WHEN TO TRUST THE FORECAST: POST-HOC FAILURE RISK ESTIMATION FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term time series forecasting requires more than low average error: in deployment, users often need to assess whether a generated trajectory is likely to contain large forecast errors. Standard evaluation protocols average errors over samples, variables, and horizons, thereby obscuring severe localized errors that may occur within an otherwise accurate forecast. We formulate post-hoc forecast failure risk estimation for long-term time series forecasting, where a fixed forecasting backbone, its historical input window, and its generated trajectory are used to estimate the risk of large forecast errors before the ground truth is observed. We propose TSGuard, a backbone-agnostic model that addresses this problem by fusing historical dynamics, forecast behavior, and diagnostic mismatches between the history and forecast trajectory. Concretely, TSGuard decomposes the history into trend and residual tokens, encodes the predicted trajectory with temporal-difference features, constructs diagnostic tokens from mismatch signals such as slope, volatility, range, jump, and lead-time position, and uses a Failure Decoder with Horizon Queries to produce fine-grained risk estimates along the forecast trajectory. By turning post-hoc forecast assessment from sequence-level scoring into fine-grained failure risk estimation along the forecast trajectory, TSGuard provides actionable risk signals for long-term forecasting systems.

1 INTRODUCTION

Long-term time series forecasting (LTSF) plays a critical role in domains such as energy management, transportation, meteorology, finance, and industrial monitoring (Ma et al., 2025; Lin et al., 2026; Ma et al., 2026a). While recent advances—spanning efficient Transformer variants and lightweight architectures (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022; Zeng et al., 2023; Nie et al., 2023; Liu et al., 2024b; 2025a)—have substantially pushed the state of the art on standard benchmarks, model evaluation remains overwhelmingly dominated by aggregate metrics such as Mean Absolute Error (MAE) and Mean Squared Error (MSE) (Cerqueira et al., 2025; Kudrat et al., 2025; Wang et al., 2025a). These metrics are typically averaged across samples, channels, and prediction horizons, collapsing heterogeneous forecasting behavior into a single scalar score. Although effective for global model ranking, such aggregation can mask substantial performance variations across forecasting horizons and overlook localized structural discrepancies or large errors at individual time steps (Cerqueira et al., 2025; Kudrat et al., 2025). In real-world deployment, a pivotal yet under-addressed question therefore arises: not merely which model exhibits lower average error overall, but whether an already generated forecast can be trusted and where its most severe risks lie.

This limitation motivates the need for fine-grained post-hoc forecast reliability assessment. Unlike sequence-level indicators that fail to pinpoint error localization within a multi-step trajectory, fine-grained, step-by-step risk profiles provide actionable reliability signals for downstream decision-making. While conceptually adjacent to uncertainty quantification, post-hoc failure risk estimation addresses a distinct objective: rather than estimating the predictive distribution or interval coverage of ground-truth future values, it diagnostically assesses the likelihood of model-specific failure modes for a fixed point forecaster. Consequently, the output is not a predictive distribution over future values, but a fine-grained risk profile indicating where the existing point forecast is likely to exceed a task-defined error tolerance.

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.

Accordingly, we formulate post-hoc forecast failure risk estimation for time series forecasting. Given a frozen forecasting backbone, a historical context window, and the resulting forecast trajectory, the goal is to estimate whether and at which forecast steps severe errors are likely to occur, before the corresponding ground truth becomes available. The risk estimator operates as a second-stage diagnostic model, using only information available at inference time: historical observations, the forecast trajectory, and optional temporal covariates. A central challenge is that failure cues arise from multiple sources. Beyond irregularities in the observed history or implausible patterns in the forecast alone, risk may emerge from structural mismatches between the historical context and the predicted trajectory, requiring the estimator to model their relationship.

We propose TSGuard, a backbone-agnostic model for post-hoc forecast failure risk estimation. The core idea of TSGuard is to estimate forecast failure risk by fusing three complementary sources of evidence: historical dynamics, forecast behavior, and diagnostic mismatches. To capture these sources, TSGuard first uses a **Historical Pattern Encoder** to decompose the input window into trend and residual components, modeling slow regime evolution and high-frequency instability. It then applies a **Forecast Behavior Encoder** to represent the backbone's predicted trajectory together with temporal-difference features, exposing smoothing, variation, and curvature patterns in the forecast itself. In parallel, a **Diagnostic Risk Feature Encoder** constructs mismatch features along the forecast trajectory, including slope deviation, volatility discrepancy, range excess, recent jump strength, and lead-time position. Finally, a **Failure Decoder with Horizon Queries** uses cross-attention to aggregate evidence from the fused memory and outputs fine-grained large-error risk estimates along the forecast trajectory. Large-error labels are defined using validation-derived lead-time-specific percentile thresholds, accounting for heterogeneous error scales across the forecast trajectory while remaining fixed before test evaluation. Our main contributions are:

- We open a **new direction** for long-term time series forecasting, post-hoc forecast failure risk estimation, which shifts the focus from average accuracy to where a fixed backbone is likely to incur severe errors, using only information available at forecast time.
- We propose TSGuard, a backbone-agnostic failure risk model that combines decomposed historical pattern encoding, forecast behavior encoding, explicit diagnostic mismatch features, and a cross-attention decoder with horizon queries.
- We construct failure labels using validation-derived lead-time-specific percentile thresholds, accounting for heterogeneous error scales across the forecast trajectory while ensuring leakage-free test-time evaluation.
- Extensive experiments across seven datasets and five forecasting backbones demonstrate that TSGuard consistently identifies failure-prone predictions across heterogeneous forecasting settings, enabling effective risk-based retention that reduces forecasting MSE by **22.6%** on average and up to **30.6%**.

2 Related Work

Time series forecasting has become a fundamental task in many real-world applications, attracting sustained attention from the research community (Miller et al., 2024; Guo et al., 2026; Zhao et al., 2026). A major line of recent work has centered on designing more effective forecasting architectures, with particular emphasis on temporal representation learning. Existing methods can be broadly grouped into several representative families.

First, Transformer-based models remain one of the dominant paradigms for long-term forecasting, mainly because self-attention provides a powerful mechanism for capturing long-range temporal dependencies (Wu et al., 2021; Ma et al., 2025; Zhang et al., 2025; Niu et al., 2026). To further enhance their forecasting capability, numerous studies have incorporated temporal decomposition (Ma et al., 2026b), frequency-domain modeling (Zhou et al., 2022), patch-wise representation learning (Nie et al., 2023), and improved embedding designs (Zhang et al., 2025). These extensions are particularly useful for characterizing periodicity, trend variation, and multi-scale temporal structures. Second, MLP-based models have emerged as an efficient alternative. Despite their simple architectures, they often achieve competitive forecasting accuracy with substantially lower computational cost (Lin et al., 2024; Stitsyuk & Choi, 2025; Murad et al., 2025). For instance, PatchMLP (Tang & Zhang, 2025) adopts a block-wise MLP design and performs channel mixing to enable cross-

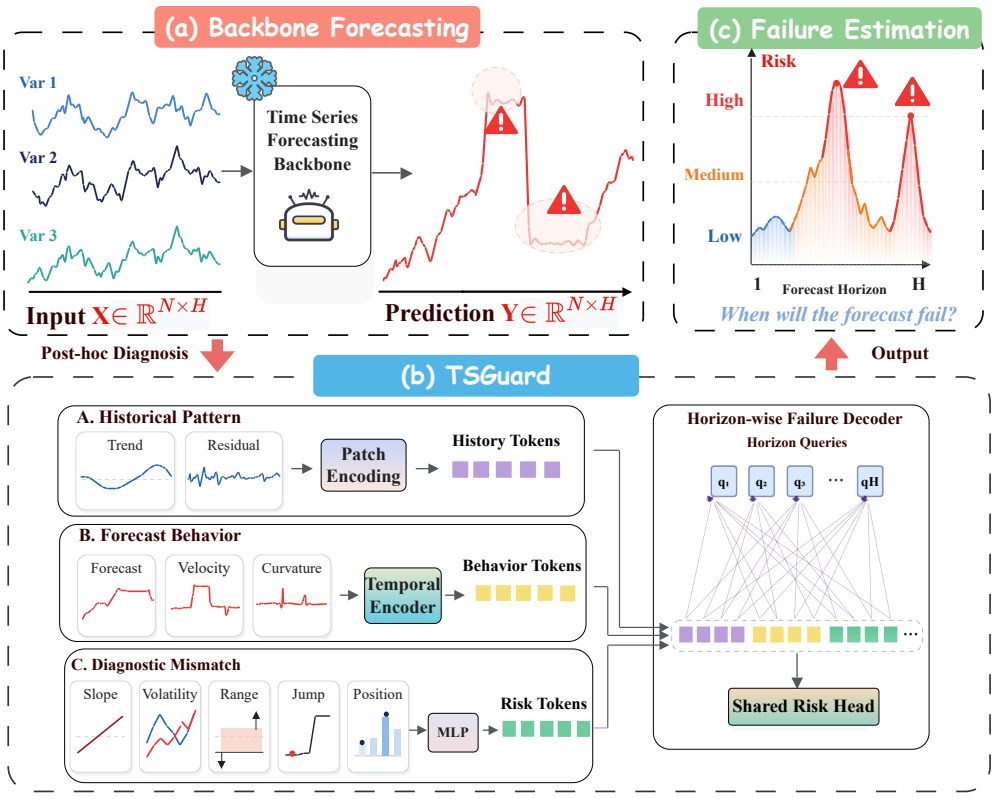

Figure 1: Overall architecture of our model. TSGuard identifies the failure risk of predictions generated by time-series forecasting backbones (e.g., DLinear and iTransformer).

variable information exchange. Third, TCN-based approaches exploit convolutional operations to model local temporal contexts, offering a favorable balance between efficiency and sequence modeling capacity (Wu et al., 2023; Wang et al., 2026).

Beyond these mainstream architectures, recent studies have also explored new modeling paradigms, including Mamba-based models (Wang et al., 2025b), KAN-based networks (Huang et al., 2025), and large language model-based forecasting frameworks (Jin et al., 2023). In parallel, complementary learning strategies such as multimodal learning (Liu et al., 2024a), self-supervised pretraining (Zhang et al., 2024), and diffusion-based generative modeling (Tashiro et al., 2021) have been introduced to enrich time series forecasting research from different perspectives.

3 Problem Setup and Notation

Backbone Forecasting. For a time series forecasting case, let $\boldsymbol{X} \in \mathbb{R}^{L\times C}$ denote the historical input window with length L and C observed variables, and let $\boldsymbol{Y} \in \mathbb{R}^{H\times D}$ denote the future target window with prediction length H and D target variables. A trained forecasting backbone f_ϕ first produces a point forecast $\widehat{\boldsymbol{Y}} \in \mathbb{R}^{H\times D}$.

Post-hoc Forecast Failure Risk Estimation. Given this fixed forecast $\widehat{\boldsymbol{Y}}$, TSGuard learns a failure risk model g_θ that outputs a fine-grained risk profile over the forecast trajectory:

$$\widehat{\boldsymbol{Y}} = f_\phi(\boldsymbol{X}), \qquad \widehat{\boldsymbol{p}} = g_\theta(\boldsymbol{X}, \widehat{\boldsymbol{Y}}, \mathcal{T}) \in [0,1]^H. \tag{1}$$

Here $\widehat{\boldsymbol{p}} = [\widehat{p}_1, \ldots, \widehat{p}_H]$, with $\widehat{p}_h$ indicating the failure risk assigned to the backbone forecast at lead time h. The optional covariate $\mathcal{T} \in \mathbb{Z}^{H\times 2}$ encodes future calendar information, i.e., hour-of-day and day-of-week. At inference, TSGuard only takes $\boldsymbol{X}$, $\widehat{\boldsymbol{Y}}$, and optional $\mathcal{T}$ as inputs, ensuring that the true future $\boldsymbol{Y}$ is never used.

