HEPA: A SELF-SUPERVISED HORIZON-CONDITIONED EVENT PREDICTIVE ARCHITECTURE FOR TIME SERIES

Anonymous authors
Paper under double-blind review

https://anonymous.4open.science/r/HEPA-8D6B

ABSTRACT

Critical events in multivariate time series, from turbine failures to cardiac arrhythmias, demand accurate prediction, yet labeled data is scarce because such events are rare and costly to annotate. We introduce HEPA (Horizon-conditioned Event Predictive Architecture), built on two key principles. First, a causal Transformer encoder is pretrained via a Joint-Embedding Predictive Architecture (JEPA): a horizon-conditioned predictor learns to forecast future *representations* rather than future values, encouraging the encoder to capture predictable temporal dynamics from unlabeled data alone. Second, we freeze the encoder and *finetune only the predictor* toward the target event, producing a monotonic survival cumulative distribution function (CDF) over horizons. With fixed architecture and optimiser hyperparameters across all benchmarks, HEPA handles water contamination, cyberattack detection, volatility regimes, and eight further event types across 11 domains, matching or exceeding leading time-series architectures including PatchTST, iTransformer, MAE, and Chronos-2 on at least 10 of 14 benchmarks each, with an order of magnitude fewer tuned parameters and, on lifecycle datasets, an order of magnitude less labeled data.

1 INTRODUCTION

A turbine blade cracks after 12,000 flight hours. A bearing degrades over weeks of vibration data. A satellite sensor drifts silently for 48 hours before triggering a cascade. These events are rare in operational data, yet they follow partially predictable precursor dynamics (Scheffer et al., 2009): temperatures rise gradually before overheating, vibration amplitudes grow before mechanical failure, and sensor readings deviate systematically before spacecraft faults. A range of machine-learning methods attempt to predict such events from multivariate sensor streams. Remaining-useful-life (RUL) models (Fan et al., 2024) estimate how long until a machine fails; anomaly detectors (Xu et al., 2022; He et al., 2026) flag when sensor readings look abnormal. Although general-purpose architectures exist for both, the two communities develop separate benchmarks, metrics, and evaluation protocols: RUL models never see anomaly benchmarks; anomaly detectors never forecast time-to-failure. Yet all these tasks share the same structure: given observations up to time t, estimate the probability $P(\text{event within } \Delta t)$ for each prediction horizon Δt.

This structural uniformity suggests a separation of concerns. The *encoder* learns temporal dynamics from unlabeled data without knowing which event matters downstream. The *predictor*, finetuned with a small number of event labels, specialises the learned dynamics to whichever event is relevant. The key design choice is what the encoder should forecast during pretraining. Value-forecasting approaches, whether supervised (Nie et al., 2023) or pretrained on large corpora (Ansari et al., 2025; Das et al., 2024), shape representations around all variation in the signal, including noise irrelevant to the downstream event. The Joint-Embedding Predictive Architecture (JEPA) (Assran et al., 2023) offers an alternative: by forecasting future *representations* rather than future values, the encoder learns a latent space that retains what is predictable about the future and discards what is not.

We apply this principle to time series as HEPA (Horizon-conditioned Event Predictive Architecture). A causal Transformer encodes observations up to time t; a horizon-conditioned predictor maps the encoding and a horizon Δt to a predicted future representation, encouraging the encoder to

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

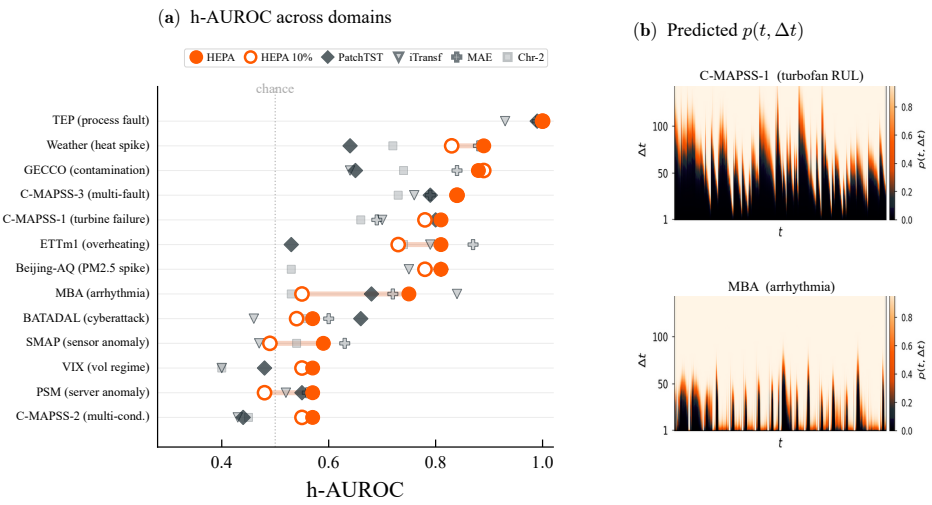

Figure 1: **One label-efficient architecture, domain- and event-agnostic.** (a) h-AUROC (↑; horizon-averaged AUROC) across 13 of the 14 benchmarks (C-MAPSS-4, added later, is in table 1). HEPA takes the best mean on 9 of 14 at full labels and matches or exceeds each baseline on at least 10; at 10% labels (open circles) it retains ≥87% of full-label performance on C-MAPSS (≥96% on the single-condition subsets). (b) Predicted probability surfaces $p(t, \Delta t)$ for turbofan degradation (top) and cardiac arrhythmia (bottom).

internalise dynamics at multiple timescales (fig. 2). After self-supervised pretraining, the standard JEPA recipe discards the predictor and trains a linear probe on the frozen encoder. We instead retain the predictor: freeze the encoder but finetune the predictor alongside a lightweight event head that outputs a discrete-time survival CDF, ensuring that the predicted event probability never decreases as the horizon grows. This "predictor finetuning" recipe tunes only 198K parameters, roughly $11\times$ fewer than end-to-end training, yet is more expressive than a linear probe because the predictor reshapes its horizon-conditioned outputs to align with the downstream event.

Our contributions are:

1. **One architecture, any event, any domain.** A single 2.16 M-parameter architecture with fixed hyperparameters, evaluated on 14 benchmarks across 11 domains via a unified probability surface $p(t, \Delta t)$. HEPA matches or exceeds each baseline on at least 10 of 14 benchmarks and takes the best mean on 9, while tuning $11\times$ fewer *label-tuned* parameters than PatchTST (198K vs. 2.26M); in downstream head capacity the baselines in fact get more (264K vs. 198K).
2. **Predictor finetuning as the downstream recipe.** Freezing the encoder and finetuning only the predictor and event head tunes $11\times$ fewer parameters than end-to-end training. On the C-MAPSS benchmark (Saxena et al., 2008), where degradation unfolds over hundreds of cycles, C-MAPSS-1 retains 92% of full-label h-AUROC at just 2% of labels. An information-theoretic bound (proposition 1) provides a condition compatible with this behaviour, and its key prediction, that lower pretraining loss implies stronger downstream performance, is validated *within* individual datasets across pretraining checkpoints (fig. 4); its constants are dataset-specific, so pooling all 14 datasets washes the correlation out ($r{=}-0.05$, $p{=}0.90$).

Two limitations belong with these claims: the surfaces are poorly calibrated out of the box (ECE up to 0.310, section O), and C-MAPSS RUL error is about twice that of supervised point predictors (section J).

2 Related Work

Self-supervised learning for time series. Self-supervised learning (SSL) for time-series representation learning falls into three families. Contrastive methods, including TS2Vec (Yue et al., 2022), TNC (Tonekaboni et al., 2021), TimesURL (Liu and Chen, 2024), CPC (van den Oord et al., 2018), and CoST (Woo et al., 2022), learn representations by contrasting positive and negative pairs.

Masked reconstruction approaches such as PatchTST (Nie et al., 2023) and SimMTM (Dong et al., 2023) recover masked patches in input space; TimesNet (Wu et al., 2023) is a supervised temporal backbone. JEPA (Assran et al., 2023; Bardes et al., 2024) takes a different path: predicting future *representations* rather than reconstructing inputs. For time series, TS-JEPA (Ennadir et al., 2024) applies temporal masking for classification, MTS-JEPA (He et al., 2026) adds codebook regularisation for anomaly prediction, and CF-JEPA (Lee and Sim, 2026) replaces masking with multi-horizon forward prediction of representations. CF-JEPA is the closest prior work and shares the multi-horizon forward-latent-prediction premise; HEPA differs in the collapse mechanism (a variance–covariance penalty rather than asymmetric encoder utilisation) and in finetuning the predictor into a monotone survival CDF. Concurrent; not benchmarked here. All these methods discard their pretraining head at inference and probe only the encoder. HEPA instead retains the predictor and finetunes it toward the downstream event, treating the predictor as a learnable bridge between frozen representations and event probabilities. Collapse is prevented by a variance–covariance penalty (Bardes et al., 2022) rather than by I-JEPA's EMA schedule. LeJEPA (Balestriero and LeCun, 2025) addresses the same failure mode with a distributional constraint; we compare the two directly in section I.5.

Foundation models for time series. Chronos-2 (Ansari et al., 2025), TimesFM-1.0 (Das et al., 2024), MOMENT (Goswami et al., 2024), Moirai (Woo et al., 2024), pretrain on large-scale corpora for generic value forecasting, while UniTS (Gao et al., 2024) is explicitly multi-task, sharing one model across forecasting, classification and anomaly detection. Generative pretraining (Liu et al., 2024a) and LLM repurposing (Zhou et al., 2023) offer alternative transfer strategies. These approaches target future channel values; HEPA targets event probabilities. See also concurrent work on industrial pretraining corpora (Othman et al., 2026; Merzouki et al., 2026). The encoder is mid-scale and pretrained per-dataset; what transfers across domains is the *recipe* (architecture + predictor finetuning), not the weights. We benchmark HEPA against four of these foundation models, using matched-capacity downstream heads to isolate encoder quality (sections 5 and G).

Prognostics, anomaly prediction, and survival modelling. C-MAPSS (Saxena et al., 2008) is the standard remaining-useful-life (RUL) benchmark, where the supervised state of the art is STAR (Fan et al., 2024) (root mean square error, RMSE, 10.61). Self-supervised pretraining is far less established in this literature than supervised regression. Anomaly detection methods such as Anomaly Transformer (Xu et al., 2022), DCdetector (Yang et al., 2023), and TranAD (Tuli et al., 2022) report point-adjusted F1, a metric shown to inflate scores dramatically by crediting entire segments from a single detection (Kim et al., 2022a; Schmidl et al., 2022). These domain-specific metrics are incomparable across tasks. HEPA's downstream parameterisation builds on discrete-time survival models: we compose per-interval hazards into a survival CDF (Gensheimer and Narasimhan, 2019) rather than modelling the event/time probability mass directly (Lee et al., 2018); we adapt this to multi-horizon event prediction. We unify evaluation through h-AUROC, the mean of per-horizon AUROC values computed over the probability surface, which is threshold-free and robust to class imbalance (section 4). Domain-specific metrics are reported as lossy projections of the same surface for comparability with published baselines.

3 METHOD

3.1 ARCHITECTURE AND PRETRAINING

HEPA consists of three components that interact across two phases (fig. 2). The **context encoder** f_θ is a causal Transformer (d=256, 2 layers, 4 heads) that maps observations $\mathbf{x}_{\leq t}$, tokenised into non-overlapping patches of size P=16 (following PatchTST (Nie et al., 2023)) with per-context instance normalisation (Kim et al., 2022b) and sinusoidal positional encodings, to a summary embedding $\mathbf{h}_t = f_\theta(\mathbf{x}_{\leq t}) \in \mathbb{R}^d$. The **predictor** g_ϕ is a multilayer perceptron (MLP) with two hidden layers (three weight matrices, $257 \to 256 \to 256 \to 256$, 197.6K parameters) that takes the encoder output $\mathbf{h}_t$ together with a prediction horizon Δt and produces a predicted embedding of the future interval. The horizon enters as a raw scalar in native time units, concatenated to $\mathbf{h}_t$ (so the input width is d+1); it is neither normalised nor embedded, which is what makes sub-patch horizons well defined:

$$\hat{\mathbf{h}}_{(t,t+\Delta t]} = g_\phi(\mathbf{h}_t, \Delta t). \tag{1}$$

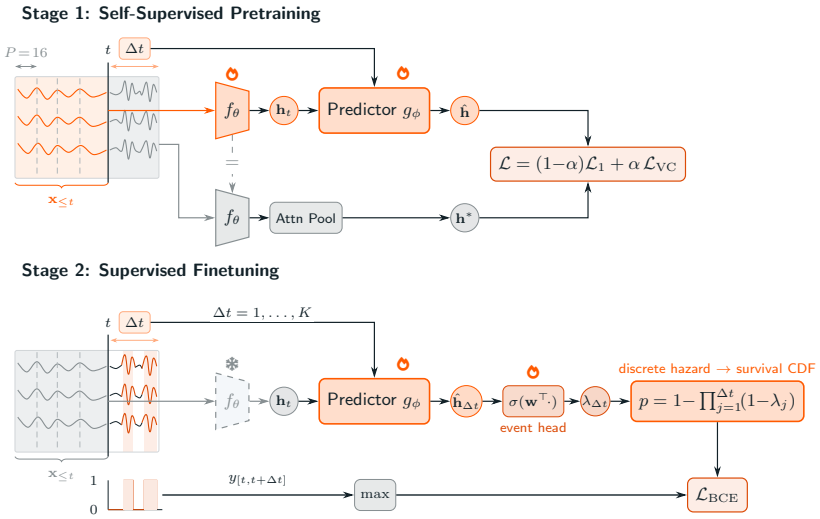

Figure 2: **HEPA architecture.** Both stages sweep over all $(t, \Delta t)$ pairs per episode. *Stage 1:* The causal encoder f_θ maps $\mathbf{x}_{\leq t}$ to $\mathbf{h}_t$; the predictor $g_\phi(\mathbf{h}_t, \Delta t)$ predicts future representations via a self-supervised JEPA objective. *Stage 2:* Encoder frozen; the predictor produces K horizon-specific hazard rates $\lambda_{\Delta t}$ composed into a survival CDF (cumulative distribution function) $p(t, \Delta t)$.

