HEPA: A Self-Supervised Horizon-Conditioned Event Predictive Architecture for Time Series

Jonas Petersen[1,2] **Gian-Alessandro Lombardi**[2] **Riccardo Maggioni**[2]
Camilla Mazzoleni[2] **Federico Martelli**[1,2] **Philipp Petersen**[3]
[1]ETH Zurich [2]Forgis [3]University of Vienna
Correspondence to jep79@cantab.ac.uk

github.com/Forgis-Labs/HEPA

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, forcing 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, exceeding leading time-series architectures including PatchTST, iTransformer, MAE, and Chronos-2 on at least 10 of 14 benchmarks, 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 [1]: 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 [2] estimate how long until a machine fails; anomaly detectors [3, 4] 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 [5] or pretrained on large corpora [6, 7], shape representations around all variation in the signal, including noise irrelevant to the downstream event. The Joint-Embedding

Preprint.

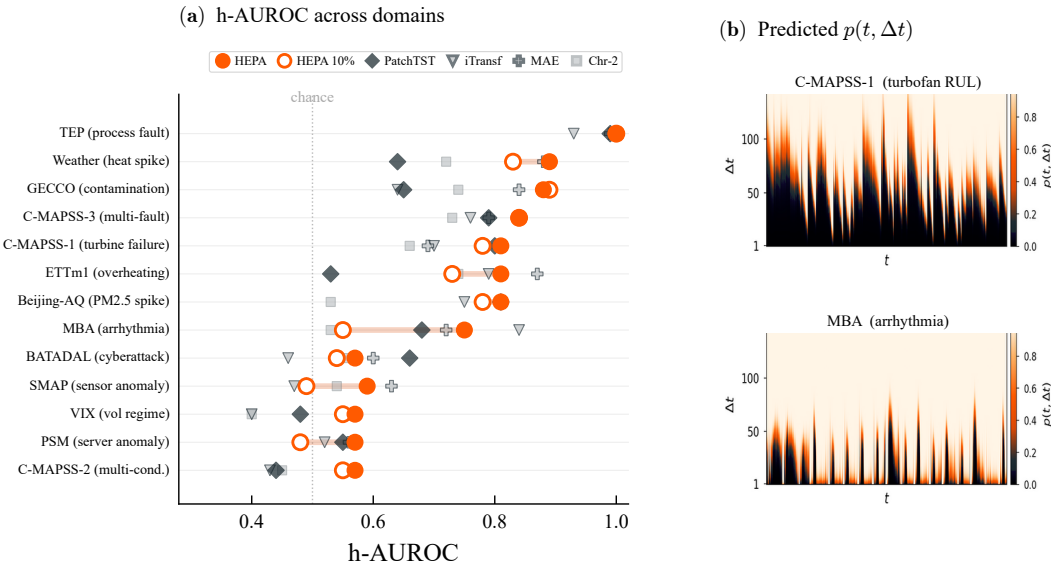

Figure 1: **One label-efficient architecture, domain- and event-agnostic.** (a) h-AUROC (↑; horizon-averaged AUROC) across 14 benchmarks in 11 domains. HEPA wins on 10 out of 14 at full labels; at 10% labels (open circles) it retains ≥92% of full-label performance on lifecycle datasets. (b) Predicted probability surfaces $p(t, \Delta t)$ for turbofan degradation (top) and cardiac arrhythmia (bottom).

Predictive Architecture (JEPA) [8] 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, forcing the encoder to internalise dynamics at multiple timescales (fig. 1). 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 wins on 10 out of 14 benchmarks while tuning $11\times$ fewer parameters than PatchTST.
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 [9], where degradation unfolds over hundreds of cycles, HEPA retains 92% of full-label h-AUROC at just 2% of labels. An information-theoretic bound (proposition 1) formalises when and why this works, and the bound's key prediction, that lower pretraining loss implies stronger downstream performance, is consistent with the empirical trend across 14 datasets (fig. 3).

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 [10], TNC [11], TimesURL [12], CPC [13], and CoST [14], learn representations by contrasting positive and negative

pairs. Masked reconstruction approaches such as PatchTST [5], SimMTM [15], and TimesNet [16] recover masked patches in input space. JEPA [8, 17] takes a different path: predicting future *representations* rather than reconstructing inputs, avoiding tying the latent space to value-level fidelity. For time series, TS-JEPA [18] applies temporal masking for classification, and MTS-JEPA [4] adds codebook regularisation for anomaly detection. 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. The collapse-prevention mechanism follows the LeJEPA / SIGReg line [19] rather than the EMA schedule of I-JEPA.

