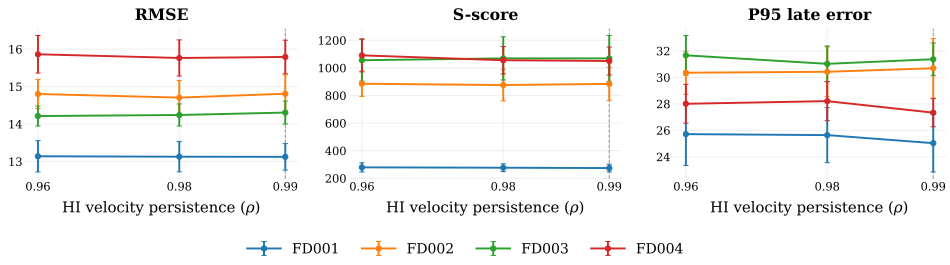

Figure G13. Sensitivity to the velocity persistence. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

S7 - Velocity persistence ρ. HAML is notably insensitive to ρ within the tested high-persistence range $0.96 - 0.995$, with a composite impact of only 1.12% (Figure G13). Lower persistence can slightly improve selected metrics, e.g., FD002 RMSE at 0.98 and FD003 RMSE at 0.96, but no lower value consistently dominates the default. Conversely, $\rho = 0.995$ gives the strongest FD001 and FD004 risk behaviour and maintains the longest degradation memory. The relatively flat response is useful: it indicates that the persistent level–velocity mechanism does not rely on a finely tuned decay constant.

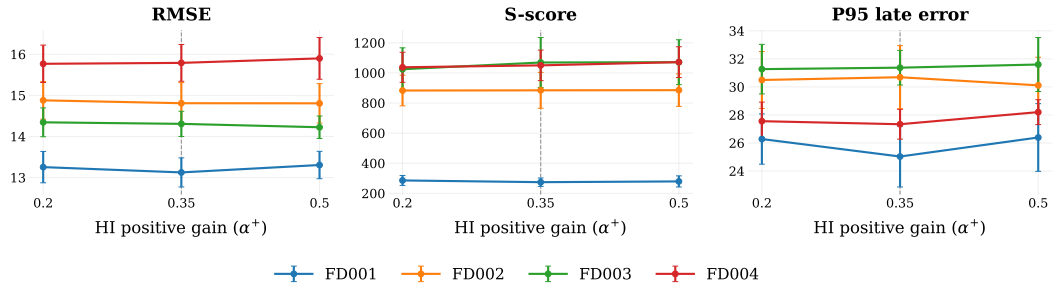

Figure G14. Sensitivity to the positive HI correction gain. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

S8a - Positive HI correction gain α^+. Prediction is relatively stable for $\alpha^+ \in \{0.2, 0.35, 0.5\}$, with a composite impact of approximately 1.52% (Figure G14). The intermediate default is especially well balanced: it gives the best FD001 RMSE and P95, whereas smaller or larger gains offer only dataset-specific improvements. For example, $\alpha^+ = 0.2$ improves FD003/FD004 S-score but worsens FD001 RMSE and P95, while $\alpha^+ = 0.5$ improves FD003 RMSE but worsens FD004. The absence of a monotonic trend is consistent with the role of α^+: overly weak correction can under-react to genuine positive innovations, whereas overly strong correction transfers more transient geometric variation into the HI.

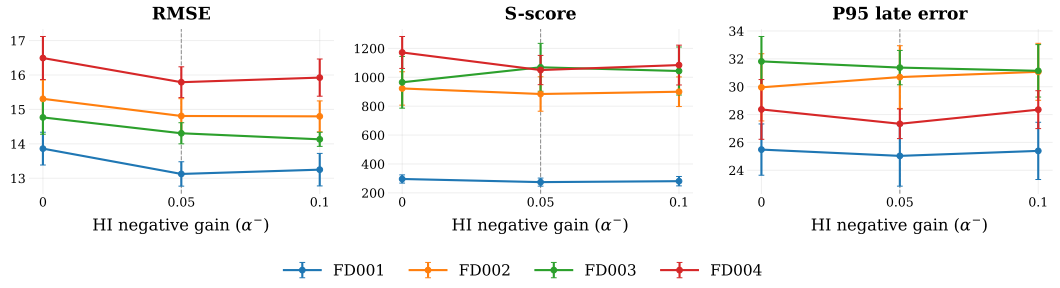

Figure G15. Sensitivity to the negative HI correction gain. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

S8b - Negative HI correction gain α^-. This is the second most sensitive parameter overall (composite impact about 3.32%) and provides particularly strong support for the asymmetric filter design (Figure G15). Setting $\alpha^- = 0$, i.e. preventing negative innovations from correcting the predicted level before monotone projection, markedly worsens RMSE on FD001 (13.13 to 13.86), FD002 (14.81 to 15.31), and FD004 (15.79 to 16.49); FD004 S-score also increases from 1050 to 1172. Although $\alpha^- = 0$ improves FD003 S-score, the cross-dataset pattern strongly favours allowing a small negative correction. Increasing the gain to 0.10 yields mixed results.

S9 - Velocity reinforcement β. The effect of β is non-monotonic, with a composite impact of approximately 1.95% (Figure G16). Small-to-moderate gains ($0.005 - 0.02$) generally provide stable behaviour, while aggressive reinforcement can produce dataset-specific changes. Most notably, $\beta = 0.08$ greatly improves FD003 S-score (1069 to 898) but worsens FD001 RMSE (13.13 to 13.72), S-score (274 to 298), and P95 Late Error (25.03 to 27.34). This is consistent

with the filter mechanism: larger β makes persistent velocity respond more strongly to positive innovations, which can accelerate tracking of abrupt degradation but can also over-propagate transient upward evidence.