During pretraining, Δt is sampled from a log-uniform distribution over $[1, \Delta t_{\text{max}}]$, encouraging the encoder to internalise dynamics at multiple timescales. A second encoder of the same architecture, applied bidirectionally to $\mathbf{x}_{(t,t+\Delta t]}$ with attention pooling, produces the **target representation** $\mathbf{h}^*_{(t,t+\Delta t]} \in \mathbb{R}^d$. It is initialised by copying the context encoder's shape-compatible weights and is then held fixed: the alignment term is computed against a detached target, so it takes no optimiser step and $\mathbf{h}^*$ is a fixed function of the future window throughout pretraining.

Collapse is prevented by a **variance–covariance regulariser** on the raw predictor output (the two single-branch terms of VICReg (Bardes et al., 2022)), which replaces the exponential moving average (EMA) momentum schedule of standard JEPA (section I.3). The variance hinge holds each coordinate's standard deviation near 1, ruling out the constant solution; the covariance term decorrelates coordinates, ruling out low-rank ones. The closed form is in section I.4. Input asymmetry reinforces it: the predictor sees only $\mathbf{x}_{\leq t}$, never the target window. A single mixing weight α=0.04 controls its contribution to the total loss (section I.3). Sub-patch horizons and context lengths that are not multiples of P are handled without special casing; see section L.

The pretraining loss combines an L1 prediction objective (chosen over L2 because L1 distributes gradient magnitude equally across samples, avoiding domination by outlier predictions) with the variance–covariance regulariser of eq. (21). Alignment is computed on L2-normalised representations, the regulariser on the raw predictor output, where a unit-variance hinge is meaningful rather than saturated:

$$\mathcal{L} = (1-\alpha)\left\|\tilde{\mathbf{h}} - \tilde{\mathbf{h}}^*\right\|_1 + \alpha\,\mathcal{L}_{\text{VC}}, \qquad \tilde{\mathbf{h}} = \hat{\mathbf{h}}/\|\hat{\mathbf{h}}\|_2,\ \tilde{\mathbf{h}}^* = \mathbf{h}^*/\|\mathbf{h}^*\|_2, \tag{2}$$

where α balances the two terms. The alignment term uses a detached target, so the target encoder stays at its initialisation and gradient reaches the encoder only through the context branch; collapse is prevented by eq. (21) rather than by an EMA schedule. No labels are used. Pretraining takes under one minute per dataset on a single A10G GPU, with the full 14-dataset, 5-seed sweep completing in under two hours. Per-dataset preprocessing details are in section L.

3.2 Downstream: Predictor Finetuning

After pretraining, we freeze the encoder f_θ and finetune only the predictor g_ϕ together with a lightweight linear event head. This "predictor finetuning" (pred-FT) recipe tunes 198K parameters,

compared to 2.16M for end-to-end training and 769 for a frozen linear probe (LayerNorm affine, 512, plus the shared linear map, 257). Finetuning reshapes the predictor's per-horizon outputs to separate event-relevant from event-irrelevant dynamics, making it more expressive than a linear probe, while the frozen encoder supplies the pretrained dynamical knowledge that makes few labels sufficient. To be explicit: what transfers is the frozen encoder and the horizon-conditioning *structure*, not the predictor's weights: the predictor is effectively re-learned, and initialising it randomly costs at most 0.003 (section I.2). End-to-end finetuning achieves equivalent h-AUROC at full labels (table 4); pred-FT's advantage is computational efficiency and robustness under label scarcity (section 5.4).

The predictor is run at each of K discrete horizons $\Delta t = 1, \ldots, K$ (unit steps; K=150 for C-MAPSS/TEP, K=200 otherwise). A shared linear head maps each predicted representation to a per-interval *conditional hazard*:

$$\lambda_{\Delta t}(t) \;=\; \sigma\big(\mathbf{w}^\top \hat{\mathbf{h}}_{(t,t+\Delta t]} + b\big) \;\in\; (0,1), \tag{3}$$

where σ is the sigmoid function and $\lambda_{\Delta t}(t)$ approximates $P(\text{event in } (\Delta t{-}1, \Delta t] \mid T^* > \Delta t{-}1, \mathbf{x}_{\le t})$, with T^* denoting the time to the first event after t. The event probability surface is then parameterised as a discrete-time survival CDF (Lee et al., 2018; Gensheimer and Narasimhan, 2019):

$$p(t, \Delta t) \;=\; 1 - \prod_{j=1}^{\Delta t} (1 - \lambda_j(t)). \tag{4}$$

Because each factor $(1 - \lambda_j) \in (0, 1)$, the survival product is non-increasing in Δt, so $p(t, \Delta t)$ increases monotonically with the prediction horizon by construction. No distributional assumptions are required: each $\lambda_{\Delta t}$ is a free function of $\mathbf{h}_t$ via the predictor network. The finetuning loss sums positive-weighted binary cross-entropy (BCE) over horizons:

$$\mathcal{L}_{\text{FT}} = -\frac{1}{K} \sum_{\Delta t=1}^{K} \Big[\, w^+ \, y(t, \Delta t) \log p(t, \Delta t) \;+\; \big(1 - y(t, \Delta t)\big) \log\big(1 - p(t, \Delta t)\big) \Big], \tag{5}$$

where $y(t, \Delta t) = \mathbb{1}[\text{event in } (t, t{+}\Delta t]]$. The weight w^+ multiplies the positive term only (placing it outside the BCE would scale positives and negatives alike, making it a learning-rate multiplier rather than class reweighting) and is a single global scalar rather than a per-horizon $w^+_{\Delta t}$ (section M).[1]

3.3 Theoretical Analysis

Predictor finetuning rests on a premise: the pretrained encoder retains enough event-relevant information that a small downstream head can extract it. We formalise when this holds and connect the bound to experiments.

Let $X_{\le t}$ denote observations up to time t, and let $E_{t+\Delta t} \in \{0, 1\}$ be a binary indicator that equals 1 if an event occurs in the interval $(t, t{+}\Delta t]$ and 0 otherwise. The encoder produces $H_t = f_\theta(X_{\le t}) \in \mathbb{R}^d$; the target encoder produces $H^* = \bar{f}_\theta(X_{(t,t+\Delta t]}) \in \mathbb{R}^d$ from the future interval; and the predictor produces $\hat{H} = g_\phi(H_t, \Delta t)$. We define the event posterior $\eta(h) \coloneqq P(E_{t+\Delta t}{=}1 \mid H^*{=}h)$ and the marginal event rate $\pi_e \coloneqq P(E_{t+\Delta t}{=}1)$, using π_e to distinguish it from the probability surface $p(t, \Delta t)$.

Proposition 1 (Event-Information Retention)**.** *Suppose* (A1) *the event $E_{t+\Delta t}$ is conditionally independent of $X_{\le t}$ given H^*,* (A2) *the pretraining loss satisfies $\mathbb{E}[\|\hat{H} - H^*\|_2^2] \le \varepsilon$,* (A3) *the event posterior $\eta(h)$ is L-Lipschitz, and* (A4) *the posterior is bounded: $\eta(H^*) \in [\underline{\eta}, \overline{\eta}] \subset (0, 1)$ a.s. Then*

$$I(H_t;\, E_{t+\Delta t}) \;\ge\; I(H^*;\, E_{t+\Delta t}) \;-\; C_\eta \, L^2 \, \varepsilon \,, \tag{6}$$

where $C_\eta = (2\,\underline{\eta}\,(1{-}\overline{\eta}))^{-1}$ and $I(\cdot;\cdot)$ denotes mutual information.

[1] We apply BCE to the cumulative event probability $p(t, \Delta t)$ rather than to the per-step hazards $\lambda_j(t)$ against per-step indicators (the standard discrete-survival likelihood, e.g. nnet-survival (Gensheimer and Narasimhan, 2019)). This is a deliberate design choice: BCE on the cumulative surface acts as a smoothing regulariser across horizons (each hazard λ_j contributes to BCE for every $\Delta t \ge j$), which empirically improves h-AUROC under our positive-weighted regime but distorts the probability scale (section O).

The proof proceeds in three steps (full details in section A). First, because $\hat{H}$ is a deterministic function of H_t, the data processing inequality gives $I(H_t; E) \geq I(\hat{H}; E)$. Second, a Jensen-gap argument on the convex KL divergence, combined with the Lipschitz condition and prediction error bound, yields $I(H^*; E) - I(\hat{H}; E) \leq C_\eta L^2 \varepsilon$. Combining these two inequalities produces the result.

The bound motivates a testable expectation: as pretraining proceeds and ε shrinks, downstream h-AUROC should rise. It is an expectation rather than a deduction: proposition 1 lower-bounds $I(H_t; E_{t+\Delta t})$, and a rising lower bound does not entail that the quantity itself rises, nor that mutual information and AUROC move together. The bound's constants L (Lipschitz of η), C_η (posterior bound), and the target sufficiency $I(H^*; E_{t+\Delta t})$ are all properties of the target representation $\mathbf{h}^*$, so a checkpoint sweep would ordinarily move them alongside ε and confound the test. It does not do so here, for a structural reason: the target branch is detached and never updated (section 3), so $\mathbf{h}^*$ is a fixed function of the future window for the whole of pretraining. Its distribution, the posterior η, and hence L, C_η and $I(H^*; E_{t+\Delta t})$ are therefore constant across the checkpoints of a run, and varying ε alone is exactly what the sweep does. The constants still differ across datasets, so the bound remains testable only *within* a dataset, by varying ε alone. We do this on three contrasting domains, turbofan lifecycle (C-MAPSS-3), cardiac arrhythmia (MBA), and spacecraft telemetry anomalies (SMAP), by snapshotting the encoder during pretraining at epochs $\{1, 3, 8, 25\}$ plus the converged best, and at each snapshot running the standard predictor finetuning recipe to obtain h-AUROC on the held-out test split (3 seeds per dataset). The bound's monotone prediction holds across all three: pooled Spearman $\rho(\varepsilon, \text{h-AUROC}) = -0.67$ (p=0.017, n=12) on C-MAPSS-3, ρ=−0.64 (p=0.026, n=12) on MBA, and ρ=−0.49 (p=0.130, n=11) on SMAP. SMAP shows the largest visible h-AUROC range (0.40 at ε=0.033 rising to 0.65 at ε=0.026). The converged-best snapshot regresses slightly relative to epoch 25 on all three datasets, consistent with mild over-pretraining at fixed labels. The same sweep on C-MAPSS-1, the original lifecycle benchmark, gives a signal of the same sign. Corollary 2 predicts a fourth regime where the bound becomes vacuous: on short-window anomaly benchmarks like GECCO the within-dataset correlation is indistinguishable from zero and finetuning is unstable across early snapshots, as expected when extended precursors are weak.

A cross-dataset scatter, by contrast, does *not* validate the bound: pooling the converged ε across the 14 Table 1 datasets gives Pearson r=−0.05 (p=0.90), because L, C_η, and the absolute scale of the target representation differ by dataset, dominating any signal from ε alone; the same incommensurability fig. 4 makes visible (C-MAPSS-3 clusters around $\varepsilon \sim 0.015$, SMAP around 0.027, MBA around 0.06). This does not contradict the bound; it shows that comparing ε across datasets compares incommensurable quantities, motivating the within-dataset protocol above. Two further caveats remain. The constants L and C_η are not estimated directly; fig. 4 validates only the monotonic relationship, not the full quantitative bound. And A1 (target sufficiency) may fail when event precursors span intervals longer than the target window; when A1 is violated, proposition 1 ceases to apply rather than merely loosening: the guarantee is unavailable, even though $I(H_t; E_{t+\Delta t})$ may remain large in practice (see section A for assumption-by-assumption failure modes).

Corollary 2 (Precursor necessity)**.** *The bound is non-vacuous if and only if the future interval contains event precursors that the target encoder captures ($I(H^*; E_{t+\Delta t}) > 0$) and the predictor approximates the target well enough ($\varepsilon < I(H^*; E_{t+\Delta t})/(C_\eta L^2)$).*

This corollary explains both HEPA's successes and its failures. On C-MAPSS, degradation unfolds over hundreds of cycles, so $I(H^*; E_{t+\Delta t})$ is large and pretraining drives ε small, yielding h-AUROC up to 0.84 on the single-condition subsets (C-MAPSS-1, C-MAPSS-3). On datasets without extended precursors, the bound is vacuous regardless of pretraining quality.

4 Evaluation Framework

The model outputs a probability surface $p(t, \Delta t)$ (eq. (4)) for each observation time t and prediction horizon Δt. This surface is the complete prediction; every metric is computed deterministically from it (fig. 3), enabling direct comparison with published baselines without retraining.