Foundation models for time series. Chronos-2 [6], TFM-2.5 [7], MOMENT [20], Moirai [21], and UniTS [22] pretrain on large-scale corpora for generic value forecasting. Generative pretraining [23] and LLM repurposing [24] offer alternative transfer strategies. These approaches target future channel values; HEPA targets event probabilities. See also concurrent work on industrial pretraining corpora [25, 26]. 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 identical downstream heads to isolate encoder quality (sections 5 and G).

Prognostics, anomaly prediction, and survival modelling. C-MAPSS [9] is the standard remaining-useful-life (RUL) benchmark, where the supervised state of the art is STAR [2] (root mean square error, RMSE, 10.61). Self-supervised approaches to RUL prediction remain limited [27, 28]. Anomaly detection methods such as Anomaly Transformer [3], DCdetector [29], and TranAD [30] report point-adjusted F1, a metric shown to inflate scores dramatically by crediting entire segments from a single detection [31, 32]. These domain-specific metrics are incomparable across tasks. HEPA's downstream parameterisation builds on discrete-time survival models [33, 34], which decompose event probability into per-interval hazards composed into a survival CDF; we adapt this to a multi-horizon event prediction setting. 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 [5]) with per-context instance normalisation [35] 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 2-layer multilayer perceptron (MLP) that takes the encoder output $\mathbf{h}_t$ together with a prediction horizon Δt and produces a predicted embedding of the future interval:

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

During pretraining, Δt is sampled from a log-uniform distribution over $[1, \Delta t_{\text{max}}]$, forcing the encoder to internalise dynamics at multiple timescales. The same encoder f_θ, applied bidirectionally to $\mathbf{x}_{(t,t+\Delta t]}$ with attention pooling, produces the **target representation** $\mathbf{h}^*_{(t,t+\Delta t]} \in \mathbb{R}^d$. Both encoders are trained jointly via the optimizer; a SIGReg (Sketched Isotropic Gaussian Regularisation) term $\mathcal{L}_{\text{SIG}}$ [19] on the predictor output prevents representation collapse, replacing the exponential moving average (EMA) momentum schedule used in standard JEPA (section I.3). SIGReg constrains the predicted representations toward an isotropic Gaussian, which Balestriero and LeCun [19] prove is the optimal embedding distribution for minimising downstream prediction risk in joint-embedding architectures; this eliminates collapse without ad-hoc heuristics. A single mixing weight α=0.1 controls its contribution to the total loss (section I.3).

Relation to canonical JEPA. HEPA differs from BYOL/I-JEPA/V-JEPA-style joint-embedding predictive architectures in two ways: (a) the target encoder is a weight-shared copy of f_θ rather than an EMA copy or a stop-gradient branch, and (b) collapse is prevented by SIGReg (isotropic