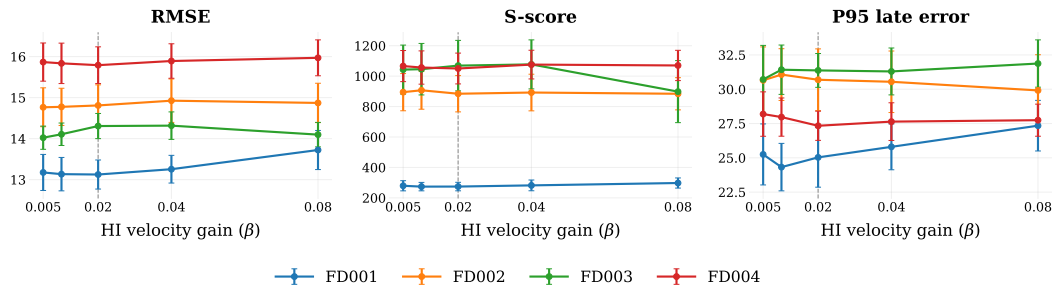

Figure G16. Sensitivity to the velocity reinforcement. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

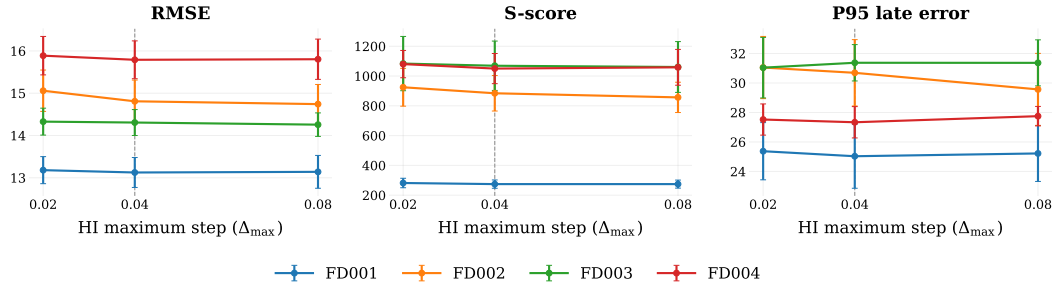

Figure G17. Sensitivity to the maximum HI step. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

S10a - Maximum HI step $\Delta_{\max}$. The step bound has relatively low sensitivity (composite impact about 1.26%) (Figure G17). Tightening it to 0.02 generally worsens prediction, particularly FD002 RMSE/S-score, suggesting that an overly restrictive bound can delay the filtered HI when degradation evolves rapidly. Relaxing it to 0.08 improves FD002 RMSE, S-score, and P95 Late Error and gives modest improvements on FD003, but provides no consistent gain on FD001 or FD004. This empirical behaviour fits well with Theorem 5, where $\Delta_{\max}$ provides a hard bounded-increment guarantee rather than serving principally as an accuracy parameter.

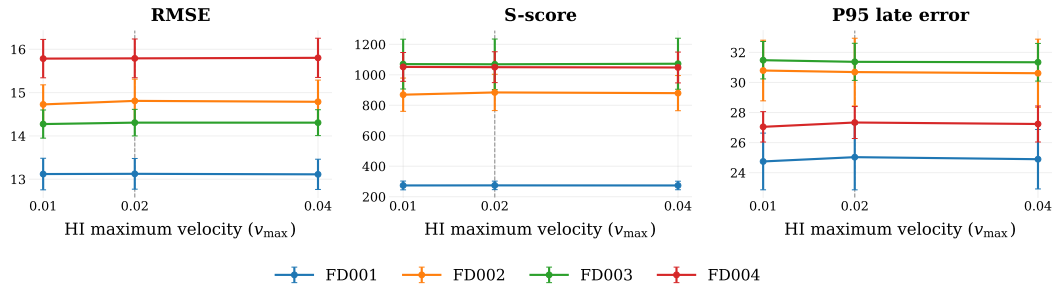

Figure G18. Sensitivity to the maximum degradation velocity. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

S10b - Maximum degradation velocity $v_{\max}$. This is the least sensitive parameter in the entire study, with a composite impact of only about 0.36% (Figure G18). RMSE, S-score, and P95 Late

Error remain almost unchanged for $v_{\max} \in \{0.01, 0.02, 0.04\}$, and differences are very small relative to the seed-wise error bars. This indicates that the velocity cap is rarely the dominant factor controlling prediction within the tested range; rather, it acts primarily as a safety bound preventing unusually large accumulated degradation momentum. The result is favourable theoretically: the bounded-velocity assumption used in the causal-filter analysis does not appear to require precise numerical tuning.

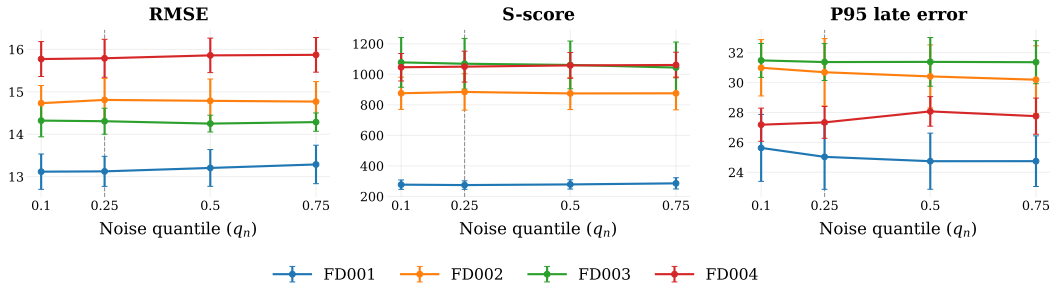

Figure G19. Sensitivity to the early-life noise quantile q_n. Markers show the mean, and whiskers show the sample standard deviation over ten fixed seeds; lower is better for all metrics.

S11 - Early-life noise quantile q_n. HAML is highly insensitive to the quantile used to estimate the early-life noise threshold, with a composite impact of only 0.94% across $q_n \in \{0.10, 0.25, 0.50, 0.75\}$ (Figure G19). RMSE remains nearly unchanged across the four datasets, and most differences in S-score and P95 Late Error are small relative to the seed-wise variability. Increasing q_n raises the learned threshold and therefore suppresses a larger fraction of small positive innovations from reinforcing the persistent degradation velocity. This can slightly improve selected risk metrics; for example, larger q_n reduces P95 Late Error on FD001-FD002 and improves the FD003 S-score, but the effect is not consistent across datasets, with FD004 slightly favouring $q_n = 0.10$. The absence of a common monotonic trend indicates that the learned dead-zone mechanism is not dependent on a finely tuned quantile.

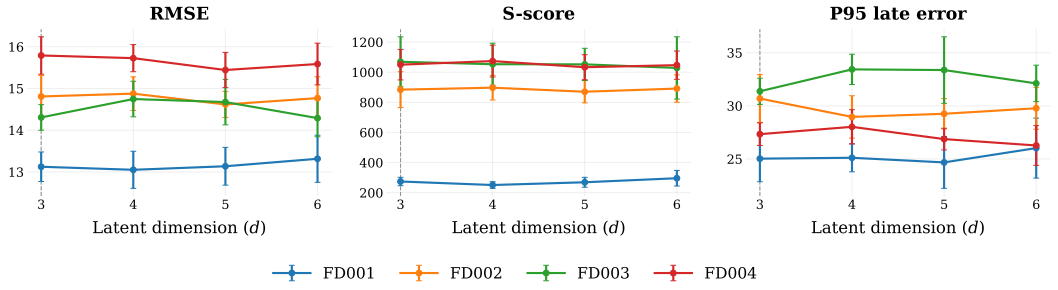

Figure G20. Sensitivity to the latent dimension. Markers show the mean and whiskers show the sample standard deviation over ten fixed seeds; lower is better for all metrics.

