STATISTICAL RISK HEADS FOR FAILURE WARNING ACROSS TIME-SERIES FORECASTERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurate time-series forecasting does not by itself specify when to issue a useful failure warning. This decision requires distinguishing approaching failures from normal operation while balancing missed events and false alarms. We introduce a statistical risk-head framework that adapts frozen forecasting models to failure warning using training targets that increase as failure approaches and degradation indicators computed from information available at prediction time. We train the risk heads using labels based on time remaining before failure, with additional losses that suppress high scores during safe operation, match changes in the target curve, and discourage falling risk as failure approaches. Warning thresholds are selected using training or calibration data. We evaluate the framework with DLinear, PatchTST, iTransformer, TTM, and FlowState on tiny-N-CMAPSS, NASA Battery Aging, XJTU-SY, and FEMTO/PRONOSTIA. Our system warns before more failures than a standard binary risk head in all 20 model–dataset pairs and improves 57 of 60 comparisons of event recall, false positive rate, and precision. Averaged equally across the 20 pairs, event recall increases from 65.98% to 87.35%, precision from 45.63% to 70.91%, and false positive rate decreases from 9.43% to 1.80%. These findings support statistical risk adaptation as a practical way to improve failure warnings across heterogeneous forecasting models.

1 INTRODUCTION

Time-series forecasting is used in various domains to aid decision making by predicting future values from historical data, spanning a broad range of models including linear models, transformer based models, and pretrained foundation models. In many contexts, however, a forecast is only an intermediate measure rather than a decision that a user needs, and an accurate forecast does not always translate to accurate failure prediction. In failure warning, a system observes the history of an asset and decides whether a failure will occur in a time horizon, and balance missed warnings against false positives. Learning these failures is difficult because they are very rare in safety critical systems (Arias Chao et al., 2021), so representative data is scarce. When the warning horizon is very short relative to the asset's life, they only form a small minority of the training data.

Existing formulations only address these limitations partially. Horizon classifiers(Susto et al., 2015) use a binary horizon label, which gives the same target to an observation right before a failure as well as one at the end, and treat the observation right after a horizon as safe, ignoring temporal dependence between samples. Point level metrics(Tatbul et al., 2018) are not a good representation of events that span time, and can misrepresent real performance.

We propose a statistical risk-head framework that turns frozen forecasts into failure warnings. Our contributions are:

- A shared causal interface that combines forecast summaries with observed degradation features, supporting different forecasting backbones without changing internal operations or weights.
- A bounded correction to a binary risk score. The head learns to adjust the score while explicitly limiting the size of the logit change.
- A training objective that combines temporally graded targets with 3 penalties: discouraging false warnings, matching changes in the target curve, and discouraging falling risk while the target rises.

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.

- An analysis of what graded supervision asks the score to learn and how bounded correction changes the binary score, and conditions under which a separate statistical procedure can recover a warning boundary.
- An evaluation across five forecasting backbones and four prognostic datasets. The risk system catches more failures than the standard binary head in all 20 model–dataset pairs, with favorable changes in 57 of the 60 comparisons of event recall, false positive rate, and precision.

2 Related Work

Time-series forecasting models. A wide range of models has been developed for time-series forecasting, spanning from simple linear approaches to highly complex architectures. DLinear decomposes a time series into trend and seasonal components, achieving performance comparable to that of larger models (Zeng et al., 2023). PatchTST cuts each channel into short segments, called patches, and feeds these to the model as tokens (Nie et al., 2023). iTransformer treats the full history of each variable as one token, so that attention learns how the variables relate to each other (Liu et al., 2024). Pretrained models can be applied to new data with little or no extra training. Chronos converts numbers into tokens and trains a language-model backbone on them (Ansari et al., 2024). TTM is a small pretrained mixer model built for zero- and few-shot forecasting (Ekambaram et al., 2024). FlowState uses a state-space encoder and a decoder that can produce forecasts at different sampling rates and horizons (Graf et al., 2026). These are trained and evaluated based on the accuracy of their forecasts but do not decide when to issue a failure warning. In this work, we keep them fixed and augment them with a risk prediction head that determines when a warning should be issued.

Failure prediction and remaining useful life. Online failure prediction commonly labels an observation positive if a failure happens within a fixed window after it(Salfner et al., 2010). In prognostics and health management, the goal is to estimate the remaining useful life (RUL) of a machine. Deep networks such as LSTMs (Zheng et al., 2017) and convolutional networks (Li et al., 2018) are trained for this task on run-to-failure datasets, (Saha & Goebel, 2007), (Wang et al., 2020), (Nectoux et al., 2012). Closest to our setting, Abdouni et al. (2026) add a small adapter to the embeddings of a frozen time-series foundation model to predict remaining useful life (RUL). However, this is designed for a single model architecture, while ours can attach to any backbone. Since failures are rare, a standard binary classifier sees very few positive examples during training and treats all time points the same. Focal loss (Lin et al., 2017), class-balanced weighting (Cui et al., 2019), and label smoothing (Szegedy et al., 2016) have been proposed to address this. Temporal label smoothing extends this idea by increasing the target value as a failure approaches(Yèche et al., 2023). Building on this, we use a time-dependent target that rises as the failure nears, and add a bounded correction, a penalty on false warnings and on changes in the target curve and mismatched risk direction.

Threshold-based anomaly detection and event-level evaluation. Alarm systems turn a running health or anomaly score into an alarm by comparing it with a threshold. The threshold can be set from forecast errors (Hundman et al., 2018), or extreme value theory (Siffer et al., 2017). Evaluating such alarms, however, is not straightforward. Point-based metrics do not describe events that last over time, which led Tatbul et al. (2018) to propose range-based versions. "Point adjustment" counts a whole event as detected if any single point in it is flagged, inflating detector performance(Kim et al., 2022). Therefore, we report event-level recall and point-level false positive rate and precision, apply no point adjustment, and choose warning thresholds from calibration scores taken during safe operation, with a randomization test to pick the warning boundary.

3 Problem formulation

We study the following decision: given an asset's observations up to the present, should the system warn of an event of interest (e.g., a failure) within the next H time units? A useful system warns before many distinct failures and raises few alarms during safe operation. We formalize the decision first, then describe how the risk head learns it.

For asset i, let $x_{i,k} \in \mathbb{R}^p$ be the observation completed at time $\tau_{i,k}$, and let $\mathcal{H}_{i,k} = x_{i,1:k}$ denote the available history. Let $\mathcal{T}_i$ be the set of onset times of events of interest for this asset. We define the

absolute time of the next event and the *remaining time to that event* separately:

$$T_{i,k} = \inf\{t \in \mathcal{T}_i : t > \tau_{i,k}\}, \qquad d_{i,k} = T_{i,k} - \tau_{i,k},$$

with $\inf \varnothing = \infty$. For failure events, the index k allows the next failure to change after an asset recovers and returns to operation. For example, if failures occur at times 100 and 160, the next failure at time 80 is 100; at time 120, after recovery, it is 160. The remaining times are 20 and 40.

Fix a warning horizon $H = h_{\max} > 0$. The outcome for a decision at (i, k) is

$$Y_{i,k} = \mathbf{1}\{0 < d_{i,k} \leq H\}.$$

An observation is *eligible* if a full input context is available, the asset is not actively in the event of interest phase (e.g., failure state), and follow-up establishes this outcome: either an event of interest is observed within H, or the asset is observed without such event throughout the next H time units. Observations with insufficient follow-up are excluded rather than labeled safe. A *safe observation* is an eligible observation with $Y_{i,k} = 0$.