Failure Supervision Label. Failure supervision is defined as a large-error event of the fixed backbone. For a forecast case, the error at lead time h is computed by averaging absolute error over target variables:

$$e_h = \frac{1}{D}\sum_{d=1}^{D}\left|\widehat{Y}_{h,d} - Y_{h,d}\right|, \qquad h = 1, \ldots, H. \tag{2}$$

Since long horizons naturally have different error scales from short horizons, TSGuard uses a separate validation-derived threshold for each lead time:

$$\delta_h = \mathrm{Percentile}_\tau\left(\{e_{m,h} : m \in \mathcal{V}\}\right), \qquad z_h = \mathbb{I}[e_h > \delta_h]. \tag{3}$$

Here $e_{m,h}$ denotes the error of validation case m at lead time h, and $\mathcal{V}$ is the validation split. The binary label $z_h \in \{0, 1\}$ indicates whether the already generated forecast at lead time h exceeds the validation-derived failure tolerance. To avoid optimistic in-sample errors, training-split forecasts are generated out-of-fold in temporal order.

4 METHOD

4.1 OVERALL FRAMEWORK

As shown in Figure 1 and Appendix A, TSGuard is a backbone-agnostic framework for post-hoc forecast failure risk estimation that can be applied to arbitrary time series forecasting backbones. To estimate this risk, TSGuard fuses three deployable sources of evidence: historical dynamics, forecast behavior, and explicit history–forecast mismatch diagnostics, and then decodes them with horizon queries into a fine-grained risk profile. The architecture contains four modules. The **Historical Pattern Encoder** takes $\boldsymbol{X}$ as input and outputs historical memory tokens $\boldsymbol{H}_x \in \mathbb{R}^{2N\times d}$, where $N = \lceil L/P \rceil$ is the number of patches per stream and d is the hidden dimension. The **Forecast Behavior Encoder** takes $\widehat{\boldsymbol{Y}}$ as input and outputs forecast-behavior tokens $\boldsymbol{H}_y \in \mathbb{R}^{H\times d}$ that summarize the shape and dynamics of the generated trajectory. The **Diagnostic Risk Feature Encoder** takes $\boldsymbol{X}$, $\widehat{\boldsymbol{Y}}$, and lead-time indices as input and outputs risk tokens $\boldsymbol{H}_r \in \mathbb{R}^{H\times d}$ built from explicit mismatch features along the forecast trajectory. The **Failure Decoder with Horizon Queries** attends over the fused memory $\boldsymbol{M} = [\boldsymbol{H}_x; \boldsymbol{H}_y; \boldsymbol{H}_r] \in \mathbb{R}^{S\times d}$ and outputs $\widehat{\boldsymbol{p}} \in [0, 1]^H$.

4.2 HISTORICAL PATTERN ENCODER

To separate slow movement from local instability, TSGuard first applies a moving-average filter to the input history $\boldsymbol{X}$ to obtain its trend component, and then subtracts this trend from $\boldsymbol{X}$ to obtain the residual component. Then, both streams have shape $L \times C$. Each stream is split into $N = \lceil L/P \rceil$ patches of length P, where the n-th patch $\boldsymbol{P}_n^{(b)} \in \mathbb{R}^{P\times C}$ belongs to branch $b \in \{\mathrm{trend}, \mathrm{res}\}$. After flattening, a patch is projected into a d-dimensional token and enriched with a positional embedding:

$$\boldsymbol{e}_n^{(b)} = \boldsymbol{W}_p\,\mathrm{vec}(\boldsymbol{P}_n^{(b)}) + \boldsymbol{\pi}_n, \qquad \boldsymbol{e}_n^{(b)} \in \mathbb{R}^d. \tag{4}$$

The trend and residual token sequences are then encoded by separate Transformer encoders. Stream-type embeddings mark which component each token comes from, and the two streams are concatenated into the historical memory:

$$\boldsymbol{H}_x = \mathrm{LN}\left(\left[\mathrm{Enc}_{\mathrm{tr}}(\boldsymbol{E}^{\mathrm{trend}}) + \boldsymbol{c}_{\mathrm{tr}}; \mathrm{Enc}_{\mathrm{res}}(\boldsymbol{E}^{\mathrm{res}}) + \boldsymbol{c}_{\mathrm{res}}\right]\right), \qquad \boldsymbol{H}_x \in \mathbb{R}^{2N\times d}. \tag{5}$$

This representation gives the decoder separate access to persistent directional behavior and high-frequency deviations, instead of forcing both signals into a single raw sequence embedding.

4.3 FORECAST BEHAVIOR ENCODER

The Forecast Behavior Encoder is included because the backbone prediction can reveal failure signals by itself. A forecast that becomes too smooth, jumps inconsistently, loses amplitude, or degenerates at long horizons may carry high failure risk even if the historical window looks ordinary. The input to this module is $\widehat{\boldsymbol{Y}} \in \mathbb{R}^{H\times D}$. TSGuard augments it with first- and second-order finite differences, which expose local slope and curvature:

$$\Delta\widehat{\boldsymbol{Y}}_h = \widehat{\boldsymbol{Y}}_h - \widehat{\boldsymbol{Y}}_{h-1}, \qquad \Delta^2\widehat{\boldsymbol{Y}}_h = \Delta\widehat{\boldsymbol{Y}}_h - \Delta\widehat{\boldsymbol{Y}}_{h-1}. \tag{6}$$

The first horizon is zero-padded for the differences. Concatenating the value, slope, and curvature terms gives $\boldsymbol{F} = [\widehat{\boldsymbol{Y}}, \Delta\widehat{\boldsymbol{Y}}, \Delta^2\widehat{\boldsymbol{Y}}] \in \mathbb{R}^{H\times 3D}$. A linear projection, positional embedding, and Transformer encoder then produce forecast-behavior tokens:

$$\boldsymbol{H}_y = \mathrm{LN}\left(\mathrm{Transformer}\left(\boldsymbol{W}_f\boldsymbol{F} + \boldsymbol{\Pi}\right)\right), \qquad \boldsymbol{H}_y \in \mathbb{R}^{H\times d}. \tag{7}$$

These tokens allow the failure decoder to reason about the forecast trajectory as an object generated by the backbone, rather than treating it merely as a set of future values.

4.4 Diagnostic Risk Feature Encoder

The diagnostic branch injects interpretable mismatch evidence along the forecast trajectory. For each lead time h, it constructs a feature vector $\boldsymbol{r}_h \in \mathbb{R}^K$, where $K = 16$ in our model. The features summarize historical statistics, forecast-prefix statistics, and history-forecast mismatches, including recent volatility, recent maximum jump, historical slope, forecast displacement from the last observed value, range excess outside historical extrema, first- and second-difference magnitudes, prefix forecast volatility, and normalized horizon position.

Two representative features are slope mismatch and volatility mismatch. Let s^{hist} denote the recent historical slope estimated from the last k observed steps. The slope mismatch compares the forecast displacement at horizon h with this recent trend:

$$r_h^{\mathrm{slope}} = \left| \frac{1}{D}\sum_{d=1}^{D} \frac{\widehat{Y}_{h,d} - X_{L,d}}{h} - s^{\mathrm{hist}} \right|. \tag{8}$$

The volatility mismatch compares the standard deviation of the forecast prefix with the recent historical volatility:

$$r_h^{\mathrm{vol}} = \left| \mathrm{Std}(\widehat{\boldsymbol{Y}}_{1:h}) - \mathrm{Std}(\boldsymbol{X}_{L-k+1:L}) \right|. \tag{9}$$

Stacking the diagnostic vectors across all lead times yields $\boldsymbol{R} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_H] \in \mathbb{R}^{H\times K}$. An MLP maps these features into the same hidden space as the other memory tokens:

$$\boldsymbol{H}_r[h] = \mathrm{MLP}(\boldsymbol{r}_h), \qquad \boldsymbol{H}_r \in \mathbb{R}^{H\times d}. \tag{10}$$

This branch is deliberately simple: it gives TSGuard explicit mismatch cues that are easy to inspect, while the learned encoders capture richer temporal patterns.

4.5 Failure Decoder with Horizon-to-Memory Attention

The decoder turns the fused evidence into fine-grained failure risk scores. Rather than pooling all memory tokens into one confidence score, TSGuard assigns each future step a query vector. This preserves temporal specificity by allowing each lead time to attend to different historical, forecast-side, and diagnostic evidence. If timestamp covariates are available, the query combines a learnable horizon embedding with hour-of-day and day-of-week embeddings:

$$\boldsymbol{a}_h = \mathrm{LN}\left(\mathrm{Emb}_{\mathrm{hour}}(\mathrm{hour}_h) + \mathrm{Emb}_{\mathrm{dow}}(\mathrm{dow}_h)\right), \qquad \boldsymbol{q}_h = \mathrm{Emb}_{\mathrm{horizon}}(h) + \boldsymbol{a}_h. \tag{11}$$

Here $\boldsymbol{q}_h \in \mathbb{R}^d$ and $\boldsymbol{Q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_H] \in \mathbb{R}^{H\times d}$. Without timestamp features, $\boldsymbol{q}_h$ reduces to the horizon embedding. We introduce a Horizon-to-Memory Attention (HMA) module, where each future horizon query adaptively retrieves the most relevant historical and variate-aware information from the fused memory $\boldsymbol{M}$. Formally,

$$\boldsymbol{A} = \mathrm{HMA}(\boldsymbol{Q}, \boldsymbol{M}, \boldsymbol{M}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{M}^\top}{\sqrt{d}}\right)\boldsymbol{M}, \qquad \boldsymbol{A} \in \mathbb{R}^{H\times d}. \tag{12}$$

To obtain stable horizon-specific representations, we preserve horizon semantics through residual connections and refine the memory-enhanced features with normalization and a position-wise feed-forward network:

$$\boldsymbol{U} = \mathrm{LN}(\boldsymbol{Q} + \boldsymbol{A}), \qquad \boldsymbol{Z} = \mathrm{LN}(\boldsymbol{U} + \mathrm{FFN}(\boldsymbol{U})), \qquad \boldsymbol{Z} \in \mathbb{R}^{H\times d}. \tag{13}$$

Finally, instead of assigning a separate predictor to each horizon, we use a shared prediction head to enforce a consistent risk-scoring function across all future steps and improve parameter efficiency:

$$\widehat{p}_h = \sigma(\boldsymbol{w}^\top \boldsymbol{Z}_h + b), \qquad h = 1, \ldots, H. \tag{14}$$

4.6 Training Objective of TSGuard

Failure labels are typically imbalanced because only the upper tail of validation errors is marked as failure. TSGuard is trained with weighted binary cross-entropy, where α_h is the positive-class weight computed separately from the training labels at each lead time:

$$\mathcal{L}_{\mathrm{WBCE}} = -\frac{1}{H}\sum_{h=1}^{H}\left[\alpha_h z_h \log \widehat{p}_h + (1 - z_h)\log(1 - \widehat{p}_h)\right]. \tag{15}$$

5 Experiment

5.1 Experiment Setup

Dataset We evaluate the effectiveness of our TSGuard on multiple real-world datasets, including the ETTh series (Zhou et al., 2021), Electricity, Exchange, and Traffic. These datasets cover diverse application domains and exhibit different characteristics. More details can be found in Appendix C.2.

Experiment Setting For each dataset, we first train the forecasting backbone following its original protocol. For forecasting, the historical look-back window L is set to 96, with prediction horizons H of 96 and 720. For failure risk estimation, the look-back window is also 96. The resulting backbone forecasts are kept fixed and used to train the second-stage risk estimator. For TSGuard, the batch size is selected from $\{32, 64, 128\}$, the hidden dimension from $\{64, 128\}$, and the dropout rate from $\{0.1, 0.2\}$. The second-stage model uses two layers and four attention heads, with a feed-forward dimension of 256, patch length of 16, and decomposition kernel size of 25. Hyperparameter search is restricted to the risk estimation module, while the backbone, dataset split, and forecasting protocol remain unchanged. In the main experiments, we use the 80th percentile of forecasting error as the failure threshold and evaluate performance using **AUROC** and **Recall@Top20%**.