S12 - Latent dimension d. Increasing the latent dimensionality from 3 to $4 - 6$ does not produce a consistent improvement (Figure G20). Higher dimensions help selected datasets; for example, $d = 5$ reduces FD004 RMSE from 15.79 to 15.44, while $d = 4$ improves FD002 P95 Late Error from 30.69 to 28.96. However, they can simultaneously degrade other metrics: FD003 P95 Late Error increases substantially to 33.43 and 33.37 for $d = 4$ and 5, respectively. The overall composite impact is about 2.48%, indicating moderate but not severe sensitivity. Thus, $d = 3$ is a reasonable parsimonious choice: it provides sufficient geometric capacity, enables direct trajectory visualisation, and avoids increasing latent complexity without consistent cross-dataset predictive benefit.

S13a - UMAP n_neighbors. The UMAP neighbourhood size is one of the more influential representation parameters, with a composite impact of approximately 3.28% (Figure G21). Its effect is nevertheless strongly dataset- and metric-dependent, with no common optimum across C-MAPSS.

A small neighbourhood (n_neighbors=8) increases RMSE and S-score on FD001-FD002, but substantially improves the FD003 S-score (934.1 versus 1069.1 at the default). Increasing n_neighbors to 24 gives the strongest overall performance on FD002 and also reduces the FD003 P95 Late Error, whereas n_neighbors=40 provides the best RMSE and S-score on FD004 but worsens the FD001 S-score. Thus, neither highly local nor broader neighbourhoods consistently dominate across operating and fault regimes.

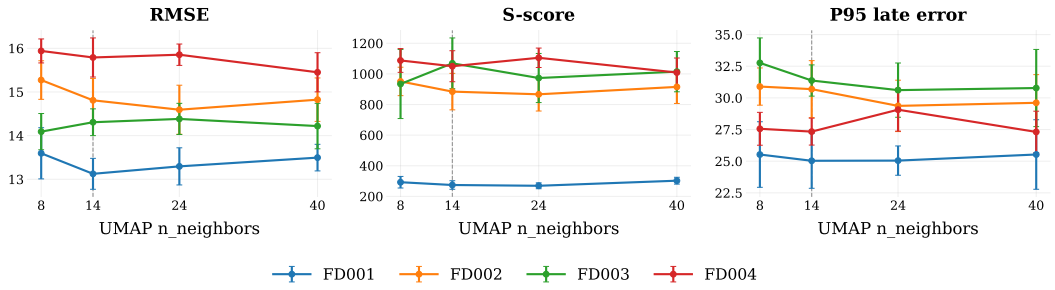

Figure G21. Sensitivity to the UMAP number of neighbours. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

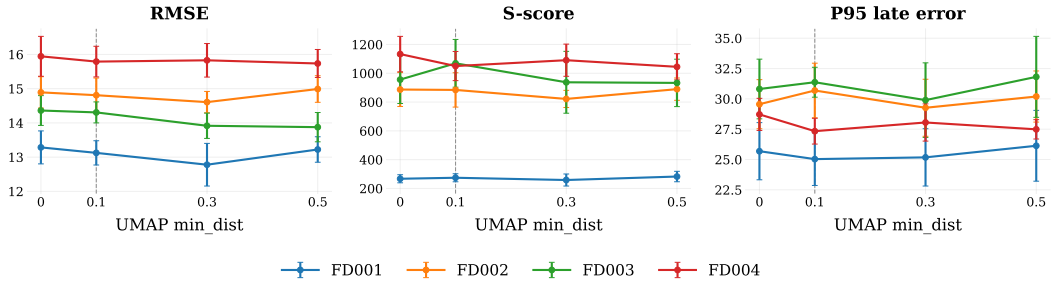

Figure G22. Sensitivity to the UMAP minimum distance. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

S13b - UMAP min_dist. Min_dist is one of the more influential representation parameters, with a composite impact of approximately 3.23% (Figure G22). The response is strongly dataset-dependent. A value of 0.3 improves FD001 and FD002 substantially and also lowers FD003 P95, while 0.5 gives the lowest FD003 RMSE/S-score and slightly improves FD004. Conversely, min_dist=0 substantially worsens FD004 S- score ($\sim$ 1133 vs. 1050 at the default). These results show that the local packing enforced by UMAP can alter the subsequent graph geometry, as expected, but no setting dominates across operating/fault regimes.

Taken together, S13a and S13b show that HAML is moderately sensitive to the local scale and packing of the representation backend, as expected because these choices affect the subsequent landmark-graph topology. However, neither UMAP parameter has a universally preferred setting across datasets. This complements A8: the downstream HAML health construction is representation-backend portable but not representation-invariant, and the primary UMAP configuration does not rely on a single test-optimal neighbourhood or packing parameter.

G5. FURTHER ANALYSIS

Effect of Landmark Sampling Strategy. The landmark-covering result in Theorem 4 assumes theoretical i.i.d. sampling and is not intended as a convergence guarantee for HAML's implemented sampler. We therefore empirically compare the released engine-/condition-balanced round-robin sampler with uniform random landmark sampling without replacement. Both strategies select the same number of landmarks from the identical full-training candidate pool and use the same seeds and frozen HAML configuration; neither uses progress or RUL information during sampling.

We computed the engine and condition Gini coefficient of HAML to quantify engine imbalance when using Uniform and Round-robin sampling. Let ψ_i denote the number of selected landmarks contributed by represented engine i, and the landmark set contains $N_{L,\,eng}$, the engine Gini coefficient is given by:

$$G_{engine} = \frac{\sum_{i=1}^{E_{L,eng}} \sum_{j=1}^{E_{L,eng}} |\psi_i - \psi_j|}{2E_{L,eng} \sum_{i=1}^{E_{L,eng}} \psi_i} \tag{G1}$$

Meanwhile, the condition Gini coefficient used for multi-condition datasets is given by

$$G_{condition} = \frac{\sum_{c=1}^{\mathcal{C}_L} \sum_{j=1}^{\mathcal{C}_L} |\varpi_c - \varpi_j|}{2\mathcal{C}_L \sum_{c=1}^{\mathcal{C}_L} \varpi_c} \tag{G2}$$

where ϖ_i denote the number of selected landmarks belonging to condition $\mathbf{C}_u$, and the landmark set contains $\mathcal{C}_L$ conditions. The results are shown in Table G8.

Table G8. Landmark-allocation balance under uniform random and round-robin sampling. Lower Gini indicates more even landmark allocation; condition-level Gini is not applicable to the single-condition subsets FD001 and FD003.