As a cross-domain metric, we use **h-AUROC**: the mean over horizons of the per-horizon AUROC, each computed over t at fixed Δt. This is the integrated cumulative/dynamic time-dependent AUC of the survival literature (Heagerty and Zheng, 2005; Blanche et al., 2013), which we introduce to RUL and anomaly benchmarks otherwise scored on incomparable metrics. Per-horizon prevalence varies

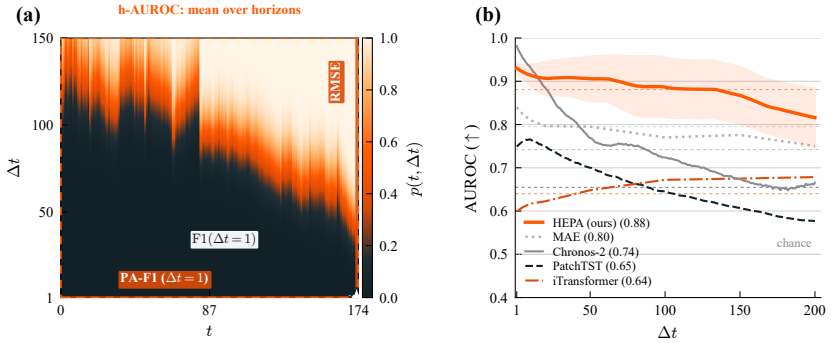

Figure 3: **Evaluation framework.** (a) The probability surface $p(t, \Delta t)$ on a representative C-MAPSS-1 engine (lifetime 174 cycles) unifies all event-prediction metrics as lossy projections. The colour scale matches Fig. 1b. RMSE requires converting the survival curve to a point estimate $\hat{\tau} = \sum_{\Delta t} \Delta t \cdot P(\text{event at } \Delta t)$; this projection is sensitive to calibration (section J). PA-F1 thresholds $p(t, 1)$ at the smallest horizon and credits entire anomaly segments from a single detection (inflated (Kim et al., 2022a)). F1 collapses to a single $(t, \Delta t)$ cell. h-AUROC averages AUROC over all horizons, using the full surface. (b) Per-horizon AUROC on GECCO (K=200 for HEPA / PatchTST / Chronos-2; sparse K=8 for iTransformer / MAE, whose per-horizon curves come from an earlier evaluation batch and are shown for shape only, not for horizon-matched comparison). Mean h-AUROC (↑) per method shown in the legend; dashed lines mark the per-method mean. HEPA holds AUROC $\geq$0.82 across the full horizon range while value-level baselines decay sharply.

wildly across datasets, and even within a single surface: on C-MAPSS-1, the event "failure within Δt steps" has prevalence 0.5% at Δt=1 and 96% at Δt=150, a $\sim$200$\times$ range. Pooled area under the precision-recall curve (AUPRC) over all $(t, \Delta t)$ cells inherits a 0.957 baseline on C-MAPSS-1, because a model predicting only per-horizon prevalence already scores there. h-AUROC solves this by decomposing the surface into independent per-horizon binary classification problems, each with a universal 0.5 baseline that does not depend on prevalence. The uniform average treats all horizons equally, though specific horizons matter more in practice (long-range for turbine maintenance, short-range for arrhythmia); we use it for cross-domain comparability and store the full surface for application-specific weighting. Domain-specific metrics (RMSE for remaining-useful-life, PA-F1 for anomaly detection) are derived as projections of the same surface for comparability with published baselines (section J). All numbers are reported as mean $\pm$ std across 5 seeds (HEPA, PatchTST, iTransformer, MAE) or 3 seeds (Chronos-2).

5 Experiments

5.1 Setup

We pretrain a separate HEPA encoder per dataset from unlabeled training data. Architecture and hyperparameters are identical across all domains; only the input projection (sensor count S) changes. All comparison methods share the loss (eq. (5)), splits, horizon grids and evaluation; they differ in the encoder and in how the head is attached to it (section C). Dense unit-step horizons are used throughout: K=150 for C-MAPSS and TEP, K=200 for all others. The dataset overview (14 datasets, 11 domains) is in table 3.

5.2 Main Results

Table 1 compares HEPA against two classes of methods. The primary comparison is *architectural*: PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024b), and a masked autoencoder (MAE) baseline use the same per-dataset regime with matched-capacity downstream heads, isolating the effect of JEPA pretraining versus alternative self-supervised and supervised objectives. The secondary comparison is against the *foundation model* Chronos-2 (Ansari et al., 2025), which pretrains on a large external corpus and operates in a fundamentally different regime. A full comparison against MTS-JEPA (He et al., 2026) (matched protocol) is in section H; HEPA wins on 8 out of 9 datasets

Table 1: **Main results** (mean ± std; 5 seeds for HEPA, PatchTST, iTransformer, MAE; 3 seeds for Chronos-2). Protocols differ by method and are set out in section C. Two rows per dataset: 100% and 10% labels (gray). **Bold** = best per row.

			FM	Unified architecture				Domain-specific SOTA metric			
Dataset	Domain	Label %	Chr-2	PatchTST	iTransf	MAE	**HEPA**	Metric	**HEPA**	SOTA	Ref
				h-AUROC ↑					domain metric		
C-MAPSS-1	Turbo.	100	.66±.00	.80±.04	.70±.05	.69±.02	**.81**±.03	RMSE↓	28.5	**12.2**	(Fan et al., 2024)
turbine failure		10	.66±.00	.69±.08	.59±.04	.63±.12	**.78**±.07		32.3	**17.0**	
C-MAPSS-2	Turbo.	100	.45±.01	.44±.03	.43±.03	.56±.01	**.57**±.01	RMSE↓	40.8	**20.0**	(Fan et al., 2024)
multi-cond.		10	.46±.02	.48±.08	.50±.08	.52±.01	**.55**±.01		**40.9**	43.1	
C-MAPSS-3	Turbo.	100	.73±.00	.79±.01	.76±.01	.78±.02	**.84**±.01	RMSE↓	34.7	**12.7**	(Fan et al., 2024)
multi-fault		10	.67±.00	.77±.03	.64±.06	.78±.03	**.84**±.01		47.1	**21.8**	
C-MAPSS-4	Turbo.	100	—	.52±.03	.45±.02	.57±.02	**.63**±.02	RMSE↓	—	—	
multi-cond.+fault		10	—	.51±.04	.49±.04	.52±.04	**.55**±.03		—	—	
SMAP	Spacecraft	100	.54±.02	.49±.03	.47±.05	**.64**±.04	.59±.05	PA-F1↑	.86	**.96**	(Xu et al., 2022)
sensor anomaly		10	.51±.01	.46±.05	**.52**±.07	.50±.15	.49±.08		.89	**.96**	
PSM	Server	100	.48±.01	.55±.02	.52±.03	.56±.02	**.57**±.02	PA-F1↑	.95	**.98**	(Xu et al., 2022)
server anomaly		10	.51±.01	.43±.03	**.53**±.01	.52±.02	.48±.04		.95	**.98**	
MBA	Cardiac	100	.53±.10	.68±.07	**.84**±.03	.73±.03	.75±.03	F1↑	**.98**	.77	LSTM-AE
arrhythmia		10	.50±.07	**.65**±.08	.31±.05	.53±.04	.55±.15		**.98**	.81	
BATADAL	ICS	100	.56±.01	**.66**±.03	.46±.13	.60±.04	.57±.03	F1↑	**.77**	.74	AEED
cyberattack		10	.55±.04	.37±.04	.54±.08	**.61**±.08	.54±.06		**.61**	.33	
TEP	Chemical	100	—	.99±.02	.93±.02	.96±.02	**1.00**±.00	F1↑	**.95**	.93	XGBoost
process fault		10	—	**1.00**±.00	.94±.05	.99±.00	**1.00**±.00		.88±.03	.86	
ETTm1	Power	100	.74±.01	.53±.03	.79±.03	**.87**±.00	.81±.00				
overheating		10	.68±.02	.50±.03	.61±.02	**.77**±.02	.73±.01				
Weather	Climate	100	.72±.02	.64±.08	.83±.04	.88±.02	**.89**±.01				
heat spike		10	.71±.02	.67±.03	**.85**±.01	.83±.02	.83±.02				
Beijing-AQ	Air	100	.53±.00	**.81**±.01†‡	.75±.02	.81±.01	.81±.01				
PM2.5 spike		10	.49±.00	.65±.08	.73±.02	.77±.01	**.78**±.02				
VIX	Finance	100	.40±.04	.48±.11	.40±.14	.57±.03	**.57**±.01				
vol regime		10	.38±.04	**.57**±.13	.48±.05	.55±.03	.55±.01				
GECCO	Water	100	.74±.02	.65±.07	.64±.15	.81±.04	**.88**±.06				
contamination		10	**.80**±.03	.52±.11	.50±.13	.55±.11	.39±.13				
Best (100%)			0	2	1	2	9				

Matched downstream heads (HEPA 198K pred-FT; baselines 264K dt-MLP), positive-weighted BCE, identical protocol. K=150 for C-MAPSS/TEP, K=200 otherwise. Domain SOTAs are detection or RUL baselines; HEPA domain metrics projected from $p(t, \Delta t)$ at the matching horizon (Δt=1 for PA-F1/F1, $\mathbb{E}[\Delta t]$ for RMSE; section J). PA-F1 = point-adjusted F1 (Kim et al., 2022a). Pairwise Welch's t-tests in section N. †Beijing-AQ PatchTST: 3 stations at 100%. ‡Beijing-AQ's top three agree to two decimals but not in rank: PatchTST .814 leads HEPA .807. Bold follows the means, so this row is not a HEPA win. HEPA and MAE are within .003 and the two stored result sets order them differently, so we do not rank them; neither gap is statistically significant (section N). Bottom block has no published domain SOTA. Additional baselines (MOMENT, TimesFM-1.0, Moirai, MTS-JEPA) in sections G and H.

where MTS-JEPA could be reproduced (TEP excluded: the public MTS-JEPA release does not include a chemical-process benchmark).

HEPA vs. architectural baselines. At 100% labels HEPA matches or exceeds each baseline on at least 10 of 14 benchmarks, and takes the best mean on 9 (fig. 1a), including all four C-MAPSS variants and the newly added FD004 (the hardest subset: two fault modes and six operating conditions). HEPA's representation-level prediction captures temporal structure that supervised training (PatchTST) and reconstruction-based SSL (MAE) miss, particularly on datasets with extended precursor dynamics (C-MAPSS, GECCO, PSM, TEP). MAE is a strong second: it matches or exceeds HEPA on spacecraft telemetry (SMAP) and power systems (ETTm1), suggesting that reconstruction-based pretraining transfers well when the dominant failure mode is gradual drift. iTransformer's variate-attention mechanism excels on MBA (h-AUROC 0.84 vs. HEPA's 0.75), where arrhythmia patterns are localised across specific leads.

HEPA vs. Chronos-2. At 100% labels HEPA leads Chronos-2 in mean h-AUROC on all twelve benchmarks Chronos-2 covers; at 10% labels Chronos-2 leads on GECCO. Per-dataset JEPA excels when events have extended precursors that the local training data fully represents; large-corpus pretraining helps when event signatures resemble patterns seen at scale.

Where HEPA loses. HEPA is below the best baseline on five datasets at 100% labels. The pattern is interpretable: BATADAL and MBA have sensor-localised events where channel-fusion tokenisation dilutes the relevant subset, so per-variate attention (iTransformer) or channel-independent training (PatchTST) wins; Beijing-AQ's top three agree to two decimals; MAE's reconstruction objective transfers well when the dominant failure mode is gradual drift (SMAP, ETTm1). Adopting a sensor-as-token strategy (Liu et al., 2024b) within the HEPA encoder is a natural way to close this gap.

5.3 What Does Pretraining Learn?

Without any labels, the encoder organises C-MAPSS-1 representations into a smooth degradation manifold (fig. 4, four test engines). Across all 100 test engines, PC1 captures 61% of variance and tracks time-to-failure monotonically within each engine (median per-engine Spearman $\rho{=}+0.97$; 84% of engines $\rho{>}0.9$), with engines from different healthy regions converging on a shared failure region. Healthy and degraded states are thus already separated before any label is seen, which is why so few suffice.

5.4 Label Efficiency

Table 2: **Label efficiency on C-MAPSS lifecycle datasets (HEPA only, 3 seeds).** h-AUROC (↑) and retention as raw quotient / above chance, the latter $(x{-}0.5)/(\text{full}{-}0.5)$. The two diverge at the smallest budgets: C-MAPSS-3 at 1% is at chance, which raw retention calls 60%.

	C-MAPSS-1 (85 eng.)			C-MAPSS-3 (100 eng.)		
Labels	h-AUROC	Ret. / >chance	Eng.	h-AUROC	Ret. / >chance	Eng.
100%	.786 ± .033	100% / 100%	85	.853 ± .004	100% / 100%	100
10%	.772 ± .059	98% / 95%	9	.830 ± .018	97% / 93%	10
5%	.730 ± .018	93% / 80%	4	.709 ± .131	83% / 59%	5
2%	**.724 ± .013**	**92% / 78%**	**2**	.635 ± .065	74% / 38%	2
1%	.670 ± .110	85% / 59%	1	.513 ± .220	60% / 4%	1

All methods in table 1 train only a downstream head on a frozen encoder at evaluation time, though PatchTST and iTransformer obtained that encoder by end-to-end supervised training rather than by pretraining (section C). The question is whether HEPA's representations degrade more gracefully. Table 2 shows that on C-MAPSS, where degradation unfolds over hundreds of cycles and the JEPA predictor achieves low pretraining loss, HEPA retains 92% of full-label h-AUROC with just 2 training engines out of 85. C-MAPSS-3 retains 97% at 10% labels. This is consistent with proposition 1: low ε on lifecycle datasets means the encoder already separates healthy from degraded states, so the finetuned predictor needs only a few labelled examples to map them to event probabilities.

The advantage is not universal. On anomaly datasets without extended precursors, HEPA at 10% labels does not compress toward chance but falls below it (table 1, gray rows). An h-AUROC under 0.5 is a ranking *inversion*: with the encoder frozen and few positive cells left, positive-weighted BCE can fit a head ordered against the event, so where precursors are weak the frozen-representation advantage reverses rather than merely vanishing. Label efficiency is strongest where pretraining loss is lowest: extended-precursor lifecycle datasets.

6 Conclusion & Future Work

HEPA demonstrates that self-supervised JEPA pretraining combined with predictor finetuning provides a practical recipe for event prediction. The encoder learns temporal dynamics from unlabelled data; the predictor learns which dynamics signal the target event. One architecture handles degradation forecasting, anomaly prediction, and arrhythmia detection across 14 benchmarks in 11 domains, matching or exceeding PatchTST, iTransformer, MAE, and Chronos-2 on at least 10 of 14 benchmarks in pairwise comparison and taking the best mean on 9, while tuning an order of magnitude fewer label-tuned parameters. On lifecycle datasets the recipe is robust to extreme label scarcity: 92% of full-label h-AUROC with 2% of labels on C-MAPSS-1, consistent with the information-retention bound of proposition 1. Because the recipe is domain-agnostic, the same architecture that predicts turbine failure from flight-recorder data can flag arrhythmia risk from ECG streams or detect water contamination from sensor networks, each time requiring only a handful of event labels.

Looking ahead, cross-domain pretraining on corpora such as FactoryNet (Othman et al., 2026) is the natural next step toward industrial deployment, and sensor-as-token strategies (Liu et al., 2024b) could close the gap on systems where event-relevant information is concentrated in a few channels. On the theory side, deriving fully empirical versions of the information-retention bound, estimating L and C_η directly from data, remains an open problem. Wherever multivariate sensors record the precursors to rare but consequential events, HEPA offers a path from unlabelled streams to actionable predictions.