Baselines and Forecasting Backbones. We compare TSGuard with representative time-series modeling baselines adapted for failure classification, including TimeMixer (Wang et al., 2024a), DLinear (Zeng et al., 2023), EdgeMTSC (Zhou et al., 2026), MambaSL (Jung & Kim, 2026), MedMamba (He et al., 2026), APTF (Zhang et al., 2026b), and AnchorMoE (Tao et al., 2026). We use DLinear (Zeng et al., 2023), TimeXer (Wang et al., 2024b), TQNet (Lin et al., 2025), iTransformer (Liu et al., 2024b), and PatchTST (Nie et al., 2023) as forecasting backbones to evaluate robustness across heterogeneous forecasting architectures. In addition, Appendix C.7 provides comparisons with reliability-oriented baselines, including uncertainty/error estimation and time-series anomaly detection methods.

5.2 Main Results

Table 1 and Table 2 compare the failure risk estimation performance across different forecasting backbones. Overall, existing methods show relatively limited and unstable performance, mainly because they lack an explicit mechanism to jointly characterize historical context, forecast trajectories, and their discrepancies across prediction horizons. Among the baselines, TimeMixer performs competitively in several settings, likely benefiting from its multi-scale temporal modeling, while DLinear also shows strong results due to its simple structural decomposition. In contrast, the more complex MambaSL architecture does not consistently translate into better failure estimation and exhibits noticeable variation across datasets and backbones. This instability is further summarized in Figure 2, where the AUROC and R@20 rankings of baseline methods span relatively broad ranges over the 35 dataset–backbone combinations. The issue is particularly evident on Exchange, where several baselines even obtain AUROC values below 0.5.

By contrast, TSGuard explicitly integrates historical temporal patterns, forecast behaviors, and diagnostic history–forecast mismatch features, and uses a failure decoder with horizon queries to produce fine-grained risk estimates along the forecast trajectory. As shown in Figure 2, TSGuard consistently achieves the top rank in both AUROC and R@20 across all 35 evaluation settings, indicating that its advantage is not tied to a specific dataset or forecasting backbone. The gains

Table 1: Failure risk estimation performance across different forecasting backbones. The historical look-back window L is set to 96 and the prediction horizon H is set to 96 ($H = 720$ in Table 2). Best results are highlighted in red bold, and second-best results are underlined in blue. R@20 means Recall@Top20%.

Dataset	Backbone	APTF WWW'2026		TimeMixer ICLR'2024		DLinear AAAI'2023		EdgeMTSC AAAI'2026		MambaSL ICLR'2026		MedMamba ICML'2026		AnchorMoE KDD'2026		TSGuard (Ours)	
		AUROC	R@20	AUROC	R@20	AUROC	R@20	AUROC	R@20	AUROC	R@20	AUROC	R@20	AUROC	R@20	AUROC	R@20
ETTh1	DLinear	0.558	0.180	0.663	0.385	0.657	0.340	0.577	0.270	0.528	0.213	0.425	0.143	0.606	0.085	**0.791**	**0.549**
	PatchTST	0.464	0.189	0.700	0.446	0.673	0.377	0.478	0.183	0.514	0.194	0.444	0.108	0.494	0.144	**0.760**	**0.525**
	iTrans.	0.442	0.192	0.747	0.512	0.695	0.426	0.501	0.214	0.477	0.170	0.374	0.069	0.609	0.116	**0.788**	**0.570**
	TQNet	0.500	0.145	0.760	0.519	0.701	0.423	0.525	0.218	0.427	0.125	0.484	0.115	0.593	0.155	**0.802**	**0.575**
	TimeXer	0.543	0.132	0.704	0.457	0.714	0.417	0.521	0.229	0.461	0.163	0.391	0.078	0.620	0.293	**0.818**	**0.619**
ETTh2	DLinear	0.496	0.137	0.488	0.205	0.458	0.180	0.506	0.188	0.510	0.192	0.510	0.205	0.403	0.123	**0.546**	**0.247**
	PatchTST	0.524	0.219	0.583	0.255	0.474	0.188	0.517	0.285	0.540	0.243	0.555	0.249	0.425	0.246	**0.616**	**0.318**
	iTrans.	0.469	0.149	0.458	0.143	0.508	0.226	0.462	0.158	0.545	0.240	0.473	0.189	0.468	0.095	**0.603**	**0.305**
	TQNet	0.497	0.111	0.451	0.156	0.504	0.218	0.404	0.083	0.539	0.239	0.541	0.250	0.484	0.226	**0.620**	**0.286**
	TimeXer	0.522	0.230	0.476	0.151	0.524	0.246	0.516	0.222	0.538	0.240	0.541	0.264	0.492	0.243	**0.593**	**0.323**
ETTm1	DLinear	0.509	0.160	0.660	0.275	0.507	0.195	0.512	0.211	0.612	0.272	0.523	0.223	0.562	0.144	**0.704**	**0.397**
	PatchTST	0.430	0.178	0.549	0.257	0.583	0.297	0.452	0.183	0.608	0.308	0.489	0.156	0.409	0.176	**0.735**	**0.411**
	iTrans.	0.474	0.176	0.624	0.332	0.639	0.323	0.480	0.186	0.635	0.319	0.458	0.157	0.469	0.322	**0.679**	**0.368**
	TQNet	0.499	0.240	0.666	0.388	0.624	0.291	0.493	0.200	0.602	0.296	0.517	0.194	0.544	0.293	**0.706**	**0.424**
	TimeXer	0.503	0.301	0.678	0.391	0.650	0.322	0.502	0.211	0.618	0.296	0.506	0.203	0.457	0.311	**0.713**	**0.427**
ETTm2	DLinear	0.456	0.178	0.517	0.220	0.517	0.229	0.464	0.162	0.472	0.149	0.556	0.257	0.436	0.223	**0.601**	**0.318**
	PatchTST	0.477	0.198	0.462	0.172	0.506	0.216	0.464	0.187	0.523	0.227	0.496	0.203	0.434	0.121	**0.556**	**0.264**
	iTrans.	0.476	0.179	0.519	0.224	0.542	0.250	0.457	0.174	0.487	0.203	0.536	0.240	0.510	0.189	**0.611**	**0.362**
	TQNet	0.507	0.209	0.518	0.198	0.539	0.240	0.540	0.245	0.512	0.223	0.498	0.177	0.498	0.184	**0.573**	**0.306**
	TimeXer	0.381	0.126	0.503	0.218	0.542	0.244	0.415	0.117	0.562	0.227	0.482	0.180	0.362	0.134	**0.636**	**0.344**
Exchange	DLinear	0.448	0.029	0.562	0.184	0.468	0.082	0.564	0.143	0.379	0.057	0.404	0.063	0.433	0.007	**0.604**	**0.258**
	PatchTST	0.354	0.040	0.308	0.031	0.453	0.088	0.430	0.097	0.308	0.069	0.322	0.067	0.400	0.077	**0.558**	**0.269**
	iTrans.	0.484	0.247	0.603	0.254	0.513	0.206	0.601	0.294	0.563	0.256	0.585	0.299	0.473	0.284	**0.649**	**0.331**
	TQNet	0.422	0.127	0.528	0.255	0.477	0.149	0.444	0.117	0.530	0.219	0.415	0.097	0.385	0.065	**0.644**	**0.342**
	TimeXer	0.476	0.099	0.503	0.133	0.499	0.148	0.501	0.151	0.565	0.219	0.571	0.218	0.476	0.077	**0.631**	**0.257**
Traffic	DLinear	0.812	0.570	0.888	0.597	0.857	0.588	0.878	0.605	0.881	0.616	0.844	0.538	0.799	0.490	**0.928**	**0.652**
	PatchTST	0.673	0.630	0.875	0.655	0.863	0.656	0.887	0.625	0.871	0.550	0.585	0.253	0.711	0.634	**0.924**	**0.695**
	iTrans.	0.747	0.578	0.887	0.653	0.859	0.631	0.842	0.560	0.850	0.636	0.592	0.254	0.845	0.216	**0.922**	**0.689**
	TQNet	0.777	0.564	0.871	0.635	0.886	0.627	0.847	0.555	0.860	0.626	0.552	0.242	0.712	0.596	**0.917**	**0.691**
	TimeXer	0.670	0.202	0.835	0.588	0.833	0.575	0.841	0.588	0.840	0.568	0.534	0.222	0.813	0.173	**0.895**	**0.623**
ECL	DLinear	0.749	0.357	0.772	0.559	0.775	0.456	0.773	0.512	0.768	0.533	0.626	0.340	0.743	0.385	**0.842**	**0.678**
	PatchTST	0.714	0.339	0.821	0.519	0.855	0.522	0.830	0.483	0.851	0.537	0.665	0.358	0.622	0.318	**0.888**	**0.581**
	iTrans.	0.835	0.376	0.839	0.502	0.842	0.499	0.829	0.481	0.841	0.487	0.670	0.348	0.813	0.440	**0.893**	**0.537**
	TQNet	0.634	0.346	0.844	0.467	0.832	0.449	0.841	0.474	0.822	0.441	0.656	0.317	0.612	0.410	**0.882**	**0.508**
	TimeXer	0.723	0.344	0.783	0.368	0.765	0.397	0.777	0.397	0.788	0.369	0.673	0.331	0.771	0.351	**0.834**	**0.437**

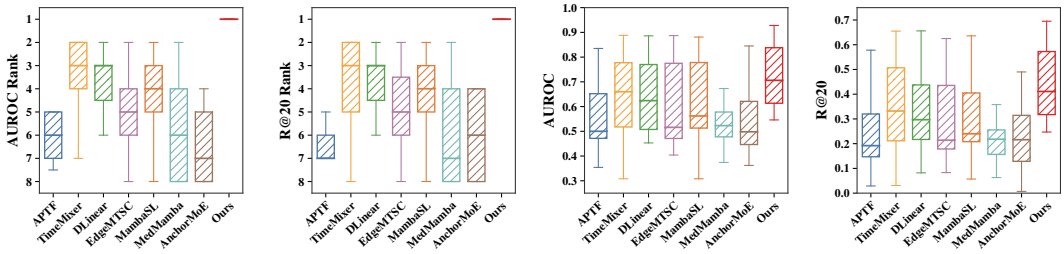

Figure 2: Overall rank and score distributions complementary to Table 1.

are also substantial in individual cases. On Exchange with PatchTST, TSGuard improves R@20 from 0.097 to 0.269 over EdgeMTSC, corresponding to a relative improvement of **177.3%**. On ETTm2 with iTransformer, it improves R@20 from 0.250 to 0.362 over DLinear, yielding a **44.8%** gain. Similarly, on ETTh1 with DLinear, TSGuard improves R@20 from 0.385 to 0.549 over TimeMixer, corresponding to a **42.6%** improvement. These results demonstrate that TSGuard achieves not only stronger failure risk estimation, but also more consistent generalization across heterogeneous forecasting settings.

5.3 Risk-Based Forecast Retention

We further evaluate post-hoc forecasting performance under risk-aware retention. Since TSGuard outputs a risk score for every lead time, retention is performed at the step level: all (case, lead-time) test forecasts are ranked by predicted risk, the top 20% highest-risk ones are discarded, and MAE and MSE are recomputed on the retained points, with the same protocol for all methods. We report the relative error reduction with respect to the full test set. **As a reference, random dropping yields an approximately 0% performance gain on average**. The results are shown in Figure 3. Among

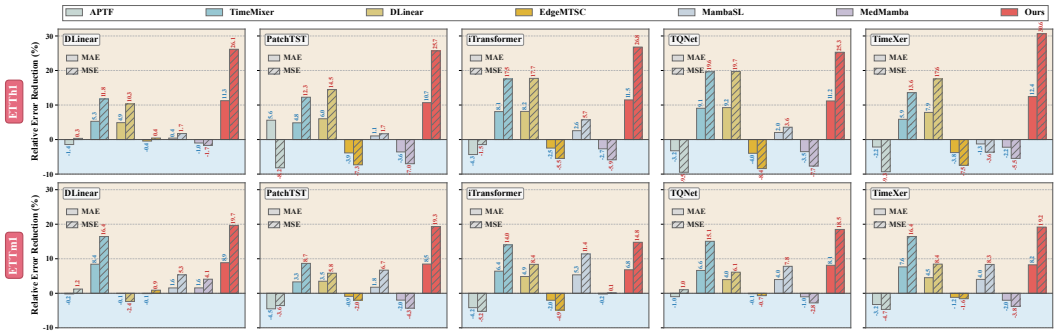

Figure 3: Comparison of post-hoc forecasting performance on ETTh1 and ETTm1 datasets. The pink regions indicate positive gains, while the blue regions indicate negative gains.