Dataset	Engine Gini Uniform	Engine Gini Round-robin	Condition Gini Uniform	Condition Gini Round-robin
FD001	0.204	0.000	–	–
FD002	0.275	0.032	0.097	0.006
FD003	0.262	0.000	–	–
FD004	0.294	0.004	0.099	0.001

Table G9. RUL performance under uniform random and round-robin landmark sampling. RMSE, S-score, and P95 Late Error are reported as mean ± sample standard deviation over ten fixed seeds; lower is better for all metrics.

Dataset	Sampler	RMSE ↓	S-score ↓	P95 Late Error ↓
FD001	Uniform random	12.76 ± 0.89	239.1 ± 52.5	23.34 ± 2.83
	Round-robin	13.13 ± 0.50	274.3 ± 39.0	25.03 ± 3.04
FD002	Uniform random	15.07 ± 0.49	939.2 ± 103.4	30.31 ± 3.67
	Round-robin	14.81 ± 0.70	884.3 ± 166.6	30.69 ± 3.15
FD003	Uniform random	14.43 ± 1.16	1028.5 ± 203.2	32.72 ± 2.71
	Round-robin	14.31 ± 0.43	1069.1 ± 231.7	31.37 ± 1.72
FD004	Uniform random	15.67 ± 0.52	1032.7 ± 133.9	29.18 ± 3.39
	Round-robin	15.79 ± 0.63	1050.1 ± 141.5	27.34 ± 1.49

The principal effect of round-robin sampling is improved population balance rather than universal predictive improvement (Table G8-G10). Engine-level landmark Gini coefficients decrease from 0.204, 0.275, 0.262, and 0.294 under uniform sampling to 0.000, 0.032, 0.000, and 0.004 under round-robin sampling on FD001–FD004, respectively. In the multi-condition subsets, condition-level Gini similarly decreases from 0.097 to 0.006 on FD002 and from 0.099 to 0.001 on FD004.

Round-robin also covers every training engine in all four datasets. These differences confirm that uniform sampling over windows can over-represent long or densely sampled trajectories, whereas round-robin explicitly controls trajectory representation.

Table G10. HI diagnostics under uniform random and round-robin landmark sampling. All metrics are reported as mean ± sample standard deviation over ten fixed seeds. Higher is better for monotonicity, HI-progress correlation, prognosability, and healthy-failure separation; lower is better for mean HI step. Monotonicity and bounded step behaviour are directly influenced by the causal-filter constraints and are therefore interpreted as constraint-satisfaction diagnostics, whereas progress correlation, prognosability, and healthy-failure separation provide complementary measures of HI quality.

Dataset	Sampler	Monotonicity ↑	Progress corr. ↑	Prognosability ↑	Mean HI step ↓	Healthy–Failure sep ↑
FD001	Uniform random	1.0000 ± 0.0000	0.9062 ± 0.0041	0.9969 ± 0.0076	0.006105 ± 0.000022	0.9153 ± 0.0088
	Round-robin	1.0000 ± 0.0000	0.9023 ± 0.0057	0.9971 ± 0.0040	0.006095 ± 0.000025	0.9133 ± 0.0103
FD002	Uniform random	1.0000 ± 0.0000	0.8896 ± 0.0098	0.9822 ± 0.0464	0.005835 ± 0.000241	0.8735 ± 0.0405
	Round-robin	1.0000 ± 0.0000	0.8923 ± 0.0084	0.9975 ± 0.0023	0.005973 ± 0.000040	0.8978 ± 0.0086
FD003	Uniform random	1.0000 ± 0.0000	0.8654 ± 0.0063	0.9998 ± 0.0005	0.004772 ± 0.000033	0.8895 ± 0.0083
	Round-robin	1.0000 ± 0.0000	0.8724 ± 0.0097	0.9983 ± 0.0028	0.004782 ± 0.000028	0.8905 ± 0.0073
FD004	Uniform random	1.0000 ± 0.0000	0.8441 ± 0.0090	0.9943 ± 0.0047	0.004628 ± 0.000049	0.8560 ± 0.0108
	Round-robin	1.0000 ± 0.0000	0.8567 ± 0.0115	0.9819 ± 0.0220	0.004461 ± 0.000408	0.8016 ± 0.1089

This balancing does not lead to uniform improvements in RUL prediction. Uniform sampling performs better on RMSE, S-score, and P95 Late Error on FD001; round-robin improves RMSE and S-score on FD002, RMSE and P95 Late Error on FD003, and P95 Late Error on FD004. The maximum latent nearest-landmark radius is also slightly larger under round-robin, showing that improved engine/condition balance is distinct from minimising worst-case Euclidean latent coverage. Hence, the empirical role of the round-robin sampler is to provide a more balanced finite representation of the training population, rather than to optimise prediction accuracy or latent covering radius.

The HI analysis leads to the same conclusion. Both samplers retain monotonicity of 1.000 on every dataset because monotonicity is imposed downstream by the causal HI filter. Round-robin yields slightly higher HI-progress correlation on FD002–FD004, but other properties are mixed. In particular, FD004 shows higher progress correlation under round-robin (0.857 versus 0.844) but lower and more variable healthy–failure separation (0.802 ± 0.109 vs. 0.856 ± 0.011). Thus, balanced landmark selection modifies the finite graph geometry without uniformly improving every HI characteristic.

Sensitivity to Progress-Supervised Engine Fraction. HAML introduces normalised lifecycle progress only after the latent representation has been constructed without lifecycle-progress or RUL supervision; progress is then used to weight health-graph edges and define healthy/failure reference regions. To test whether this geometric supervision must be available for every training trajectory, we restrict progress-supervised graph construction to nested subsets containing 10%, 25%, 50%, and 100% of the training engines. Subsets are selected at the engine level and balanced across available operating-condition and lifespan strata. The representation model remains fitted using the standard full-training protocol: landmark selection may use engine/condition metadata for balancing, but lifecycle progress and RUL targets are not used to supervise representation fitting. The results are shown in Figure G23.