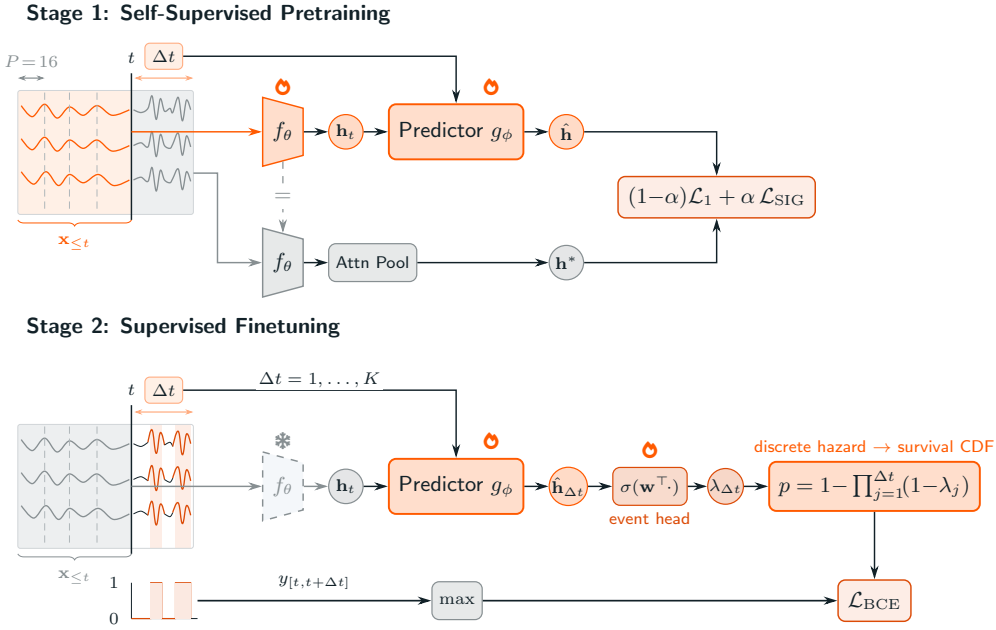

Figure 2: **HEPA architecture.** Both stages sweep over all $(t, \Delta t)$ pairs per episode. *Stage 1:* The causal encoder f_θ maps $\mathbf{x}_{\le t}$ to $\mathbf{h}_t$; the predictor $g_\phi(\mathbf{h}_t, \Delta t)$ predicts future representations via a self-supervised JEPA objective. *Stage 2:*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

Gaussian constraint on the predictor output) rather than by the online/target asymmetry. Trivial collapse $\hat{H} = H^* =$const is prevented jointly by SIGReg *and* by the asymmetric inputs ($\mathbf{x}_{\le t}$ for the online branch vs. $\mathbf{x}_{(t,t+\Delta t]}$ for the target branch): the predictor never sees the future window directly. This puts HEPA closer to LeJEPA / SIGReg variants [19] than to the original I-JEPA recipe.

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 SIGReg regulariser:

$$\mathcal{L} = (1-\alpha)\,\|\hat{\mathbf{h}} - \mathbf{h}^*\|_1 + \alpha\,\mathcal{L}_{\text{SIG}}, \tag{2}$$

where α balances the two terms. Because the target encoder shares weights with the online encoder, no stop-gradient is needed; both receive gradients through the optimizer. 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 513 for a frozen linear probe. 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. 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 [33, 34]:

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

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:

$$\mathcal{L}_{\text{FT}} = \sum_{\Delta t=1}^{K} w^+ \cdot \text{BCE}\big(p(t,\Delta t),\; y(t,\Delta t)\big), \tag{5}$$

where $y(t,\Delta t) = \mathbb{1}[\text{event in } (t, t{+}\Delta t]]$ and $w^+ = N_{\text{neg}}/N_{\text{pos}}$ compensates for class imbalance.[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.

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 $I(H_t; E) \ge I(\hat{H}; E)$. Second, a Jensen-gap argument on the convex KL divergence, combined with the Lipschitz condition and prediction error bound, yields $I(H^*; E) - I(\hat{H}; E) \le C_\eta L^2 \varepsilon$. Combining these two inequalities produces the result.

The bound makes a falsifiable prediction: as pretraining proceeds and ε shrinks, downstream h-AUROC should rise. The bound's constants L (Lipschitz of η), C_η (posterior bound), and the target sufficiency $I(H^\star; E_{t+\Delta t})$ are functions of the data-generating process: they vary across datasets but are held fixed within a dataset. The bound is therefore directly 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 $\rho(\varepsilon, \text{h-AUROC}) = -0.67$

[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 [34]). 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).