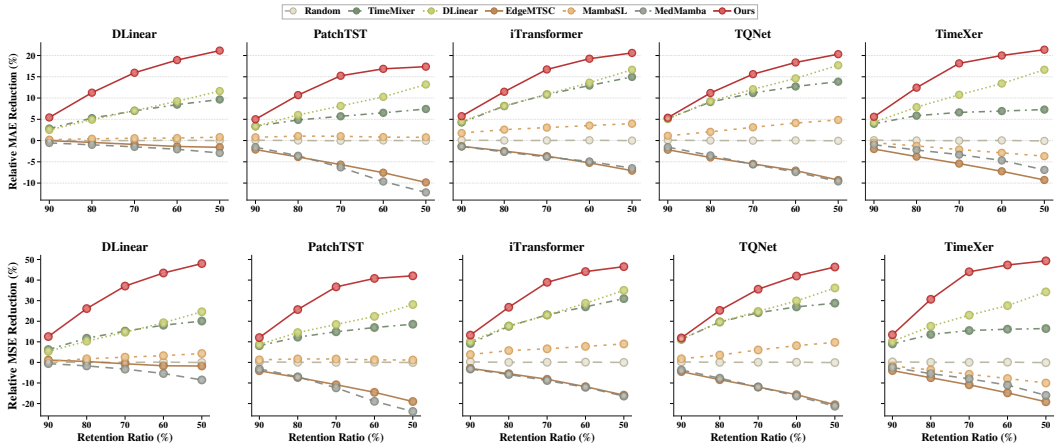

Figure 4: Post-hoc performance under retention ratios on ETTh1: (a) MAE and (b) MSE.

the baselines, TimeMixer and DLinear obtain moderate positive gains, whereas APTF, EdgeMTSC, and MedMamba frequently produce negative gains. Negative gains indicate a weak correlation between the risk scores and the true forecasting errors. In contrast, our model achieves an average MSE reduction of 22.6% across the ten ETTh1 and ETTm1 settings and a maximum reduction of 30.6%, demonstrating its effectiveness in identifying forecasting risks.

As shown in Figure 4 on ETTh1 and Figure 10 and Figure 11 on ETTm1, our method obtains larger MAE/MSE reductions as the retention ratio decreases from 90% to 50%, demonstrating effective identification of unreliable forecasts, whereas several baselines show limited or negative gains due to weaker alignment between their risk scores and actual forecasting errors.

5.4 Sensitivity to the Failure Threshold

As shown in Figure 5, the results with PatchTST provide a representative example of the robustness of our method. On ETTh1, TSGuard forms the upper performance envelope across almost the entire threshold range. Its advantage over DLinear, MambaSL, and MedMamba becomes more apparent as the failure definition becomes increasingly selective. Although TimeMixer narrows the gap under the most restrictive setting, TSGuard remains competitive, indicating that the improvement is not caused by an isolated threshold choice. A similar but more informative pattern is observed on ETTm1. MambaSL is competitive under a relatively permissive setting, but its performance gradually deteriorates as the threshold becomes stricter. In contrast, TSGuard quickly recovers and reaches its strongest regime around the default operating point while retaining a clear advantage under more selective failure definitions. This contrasting behavior suggests that TSGuard is better

able to capture the error patterns associated with rare and severe forecasting failures, rather than relying on the class distribution induced by a particular threshold.

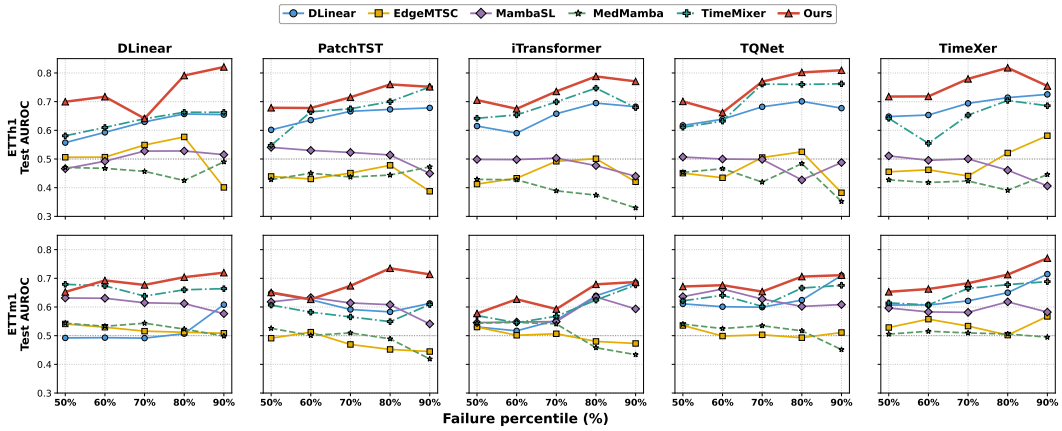

Figure 5: Failure-threshold sensitivity measured by AUROC across forecasting backbones.

5.5 Ablation Study

The *"Historical Pattern Encoder"* decomposes the observed sequence into *trend* and *residual* components and encodes them as historical memory. The *"Forecast Behavior Encoder"* represents the predicted values together with their *first- and second-order differences*. The *"Diagnostic Risk Features"* capture level, range, slope, and volatility mismatches between the historical observations and predicted trajectory. The *"Failure Decoder with Horizon Queries"* assigns an independent query to each future step, while its ablation replaces horizon-to-memory attention with a shared MLP.

As shown in Figure 6, removing the *Historical Pattern Encoder* degrades performance in all six settings, indicating that forecast reliability cannot be determined from forecast-side information alone. For DLinear, replacing the horizon decoder causes the largest degradation on the ETT datasets, while removing the *Forecast Behavior Encoder* is most detrimental on Traffic. In most settings, the *Diagnostic Risk Features* further provide complementary gains beyond learned sequence representations.

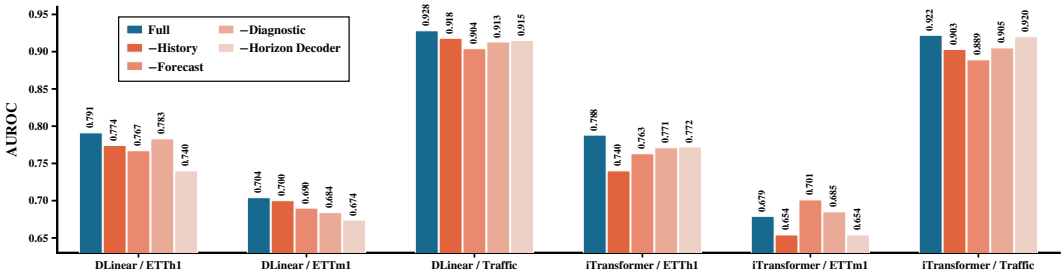

Figure 6: Ablation Study.

6 Conclusion

We study post-hoc forecast failure risk estimation for long-term time series forecasting, which aims to identify unreliable predictions before ground truth becomes available. We propose TSGuard, a backbone-agnostic post-hoc framework that integrates historical dynamics, forecast behavior, and history–forecast mismatch signals to produce fine-grained risk estimates along the forecast trajectory. Experiments on multiple datasets and forecasting backbones show that TSGuard consistently improves failure identification and enables effective risk-based forecast retention. Because TSGuard requires only the historical window and the generated forecast, it can be attached to existing forecasters without modifying or retraining them, turning point forecasts into lead-time-specific risk profiles for downstream decision-making.

REFERENCES

Lennart Bramlage and Cristóbal Curio. Principled input-output-conditioned post-hoc uncertainty estimation for regression networks, 2025. URL https://arxiv.org/abs/2506.00918.

Vitor Cerqueira, Luis Roque, and Carlos Soares. Modelradar: Aspect-based forecast evaluation. Machine Learning, 114:229, 2025. doi: 10.1007/s10994-025-06877-z.

Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pp. 4027–4035, 2021. doi: 10.1609/aaai.v35i5.16523.

Isaac Gibbs and Emmanuel Candès. Adaptive conformal inference under distribution shift. In Advances in Neural Information Processing Systems, volume 34, pp. 1660–1672, 2021.

Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. Annual Review of Statistics and Its Application, 1:125–151, 2014. doi: 10.1146/annurev-statistics-062713-085831.

Qi Guo, Baolin Zhao, Mingchen Song, and Guoqiang Zhong. A survey of deep learning for time series forecasting: Taxonomy, analysis and future directions. IEEE Transactions on Knowledge and Data Engineering, 2026.

ZhengXiao He, Huayu Li, Xiwen Chen, Janet M Roveda, Jinghao Wen, Siyuan Tian, and Ao Li. Medmamba: Recasting mamba for medical time series classification. arXiv preprint arXiv:2605.05214, 2026.

Dongjian Hu, Junxi Wu, Shu-Tao Xia, and Changliang Zou. Distribution-informed online conformal prediction. In The Fourteenth International Conference on Learning Representations, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5f8241131e6fd428aa49914da76b8ad0-Abstract-Conference.html.

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. Timekan: Kan-based frequency decomposition learning architecture for long-term time series forecasting. arXiv preprint arXiv:2502.06910, 2025.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. arXiv preprint arXiv:2310.01728, 2023.

Yoo-Min Jung and Leekyung Kim. MambaSL: Exploring single-layer mamba for time series classification. In The Fourteenth International Conference on Learning Representations, 2026. URL https://openreview.net/forum?id=YDl4vqQqGP.

Dilfira Kudrat, Zongxia Xie, Yanru Sun, Tianyu Jia, and Qinghua Hu. Patch-wise structural loss for time series forecasting. In Proceedings of the 42nd International Conference on Machine Learning, volume 267, pp. 31841–31859. PMLR, 2025.

Jinglin Li, Jun Tan, Qi Fang, and Ning Gui. Parametric prior mapping framework for non-stationary probabilistic time series forecasting. In International Conference on Machine Learning, 2026. URL https://arxiv.org/abs/2605.23402.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. Advances in Neural Information Processing Systems, 37:106315–106345, 2024.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. arXiv preprint arXiv:2505.12917, 2025.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and C. L. Philip Chen. Sparsetsf: Lightweight and robust time series forecasting via sparse modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 48(1):170–183, 2026. doi: 10.1109/TPAMI.2025.3602445.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Prabhakar Kamarthi, Aditya Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, et al. Time-mmd: Multi-domain multimodal dataset for time series analysis. Advances in Neural Information Processing Systems, 37:77888–77933, 2024a.

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, 2024b.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting. In International Conference on Learning Representations, 2025a.

Zhiding Liu, Mingyue Cheng, Guanhao Zhao, Jiqian Yang, Qi Liu, and Enhong Chen. Improving time series forecasting via instance-aware post-hoc revision. In Advances in Neural Information Processing Systems, volume 38, 2025b. doi: 10.52202/085713-1196. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/331c41353b053683e17f7c88a797701d-Abstract-Conference.html.

Aitian Ma, Dongsheng Luo, and Mo Sha. Mixlinear: Extreme low-resource multivariate time series forecasting with 0.1k parameters. In International Conference on Learning Representations, 2026a.

Jiaming Ma, Binwu Wang, Qihe Huang, Guanjun Wang, Pengkun Wang, Zhengyang Zhou, and Yang Wang. Mofo: Empowering long-term time series forecasting with periodic pattern modeling. In Advances in Neural Information Processing Systems, volume 38, 2025.

Jiaming Ma, Qihe Huang, Haofeng Ma, Guanjun Wang, Sheng Huang, Zhengyang Zhou, Pengkun Wang, Xu Wang, Binwu Wang, and Yang Wang. Phat: Modeling period heterogeneity for multivariate time series forecasting. In International Conference on Learning Representations, volume 2026, pp. 135659–135685, 2026b.

John A Miller, Mohammed Aldosari, Farah Saeed, Nasid Habib Barna, Subas Rana, I Budak Arpinar, and Ninghao Liu. A survey of deep learning and foundation models for time series forecasting. arXiv preprint arXiv:2401.13912, 2024.