Reducing the progress-supervised engine fraction produces surprisingly small changes in prediction accuracy. At 10% progress supervision, RMSE is 13.67, 15.09, 13.96, and 15.89 on FD001–FD004, compared with 13.13, 14.81, 14.31, and 15.79 under 100% progress supervision. The corresponding P95 Late Errors remain similarly close. Moreover, performance is not monotonic in the amount of

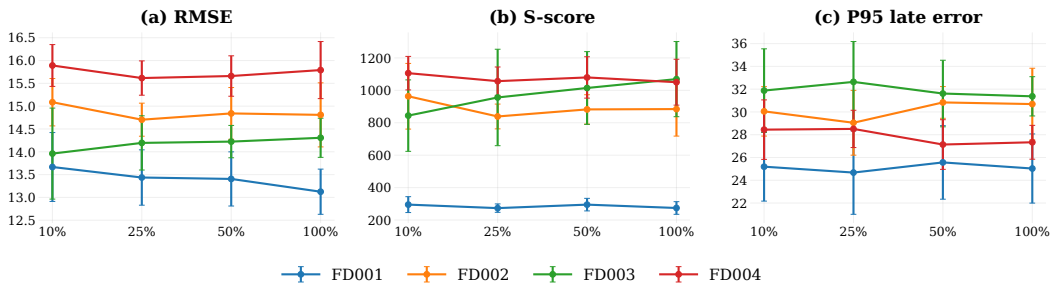

Figure G23. Sensitivity to lifecycle-progress supervision in health-graph construction. Markers show the mean and whiskers the sample standard deviation over ten fixed seeds; lower is better for all metrics.

progress supervision: 25% progress supervision gives the lowest RMSE, S-score, and P95 Late Error on FD002, while the 10% configuration improves RMSE and S-score relative to 100% on FD003. These occasional improvements should not be interpreted as evidence that reducing supervision is intrinsically beneficial; restricting the progress-supervised anchors changes the finite graph topology and can act as a representation perturbation to which the refitted full-data decoder adapts.

Overall, the experiment indicates that the degradation-aware graph does not rely on lifecycle-progress supervision from all training engines. A comparatively small and representative subset can provide sufficient lifecycle structure to preserve useful graph-geodesic coordinates. This result complements, rather than replaces, the standard HAML benchmark: the 100%-progress-supervised graph configuration exactly recovers the released round-robin HAML baseline used in the main benchmark tables. Importantly, RUL supervision for the downstream decoder remains unchanged across all configurations.

G6. ABLATION STUDY

We conduct eight controlled ablations of HAML (A1–A8) by modifying one mechanism at a time while keeping the data preprocessing, temporal descriptor encoder, representation backend, downstream evaluation protocol, and seed schedule fixed whenever the ablated component does not require refitting. A1–A4 investigate the post-representation degradation geometry and graph topology, A5–A6 investigate the conversion of the raw geometric observation into a causal degradation state, A7 investigates the information hierarchy used by the RUL readout, and A8 examines dependence on the low-dimensional representation backend. Together, these ablations test the intended separation between representation learning, degradation-aware metric construction, causal state estimation, and downstream RUL decoding.

A0 (HAML using UMAP and 49 features): This is the full HAML framework as a reference for the other ablation studies.

A1 (Graph-geodesic vs. Euclidean health coordinate): In this case, the OOS distances will be:

$$d_H^{(Euclidean)}(\mathbf{z}) = \min_{h \in \mathcal{H}} \| \mathbf{z} - \mathbf{z}_h \|_2 \tag{G3}$$

$$d_F^{(\text{Euclidean})}(\mathbf{z}) = \min_{f \in \mathcal{F}} \|\mathbf{z} - \mathbf{z}_f\|_2. \tag{G4}$$

followed by

$$s^{(Euclidean)}(\mathbf{z}) = \frac{d_H^{(Euclidean)}(\mathbf{z})}{d_H^{(Euclidean)}(\mathbf{z}) + d_F^{(Euclidean)}(\mathbf{z})} \tag{G5}$$

A1 tests whether the post-representation graph metric provides degradation information beyond direct Euclidean proximity in the same fixed latent representation.

The results indicate a modest but consistent structural benefit from graph-geodesic health coordinates. Compared to A1, the full HAML configuration achieves lower RMSE on FD001 and FD002, identical mean RMSE on FD004, and only slightly higher RMSE on FD003. More importantly, from a risk perspective, A0 gives a lower P95 Late Error on all four datasets, including 25.03 vs. 25.16 on FD001, 30.69 vs. 30.76 on FD002, 31.37 vs. 31.55 on FD003, and 27.34 vs. 28.08 on FD004. The S-score is more mixed, with the Euclidean variant performing better on FD003 and marginally on FD004.

The HI metrics provide clearer evidence for the geodesic formulation. A0 produces a higher HI-progress correlation on every dataset: 0.902 vs. 0.883 on FD001, 0.892 vs. 0.873 on FD002, 0.872 vs. 0.847 on FD003, and 0.857 vs. 0.834 on FD004. This pattern suggests that direct Euclidean proximity does not fully preserve lifecycle ordering, whereas shortest-path distances over the health-aware graph better align latent position with degradation progression. Although the improvement is not large enough to argue that Euclidean distances fail, A1 supports the more precise conclusion that graph-geodesic topology provides additional lifecycle-consistent structure and slightly better control of late-error tails beyond Euclidean proximity.

A2 (Contribution of the progress-aware edge penalty): In the full HAML framework, the spatial weight is given by $w_{uv}^{(s)} = \| \mathbf{z}_u - \mathbf{z}_v \|_2 \ (1 + \lambda_G \, |p_u - p_v|)$. A2 investigates the necessity of a health-aware penalty in graph construction. By removing the penalty, the new weight will be $w_{uv}^{(s)} = \| \mathbf{z}_u - \mathbf{z}_v \|_2$

The effect on overall RMSE is small but remarkably consistent: removing the penalty increases RMSE on all four datasets, from 13.13 to 13.24 on FD001, 14.81 to 14.86 on FD002, 14.31 to 14.37 on FD003, and 15.79 to 15.94 on FD004. The effect is stronger for FD004 tail risk, where P95 Late Error increases from 27.34 to 29.15. S-score also deteriorates on FD001, FD002, and FD004, although A2 obtains a lower S-score on FD003.

The HI results reinforce this interpretation. Removing the progress penalty reduces HI–progress correlation on every dataset, from 0.902 to 0.886, 0.892 to 0.875, 0.872 to 0.852, and 0.857 to 0.838 for FD001–FD004, respectively.

Thus, the principal benefit of the progress-aware penalty appears to be better lifecycle ordering of the graph rather than a large direct reduction in RMSE. This is consistent with HAML's intended supervision placement: lifecycle progress does not alter the fitted latent coordinates but changes the costs of traversing between them, thereby modifying the degradation metric from which the health coordinate is constructed.

