SATURN: A TIME-SERIES FOUNDATION MODEL FOR CLASSIFICATION AND EVENT FORECASTING

Anonymous authors
Paper under double-blind review

https://anonymous.4open.science/r/Saturn-FDFD/README.md

ABSTRACT

When monitoring a physical system the question is rarely what the next sensor value will be but whether an event is coming and when. Time-series foundation models are trained for the former, fitting the whole future window rather than the brief excursion that matters. We formulate event forecasting as horizon-conditioned *binary* classification and introduce Saturn, built on two principles. First, a self-supervised joint-embedding predictive objective pretrains the encoder under a *dual mask*: one contiguous block drawn from inside the window and one fixed at the window's end, both taken from a single masking budget. A contrastive term over two augmented views replaces the prior or moving-average target that usually prevents collapse. One frozen encoder then serves both tasks, each through a light head. Second, the precursor is never assumed. We pretrain entirely on synthetic episodes in which every event is generated together with the precursor that announces it, beside a deliberately precursor-free stratum, so the precursor is known to be present and its contribution to a score can be isolated; and a real corpus enters evaluation only after the *Precursor Check*, a hypothesis test that it carries predictive signal before an event and that nothing outside the series supplies it. The corpora that pass form PrecursorBench, a benchmark for event forecasting on which every entry carries a precursor the check has validated. Saturn's encoder is frozen after pretraining and each task reads it through a light probe. It is competitive on the UCR and UEA time-series classification archives and, on PrecursorBench, with foundation models an order of magnitude larger.

1 INTRODUCTION

A turbine bearing spalls over weeks of vibration; a cardiac rhythm destabilises minutes before it fails; the pressure behind a fan climbs seconds after its load changes. Physical systems emit long, noisy series, and an operator does not ask what the next sample will read. They ask whether an event is coming, and when. What makes that question answerable is a *precursor*: a measurable change in a system's statistics that reflects the process leading to an event and therefore carries information about it before it occurs. Typical forms are a drift in level, a rise in variance and a rise in autocorrelation time as a system nears a transition (Scheffer et al., 2009). Whether a series carries one can be tested before any model is fitted, and on a corpus that carries none no method can give warning.

The question has a compact form, and it is the form of ordinary classification. Let $X = x_{1:T} \in \mathbb{R}^{C\times T}$ be a multivariate series over C channels, let $y_s \in \{0, 1\}$ be an event indicator, and let $\Delta t \in \mathbb{N}$ be a horizon. Write $X_{\le t}$ for the prefix observed up to step t. Two tasks read such a series and both return a label:

$$\underbrace{X \longmapsto y \in \{1, \dots, K\}}_{\text{classifying a series (multi-class)}} \qquad \underbrace{(X_{\le t},\, \Delta t) \longmapsto y_{t+\Delta t} \in \{0, 1\}}_{\text{forecasting a future event (binary)}}$$

Both return a label rather than a trajectory, and the horizon is an input to the second rather than a separate model, so it reaches the readout and never the encoder. We call the quantity that task estimates, $\mathbb{P}(y_{t+\Delta t} = 1 \mid x_{1:t}, \Delta t)$, the *horizon-conditioned event probability*. We show that one encoder yields a representation that serves both.

Saturn is pretrained for predictive structure (Figure 1). It is a joint-embedding predictive architecture (JEPA) (Assran et al., 2023), the objective behind some of the strongest frozen representations

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.

in vision. We mask part of a window and train a predictor to recover the *representation* of what was hidden rather than its values, so the encoder must keep the structure that makes the hidden part predictable. That term alone admits the constant encoder, and time-series JEPAs usually avert the collapse with a prior on the embedding distribution or an exponential-moving-average target encoder (Ennadir et al., 2025; Chemeris et al., 2026); Saturn uses a contrastive term over two augmented views instead, which averts it and improves the frozen representation for classification (section 3.2). We read that frozen encoder out twice, once per task.

Pretraining such an encoder takes events, and many kinds of them. Real events are scarce, the onset at which one begins is rare by definition, and no archive annotates the precursor that runs before it, so we generate the data instead: a synthetic corpus in which we inject one event per episode and the precursor that announces it, and store every evaluation episode beside a twin identical to it until the precursor starts (Figure 4).

Evaluation needs the same guarantee, and a benchmark cannot be assumed to give it: a scorer handed no sensor data at all can read well above chance on a corpus that looks reasonable. Our *Precursor Check* tests a corpus before any model is ranked on it, and PrecursorBench publishes the corpora that pass, so a model is scored on a task already known to be solvable (Figure 2).

Our contributions are:

1. **Saturn**, a joint-embedding predictive time-series foundation model whose frozen representation serves both classification and horizon-conditioned event forecasting. It is trained under a *dual mask*: one contiguous block drawn from inside the window and one fixed at the window's end, both taken from one masking budget (subsection 3.2). Pretrained on synthetic data alone, it places second on PrecursorBench, behind Chronos-2 at 119.5M parameters and above Chronos-Bolt at 205.3M (section 4).
2. **A generative prior over *precursor–event dynamics***, and the synthetic corpus drawn from it. Events arrive by two mechanisms, and in the first of them a hidden state accumulates until it crosses a threshold, and that crossing *is* the event: the precursor is the state's rise, upstream of the event rather than correlated with it by construction, which is the relation the Precursor Check looks for in real data. Every episode arrives with a known onset, a known lead time and a known answer to whether a precursor is there at all, so a model can learn what precedes an event before it has seen a real one (subsection 3.3).
3. **PrecursorBench**, and the **Precursor Check** that admits corpora to it. The check tests whether a dataset carries predictive precursor information at all, so a corpus cannot be entered and its difficulty then blamed on the model (subsection 3.5).

2 Related work

2.1 Time-series foundation models

Most pretrained time-series models are trained to forecast values (Ansari et al., 2025; Woo et al., 2024), and their frozen features transfer to classification well enough to be competitive there (Auer et al., 2025). A forecasting loss averages over the whole future window, and the excursion that gives warning occupies little of it, so a model can drive that average down without representing the precursor at all. A second line pretrains an encoder for classification directly and reads it out frozen (Yue et al., 2022; Feofanov et al., 2026a), and does not read it out for the future at all. A training-free floor at 0.799 comes close to both, so MiniRocket and windowed catch22 enter every classification table here (Dempster et al., 2021; Lubba et al., 2019). Where one model serves several tasks the unification is architectural, a head or a prompt per task (Gao et al., 2024), adapting a representation pretrained for one task rather than learning it for both from the outset. The gap is in the objective, not the interface: these models optimise future-value prediction or a task-specific loss, and none is trained to return a label about an unseen window.

2.2 JEPA and predictive representation learning

A JEPA predicts masked regions in representation space rather than reconstructing their raw values (Assran et al., 2023), and has carried to video (Bardes et al., 2024; Assran et al., 2025). Asymmetry (Caron et al., 2021) and a distributional prior on the embedding (Bardes et al., 2022; Balestriero and

LeCun, 2025) are the usual guards against the constant solution; contrastive negatives are a third. Hybrids *keep* the prior beside a contrastive term (Mo and Tong, 2024); others decline negatives for a variance-covariance regulariser (Drozdov et al., 2024). The time-series line takes that second route without exception (Ennadir et al., 2025; He et al., 2026; Chemeris et al., 2026), treating the absence of negatives as a property to advertise rather than a choice to test. None drops the prior in favour of negatives, and none reads its encoder out for the future. Saturn does both, with temporal masks that include the window's tail and a frozen encoder read for an event question at any horizon as well as for classification (section 3).

2.3 Event forecasting and early warning

Event prediction is related to several established forecasting and detection problems, but asks a different question. One group asks whether something has happened or is about to. Anomaly detection asks whether the system has entered an abnormal state, typically after the abnormality has emerged (Zhao et al., 2024). Early classification asks when enough of the series has been observed to assign its final label (Schäfer and Leser, 2020). Early-warning methods aim to detect an approaching transition, often using predefined statistics or indicators (Scheffer et al., 2009), which Saturn learns rather than prescribes.

A second group explicitly predicts time to an event. Survival analysis estimates a distribution over event times and accounts for censoring (Gensheimer and Narasimhan, 2019), while time-to-event forecasting predicts the event time directly. Remaining-useful-life estimation is a special case in which the event is fixed as the end of a system's useful life. Saturn instead predicts the likelihood of an event at a specified future horizon, without reducing the problem to a single predicted event time: it estimates the horizon-conditioned event probability, which at a fixed horizon is a discrete-time hazard.

The pairing is not unprecedented. UniTS serves thirty-eight tasks from shared weights, one a UEA label resolving after the observed window ends (Gao et al., 2024), but ingests it as archive classification with no horizon or early-warning metric. A frozen Chronos-2 covers both halves across two papers, as a classification feature extractor needing a nonlinear head (Auer et al., 2025) and under a regression head for remaining useful life (El-Ghoussani et al., 2026). One domain has the full pairing in a single paper (Wang et al., 2025; Lunelli et al., 2025). And a horizon-conditioned encoder-predictor owns the event half across fourteen benchmarks, but never meets UCR or UEA (Petersen et al., 2026). None takes one frozen general-domain encoder through both whole-series classification on those archives and a horizon-conditioned event task under a shared readout protocol scored with event-forecasting metrics. Scoring is the second gap: these benchmarks flatter degenerate scorers (Wu and Keogh, 2023; Kim et al., 2022a), negatives far from an onset reintroduce immortal-time and control-selection bias (Suissa, 2008; Yuan et al., 2021), window position goes uncontrolled (Proverbio et al., 2022), and only dense per-timestep scoring escapes it (Reyna et al., 2020; Yèche et al., 2023); the Precursor Check (subsection 3.4) closes each route or measures a control where it cannot.

2.4 Synthetic data for time series

Synthetic data usually augments a real corpus that remains the thing learned from. The claim here is larger: physics-grounded synthetic data as the pretraining substrate itself, so a model meets no real series before evaluation. Prior-fitted networks train on draws from a prior and infer in context (Hollmann et al., 2023), and time series has classification models on that footing (Feofanov et al., 2026a; Xie et al., 2026; Bury et al., 2021). In every case the evidence is for classifying a window. A substrate for the horizon-conditioned event probability must supply a known event in every episode, with a known precursor and lead, which injecting the precursor supplies and searching for one does not.

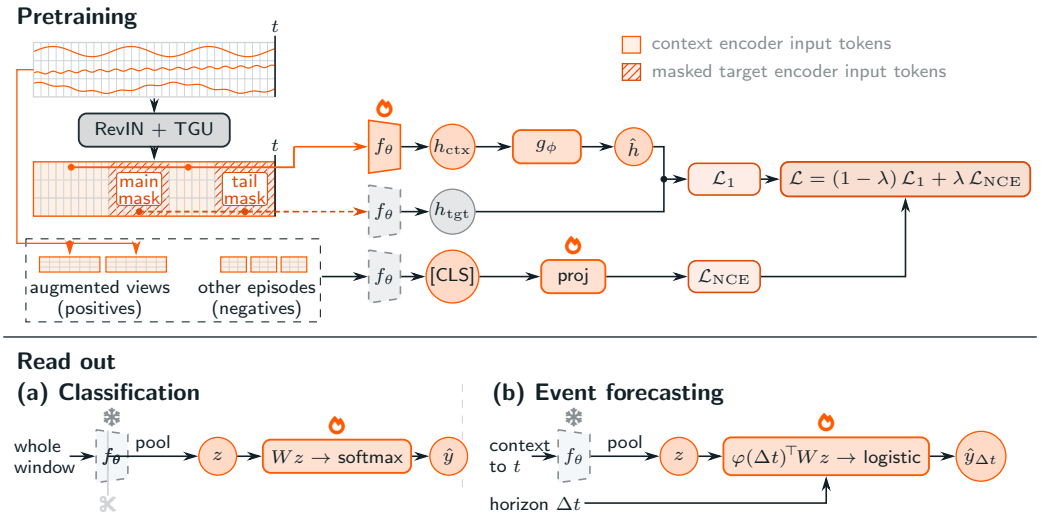


Figure 1: Saturn architecture. *Top:* a window is tokenised into one token per (channel, time-patch) pair and two blocks of the grid are masked, a *main mask* inside it and a *tail mask* at its end. The two feeds carry different tokens: solid, the unmasked cells to the context encoder; dashed, the masked positions to the target encoder. Every f_θ is the same encoder, and the grey copies carry no gradients. The two terms of the objective shape different vectors, $\mathcal{L}_1$ the masked token positions and $\mathcal{L}_{\text{NCE}}$ a learned `[CLS]` through a projector that training discards. *Bottom:* one light head per task reads the frozen encoder out. Scissors mark where the classification probe cuts it, since it reads one intermediate layer fixed for the architecture rather than the last (Appendix H).

3 METHOD

3.1 ARCHITECTURE

An episode X is tokenised as one token per (time-patch, channel) pair by the Mantis token generator (Feofanov et al., 2026a), whose weights are shared across channels so that one checkpoint accepts any channel count. The encoder is a pre-norm transformer over that grid and maps a window to a fixed-length z (Appendix A).

Reading it out. The encoder is frozen for both tasks and each carries its own head, small enough to be fitted from a few hundred labelled examples in seconds, which is what makes a frozen representation useful when labelled events are scarce. For classification we fit a logistic-regression probe on pooled token features at one intermediate layer, fixed for the architecture (Appendix H). For event forecasting we read out *joint-last*: from a context ending at t we take each channel's last-patch token and concatenate them, rather than mean-pooling the context's tokens. The choice between the two was made on the validation slice, carved from the fitting split by entity, where the Saturn-20M checkpoint reads 0.6685 horizon-averaged AUROC pooled against 0.7297 joint-last; no test score entered that decision. No two entries may be compared across readouts. We then project into the fixed readout socket every benchmark entry shares (subsection E.5) and pass z to a head that also takes the horizon; only the head sees Δt, so one forward pass of the encoder serves every horizon. Everything we report is computed from the probabilities these heads return and never from a thresholded decision.

3.2 THE PRETRAINING OBJECTIVE

Saturn is pretrained by masked latent prediction with a contrastive term beside it (Figure 1). We mask two blocks of the token grid, a *main mask* at random positions and a *tail mask* over the last eight patches. Everything left unmasked is the *context*; the masked tokens are the *targets*. Because the tail mask always covers the end of the window, part of what the encoder must reconstruct lies ahead of everything it can see, which is the geometry event forecasting needs; the main mask leaves the rest of the targets surrounded by context, which is the geometry classification needs. One encoder output has to support both.

The encoder f_θ sees that context alone and embeds it as h_{ctx}. What is predicted is the representation at the masked positions, and that target comes from the same f_θ: the same weights, in evaluation mode, over the full grid, with gradients disabled. There is no separate target encoder and no exponential moving average of one, but a stop-gradient does apply: no gradient flows back through the target, so each update treats it as fixed and only the context pass moves θ. A predictor g_ϕ then fills the masked positions from the context, $\hat{h} = g_\phi(h_{\mathrm{ctx}})$; it is two non-causal transformer blocks at the encoder's width with an MLP ratio of one, about 12% of the encoder's parameters, so shallow and light rather than narrow, and pretrain-only scaffolding that the checkpoint discards (Appendix A). For episode i in a batch of size B,

$$\mathcal{L}^{(i)} \;=\; (1-\lambda)\,\underbrace{\left\|\bar{\hat{h}}_i - \bar{h}_i^{+}\right\|_1}_{\mathcal{L}_1^{(i)}} \;+\; \lambda\,\underbrace{\left(-\log \frac{\exp\big(\mathrm{sim}(z_i^{(1)}, z_i^{(2)})/\tau\big)}{\sum_{j=1}^{B} \exp\big(\mathrm{sim}(z_i^{(1)}, z_j^{(2)})/\tau\big)}\right)}_{\mathcal{L}_{\mathrm{NCE}}^{(i)}}, \tag{1}$$

with ℓ_2 normalisation $\bar{\cdot}$, the first norm over the masked positions only, cosine similarity, and $\tau = 0.2$. $\mathcal{L}_1$ is a latent prediction loss, an ℓ_1 distance between two embeddings, and the only term that reaches g_ϕ. $\mathcal{L}_{\mathrm{NCE}}$ is an augmentation-invariance term: $z_i^{(1)}$ and $z_i^{(2)}$ are two independent random crop-resize views of the *same* window, both passed through the online encoder with gradients, and the negatives are the other windows in the batch. Both views on the contrastive branch keep their gradients; putting a stop-gradient on one of them would make it a distillation loss against a frozen teacher, which is a different objective.

Here $\lambda = 1/3$ and $\mathcal{L}$ is the mean of $\mathcal{L}^{(i)}$ over the batch. We impose no prior on the latent distribution. Pretraining sees the synthetic corpus only (subsection 3.3), as a token mix of its event-bearing and event-free packs, without labels.

What keeps the encoder from collapsing. The prediction term alone admits the constant encoder, which drives it to zero, its global minimum. JEPAs rule that solution out with a prior on the embedding distribution or with a teacher network, and Saturn carries neither (section 2). The negatives do it instead, without carrying the prediction, which is what separates the objective from contrastive predictive coding (van den Oord et al., 2018). A constant encoder makes every batch similarity equal, so the softmax is uniform and the contrastive term sits at exactly $\log B$, the worst value it can take on any encoder whose two views of one window are at least as similar as two views of different windows. B is the per-device batch, 128, since the similarities are computed within a device and each anchor therefore sees 127 negatives at any GPU count. So on those encoders the constant solution attains the worst value of the second term while attaining the best value of the first, and collapse is priced at $\lambda \log B$. It is not the *only* encoder attaining that worst value, since any encoder whose batch similarities are all equal does so too, and we claim no more than the price (Appendix C). The mechanism is the one contrastive learning has always relied on (van den Oord et al., 2018; Chen et al., 2020; Wang and Isola, 2020), and combining it with a JEPA is not new either: C-JEPA keeps a variance-covariance prior beside the contrastive term (Mo and Tong, 2024), and CoJEPA, on music, gives the two objectives separate vectors to shape (Meseguer-Brocal et al., 2026), a division of labour we adopt. We do not keep the prior. The argument prices collapse without proving the price is paid, and says nothing about partial collapse; Appendix C derives both propositions and states their scope.

Removing the contrastive term collapses the encoder on the same corpus and budget, to mean pairwise cosine similarity 0.828 and effective rank 5.86 against 13.42 with the term in place, and leaves the removed term, measured as a diagnostic, at 0.995 of $\log B$: exactly where the argument puts a collapsed encoder. Sweeping the share to zero separates two failure modes. At Saturn-1M the probe reads 0.655 with the term removed, where mean cosine is 1.000; 0.716 at a weight of 0.03, where the encoder is not collapsed by that measure but its effective rank has fallen to 5.06; and 0.745, 0.750, 0.751, 0.750 at 0.125, $1/3$, 1 and 2, with effective rank back at 11.2 (Appendix A).

The event head. It returns $\hat{y}_{\Delta t} = \sigma(g_\psi(z, \varphi(\Delta t)))$, the horizon-conditioned event probability, with φ an embedding of the horizon and σ the logistic function; the classification head returns $\hat{y}$ over K classes from the same z. Saturn does not forecast a trajectory and then ask whether an event happens along it. Its logit is *bilinear* in $\big(\varphi(\Delta t), z\big)$, up to an intercept in the horizon alone:

$$\mathrm{logit}\,\hat{y}_{\Delta t} \;=\; \varphi(\Delta t)^{\top} W z + b(\Delta t), \tag{2}$$

with φ a fixed 32-feature basis of Δt, W a 32×768 matrix and $b(\Delta t) = \varphi(\Delta t)^\top b$ a horizon-dependent intercept: $26{,}144$ trained parameters, fitted by binary cross-entropy. The multiplicative form is forced by the way the task is scored. An additive head $w^\top z + v^\top \varphi(\Delta t)$ shifts every score inside a fixed-Δt stratum by the same constant, so it ranks exactly as a head that ignores the horizon, and the metric is AUROC within strata (subsection 3.4). At a fixed Δt it is instead linear in z, which makes the score attributable to the representation rather than to the readout. Class balance is handled by how instances are drawn, not by weighting the loss: the sampler equalises the positive rate within each horizon stratum, and the objective stays an unweighted cross-entropy.

