ROBUST TIME SERIES FORECASTING VIA BOUNDED KERNEL LOSS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series forecasting models predominantly rely on Mean Squared Error (MSE) optimization, which fundamentally assumes independent, identically distributed Gaussian residuals. However, in real-world scenarios, the complex interplay between inherent data noise (aleatoric uncertainty) and model capacity limits (epistemic uncertainty) frequently yields asymmetric, heavy-tailed error distributions. Under MSE's unbounded quadratic penalty, these unlearnable extreme residuals disproportionately dominate the gradient updates, forcing the model to overfit noise and severely degrade generalization. To resolve this fundamental misalignment, we propose a robust optimization framework centered on Bounded Kernel Loss (BKLoss). Rather than penalizing all errors uniformly, BKLoss evaluates prediction quality through localized Gaussian-kernel similarity. This mechanism naturally caps the gradient contribution of extreme outliers, allowing the model to dynamically assign lower weights to unlearnable deviations and focus its capacity on generalizable patterns. We further develop a Top-β Mean Adaptive Strategy that dynamically calibrates the batch-wise kernel bandwidth based on the residual statistics. Supported by theoretical analysis of the loss formulation and gradient boundedness, extensive experiments across multiple benchmark datasets demonstrate that our method consistently outperforms strong baselines, achieving highly stable forecasting performance in highly uncertain environments. The code is provided in the supplementary material.

1 INTRODUCTION

Time series forecasting plays a pivotal role across critical domains, such as energy consumption planning (Ma et al., 2024; Wang et al., 2022; Hong & Fan, 2016), weather forecasting (Bi et al., 2023; Rasp et al., 2020; Lam et al., 2023), and public health monitoring (Morid et al., 2023; Kandula & Shaman, 2019; Lutz et al., 2019). While the rapid evolution of neural architectures has introduced diverse methodologies - from MLP-based DLinear (Zeng et al., 2023) to advanced Transformer variants like PatchTST (Nie et al., 2022) and iTransformer (Liu et al., 2023a), as well as convolution-based TimesNet (Wu et al., 2022), their training paradigms remain highly homogeneous. Mean Squared Error (MSE) is universally adopted as the de facto learning objective. MSE intrinsically treats all time points uniformly, forcing the model to fit global observations through an unbounded quadratic penalty on errors. From a probabilistic perspective, this equates to maximum likelihood estimation (MLE) under the strict—and often idealized—assumption that prediction residuals strictly follow an independent and identically distributed (i.i.d.) Gaussian distribution (Bishop & Nasrabadi, 2006; Murphy, 2012; Goodfellow, 2016).

However, this idealized Gaussian assumption is frequently violated in real-world environments, where complex time series are inevitably corrupted by a combination of data-intrinsic and model-intrinsic uncertainties. On one hand, real-world data are inherently plagued by **aleatoric uncertainty**—such as stochastic noise and extreme anomalies—which generates large, unpredictable residuals (Li et al., 2025). Because the gradient of MSE scales linearly with the error magnitude, these uninformative outliers exert a disproportionately strong pull on parameter updates. On the other hand, finite model architectures introduce **epistemic uncertainty**, as they cannot perfectly capture all non-stationary, complex temporal dynamics (Li et al., 2025). Crucially, from the optimizer's perspective, these "unlearnable" patterns are practically indistinguishable from pure random noise. When MSE indiscriminately applies a uniform, unbounded penalty to all deviations, it forces

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.

the model to fiercely overfit these massive, unreliable residuals. This highlights a fundamental flaw in standard optimization paradigms: not all errors should be penalized with the same intensity, and forcing the model to chase extreme outliers ultimately degrades generalization.

To break this limitation, we argue that a robust forecasting model must possess optimization "discernment"—the ability to naturally distinguish and de-prioritize unlearnable extreme deviations. Inspired by the insights from Information Theoretic Learning (Principe, 2010; Liu et al., 2007), which shifts the optimization paradigm from unbounded error minimization to localized similarity maximization, we propose a novel objective termed Bounded Kernel Loss (BKLoss). Unlike MSE, which relies on global second-order statistics, BKLoss evaluates the local agreement between predictions and ground-truth values using a Gaussian kernel. The core mathematical advantage of BKLoss lies in its localized gradient response: while the gradient contribution increases for normal, learnable errors, it exponentially decays once the prediction residual exceeds a certain threshold. Thus, massive residuals are mathematically quarantined as non-generalizable interference rather than being allowed to dominate the parameter updates. By suppressing the excessive influence of extreme outliers, BKLoss prevents the uncertainty-driven "optimization hijacking" discussed earlier, enabling the model to focus its finite capacity on feature patterns with genuine generalization value. Figure 1 provides an exemplar error distribution in the test-set from MSE and BKLoss. The main contributions of this work are:

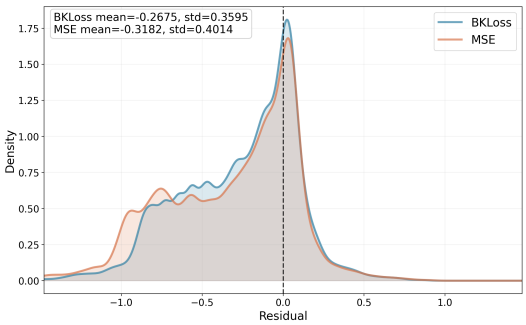


Figure 1: Residual distributions of BKLoss and MSE.