A3 (Multi-source reference regions vs. single medoids): A3 investigates the necessity of maintaining multi-source healthy and failure reference regions, denoted as $\mathcal{H}$ and $\mathcal{F}$, for constructing a robust health coordinate. In the full HAML framework, the out-of-sample graph-geodesic distances $d_H(z)$ and $d_F(z)$ of a latent state z are computed relative to the closest landmarks within these diverse reference sets. A3 simplifies this approach by collapsing each region into a single graph-central medoid:

$$h^* = \arg\min_{h \in \mathcal{H}} \sum_{u \in \mathcal{H}} d_G(h, u), \quad f^* = \arg\min_{f \in \mathcal{F}} \sum_{u \in \mathcal{F}} d_G(f, u) \tag{G6}$$

Then define

$$\begin{aligned} d_H^{A3}(\mathbf{z}) &= \min_{j \in \mathcal{N}_{K_{\text{OOS}}}(\mathbf{z})} \left[\|\mathbf{z} - \mathbf{z}_j\|_2 + d_G(j, h^*) \right], \\ d_F^{A3}(\mathbf{z}) &= \min_{j \in \mathcal{N}_{K_{\text{OOS}}}(\mathbf{z})} \left[\|\mathbf{z} - \mathbf{z}_j\|_2 + d_G(j, f^*) \right]. \end{aligned} \tag{G7}$$

The raw health coordinate, $s(\mathbf{z})$ is subsequently computed using the relative graph-geodesic distances strictly to these singular medoids, h^* and f^*:

$$s(\mathbf{z}) = \frac{d_H^{A3}(\mathbf{z})}{d_H^{A3}(\mathbf{z}) + d_F^{A3}(\mathbf{z})} \tag{G8}$$

Among the graph-related ablations, A3 provides the strongest evidence for a structurally important HAML component. Prediction results alone are mixed: RMSE increases notably on FD001 from 13.13 to 14.06 and slightly on FD003 and FD004, while remaining essentially unchanged on FD002. Interestingly, A3 produces a substantially lower S-score on FD003 (568.4 vs. 1069.1) despite increasing the P95 Late Error from 31.37 to 33.16. The unusually low FD003 S-score should therefore not be interpreted as evidence that the single-medoid representation is superior; taken together with the larger P95 Late Error and the HI results, it is more consistent with a change in prediction calibration that happens to be favoured by the asymmetric S-score.

The deterioration of the HI representation is much clearer. Healthy–failure separation falls from 0.912 to 0.585 on FD001, 0.897 to 0.654 on FD002, 0.889 to 0.359 on FD003, and 0.800 to only 0.291 on FD004. Prognosability similarly falls from 0.997 to 0.810, 0.997 to 0.899, 0.998 to 0.472, and 0.982 to 0.465, respectively. The HI–progress correlation also collapses particularly strongly on FD003 and FD004, from 0.872 to 0.531 and from 0.857 to 0.482.

These results show that a single prototype cannot adequately represent the diversity of states occurring near the healthy and failure boundaries. Multiple references provide a substantially more stable definition of the two ends of the health coordinate, particularly on FD003 and FD004. A3 therefore supports the use of distributed endpoint geometry rather than single healthy/failure prototypes: health in HAML is defined relative to regions of plausible endpoint states under the graph metric, rather than relative to one representative state or one latent coordinate.

A4 (Contribution of temporal trajectory edges): A4 removes the temporal edge with weights $w_{uv}^{(t)} = \eta_T \widetilde{d}_{uv}\sqrt{\Delta c_{uv}}$. By doing this, the final weight will be a spatial or bridge weight $w_{uv} = \min\left\{w_{uv}^{(s)}, w_{uv}^{(b)}\right\}$. Ablation A4 investigates the necessity of temporal edges in graph construction.

Unlike A3, the A4 results indicate that the incremental contribution of temporal edges is relatively modest under the current configuration. Removing them increases RMSE slightly on FD001 (13.13 to 13.39), FD002 (14.81 to 14.84), and FD004 (15.79 to 15.83), but improves RMSE on FD003 from 14.31 to 14.14. The same mixed behaviour is visible in the S-score and P95 Late Error measures, with A4 performing better than A0 on all three prediction metrics for FD003. The HI metrics are also very similar. HI–progress correlation changes only from 0.902 to 0.899 on FD001, 0.892 to 0.888 on FD002, 0.872 to 0.870 on FD003, and 0.857 to 0.853 on FD004. Healthy-failure separation is actually slightly higher for A4 on each dataset.

Consequently, the present experiment does not support a strong claim that temporal edges are indispensable. A more defensible interpretation is that they provide a mild temporal inductive bias and small improvements in several datasets, while most of the useful degradation structure can already be recovered from the progress-aware spatial graph and connectivity mechanism. The FD003 result further indicates that temporal connectivity is not universally beneficial. This is an important distinction because it prevents over-attributing HAML's performance to a component whose measured marginal effect is small.

A5 (Contribution of the causal monotonic HI filter): Raw health coordinates s_t will play the role of h_t for the RUL prediction.

The predictive results reveal a clear trade-off. On clean C-MAPSS test data, removing the filter improves all three reported prediction metrics on every dataset. RMSE decreases from 13.13 to 12.77 on FD001, 14.81 to 14.73 on FD002, 14.31 to 13.73 on FD003, and 15.79 to 15.55 on FD004. S-score also decreases consistently, while P95 Late Error falls from 25.03 to 24.17, 30.69 to 26.27, 31.37 to 30.23, and 27.34 to 26.82, respectively. Therefore, the causal filter should not be presented as improving clean-data predictive accuracy.

Its contribution is instead evident in the trajectory constraints and in several complementary HI characteristics. Without filtering, the non-decreasing constraint is no longer enforced: measured monotonicity falls from the construction-imposed value of 1.000 to 0.540, 0.527, 0.531, and 0.520 on FD001–FD004. Mean one-cycle HI variation also rises substantially, as expected when the bounded-step state update is removed: from 0.0061 to 0.0437 on FD001, 0.0060 to 0.0562 on FD002, 0.0048 to 0.0449 on FD003, and 0.0044 to 0.0649 on FD004. More importantly for representation quality, prognosability and healthy–failure separation also decline, although these properties are not directly guaranteed by the monotonic projection.

A5 therefore demonstrates one of the central design trade-offs in HAML: the causal filter exchanges a small amount of clean-data predictive accuracy for an explicitly constrained degradation state with non- decreasing evolution, bounded local variation, and improved endpoint characteristics. This is consistent with its role as a structural prior rather than a conventional accuracy-enhancement module. The result also motivates evaluating the filter through complementary criteria rather than RMSE alone: monotonicity and step size verify the intended trajectory constraints, while progress correlation, prognosability, healthy–failure separation, and robustness experiments test whether those constraints produce a useful degradation state. In particular, A5 verifies that HAML's monotonicity is introduced by the proposed filter rather than emerging incidentally from the raw graph coordinate; this should be interpreted as a mechanism-validation result rather than as independent evidence of health-coordinate quality.