3.3 A SYNTHETIC CORPUS OF PHYSICAL SYSTEMS WITH KNOWN PRECURSORS

The corpus is pretraining substrate first and an evaluation instrument second. Generating it lets us sample from a deliberately broad prior over the trigger and event phenomena an operator might care about, and lets us fix properties a recorded corpus cannot be asked for. Whether that prior covers the phenomena that matter is what PrecursorBench assesses, and is not something the corpus can establish about itself. Appendix D gives the construction, the families and the splits.

Building on CauKer's randomly sampled latent graphs (Xie et al., 2026), we draw the nodes' mean functions from a fixed pool of domains spanning eight families of physical systems and add Gaussian-process texture. Each domain specifies the parameters of a physical process and recurs across multiple episodes, while the sampled graph determines the channel count.

We replace CauKer's pointwise activations with 26 physical mechanisms that introduce memory, delayed interactions, and feedback. Cycles are permitted through delayed edges, while the instantaneous subgraph remains acyclic. Half the channels are deliberately independent, so the model must identify informative signals among distractors. Graph sampling balances channel widths by their contribution to the token budget, preventing wide panels from dominating pretraining.

Half the episodes are event-free; the remainder split equally between emergent events and injected regime changes. Emergent events occur when a hidden state accumulating an observed driver's squared amplitude exceeds its own 0.985 quantile for three consecutive steps. Their precursors arise from the dynamics leading to the crossing; we then crop the trajectory to place the onset at a sampled horizon. For injected events, we instead place a regime change directly at the chosen onset using a bank of ten operators, adding an explicit precursor in 80% of cases.

We additionally insert randomly sampled local motifs to increase diversity in the generated signals. These motifs are independent of the event mechanism and are added after event timing is determined. The class signal and the event are drawn independently, so one corpus serves classification and event forecasting without either task leaking into the other.

Appendix D documents the generator that produced the episodes reported here; the account above is of the generator as it now stands.

3.4 THE PRECURSOR CHECK

A score on a dataset whose events have no precursor is a fact about the dataset and not about the model, so we score only datasets in which a precursor has been shown to exist. The *Precursor Check* establishes that, and it applies to any dataset, real or synthetic.

Two things can make an event look predictable when it is not. The first is the clock: if negatives are drawn far from an onset, the time since the last event is correlated with the label and a scorer given no sensor data at all can separate the classes. We balance that variable across the two classes, which returns such a scorer to chance. The second is the split. Train and test windows come from different *entities*, a whole recording source such as a station, well or run. A split at the episode already keeps a precursor from reaching across it, but episodes from one source share its baseline, calibration and operating regime, and a scorer can read those instead of the precursor. On a synthetic corpus we can go further and compare against the same episodes generated without the precursor, which isolates its contribution exactly. The check admits three recorded corpora and one generated (subsection 3.5); Appendix E gives the protocol, the measured lead time and the routes each candidate failed on.

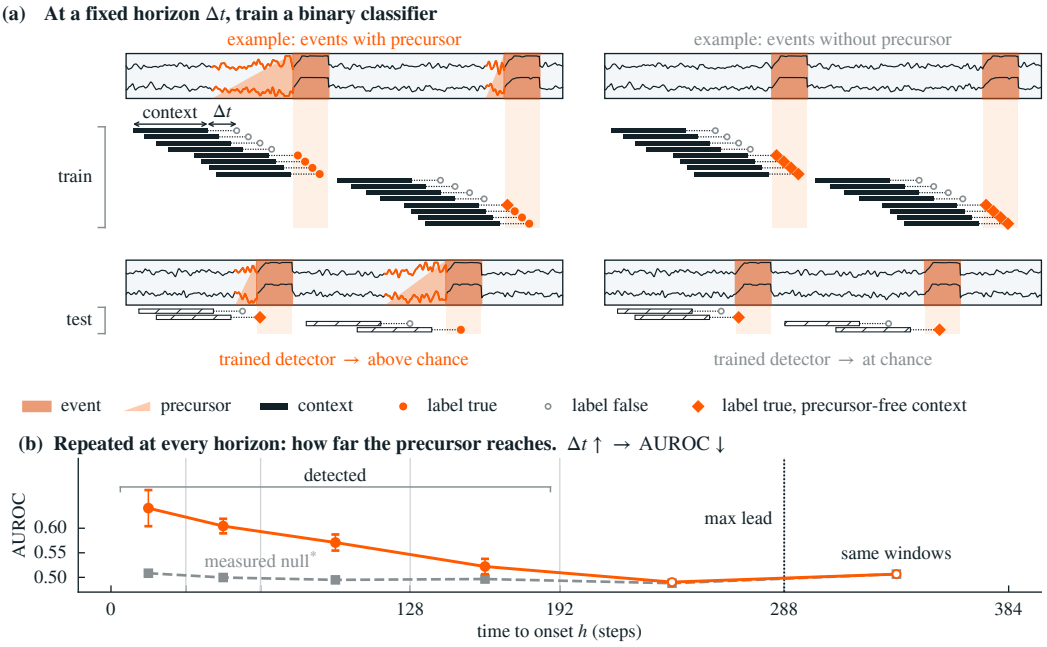


Figure 2: The Precursor Check. (a) Every context ends strictly before its event (Table 9). Training and test windows come from *different episodes*, never from one recording divided by event, because a precursor can outlast the gap between two events and would then reach across the split; the splits we report are entity-disjoint, by station, well or run, which is coarser again. On the synthetic corpus each episode carries exactly one event and contributes one context, never slid; real corpora are cut at a stride (subsection 3.5). **(b)** AUROC of the reference detector separating event-bearing from event-free episodes, with positives grouped by time to onset h, so a value is unitless and 0.5 is chance. The lower curve is the same detector scored on precursor-free twins of those episodes and is this panel's chance reference*. Bars are the spread over five entity-disjoint resamples (Table 14), so they measure sensitivity to the split and nothing else; the two curves separate by more than that spread up to $h = 192$ and not beyond, which is where this corpus's precursor stops being detectable. *Synthetic corpus only (subsection E.8).

3.5 PRECURSORBENCH

An event-forecasting number means nothing on a corpus that has not been shown to carry a precursor, or on one where something other than the series predicts the label. PrecursorBench applies the check of subsection 3.4 as an admission rule and publishes what survives, so the benchmark's own construction is the evidence that its scores are about the series.

Task. A corpus is a set of *entities*, each a multivariate series carrying event onsets. An instance is a context $x_{1:t}$ from one entity together with a horizon Δt, and its label is the event indicator at $t + \Delta t$, so a model estimates that probability directly. Horizons come from a fixed grid, the context length is fixed per corpus, and an onset must fall strictly after the context, which makes the label a statement about an unseen window.

Metric. We report *h-AUROC*: AUROC within each horizon stratum, averaged over horizons. Stratifying is the point rather than a refinement. Pooled, Δt is a free variable a scorer can rank on and prevalence varies with it; within a stratum it is a constant and can order nothing. Because the sampler draws each context's offset independently of its label, chance is 0.5 by construction rather than by correction. Secondary: h-AUPRC on the same strata, and h-AUROC against Δt, whose step-down locates a corpus's *lead time*, the horizon beyond which no detector on it reads above chance.

Integrity. Five properties are checked on measurement, not asserted: entity-disjoint *splits*; *horizon balancing*, which equalises the positive rate across strata and so removes the clock as a feature; *precursor validation*, which requires a reference detector , refit independently at every horizon, with Δt never an input , to clear chance somewhere and then to *decay* towards it, so a corpus with no detectable precursor cannot be entered and then blamed on the model; *null controls*, blind instruments reading the horizon, the context's end time and the time since the previous event and nothing else, read two-sided since an instrument below chance leaks as much as one above it; and bounded *leakage*, by defining events on a channel withheld from the model.

Table 1: PrecursorBench, the four corpora this paper reports, scored through the benchmark's own CV-ridge readout on the released entity-disjoint splits. The h-AUROC column is Saturn at 1M parameters (the golden-saturn configuration); Δ persistence is the margin over a last-value detector on the same rows, which is what the model contributed rather than what the corpus made easy. Entities are each corpus's own disjoint units: gauges, buoys, turbines, experiments, shear blocks, generated domains. The split is over them, never over windows. Test windows are counted as released, summed over the horizon grid. Every blind instrument on every admitted row sits within 0.05 of chance; the benchmark release carries the rejected candidates and the per-instrument readings. Generated at $\Delta t \geq 288$ carries no precursor by construction, which is what makes the null band a null rather than a convention.

corpus	entities	test windows	h-AUROC	Δ persistence
recorded				
wind tunnel	33	12594	0.8126	+0.0347
river reaches, discharge	41	5840	0.7417	+0.0155
wind farm SCADA	21	4694	0.6226	+0.0430
generated, precursor injected at a known lead				
switching device	150	13778	0.6937	+0.1955
recorded, mean			0.7256	
generated, mean			0.6937	

A corpus is admitted when precursor validation clears, the detection curve decays with Δt, and every blind instrument stays within 0.05 of chance, read two-sided. Ten corpora pass. This paper reports four of them (Table 1), three recorded and one generated, the subset whose full blind-instrument battery we hold; Appendix E lists what each rejected candidate failed on, which is the part of a benchmark that usually goes unpublished and is the most informative about how easily this task is got wrong.

The generated half is not padding. On a recorded corpus the precursor is argued for; on a generated one it is injected at a known lead, so the same protocol reads a quantity whose ground truth is available. The two kinds separate the baselines sharply. A last-value detector reaches 0.6946 on the recorded corpora and 0.4982 on the generated one, chance to three decimals, because a recorded onset is preceded by its own autocorrelation and an injected one is not. Any benchmark built only from recorded systems is therefore partly measuring persistence, and cannot say how much.

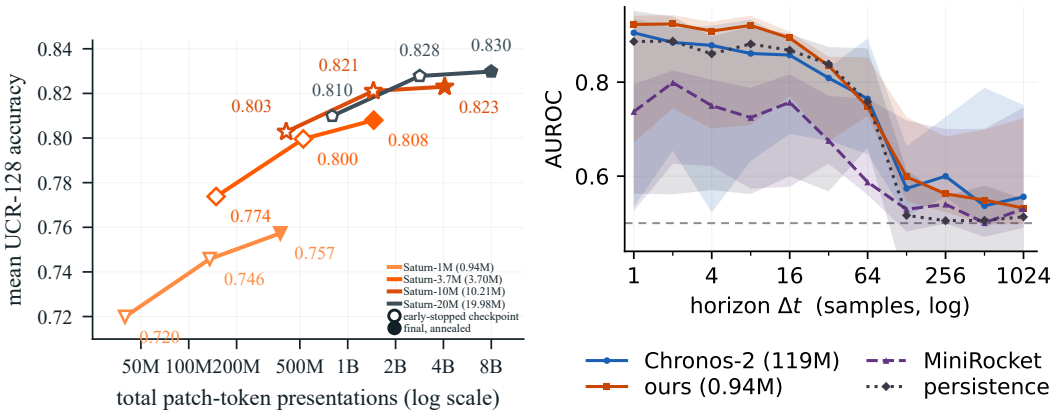
The simulator sits beside the real corpora as a positive control and is not counted as real. Two caveats bound what the rest establishes, both properties of the corpora rather than of the protocol: the bearing rig's entities are snapshots from one machine, so its entity-disjoint split is weaker than the others'; and the cluster-telemetry and bearing corpora were measured from a preprocessed store no longer available, so re-scoring those rows means rebuilding them from their public sources.

4 RESULTS

Each size contributes one checkpoint, pretrained on the synthetic corpus's `pretrain` split with its labels withheld and then read out frozen on all three benchmarks; no weights change between the columns of Table 2. Classification rises with size under that frozen readout: UCR-128 mean accuracy is 0.7768 at 0.9M parameters and 0.8297 at 20.0M. The representation supplies that rise, not the architecture or the probe: each pretrained encoder is paired with a random-initialisation twin, architecture and readout held fixed, and pretrained minus twin is $+0.119$ at Saturn-1M and $+0.156$ at Saturn-20M on UCR-128, ahead on 118 of the 128 datasets (Appendix G). Probed this way Saturn is competitive with the strongest published encoders on the UCR and UEA archives (Dau et al., 2019; Bagnall et al., 2018) (Table 20), without having seen a real series.

On PrecursorBench's four corpora (Table 1) no blind instrument reaches 0.05 from chance, and Saturn reads 0.7176 horizon-averaged AUROC against Chronos-2's 0.7114 at more than ten times the parameters, a margin a fifth of the bootstrap interval. Scale helps where a last-value detector cannot: on the generated corpus, where that detector is at chance, the encoder goes from 0.694 to 0.745 between 0.9M and 10.2M, while on the recorded corpora it does not. `usgs_reach` is the clearest case, reading 0.742, 0.694 and 0.685 across the three sizes against a persistence floor of 0.726, the same readout given only the last observed value.

Of seven design choices measured against matched single-knob controls, two move the frozen trunk and five do not (Table 6); the largest is the augmentation-contrastive term. Because the across-dataset



(a) classification (b) event forecasting

Figure 3: The two tasks one encoder is asked to do. *(a)* Fixed-layer accuracy on UCR-128 against pretraining tokens presented, at four encoder sizes of one recipe on one corpus; hollow markers are read before the learning rate anneals, filled ones after. The token axis repeats a fixed corpus, so it is a budget axis and not a measure of unique data. *(b)* AUROC against the horizon on a recorded corpus, for the encoder and for a last-value detector, with the detector carrying most of the score at short horizons.

Table 2: One frozen encoder, both tasks. Each row is a single checkpoint read out frozen. Classification is mean accuracy at one intermediate layer; UEA-27 is the panel the published anchors score. Event forecasting is horizon-averaged AUROC over the four admitted corpora, split real and synthetic (Table 1).

		classification		event forecasting	
encoder	params	UCR-128	UEA-27	real	synthetic
Saturn-1M	0.9M	0.7697		0.726	0.694
Saturn-10M	10.2M	0.8230	0.7319	0.707	0.745
Saturn-20M	20.0M	0.8298	0.7223	0.704	0.741
Chronos-2	119.5M	0.8244	0.7234	0.706	0.727
Chronos-Bolt	205.3M			0.697	0.701
MiniRocket				0.645	0.681
persistence				0.695	0.498

standard error is larger than any single lever, these are paired verdicts dataset by dataset and not a ranking of accuracy levels.

How far these results reach. Every cell is one seed, so a difference smaller than the across-dataset standard error of 0.031 is not one we can defend. The probe always reads an intermediate layer rather than the last (Appendix H), and a different layer suits different datasets. The pretraining corpus is one we generated, so a family it does not span is a gap in the evidence rather than a property of the method. Appendix B takes the remaining questions in turn, the scaling reversal on `usgs_reach` among them.

5 Conclusion

Posing event forecasting as horizon-conditioned classification puts the present and the future in one form, and Saturn learns that representation with a joint-embedding predictive objective regularised by its contrastive term rather than by a prior on the latent distribution. Read out frozen, Saturn matches the strongest foundation model on that benchmark at a fraction of its parameter count, from generated pretraining episodes alone.

AI USE STATEMENT

We used generative AI tools, principally Anthropic's Claude, throughout this work: to write and review code for the objective, the model and the evaluation harness, to design experiments and argue about methodology, to produce figures, and to draft and edit this manuscript and its bibliography. The pretraining corpus is the output of a stochastic dynamical-systems simulator we wrote with that assistance, not a sample from a generative model.

The authors have reviewed all AI-assisted work and take full responsibility for the content of this paper, including its text, its claims and its artifacts.

REFERENCES

James R. Anderson, Kevin C. Cain, and Richard D. Gelber. Analysis of survival by tumor response. *Journal of Clinical Oncology*, 1(11):710–719, 1983. doi: 10.1200/JCO.1983.1.11.710.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15619–15629, 2023. doi: 10.1109/CVPR52729.2023.01499.

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

Andreas Auer, Daniel Klotz, Sebastian Böck, and Sepp Hochreiter. Pre-trained forecasting models: Strong zero-shot feature extractors for time series classification. *arXiv preprint arXiv:2510.26777*, 2025. NeurIPS 2025 Workshop on Recent Advances in Time Series Foundation Models (BERT2S).

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

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

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=QaCCuDfBk2.

Florian Bordes, Randall Balestriero, Quentin Garrido, Adrien Bardes, and Pascal Vincent. Guillotine regularization: Why removing layers is needed to improve generalization in self-supervised learning. *Transactions on Machine Learning Research*, 2023.

Thomas M. Bury, R. I. Sujith, Induja Pavithran, Marten Scheffer, Timothy M. Lenton, Madhur Anand, and Chris T. Bauch. Deep learning for early warning signals of tipping points. *Proceedings of the National Academy of Sciences*, 118(39):e2106140118, 2021. doi: 10.1073/pnas.2106140118.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021. doi: 10.1109/ICCV48922.2021.00951.

Alexander Chemeris, Ming Jin, and Randall Balestriero. LeNEPA: No-augmentation next-latent prediction for time-series representation learning. *arXiv preprint arXiv:2607.00958*, 2026. KDD 2026 Workshop on Mining and Learning from Time Series (MILETS).

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019. doi: 10.1109/JAS.2019.1911747.

Angus Dempster, Daniel F. Schmidt, and Geoffrey I. Webb. MiniRocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 248–257. ACM, 2021. doi: 10.1145/3447548.3467231.

Dang-Khoa Do, Patrick Rockenschaub, Sebastian Daniel Boie, Oliver Kumpf, Hans-Dieter Volk, Felix Balzer, Falk von Dincklage, and Gregor Lichtner. The impact of evaluation strategy on sepsis prediction model performance metrics in intensive care data: retrospective cohort study. *Journal of Medical Internet Research*, 28:e72083, 2026. doi: 10.2196/72083.

Katrina Drozdov, Ravid Shwartz-Ziv, and Yann LeCun. Video representation learning with joint-embedding predictive architectures. *arXiv preprint arXiv:2412.10925*, 2024.

Amir El-Ghoussani, Michele De Vita, Ronald Naumann, and Vasileios Belagiannis. Time-series foundation model embeddings for remaining useful life estimation. In *European Signal Processing Conference (EUSIPCO)*, 2026. arXiv:2606.11990.

Sofiane Ennadir, Siavash Golkar, and Leopoldo Sarra. Joint embeddings go temporal. *arXiv preprint arXiv:2509.25449*, 2025. Accepted at the NeurIPS 2024 Workshop on Time Series in the Age of Large Models.

Vasilii Feofanov, Songkang Wen, Shifeng Xie, Simon Roschmann, Marius Alonso, Hongbo Guo, Romain Ilbert, Malik Tiomoko, Quentin Bouniot, Zeynep Akata, Lujia Pan, Jianfeng Zhang, and Ievgen Redko. Mantis: Lightweight foundation model for time series classification. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026a. arXiv:2502.15637.

Vasilii Feofanov, Songkang Wen, Jianfeng Zhang, Lujia Pan, and Ievgen Redko. MantisV2: Closing the zero-shot gap in time series classification with synthetic data and test-time strategies. *arXiv preprint arXiv:2602.17868*, 2026b.

Juan L. Gamella, Jonas Peters, and Peter Bühlmann. Causal chambers as a real-world physical testbed for AI methodology. *Nature Machine Intelligence*, 7(1):107–118, 2025. doi: 10.1038/s42256-024-00964-x.

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

Michael F. Gensheimer and Balasubramanian Narasimhan. A scalable discrete-time survival model for neural networks. *PeerJ*, 7:e6257, 2019. doi: 10.7717/peerj.6257. arXiv:1805.00917.

Sander Greenland and Duncan C. Thomas. On the need for the rare disease assumption in case-control studies. *American Journal of Epidemiology*, 116(3):547–553, 1982. doi: 10.1093/oxfordjournals.aje.a113439.

Yanan He, Yunshi Wen, Xin Wang, and Tengfei Ma. SC-JEPA: Stabilizing latent predictive learning for time-series anomaly prediction. *arXiv preprint arXiv:2602.04643*, 2026. To appear, SIAM International Conference on Data Mining (SDM) 2026.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations (ICLR)*, 2023.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, 2023. doi: 10.1016/j.patter.2023.100804.

Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7194–7201, 2022a. doi: 10.1609/aaai.v36i7.20680.

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 *International Conference on Learning Representations (ICLR)*, 2022b. URL `https://openreview.net/forum?id=cGDAkQo1C0p`.

Carl H. Lubba, Sarab S. Sethi, Philip Knaute, Simon R. Schultz, Ben D. Fulcher, and Nick S. Jones. catch22: CAnonical Time-series CHaracteristics: Selected through highly comparative time-series analysis. *Data Mining and Knowledge Discovery*, 33(6):1821–1852, 2019. doi: 10.1007/s10618-019-00647-x.