- **Paradigm Shift in Optimization Objective:** We propose Bounded Kernel Loss (BKLoss), a novel objective function for time series forecasting that shifts the learning paradigm from unbounded error minimization to localized similarity maximization. By mathematically bounding the gradient contribution of extreme residuals, BKLoss effectively mitigates the disruptive impact of unlearnable noise, and encourages the model to focus on more learnable and generalizable patterns.
- **Dynamic Bandwidth Calibration Strategy:** To alleviate the burden of exhaustive hyperparameter tuning and adapt to varying uncertainty levels across different datasets, we develop a Top-β Mean Adaptive Strategy to dynamically calibrate the kernel bandwidth based on batch-level residual statistics, ensuring robust and stable optimization without manual intervention.
- **Model-Agnostic Performance Gains:** Extensive experiments on eight real-world benchmark datasets demonstrate that BKLoss consistently improves the performance of various state-of-the-art forecasting models, particularly in highly uncertain environments.

2 Related Work

Time Series Forecasting Methods From the perspective of modeling capabilities and robustness, existing methods can be broadly summarized into two streams. (1) **Better models:** to reduce epistemic uncertainty. Compared to the early methods, e.g., ARIMA (Asteriou & Hall, 2011), which relied on basic statistical metrics to extract features, recent works use more advanced models. PatchTST (Nie et al., 2022) uses patches rather than points to capture variation patterns within local temporal segments. TimesNet (Wu et al., 2022) reveals the importance of multi-periodic structures in complex temporal modeling. DLinear (Zeng et al., 2023) demonstrates from another perspective that simple linear architectures can also achieve strong efficiency and competitiveness in long-term forecasting tasks. iTransformer (Liu et al., 2023a) shifts the modeling focus toward the variate dimension, enhancing the model's ability to capture inter-variate dependencies. These methods suggest that accurately capturing structurally informative and predictive features in time series is crucial for improving long-term forecasting performance. (2) **More robust models:** by reducing ineffective disturbances and extracting stable components. Autoformer (Wu et al., 2021) leverages sequence decomposition and an auto-correlation mechanism to reduce the influence of local noisy

fluctuations on forecasting. FiTS (Xu et al., 2023) preserves dominant frequency components from the frequency-domain perspective and filters high-frequency disturbances. Koopa (Liu et al., 2023b) exploits Koopman theory to weaken the interference caused by irregular dynamic variations. However, their work is partially hampered by the joint influence of aleatoric and epistemic uncertainty.

Training Objectives Beyond architectural innovations, many researchers focus on the training objectives and optimization strategies. Existing studies can be roughly divided into two categories. The first category focuses on the limitations of conventional point-wise regression objectives, aiming to design training objectives that are better suited to time series forecasting from the perspective of temporal label structures. Soft-DTW (Cuturi & Blondel, 2017) and DILATE (Lee et al., 2022) alleviate the limitations of point-wise errors by introducing sequence-level alignment. PSLoss (Kudrat et al., 2025) further enhances local structural consistency between predicted and ground-truth sequences through patch-wise statistical properties. FreDF (Wang et al., 2024) mitigates the optimization bias caused by future-label autocorrelation from the frequency-domain perspective, while Time-o1 (Wang et al., 2025) further transforms label sequences into decorrelated components with discriminative significance to reduce the influence of label autocorrelation on optimization. DBLoss (Qiu et al., 2025) approaches this issue from the perspective of trend-seasonality decomposition by constraining different temporal components within the forecasting horizon. The second category focuses on selection-based optimization strategies, whose goal is to weaken the interference of unreliable or redundant information during training. Selective Learning (Fu et al., 2025) alleviates the overfitting problem caused by uncertain and anomalous timesteps in the output targets, while AMRC (Liang et al., 2025) reduces the interference of redundant segments in long historical inputs on effective representation learning. However, whether a timestep belongs to an anomaly, noise, or redundant component is often probabilistic rather than deterministic. Therefore, their hard masking mechanisms may arbitrarily exclude samples that contain useful predictive information, leading to potential information loss.

3 METHODOLOGY

Given the historical multivariate time series input $X \in \mathbb{R}^{L_{\text{input}} \times D}$, our goal is to predict the future sequence $Y \in \mathbb{R}^{L_{\text{output}} \times D}$. Let $\hat{Y}$ denote the model prediction and $E = Y - \hat{Y}$ the error matrix.

3.1 LOSS WITH LOCALIZED SIMILARITY MAXIMIZATION

To mitigate the interference of unlearnable residuals, we use kernel similarity to measure the local agreement between predictions and ground-truth values, rather than relying solely on the error magnitude itself as MSE does. Specifically, we map the prediction error into a bounded similarity space. For a scalar error random variable e, the kernel similarity is defined as

$$K_\sigma(e) = \exp\left(-\frac{e^2}{2\sigma^2}\right), \tag{1}$$

where σ denotes the kernel bandwidth. $K_\sigma(e) \in (0, 1]$. This similarity approaches 1 when the prediction error is small and rapidly decays toward 0 as the error increases, thereby characterizing the local agreement between prediction and ground truth. In practice, we apply this measure element-wise to E and aggregate over all forecasting horizons and variables.

3.2 BOUNDED KERNEL LOSS

As different points might have different errors, we define BKLoss as the batch-wise averaged similarity value, based on Equation. 1.

$$\mathcal{L}_{\text{BKLoss}}(\mathbf{E^b}) = 1 - \frac{1}{N}\sum_{i=1}^{N} K_\sigma(e_i), \quad \forall e_i \in \mathbf{E^b} \tag{2}$$

where $\mathbf{E^b}$ denotes the prediction error vector of the current batch b, σ is the kernel bandwidth, and N is the total number of error elements in the batch. Unlike MSE, which amplifies large errors through an unbounded quadratic penalty, BKLoss suppresses the influence of extreme residuals through localized kernel responses. As shown in Equation 2, the resulting BKLoss is also bounded

$\mathcal{L}_{\text{BKLoss}}(\mathbf{E}) \in [0, 1)$. This property stands in sharp contrast to the unbounded quadratic penalty of MSE.