We seek a scoring function $s_\theta(\mathcal{H}_{i,k}) \in [0, 1]$ and a threshold η that turn the available history into a warning decision:

$$A_{i,k} = \mathbf{1}\{s_\theta(\mathcal{H}_{i,k}) \geq \eta\}.$$

For an event e on asset i at time t_e, its warning set contains eligible observations $x_{i,k}$ that occur after $t_e - H$ (i.e., $\tau_{i,k} \geq t_e - H$) and for which the next event of interest is e:

$$\mathcal{W}_e = \{x_{i,k} : T_{i,k} = t_e,\ 0 < t_e - \tau_{i,k} \leq H,\ x_{i,k} \text{ eligible}\}.$$

An event e is caught when at least one observation in $\mathcal{W}_e$ raises a warning. Event recall is the fraction of events caught. FPR is the fraction of safe observations that raise warnings, and precision is the fraction of warnings with $Y_{i,k} = 1$. Thus ten warnings before one failure still catch only one event. Section 6 specifies event eligibility and aggregation for the datasets. The scoring function is causal: it uses completed observations only. Failure times determine training labels and evaluation outcomes, not prediction inputs.

4 METHOD

The method has three stages: extract causal features from a frozen forecaster and observed history, adjust a binary logit through a bounded risk head, and convert the resulting score into a warning. Figure 1 shows the information flow.

4.1 A SHARED CAUSAL INTERFACE

A frozen forecaster maps the last L observations to K future observations, denoted $\widehat{x}_{i,k+j|k}$ for $j = 1, \ldots, K$. We summarize its output by concatenating the channelwise forecast mean and final prediction:

$$f_{i,k} = \left(\frac{1}{K} \sum_{j=1}^{K} \widehat{x}_{i,k+j|k},\ \widehat{x}_{i,k+K|k} \right) \in \mathbb{R}^{2p}.$$

The head also needs to know the asset's current condition. We summarize the observed sensor history in $\psi_{i,k}$. The summaries include the current channel values, their means over the last L observations, their linear trends over that window, and the difference between the trailing means and the asset's initial L-observation means. The trend features are least-squares slopes multiplied by $L - 1$, expressing the fitted change across the context window. Preprocessing parameters are fitted on training data and kept fixed during calibration and evaluation. The summaries themselves use only observations available at time $\tau_{i,k}$. Where a physical failure limit is available, $c_{i,k}$ adds three coordinates for the asset's current and projected position relative to that limit; Section 4.3 defines them. The complete input is $v_{i,k} = (f_{i,k}, \psi_{i,k}, c_{i,k})$. For a battery, these groups describe what the forecaster predicts, how the battery has recently changed, and how close it is to its capacity limit. Each group uses information available at the decision time.

The objective comparisons share this input and the transformations fitted on training data. A binary head b_0 learns the full-horizon label $Y_{i,k}$ and is then frozen. Its output $z^0_{i,k} = b_0(v_{i,k})$ is the starting logit at calibration and inference. To train the correction, we instead score each training asset with an auxiliary binary head fitted without that asset's fold. Risk-head training changes neither the forecasting architecture nor its weights.

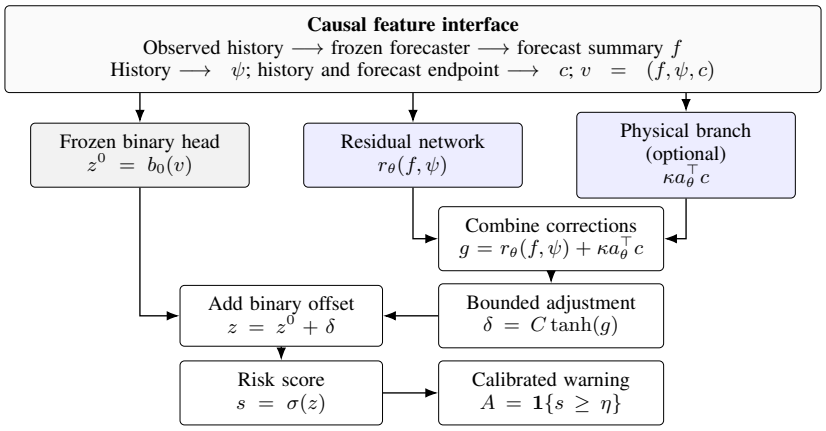


Figure 1: The common risk head at inference; asset and time indices are omitted. The binary logit bypasses the correction network and is added only after the bounded adjustment. Blue boxes contain learned correction parameters. The physical branch is absent when no suitable boundary coordinates are available. Failure labels are used for training and calibration, not as inputs to this diagram.

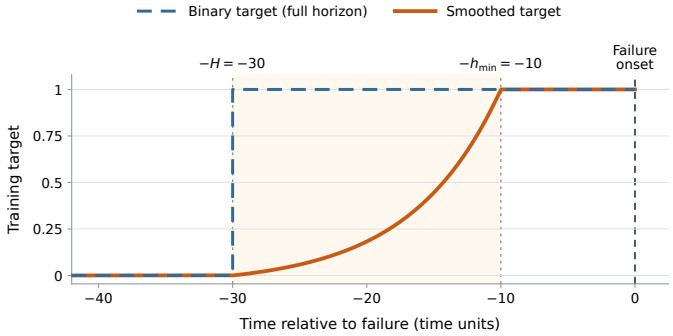


Figure 2: Training targets before a failure at time zero. The full-horizon binary target becomes one at $d = H$. The smoothed target rises gradually from zero at $d = H$ to one at $d = h_{\min}$, then remains one until failure. Here $H = 30$, $h_{\min} = 10$, and $\gamma = 0.15$.

4.2 Temporally Graded Supervision

The outer horizon $H = h_{\max}$ defines the warning interval. The *inner horizon* $h_{\min} \in (0, H)$ marks the region immediately before failure that receives target one. Between these horizons, we use the exponential temporal label smoothing, written in terms of remaining time:

$$q(d) = \begin{cases} 1, & 0 < d \leq h_{\min}, \\ \dfrac{e^{-\gamma(d-h_{\min})} - e^{-\gamma(H-h_{\min})}}{1 - e^{-\gamma(H-h_{\min})}}, & h_{\min} < d < H, \\ 0, & d \geq H, \end{cases} \qquad \gamma > 0. \tag{1}$$

Here γ controls the shape of the transition: as failure approaches, the target rises from zero at $d = H$ to one at $d = h_{\min}$ and remains one for $0 < d \leq h_{\min}$. The parameters are selected using training data. For example, with $h_{\min} = 10$, $H = 30$, and $\gamma = 0.15$, observations 5, 20, and 30 time units before failure receive targets 1, approximately 0.18, and 0. The graded target changes the supervision within that warning interval. Figure 2 compares the graded target with a binary target over the same warning horizon. Section 5 explains what the resulting score represents.

4.3 BOUNDED RISK ADAPTATION

As shown in Figure 1, the correction combines an unrestricted residual network with an optional physical branch, bounds their combined output, and adds it to the binary logit:

$$\begin{aligned} z_{i,k} &= z^0_{i,k} + \delta_{i,k}, \qquad \delta_{i,k} = C\tanh(g_{i,k}), \\ g_{i,k} &= r_\theta(f_{i,k}, \psi_{i,k}) + \kappa a_\theta^\top c_{i,k}, \qquad s_{i,k} = \sigma(z_{i,k}). \end{aligned} \tag{2}$$

The sigmoid $\sigma(z) = (1 + e^{-z})^{-1}$ converts the logit to a score in $(0, 1)$. The constant $C > 0$ sets the largest allowed logit adjustment. The residual network r_θ learns from the forecast and observed-history summaries, while the learned coefficient $\kappa \geq 0$ sets the strength of the physical branch. The binary logit bypasses both branches and is added at the end. Zero-initializing the residual output and setting $\kappa = 0$ therefore reproduces the binary score at initialization.