Jay H. Lubin and Mitchell H. Gail. Biased selection of controls for case-control analyses of cohort studies. *Biometrics*, 40(1):63–75, 1984. doi: 10.2307/2530744.

Riccardo Lunelli, Angus Nicolson, Samuel Martin Pröll, Sebastian Johannes Reinstadler, Axel Bauer, and Clemens Dlaska. BenchECG and xECG: a benchmark and baseline for ECG foundation models. *arXiv preprint arXiv:2509.10151*, 2025.

Gabriel Meseguer-Brocal, Yuexuan Kong, and Romain Hennequin. CoJEPA: Combining contrastive learning and JEPA for global-local music representations. *arXiv preprint arXiv:2608.30974*, 2026.

Shentong Mo and Shengbang Tong. Connecting joint-embedding predictive architecture with contrastive self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 37, pages 2348–2377, 2024.

Michael Moor, Bastian Rieck, Max Horn, Catherine R. Jutzeler, and Karsten Borgwardt. Early prediction of sepsis in the ICU using machine learning: a systematic review. *Frontiers in Medicine*, 8:607952, 2021. doi: 10.3389/fmed.2021.607952.

Jonas Petersen, Gian-Alessandro Lombardi, Riccardo Maggioni, Camilla Mazzoleni, Federico Martelli, and Philipp Petersen. HEPA: A self-supervised horizon-conditioned event predictive architecture for time series. *arXiv preprint arXiv:2605.11130*, 2026.

Daniele Proverbio, Françoise Kemp, Stefano Magni, and Jorge Gonçalves. Performance of early warning signals for disease re-emergence: a case study on COVID-19 data. *PLOS Computational Biology*, 18(3):e1009958, 2022. doi: 10.1371/journal.pcbi.1009958.

Matthew A. Reyna, Christopher S. Josef, Russell Jeter, Supreeth P. Shashikumar, M. Brandon Westover, Shamim Nemati, Gari D. Clifford, and Ashish Sharma. Early prediction of sepsis from clinical data: the PhysioNet/Computing in Cardiology Challenge 2019. *Critical Care Medicine*, 48 (2):210–217, 2020. doi: 10.1097/CCM.0000000000004145.

Patrick Schäfer and Ulf Leser. TEASER: early and accurate time series classification. *Data Mining and Knowledge Discovery*, 34(5):1336–1362, 2020. doi: 10.1007/s10618-020-00690-z. arXiv:1908.03405.

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. doi: 10.1038/nature08227.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Samy Suissa. Immortal time bias in pharmacoepidemiology. *American Journal of Epidemiology*, 167 (4):492–499, 2008. doi: 10.1093/aje/kwm324.

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

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *PMLR*, pages 9929–9939, 2020.

Yilin Wang, Yifei Yu, Kong Sun, Peixuan Lei, Yuxuan Zhang, Enrico Zio, Aiguo Xia, and Yuanxiang Li. RmGPT: A foundation model with generative pretrained transformer for fault diagnosis and prognosis in rotating machinery. *IEEE Internet of Things Journal*, 12(20):41562–41573, 2025. doi: 10.1109/JIOT.2025.3580823. arXiv:2409.17604.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 53140–53164, 2024.

Renjie Wu and Eamonn J. Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering*, 35 (3):2421–2429, 2023. doi: 10.1109/TKDE.2021.3112126. arXiv:2009.13807.

Shifeng Xie, Vasilii Feofanov, Jianfeng Zhang, Themis Palpanas, and Ievgen Redko. CauKer: Classification time series foundation models can be pretrained on synthetic data. In *International Conference on Learning Representations (ICLR)*, 2026.

Hugo Yèche, Alizée Pace, Gunnar Rätsch, and Rita Kuznetsova. Temporal label smoothing for early event prediction. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 39913–39938, 2023.

William Yuan, Brett K. Beaulieu-Jones, Kun-Hsing Yu, Scott L. Lipnick, Nathan Palmer, Joseph Loscalzo, Tianxi Cai, and Isaac S. Kohane. Temporal bias in case-control design: preventing reliable predictions of the future. *Nature Communications*, 12(1):1107, 2021. doi: 10.1038/s41467-021-21390-2.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. TS2Vec: Towards universal representation of time series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8980–8987, 2022. doi: 10.1609/aaai.v36i8.20881.

John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLOS Medicine*, 15(11):e1002683, 2018. doi: 10.1371/journal.pmed.1002683.

Sinong Zhao, Wenrui Wang, Hongzuo Xu, Zhaoyang Yu, Qingsong Wen, Gang Wang, Xiaoguang Liu, and Guansong Pang. Abnormality forecasting: Time series anomaly prediction via future context modeling. *arXiv preprint arXiv:2410.12206*, 2024.

A Architecture and hyperparameters

Table 4 lists every setting of the canonical recipe with the reason for it where one exists; the subsections below explain the components that need more than a table row. Every value in this appendix is read from the recipe file and the code it drives, and every parameter count comes from building the model and counting.

A.1 TOKENISER

We tokenise channel-independently with the Token Generator Unit of Mantis (Feofanov et al., 2026a;b). One token is one (time-patch, channel) pair with patch length $P = 16$, every weight is shared across channels, and no channel-identity embedding is added, so one parameter set serves any channel count and the encoder is invariant to the order of the channels by construction. The unit has three branches, each producing a d-dimensional vector per token.

- *Signal.* Each patch of each channel is first normalised by its own mean and standard deviation, so this branch sees shape and not level. A one-dimensional convolution with kernel 41 and stride 16 then maps the normalised series to one vector per patch, followed by LayerNorm. The kernel is wider than the patch, so each token also sees part of its neighbouring patches; the padding is symmetric, 12 steps before and 13 after.
- *Difference.* The same construction, with its own weights, applied to the first difference of the patch-normalised series.
- *Statistics.* The mean and standard deviation that the patch normalisation removed are each encoded by a multi-scale scalar encoder (Feofanov et al., 2026a): eight affine maps of the scalar at fixed, log-spaced scales from 10^{-3} to 10^{4}, blended by a softmax over how close the scalar lies to each scale, followed by LayerNorm. The two encodings are summed.

The three vectors are concatenated and fused by a linear map from $3d$ to d and a LayerNorm. The signal and difference branches therefore describe the shape of each patch, and the statistics branch restores its level and spread, in separate subspaces of one token. Tokens are laid out patch-major and channel-minor, and each carries its patch index and its channel index. Channels are zero-padded to the widest episode in the batch and padded tokens enter attention only through the key-padding mask.

Masking removes tokens after tokenisation, and the 41-step kernel reaches 12 to 13 steps into a neighbouring patch, so a visible token beside a masked block would carry part of that block's raw signal. A causal kernel would close that path for the tail mask but not for the interior block, so the recipe closes it directly instead, zeroing the input under masked patches in the context pass before tokenisation. What the leak was worth is a measured null: at Saturn-1M, leaving the context unzeroed moves UCR-128 by $+0.0008$ ($p = 0.239$), inside the three-seed spread of 0.0027. The window statistics below are the part of the path that is still open.

A.2 NORMALISATION

Three normalisations act in sequence, each at a different scope. At load, every episode is z-scored per channel over its whole length, which puts corpora recorded in different units on one scale. At the encoder input, every context window is z-scored per channel over the window (RevIN, Kim et al., 2022b); the removed mean and standard deviation are compressed as $\big(\mathrm{sign}(\mu)\log(1+|\mu|),\ \log(1+\sigma)\big)$, mapped by a linear layer shared across channels, averaged over the real channels, normalised and halved, and the result is added to every output token and to the pooled vector. The encoder thus sees a level-free series and can still read the level of the window it was given. Inside the tokeniser, finally, every patch is normalised by its own statistics as described above, which is the per-(channel, patch) normalisation of Mantis. The window statistics are computed over the whole context, masked patches included, so the level of a masked block still reaches the context through them: zeroing the input under masked patches closes the tokeniser's kernel path and leaves this one open.

A.3 CONTEXT ENCODER

The encoder is a pre-norm transformer of eight blocks at every size. Each block applies LayerNorm, multi-head self-attention with rotary position encoding (Su et al., 2024), a residual connection, LayerNorm, a two-layer feed-forward network of inner width $4d$ with a GELU, and a second residual. Dropout of 0.1 acts on the attention output, inside the feed-forward network and on each residual branch. A final LayerNorm closes the stack. Attention is bidirectional over the whole token grid in both passes: no time mask is applied, only the key-padding mask. A learned `[CLS]` token is prepended to the grid; it is the vector the contrastive term reads, while the prediction term and every downstream readout use the token grid (Table 4).

Table 3 gives the four sizes. Depth, patch size, tokeniser and normalisation are identical across them; width, head count and feed-forward width change.

Table 3: The four sizes, built and counted. *Encoder* is every trainable parameter of the saved encoder, tokeniser included; *tok.* is the tokeniser alone. These counts, and the parameter axis of Figure 3, are the encoders as built. *Predictor* and the *contrastive projector* are discarded after pretraining. The predictor is 12.1% of the encoder at every size.

model	d	heads	d/h	d_{ff}	encoder	tok.	predictor	contr. projector
Saturn-1M	96	2	48	384	935,520	40,032	112,800	18,720
Saturn-3.7M	192	4	48	768	3,695,808	135,360	446,784	74,304
Saturn-10M	320	5	64	1280	10,214,720	348,480	1,236,160	205,760
Saturn-20M	448	7	64	1792	19,977,664	659,904	2,418,752	402,752

A.4 Positional encoding

The encoder encodes position rotationally. Rotary embeddings rotate each consecutive pair of head dimensions by an angle proportional to position, so that the attention logit between two tokens depends on them only through their separation. The rotation is driven by the *patch* index and not by the index in the token sequence: under channel-independent tokenisation several tokens share one time index, and rotating by sequence position would encode channel identity as though it were time.

The predictor does not share this. It adds an absolute sinusoidal encoding of the patch index, plus a second sinusoidal encoding of the channel index offset by $10{,}000$. The offset breaks a symmetry rather than decorating one: an unoffset sum $\mathrm{pe}(t) + \mathrm{pe}(c)$ is symmetric in its two arguments, so tokens at $(t{=}3, c{=}7)$ and $(t{=}7, c{=}3)$ would receive identical encodings and collide.

A.5 Masking

For each context the recipe masks whole patches, and every channel at a masked patch, so a mask is a set of time-patches. A context of L steps holds $N = \lceil L/16 \rceil$ patches. The recipe masks $b = \min(\mathrm{round}(0.6N),\, N-1)$ of them in total: the last $\min(8,\, b-1)$ patches always, which is the tail mask, and the remaining budget as one contiguous block placed uniformly at random among the patches before the tail, which is the main mask. The two blocks can touch. The tail lives inside the 0.6 budget and not on top of it. At the full context of 512 steps, $N = 32$ and $b = 19$: eight tail patches and an eleven-patch block, 59.4% of the grid.

Every context is exactly 512 steps, so this is the mask of every training sample: 19 of 32 patches, the last eight and an eleven-patch block, with the block's position the only random draw. A context is cut from an episode at an end point drawn uniformly over the positions that leave 512 steps before it, so training and evaluation see windows of one length.

A.6 Predictor and projectors

The predictor is a two-block transformer over the full token grid, pre-norm, at the encoder's width and head count, with a feed-forward width equal to d rather than $4d$ and no dropout. It replaces every masked token by one learnable mask token, initialised from a normal distribution scaled by 0.02; the substitution routes zero gradient into the encoder's outputs at masked positions, so nothing the encoder never saw can reach the loss. The predictor applies no time mask, so information flows from any visible token to any masked one, as in I-JEPA even when its encoder is restricted (Assran et al., 2023). It is small by depth and feed-forward width rather than by token width: two blocks against eight, at a feed-forward ratio of one against four, which puts it at 12.1% of the encoder (Table 3).

One projector sits between the encoder and a loss, and it is discarded with the predictor. It is a two-layer MLP: a linear map from d to d without bias, batch normalisation, a GELU and a second linear map from d to d. No projector stands on either side of $\mathcal{L}_1$: the distance of Equation 1 is taken between the predictor's output and the encoder's target directly, ℓ_2-normalised at the masked positions and with the target detached, which is the distance the code computes. The *contrastive* projector maps the `[CLS]` output of each augmented view before $\mathcal{L}_{\mathrm{NCE}}$ is computed, which is the SimCLR placement (Chen et al., 2020): the invariance to cropping is imposed on the projection, and the encoder is free to keep what the augmentation destroys. The saved checkpoint contains the

encoder alone, which is what lets every downstream loader read it without knowing which objective produced it.

A.7 READOUTS

Classification. We freeze the encoder and embed the whole window. The vector every readout reads is the *mean of the grid tokens*, with the `[CLS]` excluded. We fit a logistic-regression probe on those standardised pooled features, with L-BFGS and at most 500 iterations, on each dataset's canonical split. The probe reads one intermediate layer of the encoder, and that layer is fixed for the architecture rather than chosen per dataset: choosing it per dataset on the split being reported would be selection on the test set. Reading an intermediate layer rather than the last is guillotine regularisation (Bordes et al., 2023), which is where the invariance the pretraining objective enforces stops being useful to a downstream probe. Appendix H fixes the layer, prices the alternatives, and records which rows are read at the committed block and which remain on the weaker panel-mean rule.

Event forecasting. We freeze the encoder and read it out *joint-last* over a context of $L = 512$ steps, taking each channel's last-patch token and concatenating them (subsection 3.1), then project into a shared 768-dimensional space by a fixed random orthogonal map. The socket is 768 because projecting up is free and projecting down is not: an isometric embedding into the widest entry leaves every narrower model's features spanning exactly the dimensions they occupied, with the padded directions in the null space of those features, whereas projecting down to the narrowest entry would make every wider model pay a truncation cost that no narrow model pays. It is also where the comparison class sits, with Chronos-2 at 768 (Ansari et al., 2025).

The head is the bilinear probe of Equation 2: LayerNorm on z, a 32×768 matrix W, and a 32-vector for the horizon-dependent intercept, so $26{,}144$ parameters in total. The horizon basis φ is fixed and parameter-free: Δt scaled to $[0, 1]$, an explicit linear term and bias so that a smooth monotone trend costs no frequencies, and fifteen sine-cosine pairs on a geometric ladder from one cycle to 128 cycles across a horizon grid of 384. The ladder is capped rather than octave-spaced because an octave ladder would run to 2^{14} cycles on this grid, six octaves past its Nyquist limit, giving the head aliased components to overfit instead of a smooth function of the horizon.

At any fixed horizon the head is a linear functional of z, which is the property that makes its score attributable to the representation rather than to the readout, and it is the property linear probing is the canonical instrument for (Caron et al., 2021). At $26{,}144$ parameters it is 0.1% of the widest rung, so no entry can be accused of having its score manufactured by its head. Two further heads exist and neither is an entry. A horizon-unconditioned linear head at $2{,}305$ parameters is the degeneracy ablation: a model whose conditioned and unconditioned scores agree is answering whether a precursor is present, not whether onset falls in the queried window. A horizon-modulated nonlinear head at $249{,}345$ parameters sits an order of magnitude inside the few-hundred-thousand ceiling that V-JEPA 2 sets for a nonlinear probe (Assran et al., 2025; Bardes et al., 2024), with its hidden width pinned at 256 so that widening the socket does not silently widen the head.

A.8 EVERY SETTING, AND WHY

Table 4 is the recipe in full. *Why* gives the reason where one exists and says so where the value was inherited without a test; a blank reason would read as a justification we do not have.

Table 4: The canonical recipe, every setting. Sizes in Table 3; d is the width.

component	setting	why
Data		
corpus	the synthetic corpus, labels withheld	subsection 3.3
episode normalisation	z-score per channel over the whole episode, at load	puts corpora in different units on one scale
context	exactly 512 steps, ending at a uniformly drawn cut; 40 cuts per episode, drawn once	the length every evaluation reads

component	setting	why
channels	native count, capped at 43; zero-padded to the widest in the batch and masked	any channel count, one set of weights
batch	128 contexts per GPU, grouped by channel width; the global batch is 128 times the GPU count	inherited; $\mathcal{L}_{\mathrm{NCE}}$ is computed per GPU, so each anchor has 127 negatives at any GPU count
Tokeniser		
unit	Mantis Token Generator Unit, patch 16, weights shared across channels, no channel embedding	channel-count-agnostic and permutation-invariant in channels
signal branch	per-patch z-score, then Conv1d($1\to d$, kernel 41, stride 16), non-causal (symmetric) padding, LayerNorm	shape without level; MantisV2 reports gains up to kernel 41 at patch 16
difference branch	the same on the first difference, own weights	slope information at no cost in tokens
statistics branch	patch mean and s.d. through two multi-scale scalar encoders (8 fixed scales, 10^{-3} to 10^{4}), LayerNorm	restores the level and spread the patch z-score removed, robust across magnitudes
fusion	concatenate, Linear($3d\to d$), LayerNorm	one token per (patch, channel)
Encoder		
input normalisation	RevIN per channel over the context; signed-log mean and $\log(1+\sigma)$ mapped to a d-vector, halved and added to every output token	level-free input without losing the window's level
blocks	8, pre-norm; multi-head attention, then MLP $d\to 4d\to d$ with GELU	the Saturn-3.7M is the Causal ViT-XS shape of LeNEPA (depth 8, 4 heads, MLP ratio 4) (Chemeris et al., 2026); the other sizes widen it at fixed depth and ratio, so a cut after block k is the same fraction of the encoder at every size
attention	bidirectional over the grid, key-padding mask only	the target pass needs the full grid; a causal context pass is a tested null (Table 6)
position	rotary, on the patch index	logits depend on temporal separation only; channel index not encoded
dropout	0.1 on attention output, MLP and residuals	inherited, untested
`[CLS]` token	learned, prepended; read only by the contrastive term	the two loss terms shape different vectors
Masking and prediction		
mask	whole patches across all channels, at fixed size and no ratio: the last 8 always, plus one contiguous 8-patch block at a random position before them, so 16 of 32 on a full 512-step context	tail: part of every target lies beyond all visible input; block: the rest is in-filling
context pass	encoder with gradients; masked tokens removed from attention, and the normalised input zeroed under masked patches	keeps masked signal out of attention and out of the tokeniser's kernel; the RevIN window statistics are still taken over all steps, so they carry masked input
target pass	same encoder, evaluation mode, full grid, no gradient; no EMA	no second network to keep in step
predictor	2 pre-norm blocks at width d, encoder's head count, MLP ratio 1, dropout 0; learnable mask token; absolute sinusoidal patch and channel encodings	shallow so the encoder must carry the representation; dropout 0 as in I-JEPA
$\mathcal{L}_1$	ℓ_1 between the ℓ_2-normalised predictor output and the detached encoder target, mean over masked tokens	scale-free alignment
Contrastive term		
views	two random crop-resize views of the full context, crop fraction $c \sim \mathcal{U}(0, 0.2)$ drawn per sample, both views augmented	Mantis's single augmentation

component	setting	why
readout	`[CLS]` through the contrastive projector (same MLP shape)	SimCLR placement: invariance imposed on the projection, not the encoder
$\mathcal{L}_{\text{NCE}}$	InfoNCE, cosine similarity, $\tau = 0.2$, symmetric over the two views, in-batch negatives	negatives rule out the constant encoder (section 3.2); τ below
weighting	$\mathcal{L}_1 + 0.5\,\mathcal{L}_{\text{NCE}}$, i.e. $\lambda = 1/3$	inherited; a four-to-one rebalance is a tested null (Table 6)
off	latent-distribution prior, scale grounding	no prior by design; removing both together moved UCR-128 accuracy by -0.0015 and $+0.0010$ at 142 and 400 tokens per parameter, both inside noise
Optimisation		
optimiser	AdamW, $\beta = (0.9, 0.999)$, weight decay 0.01 on all parameters	inherited
learning rate	peak 3×10^{-4}; linear warmup over 2% of steps, then cosine to zero at the last step	inherited; intermediate checkpoints are not annealed
clipping	global norm 1.0 over encoder and predictor	inherited; the contrastive projector is outside it
precision	bf16 autocast, fp32 master weights, gradient checkpointing	memory at the widest size; identical maths
budget	400 tokens per parameter, where the learning rate reaches zero; checkpoints at 142 and 400 in *realised* tokens per parameter, saved when the tokens seen cross the mark; steps from the exact expected tokens per step of the corpus	tokens per parameter rather than epochs, so every size sees the same budget per parameter
seeds	42 for initialisation, cuts and batch order; masks from seed 0	inherited

Every checkpoint states its global batch and its *realised* tokens per parameter, the tokens actually seen divided by the encoder's parameter count, because a run's nominal budget and what it delivered can differ: one earlier run registered for 400 realised 478 through a stale tokens-per-step estimate. Table 5 lists the pre-recipe runs this appendix draws on.