A6 (Persistent level–velocity dynamics vs. cumulative maximum): A6 is designed to isolate the contribution of the proposed causal monotonic Health Index (HI) filter from the inherent property of monotonicity itself. A6 replaces the persistent level–velocity dynamic state in the full HAML with a simpler cumulative-maximum monotonic envelope applied directly to the raw coordinates:

$$h_t = \max_{\vartheta \le t} s_\vartheta, \quad v_t = \max\left(h_t - h_{t-1}, 0\right) \tag{G9}$$

Therefore, the comparison between the full model and A6 does not contrast monotonic against non-monotonic behaviour, because both formulations enforce non-decreasing HI trajectories. Rather, it empirically evaluates whether the persistent level–velocity update, which admits the local convex-projection interpretation given in Proposition 6, provides useful degradation memory beyond a basic cumulative-maximum constraint.

Relative to full HAML, cumulative-max filtering of A6 increases RMSE on all four datasets: 13.13 to 14.01 on FD001, 14.81 to 15.10 on FD002, 14.31 to 14.46 on FD003, and 15.79 to 16.18 on FD004. The S-score is also worse on FD001, FD002, and FD004, although A6 gives a slightly lower S-score on FD003. P95 Late Error is mixed, indicating that persistent velocity primarily improves overall trajectory decoding rather than uniformly dominating every tail-risk statistic.

Both methods achieve monotonicity of 1.000 by construction, demonstrating that satisfaction of the monotonic constraint alone cannot explain the predictive advantage of A0. The persistent filter also yields consistently higher prognosability and healthy–failure separation. For example, separation improves from 0.879 to 0.912 on FD001, 0.854 to 0.897 on FD002, 0.858 to 0.889 on FD003, and 0.765 to 0.800 on FD004. Interestingly, cumulative maximum gives a slightly higher HI–progress correlation on FD001-FD003 and the same value on FD004.

This combination is informative. A cumulative maximum is effective at enforcing simple ordering, but it does not retain information about degradation momentum or distinguish sustained deterioration from isolated coordinate excursions. The improved RUL performance, prognosability, and endpoint separation of A0 therefore support the intended role of the persistent level-velocity state: it provides degradation memory beyond a purely monotonic envelope.

A7 (Nested RUL-decoder feature hierarchy): A7 tests whether the scalar HI alone is sufficient for RUL prediction and quantifies the incremental contribution of HI history, temporal dynamics, persistent filter state, and latent-geometric information while leaving the upstream HAML health construction unchanged. Five nested decoder configurations are evaluated:

- A7a: current filtered HI only (1 feature).
- A7b: 30-step filtered-HI history (30 features).
- A7c: HI-centred temporal representation comprising HI history and derived temporal summaries (44 features)
- A7d: A7c plus the raw graph coordinate and persistent velocity state (46 features).
- A0: A7d plus the three standardised latent coordinates (49 features).

The nested feature ladder provides direct evidence that the HAML health representation is HI-centred but not reducible to the instantaneous scalar HI. Using only the current scalar HI (A7a) results in substantial degradation on all datasets. RMSE rises to 16.08, 18.81, 17.88, and 22.11 on FD001-FD004, compared with 13.13, 14.81, 14.31, and 15.79 for the full 49-feature representation.

The effect is particularly severe on FD004, where the S-score increases from 1050.1 to 4258.2 and P95 Late Error from 27.34 to 38.22.

Adding the 30-step HI history in A7b produces a clear improvement over the current-HI-only model, showing that the instantaneous degradation stage is insufficient for RUL estimation: remaining life also depends on how the asset arrived at its current health state. A substantially larger improvement occurs when the temporal summaries are introduced in A7c. For example, RMSE decreases from 19.78 in A7b to 17.29 in A7c on FD004 and from 17.08 to 14.97 on FD003. This demonstrates that multi-horizon slopes, changes, global trajectory statistics, degradation activity, and the rate-based estimate capture useful information that cannot be recovered efficiently from the raw fixed-length history alone.

Adding the raw graph coordinate and persistent velocity in A7d produces smaller but generally favourable additional gains, showing that pre-filter geometric position and degradation momentum provide information complementary to the HI trajectory. Finally, restoring the three standardised latent coordinates to obtain the complete 49-dimensional A0 representation further reduces RMSE on all four datasets and lowers P95 Late Error substantially, including 27.20 to 25.03 on FD001, 35.23 to 31.37 on FD003, and 30.21 to 27.34 on FD004. This indicates that the one-dimensional HI and its temporal dynamics do not completely preserve the multidimensional latent state, particularly where operating-condition or fault-mode heterogeneity is present.

The A7 ladder therefore establishes a clear information hierarchy. The scalar HI provides the primary interpretable degradation-stage coordinate, but is not a sufficient statistic for RUL. Its causal history and derived temporal summaries capture how the asset reached its current state and how degradation is evolving, while the persistent velocity, raw graph coordinate, and latent coordinates retain complementary dynamic and geometric information. The full decoder should therefore be interpreted as an HI-centred causal prognostic representation rather than as an HI-only regressor or an arbitrary feature expansion. The only notable exception is the FD003 S-score, for which some reduced representations obtain lower values than A0 despite substantially worse RMSE and P95 Late Error. This again indicates that conclusions should be based jointly on accuracy and tail-risk metrics rather than on the asymmetric score alone.

A8 (Representation-backend substitution): We compare the pre-specified UMAP representation with PCA, Isomap, PHATE, and Diffusion Maps while keeping the downstream HAML health-graph construction, reference regions, causal filter, decoder, and evaluation protocol unchanged. The purpose of A8 is to test representation-backend portability and sensitivity, not to use test performance to select a replacement for the primary UMAP configuration.

On FD001, PHATE provides the lowest RMSE (12.97) and P95 Late Error (24.24), while Isomap gives the lowest S-score (261.7), all competitive with or better than UMAP. On FD002, PCA clearly outperforms UMAP across all three metrics, reducing RMSE from 14.81 to 13.55, S-score from 884.3 to 727.9, and P95 Late Error from 30.69 to 27.43. The effect is even more pronounced on FD003, where PCA reduces RMSE from 14.31 to 12.91 and the S-score from 1069.1 to 322.3, while also slightly reducing P95 Late Error. These improvements do not imply that PCA should retrospectively replace UMAP in the primary HAML results, because backend choice was not tuned per dataset using test performance. Rather, they show that the representation supplied to the downstream health-aware layer remains an important source of performance variation.

Conversely, the pre-specified UMAP configuration performs best among the tested backends on FD004: all four alternative backends produce higher RMSE, S-score, and P95 Late Error. For example, PCA gives RMSE/S-score/P95 Late Error of 16.04/1104.2/28.01 compared with 15.79/1050.1/27.34 for UMAP, while Diffusion Maps deteriorates to 16.79/1389.8/30.18. There is therefore no universally optimal representation backend. This also explains why A8 is treated as a portability/sensitivity analysis rather than as a procedure for dataset-specific backend optimisation.