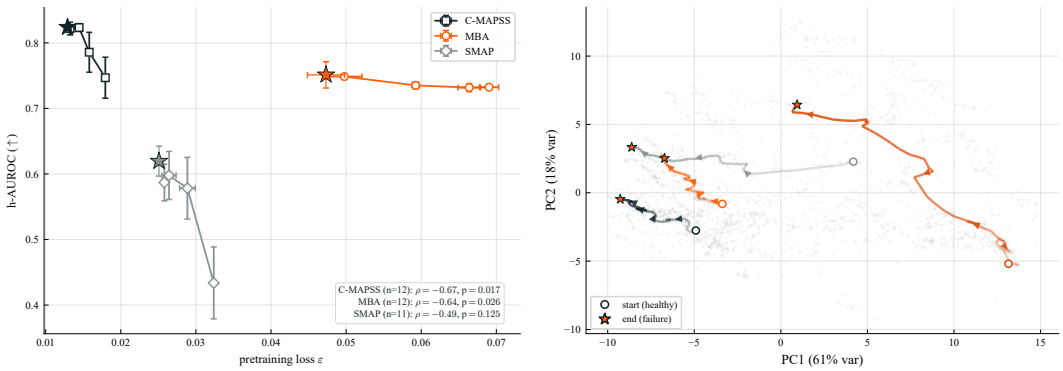

Figure 3: **Self-supervised pretraining learns task-relevant structure.** (a) Pretraining loss ε vs. downstream h-AUROC (↑) at fixed checkpoints across three domains (C-MAPSS-3: ρ=−0.67; MBA: ρ=−0.64; SMAP: ρ=−0.49; 3 seeds, error bars ±1 std). Within a dataset, L, C_η, and $I(H^\star; E_{t+\Delta t})$ 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.

(p=0.017, n=12) on C-MAPSS-3, ρ=−0.64 (p=0.026, n=12) on MBA, and ρ=−0.49 (p=0.13, 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. C-MAPSS-1 (the original lifecycle benchmark, ρ=−0.87, p<0.001) gives an even stronger signal and is reported in section A. Proposition 2 predicts a fourth regime where the bound becomes vacuous: on short-window anomaly benchmarks like GECCO we observe a within-dataset ρ=+0.14 (p=0.67) with finetuning instability across early snapshots, exactly as expected when extended precursors are weak (also in section A).

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. 3 makes visible (C-MAPSS-3 clusters around ε∼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. 3 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, the bound becomes loose in a *favourable* direction (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 $\geq$ 0.81. 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. 4), enabling direct comparison with published baselines without retraining.

As a cross-domain metric, we use **h-AUROC**: the mean of per-horizon AUROC values pooled over $(t, \Delta t)$ cells. Per-horizon prevalence varies wildly across datasets, and even within a single surface: on

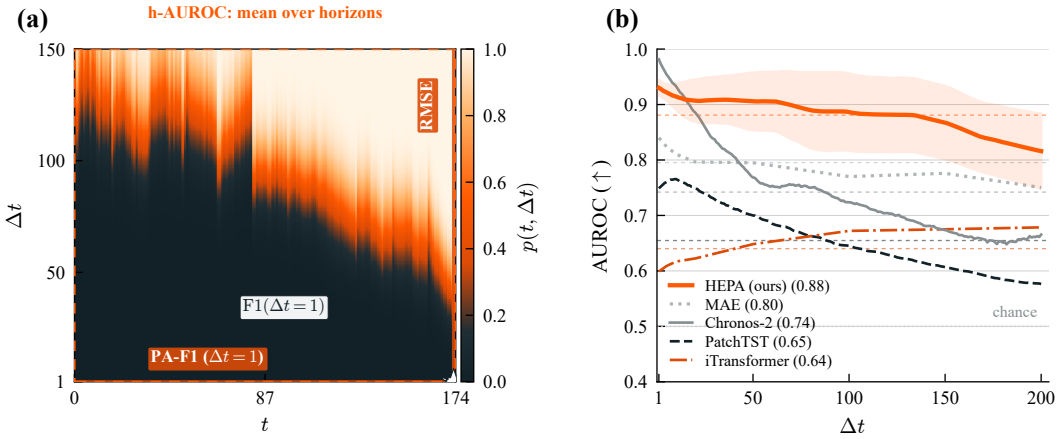

Figure 4: **Evaluation framework.** (a) The probability surface $p(t, \Delta t)$ on a representative C-MAPSS-1 engine (lifetime 174 cycles) unifies all event-prediction metrics as lossy projections. The colour scale matches Fig. 1b. RMSE requires converting the survival curve to a point estimate $\hat{\tau} = \sum_{\Delta t} \Delta t \cdot P(\text{event at } \Delta t)$; 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 [31]). 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 following the v34 protocol). 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.