Table 5: The checkpoints behind this appendix's layer and budget analyses, with the global batch each trained at and the tokens per parameter it realised. Red cells are not yet read back from their run logs. Every run in this table predates the recipe of Table 4 and differs from it as its row says, so none of them is a rung of Table 2.

checkpoint	where reported	global batch	realised tok/param
provisional Saturn-3.7M, first rung	Appendix H	128	**135.3**
3.7M budget curve, three rungs	subsection D.6	128	–
1.1M budget curve, five rungs	subsection D.6	64	41.7, 83.4, 125.1, 166.7, 208.2

A.9 Why these choices

Eight blocks at every size. The scaling question is asked by cutting the encoder at a fixed depth and reading out from the cut. If depth varied with width, a cut after the fifth block would mean a different fraction of the encoder at each size and the curves would not be comparable. Holding depth fixed and letting width carry the parameter budget makes the cut index the same axis at every size. The cost is that the head dimension is not constant, 48 at the two smaller sizes and 64 at the two larger, and we have not measured what that contributes.

Channel-independent tokenisation. It is the only scheme whose weights do not depend on the channel count, and a foundation encoder that cannot accept an unseen channel count is not one. It also preserves the channel-by-time grid that the masking, the attention mask and the per-channel diagnostics all index into. We decline a channel-embedding table so that comparisons across corpora of differing width isolate data scope rather than the channel interface.

Table 6: Every lever measured on the frozen trunk, including the ones that did nothing. Δ is paired across datasets against a matched control differing in the named knob alone, p a two-sided Wilcoxon signed-rank test on those pairs, *wins* the datasets helped. UCR-25 is the development subset; UEA is our own 30-dataset panel scoring 21–22, not the published UEA-27. Every cell is seed 42, so no error bar here is a seed error bar and the across-dataset standard error is 0.031.

	UCR-25			UEA (ours, 21–22 scored)			
Lever	Δ	wins	p	Δ	wins	p	verdict
Augmentation-contrastive term	+0.025	22/25	0.0003	+0.015	14/21	0.091	established
Width, 3.7M $\rightarrow$ 7.9M parameters	+0.009	–	0.0001	+0.007	–	0.033	established
Bidirectional attention	+0.006	17/25	0.58	–	–	–	null
Alternating time/channel attention	+0.000	–	–	–	–	–	null
Multi-scale token embedding	+0.002	–	0.485	–	–	–	null
Three-branch tokenizer	−0.004	9/25	0.77	–	–	–	null
Loss rebalancing, contrastive:predictive 4:1	−0.001	–	0.53	–	–	–	null

The width ladder. Four widths put four points on a scaling axis, roughly tripling the parameter count per step. The widest is the one to drop if compute runs short, because the axis survives on three points and it is the most expensive; a dropped size has to be stated rather than absorbed.

The temperature. It is the one loss constant with a recorded argument of its own: at our measured separation the gradient on the positive is 0.36 at $\tau = 0.2$ and 0.005 at $\tau = 0.1$, so a sharper temperature helps when separation is poor and hurts once it is good. For contrast, the leading contrastive time-series model trains at batch 256, learning rate 2×10^{-3} and $\tau = 0.1$, so our operating point is not theirs.

A.10 What we searched, and what we did not

There is no hyperparameter search behind this recipe. No learning rate, warmup, batch size, weight decay, dropout, mask ratio, tail length, patch size, kernel width or loss weight in Table 4 was swept for this model, and neither were the encoder's depth and width or the predictor's depth and width: the sizes in Table 3 were set by hand to reach parameter targets. Depth has never been ablated at any size. The eight-block choice is an argument about comparability, not a measurement that eight is better than four or sixteen, and the two-block predictor follows I-JEPA rather than a sweep of our own. Most of the recipe was carried forward from earlier runs whose single-knob measurements are in Table 6, and the honest reading is that these are inherited values and not the outcome of a search conducted here.

What does exist is a set of single-knob measurements on the frozen trunk, each paired across datasets against a control differing in that knob alone, on a single seed (Table 6). Two levers are established on both panels and five are null, and the one significance caveat that has to travel with the table is the contrastive term on the multivariate panel, where the direction agrees with the univariate result at two-thirds the size but $p = 0.091$ does not establish it. Four pretext variants span 0.001 between them, which is one number inside noise. Every cell is a single seed, so no error bar among them is a seed error bar, and the across-dataset standard error of 0.031 is larger than any single lever measured, which is why the verdicts rest on paired across-dataset tests and never on replication.

One ablation of the objective is not a null. Removing the contrastive term entirely collapses the encoder on the same corpus and budget, with mean pairwise cosine rising to 0.828, the contrastive loss reaching 0.995 of $\log B$, and effective rank falling to 5.86. That is the control the anti-collapse argument of section 3 needs, and it is the one architectural component of the objective whose removal we have shown to be fatal rather than merely unhelpful.

The contrastive weight. The sweep below reports α, the coefficient the implementation applies as $\mathcal{L}_1 + \alpha\mathcal{L}_{\mathrm{NCE}}$, which is not the λ of Equation 1: that one is a convex mixture and cannot exceed 1, and the two are related by $\lambda = \alpha/(1+\alpha)$, so the shipped $\alpha = 0.5$ is $\lambda = 1/3$. A sweep at Saturn-1M puts the shipped value at the floor of a plateau rather than at an optimum. UCR-128 reads 0.7164 at a weight of 0.03, 0.7445 at 0.125, 0.7498 at the shipped $\lambda = 1/3$, 0.7505 at 1.0 and 0.7504 at 2.0, against a three-seed spread of 0.0027. Everything from a third upward is one number; the fall below

it is the partial collapse section 3.2 predicts, with effective rank at 5.06 where the plateau holds 11.2. The value is therefore not tuned, and nothing here rests on its being $1/3$ rather than 1.

B Discussion

Is the benchmark circular, admitting corpora with a detector and then scoring models on them? Admission asks whether *any* detector reads above chance, a property of the corpus; the leaderboard asks which model reads highest, a property of the models. A corpus that fails admission is not hard but uninformative. The rule does bound what the benchmark can hold, since a precursor too subtle for the reference detector is excluded along with the corpora that have none, and we accept that bound because the alternative, admitting corpora on faith, is what produces the leaderboards this paper argues are uninterpretable.

C The contrastive term as a regulariser

C.1 Setup

Fix a batch of B windows. For window i the encoder produces two embeddings, $z_i^{(1)} = f_\theta(A_1(X_i))$ and $z_i^{(2)} = f_\theta(A_2(X_i))$, where A_1 and A_2 are independent draws of the crop-resize augmentation. Write u_i and v_j for the ℓ_2-normalised first and second views, and $s_{ij} = u_i^\top v_j$ for their cosine similarity. The contrastive term of Equation 1 is then the cross-entropy of a softmax over row i at temperature τ,

$$\mathcal{L}_i^{\mathrm{NCE}} \;=\; -\log p_{ii}, \qquad p_{ij} \;=\; \frac{\exp(s_{ij}/\tau)}{\sum_{k=1}^{B} \exp(s_{ik}/\tau)}. \tag{3}$$

The sum runs over all B candidates, the positive $k = i$ included, which is what the implementation computes and what fixes the scale of everything below. The implementation also symmetrises the term over the two views, averaging Equation 3 with the same quantity computed down the column; every statement here holds for each direction separately and therefore for their average.

C.2 A constant encoder attains $\log B$

Proposition 1. Let f_θ be constant, $f_\theta(X) = c$ for every input, with $c \neq 0$. Then $\mathcal{L}_i^{\mathrm{NCE}} = \log B$ for every i and every $\tau > 0$.

Proof of Proposition 1. Normalising a constant gives $u_i = v_j = c/\|c\|$ for all i and j, so $s_{ij} = 1$ everywhere. Every logit in row i is then $1/\tau$, the softmax is uniform, $p_{ij} = 1/B$, and $-\log p_{ii} = \log B$. The step never used the value of the similarity, only that it is the same everywhere, so any encoder whose similarity row is flat attains the same value. This completes the proof.

The constant encoder is thus a global minimiser of the prediction term, where it scores zero, and it scores $\log B$ on the contrastive term. The two terms disagree about it, and that disagreement is the whole of the mechanism the body claims.

C.3 Among view-consistent encoders, $\log B$ is the maximum

Call an encoder *view-consistent* on a batch when $s_{ii} \geq s_{ij}$ for every i and j: two views of one window are at least as similar as two views of different windows. This is the property the contrastive term exists to produce, and an encoder that violates it is being pushed towards it by every gradient step.

Proposition 2. If $s_{ii} \geq s_{ij}$ for every j, then $\mathcal{L}_i^{\mathrm{NCE}} \leq \log B$, with equality if and only if $s_{ij} = s_{ii}$ for every j.

Proof of Proposition 2. The map $x \mapsto \exp(x/\tau)$ is increasing, so $s_{ii} \geq s_{ij}$ gives $\exp(s_{ii}/\tau) \geq \exp(s_{ij}/\tau)$ for every j, hence $\sum_{k=1}^{B} \exp(s_{ik}/\tau) \leq B \exp(s_{ii}/\tau)$ and therefore $p_{ii} \geq 1/B$. Taking $-\log$ reverses the inequality, giving $\mathcal{L}_i^{\mathrm{NCE}} \leq \log B$. Equality in the sum requires $\exp(s_{ik}/\tau) =$

$\exp(s_{ii}/\tau)$ for every k, hence $s_{ik} = s_{ii}$, which is the flat row of Proposition 1. This completes the proof.

Restricted to view-consistent encoders, the two propositions together say that the constant encoder *attains* the worst value of the contrastive term. On that set the body's phrase, that the term makes a constant encoder maximally lossy, is exact rather than figurative, and the constant solution is simultaneously a best point of the prediction term and a worst point of the term beside it.

It is not the *unique* worst point, and an earlier version of this appendix said it was. Proposition 2's equality condition is $s_{ij} = s_{ii}$ for every j, which is a statement about the similarity row being flat, not about θ. The constant encoder flattens every row, but the proof does not exclude a non-constant encoder that also does, and we have not shown that none exists. Nothing downstream needs uniqueness: the argument prices collapse at $\lambda \log B$ because the constant encoder attains the maximum, and that holds whether or not others attain it too.

C.4 What the argument does not establish

The restriction in Proposition 2 is not cosmetic. Over all encoders $\mathcal{L}^{\mathrm{NCE}}$ is not bounded above by $\log B$ at all: an encoder that made every positive pair *less* similar than its negatives would score above it, up to $2/\tau + \log(B-1)$ in the worst case. So the constant encoder is the worst view-consistent solution, not the worst solution, and a reader who took the body's sentence as a statement about the whole parameter space would be taking more than we have shown.

The argument is silent in three further ways. First, it prices collapse without proving it is avoided: the objective pays $\lambda \log B$ for collapsing, and whether that outweighs what the prediction term gains depends on λ, on τ, on B and on the optimisation path. The price is also mild in absolute terms, since $\log B$ grows logarithmically in the batch: at batch 256 it is 5.5452 nats, and doubling the batch adds only $\log 2$. Second, it says nothing about partial or dimensional collapse. The term needs windows to be mutually distinguishable and a low-dimensional subspace is enough for that, which is visible in the run we score: effective rank falls from 29.94 to 13.42 over $15{,}000$ steps, so the objective contracts the representation even where it does not collapse it. Third, both propositions describe points of the loss surface rather than a trajectory, and gradient descent is under no obligation to stay away from a maximum it is not being pushed towards early in training.

C.5 The control

The propositions make one prediction that is cheap to test: an encoder that has collapsed should be found sitting at $\log B$ on the contrastive term. Setting λ to zero at the configuration we score, context 512 and batch 256 for $15{,}000$ steps, collapses the encoder, and it collapses to where Proposition 1 says it should. Mean pairwise cosine similarity reaches 0.828 against a tripwire of 0.60, effective rank falls to 5.86, and the contrastive term, computed as a diagnostic rather than optimised, reads 0.995 of $\log B$. With the term restored at its canonical weight the same configuration reaches budget on five seeds of five, holding cosine similarity between 0.043 and 0.061 and effective rank at 13.42, while the loss falls from 5.5611 to 4.245 against a chance value of 5.5452.

This is one control and not an ablation, and the distinction matters for what may be concluded from it. It removes the term at a single weight, so it does not sweep the balance; it removes the term whole, so it does not separate the two jobs the term does, which are supplying negatives and supplying augmentation invariance; and it reports collapse diagnostics rather than a downstream score, so it does not show what collapse costs a probe. An earlier setup on a different corpus and budget points the same way, with the contrastive arm reaching full budget on three seeds of three where a prior-regularised arm collapsed on two of three, and we report that as corroboration rather than as evidence at the scored configuration. The sweep that would close the question is marked in the body as outstanding.

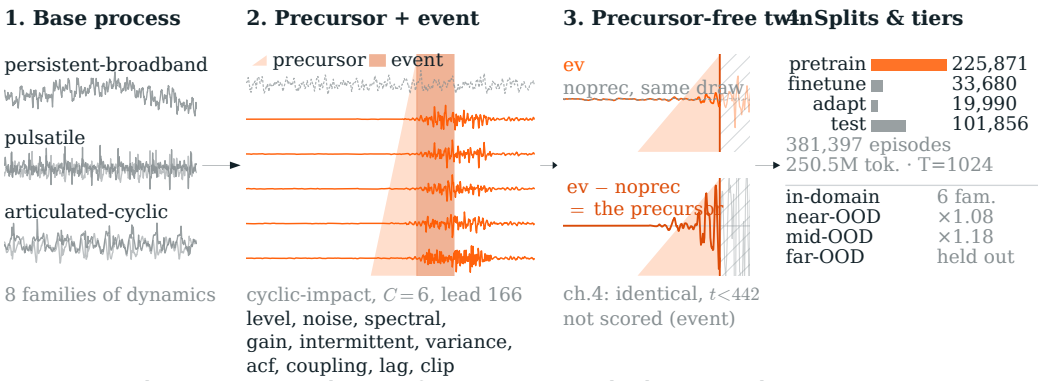


Figure 4: How we generate a synthetic episode (real episodes throughout). (1) We draw a base process from one of eight families, each defined by a measured profile of dynamics rather than by an application domain (Table 7). (2) The event bank injects exactly one event and the precursor that announces it, here on the orange channels; grey channels stay untouched. The precursor is drawn as a ramp rising to the onset and the event as an outlined column, the same marks as Figure 2. (3) One random stream also yields the twin `noprec`, discarding only the precursor: `ev` − `noprec` is exactly zero before the precursor starts and equals the precursor until the onset. After the onset the arms differ again because most operators are non-additive; no scored context reaches that far (hatched). (4) We split episodes entity-disjointly: `pretrain` trains the encoder, `finetune` fits in-domain heads, `adapt` fits heads on the out-of-domain tiers so a tier is never scored by a head fitted elsewhere, and `test` is held out. Tiers run from in-domain to two families withheld from pretraining. Episode geometry is in Table 9.

D THE SYNTHETIC CORPUS IN DETAIL

D.1 THE EIGHT FAMILIES

Table 7 gives each family's calibration target, the archive datasets it was estimated from, and its share of the corpus. The family names describe measured physics. They were previously the archives' own type labels, which both understate what the generator models and invite the objection that the corpus is fitted to one pair of archives; the exemplar column carries the old reading for anyone comparing against Figure 4, whose panels are annotated in the earlier scheme.

Ordered by how long the process remembers itself, each family holds a position the others leave empty. SMOOTH-CYCLIC is a near-deterministic slow cycle with almost all its energy at one frequency, and the lowest spectral entropy in the set. ARTICULATED-CYCLIC is that cycle with more spectral content and less step structure, a limb or a pen on a repeated path. CYCLIC-IMPACT punctuates the cycle with transients, and step fraction separates it from the last, 0.319 against 0.198. PULSATILE is mostly quiet with a repeating sharp complex, the only positively skewed tail among the persistent families. CORRELATED-STOCHASTIC puts nothing sharp anywhere, step fraction 0.150, the lowest, and carries its structure in the cross-channel correlation instead. PERSISTENT-BROADBAND is moderately rough, weakly periodic, untrended and thin-tailed, and is identified by that combination because each of its neighbours breaks on exactly one axis. SWITCHING is piecewise constant, step fraction 0.433, the highest: almost all its variation sits in a few jumps between levels. ROUGH-BROADBAND has the lowest persistence of the eight, a near-flat spectrum and near-independent channels, effective-rank fraction 0.091, so it is wideband rather than event-driven.

Table 7: The eight families of the corpus, ordered by persistence. Each row's five invariants are the median over the archive datasets of that type, and they are the calibration target the family's generator knobs are fitted to. *share* is the fraction of the released pack's 381,397 episodes drawn from that family; C is its channel count. PERSISTENT-BROADBAND and SWITCHING are univariate by construction, so no cross-channel effective rank is defined for them, and SMOOTH-CYCLIC's archive type supplies none. Excess kurtosis, so a Gaussian reads 0.

family	exemplar	share	C	ac_1	sp. ent.	period.	step	kurt.	e-rank
SMOOTH-CYCLIC	PigAirwayPressure	0.042	2–43	0.997	0.259	0.908	0.182	−0.74	–
ARTICULATED-CYCLIC	PenDigits	0.101	2–43	0.992	0.313	0.853	0.198	−0.37	0.639
CYCLIC-IMPACT	UWaveGestureLib.	0.310	2–43	0.990	0.329	0.819	0.319	−0.31	0.716
PULSATILE	ECG5000	0.109	2–43	0.985	0.472	0.524	0.352	+2.26	0.727
CORRELATED-STOCHASTIC	MotorImagery	0.071	2–43	0.978	0.514	0.523	0.150	−0.14	0.412
PERSISTENT-BROADBAND	ItalyPowerDemand	0.225	1	0.940	0.414	0.679	0.179	−0.38	–
SWITCHING	ScreenType	0.088	1	0.880	0.519	0.575	0.433	+1.67	–
ROUGH-BROADBAND	Phoneme	0.054	2–43	0.656	0.800	0.281	0.196	+4.68	0.091

Separability, all figures classifier two-sample AUROC on event-free panels. Every family reads 0.894 or above against a pool of the other seven, and 1.000 for the two withheld families. The

within-family floor, one half of a family against the other, runs 0.464 to 0.555: a two-sample test on this instrument does not read exactly 0.5 even on identically distributed halves, which is why the floor is reported beside every separation. The closest pair in the set is CORRELATED-STOCHASTIC against PERSISTENT-BROADBAND at 0.767; the next closest reads 0.974.

The names carry a testable claim, that a family is a physics rather than a domain. Measured on ten physical sources absent from both archives, general-aviation flight recordings land nearest PULSATILE at distance 0.177, closer than ROUGH-BROADBAND sits to any other family, and a turbofan lands 0.036 from SMOOTH-CYCLIC, closer than two archive families sit to each other. Extending the corpus to a new phenomenon therefore means measuring its signature and fitting knobs, not writing a new simulator, and that is the property we would want of any synthetic pretraining corpus meant to outlive the domains it was first fitted to.

D.2 THE NINTH FAMILY, FITTED AND NOT GENERATED

The eight families were estimated on the UCR and UEA archives, so whether they describe physical sensing or one archive is an open question, and we measured the five invariants on ten sources outside both: machining, spacecraft telemetry, bridge monitoring, a planetary rover, metro compressors, a turbofan, aviation flight data, a solar observatory, photovoltaic output and vehicle energy. Eight of the ten land within 0.29 of a family against a within-set spacing of 0.101. Five, however, are heavy-tailed past anything in the archives, whose maximum is ROUGH-BROADBAND at excess kurtosis 4.68 while these reach 6.2 to 57.1.

Raw kurtosis is set by a few extreme order statistics, so one sensor fault fakes it. Two robust statistics separate the cases: excess kurtosis recomputed after trimming the extreme 1% of samples, and the fraction of samples beyond five median absolute deviations. The five impulsive sources hold trimmed excess kurtosis 1.09 to 6.10 and 3.2% to 11.8% beyond five deviations, against a Gaussian's 0.07%: frequent excursions, not outliers. One further source fails that check, raw 5.06 collapsing to trimmed 0.12, and is excluded, which is what the check is for.