Furthermore, we analyze the gradient of BKLoss with respect to the prediction error e_i:

$$\frac{\partial \mathcal{L}_{\text{BKLoss}}}{\partial e_i} = \frac{1}{N} \frac{e_i}{\sigma^2} \exp\left(-\frac{e_i^2}{2\sigma^2}\right), \tag{3}$$

Equation 3 shows that the gradient of BKLoss is jointly determined by the linear error term and the exponentially decaying kernel term. For a fixed σ, as shown in Figure 6, the gradient magnitude reaches its maximum at $|e_i| = \sigma$. For samples with small errors, the gradient magnitude increases with the error, ensuring that the model can effectively learn from the majority of valid data. When the error becomes excessively large, the exponential term rapidly decays toward zero, and this decay dominates the linear growth of e_i. Consequently, the gradient magnitude decreases for extremely large residuals, preventing such samples from dominating the parameter updates. In practical training, to prevent σ from degenerating to zero, we impose a positive lower bound of 1×10^{-4} on σ, which ensures numerical stability during training. Detailed discussion can be found in Section 4.5.

3.3 Top-β Mean-based Adaptive σ Strategy

It is important to note that the prediction error might also change from batch to batch due to data distribution variations. Thus, we should not use one kernel bandwidth σ for different batches. Therefore, we adopt a Top-β-based adaptive calculation strategy to dynamically determine the kernel bandwidth σ. We use the mean of the top β (the percentage of the largest errors) of the current batch to calculate $\sigma^b_{\text{adaptive}}$. Here, β represents the proportion of large residuals that are difficult to learn due to aleatoric and/or epistemic uncertainty.

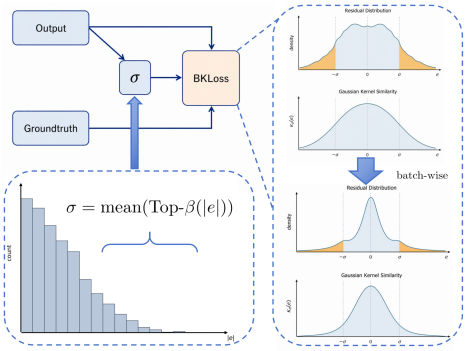


Figure 2: Overview of BKLoss with the Top-β mean adaptive bandwidth strategy.

Given a hyperparameter $\beta \in (0, 1)$, representing the estimated noise ratio, we select the top $\lfloor \beta N \rfloor$ largest errors and define the adaptive bandwidth as

$$\sigma^b_{\text{adaptive}} = \frac{1}{\lfloor \beta N \rfloor} \sum_{k=1}^{\lfloor \beta N \rfloor} |e|_{(k)}. \tag{4}$$

Here, the top $\lfloor \beta N \rfloor$ errors can be selected without fully sorting all residuals. Specifically, a Quickselect-like algorithm can be used to find the $\lfloor \beta N \rfloor$-th largest error with a time complexity of $O(N)$. With this dynamic σ, the loss function can automatically scale according to the complexity of the current batch, effectively suppressing extreme residuals while largely maintaining learning efficiency for normal samples.

3.4 Theoretical Analysis

We summarize the fixed-bandwidth properties of BKLoss and a conditional clean-test MSE result. Proofs appear in Appendix C.

Proposition 1 (MSE Limit and Local Similarity)**.** *For fixed σ, BKLoss is bounded, and $2\sigma^2 \mathcal{L}_{\text{BKLoss}} \to \text{MSE}$ as $\sigma \to \infty$. Minimizing it also maximizes the Gaussian kernel density estimate at zero when σ is held fixed.*

Proposition 2 (Bounded Per-Residual Score)**.** *For fixed σ, the per-residual BKLoss score $\psi_{\text{BK}}(e) = e\sigma^{-2} \exp(-e^2/(2\sigma^2))$ is bounded by $1/(\sigma\sqrt{\text{e}})$ and tends to zero for extreme residuals, unlike the linear squared-error score.*

Generalization under contamination. The next result makes explicit when nonzero-mean training contamination can lead to a smaller MSE on clean test targets.

Theorem 1 (Clean-Test MSE under Mean-Shift Contamination). *Fix x and let $Y_{\mathrm{tr}} = f^*(x) + \delta$, where $\delta \sim (1-\epsilon)\mathcal{N}(0, \nu^2) + \epsilon\mathcal{H}$, $0 < \epsilon < 1$, $\nu > 0$, and $\mathcal{H}$ has finite second moment and mean $\mu_{\mathcal{H}} \neq 0$. For fixed $\sigma > 0$, set $v = \sigma^2 + \nu^2$, $a = (1-\epsilon)\sigma/\sqrt{v}$, and*

$$B_{\epsilon,\sigma,\nu} = \sqrt{2v \log\left(\frac{a}{a-\epsilon}\right)}.$$

If $a > \epsilon$ and $\epsilon|\mu_{\mathcal{H}}| > B_{\epsilon,\sigma,\nu}$, the conditional population minimizers obey

$$|g_{\mathrm{BK},\sigma} - f^*(x)| < B_{\epsilon,\sigma,\nu} < |g_{\mathrm{MSE}} - f^*(x)| = \epsilon|\mu_{\mathcal{H}}|.$$

Thus, if $Y_{\mathrm{te}} = f^*(x) + \xi$ with $\mathbb{E}[\xi \mid x] = 0$ and finite variance, BKLoss has strictly lower conditional population clean-test MSE than MSE.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our method on eight widely used real-world multivariate time series datasets, including four ETT datasets, i.e., ETTh1, ETTh2, ETTm1, and ETTm2, as well as Electricity, ILI, Weather, and Solar-Energy. Detailed descriptions of these datasets are provided in Appendix A.1.