The physical branch expresses current and projected degradation in the same coordinates. For batteries and bearings, $u_{i,k}$ is a signed, normalized distance to the specified capacity or vibration limit. Zero denotes the limit, negative values lie on the healthy side, and larger values indicate greater degradation. The nonnegative increments $\Delta^{\text{obs}}_{i,k}$ and $\Delta^{\text{pred}}_{i,k}$ project deterioration over H from the trailing observed trend and the frozen forecast endpoint, respectively. The branch combines the current coordinate with these two projections:

$$\begin{aligned} c_{i,k} &= (u_{i,k},\ u_{i,k} + \Delta^{\text{obs}}_{i,k},\ u_{i,k} + \Delta^{\text{pred}}_{i,k}), \\ a_\theta &= (1 - \varepsilon)\,\mathrm{softmax}(\theta_c) + \varepsilon(0, 0, 1), \qquad \kappa \geq 0. \end{aligned} \tag{3}$$

Here $\theta_c \in \mathbb{R}^3$ is a learned vector of mixture logits, with $\mathrm{softmax}(\theta_c)_j = e^{\theta_{c,j}} / \sum_{r=1}^{3} e^{\theta_{c,r}}$. It is part of the head parameters θ. The fixed $\varepsilon \in (0, 1)$ gives the forecast coordinate a weight of at least ε; all weights are nonnegative and sum to one.

For example, $c = (-0.6, -0.3, 0.1)$ describes an asset currently below its degradation limit, still below the limit under the observed-trend projection, but beyond it under the forecast projection. Increasing one coordinate cannot reduce the physical branch's contribution when the other score components are held fixed. This conditional property does not require the complete score to increase along every observed trajectory.

4.4 FITTING AND DECISIONS

Let $\mathcal{I}_i$ contain asset i's eligible training observations, with $n_i = |\mathcal{I}_i| > 0$, and let m be the number of training assets. Write $q_{i,k} = q(d_{i,k})$ for the graded target and $s_{i,k} = \sigma(z_{i,k})$ for the predicted risk. We assign $w_{i,k} = 1/(mn_i)$, giving each asset equal total weight in the supervised loss.

With $\ell(z, q) = \log(1 + e^z) - qz$, we minimize

$$\begin{aligned} \mathcal{L} = \sum_{i=1}^{m} \sum_{k \in \mathcal{I}_i} w_{i,k}\ell(z_{i,k}, q_{i,k}) \\ + \lambda_{\text{FP}}\mathcal{L}_{\text{FP}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}} + \lambda_{\text{mono}}\mathcal{L}_{\text{mono}}, \end{aligned} \tag{4}$$

where all penalty coefficients are nonnegative. The supervised term fits the graded risk target. The three penalties discourage high risk when the target is zero, encourage changes that follow the target curve, and discourage decreasing risk while the target rises.

False-positive penalty. Let $\mathcal{B} = \{(i, k) : k \in \mathcal{I}_i,\ q_{i,k} = 0\}$ contain the training observations whose target risk is zero. We penalize their predicted risk using

$$\mathcal{L}_{\text{FP}} = \frac{1}{|\mathcal{B}|} \sum_{(i,k) \in \mathcal{B}} s^2_{i,k}.$$

This term pushes scores toward zero wherever the graded target is zero. This includes the boundary $d = H$, which has target zero but remains a positive observation under the evaluation label $Y = \mathbf{1}\{0 < d \leq H\}$. Thus the penalty follows the graded target rather than exactly matching the set used to calculate FPR.

Shape-aware smoothness penalty. We want changes in predicted risk to follow changes in the graded target. Let $\mathcal{P} = \{(i,k) : k \in \mathcal{I}_i,\ k+1 \in \mathcal{I}_i\}$ index consecutive eligible observations within each asset's trajectory. Pairs never cross asset boundaries or bridge excluded observations. Define $\Delta s_{i,k} = s_{i,k+1} - s_{i,k}$ and $\Delta q_{i,k} = q_{i,k+1} - q_{i,k}$. The penalty is

$$\mathcal{L}_{\text{smooth}} = \frac{1}{|\mathcal{P}|} \sum_{(i,k)\in\mathcal{P}} \big(\Delta s_{i,k} - \Delta q_{i,k}\big)^2.$$

When the target is flat, this term discourages fluctuations in predicted risk. When the target rises or falls, it encourages the prediction to change in the same direction and by a similar amount. It matches the local shape of the target curve rather than requiring risk to remain constant.

Monotonicity penalty. During portions of the trajectory where the target risk increases, a decrease in predicted risk is undesirable. Let $\mathcal{R} = \{(i,k) \in \mathcal{P} : \Delta q_{i,k} > 0\}$. We penalize downward changes on these pairs:

$$\mathcal{L}_{\text{mono}} = \frac{1}{|\mathcal{R}|} \sum_{(i,k)\in\mathcal{R}} \big[\max\{0, s_{i,k} - s_{i,k+1}\}\big]^2.$$

This term is zero when predicted risk stays constant or increases, and positive when it decreases while the target is rising. Unlike the shape-aware penalty, it does not specify the size of the desired increase. It adds a one-sided penalty for changes in the wrong direction.

All three penalties use training observations only. A penalty is defined as zero when its indexing set is empty. Setting its coefficient to zero removes that penalty from the objective.

For the controlled objective comparison, model selection uses held assets within training data. A separate calibration partition sets each head's pooled safe-score 95th-percentile cutoff. We advance the cutoff to the next representable value above that quantile to handle ties, and issue a warning when $s_{i,k} \geq \eta$. Calibration fixes the operating point; it does not force evaluation FPR to equal 5%.

5 Statistical properties

The analysis answers three questions: what the graded target asks the score to learn, when a warning boundary can be recovered, and how much a bounded head can change the binary score. Each result concerns a specific part of the procedure.

Meaning of the supervised score. Consider an eligible observation from Section 3. We use V for its feature vector $v_{i,k}$ and D for its remaining time $d_{i,k}$ to the next failure. These are two different quantities: V is available to the predictor, whereas D supplies the training outcome. To match the loss weights, imagine drawing an asset uniformly and then drawing an eligible observation uniformly from that asset. All expectations below use this sampling rule. When follow-up confirms no failure through H and no next failure is observed, we use $D = \infty$; this leaves the labels and target on the warning horizon unchanged. Finally, $F_v(u) = \Pr(D \leq u \mid V = v)$ denotes the conditional probability of failure within u time units.

Proposition 1 (Estimand of temporal supervision)**.** *Unrestricted minimization of conditional binary cross-entropy with target $q(D)$ gives $m_q(v) = \mathbb{E}[q(D) \mid V = v]$. For the target in Equation (1),*

$$m_q(v) = \int_{h_{\min}}^{h_{\max}} F_v(u) \frac{\gamma e^{-\gamma(u-h_{\min})}}{1 - e^{-\gamma(h_{\max}-h_{\min})}}\, du, \qquad \gamma > 0. \tag{5}$$

Thus $F_v(h_{\min}) \leq m_q(v) \leq F_v(h_{\max})$.

The logarithmic-loss argument follows the proper-scoring-rule interpretation of cross-entropy (Gneiting & Raftery, 2007). Concretely, the ideal score averages failure probabilities across the warning interval, placing greater weight on nearer horizons. With inner and outer horizons of 10 and 30, it summarizes urgency across that range rather than just the probability of failure within 30. This explains the graded target's role. The proposition characterizes unrestricted BCE minimization; the head restrictions and additional penalties mean that the complete fitted score need not equal $m_q(v)$.

Recovering a warning boundary. Appendix B develops a randomization-based estimator of the farthest lead at which pre-event observations can be distinguished from safe operation. Under independent episodes, exchangeable safe-reference pools beyond the boundary, and a positive separation gap at the boundary, it provides a finite-sample recovery bound.

Controlling changes to the binary score. Equation (2) gives $|z - z^0| \leq C$, so the sigmoid odds change by at most a factor e^C in either direction. At the same numerical cutoff $\eta \in (0, 1)$, a baseline logit more than C away from $\mathrm{logit}(\eta) = \log(\eta/(1-\eta))$ retains its warning decision. This is a deterministic constraint on the head, independent of training convergence. Likewise, the physical branch is nondecreasing in each boundary coordinate when the baseline and residual contribution are held fixed. Appendix A proves these statements. Together, the results explain the target and constrain the score transformation; comparative warning performance remains an empirical question rather than a consequence of these bounds.