AI USE STATEMENT

We used generative AI tools (Anthropic's Claude) as an assistant in preparing this work: to implement methods and write supporting evaluation and analysis code; to give feedback on our methodology and experiment design; to clean and reformat the benchmark datasets into a common schema; to help interpret experimental results; to generate figure source from data and specifications we supplied; to draft and edit parts of this manuscript; and to search and summarise related literature.

We did not use them to generate synthetic data, all fourteen benchmarks being publicly available corpora, nor to develop the theoretical framework, nor to formulate or prove the mathematical claims. We did not use them for translation. Qualitative and thematic data analysis does not arise here: every analysis in this paper is quantitative.

The research reported here is the authors' own: the architecture, the experimental design, the theory and the hypotheses this paper tests. The propositions, their assumptions, the proof strategy and the written proofs are ours. All AI-assisted code was reviewed and tested by us, and every number in this paper was produced by running that code: no experimental result, table entry or figure value was authored by a language model. We have reviewed all AI-assisted work and take responsibility for the final content of this paper, including its text, claims and artifacts.

ETHICS STATEMENT

This work uses only publicly available benchmark datasets. No new data were collected and no human-subject experimentation was performed. The cardiac benchmark (MBA) consists of de-identified ECG recordings released for research use; we used it under its published terms, made no attempt at re-identification, and report only aggregate performance. The remaining benchmarks are engineering, environmental, and financial time series.

Three benchmarks (BATADAL, GECCO, SMAP) concern attacks or faults in critical infrastructure. Our intended use is defensive: earlier warning of water contamination, intrusion into a control system, or component failure. All three are long-standing public benchmarks and the architecture is domain-agnostic, so we do not believe this work confers new offensive capability.

We caution against deploying the model as a sole decision-maker in safety-critical settings. As noted in section 3.2 and section O, training positive-weighted binary cross-entropy on the cumulative probability surface improves horizon-averaged ranking but distorts the absolute probability scale, so outputs should be read as a relative ordering of risk rather than as calibrated event probabilities. Clinical or maintenance deployment would require recalibration on the target distribution and human oversight, particularly because false negatives on rare events carry an asymmetric cost that a threshold-free metric such as h-AUROC does not represent.

REPRODUCIBILITY STATEMENT

Architecture and optimiser hyperparameters are fixed across every benchmark and listed in full in section M; only the input projection changes with sensor count. Per-dataset preprocessing, including normalisation, windowing and event labelling, is specified in section L, and the fourteen benchmarks with their sources and splits are described in section D. The matched-capacity protocol applied to every baseline, including the shared downstream head and identical splits, is given in section C, and the horizon sets in section B. Main-table numbers are means over five seeds (three for Chronos-2), with pairwise Welch's t-tests in section N. The assumptions, full proofs and assumption-by-assumption failure modes for proposition 1 and Corollary 2 are given in section A. Pretraining takes under one minute per dataset on a single A10G GPU, and the full fourteen-dataset, five-seed sweep completes in under two hours, so the entire study is reproducible on a single commodity GPU.

REFERENCES

Marten Scheffer, Jordi Bascompte, William A Brock, Victor Brovkin, Stephen R Carpenter, Vasilis Dakos, Hermann Held, Egbert H Van Nes, Max Rietkerk, and George Sugihara. Early-warning signals for critical transitions. *Nature*, 461(7260):53–59, 2009.

Zhengyang Fan, Wanru Li, and Kuo-Chu Chang. A two-stage attention-based hierarchical transformer for turbofan engine remaining useful life prediction. *Sensors*, 24(3):824, 2024. doi: 10.3390/s24030824.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *ICLR*, 2022.

Yanan He, Yunshi Wen, Xin Wang, and Tengfei Ma. MTS-JEPA: Multi-resolution joint-embedding predictive architecture for time-series anomaly prediction. *arXiv preprint arXiv:2602.04643v1*, 2026.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*, 2023.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *ICML*, 2024.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, 2023.

Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *International Conference on Prognostics and Health Management (PHM)*, 2008.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. TS2Vec: Towards universal representation of time series. In *AAAI*, 2022.

Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. In *ICLR*, 2021.

Jiexi Liu and Songcan Chen. TimesURL: Self-supervised contrastive learning for universal time series representation learning. In *AAAI*, 2024.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *ICLR*, 2022.

Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. SimMTM: A simple pre-training framework for masked time-series modeling. In *NeurIPS*, 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 *International Conference on Learning Representations (ICLR)*, 2023.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *TMLR*, 2024.

Sofiane Ennadir, Siavash Golkar, and Leopoldo Sarra. Joint embeddings go temporal. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.

Jaehoon Lee and Sunghyun Sim. CF-JEPA: Mask-free forward prediction with asymmetric encoder utilization for time-series representation learning. *arXiv preprint arXiv:2606.07031*, 2026.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *ICML*, 2024.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *International Conference on Machine Learning (ICML)*, 2024.

Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. UniTS: A unified multi-task time series model. In *NeurIPS*, 2024.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *ICML*, 2024a.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained LM. In *NeurIPS*, 2023.

Karim Othman, Jonas Petersen, Matei Ignuta-Ciuncanu, Camilla Mazzoleni, Federico Martelli, Alessandro Lombardi, Riccardo Maggioni, and Philipp Petersen. Factorynet: A large-scale dataset toward industrial time-series foundation models. *arXiv preprint arXiv:2605.09081*, 2026.

Yanis Merzouki, Coral Izquierdo, Matei Ignuta-Ciuncanu, Marcos Gomez-Bracamonte, Riccardo Maggioni, Alessandro Lombardi, Camilla Mazzoleni, Federico Martelli, Balazs Gunther, Jonas Petersen, and Philipp Petersen. Factorybench: Evaluating industrial machine understanding. *arXiv preprint arXiv:2605.07675*, 2026.

Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. DCdetector: Dual attention contrastive representation learning for time series anomaly detection. In *KDD*, 2023.

Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. TranAD: Deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment*, 15(6): 1201–1214, 2022.

Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. In *AAAI*, 2022a. arXiv:2109.05257.

Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: A comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797, 2022.

Michael F Gensheimer and Balasubramanian Narasimhan. A scalable discrete-time survival model for neural networks. *PeerJ*, 7:e6257, 2019. doi: 10.7717/peerj.6257.

Changhee Lee, William R Zame, Jinsung Yoon, and Mihaela van der Schaar. DeepHit: A deep learning approach to survival analysis with competing risks. In *AAAI*, 2018.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *ICLR*, 2022b.

Patrick J Heagerty and Yingye Zheng. Survival model predictive accuracy and ROC curves. *Biometrics*, 61(1):92–105, 2005.

Paul Blanche, Jean-François Dartigues, and Hélène Jacqmin-Gadda. Estimating and comparing time-dependent areas under receiver operating characteristic curves for censored event times with competing risks. *Statistics in Medicine*, 32(30):5381–5397, 2013.

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 (ICLR)*, 2024b.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

Yury Polyanskiy and Yihong Wu. *Information Theory: From Coding to Learning*. Cambridge University Press, 2025.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

A Theoretical Analysis: Full Proofs and Discussion

A.1 Notation and Preliminaries

We work with the following random variables on a common probability space: $X_{\leq t}$ (observations up to time t), $X_{(t,t+\Delta t]}$ (future observations), $E_{t+\Delta t} \in \{0,1\}$ (event indicator), $H_t = f_\theta(X_{\leq t}) \in \mathbb{R}^d$ (encoder output), $H^* = \bar{f}_\theta(X_{(t,t+\Delta t]}) \in \mathbb{R}^d$ (target encoder output), $\hat{H} = g_\phi(H_t, \Delta t) \in \mathbb{R}^d$ (predicted representation). All mutual informations $I(\cdot;\cdot)$ and entropies $\mathbb{H}(\cdot)$ are well-defined: $E_{t+\Delta t}$ is discrete (binary), and for the continuous variables $H_t, H^*, \hat{H}$ we use differential entropy and the standard extension of mutual information to mixed discrete-continuous pairs (Cover and Thomas, 2006, Ch. 8–9). We use $\mathbb{H}$ (blackboard bold) for entropy to avoid confusion with the encoder embedding H_t.

We define the event posterior $\eta(h) \coloneqq P(E_{t+\Delta t} = 1 \mid H^* = h)$ and the marginal event rate $\pi_e \coloneqq P(E_{t+\Delta t} = 1)$, using π_e to distinguish it from the probability surface $p(t, \Delta t)$ in the main text.

A.2 Assumptions

(A1) Target sufficiency. $E_{t+\Delta t} \perp\!\!\!\perp X_{\leq t} \mid H^*$.

Interpretation. The target encoder's representation of the future interval is a sufficient statistic for the event, given the past. This holds when: (a) the event is determined by the dynamics in $(t, t+\Delta t]$, and (b) the target encoder has enough capacity and sees the relevant future interval. Because $\bar{f}_\theta$ is bidirectional with attention pooling over the full interval, it is strictly more expressive than the causal encoder for summarising the future, making this assumption mild for well-trained target encoders.

When it fails. If the event depends on context outside $(t, t+\Delta t]$ (for instance, a slow trend visible only in $X_{\leq t}$ that the target encoder cannot see), then A1 is violated and the past observations carry event information not mediated by H^*. A1 is load-bearing rather than cosmetic: Step 2 uses it to write $P(E{=}1 \mid H^*, \hat{H}) = \eta(H^*)$ and then applies the tower property (eq. (11)). Without A1 that step does not go through and proposition 1 ceases to apply: the conclusion is not a valid-but-loose inequality, it is simply unavailable. Empirically $I(H_t; E_{t+\Delta t})$ may still exceed the value the bound would have returned, but that is an observation about a particular dataset, not a guarantee the proposition provides.

(A2) Bounded prediction error. $\mathbb{E}[\|\hat{H} - H^*\|_2^2] \leq \varepsilon$.

Interpretation. The pretraining loss (L1 on L2-normalised representations in practice) drives the prediction residual small. The bound uses L2 squared error for analytical tractability. The pointwise inequality $\|u\|_2 \leq \|u\|_1$ does not by itself order two expectations taken at different points in training, so it cannot on its own license reading the L1 curve as an L2 curve. The chain that does hold here uses the normalisation: both representations are projected onto the unit sphere, so $\|u\|_2 \leq 2$, and on that bounded range $\|u\|_2^2 \leq 2\|u\|_2 \leq 2\|u\|_1$ pointwise; taking expectations gives $\varepsilon = \mathbb{E}\|u\|_2^2 \leq 2\,\mathbb{E}\|u\|_1$, so the L1 training loss upper-bounds ε up to a constant and driving it down drives the bound's slack down. This is a one-sided control, not a monotone equivalence: an L1 decrease cannot certify an ε decrease, only a lower ceiling on it. We use this monotonic relationship in our empirical validation (fig. 4) to test the bound's qualitative prediction without requiring a precise norm conversion.

When it fails. Early in training or on out-of-distribution horizons, ε can be large and the bound becomes vacuous.

(A3) Smooth event dependence. The conditional distribution $P(E_{t+\Delta t} = 1 \mid H^* = h)$ is Lipschitz continuous in h with constant L: for all $h, h' \in \mathbb{R}^d$, $|P(E_{t+\Delta t} = 1 \mid H^* = h) - P(E_{t+\Delta t} = 1 \mid H^* = h')| \leq L\|h - h'\|_2$.

Interpretation. Small perturbations of the target representation do not drastically change the event probability. This is a regularity condition on the relationship between the learned representation space and event occurrence; it holds whenever the event boundary in representation space is not a fractal or highly irregular set.

When it fails. If the event probability is a discontinuous function of H^* (e.g. a hard threshold on a single component), the Lipschitz constant L diverges and our continuity argument requires replacement by a discrete analysis.

(A4) Bounded event posterior. There exist $0 < \underline{\eta} \leq \overline{\eta} < 1$ such that $P(\eta(H^*) \in [\underline{\eta}, \overline{\eta}]) = 1$, where $\eta(h) = P(E_{t+\Delta t} = 1 \mid H^* = h)$.

Interpretation. The event posterior, evaluated on the support of the target encoder's output, is bounded away from 0 and 1 almost surely. This matches the main-text assumption A4, and it is a genuine additional assumption rather than a consequence of the others. Since H^* is L2-normalised onto the unit sphere, its support is compact and A3 makes η continuous there, so the range of $\eta(H^*)$ is a closed bounded interval; but neither compactness nor detectability of the event forces that interval away from the endpoints 0 and 1, so we assume it. Note that A4 implies the marginal event rate $\pi_e = \mathbb{E}[\eta(H^*)]$ is bounded in $[\underline{\eta}, \overline{\eta}]$, so the event is neither impossible nor certain.

Role in the proof. This assumption is needed to bound $\sup_q \varphi''(q) = \sup_q 1/(q(1-q))$ over the support of $\eta(H^*)$ (Step 2 of the proof). Without it, the KL second derivative φ'' is unbounded near 0 and 1, making the Jensen-gap bound vacuous. The constant in the bound becomes $C_\eta = (2\underline{\eta}(1-\overline{\eta}))^{-1}$. As $\pi_e \to 0$ or $\pi_e \to 1$, the posterior bounds are forced toward 0 or 1, driving $C_\eta \to \infty$; this reflects a genuine difficulty: distinguishing $P(E|\hat{H})$ from the prior requires high precision when events are extremely rare.

When it fails. If $\eta(H^*)$ concentrates near 0 or 1, then $C_\eta \to \infty$ and the bound degrades. Empirically, this is the case on CHB-MIT, a scalp-EEG seizure benchmark we ran as a deliberate negative control, outside the formal suite (section A.6), where seizure onset cannot be reliably predicted from any 16-second past context, and $\eta(h) \approx \pi_e$ for all h (consistent with $I(H^*; E) \approx 0$).

A.3 Full Proof of Proposition 1

Proof. We proceed in three steps.