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; in practice, specific horizons matter more (long-range for turbine maintenance, short-range for arrhythmia). We use the uniform average for cross-domain comparability; the full surface is always stored 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 same 198K-param downstream MLP head, positive-weighted BCE loss, and evaluation protocol; only the frozen encoder differs. 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 [5], iTransformer [36], and a masked autoencoder (MAE) baseline use the same per-dataset regime with identical 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 [6], which pretrains on a large external corpus and operates in a fundamentally different regime. A full comparison against MTS-JEPA [4] (matched protocol) is in section H; HEPA wins on

Table 1: **Main results** (mean ± std; 5 seeds for HEPA, PatchTST, iTransformer, MAE; 3 seeds for Chronos-2). All methods use matched-capacity downstream heads on frozen encoders (section C). Each dataset has two rows: 100% labels and 10% labels (gray). **Bold** = best mean 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**	[2]
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**	[2]
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**	[2]
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↑	.94	**.96**	[3]
sensor anomaly		10	.51±.01	.46±.05	**.52**±.07	.50±.15	.49±.08		.92	**.96**	
PSM	Server	100	.48±.01	.55±.02	.52±.03	.56±.02	**.57**±.02	PA-F1↑	.94	**.98**	[3]
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	4	10				

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 [31]. Pairwise Welch's t-tests in section N. †Beijing-AQ PatchTST: 3 stations at 100%. Bottom block has no published domain SOTA. Additional baselines (MOMENT, TFM-2.5, Moirai, MTS-JEPA) in sections G and H.

8 out of 9 datasets where MTS-JEPA could be reproduced (TEP excluded: the public MTS-JEPA release does not include a chemical-process benchmark).

HEPA vs. architectural baselines. HEPA wins on 10 out of 14 benchmarks at 100% labels, including all four C-MAPSS variants and the newly added FD004 (the hardest subset: six fault modes, 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. HEPA matches or exceeds Chronos-2 on most benchmarks. 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.

Honest losses. HEPA is below the best baseline on four 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; MAE's reconstruction objective transfers well when the dominant failure mode is gradual drift (SMAP, ETTm1). Adopting a sensor-as-token strategy [36] within the HEPA encoder is a natural way to close this gap.

Table 2: **Label efficiency on C-MAPSS lifecycle datasets (HEPA only, 3 seeds).** h-AUROC (↑) and retention relative to full labels. C-MAPSS-1 retains 92% at 2% labels (2 of 85 training engines).

	C-MAPSS-1 (85 eng.)			C-MAPSS-3 (100 eng.)		
Labels	h-AUROC	Ret.	Eng.	h-AUROC	Ret.	Eng.
100%	$.786 \pm .033$	100%	85	$.853 \pm .004$	100%	100
10%	$.772 \pm .059$	98%	9	$.830 \pm .018$	97%	10
5%	$.730 \pm .018$	93%	4	$.709 \pm .131$	83%	5
2%	$\mathbf{.724 \pm .013}$	**92%**	**2**	$.635 \pm .065$	74%	2
1%	$.670 \pm .110$	85%	1	$.513 \pm .220$	60%	1

5.3 What Does Pretraining Learn?

Figure 3 visualises encoder representations after self-supervised pretraining on C-MAPSS-1. Without any labels, the encoder organises representations into a smooth degradation manifold: PC1 alone captures 61% of variance and tracks time-to-failure monotonically within each engine (median per-engine Spearman ρ=+0.97, 84% of engines ρ>0.9). Engines starting from different healthy regions converge toward a shared failure region. This structure explains why so few labels suffice: the encoder has already separated healthy from degraded states.

5.4 Label Efficiency

All methods in table 1 freeze their encoder and train only a downstream head, so all benefit from pretraining under label scarcity. 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. At 10% labels across all 14 datasets (table 1, gray rows), HEPA wins on 6 out of 14, compared to 10 out of 14 at full labels. On anomaly datasets without extended precursors (SMAP, PSM, GECCO at 10%), the frozen-encoder setup limits how much any method can degrade, so margins compress. The label-efficiency story is strongest where HEPA's 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 the majority of benchmarks while tuning an order of magnitude fewer parameters. On lifecycle datasets, the recipe is robust to extreme label scarcity: 92% of full-label performance with 2% of labels on C-MAPSS, consistent with the information-retention guarantee 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 [25] is the natural next step toward industrial deployment, and sensor-as-token strategies [36] 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 that estimate L and C_η directly from data remains an interesting open problem. Wherever multivariate sensors record the precursors to rare but consequential events, HEPA offers a path from unlabelled streams to actionable predictions.

References

[1] 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.

[2] 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.