The resulting signature, lowest persistence in the set at ac_1 0.522 together with the heaviest tails at excess kurtosis 6.07, at moderate periodicity, is not reachable from any existing family by shifting knobs. Physically it is an intermittently excited mechanical or energetic system. Its generator should be a heavy-tailed innovation driving a lightly damped resonator, one mechanism for all three properties, and specifically not spikes added to a Gaussian base, which reproduces raw kurtosis without trimmed kurtosis and so reproduces the artefact the excluded source shows. The acceptance gate is trimmed excess kurtosis above 1.0 together with a beyond-five-deviations rate above 3%; passing on raw kurtosis alone is a rejection.

It is not in the corpus. Every generator that built a corpus in this paper draws from a mix of exactly eight families, and the ninth exists as a fitted signature, a generator design and that gate, not as generated data. What this changes is the scope of the claim rather than the corpus. The eight cover the central mass of physical sensing and the fitting protocol generalises off the archives, but the archives under-represent impulsive processes and a generator calibrated to them alone inherits that blind spot. The corpus therefore predicts where it should be weakest, which is anomaly detection in settings where excursions are themselves the signal.

D.3 THREE MATCHED ARMS, AND WHAT IS STORED

We generate every episode from one random stream in three versions: `ev` (base, precursor and event), `noprec` (base and event) and `noev` (base only). The injector consumes identical randomness in `ev` and `noprec` and discards only the precursor's effect, so the two share the onset, operator, amplitude and channels and are bit-identical before the precursor starts, in 250 of 250 sampled episodes. On any context ending before the onset their difference is exactly the injected precursor, which subsection 3.4 turns into the benchmark's null. After the onset the arms diverge again for an unrelated reason, since most operators are non-additive and the event is then applied to two different base signals; no scored context reaches that far.

Scoring happens only on the `test` and `adapt` splits, so the twins are stored for those and not for the rest. Three arms everywhere would cost about 48 GB against 25 GB this way, for data no protocol reads.

Splits and tiers come from stable hashes of the episode index rather than from any global shuffle, so the corpus is append-only: generating more episodes never moves an existing episode between splits and never changes one that already exists. The corpus holds $1{,}422{,}932$ episodes and about 800M unique patch tokens; the episodes reported in this paper come from a released pack of $381{,}397$ of them. An entity appears in exactly one split, and that pack's four splits hold `pretrain` $225{,}871$, `finetune` $33{,}680$, `adapt` $19{,}990$ and `test` $101{,}856$ episodes, with zero shared entities between any two. The four tiers hold $282{,}090$ in-domain, $22{,}848$ and $22{,}307$ at the two knob shifts, and $54{,}152$ in the two withheld families.

Labels ship with the `pretrain` split as well, so a reader can audit our claim to have withheld them rather than take it on trust. The protocol forbids using them, and nothing in the pretraining objective reads a label array.

D.4 Every sampled axis

Table 8 lists the generator's draws. Read with Table 9, which carries the timing, it is the complete specification of what varies between two episodes.

Table 8: The generator's sampled axes. Every draw is a deterministic function of the episode index, so the corpus is reproducible from the index alone and extends by appending.

axis	draw	what it varies
family	one of 8, at the shares of Table 7	the dynamics
knob tier	$\times 1.00$, $\times 1.08$, $\times 1.18$, or a withheld family	the operating regime, coarsely
knob jitter	$\pm 10\%$ per knob, clipped to $[0.7, 1.4]$	the operating regime, per episode
channels C	1, or one of $\{2, 3, 4, 6, 8, 12, 20, 32, 43\}$	panel width
coupling	centred on the family's own mean $\lvert r\rvert$, clipped to $[0.05, 0.97]$	how much channels share
effective rank	the family's own, from Table 7	how many directions the panel spans
amplitude scale	log-uniform over two decades	absolute signal level
time warp	$\pm 6\%$, clipped to $[0.85, 1.15]$	rate
local motif	2 to 6 per domain, 8 to 27 steps, gain 1.6 to 3.5	the class-carrying transient
white noise	3% of each channel's standard deviation	the noise floor
event operators	1 of 10, a second with probability 0.35	what changes
event amplitude	log-uniform over 8 to 20 channel standard deviations	how much
event profile	one of 4 ramp shapes, exponent log-uniform over $[0.4, 3.0]$	how fast
transience	probability 0.45, duration to the episode end	permanent or an excursion
precursor operator	the event's natural indicator with probability 0.65, else free	what announces it
precursor lead	$\mathcal{U}\{48, 288\}$ steps	how much warning
precursor amplitude	0.15 to 0.90 of the event's	how quiet the warning is
channel placement	disjoint from the event's 10%, partial 15%, else nested	where to look
event presence	65% of episodes bear exactly one	the base rate

Two gaps in what we can quantify. The realised distribution over the 55 reachable event compositions is not recorded anywhere, only the marginal over the ten operators (Table 15), so we cannot say which compositions are rare in the shipped episodes or whether any is unreachable in practice. And noise enters only twice, as the white term above and as one of the ten injected operators; the corpus records no observation-model axis beyond that, so quantisation, missingness and sensor faults are not swept and no per-family noise-to-signal range is audited.

D.5 DIVERSITY, MEASURED RATHER THAN ASSERTED

Operating regimes are a dose, not a switch. A classifier two-sample test against the same family at its unshifted knobs reads 0.44 to 0.51 at $\times 1.00$, 0.66 to 0.78 at $\times 1.08$ and 0.84 to 0.93 at $\times 1.18$, measured on four of the eight families. The $\times 1.00$ row is the instrument's own floor and it sits below 0.51, so the two shifted rows are separations rather than artefacts, and an 8% or 18% change of knobs is recoverable but not trivially so. The two families withheld from pretraining read 1.000 against the pool of the other seven, which is the far tier's separation and the top of the same ladder. Read against the pretraining pool rather than against each family's own unshifted knobs the ladder should be shallower, and we quote no figure for it, nor a dose response for the four families the knob-shift measurement does not cover.

The precursor bank is used unevenly, and we report the spread rather than smoothing it. Over the $5{,}843$ held-out event-bearing episodes of the detectability check the realised precursor mix runs from 193 episodes on the rarest operator to $1{,}117$ on the most common (Table 15). The imbalance follows the operators' own applicability: `coupling` needs at least two channels and 31.3% of episodes are univariate, and `gain` and `clip` are reached mainly through the natural-indicator map rather than drawn directly.

A single episode length does not fix the calibration to that length. Episodes are $T = 1024$ steps throughout, which is the one axis the corpus does not vary, so the natural objection is that the five invariants were fitted at one length and describe no other. Measured on the generator itself, extending an episode from $1{,}199$ to $4{,}096$ steps moves lag-one autocorrelation by at most 0.0054 and spectral entropy by at most 0.0671 across the eight families. Excess kurtosis moves more, by up to 1.1, and the heavy-tailed families dominate that, which is consistent with a longer window simply sampling more excursions. The persistence and spectral targets therefore transfer to other lengths; the tail target does not, and a corpus meant for much longer episodes should refit it.

The corpus is close to its event-free sibling, and the gap is the injection. Against the event-free corpus it shares a base generator with, a classifier two-sample test reads 0.582 AUROC on windows drawn anywhere in an episode and 0.547 on the pre-onset evaluation contexts. Both are near the instrument's floor, which is what we want: the two corpora differ by the injected precursors and events and not by a drift in the base process, and the smaller figure on the scored contexts follows from the precursor being the quieter of the two injections.

D.6 SCALING MODEL SIZE AND TRAINING EXPOSURE

Figure 3 measures how classification performance changes with encoder size and training exposure to a fixed synthetic corpus. The central distinction is between presenting more tokens and generating more independent trajectories. Increasing the former can improve learning from existing data without increasing the latter.

Budget and data are separate axes. Let N be the number of encoder parameters, D_{pool} the number of source patch tokens in the available corpus, and T the cumulative number of patch-token presentations during training, including repeated exposure. The training budget per parameter and the corpus-equivalent exposure are

$$b = \frac{T}{N}, \qquad \rho = \frac{T}{D_{\mathrm{pool}}}. \tag{4}$$

Holding the corpus fixed makes ρ grow with the training budget, while the number of distinct source tokens encountered remains bounded by D_{pool}. This ratio is an exposure measure, not a count of uniform passes: sampling and overlapping windows can revisit some trajectories more often than others. Different masks and augmentations provide new views of those trajectories, but do not generate new physical systems. Thus the horizontal axis in Figure 3 measures T, not unique dataset size.

Experimental setting. The four encoder sizes use the same synthetic corpus and training recipe, with seed 42, four accelerators and global batch size 512. We evaluate checkpoints at nominal budgets

$b \in \{40, 142, 400\}$, using a learning-rate schedule defined for 400 tokens per parameter. Only the final checkpoints complete annealing.

Observed scaling. Accuracy increases at every measured budget, but the gain from 142 to 400 tokens per parameter is smaller than the gain from 40 to 142 for every model. For the 10M and 20M encoders, the later interval adds only 0.19 and 0.20 percentage points, respectively, despite approximately 2.8 times as many total presentations. Larger encoders also achieve higher accuracy at a matched budget per parameter, with smaller gains between the largest sizes. Since $T = bN$, this comparison jointly increases model capacity and total training exposure.

Apparent saturation and unique-token coverage. The scaling curves flatten at larger model sizes and higher token budgets. A finite corpus provides a possible explanation for both trends: as N grows, the number of unique source tokens per parameter, D_{pool}/N, falls, while exposure at a fixed budget per parameter grows as $\rho = bN/D_{\text{pool}}$. Larger models therefore receive more presentations of the same available trajectories, rather than a proportionally larger supply of unique data. Repeated exposure can refine existing features while adding progressively less information about unseen systems. The apparent saturation may thus reflect limited unique-token coverage relative to model size and training budget, rather than an intrinsic limit of the model or the synthetic generator. At the narrowest size the coverage is not yet what binds: holding the presented-token budget fixed and pretraining on a tenth of the entities moves UCR-128 by $+0.0047$ at Saturn-1M ($p = 0.093$), and on three tenths by -0.0002, both inside the three-seed spread of 0.0027.

E The Precursor Check

E.1 The question, and the geometry that answers it

The question. Given a context of $L = 512$ steps ending at t and a horizon Δt, is there an event at $t + \Delta t$? The label is $y = \texttt{event_mask}[t + \Delta t]$. The mask is a single contiguous run beginning at the recorded onset in 248,163 of 248,163 event-bearing episodes, so the label reduces to onset $\leq t + \Delta t <$ onset + duration. The benchmark chooses no tolerance window, bucket width or threshold.

The geometry. Four inequalities force every quantity in Table 9: the context needs 512 steps, the event must finish inside the episode, the precursor cannot start before step 0, and the precursor reaches the context only if its lead exceeds the horizon.

Table 9: Synthetic episode geometry. o is the context offset, h the horizon of the onset from the context's last step.

quantity	value	why
episode length T	1024	
context	$[o,\, o + 511]$, $L = 512$	one context per episode; never slid
offset o	$\mathcal{U}\{0, 64\}$, independent of the label	window position carries no label information
onset t_0	$o + 511 + h$, $h \sim \mathcal{U}\{1, 384\}$	the context ends strictly before the event
precursor lead	$\mathcal{U}\{48, 288\}$	capped below 384, so the far horizons are a control
events per episode	exactly 1	no time-since-last-event shortcut

The measured consequences: $\text{corr}(o, \text{has_event}) = -0.0008$, $\text{corr}(o, h) = +0.0003$, horizon mean 192.6 on $[1, 384]$ against a target of 192.5 (per-horizon coefficient of variation 0.075), and a data-free scorer given t reads 0.4965–0.5014 AUROC across the four tiers. The model never sees o or t; the only temporal quantity that reaches it is Δt, and Δt reaches the head, not the encoder.

The horizon grid, and why it is capped where it is. The grid is $\Delta t \in [1, 384]$ and the canonical reporting window is $[1, 256]$. Two separate bounds argue for a cap and they are not the same bound. The first belongs to the corpus: 288 is the maximum injected lead, so an episode whose onset lies further than that beyond t cannot have its precursor in the context at all, and horizons past 288 are increasingly made of such episodes and measure the geometry of the corpus rather than a model. The

second is the *model's*, and it is the tighter one. Pretraining masks $\mathrm{round}(0.6 \times 32) = 19$ patches at a random position and 8 patches at the window's end, and the tail is taken *out of* that budget rather than added to it. Nineteen patches are masked in total: the last eight always, and a random contiguous block of eleven drawn from the patches before them. So of the 512-step window the encoder is *always* asked to reconstruct the final $8 \times 16 = 128$ steps, and *at most* $19 \times 16 = 304$, the latter only when the random block lands immediately before the tail. Reporting past 304 asks for a horizon the model was never trained to fill; we cap at 256, inside that ceiling. The scores reported in this paper were computed on the $[1, 288]$ window and have not been re-scored at that cap, so their horizon average still includes episodes out to 288 steps, past the span the mask geometry guarantees.

A prediction about the shape of the decline. The two spans above are not interchangeable. If the mask geometry is what sets forecasting accuracy, the decline against Δt should *kink* near 128 steps: gentle while the horizon sits inside the tail mask's own span, which every draw covers, then steeper from 128 to 304, which only the contiguous draws cover. A smooth decline across that range would instead say the decline is ordinary long-horizon difficulty and that the mask geometry is not what limits us. The distinction matters for what to change next, since only the first reading makes the tail mask's width a lever.

E.2 Three arms, and exactly where they agree

One random stream produces the `ev`, `noprec` and `noev` arms: `noev` is the base process itself, and the injector consumes identical randomness in `ev` and `noprec`, discarding in the second only the precursor's contribution through a single flag. Measured on 250 episodes, `ev` and `noprec` are bit-identical *before* the precursor starts (250 of 250), differ inside the precursor window (248 of 250; the other two are sub-threshold), and also differ *after* the onset (249 of 250). The last is not a defect. Nine of the ten operators are non-additive (`gain` is multiplicative, `variance` scales the residual, `clip` truncates the support), so an event applied to base-plus-precursor is not the same as that event applied to the base. The difference `ev` − `noprec` is therefore exactly the injected precursor on the interval the benchmark scores, since every context ends before the onset, and is *not* a precursor extraction over the whole episode. We store twins for the `test` and `adapt` splits, where scoring happens, and they cover 100% of their episodes. Event-free episodes also carry `noprec` twins, identical over the whole episode, so the arms literally share their negatives and only the positives differ.

One operator weakens the ceiling without affecting validity: `clip` precursors leave the context unchanged in 63.7% of univariate and 33.1% of multivariate cases, where the signal never reaches the clipping threshold. They cannot inflate a score, and they explain `clip` having the lowest recall of any operator (Table 15).

E.3 The balance invariant

Table 10: The same test episodes under four samplings, scored by a scorer that reads only Δt. Measured on all 101,856 test episodes across 384 horizons.

sampling	prevalence	marginal across Δt	Δt-only AUROC
natural, cumulative label (time to event $\leq \Delta t$)	32.55%	0.19% → 64.96%	0.7475
natural, event in progress at $t + \Delta t$	26.11%	0.19% → 45.66%	0.6888
natural, six unequal-width horizon buckets	–	5.44% → 16.53%	0.6253
balanced at every Δt (this protocol)	**50.00%**	**0.5000 → 0.5000**	**0.5000**

Onsets accumulating causes the ramp in Table 10, not events failing to end: with the onset offset uniform on $[1, 384]$, few episodes have had their onset by a small Δt and most have by a large one. The invariant is $P(y{=}1 \mid \Delta t) = 0.5$ exactly, at every Δt, in training and in evaluation. Each event-bearing episode contributes a positive at a horizon where its event is in progress; we draw negatives at the same horizon, so balance holds stratum by stratum and not merely overall, which would fix nothing while the rate still varied with the horizon. The index builder verifies the invariant and refuses to return an index that violates it, and we publish its fingerprint with every number, because an index subsamples positives per horizon under a seed and two runs are comparable only if they share it.

Balancing makes any label safe, so we choose the label on meaning alone. Scoring within each Δt is a second, independent guard: inside a stratum the horizon is constant and orders nothing. On a constant score the stratified estimator returns 0.500000, and on a Δt-only score it returns 0.500000. The two guards fail independently, which is why we keep both.

Balancing discards most negatives, so the stratified estimator carries the power. And 50% is not the real prior: before reading any output as a hazard we restore the fixed logit offset $\log\big(\pi/(1-\pi)\big)$ with $\pi = P(\text{event})/384$, which cannot change AUROC. The sampler carries the balance and never the loss; reweighting per horizon during training would force the fitted hazard flat in Δt and erase the timing signal the task exists to measure.

E.4 Negatives: four cuts, four questions

Each Δt receives equal numbers of two kinds of negative, so both sub-scores are estimable at equal n (Table 11).

Table 11: The four negative cuts.

cut	negatives are	question
`detection`	episodes with no event at all	is a precursor present?
`timing`	event-bearing episodes not in an event state at this Δt	is it happening now?
`timing_reach_matched`	as `timing`, both sides restricted to reaching precursors	as above, without presence as a shortcut
`combined`	both kinds	the mixture a deployment faces

`detection` is what a conventional event-forecasting AUROC reports. The separation is not cosmetic: a synthetic feature that encodes each episode's horizon value scores 0.8183 on `detection` and 0.5448 on `timing`. `timing_reach_matched` closes a further shortcut. At a large Δt an unmatched timing negative is often an episode whose event has already ended and whose precursor is plainly visible, while the positive at that horizon may have no precursor in reach; "see a precursor, answer no" then scores above chance with no temporal resolution. Requiring both sides to carry a visible precursor leaves only its progression.

E.5 Readout: one socket, one head

The socket. Every encoder reduces $(512, C)$ to one vector, which a frozen random orthogonal projection maps to 768 dimensions with no trainable parameters. A head whose input width varies by model is not a shared head: readout width alone moved a trivial baseline's score by 0.577 AUROC (0.211 at width 256, 0.788 at 576) on a byte-identical feature map. The projection is an isometry when the source is no wider than the target and lossy otherwise; measured on $4{,}000$ vectors, $\|y\|/\|x\| = 1.0000 \pm 0.0000$ for $768 \to 768$ and $768 \to 1024$ against 0.5770 ± 0.0205 for $768 \to 256$, which discards 512 dimensions. We therefore set the socket at or above the widest entry, and only ever move it up.

The head. Table 12 lists the three rungs, fixed for every entry. φ is fixed, never learned: 32 features, a linear term, a bias and 15 frequency pairs on a geometric ladder from 1 to 128 cycles across the grid, capped where a period reaches three steps, so a pretrained encoder and its random twin receive an identical horizon pathway.

Table 12: Heads. The linear probe is the headline because at any fixed Δt it is linear in z.

rung	form	parameters
`probe` (headline)	$\text{logit} = \varphi(\Delta t)^\top W z + b(\Delta t)$	26,144
`mlp`	FiLM: $\gamma, \beta = A\varphi(\Delta t)$; $\text{logit} = u^\top \text{GELU}(\gamma \odot z + \beta)$	249,345
`unconditioned` (ablation)	$\text{logit} = w^\top z$	2,305

The `unconditioned` head ships to measure one gap: if an entry's conditioned and unconditioned scores match, it is answering *whether*, not *when*. The loss is plain binary cross-entropy, the proper

scoring rule for a calibrated hazard. We run each entry as `probe`, `mlp` and `lpft` (probe to convergence, then the encoder unfrozen at $0.1\times$ the learning rate), under one budget: the same maximum epochs and learning-rate grid, early stopping on a validation slice carved from the fitting split *by entity*, the best checkpoint restored, and the number of steps to that checkpoint published, so an undertrained entry is visible rather than scored as a weak representation. A closed-form ridge readout ships as a determinism control.

The conditioning is non-degenerate, and we can see it in the head. The `unconditioned` ablation detects a collapsed conditioning only through its score. The probe's geometry shows the same thing directly. Because $\text{logit} = \varphi(\Delta t)^\top W z + b(\Delta t)$ is linear in z at any fixed horizon, each Δt reads the representation along a single direction $w(\Delta t) = W^\top \varphi(\Delta t)$, and the cosine between those directions separates the two cases a score cannot: a head that asks a genuinely different question at each horizon, and a head that applies one direction with a threshold that slides. Measured on the 11 horizons of the committed cell, the cosine between the $\Delta t{=}12$ and $\Delta t{=}192$ directions is $+0.030$ and the mean off-diagonal cosine is 0.233 (Figure 5). Near and far horizons are therefore close to orthogonal readings of the same vector. Two caveats travel with this. The random-init twin gives $+0.049$, so the property belongs to the head and the fixed horizon basis, not to pretraining; and it is a statement about what the readout *can* express, not evidence that any encoder has learned timing, which is what the `timing` and `timing_reach_matched` cuts are for.