Step 1: Data processing inequality. For fixed Δt (which we condition on throughout), $\hat{H} = g_\phi(H_t, \Delta t)$ is a deterministic function of H_t. Since a deterministic function cannot introduce new information, $E_{t+\Delta t} \perp\!\!\!\perp \hat{H} \mid H_t$, so the triple $(E_{t+\Delta t}, H_t, \hat{H})$ satisfies the Markov chain $E_{t+\Delta t} \to H_t \to \hat{H}$, and the data processing inequality (Cover and Thomas, 2006, Thm. 2.8.1) gives

$$I(H_t;\, E_{t+\Delta t}) \;\geq\; I(\hat{H};\, E_{t+\Delta t}). \tag{7}$$

This step uses only the functional relationship between H_t and $\hat{H}$; no assumptions on the data-generating process are needed.

Step 2: Jensen gap bound on mutual information loss. We bound $I(H^*; E_{t+\Delta t}) - I(\hat{H}; E_{t+\Delta t})$.

Expressing MI as expected KL divergence. For any representation R jointly distributed with binary $E = E_{t+\Delta t}$:

$$I(R;\, E) = \mathbb{E}_R\big[D_{\mathrm{KL}}\big(\mathrm{Ber}(\eta_R(R)) \,\big\|\, \mathrm{Ber}(\pi_e)\big)\big], \tag{8}$$

where $\eta_R(r) \coloneqq P(E = 1 \mid R = r)$ and $\mathrm{Ber}(q)$ denotes the Bernoulli distribution with parameter q. Equation (8) is the standard expression of mutual information as the expected KL divergence between the conditional and marginal label distributions; see Cover and Thomas (2006, Eq. 2.30) or

Polyanskiy and Wu (2025, Ch. 3). Applying (8) to $R = H^*$ and $R = \hat{H}$:

$$I(H^*; E) = \mathbb{E}_{H^*}\big[D_{\mathrm{KL}}\big(\mathrm{Ber}(\eta(H^*)) \,\big\|\, \mathrm{Ber}(\pi_e)\big)\big], \tag{9}$$

$$I(\hat{H}; E) = \mathbb{E}_{\hat{H}}\big[D_{\mathrm{KL}}\big(\mathrm{Ber}(\eta_{\hat{H}}(\hat{H})) \,\big\|\, \mathrm{Ber}(\pi_e)\big)\big], \tag{10}$$

where $\eta_{\hat{H}}(\hat{h}) \coloneqq P(E = 1 \mid \hat{H} = \hat{h})$.

Relating $\eta_{\hat{H}}$ to η via A1. Under **(A1)**, $E \perp\!\!\!\perp X_{\le t} \mid H^*$. Since $\hat{H} = g_\phi(f_\theta(X_{\le t}), \Delta t)$ is a composition of measurable functions, it is $\sigma(X_{\le t})$-measurable, and therefore $E \perp\!\!\!\perp \hat{H} \mid H^*$. It follows that $P(E{=}1 \mid H^*, \hat{H}) = P(E{=}1 \mid H^*) = \eta(H^*)$. By the tower property of conditional expectation, for any value $\hat{h}$:

$$\eta_{\hat{H}}(\hat{h}) = P(E = 1 \mid \hat{H} = \hat{h}) = \mathbb{E}\big[\eta(H^*) \mid \hat{H} = \hat{h}\big]. \tag{11}$$

Applying Jensen's inequality. The function $q \mapsto D_{\mathrm{KL}}(\mathrm{Ber}(q) \,\|\, \mathrm{Ber}(\pi_e))$ is convex on $(0, 1)$ (its second derivative is $1/(q(1-q)) > 0$). *Bounding the Jensen gap.* For a twice-differentiable convex φ, a second-order Taylor expansion about $\mathbb{E}[Y]$ with Lagrange remainder gives $\varphi(y) = \varphi(\mathbb{E}[Y]) + \varphi'(\mathbb{E}[Y])(y - \mathbb{E}[Y]) + \frac{1}{2}\varphi''(\xi_y)(y - \mathbb{E}[Y])^2$ for some ξ_y between y and $\mathbb{E}[Y]$; taking expectations kills the linear term and bounding $\varphi''(\xi_y)$ by its supremum yields the Jensen gap bound

$$\mathbb{E}[\varphi(Y)] - \varphi(\mathbb{E}[Y]) \;\le\; \tfrac{1}{2} \sup_y \varphi''(y)\, \mathrm{Var}(Y). \tag{12}$$

Here $\varphi(q) = D_{\mathrm{KL}}(\mathrm{Ber}(q) \,\|\, \mathrm{Ber}(\pi_e)) = q \ln(q/\pi_e) + (1{-}q) \ln((1{-}q)/(1{-}\pi_e))$ and $Y = \eta(H^*)$ conditioned on $\hat{H}$. The second derivative is $\varphi''(q) = 1/(q(1{-}q))$. Under **(A4)**, $\eta(H^*) \in [\underline{\eta}, \overline{\eta}]$ almost surely. The supremum in (12) is over the support of the conditional distribution $Y \mid \hat{H} = \hat{h}$, which is a subset of $[\underline{\eta}, \overline{\eta}]$; we relax it to the marginal support, giving $\sup_q \varphi''(q) \le 1/(\underline{\eta}(1{-}\overline{\eta}))$. Applying (12) conditionally on $\hat{H} = \hat{h}$:

$$\mathbb{E}\big[D_{\mathrm{KL}}\big(\mathrm{Ber}(\eta(H^*)) \,\big\|\, \mathrm{Ber}(\pi_e)\big) \,\big|\, \hat{H} = \hat{h}\big] - D_{\mathrm{KL}}\big(\mathrm{Ber}(\eta_{\hat{H}}(\hat{h})) \,\big\|\, \mathrm{Ber}(\pi_e)\big) \;\le\; \frac{\mathrm{Var}(\eta(H^*) \mid \hat{H} = \hat{h})}{2\,\underline{\eta}\,(1 - \overline{\eta})}, \tag{13}$$

where we have used $q(1-q) \ge \underline{\eta}(1-\overline{\eta})$ for $q \in [\underline{\eta}, \overline{\eta}]$ (from A4). Note that $q(1-q)$ is concave with minimum at the endpoints of $[\underline{\eta}, \overline{\eta}]$; since $\underline{\eta}(1-\overline{\eta}) \le \min(\underline{\eta}(1-\underline{\eta}), \overline{\eta}(1-\overline{\eta}))$, the bound on φ'' is valid (though not tight when the interval is asymmetric around $1/2$).

Bounding the conditional variance via A2 and A3. By **(A3)**: $|\eta(H^*) - \eta(\hat{H})| \le L\|H^* - \hat{H}\|_2$ (where we evaluate η at $\hat{H}$, using that η is defined on all of $\mathbb{R}^d$). The random variable $\eta(H^*)$ is bounded in $[\underline{\eta}, \overline{\eta}] \subset (0, 1)$ by **(A4)**, so all conditional second moments below are finite. (Note that $\eta(\hat{H})$ need not lie in $[\underline{\eta}, \overline{\eta}]$ since $\hat{H}$ may fall outside the support of H^*, but this does not affect the bound: the variance inequality below requires only that the right-hand side is finite, which follows from the Lipschitz condition and bounded L2 error.) For any square-integrable random variable Y, $\mathrm{Var}(Y \mid Z) \le \mathbb{E}[(Y - c)^2 \mid Z]$ for any Z-measurable c; taking $c = \eta(\hat{H})$:

$$\mathrm{Var}\big(\eta(H^*) \mid \hat{H}\big) \;\le\; \mathbb{E}\big[(\eta(H^*) - \eta(\hat{H}))^2 \mid \hat{H}\big] \;\le\; L^2\, \mathbb{E}\big[\|H^* - \hat{H}\|_2^2 \mid \hat{H}\big]. \tag{14}$$

Taking expectations over $\hat{H}$ in (13) and substituting (14):

$$\begin{aligned} I(H^*; E) - I(\hat{H}; E) &\le \frac{L^2}{2\,\underline{\eta}\,(1 - \overline{\eta})}\, \mathbb{E}\big[\|H^* - \hat{H}\|_2^2\big] \\ &\le \frac{L^2\, \varepsilon}{2\,\underline{\eta}\,(1 - \overline{\eta})} \;=\; C_\eta\, L^2\, \varepsilon, \end{aligned} \tag{15}$$

where the last line uses **(A2)** and defines $C_\eta \coloneqq (2\,\underline{\eta}\,(1 - \overline{\eta}))^{-1}$ using the posterior bounds from **(A4)**. The constant C_η depends on the posterior bounds rather than the marginal event rate, making the bound valid without assumptions on the concentration of $\eta(H^*)$ around π_e.

Step 3: Assembling the bound. Combining (7) and (15):

$$I(H_t;\, E_{t+\Delta t}) \;\ge\; I(\hat{H};\, E_{t+\Delta t}) \;\ge\; I(H^*;\, E_{t+\Delta t}) \;-\; C_\eta\, L^2\, \varepsilon. \tag{16}$$

□

Remark 1 (Comparison with $\sqrt{\varepsilon}$ bounds)**.** *The bound is* linear *in ε, which for small ε (the regime of interest, i.e., well-pretrained models) is tighter than bounds obtained via Pinsker's inequality (which yield $\sqrt{\varepsilon}$ dependence). For large ε, Pinsker-based bounds may be tighter; the constants also differ, so the comparison is regime-dependent. This improvement comes from exploiting the Jensen-gap structure: the KL divergence's convexity provides a direct second-order bound on the information loss, rather than going through total variation as an intermediate. The price is the constant C_η, which diverges as the posterior bounds $\underline{\eta}$ or $\overline{\eta}$ approach 0 or 1, reflecting the genuine difficulty of detecting events when the posterior concentrates near certainty or impossibility.*

Remark 2 (Role of the Lipschitz constant)**.** *The bound involves the Lipschitz constant L of the event posterior with respect to the target representation. In practice, this is controlled by the smoothness of the learned representation space: well-regularised encoders (the variance–covariance regulariser, L2 normalisation) produce representations where L is moderate.*

A.4 Tightness Analysis

When is the bound tight? The bound is approximately tight when: (a) the conditional variance $\mathrm{Var}(\eta(H^*) \mid \hat{H})$ is close to $L^2\mathbb{E}[\|H^* - \hat{H}\|_2^2 \mid \hat{H}]$ (the Lipschitz bound on variance is tight, which occurs when the prediction error aligns with the direction of steepest change in η), and (b) the Jensen gap bound (12) is close to equality (which occurs when $\eta(H^*)$ is approximately symmetrically distributed around its conditional mean given $\hat{H}$). In the high-SNR regime ($\varepsilon \ll I(H^*; E_{t+\Delta t})/(C_\eta L^2)$), the penalty term is small relative to the mutual information and the bound approaches $I(H_t; E_{t+\Delta t}) \gtrsim I(H^*; E_{t+\Delta t})$: nearly all event information is retained.

When is the bound vacuous? Setting the right-hand side of (6) to zero gives the vacuity threshold:

$$\varepsilon_{\mathrm{vac}} = \frac{I(H^*; E_{t+\Delta t})}{C_\eta\, L^2}. \tag{17}$$

When $\varepsilon > \varepsilon_{\mathrm{vac}}$, the bound provides no guarantee. This occurs when (i) $I(H^*; E_{t+\Delta t}) \approx 0$ (no precursors), or (ii) ε is large (poor pretraining), or (iii) L is large (irregular event boundary). Importantly, a vacuous bound does not imply that $I(H_t; E_{t+\Delta t}) = 0$; the encoder may retain information through paths not captured by our analysis (e.g. the encoder directly encodes precursor patterns without going through the predictor).

A.5 Why Predictor Weights Do Not Transfer

Our empirical finding (section I.2) that predictor *architecture* matters but predictor *pretrained weights* do not is explained by a codomain mismatch.

The predictor's architecture and codomain are the same in both stages: $g_\phi \colon \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ throughout, with no layer replaced. What changes is the objective the same map is asked to serve, and what reads its output. During pretraining, g_ϕ is scored directly against the target representation and is rewarded for reproducing all d components of H^*. During finetuning, its output is read by the shared linear event head (LayerNorm and a linear map to one logit per horizon, 769 parameters in total; section 3.2), and only the directions of $\mathbb{R}^d$ that this head can turn into event probability are rewarded.

Formally, pretraining minimises $\mathbb{E}[\|g_\phi(H_t, \Delta t) - H^*\|_1]$, while finetuning minimises $\mathbb{E}[\ell_{\mathrm{BCE}}(\sigma(g'_\phi(H_t, \Delta t)),\, E)]$. The pretraining objective rewards the predictor for reconstructing *all* d components of H^*, most of which encode event-irrelevant dynamics (channel-level forecasting, trend, seasonality). The finetuning objective rewards it for surfacing only the event-relevant directions, which the head then reads out. (Because no layer is replaced, the pretrained weights are a legitimate initialisation for the finetuned map, which is what makes the pretrained-versus-random comparison in section I.2 meaningful at all.) The optimal weight matrices for these two objectives need not be correlated, and indeed our experiments confirm they are not: at 100% labels, pred-FT with pretrained predictor weights lands within ± 0.003 of pred-FT with randomly initialised predictor weights (pooled AUPRC, the metric that ablation was scored in; section I.2).

This is analogous to the observation in vision that pretrained *classifier heads* do not transfer across tasks while pretrained *backbones* do: the backbone compresses the input into a general-purpose representation, while the head specialises to a specific output space.

A.6 Connection to Empirical Results

C-MAPSS (turbofan degradation). Degradation in turbofan engines develops over hundreds of cycles, with sensor readings progressively deviating from healthy baselines. The target encoder, seeing the future interval bidirectionally, captures degradation state with high fidelity: $I(H^*; E_{t+\Delta t})$ is large (the future interval is highly informative about whether failure occurs in that interval). Pretraining achieves low prediction error ε because the dynamics are smooth and near-deterministic. Proposition 1 predicts strong event-information retention, consistent with h-AUROC $= 0.806$ on C-MAPSS-1 and $= 0.568$ on C-MAPSS-2 (5 seeds, dense K=150).