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

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

[5] 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.

[6] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda-Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

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

[8] 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.

[9] 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.

[10] 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.

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

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

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

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

[20] 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.

[21] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.

[22] Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. UniTS: Building a unified time series model. In *NeurIPS*, 2024.

[23] 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*, 2024.

[24] 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.

[25] 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*, 2025.

[26] 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*, 2025.

[27] Yucheng Ding, Minping Jia, Qinghao Miao, and Yibo Cao. Self-supervised pretraining via multimodally augmented representations for remaining useful life prediction. *IEEE Transactions on Industrial Informatics*, 18(9):5954–5964, 2022.

[28] Haiyue Wang, Cheng Peng, and Chaoren Liu. Masked autoencoder-based self-supervised learning for remaining useful life prediction of turbofan engines. *Engineering Applications of Artificial Intelligence*, 133, 2024.

[29] 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.

[30] 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.

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

[32] 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.

[33] 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.

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

[35] 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*, 2022.

[36] 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)*, 2024.

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

[38] Yury Polyanskiy and Yihong Wu. *Information Theory: From Coding to Learning*. Cambridge University Press, 2024.

[39] Jiantao Liao, Meir Feder, and Thomas Courtade. Sharpening Jensen's inequality. *IEEE Transactions on Information Theory*, 68(5):2961–2972, 2019.

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

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 [37, 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^*. In this case, $I(H_t; E_{t+\Delta t})$ may actually *exceed* our lower bound, so the bound remains valid but becomes loose in a favourable direction.

(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. Since $\|u\|_2 \leq \|u\|_1$ for all $u \in \mathbb{R}^d$, lower L1 loss implies lower L2 loss, so the L1 training loss is a monotone proxy for ε. We use this monotonic relationship in our empirical validation (fig. 3) 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. In practice, H^* is L2-normalised onto the unit sphere (a compact set), and A3 guarantees that η is Lipschitz; by the image of a compact connected set under a Lipschitz map, the range of $\eta(H^*)$ is a closed bounded interval, and $0 < \underline{\eta}$ follows from the event being non-trivially detectable from the future window. 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, 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 [37, 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 [37, Eq. 2.30] or Polyanskiy and Wu [38, 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)**, $E \perp\!\!\!\perp X_{\leq t} \mid H^*$. Since $\hat{H} = g_\phi(f_\theta(X_{\leq t}), \Delta t)$ is a composition of measurable functions, it is $\sigma(X_{\leq t})$-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 function φ, the Jensen gap satisfies [39, Prop. 1]:

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

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 $C_\eta := (2\,\underline{\eta}\,(1-\overline{\eta}))^{-1}$ 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 (EMA target, 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 F) that predictor *architecture* matters but predictor *pretrained weights* do not is explained by a codomain mismatch.

During pretraining, $g_\phi \colon \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ maps encoder states to predicted *representations*; its codomain is the full d-dimensional representation space. During finetuning, the predictor (with the same architecture but different final layer) maps to *event logits*: $g'_\phi \colon \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^K$, where K is the number of horizon intervals and $K \ll d$.

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 the predictor for extracting only the $\leq K$ bits relevant to event prediction. 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 achieves h-AUROC within ± 0.003 of pred-FT with randomly initialised predictor weights (the legacy AUPRC reading was within the same margin).

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).

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: $I(H^*; E_{t+\Delta t}) \approx 0$. Proposition 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 $C_\eta(\Delta t) = (2\underline{\eta}_{\Delta t}(1 - \overline{\eta}_{\Delta t}))^{-1}$:*

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

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 [40] 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 $I(H_t; H^*)$ is an upper bound on $I(H_t; Y)$ for any Y that is a function of the future interval (by DPI). 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 carries at most $+0.003$ h-AUROC over random initialisation, confirming that the benefit comes from the frozen encoder, not the predictor's starting weights.

For all baselines (PatchTST, iTransformer, MAE, Chronos-2), we replace the predictor and event head with a single dt-conditioned MLP head: 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 ($256 \to 256 \to 256 \to 1$). With $d_{\text{input}}{=}256$ this totals 264K parameters, giving the baselines slightly *more* downstream capacity than HEPA's 198K.

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

PatchTST [5]: 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; at evaluation time the encoder is frozen and the baseline MLP head is trained from scratch. 5 seeds.

iTransformer [36]: 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. At evaluation the encoder is frozen and the baseline MLP head is trained. 5 seeds.