Md Mahmuddun Nabi Murad and Yasin Yilmaz. Cluster-aware causal mixer for online anomaly detection in multivariate time series, 2025. URL https://arxiv.org/abs/2506.00188.

Md Mahmuddun Nabi Murad, Mehmet Aktukmak, and Yasin Yilmaz. Wpmixer: Efficient multi-resolution mixing for long-term time series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, pp. 19581–19588, 2025.

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.

Yiming Niu, Jinliang Deng, and Yongxin Tong. Phaseformer: From patches to phases for efficient and effective time series forecasting. In The Fourteenth International Conference on Learning Representations, 2026. URL https://openreview.net/forum?id=Lk9SqMQzhX.

Min-Yeong Park, Won-Jeong Lee, Seong Tae Kim, and Gyeong-Moon Park. When will it fail?: Anomaly to prompt for forecasting future anomalies in time series. In Proceedings of the 42nd International Conference on Machine Learning, volume 267 of Proceedings of Machine Learning Research, pp. 48086–48103. PMLR, 2025. URL https://proceedings.mlr.press/v267/park25e.html.

Kamile Stankeviciute, Ahmed M. Alaa, and Mihaela van der Schaar. Conformal time-series forecasting. In Advances in Neural Information Processing Systems, volume 34, pp. 6216–6228, 2021.

Artyom Stitsyuk and Jaesik Choi. xpatch: Dual-stream time series forecasting with exponential seasonal-trend decomposition. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, pp. 20601–20609, 2025.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2828–2837, 2019. doi: 10.1145/3292500.3330672.

Xin Sun, Heng Zhou, Yuhao Wu, and Chao Li. State-derivative-aware neural controlled differential equations for multivariate time series anomaly detection and diagnosis. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 40, pp. 25727–25735, 2026. doi: 10.1609/aaai.v40i30.39770. URL https://ojs.aaai.org/index.php/AAAI/article/view/39770.

Peiwang Tang and Weitai Zhang. Unlocking the power of patch: Patch-based mlp for long-term time series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, pp. 12640–12648, 2025.

Xie Tao, Tan Zexi, Xiao Haoyi, Li Mengke, Zhang Yiqun, Lu Yang, Yang Cuie, and Cheung Yiu-Ming. Anchormoe: Interpretable time series classification via anchor-routed moe. In Proceedings of the 32nd ACM SIGKDD conference on knowledge discovery and data mining (KDD), 2026.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. Advances in neural information processing systems, 34:24804–24816, 2021.

Hao Wang, Licheng Pan, Zhichao Chen, Xu Chen, Qingyang Dai, Lei Wang, Haoxuan Li, and Zhouchen Lin. Time-o1: Time-series forecasting needs transformed label alignment. In Advances in Neural Information Processing Systems, volume 38, pp. 9591–9621, 2025a.

Huiqiang Wang, Jieming Shi, and Li Qing. FACT: Fine-grained across-variable convolution for multivariate time series forecasting. In The Fourteenth International Conference on Learning Representations, 2026. URL https://openreview.net/forum?id=j3gNYqrHtl.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In International conference on learning representations, volume 2024, pp. 38626–38652, 2024a.

Xiaoqian Wang and Rob J. Hyndman. Online conformal inference for multi-step time series forecasting, 2024. URL https://arxiv.org/abs/2410.13115.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. Advances in Neural Information Processing Systems, 37:469–498, 2024b.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is mamba effective for time series forecasting? Neurocomputing, 619:129178, 2025b.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in neural information processing systems, 34:22419–22430, 2021.

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

Junxi Wu, Dongjian Hu, Yajie Bao, Shu-Tao Xia, and Changliang Zou. Error-quantified conformal inference for time series. In The Thirteenth International Conference on Learning Representations, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/cab5ae2704d3e01f06a92512a5376b87-Abstract-Conference.html.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly Transformer: Time series anomaly detection with association discrepancy. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=LzQQ89U1qm_.

Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. DCdetector: Dual attention contrastive representation learning for time series anomaly detection. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 3033–3045, 2023. doi: 10.1145/3580305.3599295.

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.

Fan Zhang, Sinchee Chin, Jing-Hao Xue, and Wenming Yang. ICDiffAD: Implicit conditioning diffusion model for time series anomaly detection. In The Fourteenth International Conference on Learning Representations, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/59a9cc95f046e9125d8816ef971873e7-Abstract-Conference.html.

Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Y Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, et al. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. IEEE transactions on pattern analysis and machine intelligence, 46(10):6775–6794, 2024.

Xu Zhang, Peng Wang, Yichen Li, and Wei Wang. Amortized predictability-aware training framework for time series forecasting and classification. In Proceedings of the ACM Web Conference 2026, pp. 5624–5635, 2026b.

Zhiwei Zhang, Xinyi Du, Xuanchi Guo, Weihao Wang, and Wenjuan Han. Emaformer: Enhancing transformer through embedding armor for time series forecasting. arXiv preprint arXiv:2511.08396, 2025.

Zhiyuan Zhao, Juntong Ni, Shangqing Xu, Haoxin Liu, Wei Jin, and B Aditya Prakash. Timerecipe: A time-series forecasting recipe via benchmarking module level effectiveness. In International Conference on Learning Representations, volume 2026, pp. 150745–150792, 2026.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In International conference on machine learning, pp. 27268–27286. PMLR, 2022.

Xueyi Zhou, Zhenyu Li, and Dong-Kyu Chae. Edgemtsc: A lightweight large-kernel convnet for multivariate time series classification. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 40, pp. 16531–16539, 2026.

A PSEUDOCODE OF TSGUARD

Algorithm 1 Forward Pass and Training Pipeline of TSGuard

Input: historical window $\boldsymbol{X} \in \mathbb{R}^{L \times C}$, trained backbone forecast $\widehat{\boldsymbol{Y}} \in \mathbb{R}^{H \times D}$, optional timestamps $\mathcal{T}$, and failure percentile τ.
Output: fine-grained failure risk scores $\widehat{\boldsymbol{p}} = [\widehat{p}_1, \ldots, \widehat{p}_H]$.
1. Validation-only Failure Label Construction
Generate out-of-fold backbone predictions on train windows and predictions $\widehat{\boldsymbol{Y}} = f_\phi(\boldsymbol{X})$ on validation and test windows.
Compute validation errors $e_{m,h} = D^{-1} \sum_d |\widehat{Y}_{m,h,d} - Y_{m,h,d}|$ for $m \in \mathcal{V}$.
Define thresholds $\delta_h \leftarrow \text{Percentile}_\tau(\{e_{m,h} : m \in \mathcal{V}\})$ and label each forecast by $z_h \leftarrow \mathbb{I}[e_h > \delta_h]$.
2. Historical Pattern Encoder
Decompose the history: $\boldsymbol{X}^{\text{trend}} \leftarrow \text{MA}_K(\boldsymbol{X})$, $\boldsymbol{X}^{\text{res}} \leftarrow \boldsymbol{X} - \boldsymbol{X}^{\text{trend}}$.
Patch, embed, and encode trend/residual streams to obtain historical memory tokens $\boldsymbol{H}_x$.
3. Forecast Behavior Encoder
Build forecast dynamics $\boldsymbol{F} = [\widehat{\boldsymbol{Y}}, \Delta\widehat{\boldsymbol{Y}}, \Delta^2\widehat{\boldsymbol{Y}}]$.
Project and encode the forecast trajectory to obtain forecast-behavior tokens $\boldsymbol{H}_y$.
4. Diagnostic Risk Feature Encoder
Construct diagnostic features $\boldsymbol{r}_h$ for each lead time, including slope mismatch, volatility mismatch, range excess, recent jumps, prefix volatility, and lead-time position.
Project diagnostics with an MLP to obtain risk tokens $\boldsymbol{H}_r[h] \leftarrow \text{MLP}(\boldsymbol{r}_h)$.
5. Failure Decoder with Horizon Queries
Fuse memory tokens $\boldsymbol{M} \leftarrow [\boldsymbol{H}_x; \boldsymbol{H}_y; \boldsymbol{H}_r]$.
Build horizon queries with horizon embeddings and optional hour/day-of-week embeddings from $\mathcal{T}$.
Apply cross-attention from horizon queries to $\boldsymbol{M}$, followed by residual feed-forward refinement.
Return $\widehat{p}_h \leftarrow \sigma(\boldsymbol{w}^\top \boldsymbol{Z}_h + b)$ for $h = 1, \ldots, H$.
6. Optimization
Train TSGuard with weighted binary cross-entropy using labels z_h.
At inference time, use only $\boldsymbol{X}$, $\widehat{\boldsymbol{Y}}$, and optional timestamp features; never use $\boldsymbol{Y}$.
return $\widehat{\boldsymbol{p}}$.

B RELATED WORK

Uncertainty Quantification. Uncertainty quantification characterizes predictive reliability beyond point estimates. Probabilistic forecasting represents future outcomes through predictive distributions, emphasizing calibration and sharpness (Gneiting & Katzfuss, 2014). PPM combines adaptive parametric priors with learnable mappings for non-stationary probabilistic forecasting (Li et al., 2026), while IO-CUE estimates post-hoc regression uncertainty conditioned on inputs and frozen model outputs (Bramlage & Curio, 2025). Conformal methods offer a complementary approach through prediction sets and coverage control. Conformal time-series forecasting extends uncertainty estimation to multiple horizons (Stankeviciute et al., 2021), and adaptive conformal inference targets long-run coverage under distribution shifts (Gibbs & Candès, 2021). Recent developments exploit error-magnitude feedback in ECI (Wu et al., 2025), estimated nonconformity-score distributions in COP (Hu et al., 2026), and autocorrelated multi-step forecast errors in AcMCP (Wang & Hyndman, 2024). These approaches provide useful reliability signals, but distributional estimation and coverage control have different objectives from identifying task-defined forecasting failures. Complementing them, TSGuard directly learns lead-time-specific risk scores for large-error events of a frozen point forecaster.

Anomaly Detection. Time series anomaly detection identifies deviations from normal temporal behavior. OmniAnomaly learns stochastic recurrent representations and uses reconstruction probabilities for anomaly scoring (Su et al., 2019), while GDN models inter-variable dependencies and detects deviations from graph-based predictions (Deng & Hooi, 2021). Anomaly Transformer exploits association discrepancies (Xu et al., 2022), and DCdetector learns discriminative representations through dual-attention contrastive learning (Yang et al., 2023). Recent methods further enrich the modeling of temporal abnormalities. ICDiffAD combines adaptive noise scheduling with implicit conditioning for diffusion-based reconstruction (Zhang et al., 2026a), whereas SDA2D incorporates

state derivatives from neural controlled differential equations to capture anomalous system dynamics (Sun et al., 2026). For online detection, CCM-TAD combines cluster-aware causal mixing with sequential evidence accumulation (Murad & Yilmaz, 2025). Beyond detecting observed anomalies, A2P integrates anomaly-aware forecasting and synthetic anomaly prompting to anticipate future anomalous time points (Park et al., 2025). However, detecting or forecasting data abnormalities is not equivalent to estimating the error risk of a particular forecasting model. In contrast, TSGuard targets horizon-wise large-error events of a frozen forecaster, using historical observations, the generated trajectory, and their diagnostic mismatches before future ground truth becomes available.

C Additional Experiments

C.1 Baseline and Dataset Details

To comprehensively evaluate TSGuard, we compare it with representative time series classification methods covering linear, convolutional, state-space, and mixture-of-experts architectures. In addition, several widely used forecasting models are adopted as backbones to assess the effectiveness of TSGuard across different forecasting architectures.

Time series classification baselines.