Baselines. BKLoss can be used in almost all existing time series forecasting baselines. Here, four representative models spanning different paradigms are selected: Transformer-based iTransformer (Liu et al., 2023a), MLP-based DLinear (Zeng et al., 2023), Patch-based PatchTST (Nie et al., 2022), and the Frequency-based FITS (Xu et al., 2023).

Settings. All models are implemented based on the PyTorch deep learning framework and are trained on a single NVIDIA A100 40G. All baseline models adopt a unified experimental setup. For ILI, we fix the input sequence length to 36 and set the prediction lengths to $\{24, 36, 48, 60\}$; for all other datasets, the input sequence length is fixed at 96, and the prediction lengths are set to $\{96, 192, 336, 720\}$. For all baselines and loss functions, we adopt the default hyperparameters reported in the original papers. For BKLoss, we fix its only hyperparameter β to 0.2. We train and evaluate the models with five runs using MSE and BKLoss and report the averaged results of MSE and MAE.

4.2 Main Results

Table 1 reports the long-term forecasting results across eight benchmark datasets, where all results are averaged over five random seeds. Overall, replacing the original training objective with BKLoss leads to lower MSE and MAE in most cases across different models, datasets, and prediction lengths, demonstrating its broad applicability across both architectures and data distributions. The improvement is particularly pronounced for models with relatively limited representational capacity. Taking the linear-based DLinear (Zeng et al., 2023) as an example, BKLoss brings substantial performance gains on multiple datasets. On ETTh2, the average MSE and MAE of DLinear (Zeng et al., 2023) are reduced by 17.3% and 11.6%, respectively. On ETTm2, BKLoss also reduces the average MSE and MAE of DLinear (Zeng et al., 2023) by 16.7% and 14.4%, respectively. These results suggest that when the model capacity is limited and the underlying temporal patterns are difficult to fully fit, BKLoss can suppress the influence of abnormal residuals and unpredictable disturbances, allowing the model to focus more on learnable structures with stronger generalization value. For models with stronger representational capacity, BKLoss still provides consistent benefits. For instance, on ILI, iTransformer (Liu et al., 2023a) achieves relative improvements of 4.0% and 3.4% in average MSE and MAE, respectively. These results indicate that BKLoss not only improves robustness in scenarios with strong noise or abnormal fluctuations, but also serves as a general training objective that enhances the generalization performance of different models.

4.3 Other Loss Functions

Comparisons with other loss functions. We further compare our method with five representative loss functions, including FreDF (Wang et al., 2024), DBLoss (Qiu et al., 2025), AMRC (Liang et al., 2025), Selective Learning (SL) (Fu et al., 2025), and the hybrid MSE+MAE objective. As

Table 1: Experimental results aggregated over five random seeds. We report the mean value for each metric. **Bold** indicates the superior performance.