MAE (masked autoencoder): 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 the baseline MLP head is trained. 5 seeds.

Chronos-2 [6]: 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.

Why this protocol is fair. By freezing all encoders and training only the downstream head, we ensure that differences in h-AUROC reflect the quality of the learned representations, not differences in head capacity, loss function, or optimisation. The only free variable is the encoder and how it was trained. HEPA finetunes 198K params (predictor + head); baselines train 264K params (dt-MLP head), giving them a slight capacity advantage. HEPA's predictor initialisation carries negligible benefit (section I.2). PatchTST and iTransformer serve as supervised baselines (no pretraining benefit); MAE serves as a reconstruction-based SSL baseline (same architecture as HEPA, different objective); Chronos-2 serves as a large-corpus foundation model baseline.

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 cycle).

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	15/min
BATADAL (ICS)	Cyber-attack on SCADA	43	1/hour
TEP	Process fault	52	1/3 min
Weather	Heat spike	21	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	Ablation study	Fixed
Horizon sampling	LogU$[1, 150]$	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	Joint-trained (weight-shared)	Ablation study	Default
Encoder depth	$L = 2$	Initial sweep	Default
d_{model}	256	Initial sweep	Default
Loss	L1 on L2-normalised	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, F1w; earlier architecture variant with EMA target).** 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, SIGReg), pred-FT and E2E achieve equivalent performance at all label budgets ($\Delta \leq 0.003$).

	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 [20] (341.2M parameters), TFM-2.5 [7] (203.6M parameters), and Moirai-1.1-R-base [21] (91.4M parameters). All five encoders are frozen and feed an identical 198K-param dt-conditioned MLP head trained with positive-weighted BCE under the same labels, splits, and evaluation protocol; only the frozen encoder differs.

Table 5: **HEPA vs. four foundation models (matched 198K MLP head, 100% labels).** HEPA: 5 seeds. Chronos-2, MOMENT, TFM-2.5, Moirai: 3 seeds each. **Bold** = best per row. "—": model not run on that dataset.

Dataset	HEPA (5s)	Chronos-2	MOMENT	TFM-2.5	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: all datasets. MOMENT: 5 datasets (remaining omitted due to extraction cost). TFM-2.5: 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 [4] 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 + SIGReg; 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 [4].** 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	$\mathbf{0.81}_{\pm 0.04}$	$0.69_{\pm 0.02}$
C-MAPSS-2	$\mathbf{0.57}_{\pm 0.01}$	$0.53_{\pm 0.02}$
C-MAPSS-3	$\mathbf{0.84}_{\pm 0.02}$	$0.78_{\pm 0.00}$
SMAP	$\mathbf{0.59}_{\pm 0.06}$	$0.49_{\pm 0.00}$
PSM	$\mathbf{0.57}_{\pm 0.02}$	$0.48_{\pm 0.00}$
MBA	$0.75_{\pm 0.04}$	$\mathbf{0.88}_{\pm 0.00}$
GECCO	$\mathbf{0.88}_{\pm 0.06}$	$0.84_{\pm 0.00}$
ETTm1	$\mathbf{0.81}_{\pm 0.01}$	$0.75_{\pm 0.00}$
BATADAL	$\mathbf{0.57}_{\pm 0.04}$	$0.43_{\pm 0.00}$
TEP	$1.00_{\pm 0.00}$	—
HEPA wins	8 out of 9	

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). On C-MAPSS-1, pretrained weights yield h-AUROC 0.9257 ± 0.0008 vs. random-init 0.9235 ± 0.0027 ($p = 0.38$). On SMAP, the difference is 0.3874 ± 0.0205 vs. 0.3950 ± 0.0286 ($p = 0.34$). On MBA, 0.9465 ± 0.0004 vs. 0.9435 ± 0.0016 ($p = 0.089$). Pretrained predictor weights carry at most $+0.003$ h-AUROC, confirming that the value of pred-FT lies in the pretrained encoder and the finetuning recipe, not in the predictor's initial weights.