6 Experimental setup

Datasets and splits. We consider five forecasters across four prognostic datasets, giving 20 model–dataset pairs. Table 3 summarizes the datasets and evaluation windows. Table 1 compares complete warning systems, while Table 2 compares training objectives using the shared head architecture described in Section 4. For Tiny-N-CMAPSS, we use its 90 training-file engines and form a fixed 20/10/60 training/calibration/evaluation split (alovberg, 2021). NASA Battery Aging contributes seven eligible capacity-threshold events and eight right-censored cells (Saha & Goebel, 2007). XJTU-SY contributes 15 complete bearing trajectories, and FEMTO/PRONOSTIA contributes 17 (Wang et al., 2020; Nectoux et al., 2012). External datasets use rotating splits by physical unit, with each included unit evaluated once per model. All records from one unit remain in one role within a fold. The Appendix gives preprocessing, unit assignments, and event definitions.

Forecasters and comparator. DLinear (Zeng et al., 2023) and iTransformer (Liu et al., 2024) use local implementations; PatchTST (Nie et al., 2023) uses the Hugging Face implementation. TTM (Ekambaram et al., 2024) starts from a publicly available pretrained checkpoint and is fine-tuned; FlowState (Graf et al., 2025) uses a publicly available pretrained checkpoint and keeps it frozen to produce median forecasts. DLinear, PatchTST, iTransformer, and TTM receive 40 forecasting epochs, after which their weights are frozen. The standard head is a 32-unit ReLU multilayer perceptron on the channelwise forecast means and endpoints, trained for 50 epochs with binary cross-entropy. The Table 1 baseline uses a 10-unit binary training target, while event evaluation uses a 30-unit warning interval. The objective comparison in Table 2 includes both near-horizon and full-horizon binary controls.

Metrics. At decision time t, a positive point lies strictly before an observed event and at most $H = 30$ native time units from it; a safe point has no event within H and sufficient observed follow-up. We exclude active-event rows, incomplete contexts, and unobserved tails of censored units. An event is eligible only when its entire warning interval begins after the first scorable point. Event recall is the fraction of eligible events with at least one warning in that interval. FPR is false-warning points divided by safe points; precision is true-warning points divided by all warning points. Recall counts events, while FPR and precision count decision points. No point adjustment is applied, and multiple warnings before one event count separately for precision. We pool integer counts across external folds within each model–dataset pair, then report unweighted means across the 20 pairs. All five models share each dataset's events; the pairs are not independent experimental replications.

6.1 Controls for the contribution of the risk head

Every method in the Table 2 objective comparison uses the pooled empirical 95th percentile of safe calibration scores, with the same tie convention and no additional confirmation rule or score floor, and receive the same information as the proposed head. A common binary predictor for the full warning horizon supplies the frozen offset; the compared objectives train an identically parameterized bounded correction. This design separates a change in supervision from a change in available degradation information.

Table 1: System comparisons. Entries show standard → risk. E: events caught; R: event recall; P: precision. Rates are percentages; bold marks the better rate. Each dataset has n eligible events. iTrans.: iTransformer.

	tiny-N-CMAPSS ($n = 60$)				**NASA Battery** ($n = 7$)			
Backbone	E	R↑	FPR↓	P↑	E	R↑	FPR↓	P↑
DLinear	55 → 59	91.67 → **98.33**	**0.29** → 0.58	**99.44** → 98.96	4 → 6	57.14 → **85.71**	7.84 → **1.76**	65.52 → **88.46**
PatchTST	56 → 60	93.33 → **100.00**	1.85 → **0.81**	96.51 → **98.60**	3 → 5	42.86 → **71.43**	5.49 → **0.78**	60.56 → **93.22**
iTrans.	55 → 59	91.67 → **98.33**	**1.38** → 1.56	97.27 → **97.61**	2 → 5	28.57 → **71.43**	4.71 → **1.18**	56.36 → **90.48**
TTM	56 → 60	93.33 → **100.00**	2.19 → **0.35**	95.86 → **99.45**	1 → 6	14.29 → **85.71**	4.31 → **1.76**	51.11 → **88.89**
FlowState	54 → 59	90.00 → **98.33**	2.88 → **1.04**	94.66 → **98.42**	4 → 5	57.14 → **71.43**	6.08 → **2.35**	53.03 → **82.61**
	XJTU-SY ($n = 15$)				**FEMTO/PRONOSTIA** ($n = 17$)			
Backbone	E	R↑	FPR↓	P↑	E	R↑	FPR↓	P↑
DLinear	10 → 13	66.67 → **86.67**	8.50 → **1.69**	16.86 → **54.66**	12 → 15	70.59 → **88.24**	27.35 → **4.40**	10.05 → **37.47**
PatchTST	10 → 12	66.67 → **80.00**	9.73 → **1.70**	15.73 → **53.48**	12 → 15	70.59 → **88.24**	24.92 → **2.20**	9.07 → **44.05**
iTrans.	8 → 12	53.33 → **80.00**	8.48 → **1.52**	16.99 → **56.62**	12 → 16	70.59 → **94.12**	11.83 → **2.21**	11.76 → **44.15**
TTM	9 → 12	60.00 → **80.00**	8.37 → **1.73**	17.26 → **54.55**	12 → 15	70.59 → **88.24**	25.31 → **2.12**	9.73 → **45.19**
FlowState	9 → 13	60.00 → **86.67**	5.77 → **1.69**	24.74 → **54.38**	12 → 16	70.59 → **94.12**	21.35 → **4.64**	10.12 → **36.99**

Mean (20 pairs): R: 65.98 → **87.35** FPR: 9.43 → **1.80** P: 45.63 → **70.91**

We compare binary cross-entropy at both the near-event and full warning horizons, class-balanced cross-entropy, uniform label smoothing (Szegedy et al., 2016), focal loss (Lin et al., 2017), and exponential temporal label smoothing (TLS) (Yèche et al., 2023). The full-horizon binary control is very important as it tests whether a gain over the original near-event head follows simply from matching the training labels to the evaluation window. TLS uses the same graded target as the proposed objective. The comparison therefore isolates the additional penalties: suppressing risk at zero-target observations, matching changes in the target curve, and discouraging decreases while the target rises. Each component ablation removes one of these penalties while retaining the other two.

7 RESULTS

Table 1 reports all 20 recorded comparisons of the retained complete systems. The retained risk system catches more eligible failures in every pair. Event recall improves in 20 of 20 comparisons, FPR decreases in 18, and precision increases in 19. All three improve together in 18 pairs, giving 57 favorable changes among 60 metric comparisons. Equal-weight means across the pairs increase from 65.98% to 87.35% for event recall and from 45.63% to 70.91% for precision; mean FPR decreases from 9.43% to 1.80%. The corresponding changes are $+21.37$, $+25.28$, and -7.63 percentage points.

The gains differ by dataset. Mean event recall increases from 92.00% to 99.00% on tiny-N-CMAPSS, from 40.00% to 77.14% on NASA batteries, from 61.33% to 82.67% on XJTU-SY, and from 70.59% to 90.59% on FEMTO/PRONOSTIA. The bearing datasets also show substantial reductions in false warnings: mean FPR falls from 8.17% to 1.67% on XJTU-SY and from 22.15% to 3.11% on FEMTO/PRONOSTIA. Their mean precision rises from 18.31% to 54.74% and from 10.15% to 41.57%, respectively.

All three unfavorable changes occur on tiny-N-CMAPSS. For DLinear, false warning points increase from 5 to 10 among 1,734 safe points, raising FPR from 0.29% to 0.58%; precision decreases from 99.44% to 98.96%. For iTransformer, false warning points increase from 24 to 27, raising FPR from 1.38% to 1.56%. Both systems catch four additional failures.