- **TimeMixer** (Wang et al., 2024a) is an MLP-based architecture that models temporal patterns at multiple scales. It progressively mixes fine- and coarse-grained representations to capture complementary temporal dynamics.
- **DLinear** (Zeng et al., 2023) is a lightweight linear model that decomposes a time series into trend and seasonal components and models them separately. Despite its simple architecture, it provides a strong and efficient baseline for time series modeling.
- **EdgeMTSC** (Zhou et al., 2026) is a lightweight large-kernel convolutional network designed for multivariate time series classification. It combines multi-scale temporal convolutions with inter-channel message passing to capture both temporal patterns and cross-variable dependencies.
- **MambaSL** (Jung & Kim, 2026) adapts the selective state-space model Mamba to time series classification with a lightweight single-layer architecture. It introduces task-specific modifications to efficiently capture discriminative temporal representations.
- **MedMamba** (He et al., 2026) employs a multi-scale bidirectional state-space architecture for medical time series classification. It combines channel mixing, multi-scale temporal tokenization, and bidirectional Mamba blocks to model cross-channel interactions and long-range temporal dependencies.
- **APTF** (Zhang et al., 2026b) is a predictability-aware training framework for time series forecasting and classification. It dynamically identifies low-predictability samples and adjusts their training contribution to reduce the adverse effect of noisy or difficult samples.
- **AnchorMoE** (Tao et al., 2026) is an interpretable mixture-of-experts framework for multivariate time series classification. It routes local temporal patches to specialized experts and aggregates their contributions to obtain both accurate and interpretable predictions.

Adaptation to failure risk estimation. All classification baselines take the same inputs as TSGuard: the historical window $\boldsymbol{X}$ and the frozen forecast $\widehat{\boldsymbol{Y}}$ are concatenated along the time axis into a sequence of length $L + H$, with variables zero-padded to a common dimension when $C \neq D$ (DLinear operates on the variable-averaged sequence). The original classification head of each model is replaced by a linear layer that outputs H logits, one per lead time, followed by a sigmoid. All baselines are trained on the same failure labels with the lead-time-weighted binary cross-entropy $\mathcal{L}_{\mathrm{WBCE}}$ and the same optimization settings as TSGuard, with early stopping on the validation loss.

Time series forecasting backbones.

- **DLinear** (Zeng et al., 2023) decomposes historical observations into trend and seasonal components and forecasts them using lightweight linear layers, providing a simple yet competitive forecasting backbone.

- **TimeXer** (Wang et al., 2024b) is a Transformer-based forecasting model designed to incorporate exogenous variables. It uses patch-level representations for endogenous series and variate-level representations for exogenous variables, with global tokens bridging their interactions.
- **TQNet** (Lin et al., 2025) introduces learnable temporal queries to capture global inter-variable relationships while extracting sample-specific information from the observed series. Its lightweight attention architecture provides an effective balance between forecasting accuracy and efficiency.
- **iTransformer** (Liu et al., 2024b) inverts the conventional Transformer formulation by treating individual variables as tokens. Self-attention is therefore performed across variables to explicitly model multivariate correlations.
- **PatchTST** (Nie et al., 2023) segments each time series into temporal patches and treats these patches as Transformer tokens. Its channel-independent design efficiently captures long-range temporal dependencies while reducing sequence length.

C.2 Dataset Description

We conduct experiments on seven widely used real-world multivariate time series datasets spanning energy, finance, and transportation domains. Their detailed descriptions are as follows:

- **ETTh1**: ETTh1 is an hourly subset of the Electricity Transformer Temperature (ETT) dataset. It contains seven variables, including six power load-related features and oil temperature, with 17,420 observations collected over approximately two years.
- **ETTh2**: ETTh2 is collected from a different electricity transformer and follows the same hourly sampling frequency and seven-variable configuration as ETTh1. It exhibits different temporal patterns and operating characteristics, providing a complementary benchmark for evaluating forecasting robustness.
- **ETTm1**: ETTm1 is the 15-minute-resolution counterpart of ETTh1 and contains the same seven transformer-related variables. Its finer temporal resolution results in 69,680 observations and captures richer short-term variations and local temporal dynamics.
- **ETTm2**: ETTm2 is sampled every 15 minutes from the second electricity transformer. Similar to ETTm1, it contains seven variables and 69,680 observations, providing fine-grained measurements of transformer loads and oil temperature.
- **Electricity**: Electricity records the hourly electricity consumption of 321 clients from 2012 to 2014, resulting in 26,304 observations. The dataset is highly multivariate and exhibits pronounced daily and weekly consumption patterns.
- **Exchange**: Exchange contains the daily exchange rates of eight currencies from 1990 to 2016, with 7,588 observations. Compared with energy datasets, it exhibits weaker periodicity and stronger non-stationary temporal variations.
- **Traffic**: Traffic contains hourly road occupancy measurements collected from 862 sensors deployed on freeways in the San Francisco Bay Area. It consists of 17,544 observations and is characterized by high dimensionality and strong daily and weekly traffic patterns.

C.3 Out-of-Fold Training Forecasts

A backbone fitted on the full training split produces optimistic in-sample errors on that split, so thresholding them with the validation-derived δ_h would yield training labels that do not reflect out-of-sample failure behavior. We therefore generate training-split forecasts out-of-fold in temporal order. The first 50% of the training windows are used only for backbone fitting. The remaining windows are divided into three consecutive blocks, and each block is predicted by a backbone trained on all windows preceding it, with 10% of these windows held out for early stopping. The failure risk estimator is trained on these out-of-fold forecasts and their labels $z_h = \mathbb{I}[e_h > \delta_h]$, whereas validation and test forecasts are produced by the backbone trained on the full training split.

Table 2: Long-horizon failure risk estimation across forecasting backbones with $L = 96$ and $H = 720$. Failure thresholds use the 80th percentile of validation forecasting errors. AUROC and R@20 (Recall@Top20%) are higher-is-better test metrics. Red bold and blue underlining identify the best and second-best available entries in each row, respectively.

Dataset	Backbone	APTF WWW'2026		TimeMixer ICLR'2024		DLinear AAAI'2023		EdgeMTSC AAAI'2026		MambaSL ICLR'2026		MedMamba ICML'2026		AnchorMoE KDD'2026		Ours	
		AUROC	R@20	AUROC	R@20	AUROC	R@20	AUROC	R@20	AUROC	R@20	AUROC	R@20	AUROC	R@20	AUROC	R@20
ETTh1	DLinear	0.518	0.182	0.520	0.206	0.589	0.278	0.612	0.262	0.554	0.265	0.471	0.185	0.529	0.234	**0.796**	**0.615**
	PatchTST	0.496	0.191	0.494	0.192	0.485	0.213	0.470	0.173	0.469	0.179	0.495	0.198	0.508	0.203	**0.571**	**0.263**
	iTrans.	0.657	0.270	0.775	0.573	0.661	0.375	0.575	0.320	0.503	0.219	0.653	0.345	0.469	0.157	**0.843**	**0.711**
	TQNet	0.534	0.113	0.799	0.601	0.731	0.491	0.637	0.351	0.567	0.239	0.644	0.332	0.500	0.193	**0.815**	**0.655**
	TimeXer	0.670	0.371	0.716	0.405	0.733	0.476	0.573	0.341	0.542	0.233	0.633	0.332	0.470	0.159	**0.779**	**0.582**
ETTh2	DLinear	0.503	0.290	0.469	0.190	0.484	0.191	0.487	0.235	0.419	0.150	0.563	0.284	0.479	0.180	**0.616**	**0.376**
	PatchTST	0.473	0.227	0.520	0.216	0.471	0.183	0.427	0.152	0.440	0.155	0.486	0.219	0.530	0.199	**0.558**	**0.258**
	iTrans.	0.472	0.152	0.401	0.140	0.456	0.178	0.448	0.161	0.470	0.173	0.444	0.166	0.562	0.271	**0.605**	**0.298**
	TQNet	0.529	0.198	0.394	0.128	0.476	0.194	0.442	0.174	0.418	0.134	0.537	0.214	0.530	0.229	**0.578**	**0.242**
	TimeXer	0.530	0.291	0.436	0.164	0.470	0.185	0.504	0.275	0.487	0.187	0.490	0.190	0.509	0.210	**0.613**	**0.314**
ETTm1	DLinear	0.608	0.339	0.498	0.194	0.549	0.220	0.559	0.273	0.528	0.231	0.499	0.181	0.486	0.188	**0.787**	**0.582**
	PatchTST	0.599	0.326	0.509	0.208	0.635	0.284	0.696	0.429	0.556	0.188	0.470	0.158	0.413	0.128	**0.776**	**0.556**
	iTrans.	0.684	0.455	0.704	0.447	0.672	0.390	0.515	0.172	0.519	0.179	0.560	0.226	0.440	0.150	**0.816**	**0.624**
	TQNet	0.673	0.434	0.724	0.437	0.706	0.418	0.741	0.456	0.507	0.182	0.530	0.187	0.513	0.196	**0.817**	**0.619**
	TimeXer	0.706	0.414	0.589	0.319	0.712	0.426	0.516	0.214	0.543	0.207	0.540	0.214	0.482	0.174	**0.779**	**0.540**
ETTm2	DLinear	0.458	0.240	0.501	0.206	0.484	0.178	0.472	0.201	0.426	0.128	0.483	0.215	0.502	0.200	**0.571**	**0.271**
	PatchTST	0.534	0.232	0.502	0.200	0.495	0.183	0.544	0.231	0.498	0.206	0.497	0.220	0.495	0.193	**0.599**	**0.276**
	iTrans.	0.510	0.236	0.543	0.248	0.554	0.238	0.549	0.249	0.537	0.234	0.526	0.210	0.510	0.208	**0.578**	**0.254**
	TQNet	0.436	0.213	0.499	0.200	0.513	0.206	0.634	0.322	0.645	0.331	0.540	0.305	0.503	0.202	**0.675**	**0.365**
	TimeXer	0.550	0.253	0.479	0.167	0.544	0.247	0.589	0.240	0.596	0.276	0.568	0.281	0.498	0.195	**0.626**	**0.299**
Exchange	DLinear	0.574	0.267	0.454	0.218	0.557	0.152	0.843	0.469	0.788	0.355	0.642	0.212	0.307	0.072	**0.881**	**0.521**
	PatchTST	0.453	0.008	0.750	0.745	0.748	0.647	0.818	0.812	0.839	0.825	0.820	0.813	0.561	0.422	**0.871**	**0.834**
	iTrans.	0.552	0.266	0.651	0.263	0.473	0.210	0.559	0.236	0.576	0.253	0.578	0.231	0.491	0.212	**0.674**	**0.354**
	TQNet	0.480	0.179	0.553	0.193	0.446	0.097	0.452	0.157	0.712	0.362	0.638	0.243	0.668	0.381	**0.827**	**0.458**
	TimeXer	0.468	0.182	0.764	0.334	0.398	0.060	0.575	0.184	0.772	0.394	0.531	0.133	0.466	0.096	**0.820**	**0.464**
Traffic	DLinear	0.825	0.536	0.917	0.594	0.859	0.539	0.907	0.576	0.906	0.587	0.521	0.228	0.870	0.559	**0.928**	**0.647**
	PatchTST	0.837	0.596	0.501	0.200	0.871	0.575	0.859	0.553	0.804	0.539	0.493	0.195	0.769	0.504	**0.888**	**0.646**
	iTrans.	0.829	0.577	0.502	0.200	0.891	0.633	0.898	0.593	0.876	0.582	0.550	0.238	0.833	0.585	**0.935**	**0.698**
	TQNet	0.845	0.623	0.500	0.201	0.884	0.605	0.918	0.647	0.830	0.470	0.509	0.213	0.824	0.605	**0.940**	**0.717**
	TimeXer	0.833	0.595	0.927	0.669	0.913	0.654	0.853	0.530	0.810	0.521	0.536	0.224	0.916	0.651	**0.930**	**0.690**
ECL	DLinear	0.524	0.214	0.534	0.233	0.756	0.353	0.808	0.433	0.781	0.394	0.533	0.213	0.489	0.190	**0.828**	**0.439**
	PatchTST	0.743	0.519	0.502	0.183	0.744	0.398	0.840	0.515	0.833	0.511	0.475	0.173	0.492	0.202	**0.900**	**0.605**
	iTrans.	0.829	0.458	0.815	0.501	0.851	0.514	0.830	0.520	0.780	0.447	0.622	0.301	0.495	0.212	**0.866**	**0.524**
	TQNet	0.878	0.504	0.489	0.187	0.796	0.412	0.864	0.492	0.728	0.395	0.496	0.196	0.486	0.185	**0.879**	**0.511**
	TimeXer	0.840	0.504	0.835	0.502	0.832	0.506	0.845	0.501	0.813	0.457	0.607	0.296	0.498	0.204	**0.861**	**0.522**

C.4 Failure Risk Estimation with a 720-Step Forecasting Horizon

Table 2 compares failure risk estimation performance with a prediction length of 720. The results reveal that TSGuard maintains favorable performance across long-horizon settings. Although several existing methods provide competitive estimates under certain dataset–backbone combinations, their performance varies considerably across evaluation settings. This instability becomes more pronounced over such a long forecasting horizon, where prediction uncertainty accumulates and failure patterns may evolve substantially across future steps. Modeling historical observations or forecast outputs in isolation therefore provides only limited information for identifying when and where predictions are likely to fail. The broad ranking distributions in Figure 7 further indicate that this limitation is systematic across different datasets and forecasting architectures.