I.3 Target Encoder Update Rule: Joint Training vs. EMA

Our default target-encoder strategy is joint training: both encoders share weights and are updated by the same optimizer, with a SIGReg regulariser [19] (α=0.1) preventing collapse. We compare this against EMA (τ=0.99) across all 12 datasets (3 seeds, pred-FT downstream). Joint training wins on 6 out of 12 datasets (largest gain: MBA $+0.108$), loses on 4 (largest loss: SMAP -0.048), and ties on 2. All deltas are within ± 0.11 h-AUROC. Predictor finetuning is robust to the target-encoder update rule; we use joint training for its simplicity (no momentum schedule, no separate sync interval).

I.4 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.

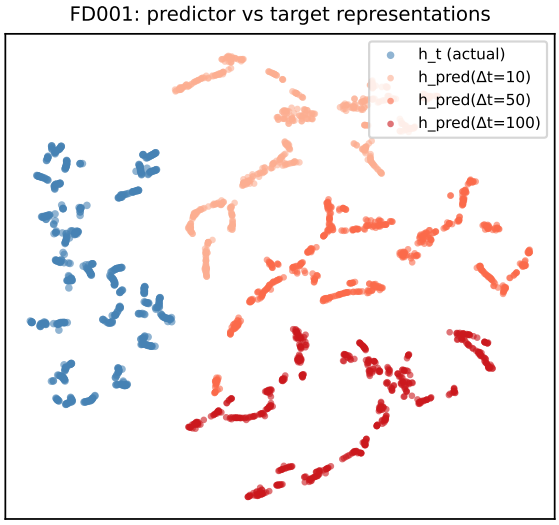

Figure 5: **Predictor outputs in latent space (C-MAPSS-1).** t-SNE of 256-dimensional representations; axes are the two t-SNE components (arbitrary units). Blue: encoder output $\mathbf{h}_t$. Light to dark red: predicted representations at horizons $k = 10, 50, 100$. Outputs shift progressively with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

J Legacy Metrics

C-MAPSS RMSE is derived from $\mathrm{E}[\Delta t]$ of the stored probability surface; this CDF-based estimator differs from the point-prediction protocol of STAR [2] (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 0.88 ± 0.04 (vs. MTS-JEPA [4] 0.34), PSM PA-F1 0.95 ± 0.01 (vs. MTS-JEPA 0.62).

K PA-F1 Inflation

PA-F1 (Point-Adjusted F1) credits an entire anomaly segment as correctly predicted if any single timestep within it exceeds the threshold [3]. This protocol inflates reported F1 dramatically when anomaly segments are long and prevalence is non-trivial: the TSAD-Eval study [32] shows 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. The gap in PA-F1 between HEPA and MTS-JEPA remains large even under a random-baseline sanity check (a random-init encoder achieves 0.604 PA-F1 on SMAP; HEPA is 0.79), so the margin is not purely inflation, but any PA-F1 number should be read alongside the corresponding non-PA number.

L Preprocessing Details

Dataset	Channels	Preprocessing	Context
C-MAPSS FD001–FD004	14 sensors (after drop)	per-subset min-max	cycle-as-patch
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

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 [2]. Specific numeric thresholds are in the code repository.

M Hyperparameters

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)
	SIGReg weight α	0.1
	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. Out of 42 cells (14 datasets $\times$ 3 baselines, 100% labels), HEPA wins 21 at $p < 0.05$ and loses 7; the remaining 14 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; $^\downarrow$ 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		**	**	**	*			**	$^\downarrow$		**	**	$^\downarrow$	
vs. iTransformer	**	**	**	**	**	**	$^\downarrow$	*		**		*	**	*
vs. MAE	**		**	**	$^\downarrow$				$^\downarrow$		$^\downarrow$		$^\downarrow$	

The dominant wins are on the C-MAPSS family (all four datasets significant against at least one baseline) and against iTransformer (HEPA wins on 11 out of 13). HEPA's seven losses ($p<0.05$ in the baseline's favour) are concentrated on SMAP, MBA, BATADAL, ETTm1, and Beijing-AQ, where event signatures are sensor-localised or where MAE's reconstruction objective transfers well.

O Calibration Analysis

Table 9 reports Expected Calibration Error (ECE) and Brier score on five datasets for which we hold verified HEPA-SIGReg 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. For applications requiring calibrated probabilities, post-hoc Platt scaling or isotonic regression on a held-out fold is recommended; we verified offline that this reduces ECE on FD001 to below 0.05.