7.1 CONTROLLED OBJECTIVE COMPARISONS

Tables 1 and 2 compare different configurations. Table 1 evaluates complete warning systems, whereas Table 2 holds the input features, correction architecture, and calibration rule fixed to compare objectives. Their corresponding rows therefore need not have identical averages.

Uniform smoothing, class reweighting, and focal loss reduce some of the remaining errors, but temporal supervision gives a stronger result. The full procedure improves on TLS after both methods receive the same graded target.

Table 2: Controlled objectives and component ablations. Entries are equally weighted means across the 20 model–dataset pairs, in percent. All rows train the same bounded correction to a full-horizon binary base and use the same 5% safe calibration target. A shared calibration target does not impose equal evaluation FPR. Bold emphasizes the full procedure.

Objective	Event recall ↑	FPR ↓	Precision ↑
Binary BCE, near horizon	66.53	7.45	50.37
Binary BCE, full horizon	69.96	6.55	52.86
Uniform label smoothing	72.10	6.09	53.90
Class-balanced BCE	74.61	6.25	52.25
Focal loss	76.04	5.50	55.21
Temporal label smoothing	78.93	3.78	60.99
Full risk head	**85.67**	**2.06**	**66.99**
Without false-positive penalty	87.89	4.32	61.25
Without shape-aware smoothness	80.61	1.93	66.27
Without monotonicity penalty	82.74	2.73	64.75

Relative to TLS, the full head adds 6.74 percentage points of event recall and 6.00 points of precision while reducing FPR by 1.72 points. The ablations show why the three metrics must be read together. Removing the false-positive term increases recall to 87.89%, but FPR rises to 4.32%. Removing shape-awareness smoothness lowers recall to 80.61% despite a slightly lower FPR. The penalties have different effects. The false-positive penalty reduces false warnings at some cost to recall, while shape-aware smoothness increases recall and precision with a small increase in FPR. Adding the monotonicity penalty improves all three reported metrics.

8 Discussion

Event coverage and the density of warnings within an event window measure different behavior. On NASA batteries, DLinear catches six failures instead of four even though its true warning points decrease from 76 to 69. It covers more failure events with fewer positive-window alarms. Conversely, repeated warnings near a small number of failures can produce many true warning points while leaving other failures undetected. Reporting event recall alongside pointwise FPR and precision exposes this distinction. Improved event recall does not establish longer warning lead time, which these results do not measure.

The warning objective does not account for how much time an intervention requires. A warning that allows an operator to stop a machine may still arrive too late to schedule maintenance or obtain replacement parts. The graded target reflects proximity to failure, but does not directly reward meeting such deadlines. Future work should evaluate the fraction of failures detected with a specified minimum lead time and incorporate these requirements into training.

9 Conclusion

Accurate forecasts do not by themselves tell us when to issue a failure warning. We address this gap with a risk head that combines frozen forecast summaries and observed degradation features through a shared causal interface. Temporally graded targets and additional penalties guide training, while a bounded correction limits changes to the binary logit. Our analysis explains the quantity learned under graded supervision, establishes bounds on score changes, and studies a separate procedure for selecting a warning boundary. Across five forecasters and four datasets, the risk system catches more failures in all 20 model–dataset pairs, reduces false positive rate in 18, and improves precision in 19. These results show the value of adapting forecasting models specifically for warning decisions.

AI USE STATEMENT

Generative AI tools assisted with experiment design, debugging, developing mathematical arguments, literature searches, and polishing the manuscript. The authors take responsibility for the final text, code, mathematical claims, and reported results.

REPRODUCIBILITY STATEMENT

The full source bundle for the 20 model–dataset system comparisons is provided in the supplementary materials, including data preparation, training, inference, calibration, and metric computation. Appendix C summarizes the software environment, entry points, and required inputs. The theoretical assumptions and proofs are given in the appendix.

REFERENCES

Ilias Abdouni, Alexandre Voisin, and Christophe Cerisara. Leveraging time series foundation models embeddings for remaining useful life prediction. In *PHM Society European Conference*, volume 9, pp. 1–7, 2026. doi: 10.36001/phme.2026.v9i1.4906. URL https://papers.phmsociety.org/index.php/phme/article/view/4906.

alovberg. tiny-N-CMAPSS: A tiny version of the 2021 PHM data challenge dataset. GitHub repository, 2021. URL https://github.com/alovberg/tiny-N-CMAPSS/tree/b915997ff8d571f4e9d091954d6556835f212ded. Revision b915997ff8d571f4e9d091954d6556835f212ded, December 20, 2021.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=gerNCVqqtR.

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

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9268–9277, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Cui_Class-Balanced_Loss_Based_on_Effective_Number_of_Samples_CVPR_2019_paper.html.

Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H. Nguyen, Wesley M. Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (TTMs): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. In *Advances in Neural Information Processing Systems*, volume 37, pp. 74147–74181, 2024. doi: 10.52202/079017-2359. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/874a4d89f2d04b4bcf9a2c19545cf040-Abstract-Conference.html.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://sites.stat.washington.edu/people/raftery/Research/PDF/Gneiting2007jasa.pdf.

Lars Graf, Thomas Ortner, Stanisław Woźniak, and Angeliki Pantazi. FlowState: Sampling rate invariant time series forecasting, 2025. URL https://arxiv.org/abs/2508.05287v1. Version 1, August 7, 2025.

Lars Graf, Thomas Ortner, Stanisław Woźniak, and Angeliki Pantazi. Flowstate: Sampling-rate-equivariant time-series forecasting. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://arxiv.org/abs/2508.05287v3.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 387–395, 2018. doi: 10.1145/3219819.3219845. URL https://doi.org/10.1145/3219819.3219845.

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, pp. 7194–7201, 2022. doi: 10.1609/aaai.v36i7.20680. URL https://ojs.aaai.org/index.php/AAAI/article/view/20680.

Xiang Li, Qian Ding, and Jian-Qiao Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172:1–11, 2018. doi: 10.1016/j.ress.2017.11.021. URL https://doi.org/10.1016/j.ress.2017.11.021.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017. URL https://openaccess.thecvf.com/content_iccv_2017/html/Lin_Focal_Loss_for_ICCV_2017_paper.html.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2ea18fdc667e0ef2ad82b2b4d65147ad-Abstract-Conference.html.

Patrick Nectoux, Rafael Gouriveau, Kamal Medjaher, Emmanuel Ramasso, Brigitte Morello, Noureddine Zerhouni, and Christophe Varnier. PRONOSTIA: An experimental platform for bearings accelerated degradation tests. In *Proceedings of the IEEE International Conference on Prognostics and Health Management*, pp. 1–8, Denver, Colorado, USA, 2012. IEEE. URL https://publiweb.femto-st.fr/tntnet/entries/1528/documents/author/data.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.14730.

Belinda Phipson and Gordon K. Smyth. Permutation p-values should never be zero: Calculating exact p-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1):Article 39, 2010. doi: 10.2202/1544-6115.1585. URL https://gksmyth.github.io/pubs/PermPValuesPreprint.pdf.

B. Saha and K. Goebel. Battery data set. NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA, 2007. URL https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/.

Felix Salfner, Maren Lenk, and Miroslaw Malek. A survey of online failure prediction methods. *ACM Computing Surveys*, 42(3):10:1–10:42, 2010. doi: 10.1145/1670679.1670680. URL https://doi.org/10.1145/1670679.1670680.

Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. Anomaly detection in streams with extreme value theory. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1067–1075, 2017. doi: 10.1145/3097983.3098144. URL https://doi.org/10.1145/3097983.3098144.

Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Seán McLoone, and Alessandro Beghi. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11(3):812–820, 2015. doi: 10.1109/TII.2014.2349359. URL https://doi.org/10.1109/TII.2014.2349359.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016. URL https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.html.