Dataset	Len	iTransformer				DLinear				PatchTST				FITS			
		original		our		original		our		original		our		original		our	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.394	0.409	**0.391**	**0.404**	0.396	0.411	**0.390**	**0.403**	0.382	0.400	**0.380**	**0.391**	0.489	0.462	**0.430**	**0.428**
	192	0.447	0.440	**0.444**	**0.437**	0.446	0.441	**0.439**	**0.432**	**0.428**	0.432	0.433	**0.424**	0.545	0.494	**0.466**	**0.449**
	336	0.491	0.464	**0.483**	**0.458**	0.489	0.467	**0.480**	**0.457**	0.469	0.457	**0.468**	**0.445**	0.612	0.530	**0.513**	**0.477**
	720	0.517	0.501	**0.501**	**0.491**	0.513	0.511	**0.498**	**0.498**	0.518	0.503	**0.506**	**0.488**	0.688	0.582	**0.509**	**0.497**
	Avg	0.462	0.454	**0.455**	**0.447**	0.461	0.457	**0.452**	**0.448**	0.449	0.448	**0.447**	**0.437**	0.583	0.517	**0.479**	**0.463**
ETTh2	96	0.300	0.350	**0.296**	**0.342**	0.347	0.400	**0.306**	**0.359**	0.300	0.350	**0.293**	**0.336**	0.314	0.360	**0.305**	**0.351**
	192	0.380	0.399	**0.375**	**0.392**	0.476	0.476	**0.401**	**0.421**	0.375	0.400	**0.364**	**0.387**	0.404	0.412	**0.390**	**0.401**
	336	0.423	0.432	**0.422**	**0.428**	0.587	0.539	**0.472**	**0.470**	0.427	0.439	**0.416**	**0.425**	0.450	0.450	**0.432**	**0.436**
	720	0.428	0.446	**0.422**	**0.440**	0.831	0.658	**0.678**	**0.583**	0.442	0.457	**0.431**	**0.447**	0.478	0.476	**0.434**	**0.448**
	Avg	0.383	0.407	**0.379**	**0.401**	0.561	0.518	**0.464**	**0.458**	0.386	0.411	**0.376**	**0.398**	0.412	0.425	**0.390**	**0.409**
ETTm1	96	0.342	0.376	**0.325**	**0.353**	0.345	0.372	**0.332**	**0.354**	0.326	0.365	**0.313**	**0.345**	0.355	0.376	**0.339**	**0.358**
	192	0.382	0.395	**0.377**	**0.380**	0.382	0.391	**0.375**	**0.377**	0.369	0.391	**0.366**	**0.374**	0.393	0.394	**0.384**	**0.380**
	336	0.418	0.418	**0.413**	**0.405**	0.414	0.414	**0.407**	**0.401**	0.398	0.408	**0.396**	**0.396**	0.426	0.416	**0.416**	**0.402**
	720	0.489	0.457	**0.479**	**0.444**	0.473	0.450	**0.467**	**0.437**	**0.460**	0.445	0.461	**0.434**	0.487	0.449	**0.481**	**0.439**
	Avg	0.408	0.412	**0.399**	**0.396**	0.403	0.407	**0.395**	**0.392**	0.388	0.402	**0.384**	**0.387**	0.415	0.409	**0.405**	**0.395**
ETTm2	96	0.185	0.270	**0.177**	**0.255**	0.193	0.292	**0.182**	**0.265**	0.184	0.267	**0.178**	**0.258**	**0.183**	0.267	**0.183**	**0.260**
	192	0.253	0.314	**0.244**	**0.300**	0.282	0.359	**0.246**	**0.312**	0.249	0.309	**0.244**	**0.301**	0.248	0.305	**0.247**	**0.300**
	336	0.315	0.352	**0.307**	**0.340**	0.378	0.424	**0.309**	**0.357**	0.313	0.351	**0.304**	**0.340**	0.308	0.342	**0.307**	**0.338**
	720	0.413	0.407	**0.408**	**0.400**	0.534	0.512	**0.417**	**0.428**	0.416	0.411	**0.409**	**0.403**	**0.408**	0.398	**0.408**	**0.395**
	Avg	0.292	0.336	**0.284**	**0.324**	0.347	0.397	**0.289**	**0.340**	0.290	0.334	**0.284**	**0.326**	0.287	0.328	**0.286**	**0.323**
Electricity	96	**0.148**	0.240	0.149	**0.235**	0.210	0.302	**0.205**	**0.288**	**0.180**	0.272	0.183	**0.268**	0.214	0.305	**0.202**	**0.275**
	192	**0.165**	0.256	0.166	**0.251**	0.210	0.305	**0.204**	**0.291**	**0.187**	0.280	0.189	**0.275**	0.215	0.309	**0.200**	**0.277**
	336	**0.179**	0.272	0.180	**0.266**	0.223	0.319	**0.216**	**0.306**	**0.204**	0.296	0.205	**0.291**	0.238	0.335	**0.214**	**0.292**
	720	0.213	0.301	**0.208**	**0.292**	0.258	0.350	**0.251**	**0.337**	**0.246**	0.328	**0.246**	**0.324**	0.311	0.400	**0.257**	**0.329**
	Avg	**0.176**	0.267	**0.176**	**0.261**	0.225	0.319	**0.219**	**0.305**	**0.204**	0.294	0.206	**0.289**	0.245	0.337	**0.218**	**0.293**
Weather	96	0.175	0.215	**0.172**	**0.207**	**0.196**	0.258	0.200	**0.240**	0.176	0.217	**0.175**	**0.211**	**0.170**	0.218	0.183	**0.216**
	192	0.225	0.258	**0.220**	**0.252**	**0.237**	0.296	**0.237**	**0.279**	0.222	0.259	**0.221**	**0.253**	**0.218**	0.260	0.226	**0.258**
	336	**0.281**	0.299	**0.281**	**0.296**	0.282	0.332	**0.280**	**0.312**	0.280	0.298	**0.277**	**0.293**	**0.275**	0.300	0.279	**0.298**
	720	0.359	0.351	**0.357**	**0.347**	0.347	0.383	**0.341**	**0.363**	0.356	0.348	**0.354**	**0.343**	**0.353**	0.349	0.356	**0.348**
	Avg	0.260	0.281	**0.258**	**0.275**	0.266	0.317	**0.265**	**0.299**	0.259	0.280	**0.257**	**0.275**	**0.254**	0.282	0.261	**0.280**
ILI	24	3.068	1.183	**2.993**	**1.150**	2.826	1.161	**2.738**	**1.112**	**2.670**	**1.084**	2.720	1.088	5.206	1.663	**4.797**	**1.587**
	36	2.901	1.148	**2.760**	**1.104**	2.741	1.132	**2.495**	**1.063**	2.722	1.085	**2.650**	**1.055**	5.424	1.667	**4.895**	**1.569**
	48	2.863	1.143	**2.740**	**1.102**	2.752	1.138	**2.473**	**1.067**	2.749	1.098	**2.694**	**1.081**	5.555	1.702	**5.016**	**1.601**
	60	2.958	1.170	**2.830**	**1.130**	2.944	1.186	**2.599**	**1.108**	2.719	1.089	**2.651**	**1.066**	5.854	1.762	**5.278**	**1.659**
	Avg	2.948	1.161	**2.831**	**1.122**	2.816	1.154	**2.576**	**1.088**	2.715	1.089	**2.679**	**1.072**	5.510	1.698	**4.997**	**1.604**
Solar	96	0.201	0.236	**0.196**	**0.218**	0.290	0.378	**0.283**	**0.336**	0.225	0.281	**0.221**	**0.269**	0.343	0.370	**0.336**	**0.354**
	192	0.236	0.262	**0.234**	**0.244**	0.320	0.398	**0.313**	**0.356**	0.257	0.299	**0.256**	**0.289**	0.384	0.396	**0.380**	**0.385**
	336	**0.250**	0.275	0.253	**0.259**	**0.352**	0.422	**0.352**	**0.375**	0.275	0.309	**0.273**	**0.298**	0.424	0.409	**0.423**	**0.399**
	720	**0.250**	0.276	0.255	**0.256**	**0.357**	0.413	0.359	**0.370**	0.277	0.309	**0.272**	**0.295**	**0.422**	0.399	0.423	**0.387**
	Avg	0.235	0.262	**0.234**	**0.245**	0.330	0.403	**0.327**	**0.359**	0.259	0.299	**0.255**	**0.288**	0.393	0.394	**0.390**	**0.381**

Table 2: Averaged results of **iTransformer** (Liu et al., 2023a) across four prediction lengths. **Bold** indicates the best performance, and underlined indicates the second best.