Vacuity in rare-event regimes. The corollary's non-vacuity condition tightens quickly as events become rare. The numerator is at most the event entropy, $I(H^*; E) \le H(E) = O(\pi \log(1/\pi))$ for marginal prevalence π, while $C_\eta = (2\underline{\eta}(1-\overline{\eta}))^{-1}$ grows as the posterior is squeezed toward the prior, so the admissible prediction error scales as $\varepsilon_{\text{vac}} = O\big(\pi^2 \log(1/\pi)/L^2\big)$: halving the event rate cuts it roughly fourfold. The bound therefore says little about benchmarks of isolated, very rare events. That is not the regime most of our surface occupies: prevalence is a function of the horizon, from 0.5% at Δt=1 to 96% at Δt=150 on C-MAPSS-1, so the horizon-averaged metric is dominated by cells where π is far from zero. The vacuous regime is the short-horizon end, which is also where every method is weakest.

CHB-MIT (seizure prediction). We evaluated CHB-MIT as a pilot to test the bound's predictions on a dataset expected to fail; it is excluded from the formal benchmark in table 1 because scalp EEG seizure prediction is a fundamentally different signal regime, with no reliable electrographic precursors observable in the 16-second context window used by all other datasets in our benchmark. Seizure onset in scalp EEG has minimal electrographic precursors in the pre-ictal period, particularly within a 16-second context. The target encoder's representation H^* carries little event information: $I(H^*; E_{t+\Delta t}) \approx 0$. Corollary 2, condition (i), fails regardless of pretraining quality. This is consistent with the empirical result of h-AUROC $= 0.497$ (chance level), and turns CHB-MIT into a clean negative control for the theory rather than an inappropriate target.

C-MAPSS-2 (multi-operating-condition). C-MAPSS-2 adds operating-condition variability to C-MAPSS-1. This increases ε (harder prediction task) without proportionally increasing $I(H^*; E_{t+\Delta t})$ (event information is the same; only noise increases). The bound predicts that C-MAPSS-2 should be harder than C-MAPSS-1 for the same architecture, consistent with the observed drop from 0.806 to 0.568 h-AUROC.

Label efficiency. At 5% labels, pred-FT achieves F1 $= 0.261$ vs. scratch $= 0.035$. The bound supports this observation: the encoder's $I(H_t; E_{t+\Delta t})$ is established during pretraining and does not depend on labels. Reducing labels degrades the finetuning optimisation (finding the right head weights) but cannot reduce the information available in the frozen encoder. Note that the proposition addresses information retention in the encoder, not sample complexity of the finetuning step; the label-efficiency advantage is consistent with, but not directly formalised by, the bound.

A.7 Per-Horizon Generalisation

Proposition 1 is stated for a fixed horizon Δt. We now make the horizon dependence explicit. All quantities on the right-hand side of the bound can vary with Δt: the event posterior $\eta_{\Delta t}(h) = P(E_{t+\Delta t} = 1 \mid H^*_{\Delta t} = h)$, the Lipschitz constant $L(\Delta t)$, the prediction error $\varepsilon(\Delta t) = \mathbb{E}[\|H^*_{\Delta t} - \hat{H}_{\Delta t}\|_2^2]$, and the posterior bounds $\underline{\eta}_{\Delta t}, \overline{\eta}_{\Delta t}$ from **(A4)**.

Proposition 3 (Per-horizon information retention)**.** *Under the assumptions of proposition 1 applied at each horizon $\Delta t > 0$ separately, with horizon-dependent quantities $L(\Delta t)$, $\varepsilon(\Delta t)$, and $C_\eta(\Delta t) = (2\underline{\eta}_{\Delta t}(1-\overline{\eta}_{\Delta t}))^{-1}$:*

$$I\big(H_t;\, E_{t+\Delta t}\big) \;\ge\; I\big(H^*_{\Delta t};\, E_{t+\Delta t}\big) \;-\; C_\eta(\Delta t)\, L(\Delta t)^2\, \varepsilon(\Delta t). \tag{18}$$

Proof. Apply the proof of proposition 1 verbatim at each fixed Δt, with all quantities carrying a subscript Δt. □

Predicted shape of the per-horizon AUROC curve. Equation (18) predicts how the encoder's event information varies with horizon. Three effects combine:

- $I(H^*_{\Delta t}; E_{t+\Delta t})$ is typically non-monotone: very short horizons may not yet encompass precursors; very long horizons dilute the event signal with unrelated dynamics.
- $\varepsilon(\Delta t)$ is monotonically increasing (representations further into the future are harder to predict from H_t).
- $L(\Delta t)$ can increase with Δt if the event boundary sharpens as the event approaches.

The net effect is that $I(H_t; E_{t+\Delta t})$ peaks at intermediate Δt and decays at large Δt. While MI and AUROC are not monotonically related in general, this qualitative shape is compatible with the per-horizon AUROC curves observed in our experiments: AUROC tends to be highest at $\Delta t \in [10, 50]$ cycles for C-MAPSS and at $\Delta t \in [1, 20]$ steps for anomaly datasets, then declines. The decay is more pronounced for the anomaly datasets because $\varepsilon(\Delta t)$ grows faster when the dynamics are irregular.

A.8 Relationship to the Information Bottleneck

The Information Bottleneck (IB) framework (Tishby et al., 2000) seeks a representation T of input X that maximises $I(T; Y)$ (relevance) while minimising $I(T; X)$ (compression):

$$\min_{P(T|X)} \; I(T; X) - \beta\, I(T; Y). \tag{19}$$

JEPA pretraining differs from IB in three ways:

1. **No explicit compression.** JEPA does not penalise $I(H_t; X_{\leq t})$. The encoder is free to retain all information about the past. Implicit compression arises only from the architecture bottleneck ($d = 256$).
2. **Predictive, not discriminative.** The IB target Y is a label; the JEPA target is a future representation H^*. JEPA implicitly encourages high $I(H_t; H^*)$ by minimising prediction error (through the predictor), and, under A1 (target sufficiency), $I(H_t; H^*)$ upper-bounds $I(H_t; Y)$ for any downstream label Y that is a function of H^*, via the Markov chain $H_t \to H^* \to Y$. Without A1 this ordering does not follow: the data-processing inequality bounds $I(H_t; Y)$ and $I(H_t; H^*)$ by $I(H_t; X_{(t,t+\Delta t]})$ separately, and does not order them against each other. This makes JEPA a *looser* but *more general* objective: it retains information about all downstream tasks, not just one.
3. **Asymmetric architecture.** The IB is typically symmetric in X and Y. JEPA imposes causal structure: the encoder sees only the past, the target encoder sees only the future. This causal asymmetry is essential for event prediction, where we cannot condition on the future at test time.

The connection to our result is direct. Let $Y = E_{t+\Delta t}$. By proposition 1:

$$I(H_t; Y) \;\geq\; I(H^*; Y) \;-\; C_\eta\, L^2\, \varepsilon. \tag{20}$$

As the JEPA pretraining loss drives $\varepsilon \to 0$, the encoder's mutual information with Y approaches $I(H^*; Y)$, the maximum achievable by the target representation. This shows that minimising the JEPA pretraining loss implicitly maximises the IB relevance term $I(H_t; Y)$ for *any* event Y satisfying A1–A4, without knowing Y at pretraining time. The price paid relative to IB is that JEPA also retains event-irrelevant information (it does not compress), which manifests as higher representation dimensionality but not as reduced downstream performance.

B Horizon Interval Design

All methods (HEPA, PatchTST, iTransformer, MAE, Chronos-2) use dense unit-step horizons: K=150 for C-MAPSS and TEP ($\Delta t \in \{1, 2, \ldots, 150\}$), and K=200 for all other datasets ($\Delta t \in \{1, 2, \ldots, 200\}$). All methods share the same horizon set per dataset, ensuring fair comparison. The h-AUROC evaluation skips degenerate horizons (event prevalence < 0.001 or > 0.999).

C BASELINE COMPARISON PROTOCOL

Table 1 compares HEPA against four baselines under a matched protocol designed to isolate the effect of the pretraining objective. All five methods share:

- **Matched downstream capacity**: a horizon-conditioned MLP that maps a frozen representation $\mathbf{h}_t$ and horizon Δt to per-horizon event logits (details below), trained with positive-weighted BCE (section 3.2).
- **Identical evaluation**: h-AUROC averaged over non-degenerate horizons, same train/val/test splits, same horizon sets (K=150 or K=200).
- **Identical label budgets**: 100% and 10% label fractions with the same subsampling procedure.

Downstream head architecture. For HEPA, the finetuned component is the pretrained predictor MLP (a 3-layer MLP mapping $[\mathbf{h}_t; \Delta t] \to \hat{\mathbf{h}} \in \mathbb{R}^{256}$; 197.6K params) plus a shared linear event head (LayerNorm + linear $\to$ logit; 769 params), totalling 198K finetuned parameters. The predictor is initialised from pretraining; however, section I.2 shows that this initialisation is worth at most $+0.003$ pooled AUPRC over random initialisation, the metric that ablation was scored in. Note what that does and does not establish: both arms freeze the *same pretrained* encoder, so the comparison isolates the predictor's starting weights and says nothing about the encoder's contribution. The control that would isolate it, pred-FT on a frozen randomly-initialised encoder, is not run in this paper.

The baselines fall into three groups, which differ in what is trained with labels and in how a scalar output becomes a probability surface.

(i) Supervised end-to-end (PatchTST, iTransformer). The encoder is trained from labels together with the same horizon-conditioned predictor and shared linear event head that HEPA uses, and the surface is produced by the same discrete-hazard parameterisation (eqs. (3) and (4)). These baselines tune their whole network against the labels (2.26M parameters for PatchTST), an order of magnitude more label-tuned capacity than HEPA's 198K.

(ii) Frozen encoder, HEPA recipe (MAE). The MAE encoder is pretrained with masked reconstruction, then frozen, and HEPA's predictor and event head are finetuned on top of it exactly as in pred-FT.

(iii) Frozen features, dt-conditioned MLP head (Chronos-2, and the foundation models of table 5). Features are extracted once from the frozen encoder and a dt-conditioned MLP head is trained on them: a linear projection from d_{input} to 256, a learned horizon embedding (256 entries $\times$ 256 dims), followed by LayerNorm and a 3-layer MLP ($256 \to 256 \to 256 \to 1$). At d_{input}=256 this totals 264K parameters, more downstream capacity than HEPA's 198K; for Chronos-2's 768-dimensional features it is 395K. This head emits an unconstrained logit per $(t, \Delta t)$ cell, so, unlike groups (i) and (ii), its surface is *not* monotone in Δt by construction. The TimesFM-1.0 baseline uses the `google/timesfm-1.0-200m-pytorch` checkpoint.

All three groups use the same positive-weighted BCE (eq. (5)), the same splits, labels and dense horizon grids, and the same evaluation.

Encoder training. The methods differ only in how the frozen encoder is obtained:

PatchTST (Nie et al., 2023): channel-independent patching with bidirectional self-attention, trained end-to-end (supervised, no self-supervised pretraining) on each dataset. Same patch size (P=16) and context length (512 steps) as HEPA. 2.26M trained parameters, all tuned against labels. Group (i): the surface comes from HEPA's predictor and shared event head, trained jointly with the encoder. 5 seeds.

iTransformer (Liu et al., 2024b): inverted Transformer that treats each variate (sensor channel) as a token and applies self-attention across variates rather than across time steps. Trained end-to-end (supervised) on each dataset with the same context window and horizon set, also group (i). 5 seeds.

MAE (masked autoencoder (He et al., 2022)): uses the same causal Transformer encoder architecture as HEPA but pretrained with a masked reconstruction objective: random patches are masked and the decoder reconstructs the masked input values. This isolates the effect of predicting future

representations (JEPA) versus reconstructing masked *values* (MAE) under an otherwise identical architecture. After pretraining the decoder is discarded, the encoder is frozen, and HEPA's predictor and event head are finetuned on top of it: group (ii), the matched comparison. 5 seeds.

Chronos-2 (Ansari et al., 2025): foundation model (119M parameters) pretrained on a large external time-series corpus. Representations are extracted per-channel (univariate) and mean-pooled across channels for multivariate datasets. The baseline MLP head is trained on the frozen features. 3 seeds. This is the only method that uses external pretraining data; all others are trained exclusively on the target dataset.

What this protocol isolates, and what it does not. Group (ii) is the controlled comparison: encoder architecture, downstream recipe and output parameterisation are all held fixed, so the h-AUROC difference between HEPA and MAE is attributable to the pretraining objective: predicting future representations versus reconstructing masked values. Group (i) gives the supervised baselines strictly more label-tuned capacity than HEPA (2.26M versus 198K), so a HEPA win there is a conservative result rather than a capacity artefact. Group (iii) does *not* isolate encoder quality on its own: HEPA's surface is monotone in Δt by construction while the dt-MLP head's is unconstrained, so a HEPA win over a frozen foundation model confounds representation quality with that structural prior. We therefore read the foundation-model comparison as a comparison of complete pipelines, not of encoders alone; giving the foundation-model features HEPA's discrete-hazard head would separate the two, and we have not run that ablation. HEPA's predictor initialisation carries negligible benefit in any of the three groups (section I.2).

D Dataset Overview

Table 3: Dataset overview (14 datasets, 11 domains). Default: patch size P=16, sliding window of 512 steps (32 tokens), except C-MAPSS, which uses full engine history (8–23 tokens per engine trajectory).

Dataset	Target event	Sensors	Rate
C-MAPSS FD001–FD004	Engine failure	14	1/cycle
SMAP	Spacecraft fault	25	1 Hz
PSM	Server fault	25	1/min
MBA (ECG)	Cardiac arrhythmia	2	275 Hz
GECCO (water)	Water contamination	9	1/min
ETTm1	Transformer overheating	7	one sample per 15 min
BATADAL (ICS)	Cyber-attack on SCADA	43	1/hour
TEP	Process fault	52	one sample per 3 min
Weather	Heat spike	21	one sample per 10 min
Beijing-AQ	PM2.5 spike	11	1/hour
VIX	Volatility regime	6	1/day

E Architectural Decisions

Decision	Value	Source	Status
Attention mask	Causal (context enc.)	Ablation study	Fixed
Horizon sampling	LogU$[1, K]$	Initial sweep	Default
Target interval	Cumulative $(t, t{+}\Delta t]$	Architecture design	Default
Output param.	Discrete hazard $\rightarrow$ CDF	section 3.2	Default
Target encoder	Bidirectional, detached target	Ablation study	Default
Encoder depth	$L = 2$	Initial sweep	Default
d_{model}	256	Initial sweep	Default
Loss	L1 on L2-norm.; VC on raw	Initial sweep	Default

F FINETUNING-MODE ABLATION

Table 4 compares five finetuning strategies on C-MAPSS under an earlier architecture variant (EMA target, 790K pred-FT params). Predictor finetuning is the only mode that remains competitive at both full and low label budgets; scratch training collapses at 5% labels.