Nesime Tatbul, Tae Jun Lee, Stan Zdonik, Mejbah Alam, and Justin Gottschlich. Precision and recall for time series. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/8f468c873a32bb0619eaeb2050ba45d1-Abstract.html.

Biao Wang, Yaguo Lei, Naipeng Li, and Ningbo Li. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability*, 69(1): 401–412, 2020. doi: 10.1109/TR.2018.2882682. URL https://doi.org/10.1109/TR.2018.2882682.

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*, pp. 39913–39938. PMLR, 2023. URL https://proceedings.mlr.press/v202/yeche23a.html.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317.

Shuai Zheng, Kosta Ristovski, Ahmed Farahat, and Chetan Gupta. Long short-term memory network for remaining useful life estimation. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pp. 88–95. IEEE, 2017. doi: 10.1109/ICPHM.2017.7998311. URL https://doi.org/10.1109/ICPHM.2017.7998311.

A Proofs and further properties of the risk head

A.1 Sampling, eligibility, and the conditional objective

The training loss assigns weight $w_{ik} = 1/(mn_i)$ to observation k of entity i, where m is the number of training entities and n_i is its number of eligible observations. Its population interpretation first samples an entity uniformly, then samples an eligible observation within that entity. All conditional expectations in Proposition 1 use this sampling measure. They need not equal expectations under the distribution obtained by sampling every recorded observation with equal probability.

An eligible observation has a positive time to a subsequent recorded failure or enough follow-up to establish the absence of failure throughout its warning window. Observations without such follow-up are excluded. We use $D = \infty$ for the latter observations when no subsequent failure is recorded. This convention preserves $\mathbf{1}\{D \le u\}$ for $u \le h_{\max}$ and the training target. It makes no claim about the failure time after the observed follow-up.

Let $\ell(p, y) = -y \log p - (1 - y) \log(1 - p)$ and $m = \mathbb{E}[q(D) \mid V = v]$. For $0 < p < 1$, linearity in the target gives

$$\mathbb{E}[\ell(p, q(D)) \mid V = v] = -m \log p - (1 - m) \log(1 - p) \tag{6}$$

$$= H(m) + \mathrm{KL}(\mathrm{Ber}(m) \| \mathrm{Ber}(p)), \tag{7}$$

where $H(m) = -m \log m - (1 - m) \log(1 - m)$, with $0 \log 0 = 0$. The Kullback–Leibler divergence is nonnegative and vanishes only when $p = m$. Thus m is the unique minimizer on $[0, 1]$ under the usual lower-semicontinuous extension of the loss. If $m \in \{0, 1\}$, a finite sigmoid logit

approaches the optimum only as its magnitude tends to infinity. This proves the estimand statement in Proposition 1.

This statement concerns the BCE component over unrestricted measurable scores. The proposed objective includes other terms and optimizes within a restricted head family. The decomposition neither asserts that its fitted scores equal m_q nor provides a consistency result for a selected, regularized head.

A.2 The exponential target as an average over horizons

Put $a = h_{\min}$ and $b = h_{\max}$, with $0 < a < b$. On eligible distances $d > 0$, the exponential TLS target (Yèche et al., 2023) is

$$q(d) = \begin{cases} 1, & 0 < d \le a, \\ \dfrac{e^{-\gamma(d-a)} - e^{-\gamma(b-a)}}{1 - e^{-\gamma(b-a)}}, & a < d < b, \\ 0, & d \ge b. \end{cases} \tag{8}$$

The convention includes $q(\infty) = 0$. For $\gamma > 0$, define

$$\rho_\gamma(u) = \frac{\gamma e^{-\gamma(u-a)}}{1 - e^{-\gamma(b-a)}} \mathbf{1}\{a \le u \le b\}. \tag{9}$$

This density is nonnegative and integrates to one. Integrating from $\max(a, d)$ to b, or inspecting the boundary cases, gives

$$q(d) = \int_a^b \mathbf{1}\{d \le u\} \rho_\gamma(u)\, du. \tag{10}$$

Conditional Tonelli's theorem applies because the integrand is nonnegative. Taking conditional expectations gives Equation equation 5. Since F_v is nondecreasing and ρ_γ integrates to one, the integral lies between $F_v(a)$ and $F_v(b)$. In the limiting case $\gamma = 0$, the target is linear on (a, b) and the same result holds with the uniform density $1/(b - a)$.

The averaging identity allows atoms in the distribution of D. For example, an atom at b contributes to $F_v(b)$ but contributes zero to m_q, since the target is zero at that endpoint. A single scalar $m_q(v)$ therefore cannot in general recover the full conditional failure-time distribution or a particular warning-onset time.

On positive finite distances, q is globally Lipschitz with constant

$$L_q = \frac{\gamma}{1 - e^{-\gamma(b-a)}} \tag{11}$$

for $\gamma > 0$, and $L_q = 1/(b - a)$ for $\gamma = 0$. Its derivative has magnitude at most this constant inside (a, b) and is zero outside, while the function is continuous at a and b. Splitting any interval at those two points proves the global bound. Therefore $|d - d'| \le e$ implies $|q(d) - q(d')| \le L_q e$. This bound characterizes sensitivity of the target to failure-time errors; it does not bound changes in a refitted model.

A.3 Bounded correction of the binary base

Proposition 2 (Effect of the bounded correction). *For finite logits $z = z^0 + C \tanh g_\theta(V)$, $s = \sigma(z)$, and $s^0 = \sigma(z^0)$,*

$$e^{-C} \le \frac{s/(1-s)}{s^0/(1-s^0)} \le e^C, \qquad |s - s^0| \le \tanh(C/4). \tag{12}$$

At a common threshold $\vartheta \in (0, 1)$, a baseline logit below $\operatorname{logit}(\vartheta) - C$ remains below threshold after correction; one at least $\operatorname{logit}(\vartheta) + C$ remains above it.

Write $z = z^0 + \delta$ with $\delta = C \tanh g_\theta(V)$, $C > 0$, $s = \sigma(z)$, and $s^0 = \sigma(z^0)$. Since $|\tanh x| \le 1$ and sigmoid odds equal the exponential of the logit,

$$|z - z^0| \le C, \qquad e^{-C} \le \frac{s/(1-s)}{s^0/(1-s^0)} \le e^C. \tag{13}$$

These bounds hold for any learned correction parameters and do not depend on optimization convergence.

For the score bound, fix a shift $d \geq 0$ and maximize $f(x) = \sigma(x + d) - \sigma(x)$ over x. Its derivative vanishes at $x = -d/2$, and this is its maximum because σ' is symmetric about zero and decreases with absolute argument. At that point,

$$f(-d/2) = 2\sigma(d/2) - 1 = \tanh(d/4). \tag{14}$$

Negative shifts give the same bound by exchanging the two logits. Since $|\delta| \leq C$, this proves $|s - s^0| \leq \tanh(C/4)$. For example, $C = 5$ allows a uniform score difference up to about 0.848; bounding the logit correction does not by itself make every change small.

At a common fixed threshold $\vartheta \in (0, 1)$, let $b_\vartheta = \text{logit}(\vartheta)$. An observation with $z^0 < b_\vartheta - C$ cannot become a warning, while one with $z^0 \geq b_\vartheta + C$ remains a warning, by substituting the extreme allowed corrections. This result requires the same numerical threshold. Calibrating two heads at the same target FPR need not produce the same numerical score threshold, so it alone does not ensure preservation of their warning decisions.