Dataset	BKLoss		FreDF		DBLoss		AMRC		SL		MSE+MAE	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.455	**0.447**	0.499	0.474	0.456	0.449	0.463	0.455	**0.452**	0.448	0.459	0.451
ETTh2	**0.379**	**0.401**	0.385	0.406	**0.379**	**0.401**	0.384	0.407	0.384	0.407	0.381	0.403
ETTm1	**0.399**	**0.396**	0.424	0.416	0.401	0.403	0.407	0.411	0.402	0.405	**0.399**	0.402
ETTm2	0.284	0.324	0.288	0.327	**0.282**	**0.323**	0.290	0.333	0.286	0.327	0.288	0.330
ECL	0.176	**0.261**	0.174	0.262	0.179	0.268	0.179	0.270	**0.172**	**0.261**	0.179	0.266
Weather	0.258	0.275	0.259	0.278	**0.256**	**0.274**	0.259	0.279	0.257	0.276	**0.256**	0.275
ILI	**2.831**	**1.122**	3.154	1.226	3.233	1.249	2.981	1.172	2.921	1.155	2.885	1.140
Solar	0.234	**0.245**	0.236	0.259	**0.231**	0.253	0.232	0.262	0.232	0.252	0.232	0.246

shown in Table 2, our method achieves the best performance in 9 out of 16 cases (56.3%) and ranks second best in 4 out of 16 cases (25.0%) across eight datasets and two evaluation metrics. These results demonstrate that our loss function consistently maintains strong competitiveness against recent advanced objectives. More results are provided in Appendix B.3.

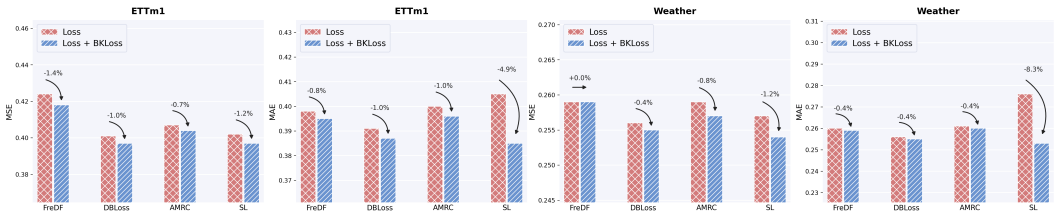


Figure 3: Performance gains brought by integrating BKLoss with different advanced training objectives on ETTm1 and Weather. The colored bars denote the average results over prediction horizons.

Integration with existing losses. In this section, we verify whether BKLoss can enhance the performance of other loss functions. We integrate BKLoss with other loss functions using a simple additive combination. Figure 3 visualizes the performance of iTransformer (Liu et al., 2023a) trained with different original loss functions and their BKLoss-enhanced counterparts. Figure 3 shows that incorporating BKLoss consistently brings additional improvements across almost all loss functions. Thus, BKLoss can serve as an effective plug-in component into existing designs.

4.4 Analysis under Different Uncertainty Settings

In real-world time series, epistemic uncertainty and aleatoric noise are often intertwined and difficult to distinguish directly. Therefore, we construct four synthetic datasets with controlled settings to separately analyze the effects of these two types of uncertainty.

Four synthetic datasets: *Short-Period Seasonal*, containing one or more complete cycles, can be observed within the input window. *Long-Period Trend* has a higher forecasting difficulty as the input window contains a partial cycle. These two datasets are noise-free and are used to analyze the backbone capacities concerning *epistemic uncertainty*. *Chaotic Periodic* and its noisy version, *Chaotic + Strong Spike* , contain short-period signals (55%) and 45% unpredictable chaotic components (May, 1976; Strogatz). They are mainly designed to identify the performance divergence on *aleatoric uncertainty*. *Chaotic + Strong Spike* was injected with strong spike noise (Huber, 1992) into the data to analyze the influence of aleatoric uncertainty. Detailed settings are in Appendix A.3.

Results (1) As shown in Table 3, on both the short-period dataset and the long-period dataset, BKLoss achieves stable improvement over MSE. When there is no aleatoric uncertainty in the data, BKLoss can improve performance by reducing the influence of epistemic uncertainty. (2) On the *Chaotic Periodic* dataset without noise, MSE and BKLoss have almost identical performance, with 15 out of 16 metrics remaining unchanged, indicating that the epistemic uncertainty in this dataset is relatively small. When strong spike noise is added (*Chaotic + Strong Spike*), significant aleatoric uncertainty is introduced. In this case, BKLoss shows a clearer advantage over MSE. Among the 16 metrics, 12 metrics are improved, accounting for 75.0%, while the remaining metrics also maintain the same performance as MSE. These results show that BKLoss can improve performance over MSE by reducing the influence of aleatoric uncertainty and epistemic uncertainty.

4.5 Analysis Experiments

Residual analysis. Here, we analyze the residuals in the test set using six statistics: *Mean*, *Variance*, *Median*, *RMSE*, $|r|$-*Q90*, and $|r|$-*Q95*. Mean and Median characterize the overall bias and central location of the residual distribution, where values closer to zero are better. Variance measures the dispersion of residuals, with smaller values corresponding to a more concentrated distribution. RMSE reflects the overall error magnitude and is more sensitive to large residuals. $|r|$-Q90 and $|r|$-Q95 denote the 90th and 95th percentiles of absolute residuals, respectively. They are used to characterize the magnitude of tail errors. As shown in Table 4, BKLoss achieves better results in most residual statistics. The Mean and Median values are closer to zero, along with the smaller Variance and RMSE, indicate that BKLoss tends to produce a more concentrated residual distribution with reduced overall deviation. Moreover, BKLoss also yields smaller $|r|$-Q90 and $|r|$-Q95 on most datasets, suggesting that tail residuals are effectively suppressed. This observation indicates that down-weighting large-error samples can effectively reduce the interference caused by uncertainty.