Table 4: **Finetuning-mode ablation on C-MAPSS (5 seeds; earlier architecture variant with EMA target).** F1w is the prevalence-weighted F1 over the $(t, \Delta t)$ cells, the metric this earlier ablation was scored in; it is not comparable to the h-AUROC used elsewhere. This variant uses 790K pred-FT params and 2.37M E2E. Pred-FT outperforms scratch by $+0.226$ at 5% labels (p=0.039, d=1.35). Scratch training collapses at low label budgets (RMSE 33), confirming pretraining value. Under the final architecture (198K pred-FT), pred-FT and E2E achieve equivalent performance at all label budgets ($\Delta \leq 0.003$). † The probe row counts the linear map only; the probe head of this earlier variant, like the final one, also carries a LayerNorm affine (512 parameters).

	100% labels		**5% labels**	
Mode	F1w ↑	RMSE ↓	F1w ↑	RMSE ↓
probe_h (257 p)†	0.30 ± 0.06	16.0 ± 1.5	0.06 ± 0.10	20.4 ± 1.2
frozen_multi (4K p)	0.15 ± 0.03	19.0 ± 0.1	0.18 ± 0.14	24.4 ± 4.9
pred_ft (790K p)	0.39 ± 0.09	16.9 ± 1.7	$\mathbf{0.26 \pm 0.17}$	24.3 ± 6.8
e2e (2.37M p)	0.41 ± 0.12	15.0 ± 1.2	0.18 ± 0.24	$\mathbf{20.1 \pm 1.9}$
scratch (2.37M p)	0.40 ± 0.08	14.5 ± 0.7	0.04 ± 0.05	32.9 ± 2.0

G FOUNDATION MODEL COMPARISONS

Table 5 consolidates the matched-head comparison between HEPA (5 seeds) and four time series foundation models: Chronos-2, MOMENT-1-large (Goswami et al., 2024) (341.2M parameters), TimesFM-1.0 (Das et al., 2024) (203.6M parameters), and Moirai-1.1-R-base (Woo et al., 2024) (91.4M parameters). All five encoders are frozen and feed a dt-conditioned MLP head trained with positive-weighted BCE under the same labels, splits, and evaluation protocol; only the frozen encoder differs. The head is 264K parameters at d_{input}=256 and 395K for Chronos-2's 768-dimensional features (section C).

Table 5: **HEPA vs. four foundation models under a matched frozen-feature protocol** (100% labels). HEPA: 5 seeds. Chronos-2, MOMENT, TimesFM-1.0, Moirai: 3 seeds each. **Bold** = best per row. "—": model not run on that dataset. **HEPA's numbers here differ from table 1** because this table gives every method, HEPA included, the same dt-conditioned MLP head on frozen features, whereas table 1 gives HEPA its predictor-finetuning recipe. The head costs HEPA on some datasets (C-MAPSS-1 0.73 vs. 0.81; GECCO 0.81 vs. 0.88) and helps on others (ETTm1 0.87 vs. 0.81; BATADAL 0.64 vs. 0.57), so the direction is not uniform. Under this encoder-isolating protocol HEPA is best on 4 of the 9 datasets it covers. This is a comparison of encoders, not HEPA's reported operating point.

Dataset	HEPA (5s)	Chronos-2	MOMENT	TimesFM-1.0	Moirai
C-MAPSS-1	$\mathbf{0.73 \pm .02}$	$0.66 \pm .00$	$0.56 \pm .01$	$0.53 \pm .00$	$0.61 \pm .00$
C-MAPSS-2	$0.58 \pm .01$	$0.50 \pm .01$	$\mathbf{0.70 \pm .00}$	$0.60 \pm .01$	$0.66 \pm .00$
C-MAPSS-3	$\mathbf{0.82 \pm .02}$	$0.72 \pm .02$	$0.47 \pm .01$	$0.62 \pm .01$	$0.70 \pm .00$
SMAP	$\mathbf{0.60 \pm .03}$	$0.53 \pm .01$	—	$0.51 \pm .03$	—
PSM	$0.55 \pm .02$	$0.49 \pm .00$	—	$\mathbf{0.57 \pm .01}$	$0.53 \pm .01$
MBA	$0.75 \pm .01$	$0.55 \pm .01$	$\mathbf{0.79 \pm .01}$	$0.76 \pm .01$	$0.57 \pm .02$
GECCO	$0.81 \pm .07$	$0.81 \pm .01$	—	$\mathbf{0.93 \pm .01}$	$0.82 \pm .01$
BATADAL	$0.64 \pm .02$	$0.58 \pm .01$	$0.54 \pm .07$	$\mathbf{0.65 \pm .01}$	$0.36 \pm .01$
ETTm1	$\mathbf{0.87 \pm .00}$	$0.78 \pm .01$	—	$0.59 \pm .01$	$0.60 \pm .00$

Coverage. Chronos-2: 12 of 14 (C-MAPSS-4 and TEP not run; they are "—" in table 1 too). MOMENT: 5 datasets (remaining omitted due to extraction cost). TimesFM-1.0: 9 datasets. Moirai: 7 datasets (SMAP infeasible). MOMENT C-MAPSS-3 (0.47) is below chance, indicating per-channel embeddings anti-predict turbofan degradation. BATADAL Moirai (0.36) is below chance: univariate patching discards cross-sensor correlations. Foundation-model wins cluster on benchmarks favoured by their pretraining corpus; HEPA wins on lifecycle benchmarks (C-MAPSS-1, C-MAPSS-3, ETTm1).

H FULL MTS-JEPA COMPARISON

Table 6 compares HEPA against MTS-JEPA (He et al., 2026) on event-prediction benchmarks from table 1. Both methods use the same context length, patch size, sequence batching, and matched-capacity downstream head; the only difference is the self-supervised objective (HEPA: predictive JEPA with the variance–covariance regulariser; MTS-JEPA: published contrastive + codebook loss). HEPA wins on 8 out of the 9 datasets where MTS-JEPA could be run, with the largest margins on lifecycle and ICS benchmarks (C-MAPSS-1, ETTm1, BATADAL); MTS-JEPA wins on cardiac arrhythmia (MBA). TEP is excluded because the public MTS-JEPA release does not include a chemical-process benchmark and the encoder fails to converge on its 52-dimensional input.

Table 6: **HEPA vs. MTS-JEPA (He et al., 2026).** Mean ($\pm$ std) over available seeds. HEPA: 5 seeds. MTS-JEPA reproduction: 1–3 seeds. Both methods use identical patch size, sequence length, and matched-capacity downstream head; only the SSL objective differs.

Dataset	**HEPA**	MTS-JEPA
C-MAPSS-1	**0.81**$\pm$0.04	0.69$\pm$0.02
C-MAPSS-2	**0.57**$\pm$0.01	0.53$\pm$0.02
C-MAPSS-3	**0.84**$\pm$0.02	0.78$\pm$0.00
SMAP	**0.59**$\pm$0.06	0.49$\pm$0.00
PSM	**0.57**$\pm$0.02	0.48$\pm$0.00
MBA	0.75$\pm$0.04	**0.88**$\pm$0.00
GECCO	**0.88**$\pm$0.06	0.84$\pm$0.00
ETTm1	**0.81**$\pm$0.01	0.75$\pm$0.00
BATADAL	**0.57**$\pm$0.04	0.43$\pm$0.00
TEP	1.00$\pm$0.00	—
HEPA wins	8 out of 9	

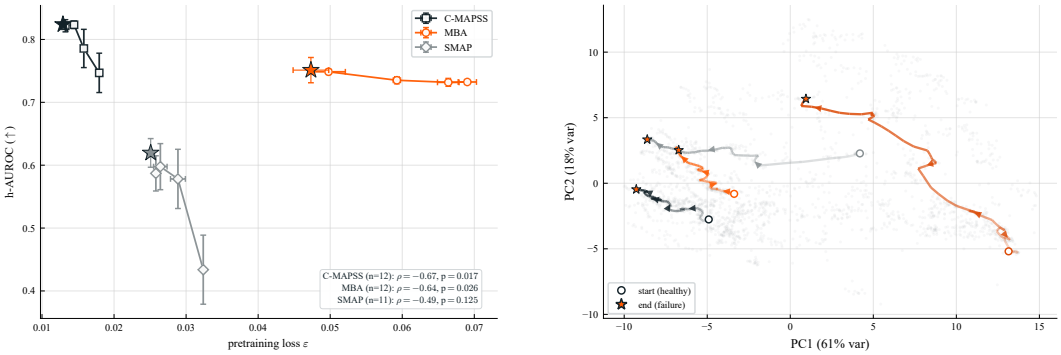

Figure 4: **Self-supervised pretraining learns task-relevant structure.** (a) Pretraining loss ε vs. downstream h-AUROC (↑) at fixed checkpoints across three domains (C-MAPSS-3: ρ=−0.67; MBA: ρ=−0.64; SMAP: ρ=−0.49; 3 seeds, error bars $\pm$1 std). Within a dataset, L, C_η, and $I(H^*; E_{t+\Delta t})$ are constant, so the bound's monotone prediction is directly testable. ★ marks the converged-best snapshot; ε scales differ across datasets so curves cannot be compared horizontally. (b) Principal component analysis (PCA) of pretrained C-MAPSS-1 representations for four test engines. Open circles: first observation (healthy); stars: last observation (near failure). PC1 captures 61% of variance; the encoder organises representations into a smooth degradation manifold without any labels.

I ADDITIONAL ABLATIONS

I.1 DOES THE PREDICTOR HELP AT INFERENCE?

With all weights frozen, a probe on $\mathbf{h}_{\text{past}}$ alone achieves 0.299 F1w at 100% labels, while a frozen multi-horizon probe on $[\hat{\mathbf{h}}_1; \ldots; \hat{\mathbf{h}}_{16}]$ achieves only 0.148. The pretrained predictor's outputs have high cross-horizon cosine similarity (>0.98), so the concatenation is nearly redundant. Finetuning

the predictor pushes the per-horizon outputs apart: pred-FT's value is in reshaping the predictor, not in extracting fixed rollouts.

I.2 Pretrained vs. Random-Init Predictor

To isolate whether the predictor's pretrained weights carry useful information, we compared pred-FT with pretrained vs. random-initialised predictor weights (encoder frozen, 3 seeds each). This ablation predates the switch to h-AUROC as the headline metric and is scored in *pooled AUPRC*: on C-MAPSS-1, pretrained weights yield 0.9257 ± 0.0008 vs. random-init 0.9235 ± 0.0027 ($p = 0.38$); on SMAP, 0.3874 ± 0.0205 vs. 0.3950 ± 0.0286 ($p = 0.34$); on MBA, 0.9465 ± 0.0004 vs. 0.9435 ± 0.0016 ($p = 0.089$). These absolute values are not comparable to table 1: pooled AUPRC inherits the per-horizon prevalence, which on C-MAPSS-1 puts the trivial baseline at 0.957 (section 4), and on the sparse SMAP surface puts it far below 0.5. What the ablation measures is the paired difference, where both arms share metric, splits and seeds: pretrained predictor weights are worth at most $+0.003$, and on SMAP the random-initialised predictor is marginally ahead. The value of pred-FT therefore lies in the frozen pretrained encoder and the finetuning recipe, not in the predictor's initial weights. We have not re-scored this ablation in h-AUROC, so its conclusion is stated for pooled AUPRC only.

I.3 Target Encoder Update Rule: Frozen Target vs. EMA

Our default target-encoder strategy initialises the target encoder from the context encoder's weights and computes the alignment term against a detached target, so the target encoder is not updated during pretraining, with the variance–covariance regulariser (Bardes et al., 2022) (α=0.04) preventing collapse. We compare this against EMA (τ=0.99) at 3 seeds with pred-FT downstream. This ablation predates the final benchmark suite and was run on a 12-dataset set that overlaps it but is not identical: it includes MSL, SMD and SKAB, which are not in table 1, and omits C-MAPSS-4, TEP, Weather, Beijing-AQ and VIX. On the 11 of those 12 with a valid 3-seed comparison (MSL's EMA arm has a single seed), joint training wins on 6 (largest gain: MBA $+0.108$), loses on 3 (largest loss: SMAP -0.048), and ties on 2 at a ± 0.02 threshold. All deltas are within ± 0.11 h-AUROC. Predictor finetuning is robust to the target-encoder update rule; we use the frozen target for its simplicity (no momentum schedule, no separate sync interval).

I.4 Closed Form of the Variance–Covariance Regulariser

Writing s_j for the standard deviation of coordinate j of the raw predictor output $\hat{\mathbf{h}}$ across a batch, and C for its covariance matrix over that batch, the two single-branch terms of VICReg (Bardes et al., 2022) are

$$\mathcal{L}_{\mathrm{VC}} \;=\; \underbrace{\frac{1}{d}\sum_{j=1}^{d}\max\bigl(0,\; 1 - s_j\bigr)}_{\text{variance hinge}} \;+\; \underbrace{\frac{1}{d}\sum_{j\neq k} C_{jk}^{2}}_{\text{covariance penalty}} \;, \tag{21}$$

with a numerical ε=10^{-4} added to each s_j. We take this term as published and claim no part of it; it appears here only so the pretraining objective is reproducible from the paper alone.

I.5 Faithful SIGReg vs. the Variance–Covariance Regulariser

HEPA's collapse term constrains second moments (eq. (21)). LeJEPA (Balestriero and LeCun, 2025) instead constrains the embedding *distribution*, testing isotropic Gaussianity along random projections with an Epps–Pulley statistic (SIGReg). Because that is the principled version of what our regulariser approximates, we implemented it faithfully and ran it head-to-head rather than arguing from analogy.