A.4 The physical branch, conditional on the other inputs

Let $r = r_\theta(f, \psi)$ and write the optional branch as

$$g = r + \kappa a_\theta^T c, \qquad a_\theta = (1 - \varepsilon)\,\text{softmax}(\theta) + \varepsilon e_3, \qquad \kappa \geq 0, \quad 0 < \varepsilon < 1. \tag{15}$$

The following statements fix the learned parameters, the baseline logit z^0, and the unrestricted branch value r. Under these conditions,

$$\frac{\partial s}{\partial c_j} = C\kappa a_{\theta,j} s(1 - s)\,\text{sech}^2(g) \geq 0. \tag{16}$$

The forecast coordinate has $a_{\theta,3} \geq \varepsilon$, and its derivative is strictly positive for finite inputs when $\kappa > 0$. No strict influence is guaranteed when $\kappa = 0$. The nonnegative derivative is a consequence of the chain rule and the monotonicity of both sigmoid and hyperbolic tangent.

For two physical coordinate vectors with the other quantities fixed, the derivative bounds for sigmoid and hyperbolic tangent imply

$$|s(c) - s(c')| \leq \frac{C\kappa}{4} |a_\theta^T (c - c')| \leq \frac{C\kappa}{4} \|c - c'\|_\infty. \tag{17}$$

The mixture also has a useful conditional upper envelope. Define

$$U(c) = \varepsilon c_3 + (1 - \varepsilon) \max_j c_j. \tag{18}$$

Because $a_\theta^T c$ is εc_3 plus $(1 - \varepsilon)$ times a convex combination of the three coordinates, $a_\theta^T c \leq U(c)$. Thus

$$s(c) \leq \sigma\big(z^0 + C \tanh[r + \kappa U(c)]\big). \tag{19}$$

This bounds what changes to the mixture weights alone can achieve when the other branches are fixed. It is not an upper bound over all unrestricted branch parameters.

The baseline can itself read the physical coordinates, and raw observations can affect both r and c. Their total effect therefore need not satisfy Equation equation 16. Clipping can also make derivatives with respect to a raw forecast zero. Neither the branch constraint nor the positive forecast weight guarantees that forecasts contribute useful information, or that scores increase with observation time.

A.5 Scope of the score analysis

The results above characterize targets and score families conditional on stated inputs and thresholds. Event recall depends on scores across each warning window; false positive rate and precision additionally depend on safe scores, event prevalence, and the policy. The properties do not order all three metrics between two fitted systems. The onset-estimation result below uses separate assumptions and does not establish a selected-head generalization guarantee.

B AN ESTIMATOR OF WARNING ONSET

The following construction estimates how far before failure a chosen abnormality score separates from a safe reference.

B.1 RECOVERING A WARNING BOUNDARY.

The onset selector asks how far before failure an abnormality score can be distinguished from safe operation. It uses N failure episodes. At each lead $\ell \in \{1, \dots, H\}$, measured in a fixed time unit, a randomization test compares the pre-event state with safe reference states from the same episode. The test repeatedly exchanges which state is treated as the candidate and recomputes a score in $[0, 1]$ that does not depend on the ordering of the reference states. A small plus-one Monte Carlo value $p_{N,B}(\ell)$ indicates separation from the safe reference. With B resampling replicates and testing level α, the selector returns the farthest rejected lead,

$$\widehat{h} = \max\{\ell : p_{N,B}(\ell) \le \alpha/H\},$$

or zero if there is no rejection. The construction below defines the test precisely. Its randomization score is distinct from the risk-head logit. Let $h^\star$ denote the farthest distinguishable lead. Beyond this boundary, we assume joint exchangeability: swapping the candidate state with a safe reference leaves their joint distribution unchanged. At the boundary, $\delta > 0$ denotes a lower bound on the mean difference between observed and randomized scores.

Theorem 1 (Recovery under exact safe nulls)**.** *Suppose the N episodes are independent, randomization scores lie in $[0, 1]$, and pooled states are jointly exchangeable at every lead beyond $h^\star \in \{1, \dots, H\}$. Suppose the mean population gap between the observed and randomized scores at $h^\star$ is at least $\delta > 0$. For the stated randomization construction, if $B \ge 4H/\alpha$, then*

$$\Pr(\widehat{h} \ne h^\star) \le \alpha + \frac{4H}{\alpha} e^{-N\delta^2/2} + e^{-\alpha B/(12H)}. \tag{20}$$

If all tested leads satisfy the null, $\Pr(\widehat{h} \ne 0) \le \alpha$.

For example, a boundary at lead 20 means that more distant leads satisfy the safe null, while lead 20 has a detectable population difference. The theorem bounds the chance of selecting the wrong boundary. Its three terms account for false rejections beyond the boundary, limited independent episodes, and finite resampling. More episodes help when the gap is positive; more resampling reduces Monte Carlo error.

This is a finite-sample guarantee for boundary selection under the stated assumptions. It neither selects both target endpoints nor guarantees the warning classifier's event recall, FPR, or precision.

B.2 RANDOMIZATION CONSTRUCTION

For each of N failure episodes and a lead $\ell \in \{1, \dots, H\}$, form a pool

$$\mathcal{P}_{j\ell} = (Y_{j\ell 0}, Y_{j\ell 1}, \dots, Y_{j\ell m}), \tag{21}$$

where index zero is the state at lead ℓ and the remaining $m \ge 1$ states are from the specified safe reference. Fix a scoring rule $g(y; \mathcal{R}) \in [0, 1]$ that is invariant to the ordering of its reference collection $\mathcal{R}$. For each possible held-out index, compute

$$a_{j\ell}(r) = g\big(Y_{j\ell r}; \{Y_{j\ell q} : q \ne r\}\big), \qquad r = 0, \dots, m. \tag{22}$$

The construction recomputes reference-dependent fitting and normalization for each held-out index. Externally learned representations and scoring parameters are fixed independently of the sampled episodes. The statements below are conditional on those fixed quantities.

The observed statistic is $A^0_{N\ell} = N^{-1} \sum_j a_{j\ell}(0)$. For Monte Carlo replicate b, draw $U^b_{j\ell}$ independently and uniformly from $\{0, \dots, m\}$ for every episode, and set $A^b_{N\ell} = N^{-1} \sum_j a_{j\ell}(U^b_{j\ell})$. Replicates are independent conditional on the episode data and are sampled with replacement. Define the conservative plus-one value (Phipson & Smyth, 2010)

$$p_{N,B}(\ell) = \frac{1 + \sum_{b=1}^{B} \mathbf{1}\{A^b_{N\ell} \ge A^0_{N\ell}\}}{B + 1}. \tag{23}$$

For a prespecified $0 < \alpha < 1$, put $t = \alpha/H$ and

$$\widehat{h} = \max\{\ell : p_{N,B}(\ell) \le t\}, \tag{24}$$

with the maximum of the empty set defined as zero. The reference periods, score, lead grid, and testing level are fixed before examining these tests. The construction does not require tests at different leads to be independent.

B.3 Assumptions and finite-sample bound

Assume that episode data are independent across j. Within an episode, the safe states need not be independent. Suppose $h^\star \in \{0, \dots, H\}$ satisfies exact joint exchangeability of $(Y_{j\ell 0}, \dots, Y_{j\ell m})$ at every $\ell > h^\star$. This is stronger than equality of marginal distributions or stationarity of a time series. If $h^\star > 0$, require a positive population gap at that boundary:

$$\frac{1}{N}\sum_{j=1}^{N} \mathbb{E}\left[a_{jh^\star}(0) - a_{jh^\star}(U_j)\right] \ge \delta > 0, \tag{25}$$

where the U_j are independent uniform indices and the expectation includes both episode data and resampling. Episode distributions may differ. No separation assumption at all shorter leads is needed to recover the maximum rejected lead.

These assumptions imply Theorem 1. If $h^\star = 0$, the false-positive bound holds for every Monte Carlo budget B.