Table 3: Forecasting results of **iTransformer** (Liu et al., 2023a) and **DLinear** (Zeng et al., 2023) on four synthetic settings. "Impr.(%)" denotes the percentage reduction achieved by BKLoss relative to MSE training.

Dataset	Model	Metric	Pred Len = 96			Pred Len = 192			Pred Len = 336			Pred Len = 720		
			MSE	BKLoss	Impr.(%)	MSE	BKLoss	Impr.(%)	MSE	BKLoss	Impr.(%)	MSE	BKLoss	Impr.(%)
Short-Period Seasonal	iTransformer	MSE	0.0003	**0.0001**	66.67	0.0003	**0.0001**	66.67	0.0003	**0.0001**	66.67	0.0003	**0.0001**	66.67
		MAE	0.0139	**0.0076**	45.32	0.0137	**0.0077**	43.80	0.0136	**0.0077**	43.38	0.0136	**0.0077**	43.38
	DLinear	MSE	0.0002	**0.0000**	100.00	0.0004	**0.0000**	100.00	0.0002	**0.0000**	100.00	0.0003	**0.0000**	100.00
		MAE	0.0081	**0.0002**	97.53	0.0102	**0.0004**	96.08	0.0081	**0.0003**	96.30	0.0104	**0.0004**	96.15
Long-Period Trend	iTransformer	MSE	0.0103	**0.0066**	35.92	0.0249	**0.0198**	20.48	0.0557	**0.0519**	6.82	0.0782	**0.0740**	5.37
		MAE	0.0644	**0.0481**	25.31	0.1108	**0.0948**	14.44	0.1749	**0.1630**	6.80	0.2235	**0.2146**	3.98
	DLinear	MSE	0.0663	**0.0635**	4.22	0.0885	**0.0863**	2.49	0.0830	**0.0809**	2.53	0.0782	**0.0753**	3.71
		MAE	0.1928	**0.1853**	3.89	0.2362	**0.2313**	2.07	0.2314	**0.2271**	1.86	0.2250	**0.2195**	2.44
Chaotic Periodic	iTransformer	MSE	**0.108**	**0.108**	0.000	**0.108**	**0.108**	0.000	**0.108**	**0.108**	0.000	**0.108**	**0.108**	0.000
		MAE	**0.292**	**0.292**	0.000	**0.292**	**0.292**	0.000	**0.292**	**0.292**	0.000	**0.292**	**0.292**	0.000
	DLinear	MSE	**0.109**	**0.109**	0.000	**0.109**	**0.109**	0.000	**0.109**	**0.109**	0.000	**0.109**	**0.109**	0.000
		MAE	**0.292**	**0.292**	0.000	**0.292**	**0.292**	0.000	**0.292**	**0.292**	0.000	0.293	**0.292**	0.341
Chaotic + Strong Spike	iTransformer	MSE	0.155	**0.151**	2.581	0.159	**0.153**	3.774	0.160	**0.158**	1.250	0.161	**0.160**	0.621
		MAE	0.329	**0.326**	0.912	0.332	**0.327**	1.506	0.333	**0.331**	0.601	0.334	**0.333**	0.299
	DLinear	MSE	**0.129**	**0.129**	0.000	**0.130**	**0.130**	0.000	0.130	**0.128**	1.538	0.133	**0.128**	3.759
		MAE	**0.308**	**0.308**	0.000	**0.309**	**0.309**	0.000	0.309	**0.308**	0.324	0.312	**0.307**	1.603

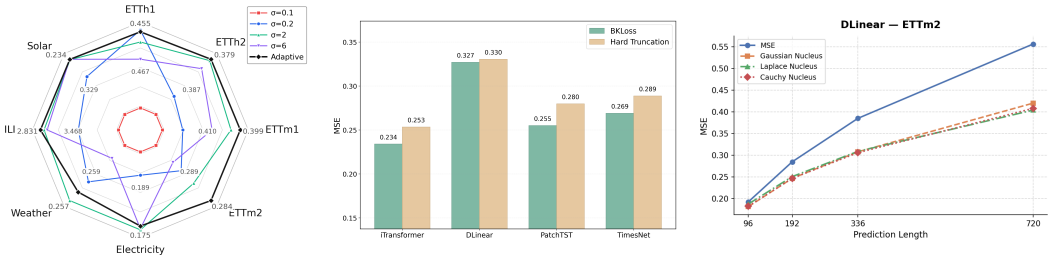
Table 4: Residual distribution statistics of **DLinear** on the full test set at prediction length 96. For *Mean* and *Median*, **bold** indicates the closer value to zero; for others, **bold** indicates the smaller value.

Dataset	**Mean**		**Variance**		**Median**		**RMSE**		$\lvert r\rvert$**-Q90**		$\lvert r\rvert$**-Q95**	
	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE
ETTh1	**0.0094**	0.0100	**0.3950**	0.3961	**-0.0072**	-0.0075	**0.6286**	0.6294	**0.8943**	0.8970	**1.3316**	1.3337
ETTm1	**0.0048**	0.0056	**0.3442**	0.3459	-0.0034	**-0.0030**	**0.5867**	0.5882	**0.8068**	0.8107	**1.2116**	1.2170
Weather	**-0.0318**	-0.0444	0.1956	**0.1942**	**-0.0273**	-0.0429	0.4434	**0.4430**	**0.5852**	0.5914	0.8162	**0.8158**
Solar	**-0.1358**	-0.1412	0.2701	**0.2699**	**-0.1362**	-0.1430	**0.5371**	0.5384	**0.8810**	0.8829	1.1799	**1.1770**

Impacts of σ. To verify the effectiveness of the proposed adaptive strategy, we compare adaptive BKLoss with BKLoss using different fixed kernel bandwidths. Specifically, we evaluate iTransformer on eight datasets and report the average results over four prediction horizons, as shown in Figure 4(a), where points closer to the outer ring indicate better performance. As can be seen, different datasets often favor different fixed σ values. For example, BKLoss with $\sigma = 0.2$ performs well on ETTh1, but shows weaker results on datasets such as Electricity. In contrast, BKLoss with $\sigma = 6$ achieves strong performance on Electricity, yet is less competitive on ETTm1 and Weather. By comparison, adaptive BKLoss achieves better or more balanced performance on almost all datasets, validating the effectiveness of our proposed adaptive strategy.