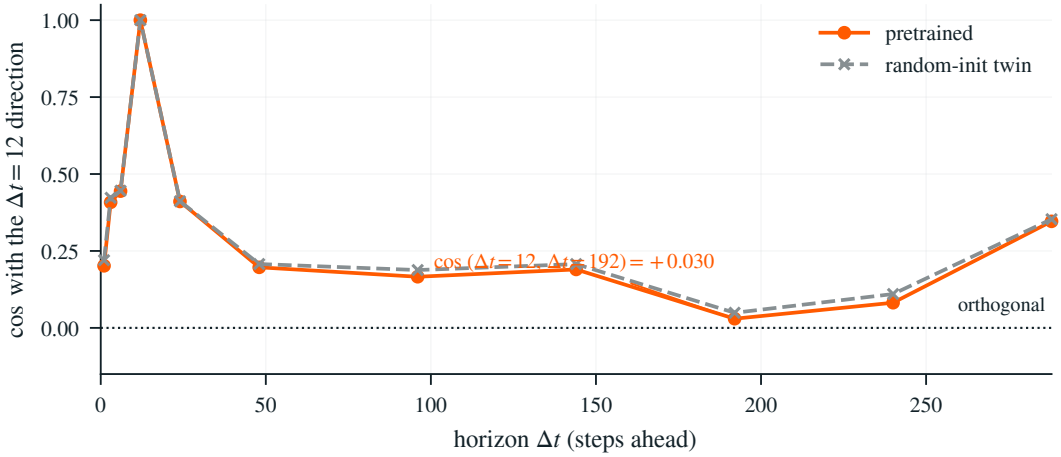


Figure 5: The readout asks a different question at each horizon. Cosine between the probe's horizon direction $w(\Delta t) = W^\top \varphi(\Delta t)$ and its direction at $\Delta t{=}12$, over the 11 horizons measured. The curve falls to $+0.030$ by $\Delta t{=}192$ and the mean off-diagonal cosine is 0.233, so the near and far horizons read the representation along nearly orthogonal directions rather than applying one direction with a shifted threshold. The random-init twin behaves the same way ($+0.049$), which places the property in the head rather than in pretraining: it shows the horizon conditioning is non-degenerate, not that any encoder has learned timing.

Where the head is fitted. We fit in-domain heads on `finetune`. We fit each out-of-domain tier twice and publish both: on `finetune` (*zero-shot*: does the event concept transfer with no target labels?) and on that tier's `adapt` episodes (*adapted*: can the representation support event forecasting here?). The `adapt` split holds 19,990 out-of-domain episodes, entity-disjoint from `test` (4,553 near, 4,541 mid, 10,896 far).

E.6 Scoring and floors

Estimator. We compute AUROC within each Δt and combine it by comparable-pair count, the stratified Mann–Whitney (van Elteren) estimator. We report AUPRC beside it, reweighted to the natural prevalence, with its chance line drawn at that prevalence.

Channel mix. We score each tier per channel group $\{1\}$, $\{2,3,4\}$, $\{6,8,12\}$, $\{20,32,43\}$ and recombine with the in-domain tier's mix. Without this a tier comparison is invalid: the far tier is 62.0% univariate against about 26% elsewhere, with mean channel count 6.18 against 11.21, because device power is univariate, and a raw in-domain to far drop would confound distribution shift with channel mix and penalise exactly the models that read channels jointly.

Intervals. Error bars come from an entity-clustered paired bootstrap. One episode contributes pairs at many horizons, so resampling pairs understates the interval; we resample over entities and reuse the same resamples for every arm and model, so differences are paired. Spread over head seeds is not a sampling distribution and we do not report it as one.

Floors. A cell is reportable only when all four floors in Table 13 read what they must.

Table 13: Mandatory floors, computed in every cell.

floor	must read	on synthetic
Δt-only	0.5000 exactly	structural: under the balance every score ties
`clock-only`	0.5 ± 0.02	0.4965–0.5014 across tiers
`noprec` (matched null)	0.5	0.4881–0.5084 for the reference detector (Table 14)
bit-identity	$\Delta = 0.0000$ exactly	where the precursor starts after t the arms are the same windows; $4{,}962$ of $4{,}962$

The bit-identity control lives inside the Δt strata rather than in one reserved bucket: a positive qualifies when its precursor starts after t, i.e. when its lead is shorter than its time to onset. Its share of positives tracks the lead distribution: impossible at $\Delta t \leq 48$, then 0.2%, 5.4%, 18.2%, 38.5% and 58.8% across successive bands. It is also why the canonical window extends to 288: only 96 control episodes exist at $\Delta t \leq 64$, against $20{,}965$ at $\Delta t \leq 288$. Where a precursor's lead ℓ is shorter than its time to onset h, the precursor starts after t, the two arms are the same windows and their scores must agree exactly, so anything else is a bug in the harness rather than a finding. Sitting inside the strata, the control catches a scoring bug that a bucket-level check would average away.

The matched null, and the margin it licenses. On the synthetic corpus we score every cell twice, on the `ev` contexts and on their `noprec` twins, fitting the head once and applying it unchanged to both, on the same episodes and labels, and we report the margin

$$\Delta \;=\; \mathrm{AUROC}(\texttt{ev}) \;-\; \mathrm{AUROC}(\texttt{noprec}). \tag{5}$$

Within the context the two arms differ only by the precursor, so family, channel count, amplitude, class motif and every route nobody has listed sit in both arms and cancel in the difference; what survives is precursor information and nothing else. This is a sanity check the synthetic corpus affords and a real one does not: a corpus we did not generate has no precursor-free version of itself (subsection E.9), so Δ never carries a claim outside the synthetic corpus. Because the null sits at chance here, between 0.4881 and 0.5084 AUROC across time-to-onset buckets for the reference detector (Table 14), the absolute AUROC is readable, and we report it as a headline with $\mathrm{AUROC}(\texttt{noprec})$ beside it as the floor that licenses reading it. Δ remains the tighter estimator of the two, since the arms share episodes, labels and head and episode-level noise cancels in the pair. On a sound corpus the two agree within their intervals, and a disagreement is a finding about the corpus, not about the model.

The canonical cell. Thousands of cells exist and we publish every one, and we fix one before any model runs: in-domain, `probe`, `combined`, $\Delta t \in [1, 288]$, channel-reweighted, reporting $\Delta(\text{pretrained}) - \Delta(\text{matched random twin})$ at identical adaptation budget, which cancels architecture, width, pooling and head capacity and leaves what pretraining bought. We also pre-registered two secondary cells, far-OOD `combined` and in-domain `timing_reach_matched`. We fix the horizon window in advance and never choose it after the fact, because a lead time chosen to maximise the margin is a maximum over a free axis. Over a family of 14 horizon buckets such a maximum read $+0.1634$ from a set whose median was $+0.0165$.

E.7 The precursor is detectable, and only up to a point

The detectability check runs a reference detector, MiniRocket features (Dempster et al., 2021) with a ridge readout fitted on each arm, on the detection question, grouping event-bearing episodes by their time to onset h. The lead time is about 192 steps, with no bucket clearing by 288, and the null stays at chance throughout, so the margin is the precursor and not the protocol. At a 10% false-alarm rate the reference detector recovers 16.4% of event-bearing episodes (11.0% at 5%, 26.0% at 20%). That is low by design, since a corpus whose baseline recovers everything measures nothing, and the difficulty spreads across operators (Table 15).

Table 14: The detectability check for a corpus. Reference detector: MiniRocket with 500 kernels and a ridge readout, fitted on each arm, on the detection question (positives are event-bearing episodes, negatives event-free ones), with positives grouped by their time to onset h. Scored per channel-count group over five entity-disjoint resamples on identical windows in both arms. AUROC is unitless; Δ is an absolute AUROC difference. A cell clears when $\Delta > 2$ sd. Beyond the maximum lead every precursor starts after t, so the two arms are the same windows.

time to onset h	positives	AUROC(ev)	AUROC(noprec)	Δ (sd)	
$(0, 32]$	1,356	0.6409	0.5084	$+0.1325$ (0.0184)	cleared
$(32, 64]$	1,361	0.6044	0.4997	$+0.1047$ (0.0074)	cleared
$(64, 128]$	2,772	0.5709	0.4949	$+0.0760$ (0.0082)	cleared
$(128, 192]$	2,777	0.5221	0.4966	$+0.0255$ (0.0078)	cleared
$(192, 288]$	4,188	0.4904	0.4881	$+0.0023$ (0.0017)	not cleared
$(288, 384]$	4,199	0.5066	0.5066	$+0.0000$ (0.0000)	identity

Table 15: Reference-detector recall at a 10% false-alarm rate, by precursor operator, over the held-out event-bearing episodes of the detectability check (n as listed). Chance is 10%.

operator	n	recall	operator	n	recall
level	972	22.6%	variance	1,117	13.4%
noise	661	22.2%	acf	601	13.1%
spectral	605	20.5%	coupling	281	10.7%
gain	193	16.6%	lag	593	10.5%
intermittent	618	15.5%	clip	202	8.4%

level, noise and spectral move a channel's own marginal, the easiest change for the reference detector to pick up. coupling and lag leave every marginal untouched and act on the relation between channels, and the detector sits at the chance floor on both. That is the headroom a representation reading channels jointly has to fill. Recall here includes event-bearing episodes whose precursor cannot reach the context, which pulls every figure towards the 10% chance level. Two checks confirm that the spread is real rather than noise: the detector recalls precursors that the generator's own visibility test rejects at 11.9%, against 17.1% for those it accepts, and per-episode scores spread widely, with 10th, 50th and 90th percentiles of -1.09, $+0.12$ and $+1.66$ in units of the negatives' own standard deviation.

E.8 Why the twin is load-bearing

The only test available without a twin is an absolute one, AUROC(ev) against 0.5. Its guarantee is one-sided and does not hold when the sampling selects negatives differently from positives. On a corpus that drew its negatives as windows far from the next onset, and whose ground truth we know exactly (a precursor is present if and only if the minimum lead is below the maximum injected lead), calling a precursor whenever AUROC exceeds 0.5 by more than two null standard deviations returns 10 of 14 cells correct, no false negatives, and **four false positives**, at $z = 4.9$, 5.1, 2.8 and 6.1, all four in cells where the two arms are identical. On that corpus the matched null itself sat at 0.5328 (sd 0.039) across 98 cells, above 0.5 in 87% of them, while a label permutation on the same cells returned 0.500001: deleting the precursor does not make two differently selected pools alike, and permuting labels destroys the selection that deletion preserves. More data makes the absolute rule worse, because the null's standard deviation shrinks while the offset does not. Under the twin margin of Equation 5 a detection implies a precursor by construction; under the absolute rule it does not. The corpus's geometry (subsection E.1) removes the selection offset, which is why its null reads 0.4881–0.5084 and why its absolute AUROC is readable; the twin is what proves that it did.

E.9 Real corpora: the Precursor Check

A corpus we did not generate has no twin, rarely has one event per episode, and does not label its own precursor, so we must show that a precursor exists there before we rank any model on it, and we cannot run four clauses of subsection E.6 on it: the matched null, the bit-identity control, the reach-matched timing cut and, on a single fixed sensor panel, the channel-mix reweighting. We declare these rather than approximate them and keep one question per horizon, balance at every Δt, the multiplicative head, horizon-averaged AUROC and entity-disjoint splits. Three further settings differ from the synthetic protocol. Contexts are shorter ($L = 64$ to 128), and we cut them from each recording at a fixed stride. We match negatives to positives in quantile bins on the context's end time,

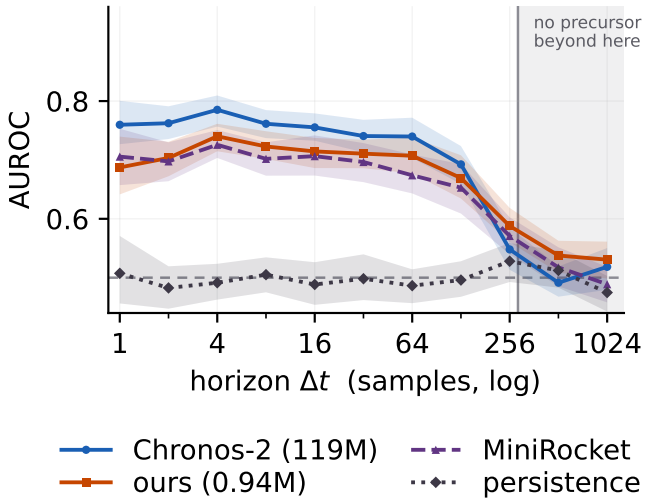


Figure 6: The same detector is exactly chance once the precursor is injected rather than inherited. AUROC against horizon on `scm_switching`, a generated corpus, drawn to the same scale as Figure 3. Persistence reads 0.4982 over the nine core horizons — chance to three decimals — against 0.7779 on the recorded corpus, while the ordering among the learned models is unchanged. The generated half is therefore the same task with the ground truth available, not a different task. The shaded region is a null *by construction*: the injected lead is drawn $\mathcal{U}\{48, 288\}$, so the fraction of positive windows still carrying a precursor is exactly $(288 - \Delta t)/241$ and is zero beyond 288 samples. Every curve landing on chance there is the instrument reading a known zero, which is what licenses reading it anywhere else. Bands as in Figure 3; this corpus has 150 generated domains, which is most of why they are narrower.

and on the wind tunnel, the cluster telemetry and the original simulator build also on the time since the previous event; elsewhere we leave that quantity unmatched and only measure it. The horizon grid is sparse, $\Delta t \in \{4, 8, 16, 24, 32, 48, 64, 96, 128, 160, 192\}$ samples, cut to 96 on the industrial testbed, whose series are short. And because these corpora supply an order of magnitude or more fewer pairs than the synthetic corpus, we hold the blind floors to 0.05 of chance rather than 0.02.

Where the precursor comes from. We look for corpora in which the system's own dynamics place a precursor: one sensor that leads another. The event is an upward crossing of a high quantile of the *lagging* signal, with nearby onsets merged, and we withhold every channel that defines the event from the model's input, so the label is not a threshold on anything the model reads. For most corpora a label-free screen chooses the pair: it measures for every pair of channels how much their correlation grows when a delay is allowed and takes the strongest pair whose lagging channel yields a well-posed event. In the wind tunnel (Gamella et al., 2025) the airflow path fixes the pair instead: we define the event on the downstream pressure relative to ambient, and on one recording the downstream pressure correlates with the intake fan's load at $r = +0.2841$ at zero lag and $+0.8185$ at a delay of 8.79 s. Its input keeps only actuators and fan speeds.

Supply, computed before any model is fitted. A context of length L is unusable if an event starts inside it. For prevalence p and mean event duration d,

$$P(\text{clean context}) \;\approx\; \exp\big(-p\,(1 + L/d)\big), \tag{6}$$

so the controlling quantity is L/d, not prevalence. A corpus of one-step events admits no clean context at any realistic L: the formula predicts two such datasets at 5×10^{-23} and 7×10^{-10} and both deliver exactly zero, and it predicts a third, with $p = 0.12$ and $d = 4.9$, at 0.0%, which delivers none. Across datasets Equation 6 agrees with the measured supply at Spearman $+0.770$ to $+0.890$, with one recorded miss: a testbed at $L = 1024$ predicted 27.8% clean that delivered none. We merge onsets that fall too densely for the same reason.

Blind instruments, read two-sided. We compute three scorers that read no sensor data (the horizon, the window's end time, and the time since the previous event) on every corpus, pooled across horizons. Where a quantity is matched the instrument checks the matching, which in bins leaves residual structure; where it is not, the instrument is the only guard, and with recurring events it can carry a corpus on its own: on an industrial testbed the time since the last event reads 0.6386, and on an oil-well corpus 0.8003. We read them two-sided, because an instrument that separates the classes

voids a result whichever way it points. One recorded window-end score of 0.2236 beside a detector at 0.7754 passes a one-sided test at $p = 1.000$, and is a separation of 0.2764 with its sign flipped. The sign is corpus-specific, 0.5824 on one corpus and 0.2040 on another. How negatives are drawn decides whether the instruments pass: with negatives from the same entity, the worse instrument reaches 0.2175 on a synthetic stratum whose precursor is known to exist, and 0.1566, 0.2174 and 0.2960 on three real corpora, while cross-entity negatives with fold-wise matching bring the first three to 0.0210, 0.0325 and 0.0211. The synthetic row carries the argument: a sampling rule alone can supply 0.2175 of separation to an instrument with no access to the data.

The reference detector, and why it is not horizon-conditioned. A separate model is fit for *every horizon*: MiniRocket features of the per-context z-scored input, refit on that horizon's training windows alone, read by a ridge classifier with an internally cross-validated penalty. Δt is never an input. Nothing is shared between horizons: not the transform, not the readout, not a normalisation constant.

This is deliberate, and it is a change from an earlier version of this protocol that used the bilinear conditioned probe of subsection E.5 as the reference detector. A single model shown every horizon at once can let a horizon with strong evidence carry one with none, because the shared parameters are fit against the pooled objective; the per-horizon train/test gap is then not merely unmeasured but undefined. Refitting independently is the price of a per-horizon number that can be read on its own, and it is affordable precisely because this detector's only job is to certify a corpus, never to be deployed. A deployed forecaster wants the opposite, one representation answering every horizon, which is what Saturn's own event head does (section 3). The two jobs need two architectures, and using the deployable one to certify the data is what let a weak horizon look admissible.

The rule. We admit a corpus when all three of the following hold on the timing cut, where negatives are event-bearing episodes not in an event state at that horizon: (i) at every horizon up to a step-down horizon greater than zero, the per-horizon reference detector beats chance with an excess recall, $(\mathrm{TPR} - \alpha)/(1 - \alpha)$ at $\alpha = 0.10$, of at least 0.05; (ii) the AUROC-against-Δt curve *decays* towards the permutation floor, since information that arrives before an event and does not fade as one asks further ahead was not a precursor; (iii) every blind instrument stays within 0.05 of chance, read two-sided. The instruments are re-read for every entry and not only at admission, since a gain on one would mean the corpus produced the score and not the model.

Clause (ii) replaces an earlier requirement that a horizon-conditioned head beat an unconditioned one. That comparison is not available under a per-horizon detector, since there is no conditioning to remove, and the property it was standing in for, that the corpus says *when* and not only *whether*, is what the decay of the curve tests directly.

Three real corpora pass: a physical wind tunnel (Gamella et al., 2025) whose downstream pressure follows the fan load by about nine seconds, a bearing test rig, and computing-cluster telemetry, the last a real system though not a physical one. Two caveats bound what they establish. The bearing corpus contributes 360 entities, which are vibration snapshots from one test rig rather than independent machines, so its entity-disjoint split is weaker than the others'. And we measured the cluster-telemetry and bearing corpora from a preprocessed store that is no longer available, so anyone re-scoring those rows must first rebuild the corpora from their public sources. A chemical-process simulator passes as well, twice on independently read copies of its data; because it is a simulation we report it beside the synthetic corpus as a second positive control and do not count it as real. Each rejected corpus fails at least one criterion on measurement: no horizon clears, the detection curve does not decay, or a blind instrument leaves the band. We set other candidates aside before scoring: too few admissible contexts or entities, no channel pair with a usable lag, or, for one body-worn activity corpus with a strong lead-lag, a domain outside physical systems.

What a twin-free floor cannot establish. Without a twin, a label-permutation floor, the usual substitute, is blind to a bias a detector can carry. On cells whose truth is exactly 0.5 the detector reads 0.05 to 0.09 below it while the permutation floor reads 0.4936–0.5046, because permuting labels destroys the structure that produces the bias. The smallest true effect the design can resolve is therefore $0.5 + |b| + 2\sigma_{\text{floor}}$, which evaluates to 0.585–0.607 across the real corpora we measured: a real-data null rules out a precursor at or above about 0.59 and says nothing about the band below, which is where a weak but useful precursor would live. An entity-aware null that would see this bias

does not exist on this design. A valid null must conserve the class marginal and decorrelate entity from class, and three constructions each fail one of the two (marginal drift $+0.2365$; entity–class correlation $+0.477$ and $+0.743$), with the obstruction itself measurable, $\mathrm{corr}(\text{entity size}, \text{entity rate}) = -0.597$ and 71.3% single-class entities. The fix belongs in the matching, not the permutation.