Proof. First consider a lead beyond $h^\star$. Conditional on each unordered episode pool, joint exchangeability makes the observed candidate index uniform among its $m + 1$ positions. Symmetry of g in the reference states ensures that relabeling acts only by changing which state is held out. Across independent episodes the vector of observed held-out indices is uniform on the product of these positions. The independent Monte Carlo vectors have the same conditional distribution. Thus $A^0_{N\ell}, A^1_{N\ell}, \dots, A^B_{N\ell}$ are exchangeable conditional on the unordered pools. Ties handled with the weak comparison in Equation equation 23 give a conservative rank: for every $u \in [0, 1]$, $\Pr(p_{N,B}(\ell) \le u) \le u$. A union bound over the at most H null leads gives

$$\Pr(\exists \ell > h^\star : p_{N,B}(\ell) \le t) \le Ht = \alpha. \tag{26}$$

This already proves the $h^\star = 0$ case.

For $h^\star > 0$, let $\mathcal{D}$ contain the labeled episode pools at that lead, and define their exact randomization tail probability

$$Q_N = \Pr(A^1_{Nh^\star} \ge A^0_{Nh^\star} \mid \mathcal{D}). \tag{27}$$

For one independent resampling vector, set $X_j = a_{jh^\star}(0) - a_{jh^\star}(U_j)$. The X_j are independent across episodes, belong to $[-1, 1]$, and have average expectation at least δ. Hoeffding's bounded-sum inequality (Hoeffding, 1963) gives

$$\mathbb{E}[Q_N] = \Pr\left(\frac{1}{N}\sum_j X_j \le 0\right) \le \exp(-N\delta^2/2). \tag{28}$$

For completeness, the exponential-moment bound is

$$\Pr\left(\sum_j (X_j - \mathbb{E}X_j) \le -N\delta\right) \le \inf_{\lambda > 0} \exp(-\lambda N\delta + N\lambda^2/2) = e^{-N\delta^2/2}, \tag{29}$$

where the quadratic term uses interval length two for each X_j. Markov's inequality therefore yields

$$\Pr(Q_N > t/4) \le \frac{4}{t} e^{-N\delta^2/2}. \tag{30}$$

Conditional on $\mathcal{D}$, the count $C = \sum_{b=1}^{B} \mathbf{1}\{A^b_{Nh^\star} \ge A^0_{Nh^\star}\}$ is binomial with parameters (B, Q_N). On $Q_N \le t/4$, it is stochastically dominated by $C' \sim \mathrm{Bin}(B, t/4)$. Put $M = Bt/4$. If the plus-one

test fails to reject, then $C > t(B+1) - 1 \geq tB/2 = 2M$, using $B \geq 4/t$. The exponential Markov bound at $\lambda = \log 2$ gives

$$\Pr(C' \geq 2M) \leq e^{-2M \log 2}(1 - t/4 + (t/4)e^{\log 2})^B \tag{31}$$

$$\leq e^{-(2\log 2 - 1)M} \leq e^{-M/3} = e^{-Bt/12}. \tag{32}$$

Together with Equation equation 30, this bounds the probability of failing to reject at $h^\star$ by $(4/t)e^{-N\delta^2/2} + e^{-Bt/12}$. If that lead is rejected and no larger lead is rejected, the maximum in Equation equation 24 equals $h^\star$, irrespective of decisions at shorter leads. Combining this fact with Equation equation 26 and substituting $t = \alpha/H$ proves Equation equation 20. □

B.4 Consistency and the role of a gradual boundary

For fixed H and $\delta > 0$, the recovery bound converges to zero if $\alpha_N \to 0$, $\log(H/\alpha_N) = o(N\delta^2)$, and $\alpha_N B_N/H \to \infty$, with the required lower bound on B_N eventually satisfied. For example, $\alpha_N = N^{-1}$ and $B_N = \lceil HN \log^2 N \rceil$ satisfy these conditions for sufficiently large N. Holding α fixed leaves a nonvanishing family-wise error bound and does not by itself establish consistency.

The same argument gives a localization statement under a specified transition band. Let integers $1 \leq \ell_- \leq \ell_+ \leq H$ satisfy exact exchangeability at all $\ell > \ell_+$ and a gap $\delta_- > 0$ of the form in Equation equation 25 at ℓ_-. Then

$$\Pr(\widehat{h} \notin [\ell_-, \ell_+]) \leq \alpha + \frac{4H}{\alpha} e^{-N\delta_-^2/2} + e^{-\alpha B/(12H)}. \tag{33}$$

Indeed, rejection at ℓ_- prevents underestimating the band, and the null tests above ℓ_+ control overestimation. A rate for the band's width requires an additional quantitative relation between its width and δ_-, as well as a specified lead-time grid. A universal $N^{-1/2}$ localization rate does not follow from exchangeability and a positive gap alone.

C Reproducibility details

Table 3: Context and forecast lengths count observed sequence rows. A bearing row summarizes 1 acquired waveform: XJTU-SY samples once per minute and FEMTO once per 10 seconds. FEMTO's 8-step forecast spans 80 seconds. Event counts refer to the aggregate evaluation cohort.

Dataset	Input channels	Context	Forecast	Events	Warning interval
Tiny-N-CMAPSS	18	16	8	60	30 flights
NASA Battery Aging	42	8	8	7	30 discharge cycles
XJTU-SY	45	8	8	15	30 minutes
FEMTO/PRONOSTIA	34	16	8	17	30 minutes

- **Evaluation scope.** The replay command evaluates saved standard and risk heads on recorded forecast representations. It recomputes calibration and metrics without retraining the forecasters. Reference counts are checked after scoring and do not replace computed results.
- **Aggregation.** Counts are pooled across external folds within each model–dataset pair. Event recall counts distinct eligible events; FPR and precision count observations. Reported means give each of the 20 pairs equal weight. No point adjustment is applied.

C.1 Selection and calibration

For the system comparison in Table 1, tiny-N-CMAPSS selects among 12 head recipes using leave-one-engine-out calibration. The external datasets select six head recipes and three threshold policies using partitions of the outer training units, with the forecaster fixed across these internal comparisons. Final cutoffs are calculated from safe calibration scores. The standard head uses a nominal

Table 4: Software and execution settings for the supplied code.

Item	Setting
Verified replay environment	Python 3.13.5, NumPy 2.3.1, PyTorch 2.7.1; CPU with four threads.
Training dependencies	pandas 2.3.1, SciPy 1.16.1, Transformers 4.56.0, scikit-learn 1.7.2. The pretrained models additionally require the `tsfm_public` implementation.
Pretrained model inputs	TTM and FlowState checkpoint locations are supplied through `RISK_TTM_CHECKPOINT` and `RISK_FLOWSTATE_CHECKPOINT`; checkpoint revisions remain specified in the source.
Training defaults	The supplied preparation scripts use 40 forecasting epochs where fitting is required, followed by 50 epochs for the standard head. FlowState remains frozen. Risk-head schedules are specified by the respective training runners.
Random seeds	tiny-N-CMAPSS model seed: 17; split seeds: 405 and 913. External model seed: $3072 + 100f$, where f is the outer-fold index.
Integrity checks	Input hashes, disjoint unit assignments, saved selections, and metric counts are checked by the corresponding runners.

Table 5: Entry points in the supplementary source bundle. Paths are relative to the extracted code directory.

Task	Entry point or location
Install dependencies	`python install.py`; add `-training` for training dependencies and `-tsfm-source` for the pretrained-model package source.
Replay all 20 pairs	`python run.py -assets support.zip -output results/replay`
Train tiny-N-CMAPSS	`python training/run.py -data-root data -output runs/ncmapss`
External training sources	`training/work/external_v6/`; dataset-specific preparation, head fitting, selection, and evaluation runners.
Recorded external splits	`training/protocol_settings.py`; `training/write_protocols.py` binds processed files to these splits and checks their hashes.
Evaluation outputs	`results.csv`, per-dataset scores, measured counts, calibration settings, and `verification.json`.

5% calibration FPR target; the engine risk head uses 2.5%. External risk systems use per-entity calibration policies at nominal levels of 5% or 10%, together with their dataset-specific warning rules. The controlled objective comparison in Table 2 instead uses the common calibration procedure in Section 4.4.