Protocol. Both arms share the encoder, predictor, head, data, splits, horizon grids, optimiser and finetuning recipe; only the regularisation term differs. SIGReg is applied to the target-encoder branch at α=0.1, matched by the variance–covariance arm at the same weight. Both arms of this ablation therefore run at α=0.1, the released code's default, rather than at the α=0.04 used for the main results; the comparison is internally matched, which is what it is for. Five seeds per cell, on the twelve of the fourteen benchmarks whose public loaders yield non-degenerate event labels; TEP (degenerate

labels) and VIX (no sequences long enough for the event loader) are excluded, and we report that rather than dropping them silently.

Result: parity overall, with a real C-MAPSS gain for SIGReg. Faithful SIGReg is at least as good as the variance–covariance term on 9 of 12 benchmarks, with a mean difference of $+0.016$ h-AUROC. The gain concentrates on C-MAPSS (4 of 4, mean $+0.040$: C-MAPSS-1 $+0.005$, C-MAPSS-2 $+0.047$, C-MAPSS-3 $+0.017$, C-MAPSS-4 $+0.092$). Elsewhere the two are within noise (SMAP $+0.004$, MBA $+0.014$, BATADAL $+0.057$, ETTm1 $+0.002$, Weather $+0.002$, PSM -0.004, Beijing-AQ -0.004), with one real loss on GECCO (-0.045, driven by a single unstable seed; the other four sit at 0.84–0.89).

Why the numbers in this paper use the variance–covariance term. Every result reported elsewhere in this paper was produced with eq. (21), and we report what we ran. The comparison above says the choice is not load-bearing for the paper's claims: the two regularisers agree within noise on most benchmarks, and where they differ the differences run both ways: SIGReg is ahead across C-MAPSS, while the variance–covariance term is ahead on GECCO. The reported numbers are therefore conservative on lifecycle data rather than uniformly conservative. We flag the C-MAPSS gap as worth pursuing rather than folding it into results it did not produce.

Validity of the comparison. An earlier internal run reported that faithful SIGReg damaged performance; those runs were discarded because their pretraining took 1–4 seconds where the matched variance–covariance runs on the same datasets took 60–130, i.e. they had not trained. The runs above do not have that defect: per-dataset pretraining time matches between arms to within 15% (C-MAPSS-2: 138 s vs. 157 s; GECCO: 3.9 s vs. 3.5 s; short times reflect small datasets, not truncated runs), no run raised a collapse flag, and predictor-output standard deviation is healthy in both arms (SIGReg 0.23–0.71, variance–covariance 0.10–0.31).

I.6 Predictor Dynamics Visualisation

Figure 5 shows how the finetuned predictor transforms the encoder's representations across prediction horizons. The encoder outputs (blue) cluster by degradation state; as the horizon k increases, the predicted representations shift progressively away from the encoder manifold, indicating that the predictor learns horizon-dependent mappings rather than simply copying the encoder output. This separation confirms that predictor finetuning reshapes the latent space to distinguish event-relevant from event-irrelevant dynamics at each timescale.

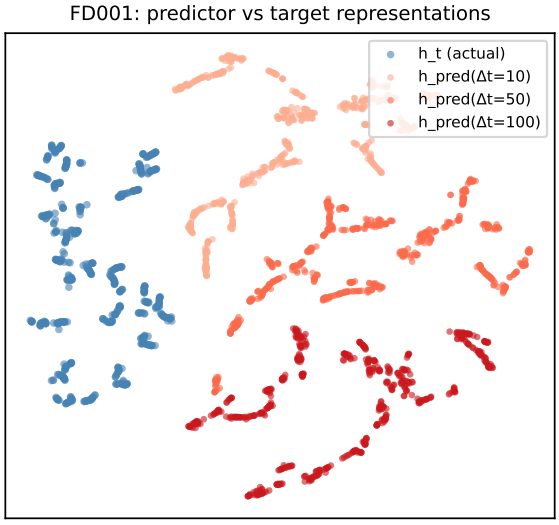

Figure 5: **Predictor outputs in latent space (C-MAPSS-1).** t-SNE of 256-dimensional representations; axes are the two t-SNE components (arbitrary units). Blue: encoder output $\mathbf{h}_t$. Light to dark red: predicted representations at horizons $k = 10, 50, 100$. Outputs shift progressively with k.

J LEGACY METRICS

C-MAPSS RMSE is derived from E$[\Delta t]$ of the stored probability surface; this CDF-based estimator differs from the point-prediction protocol of STAR (Fan et al., 2024) (replicated RMSE 12.2), so the gap reflects both protocol and SSL-vs-supervised differences. PA-F1 matches the cited baselines' protocol. HEPA domain-metric values: SMAP PA-F1 0.862 ± 0.014 (vs. MTS-JEPA (He et al., 2026) 0.34), PSM PA-F1 0.95 ± 0.01 (vs. MTS-JEPA 0.62).

K PA-F1 INFLATION

How the threshold is chosen. Both F1 and PA-F1 are reported at the threshold that maximises F1 on the surface being scored: the test surface itself. This is an oracle threshold, not a deployable decision rule: it is the convention used by the point-adjusted anomaly-detection results we compare against, and we adopt it for comparability, but it is an upper bound on what a threshold picked on validation data would achieve. It also interacts with w^+ (eq. (5)), which shifts the probability scale away from 0.5, so the maximising threshold is generally not 0.5.

PA-F1 (Point-Adjusted F1) credits an entire anomaly segment as correctly predicted if any single timestep within it exceeds the threshold (Xu et al., 2022). This protocol inflates reported F1 dramatically when anomaly segments are long and prevalence is non-trivial: Kim et al. (2022a) show that a random detector can exceed $F_1 = 0.9$ PA on some datasets. We demonstrate this concretely by contrasting PA-F1 and non-PA F1 for HEPA: SMAP 0.862 PA vs. 0.474 non-PA, PSM 0.950 PA vs. 0.575 non-PA. Any PA-F1 number should therefore be read alongside the corresponding non-PA number.

L PREPROCESSING DETAILS

Horizons shorter than a patch, and context lengths that are not multiples of P. Δt enters the predictor as a continuous scalar rather than a token index, so a horizon shorter than the patch size ($\Delta t < P{=}16$) is well defined and needs no special handling. A context length that is not a multiple of P (SMAP, PSM and MBA use 100 steps) is right-padded to the next multiple (112 steps, 7 tokens); the padded positions are marked in the attention mask and the context summary $\mathbf{h}_t$ is read from the last *valid* token, so the padding cannot enter the representation.

Dataset	Channels	Preprocessing	Context
C-MAPSS FD001–FD004	14 sensors (after drop)	per-subset min-max	full trajectory
SMAP	25 (after drop)	z-score on train	100 steps
PSM	25 (after drop-constant)	pre-normalised	100 steps
MBA ECG	2 leads	z-score on train	100 steps
GECCO water	9 quality sensors	z-score on train	512 steps
BATADAL SCADA	43 tank/pump/pressure	z-score on train	512 steps
ETTm1 power	7 load/temperature	z-score on train	512 steps
TEP	52 process vars	z-score on train	512 steps
Weather	21 climate vars	z-score on train	512 steps
Beijing-AQ	11 air quality vars	z-score on train	512 steps
VIX	6 market vars	z-score on train	512 steps

For C-MAPSS one timestep is one flight cycle and the context is the engine's full cycle history rather than a sliding window; patching is unchanged ($P{=}16$ cycles per token), which is why a trajectory yields 8–23 tokens. Elsewhere in this paper that convention is sometimes called "cycle-as-patch", which is a misnomer we avoid here: the cycle is the timestep, not the patch. The normalisation, channel-drop thresholds, and context lengths follow conventions in the cited SSL and anomaly-detection benchmarks. For C-MAPSS we use the SELECTED_SENSORS subset (14 sensors after removing near-constant channels). RUL cap is 125 cycles, following STAR (Fan et al., 2024). Specific numeric thresholds are in the code repository.

M HYPERPARAMETERS

Positive weight. w^+ in eq. (5) is estimated once per run, before finetuning, as $N_{\text{neg}}/N_{\text{pos}}$ counted over every $(t, \Delta t)$ cell of the finetuning training split, then clamped to $[1, 1000]$, or to $[1, 200]$ when that split holds fewer than 50 positive cells, so that an almost event-free low-label subset cannot drive the weight arbitrarily high. It is a single scalar per run: the same value is applied at every horizon, even though per-horizon prevalence varies by two orders of magnitude across Δt (section 4).

All hyperparameters are fixed across datasets unless noted. Seeds $\{0, 1, 2, 3, 4\}$ for 5-seed runs and $\{0, 1, 2\}$ for 3-seed runs. Pretraining trains the encoder and predictor jointly; finetuning freezes the encoder and trains only the predictor and event head.

Table 7: **Hyperparameters.** Fixed across all datasets. Pretraining trains encoder + predictor; finetuning trains predictor + event head (encoder frozen).

Stage	Hyperparameter	Value
Pretraining (encoder + predictor)	Optimizer	AdamW
	Learning rate	3×10^{-4}
	Weight decay	1×10^{-2}
	Batch size	64
	Epochs	100 (patience 10)
	VC weight α	0.04
	Horizon sampling	LogUniform$[1, K]$
Finetuning (predictor + event head)	Optimizer	AdamW
	Learning rate	1×10^{-3}
	Weight decay	1×10^{-2}
	Batch size	64
	Epochs	50 (patience 10)
	Pos-weight w^+	$N_{\text{neg}}/N_{\text{pos}}$
Architecture	Encoder layers	2
	d_{model}	256
	Attention heads	4
	Patch size P	16
	Dropout	0.1

N PAIRWISE SIGNIFICANCE TESTS

Table 8 reports Welch's two-sided t-test (unpaired, unequal variance) between HEPA and each architectural baseline at each (dataset, label-fraction) cell with at least 3 seeds per arm. Of the 42 (dataset, baseline) cells at 100% labels, 39 carry a test (C-MAPSS-4 has no significance run). HEPA wins 19 at $p < 0.05$ and loses 3; the remaining 17 are not significantly different.

Table 8: **Pairwise Welch's t-test, HEPA vs. each baseline.** Markers: * $p<0.05$, ** $p<0.01$ in HEPA's favour; ↓ denotes HEPA loses at $p<0.05$; blank = not significant.

	FD001	*FD002*	*FD003*	*FD004*	*SMAP*	*PSM*	*MBA*	*GECCO*	*BATADAL*	*TEP*	*ETTm1*	*Weather*	*Beijing*	*VIX*
vs. PatchTST		**	**	—	*			**	↓		**	**		
vs. iTransformer	**	**	**	—	**	**	↓	*		**		*	**	*
vs. MAE	**		**	—						*	↓			

Every marker above is taken from the stored per-seed results rather than from which mean is larger: a baseline ahead of HEPA is recorded as a loss only when the test says so. On that basis HEPA has **three** significant losses at $p<0.05$, not one per dataset where a baseline leads: BATADAL to PatchTST (p=0.004), MBA to iTransformer (p=0.005) and ETTm1 to MAE (p=10^{-7}). Several gaps that look like losses in table 1 are not significant: SMAP vs. MAE (p=0.12), BATADAL vs. MAE (p=0.28) and Beijing-AQ vs. both PatchTST (p=0.37) and MAE (p=0.83). The strongest

wins are on C-MAPSS and against iTransformer (10 significant wins, 1 loss and 2 not significant, across the 13 benchmarks tested).

These p-values are uncorrected. Applying Holm–Bonferroni over the full family of 78 tests (39 per label fraction; C-MAPSS-4 carries no test) leaves 11 wins and 1 loss significant at α=0.05; the per-baseline sub-families are reported in the stored results. C-MAPSS-4 is "—" throughout because no significance run covers it: its table 1 cells come from a later batch than the paired tests.

O CALIBRATION ANALYSIS

Table 9 reports Expected Calibration Error (ECE) and Brier score on five datasets for which we hold verified HEPA probability surfaces (single-seed, s=42; 10 equal-width bins; Murphy decomposition: Brier = Reliability − Resolution + Uncertainty).

Table 9: **Calibration of HEPA probability surfaces on five datasets.**

Dataset	Base rate	ECE ↓	Brier ↓	Reliability	Resolution
FD001	0.78	0.272	0.238	0.129	0.062
FD004	0.34	0.030	0.124	0.001	0.102
Weather	0.05	0.196	0.124	0.082	0.009
BeijingAQI	0.24	0.183	0.177	0.050	0.054
VIX	0.21	0.310	0.241	0.102	0.026

Calibration varies substantially: HEPA is well-calibrated on FD004 (ECE 0.030) but miscalibrated on FD001 (ECE 0.272) and VIX (ECE 0.310). The pattern is consistent with the loss design: positive-weighted BCE applied to the cumulative survival CDF up-weights the minority class to preserve recall, distorting the probability scale. Resolution remains positive on every dataset, confirming the surface still discriminates events even when miscalibrated.

ECE is pooled, and that hides where the error is. The table pools every $(t, \Delta t)$ cell into one reliability diagram, but per-horizon prevalence moves by roughly two orders of magnitude along Δt (section 4), so a pooled ECE is a statement about the surface, not about any horizon. Stratifying by horizon band on stored C-MAPSS-1 surfaces (5 seeds, thirds of the horizon axis) puts the miscalibration where the events are rare: ECE 0.043 for $\Delta t \in [1, 50]$ at base rate 0.31, against 0.032 for $[51, 100]$ and 0.030 for $[101, 150]$ at base rate 0.94. Reporting a single pooled figure averages the poorly calibrated short-horizon region into a long-horizon region that is nearly saturated.

Recalibration should act on the hazards, or through a single shared map. For applications requiring calibrated probabilities, post-hoc Platt scaling or isotonic regression on a held-out fold is the usual remedy, but it should be applied to the per-interval hazards $\lambda_{\Delta t}$ rather than to $p(t, \Delta t)$ horizon by horizon: any monotone map applied to each λ_j leaves $p = 1 - \prod_j (1 - \lambda_j)$ non-decreasing in Δt, as does a single non-decreasing map shared across horizons, whereas fitting an *independent* transform per horizon to p can reorder adjacent horizons and destroy the monotonicity the parameterisation exists to guarantee. We did not run this recalibration for the reported results, so no recalibrated ECE is quoted here.