Independent units are necessary and not sufficient. Repacking a telemetry corpus one recording per file raises it from three fold groups to 27 entity-disjoint ones, yet all five of its scorable cells remain void, the blind instruments reaching an excess of 0.1274 to 0.1799 against the detector's 0.0146 to 0.0649. The widest bucket even clears its own floor, 0.5146 against 0.4955 at $z = +2.99$, and is void anyway, because the window's end time or the gap since the last event separates the classes about twelve times better than the detector does. A usable real corpus needs independent units *and* a matching that closes the clock, and a label that is not a threshold on a channel the model reads, which is why the Precursor Check defines events on a withheld lagging sensor.

Normalisation can hide a precursor. Context-only z-scoring is leak-free, and it is also blind to a precursor that is a sustained level shift: a context lying entirely inside the new level, rescaled to its own mean and standard deviation, is indistinguishable from a normal one. On an expert-annotated oil-well corpus, same windows and same detector, AUROC is 0.5336 under per-window normalisation and 0.8072 when we normalise the context against the episode's own opening baseline. The synthetic protocol keeps context-only normalisation for comparability and reports this as a limit of every model scored under it.

E.10 Related work

The confound the geometry of subsection E.1 removes is old and already named: control-selection bias under incidence-density sampling (Greenland and Thomas, 1982; Lubin and Gail, 1984), guarantee-time bias (Anderson et al., 1983), immortal-time bias moved from effect estimation to evaluation (Suissa, 2008), temporal bias in case–control design (Yuan et al., 2021), case–control alignment in sepsis prediction (Moor et al., 2021), and run-to-failure bias on the detection side (Wu and Keogh, 2023). It is not temporal leakage in the sense of Kapoor and Narayanan (2023): nothing from the future of the test set enters training. The remedy is as old as the name, whether risk-set negatives, a landmark design, or dense scoring at every timestep, as two lines of clinical work already do by construction (Reyna et al., 2020; Yèche et al., 2023); in intensive care, controls drawn from stable pre-discharge stretches cost 0.06 to 0.14 AUROC when matched to case onset (Do et al., 2026), the same order as the magnitudes above. Yuan et al. (2021) are the closest prior work: they show a case–control model's discrimination collapsing under evaluation from a prospective origin, using simulation and a second model built under a second design. What we add is the quantity itself, measured per cell by data-free instruments through the same pipeline and folds as the detector, and an exact synthetic null in which we know the precursor exists. This is the instrument-versus-shortcut move of Zech et al. (2018) and the degenerate-scorer move of Kim et al. (2022a), applied to observation time.

Anomaly-detection benchmarks (Wu and Keogh, 2023) score a different task, whether a model flags an event already in progress, whereas every context here ends before the onset; most of that family cannot supply a negative class at all, and one representative set carries $10{,}246$ onsets in roughly $10{,}000$ steps. The early-warning-signal literature asks the question at the centre of this protocol, on critical slowing down (Scheffer et al., 2009; Bury et al., 2021), and its performance study draws positives from windows around the transition and negatives from windows well before it with no control for window position (Proverbio et al., 2022); its ceiling, variance at AUC 0.72–0.78 where its assumptions hold and 0.28–0.32 where they break, is the calibration this task needs. The protocol must deliver power to resolve a few hundredths against a floor we measure rather than assume.

F Auditing the event-forecasting protocol

We audited this protocol route by route. Table 16 lists every way we could find in which a score on this benchmark might come from something other than the ability it claims to measure, and what we did about each: the routes the construction closes outright, the routes a per-cell measurement keeps under a stated bar, and the few that remain open and therefore shape how a number should be read.

Table 16: Route-by-route audit of the horizon-conditioned event-forecasting protocol. *Construction* routes cannot exist given how the corpus or the evaluation index is built. *Control* routes are monitored by a quantity computed in every cell, with a stated bar. *Reported* routes are real and not removable; they are measured and published, and they constrain how a number should be read rather than invalidating it.

Route	What it would exploit	Status	Evidence / treatment
Closed by how the corpus or the index is built			
Horizon clock	$P(\text{event})$ rises with Δt, so the horizon index alone separates the classes	construction	The builder balances the index to $P(y{=}1 \mid \Delta t) = 0.5$ at every Δt and refuses to return one that is not. A Δt-only scorer reads 0.5000 exactly in all four tiers, against 0.7475 on a cumulative label and 0.6253 on unequal-width buckets.
Absolute clock	Window position correlates with the label	construction	One window per episode at an offset drawn independently of the label, and the model never sees o or t. $\mathrm{corr}(o, \text{has_event}) = -0.0008$; $\mathrm{corr}(o, \text{horizon}) = +0.0003$; `clock-only` 0.4965–0.5014.
Gap leak	"Time since the last event" predicts the next one	construction	Exactly one event per episode, verified on $248{,}163$ of $248{,}163$: there is no previous event to measure from.
Post-onset contamination	The context already contains the excursion, so forecasting is detection	construction	horizon ≥ 1 by construction, so the context always ends strictly before the onset.
Entity leakage	The same episode appears in fitting and in scoring	construction	A hash of `entity_id` assigns the split; measured pairwise entity overlap is 0 across all six split pairs. We carve the head's validation slice by entity as well.
Normalisation leak	Episode-wide statistics import the event into the context	construction	We compute mean and standard deviation on the context only.
Readout-width advantage	A wider feature vector buys head capacity rather than representation quality	construction	One socket for every entry; the map is an isometry whenever native ≤ 768. Readout width alone can move a trivial floor's score by 0.577.
Closed by a quantity measured in every cell			
Base-process separation	Event-bearing and event-free episodes differ before any precursor	control	Matched null: `noprec` must read 0.5. Measured 0.4916–0.5072 in the leaderboard's canonical cells (0.4691–0.5207 across all 250 of its cells that carry a value) and 0.4881–0.5084 for the reference detector (Table 14). A cell whose null departs from 0.5 beyond its interval is not reportable.
Scoring-code error	A margin appears where the two arms are the same windows	control	Where `prec_start` $> t$ the arms are bit-identical ($4{,}962/4{,}962$), so their scores must agree exactly, in every Δt stratum above 48. This control detected a non-deterministic CUDA reduction in one baseline at 3.9×10^{-6}.
Horizon conditioning that is not	An additive head leaves the within-stratum ranking unchanged, so "conditioned" means nothing	control	An unconditioned head ships as an ablation; it measures 0.0000 rank change between $\Delta t{=}8$ and $\Delta t{=}320$, which is the degeneracy an additive head would share.

Route	What it would exploit	Status	Evidence / treatment
Detection reported as timing	Precursor *presence* stands in for precursor *timing*	control	Three cuts per cell. A synthetic feature encoding the horizon value scores 0.8183 detection against 0.5448 timing. `timing_reach_matched` also requires a precursor in the context on both sides, removing presence as a usable direction.
Selective reporting	Many cells, best one published	control	One cell pre-registered before any model ran; every cell published; intervals from an entity-clustered paired bootstrap rather than seed spread.
Real, not removable, and reported			
Tier composition	An OOD drop is read as distribution shift when it is partly channel mix	reported	far-OOD is 62.0% univariate against $\sim 26\%$ for the other tiers (mean C 6.18 vs 11.21), because `DEVICE` is univariate and is 62% of it. We post-stratify scores per channel group and recombine them to the in-domain mix.
Duration weighting	A long event is labelled positive at more horizons, so it counts more	reported	Within a Δt stratum an episode appears at most once, so the per-horizon curve carries none of it; only the combining scalar does. Training draws one Δt per episode per epoch. We report by duration tercile.
Horizon–duration coupling	The far-horizon region is not a fair sample of events	reported	The event must finish inside the episode, so duration $\leq 481 -$ horizon and $\mathrm{corr} = -0.538$: a far onset forces a short event. Corpus geometry at $T{=}1024$, not physics. Removing it requires longer episodes.
Head transfer confounded with encoder transfer	A zero-shot OOD number measures the head's ignorance of a new event class	reported	We fit OOD tiers twice, on the target tier's `adapt` split and on the in-domain `finetune` split, and publish both.
Cross-channel blindness	Channel-independent encoders cannot see part of the signal	reported	`coupling` and `lag` change nothing in any channel's marginal and are 17.2% of multivariate positives. We state each entry's channel handling and run Chronos-2 in both its joint and its channel-independent mode.
Sub-threshold operators	Some injected precursors do nothing	reported	`clip` is a no-op in 63.7% of univariate and 33.1% of multivariate cases. This caps the achievable ceiling; it cannot inflate a score.

Route	What it would exploit	Status	Evidence / treatment
Pretraining overlap	The in-house entry is pretrained on this corpus; the external entries are not	reported	Structural, and not symmetric. Saturn draws 40.1% of its pretraining tokens from the synthetic corpus's `pretrain` split; Chronos-2 and Chronos-Bolt have no exposure to this corpus at all. Three facts bound what that overlap can be: the splits are entity-disjoint (0 shared entities across all six split pairs, every entity in exactly one split), the pretraining loader reads only the series and its shape (`ev`, `offsets`, `C`, `split`) and no label array, and it never reads `noprec`, so the encoder never sees the matched null on which every Δ rests. What remains is not leakage but an advantage in *domain*: Saturn has seen this generator's dynamics and the external entries have not. We do not run a variant pretrained on the event-free base corpus, so this table cannot report what that advantage is worth, and a reader should take the in-house and external columns as two measurements rather than one ranking. Saturn carries a random-init twin and leads its external comparison on far-OOD.

On the three statuses. The distinction matters more than the individual rows. A *construction* claim is the strongest available and the cheapest to audit, because one check against the generator settles it rather than one per run. A *control* claim is weaker but more general: a floor test says that one named shortcut measured chance, whereas the matched null says that whatever the shortcut is, it sits in both arms and cancels in the difference, which is the only defence that covers routes nobody has enumerated. A *reported* row is an honest limit, and its number lets a reader discount the result by the right amount instead of either ignoring the issue or dismissing the benchmark.

What is not on this list. The corpus cannot currently supply two properties we would like. We cannot test horizons beyond 384 steps at $T{=}1024$: the arithmetic $1024 - 64 - 511 - 64 = 385$ leaves no room, and lifting the ceiling to 512 needs $T{=}1152$. And a two-sample classifier statistic orders the far-OOD tier but saturates at 1.0000 for both of its families, so we can call that tier far without ranking its members against each other.

G THE RANDOM-INIT TWIN: DOES PRETRAINING CARRY THE RESULT?

Every row of Table 2 has a twin: the same architecture at the same parameter count, with the same readout fitted on the same data, whose encoder we never pretrained. The twin controls for the only alternative explanation the main table cannot rule out by itself, that the readout rather than the representation does the work. Reading a pretrained score without it says nothing, because a frozen random encoder followed by a fitted head is already a strong baseline on many time-series tasks. A *negative* answer here would invalidate the paper.

H WHICH LAYER THE PROBE READS

The last layer is not the natural readout. A self-supervised objective is enforced at one particular place in the network, and the representation at that place is shaped to satisfy it. Satisfying a pretext means discarding whatever the pretext treats as nuisance, so the layer the objective acts on is the layer most specialised to the objective and the least likely to suit a task the objective never saw. Bordes et al. (2023) name the remedy after the instrument: cut the last layers away and probe what remains.

Table 17: Planned; no cell holds a measured value. Each pretrained model against its own random-init twin, architecture and readout held fixed. Δ is pretrained minus twin, so a positive number is the part of the score that pretraining supplies.

	classification	synthetic (canonical cell)	
model	Δ UCR/UEA acc.	Δ in-domain	Δ far-OOD
Saturn-1M	–	–	–
Saturn-10M	–	–	–
Saturn-20M	–	–	–
Chronos-2	–	–	–

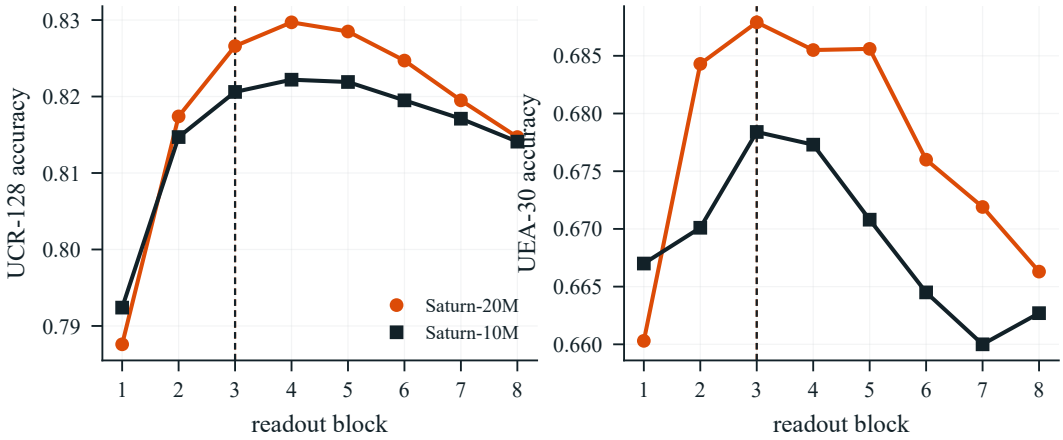


Figure 7: Readout depth. Mean accuracy with the frozen encoder read out at each transformer block, every candidate scored on the whole panel. The dashed line marks the block the probe is committed to, fixed for the architecture rather than chosen per panel. At Saturn-20M on UCR-128 the last block reads 0.8147 against 0.8297 at the best block, the specialisation guillotine regularisation predicts (Bordes et al., 2023).

They report differences above thirty accuracy points on ImageNet between the projector output and an earlier layer, they show that the depth of the optimum moves with the pretext rather than staying at the end, and they show that the dependence weakens as the pretext and the downstream task come into alignment.

Our setting sits at the far end of that axis. The invariance term of Equation 1 acts on a pooled vector through a projector of 74,304 parameters that pretraining discards, the prediction term acts on the encoder's own final token states, and the archives we probe share neither labels nor distribution with the synthetic episodes we pretrain on. Discarding the projector is therefore necessary and not sufficient, because the blocks that fed it were shaped by the same pressure. A recent time-series method also reads its representation from a space other than the one its objective projects into (Chemeris et al., 2026), and does not report where in depth that representation is best taken.

The shape of the profile. The encoder has eight blocks and the harness offers ten readout points: the tokeniser state, before any attention has run; each block's hidden state; and the encoder's own pooled output, which is what a downstream user gets by default. Accuracy across those ten points is single-peaked, and the peak is interior. On the provisional Saturn-3.7M model the fourth block stands **0.0151** above the encoder's own pooled output, while the tokeniser state, which has seen no attention at all, sits about nine accuracy points below that pooled output and about eleven below the peak. The peak falls on the third, the fourth or the fifth of the eight blocks in every panel and checkpoint we have probed, across two pretraining corpora and two variants of the objective, so the default output is not the right place to read this encoder and the right place is stable.

Naming the layer, which is easy to get wrong by one. Three conventions are in use around this project and they disagree. The harness indexes hidden states from zero, so its layer 2 is the *third* block. The readout script names cuts from one, so its cut `b3` is `hidden_states[2]` and therefore also the third block, while its `b4` is the harness's layer 3. This appendix counts blocks from one throughout: *block 3* is the third of the eight, the harness's layer 2, and the cut `b3`. A reader comparing a number here against a ledger or a results file should check which convention that file uses before concluding the two disagree.

How stable, measured. We pooled every arm, panel and checkpoint on which a full set of per-layer scores exists: 75 cells in all, spanning both probes and both pretraining corpora. The profile is single-peaked in every one, and the peak lies in blocks 2 to 5 in 57 of the 75. Block 3 is the modal peak, at 21 cells, and block 4 follows at 18. Table 18 prices the alternative to choosing at all: committing to one block in advance and reading every cell there. Committing to block 3 costs **0.0031** on average, which is a fifth of the across-dataset standard error of 0.0173 and a tenth of the seed standard error of 0.031, so within the noise this paper can resolve it costs nothing. Committing to the encoder's own pooled output, the choice a downstream user makes by doing nothing, costs **0.0268** and is the peak in none of the 75. Every figure in that table is a difference between two readings of one run under one probe, which is the comparison `probe.py` sanctions for the screen, so no cell mixes instruments.

Table 18: The cost of committing to one probe layer in advance, over 75 arm, panel and checkpoint cells. *Mean* and *worst* are the shortfall against that cell's own best layer, in absolute accuracy. *Exact peak* counts the cells where the committed block is already the best. The worst case for block 3 falls on a 25-dataset development subset that this paper does not report; on the panels it does report the worst is **0.0129**.

committed readout	mean cost	worst cost	exact peak
block 2	**0.0056**	**0.0216**	12 of 75
block 3	**0.0031**	**0.0305**	21 of 75
block 4	**0.0040**	**0.0313**	18 of 75
block 5	**0.0063**	**0.0302**	6 of 75
encoder's pooled output	**0.0268**	**0.0647**	0 of 75

The twin does not share the profile, and the caption must say so. A random-initialisation twin's accuracy rises with depth rather than peaking in the middle: over the eight blocks it runs from 0.6901 to **0.7024** at block 7, its best. Pretraining therefore does something to *where* the information sits, concentrating what a linear probe can read into the middle of the stack, while an untrained encoder merely accumulates it. That is a result worth having, and it is also a bias to declare: committing to block 3 costs the twin **0.0052** against about half that for a pretrained arm, so on the arms where we can measure both, a fixed shallow layer flatters the paired gain by the difference between those two. The effect is not uniform. On the canonical checkpoint the fixed-layer gain over the twin is *smaller* than the per-dataset one on the univariate panel, 0.0225 against 0.0247, and larger on the multivariate panel, so the direction depends on the panel. Any table reporting a fixed-layer twin comparison carries this sentence.

The rule, and what the numbers in this paper currently use. The protocol is one layer per model, committed on the corpus's own selection split, before any archive is touched. Because it is the pretraining corpus and the archives are the evaluation, a layer committed there carries no information about the panel it will be scored on, and the degree of freedom this section has been pricing does not exist under it. That rule also makes the comparison against published anchors conservative in the anchors' favour, since at least one of them reports a maximum over layers. The committed block for the shipping rungs is block 3, which Table 18 independently shows is the modal peak and free within noise.

The headline classification rows of Table 2 are scored under that rule: the shipping checkpoints are read at the committed block 3, giving mean UCR-128 accuracy 0.7697 at Saturn-1M and 0.8266 at Saturn-20M, with no layer chosen on an archive. The remaining classification cells, the per-lever rows of this appendix and the leaderboard comparison, still use the weaker fixed rule, one layer for the whole panel chosen by the panel mean on the evaluation split, which is one degree of freedom over ten candidates shared by every dataset rather than one per dataset. The per-dataset maximum appears only where it is labelled an upper bound. The rest of this section prices the difference between the weaker rule and a selection-free one, and the price is small, but small is not zero and a priced exposure is not an absent one.

The price of choosing on the evaluation split. Table 19 sets the three rules beside each other on the canonical checkpoint. Per-dataset selection buys 0.0173 on the univariate panel, 111 datasets scored, and more than that on the multivariate panel, where the two readings are 0.6830 and 0.6553. The fixed rule's own degree of freedom costs far less. We price it by leaving each dataset out of the choice: for every dataset the layer is selected on the remaining ones and the held-out dataset is then scored at that layer, which is the reading a protocol with no access to the dataset it scores would

have produced. Over every panel and checkpoint we could compute it on, that cost is at most **0.0088** and exactly zero to four decimal places in five cells of eight. It is zero whenever one layer is a clear peak, and positive only where two adjacent layers sit within a thousandth of each other, in which case the tie is settled by the noise of the dataset being held out. Choosing the layer on one archive and applying it to the other costs **0.0024** in the worse of the two directions.