The HI results lead to the same conclusion. Because the same causal monotonic filter is retained, every backend satisfies the imposed non-decreasing constraint (monotonicity $= 1.000$); this metric therefore does not distinguish representation quality across A8. Nevertheless, the pre-specified UMAP representation achieves the highest HI-progress correlation among the tested backends on all four datasets, whereas other representations can provide stronger endpoint separation. Diffusion Maps, for example, produces healthy-failure separations of 0.941, 0.925, 0.910, and 0.891 on FD001-FD004. This illustrates that no single HI property determines final RUL performance

and that different representation geometries preserve different aspects of the degradation structure. Hence, backend portability should not be confused with backend equivalence.

Overall, A8 supports representation-backend portability with representation-dependent performance, rather than representation invariance. The ability to retain the same health-aware graph, reference-distance construction, causal filter, and decoder over substantially different representations indicates that the downstream HAML machinery is not architecturally restricted to UMAP. However, the sizeable performance differences across backends, particularly the strong PCA results on FD002/FD003 and the stronger UMAP result on FD004, show that the quality and geometry of the supplied representation remain consequential. A8 therefore establishes portability of the health-aware layer, not interchangeability or universal optimality of the tested backends.

H. Limitations and Future Work

Several limitations define the scope of the present study. First, HAML assumes that degradation is irreversible at the sampling scale and therefore enforces a non-decreasing HI. This prior is appropriate for the run-to-failure processes considered here but may be unsuitable for systems exhibiting genuine recovery, maintenance interventions, component replacement, or reversible operating effects. Extending the state model to distinguish degradation from temporary recovery or intervention events is an important direction for future work.

Second, although the latent representation and landmark selection are label-free, HAML uses normalised lifecycle progress from run-to-failure training trajectories to construct the health-aware graph and its healthy/failure reference regions. The progress-supervision experiment indicates that useful geometry can be retained when only a subset of engines supplies this information; however, the downstream RUL decoder still uses the complete training targets. The current results therefore do not establish semi-supervised or weakly supervised RUL learning.

Third, HAML is backend-portable but not representation-invariant. The ablation study shows that PCA, Isomap, PHATE, Diffusion Maps, and UMAP produce different predictive and HI characteristics, while the sensitivity analysis identifies the landmark budget and selected representation/filter parameters as comparatively influential. The present implementation also constructs the representation and health-aware graph sequentially rather than learning them jointly. Developing representations whose geometry is intrinsically compatible with degradation-aware graph construction may further improve robustness and reduce dependence on backend choices.

Fourth, landmark-based graph construction introduces additional computational and memory costs relative to lightweight recurrent or sparse graph models. In particular, the landmark graph requires shortest-path computation and storage of landmark-to-landmark geodesic distances. Although the fixed 1,500-landmark configuration remains computationally practical and is substantially faster to train than several heavier Transformer and modern time-series baselines, scalability to much larger landmark sets and high-frequency industrial fleets warrants further investigation.

Fifth, the PHM12 experiments follow the established per-bearing normalised benchmark protocol (Wang et al., 2025b), in which each bearing's full-life maximal RUL is used to construct normalised ground truth and to express selected evaluation errors in absolute benchmark RUL units. RMSE and late-error statistics are reported in absolute benchmark RUL units after applying the bearing-specific benchmark scaling described above. This quantity is not available to or used by the predictor during forward inference, but the resulting absolute-cycle scores should nevertheless be interpreted as benchmark evaluations rather than as a deployment protocol in which an unseen asset's eventual lifetime is known a priori.

Finally, evaluation is restricted to the C-MAPSS and PHM12 run-to-failure benchmarks, and the robustness study considers controlled additive Gaussian sensor noise applied after training. These experiments do not capture the full range of deployment conditions, including sensor drift, missing channels, impulsive faults, calibration changes, unseen operating regimes, maintenance interventions, or cross-domain distribution shifts. HAML also retains concentrated late-prediction risk on some FD003 engines, demonstrating that monotonic health-state construction does not eliminate extreme RUL errors. Future work should therefore evaluate the framework on industrial longitudinal datasets, broader corruption and domain-shift scenarios, and safety-aware decoders explicitly designed to control extreme RUL overestimation. Finally, the theoretical results characterise specific

components of HAML under stated assumptions, including locally fixed OOS neighbourhoods and conditional landmark/reference-distance approximation, and should not be interpreted as an end-to-end convergence guarantee for the complete implemented pipeline.

REFERENCES FOR THE APPENDIX

Dovile Juodelyte, Veronika Cheplygina, Therese Graversen, and Philippe Bonnet. Predicting bearings degradation stages for predictive maintenance in the pharmaceutical industry. In *Proceedings of the 28th acm sigkdd conference on knowledge discovery and data mining*, pp. 3107–3115, 2022.

Yaguo Lei, Zhengjia He, and Yanyang Zi. A new approach to intelligent fault diagnosis of rotating machinery. *Expert Systems with applications*, 35(4):1593–1600, 2008.

Jiayang Liu, Xiaosun Wang, Fuqi Xie, Shijing Wu, and Deng Li. Condition monitoring of wind turbines with the implementation of spatio-temporal graph neural network. *Engineering Applications of Artificial Intelligence*, 121:106000, 2023.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

P Nectoux, R GOURIVEAU, K MEDJAHER, et al. An experimentalplatform forbearingsaccelerated degradation tests. In *IEEE. International Conference on Prognostics and Health Management. Beijing:[sn] 2012*, pp. 23–25, 2012.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

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. IEEE, 2008.

Trung Hieu Vu, Eyad Elyan, Will Vorley, Joe Goodlad, Truong Dang, and Tien Thanh Nguyen. Fat: Fusion-attention transformer for remaining useful life prediction. In *International Conference on Pattern Recognition*, pp. 286–301. Springer, 2024.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. In *International Conference on Learning Representations*, volume 2025, pp. 16980–17016, 2025a.

Yucheng Wang, Min Wu, Ruibing Jin, Xiaoli Li, Lihua Xie, and Zhenghua Chen. Local–global correlation fusion-based graph neural network for remaining useful life prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):753–766, 2023.

Yucheng Wang, Yuecong Xu, Jianfei Yang, Min Wu, Xiaoli Li, Lihua Xie, and Zhenghua Chen. Fully-connected spatial-temporal graph for multivariate time-series data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 15715–15724, 2024.

Yucheng Wang, Min Wu, Xiaoli Li, Lihua Xie, and Zhenghua Chen. A survey on graph neural networks for remaining useful life prediction: Methodologies, evaluation and future trends. *Mechanical Systems and Signal Processing*, 229:112449, 2025b.

Yupeng Wei, Dazhong Wu, and Janis Terpenny. Bearing remaining useful life prediction using self-adaptive graph convolutional networks with self-attention mechanism. *Mechanical Systems and Signal Processing*, 188:110010, 2023.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023.