TSGuard is particularly effective in this long-horizon setting because it treats failure estimation as a trajectory-aware diagnostic problem. By jointly modeling historical dynamics, forecast trajectories, and their discrepancies, TSGuard can capture how prediction reliability evolves throughout the 720-step forecasting horizon. Its horizon-query-based failure decoder further enables localized risk estimation at different future steps rather than relying on a single global representation. Consequently, TSGuard maintains favorable AUROC and R@20 rankings across diverse datasets and forecasting backbones. These results demonstrate that TSGuard remains robust even under long-term forecasting, where longer horizons induce substantially different and more heterogeneous failure patterns, making robust risk estimation particularly important.

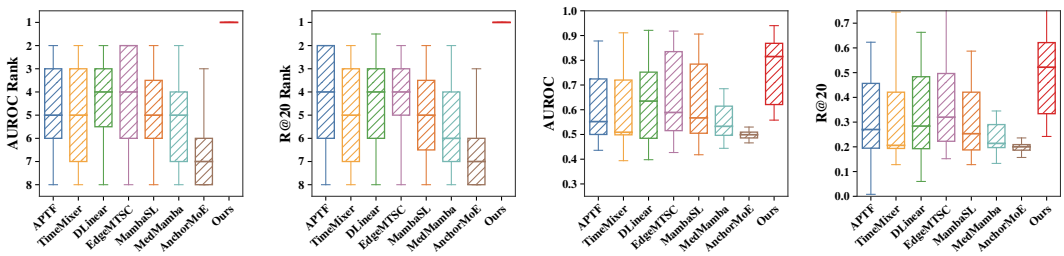

Figure 7: Overall rank and score distributions complementary to Table 2.

C.5 Comparison with Lightweight Risk Estimators

We further compare our method with lightweight alternatives to examine whether failure risk can be reliably estimated from simple temporal priors, scalar error surrogates, or generic sequence models. **Calendar-LR** predicts failure risk using only calendar and horizon-position features, providing a simple temporal-prior baseline. **Error-LGBM** learns a nonlinear mapping from diagnostic information to normalized forecasting errors and uses the predicted errors for risk ranking. We additionally include **TimeMixer** and **MambaSL** as representative neural sequence models with multi-scale temporal modeling and state-space modeling capabilities, respectively. All methods are evaluated on the same fixed forecasting outputs and failure labels, allowing the comparison to isolate their ability to identify unreliable predictions rather than forecasting accuracy itself.

As shown in Figure 8, we divide the 96-step prediction horizon into four consecutive regions, i.e., horizons 1–24, 25–48, 49–72, and 73–96, and report AUROC across different datasets and forecasting backbones. This horizon-wise comparison reveals a clearer distinction between the methods than aggregate evaluation alone. Lightweight baselines can be competitive in individual datasets or local horizon regions, but their performance varies substantially across forecasting stages and backbones. In particular, a method that performs well in the early prediction region does not necessarily maintain the same discrimination ability toward more distant horizons, indicating that failure patterns evolve along the forecast trajectory rather than remaining stationary.

In contrast, TSGuard exhibits consistently strong performance across the four horizon regions under both DLinear and PatchTST backbones. The advantage is especially informative in the later prediction regions, where forecasting uncertainty and error accumulation become more pronounced. While Calendar-LR relies primarily on recurring temporal regularities and Error-LGBM reduces risk estimation to scalar error prediction, our model jointly characterizes historical dynamics, forecast behavior, and horizon-specific diagnostic evidence. TimeMixer and MambaSL provide stronger temporal representation capacity, but do not explicitly model the relationship between the observed history, the generated forecast, and horizon-dependent failure behavior. These results suggest that effective failure-risk estimation requires not only temporal representation or error approximation, but also explicit horizon-aware reasoning over heterogeneous evidence throughout the forecast trajectory.

C.6 Evaluation under Precision–Recall Trade-offs

Failure events correspond to relatively large forecasting errors and therefore constitute only a subset of all prediction horizons, making failure risk estimation naturally class-imbalanced. To complement AUROC and Recall@Top20%, we further evaluate all methods using the Area Under the Precision–Recall Curve (AUPRC), which places greater emphasis on the ability to identify true failures while avoiding excessive false alarms. This experiment is intended to examine whether the predicted risk scores remain discriminative under an evaluation criterion that is more sensitive to rare positive events, rather than benefiting primarily from the large number of non-failure cases.

As shown in Figure 9, across different datasets and forecasting backbones, TSGuard consistently achieves the strongest AUPRC performance. The improvement is particularly evident in settings where failure events are more difficult to distinguish and several generic sequence models exhibit limited precision–recall discrimination. On highly multivariate datasets, where some competing methods already provide relatively strong risk estimates, TSGuard still maintains a consistent ad-

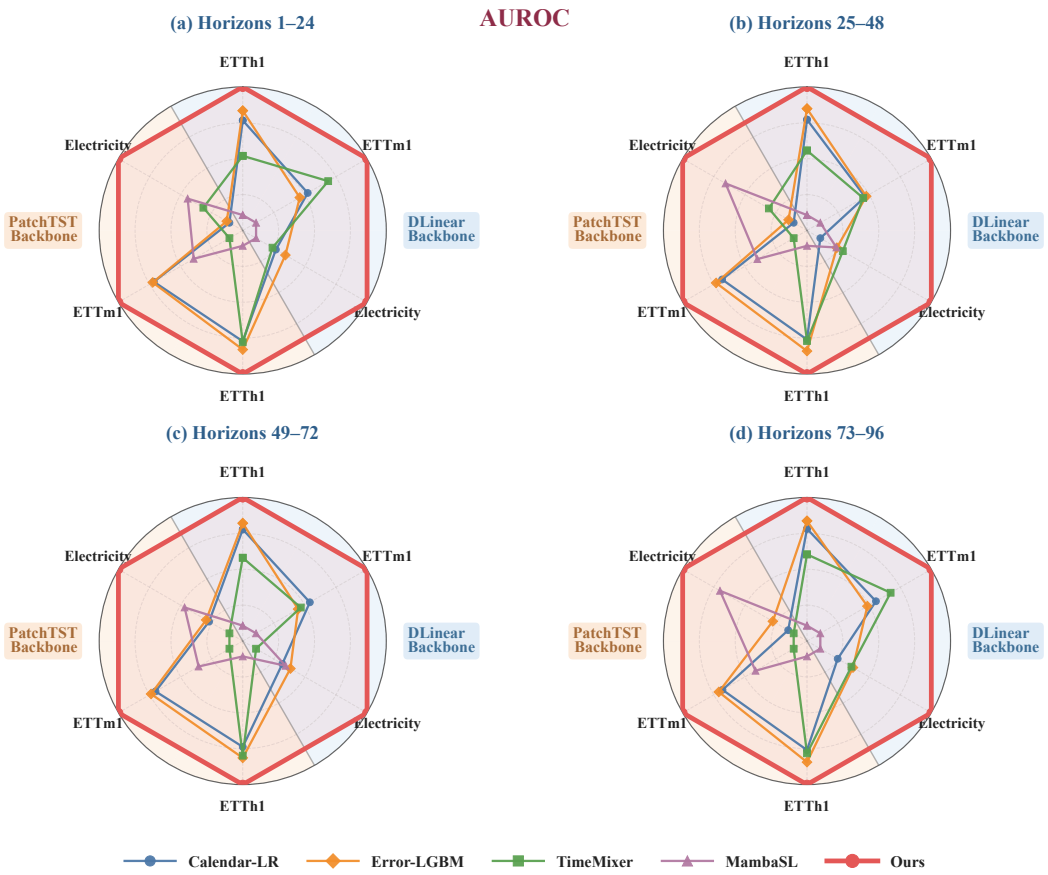

Figure 8: Horizon-wise AUROC comparison across datasets and forecasting backbones. Blue and orange sectors correspond to DLinear and PatchTST, respectively. Each spoke is independently normalized within its dataset–backbone setting, with the outer boundary defined by the performance of TSGuard. Higher is better.

vantage across different forecasting architectures. These results complement the AUROC and Recall@Top20% evaluations and demonstrate that the effectiveness of TSGuard is not tied to a particular ranking metric. Instead, its joint modeling of historical dynamics, forecast behavior, and history–forecast mismatch provides more reliable identification of genuinely failure-prone predictions under class-imbalanced conditions.

C.7 Comparison with Reliability-Oriented Baselines

Beyond conventional generic sequence-model baselines, we further compare with methods from two closely related reliability-oriented directions: uncertainty and error estimation and time-series anomaly detection. The former quantifies predictive uncertainty, expected forecasting error, or calibrated error bounds, while the latter identifies abnormal temporal patterns in observed sequences. Neither objective is directly equivalent to our post-hoc forecast failure risk estimation, which aims to identify whether the output of a fixed forecasting backbone will incur a large error at each future horizon before the corresponding ground truth is observed. We therefore adapt representative methods from both directions to provide a broader comparison.

C.7.1 Uncertainty and Error-Based Reliability Estimation

We consider six representative uncertainty and error estimation methods and retain their original learning or calibration objectives when adapting them for failure-risk ranking.

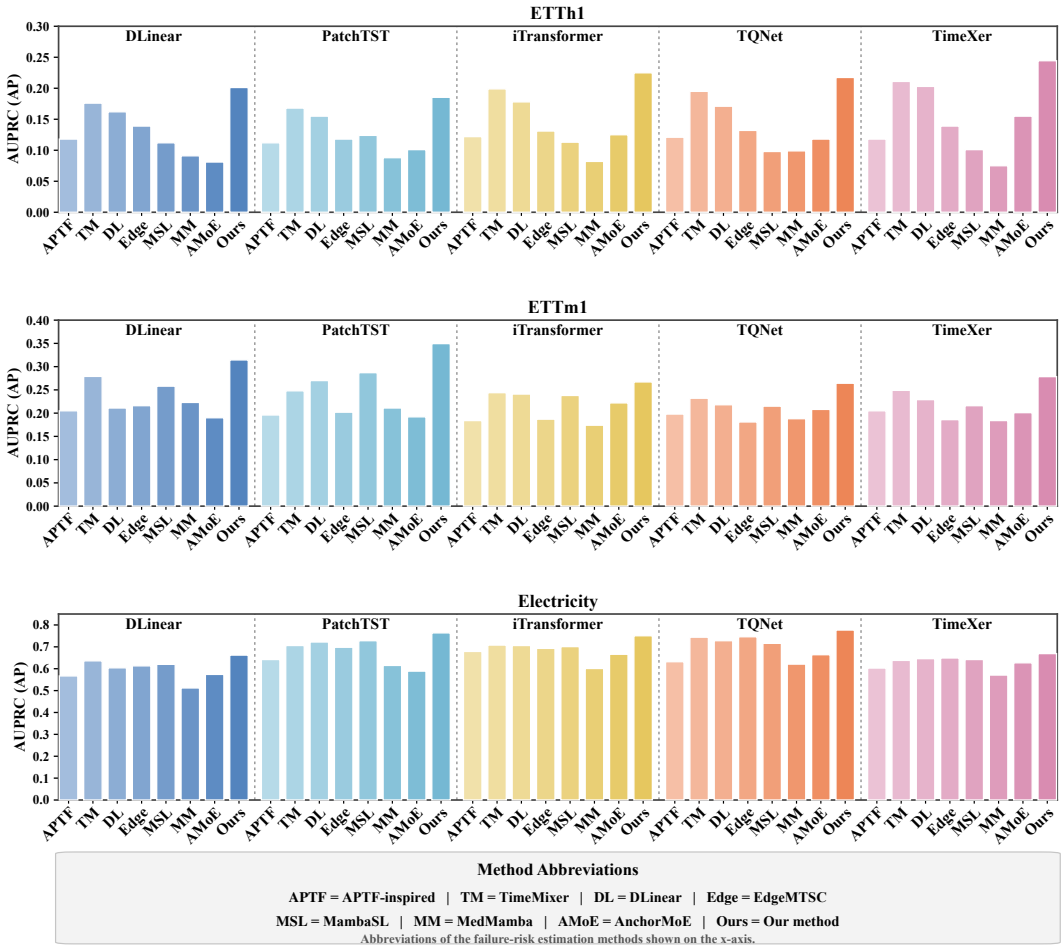

Figure 9: Comparison of AUPRC for failure risk estimation across different datasets and forecasting backbones. Higher values indicate better precision–recall discrimination under class-imbalanced failure labels. Abbreviations: TM = TimeMixer, DL = DLinear, Edge = EdgeMTSC, MSL = MambaSL, MM = MedMamba, AMoE = AnchorMoE. TSGuard consistently achieves the strongest performance across heterogeneous forecasting settings.