Table 19: Three rules for choosing the probe layer, on the canonical checkpoint. The reported rule is the middle one. The per-dataset maximum is an upper bound and never a headline. The third rule selects the layer on the datasets other than the one being scored; we report it in the prose above for the provisional model, and have not computed it for this checkpoint.

rule	the layer is chosen on	univariate	multivariate
per-dataset maximum	each dataset's own test split	0.7850	0.6830
one layer per panel	the panel mean, on the test split	0.7676	0.6553
one layer per panel	the datasets other than the scored one	–	–

Fixing the layer does not rest the result on the choice. Under one common layer the paired comparison against the matched random-initialisation twin survives on both archives, and on the multivariate archive it grows. On the univariate panel the pretrained encoder reads 0.7676 against the twin's 0.7451, a paired gain of 0.0225 ± 0.0050 on 73 of 111 datasets ($p = 8.0 \times 10^{-6}$). On the multivariate panel it reads 0.6553 against 0.6250, a gain of 0.0303 ± 0.0112 on 18 of 21 datasets ($p = 0.006$). The conclusion the table rests on is therefore a statement about the representation and not about the readout depth.

Two facts about the panels bound all of this. Our multivariate panel is every dataset in the archive, while the published anchors report twenty-seven of the thirty, so its membership differs from the published UEA-27 and only the sign of a gap to those columns is comparable, never its size. And the provisional model scores all 128 datasets of the univariate archive against 111 for the canonical checkpoint, so those two columns share a panel name and not a membership.

I PER-DATASET CLASSIFICATION RESULTS

Table 20 lists every dataset of UCR-128 and UEA-27 for every baseline that is comparable with Saturn, under one probe. The published columns are the logistic-regression tables of Feofanov et al. (2026b), the only report that scores this whole field under a linear probe. We score Saturn with the same probe: StandardScaler and scikit-learn logistic regression, the configuration those tables state. We leave out four published models. TiViT-H and TiConvNeXt render a series as an image and read it with a vision backbone. Mantis+ and MantisV2 are classification encoders. None of the four is a model of the task this paper is about: none can state the probability that an event is in progress Δt steps ahead.

The Saturn column is Saturn-10M, the golden-recipe checkpoint trained to 400 tokens per parameter. It is read at *one* probe layer, the output of the fourth of eight blocks (L3), on both panels and for every dataset; selecting the best layer for each dataset would raise its averages, and we do not do it. The feature keeps channels apart. On UCR-128, where every dataset has one channel, it is the mean over the channel's patch tokens and the CLS token. On UEA-27 it is the channel mean: all channels are encoded together, as in pretraining, each channel's tokens are averaged over its valid time patches, and the per-channel vectors are concatenated, so the probe can weight each channel on its own. For InsectWingbeat we follow the published $1{,}000/1{,}000$ subsample with our own draw, since theirs is not released.

Table 20: Every dataset, every comparable baseline, frozen encoder and the same probe throughout: StandardScaler and logistic regression (scikit-learn, L-BFGS, at most 500 iterations), canonical splits, seed 42. Published columns are the logistic-regression tables of Feofanov et al. (2026b) (UCR-128: their Tables 11–12; UEA-27: Table 13). Image-based encoders (TiViT-H, TiConvNeXt) and the Mantis family are omitted: they cannot make the event predictions this paper is about. Saturn-10M is the golden-recipe checkpoint at 400 tokens per parameter, read at one probe layer, block L3, on both panels. On UCR-128, where every dataset has one channel, its feature is the mean over the channel's patch tokens and the CLS token. On UEA-27 it is the *channel mean*: all channels are encoded together, each channel's patch tokens are averaged over time, and the per-channel vectors are concatenated. InsectWingbeat is a $1{,}000/1{,}000$ subsample on both sides (ours is our own draw). Bold: best in the row, ties included. A superscript gives the number of datasets an average covers when a model is missing some.

	published, logistic regression							ours
Dataset	C22+	TabPFN	TabICL	MOMENT	TiRex	Chr.-2	NuTime	**Saturn-10M**
UCR-128 (128 datasets)								
ACSF1	0.700	0.800	0.810	0.750	0.750	**0.860**	0.700	0.770
Adiac	0.665	0.803	0.803	0.777	0.783	**0.829**	0.800	0.806
AllGestureWiimoteX	0.549	0.623	0.504	0.661	0.701	0.684	0.607	**0.723**
AllGestureWiimoteY	0.523	0.633	0.511	0.703	**0.731**	0.704	0.653	0.726
AllGestureWiimoteZ	0.483	0.533	0.453	0.606	0.634	0.660	0.541	**0.686**
ArrowHead	0.714	0.754	0.743	0.777	0.783	**0.851**	0.726	0.829
BME	0.993	**1.000**	0.980	0.993	0.993	**1.000**	0.967	**1.000**
Beef	0.633	0.800	0.767	0.800	**0.933**	0.700	0.733	0.700
BeetleFly	0.700	0.900	0.800	**0.950**	0.900	0.800	0.700	0.800
BirdChicken	0.800	0.850	0.750	**1.000**	0.900	0.900	**1.000**	0.900
CBF	0.973	0.913	0.924	0.989	0.997	**1.000**	0.990	0.994
Car	0.833	0.783	0.817	**0.900**	0.800	0.850	0.733	0.867
Chinatown	0.977	**0.985**	0.980	0.983	0.983	**0.985**	0.962	0.971
ChlorineConcentration	0.589	0.950	**0.977**	0.779	0.802	0.767	0.609	0.641
CinCECGTorso	0.843	0.834	0.823	0.768	**0.957**	0.899	0.855	0.950
Coffee	**1.000**	0.964	**1.000**	0.964	**1.000**	**1.000**	**1.000**	**1.000**
Computers	0.668	0.620	0.656	0.612	**0.800**	0.768	0.788	0.728
CricketX	0.610	0.667	0.664	0.744	0.700	**0.756**	0.705	0.736
CricketY	0.595	0.700	0.633	0.708	**0.754**	0.736	0.690	0.746
CricketZ	0.633	0.672	0.669	**0.744**	0.733	0.723	0.659	0.736
Crop	0.687	0.799	**0.812**	0.723	0.703	0.716	0.690	0.740
DiatomSizeReduction	**0.967**	0.961	0.951	0.941	0.928	0.948	0.915	0.928
DistalPhalanxOutlineAgeGroup	0.712	**0.763**	**0.763**	0.669	0.748	0.741	0.705	0.719
DistalPhalanxOutlineCorrect	0.754	**0.783**	0.775	0.754	0.750	0.764	0.743	0.764
DistalPhalanxTW	0.619	**0.698**	0.683	0.626	0.655	0.655	0.683	0.683
DodgerLoopDay	0.575	0.613	**0.725**	0.438	0.550	0.525	0.550	0.575
DodgerLoopGame	0.855	0.790	0.797	0.841	0.790	0.891	0.819	**0.899**
DodgerLoopWeekend	0.978	**0.986**	0.978	**0.986**	0.957	0.964	0.964	0.964
ECG200	0.850	0.890	0.880	**0.900**	0.830	0.840	0.850	0.870
ECG5000	0.939	0.942	**0.945**	0.933	0.939	0.934	0.919	0.934
ECGFiveDays	0.870	0.924	0.983	0.970	0.976	**0.991**	0.899	0.956
EOGHorizontalSignal	0.528	0.528	0.500	0.564	0.572	**0.588**	0.533	0.575
EOGVerticalSignal	0.478	0.489	0.434	**0.500**	0.417	0.439	0.332	0.384
Earthquakes	0.741	**0.748**	**0.748**	0.705	0.691	**0.748**	0.734	0.712
ElectricDevices	0.675	0.703	0.661	**0.740**	0.671	0.717	0.679	0.663
EthanolLevel	0.568	**0.848**	0.694	0.682	0.508	0.654	0.520	0.566
FaceAll	0.802	**0.808**	0.771	0.779	0.788	0.753	0.721	0.744
FaceFour	0.886	0.909	0.886	0.875	0.955	0.909	**0.966**	0.875
FacesUCR	0.850	0.877	0.877	**0.891**	0.878	0.858	0.820	0.842
FiftyWords	0.723	0.739	0.717	0.776	0.723	**0.796**	0.717	0.776
Fish	0.863	0.880	0.886	**0.960**	0.903	0.943	0.914	0.914
FordA	0.874	0.897	0.876	0.906	**0.947**	0.928	0.906	0.940
FordB	0.727	0.756	0.714	0.756	**0.816**	0.807	0.765	0.806
FreezerRegularTrain	0.992	**0.999**	0.988	0.988	0.991	0.995	0.994	0.986
FreezerSmallTrain	0.941	0.893	0.810	0.819	0.891	0.969	**0.986**	0.908
Fungi	0.919	0.866	0.785	**1.000**	0.952	0.946	0.747	0.968
GestureMidAirD1	0.623	0.623	0.677	0.685	0.692	**0.731**	0.685	0.692
GestureMidAirD2	0.638	0.561	0.585	0.577	0.615	**0.700**	0.561	0.646
GestureMidAirD3	0.408	0.369	0.361	0.369	0.408	0.354	**0.446**	**0.446**
GesturePebbleZ1	0.820	0.849	0.878	0.895	0.884	0.901	0.895	**0.930**
GesturePebbleZ2	0.703	0.785	0.753	0.892	0.842	0.867	0.772	**0.962**
GunPoint	0.927	0.967	0.953	**1.000**	0.980	0.993	0.973	0.973
GunPointAgeSpan	0.943	**0.991**	0.984	0.978	**0.991**	**0.991**	0.968	0.981
GunPointMaleVersusFemale	**1.000**	0.997	**1.000**	0.997	0.994	0.994	0.987	0.997
GunPointOldVersusYoung	**1.000**	**1.000**	**1.000**	0.927	0.965	0.978	**1.000**	0.997
Ham	0.629	**0.743**	0.724	0.657	0.705	0.695	0.733	0.705
HandOutlines	0.884	0.916	0.927	**0.941**	0.876	0.938	0.886	0.924
Haptics	0.419	0.471	0.461	0.490	0.477	0.503	0.425	**0.542**
Herring	0.578	0.594	0.641	0.641	0.609	0.516	0.531	**0.719**
HouseTwenty	0.950	0.849	0.756	0.933	**0.983**	0.975	0.899	0.975
InlineSkate	0.378	0.335	0.344	0.364	0.438	0.436	0.356	**0.489**
InsectEPGRegularTrain	0.992	**1.000**	**1.000**	0.948	0.996	**1.000**	**1.000**	**1.000**
InsectEPGSmallTrain	0.932	**1.000**	**1.000**	0.835	0.956	0.920	**1.000**	0.988
InsectWingbeatSound	0.582	**0.667**	0.656	0.621	0.625	0.631	0.510	0.577
ItalyPowerDemand	0.939	**0.970**	0.963	0.954	0.960	0.964	0.912	0.960
LargeKitchenAppliances	0.837	0.637	0.696	0.760	0.829	**0.861**	0.771	0.821
Lightning2	0.754	0.672	0.705	**0.787**	0.738	**0.787**	0.738	**0.787**
Lightning7	0.699	0.699	0.740	0.740	**0.808**	0.794	0.781	**0.808**

Table 20, continued

Dataset	C22+	TabPFN	TabICL	MOMENT	TiRex	Chr.-2	NuTime	**Saturn-10M**
Mallat	0.887	**0.969**	0.948	0.919	0.955	0.947	0.849	0.901
Meat	0.933	0.983	0.933	**1.000**	0.900	0.933	0.933	0.833
MedicalImages	0.679	0.795	**0.808**	0.746	0.725	0.747	0.709	0.754
MelbournePedestrian	0.903	**0.980**	**0.980**	0.895	0.894	0.905	0.933	0.909
MiddlePhalanxOutlineAgeGroup	0.558	0.623	**0.630**	0.506	0.487	0.552	0.552	0.597
MiddlePhalanxOutlineCorrect	0.777	**0.852**	0.835	0.808	0.808	0.835	0.749	0.825
MiddlePhalanxTW	0.545	0.617	**0.623**	0.545	0.500	0.500	0.442	0.532
MixedShapesRegularTrain	0.915	0.934	0.930	0.936	**0.970**	0.967	0.936	0.965
MixedShapesSmallTrain	0.869	0.829	0.877	0.890	0.936	**0.939**	0.921	0.927
MoteStrain	0.850	0.889	0.879	0.905	0.924	0.926	**0.955**	0.859
NonInvasiveFetalECGThorax1	0.899	**0.941**	0.927	0.924	0.899	0.884	0.855	0.901
NonInvasiveFetalECGThorax2	0.918	**0.948**	0.941	0.937	0.903	0.913	0.903	0.913
OSULeaf	0.736	0.566	0.595	0.793	**0.938**	**0.938**	0.860	**0.938**
OliveOil	0.833	**0.933**	0.900	0.900	**0.933**	0.833	0.767	0.833
PLAID	0.654	0.790	0.566	0.786	0.870	**0.886**	0.756	0.885
PhalangesOutlinesCorrect	0.746	0.840	**0.861**	0.814	0.792	0.823	0.747	0.795
Phoneme	0.242	0.110	0.136	0.300	0.390	**0.417**	0.276	0.382
PickupGestureWiimoteZ	0.660	0.760	0.740	0.700	**0.840**	0.660	0.640	0.780
PigAirwayPressure	0.144	0.019	0.154	0.111	0.418	0.418	**0.452**	0.370
PigArtPressure	0.764	0.034	0.255	0.548	0.889	0.841	0.942	**0.952**
PigCVP	0.418	0.019	0.173	0.534	0.865	0.788	0.885	**0.889**
Plane	0.991	0.991	0.991	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
PowerCons	0.989	**1.000**	**1.000**	0.956	0.939	0.933	0.972	0.961
ProximalPhalanxOutlineAgeGroup	0.839	**0.859**	0.839	0.829	0.849	0.834	0.854	0.854
ProximalPhalanxOutlineCorrect	0.845	0.904	**0.924**	0.887	0.887	0.883	0.839	0.852
ProximalPhalanxTW	0.790	0.810	**0.829**	0.795	0.761	0.756	0.785	0.771
RefrigerationDevices	0.533	0.504	0.493	0.515	0.541	**0.568**	0.563	0.504
Rock	0.800	0.760	0.640	**0.900**	0.880	**0.900**	0.760	0.780
ScreenType	0.469	0.419	0.411	0.392	0.528	**0.539**	0.469	0.504
SemgHandGenderCh2	0.920	**0.947**	0.887	0.765	0.858	0.888	0.862	0.917
SemgHandMovementCh2	0.658	**0.771**	0.569	0.444	0.520	0.576	0.624	0.682
SemgHandSubjectCh2	0.827	**0.936**	0.833	0.709	0.813	0.844	0.756	0.878
ShakeGestureWiimoteZ	0.820	0.820	0.740	0.840	0.880	**0.920**	0.900	**0.920**
ShapeletSim	0.956	0.478	0.506	0.967	0.967	**1.000**	0.928	0.994
ShapesAll	0.780	0.802	0.792	0.848	0.863	**0.897**	0.872	0.853
SmallKitchenAppliances	0.773	0.787	0.763	0.611	0.813	0.811	**0.829**	0.808
SmoothSubspace	0.940	**1.000**	**1.000**	0.980	0.967	0.967	0.947	0.900
SonyAIBORobotSurface1	0.759	0.772	0.672	0.825	**0.937**	0.824	0.815	0.849
SonyAIBORobotSurface2	**0.928**	0.809	0.828	0.891	0.891	0.885	0.883	0.909
StarLightCurves	0.960	0.973	0.972	0.964	0.974	0.975	0.975	**0.976**
Strawberry	0.943	0.981	**0.984**	0.962	0.965	0.973	0.960	0.943
SwedishLeaf	0.898	0.950	0.946	0.931	**0.954**	0.947	0.939	0.947
Symbols	0.955	0.882	0.894	0.956	0.967	**0.984**	0.969	0.979
SyntheticControl	0.980	0.990	0.983	0.977	0.993	**0.997**	0.973	0.993
ToeSegmentation1	0.759	0.575	0.667	0.947	**0.969**	0.917	0.842	0.961
ToeSegmentation2	0.769	0.654	0.815	0.908	**0.915**	0.862	0.869	0.908
Trace	**1.000**	0.910	0.980	0.990	**1.000**	**1.000**	**1.000**	**1.000**
TwoLeadECG	0.951	0.951	0.925	0.992	0.985	**0.993**	0.962	0.846
TwoPatterns	0.979	0.995	0.903	0.992	**0.996**	0.995	0.942	0.990
UMD	0.993	**1.000**	0.938	0.993	0.958	0.993	0.972	0.972
UWaveGestureLibraryAll	0.917	**0.967**	0.963	0.960	0.943	0.959	0.932	0.932
UWaveGestureLibraryX	0.781	0.808	0.798	0.793	0.800	0.821	0.800	**0.834**
UWaveGestureLibraryY	0.688	0.715	0.716	0.714	0.707	0.739	0.696	**0.759**
UWaveGestureLibraryZ	0.717	0.751	0.738	0.740	0.733	0.755	0.755	**0.777**
Wafer	0.995	0.995	0.996	0.994	**0.999**	0.998	0.996	0.992
Wine	0.704	0.778	0.722	0.833	**0.870**	0.852	0.778	0.796
WordSynonyms	0.564	0.644	0.597	**0.685**	0.599	0.668	0.600	0.668
Worms	0.584	0.571	0.558	0.688	0.792	0.779	0.766	**0.818**
WormsTwoClass	0.727	0.610	0.571	0.753	**0.805**	**0.805**	0.675	0.779
Yoga	0.711	0.860	**0.865**	0.851	0.795	0.819	0.749	0.852
Average	0.770	0.781	0.771	0.798	0.820	0.824	0.791	0.823
Best (incl. ties)	6	31	22	18	25	34	13	25
UEA-27 (27 datasets)								
ArticularyWordRecognition	0.933	0.930	0.910	0.987	0.990	**0.993**	**0.993**	**0.993**
BasicMotions	**1.000**	**1.000**	0.950	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
CharacterTrajectories	0.984	0.967	0.969	0.984	0.969	0.974	**0.985**	0.980
Cricket	0.903	0.819	0.847	0.986	**1.000**	**1.000**	**1.000**	0.986
DuckDuckGeese	0.420	0.320	0.280	0.480	0.380	0.340	0.500	**0.540**
ERing	0.907	0.889	0.852	0.970	0.978	0.978	**0.982**	0.981
EigenWorms	0.763	0.420	0.565	0.832	0.840	0.840	0.870	**0.939**
Epilepsy	0.964	0.913	0.957	0.993	**1.000**	**1.000**	**1.000**	0.993
EthanolConcentration	0.373	**0.783**	0.605	0.471	0.403	0.547	0.411	0.464
FaceDetection	0.547	0.632	0.640	0.621	**0.660**	0.636	0.619	0.592
FingerMovements	0.520	0.500	0.490	0.570	**0.620**	0.600	0.610	0.500
HandMovementDirection	0.351	**0.432**	0.392	0.311	0.324	0.338	0.378	0.311
Handwriting	0.231	0.139	0.226	0.379	**0.416**	0.354	0.253	0.409

Table 20, continued

Dataset	C22+	TabPFN	TabICL	MOMENT	TiRex	Chr.-2	NuTime	**Saturn-10M**
Heartbeat	0.756	0.722	0.727	0.702	0.771	0.751	0.742	**0.785**
InsectWingbeatSubset	0.324	0.238	–	0.298	0.393	0.344	**0.618**	0.297
JapaneseVowels	0.965	0.814	0.816	0.924	0.941	0.900	**0.981**	0.914
LSST	0.509	0.548	0.502	0.605	0.592	0.601	0.557	**0.620**
Libras	0.844	0.672	0.639	0.856	0.894	0.906	**0.917**	0.894
MotorImagery	0.500	0.580	0.570	0.490	0.480	0.530	**0.590**	0.500
NATOPS	0.817	0.778	0.794	0.844	0.839	0.822	0.867	**0.878**
PEMS-SF	0.711	**0.948**	0.942	0.786	0.775	0.815	0.896	0.838
PhonemeSpectra	0.206	0.148	0.171	0.199	0.236	0.243	0.247	**0.280**
RacketSports	0.730	0.816	0.849	0.849	0.888	0.868	**0.914**	0.822
SelfRegulationSCP1	0.744	**0.894**	0.887	0.826	0.850	0.850	0.857	0.853
SelfRegulationSCP2	0.478	0.478	0.506	0.511	0.494	0.483	**0.550**	0.472
SpokenArabicDigits	0.900	0.959	0.961	0.979	0.964	0.916	0.965	**0.982**
UWaveGestureLibrary	0.859	0.806	0.762	0.916	0.872	**0.925**	**0.925**	0.887
Average	0.676	0.672	0.685$^{(26)}$	0.717	0.725	0.724	0.749	0.730
Best (incl. ties)	1	5	0	1	6	5	13	9