- **PIR-FI** (Liu et al., 2025b). PIR identifies unreliable forecasting instances within an instance-aware post-hoc revision framework. We use its identification component to estimate forecasting loss from historical observations and frozen forecasts, and broadcast the resulting instance-level score across prediction horizons.
- **IO-CUE** (Bramlage & Curio, 2025). IO-CUE estimates input–output-conditioned predictive uncertainty. We keep the forecasting backbone fixed and learn a Gaussian predictive scale using its likelihood objective, treating the estimated scale as a failure-ranking score.
- **PPM** (Li et al., 2026). PPM constructs predictive distributions through an input-conditioned parametric prior and a learnable mapping. We retain its probabilistic objective and use the expected absolute deviation of predictive samples from the frozen backbone forecast as the risk score.
- **ECI** (Wu et al., 2025). ECI performs online conformal inference by adapting its calibration state according to the magnitude of previous prediction errors. We use its adaptive error bounds as horizon-wise risk scores.
- **COP** (Hu et al., 2026). COP incorporates the estimated distribution of nonconformity scores into online conformal prediction. Its calibrated error quantiles are used as horizon-wise risk scores.

- **AcMCP** (Wang & Hyndman, 2024). AcMCP models serial dependence in multi-step forecasting errors for online conformal inference. We use its prediction-interval radii as uncertainty surrogates for failure-risk ranking.

Except for the instance-level score of PIR-FI, horizon-specific uncertainty and error scores are normalized by the corresponding validation-derived failure thresholds. Importantly, these methods retain their native regression, likelihood, probabilistic, or conformal objectives rather than being retrained with our failure-classification objective.

C.7.2 Anomaly-Detection-Based Reliability Estimation

We further examine whether representations developed for time-series anomaly detection can transfer to forecasting-failure estimation. Since their original anomaly scores characterize abnormalities in observed sequences rather than errors of a forecasting backbone, we retain their core representation mechanisms and adapt them with supervised horizon-wise failure prediction.

- **ICDiffAD** (Zhang et al., 2026a). ICDiffAD combines adaptive noise scheduling and implicit conditioning in a diffusion-based reconstruction framework. We retain its diffusion-based temporal representation network and attach a horizon-wise risk head for forecasting-failure prediction.
- **CCM-TAD** (Murad & Yilmaz, 2025). CCM-TAD models correlated variables using cluster-specific embeddings and causal temporal mixing with sequential evidence accumulation. We preserve its cluster-aware and causal temporal representations while replacing the original sequential anomaly-scoring mechanism with supervised horizon-wise risk prediction.
- **SDA2D** (Sun et al., 2026). SDA2D models temporal dynamics using neural controlled differential equations and combines reconstruction deviations with hidden-state derivatives for anomaly detection. We retain its dynamics-aware representation and use it to predict horizon-specific forecasting failures.

Evaluation protocol. All methods are evaluated on the same frozen backbone forecasts and the same validation-derived failure labels. At inference time, no method is allowed to access the future ground truth corresponding to the risk score being evaluated. Offline estimators remain fixed throughout testing, whereas online conformal methods update their calibration states only after earlier forecast outcomes have become observable. This protocol ensures that the comparison evaluates failure-risk estimation rather than differences in forecasting accuracy or access to future information.

Results. As shown in Table 3, TSGuard achieves the strongest and most consistent performance across datasets and forecasting backbones. Uncertainty and error-based methods are competitive in terms of AUROC, indicating that predictive uncertainty and error surrogates provide useful signals for separating reliable and unreliable forecasts. In contrast, the adapted anomaly-detection methods show stronger performance in several R@20 settings, particularly on ETTm1 and Traffic, suggesting that temporal abnormality representations can help identify highly failure-prone predictions. Nevertheless, both groups exhibit noticeable variation across datasets and backbones. By explicitly integrating historical dynamics, forecast behavior, and history–forecast discrepancies, TSGuard provides more consistent failure-risk estimation across heterogeneous forecasting settings.

C.8 Risk-Based Retention on ETTm1

Figures 10 and 11 report the relative MAE and MSE reductions on ETTm1 as the retention ratio decreases from 90% to 50%, complementing the ETTh1 results in Figure 4.

D Discussion

Beyond the empirical results, several aspects of forecast failure risk estimation deserve further discussion. Our study establishes a post-hoc formulation that separates forecast generation from reliability assessment, enabling failure risk estimation without modifying the underlying forecasting

Table 3: Failure risk estimation performance across different forecasting backbones. The historical look-back window L is set to 96 and the prediction horizon H is set to 96. Best results are highlighted in red bold, and second-best results are underlined in blue. R@20 means Recall@Top20%.

(a) AUROC

Dataset	Backbone	Uncertainty / Error Estimation						Anomaly Detection			TSGuard (Ours)
		PIR-FI	IO-CUE	PPM	ECI	COP	AcMCP	ICDiffAD	CCM-TAD	SDA2D	
ETTh1	DLinear	0.4922	0.6585	0.6764	0.4768	0.4730	0.5120	0.6708	0.5751	0.4595	**0.7911**
	PatchTST	0.5186	0.6794	0.6788	0.4694	0.4750	0.5060	0.6306	0.6063	0.5562	**0.7600**
ETTm1	DLinear	0.5646	0.5561	0.6268	0.5008	0.5008	0.5230	0.6218	0.5724	0.5184	**0.7040**
	PatchTST	0.5631	0.4962	0.4023	0.5105	0.5087	0.5180	0.5830	0.5863	0.5566	**0.7350**
Traffic	DLinear	0.5438	0.8349	0.8231	0.5139	0.5163	0.5370	0.7797	0.7351	0.7027	**0.9280**
	PatchTST	0.5537	0.8248	0.8370	0.5136	0.5137	0.5290	0.7836	0.7502	0.7451	**0.9240**

(b) Recall@Top20% (R@20)

Dataset	Backbone	Uncertainty / Error Estimation						Anomaly Detection			TSGuard (Ours)
		PIR-FI	IO-CUE	PPM	ECI	COP	AcMCP	ICDiffAD	CCM-TAD	SDA2D	
ETTh1	DLinear	0.2445	0.4324	0.4812	0.1398	0.1350	0.2070	0.4628	0.3443	0.2457	**0.5490**
	PatchTST	0.2474	0.4313	0.4602	0.1510	0.1554	0.1960	0.4302	0.4212	0.3511	**0.5250**
ETTm1	DLinear	0.2535	0.3312	0.3456	0.1870	0.1867	0.2190	0.3625	0.3470	0.2852	**0.3970**
	PatchTST	0.2731	0.2481	0.2368	0.2011	0.2040	0.2110	0.3406	0.3524	0.3054	**0.4110**
Traffic	DLinear	0.2309	0.5251	0.5042	0.2215	0.2234	0.2360	0.5361	0.4759	0.4698	**0.6520**
	PatchTST	0.2400	0.5643	0.5809	0.2124	0.2129	0.2280	0.6116	0.5769	0.5111	**0.6950**

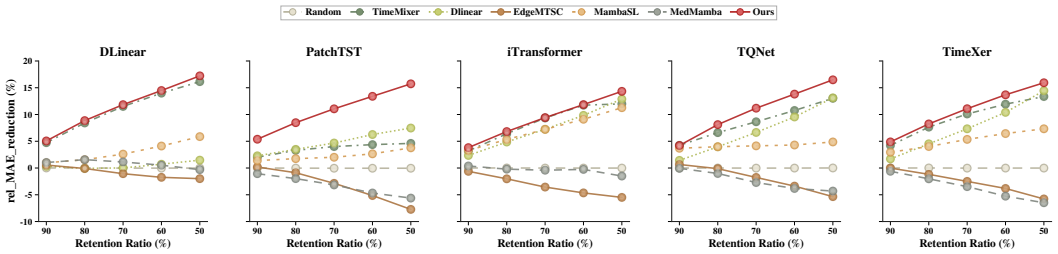

Figure 10: Post-hoc forecasting performance under different retention ratios on ETTm1 (MAE).

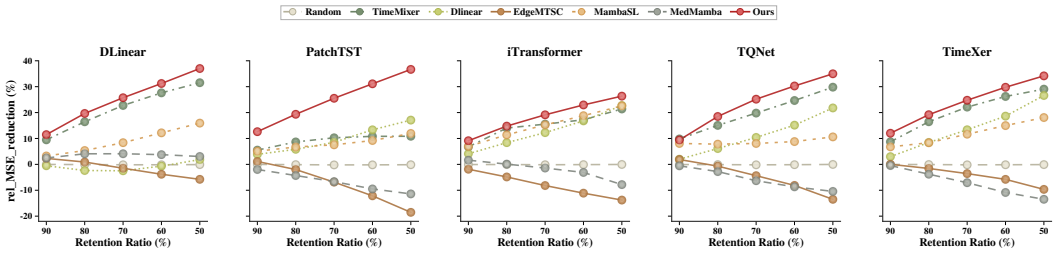

Figure 11: Post-hoc forecasting performance under different retention ratios on ETTm1 (MSE).

backbone. Building on this formulation, we highlight several directions that may further broaden its scope and practical relevance.

- **Broader contextual information.** The current framework mainly relies on historical dynamics, forecast behavior, and their structural discrepancies. In practical forecasting systems, additional signals such as external variables, operational states, or domain-specific metadata may provide complementary evidence for identifying difficult forecasting

regimes. Incorporating such information while preserving the backbone-agnostic nature of the framework is a promising direction.

- **Broader forecasting scenarios.** Our evaluation primarily focuses on standard long-term forecasting benchmarks and deterministic point forecasters. Extending post-hoc forecast failure risk estimation to more diverse settings, such as distribution shifts, evolving temporal regimes, and probabilistic forecasting systems, may further reveal its applicability under more realistic forecasting conditions.
- **Connections to forecasting reliability.** Failure risk estimation is complementary to other reliability-oriented paradigms, including predictive uncertainty and selective prediction. These approaches characterize forecasting reliability from different perspectives, and studying their interaction may enable more comprehensive risk-aware forecasting and downstream decision-making.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we make the source code for TSGuard publicly available in an anonymous repository[1]. The main paper specifies the model architecture, failure-label construction, training objective, hyperparameter search space, and evaluation metrics. Appendix A provides pseudocode for the forward pass and training pipeline, while Appendix C.2 describes the datasets and Appendix C.7 details the reliability-oriented baselines. The experimental protocol keeps backbone forecasts fixed and determines lead-time-specific failure thresholds from validation data before test evaluation. These descriptions, together with the released implementation, support reproduction of the proposed method and its evaluation.

[1] https://anonymous.4open.science/r/ICLR2027_TIMEGAURD-68E5/README.md