Hard truncation. Recent methods such as Selective Learning (SL) often mask samples identified as noisy, i.e., apply hard truncation. However, such a strategy may not be suitable for BKLoss. To investigate this, we compare BKLoss with a hard-truncated variant, where samples with errors larger than σ are directly discarded, while those with errors smaller than σ are treated in the same way as in BKLoss. As shown in Figure 4(b), we compare the results of four forecasting backbones on the Solar dataset. Hard truncation consistently degrades performance across all four models. The drop is more pronounced for iTransformer, PatchTST, and TimesNet, while DLinear, which is subject to larger epistemic uncertainty, also shows a decline. These results indicate that discarding large-error samples does not lead to better performance, further supporting the rationality of BKLoss.

Sensitivity analysis of β. A key hyperparameter in BKLoss is β, which controls the proportion of large-error samples used to estimate the adaptive kernel width. To study its effect, we evaluate a range of β values from 0 to 1 on different forecasting tasks in Figure 5. The results reveal a clear connection between the optimal β and uncertainty. For example, ETTh1 exhibits larger aleatoric uncertainty than Electricity. Thus, its optimal β under DLinear is larger. Similarly, since DLinear is subject to larger epistemic uncertainty than iTransformer, its optimal β on ETTm1 is also larger than



(a) different σ settings (b) hard truncation on Solar (c) different kernel functions

Figure 4: (a) Comparisons of the performance of BKLoss under different fixed σ values, black line denotes the performance of the adaptive strategy. (b) Comparison between BKLoss and hard truncation on the Solar dataset. (c) Performance of BKLoss with different kernel functions.

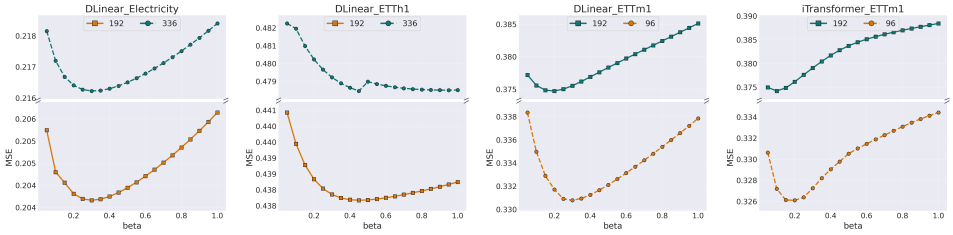


Figure 5: Performance comparison under different β values across different forecasting tasks.

that of iTransformer. Overall, these observations suggest that tasks with higher uncertainty tend to favor larger β values, which further supports the design of the proposed adaptive strategy.

Different kernel functions. We analyze the influence of different kernel functions: Gaussian, Laplace, and Cauchy kernels. As shown in Figure 4(c), BKLoss with different kernel choices still consistently outperforms the MSE on ETTm2 across different prediction horizons. This indicates that the effectiveness of BKLoss is not restricted to a specific kernel form. As can be seen in Figure 6, the gradient curves of these three kernels exhibit a common characteristic: the gradient magnitude becomes smaller for large errors. In other words, samples with large errors no longer dominate parameter updates as they do under MSE, but are instead adaptively down-weighted during optimization. It is the key reason why BKLoss with those kernels can consistently improve forecasting performance.

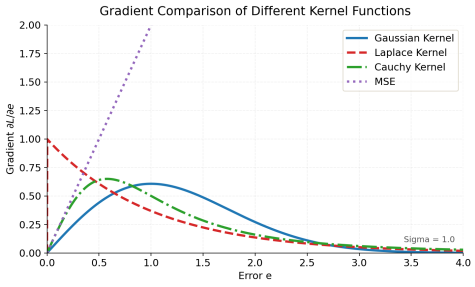


Figure 6: Gradient curves of three kernel functions: Gaussian, Laplace, and Cauchy.

5 Conclusion

In this paper, we present BKLoss, a bounded kernel-based objective for robust time series forecasting in uncertain environments. By leveraging localized Gaussian-kernel similarity, our method naturally suppresses the dominance of extreme residuals and encourages models to focus on more generalizable patterns. Combined with the Top-β Mean Adaptive Strategy, BKLoss can dynamically calibrate the kernel bandwidth according to batch-level residual statistics and maintain stable performance across different datasets and forecasting settings. Extensive experiments demonstrate that BKLoss consistently outperforms standard training objectives and serves as an effective plug-in loss for diverse forecasting models. Nevertheless, BKLoss still has limitations. When using the Top-β Mean Adaptive Strategy, we only consider the influence of residual magnitude on σ, while ignoring higher-order characteristics of the error distribution, such as skewness and kurtosis. In future work, we will further investigate how to dynamically calculate β according to the residual distribution.

LLM USAGE DISCLOSURE

Generative AI tools were used only to polish the manuscript's language and improve its readability. They were not used to generate data, develop theoretical models, formulate or prove mathematical claims, design or interpret experiments, or implement methods. The authors reviewed all AI-assisted edits and take responsibility for the final content.

REFERENCES

Dimitros Asteriou and Stephen G Hall. Arima models and the box–jenkins methodology. *Applied Econometrics*, 2(2):265–286, 2011.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*, pp. 894–903. PMLR, 2017.

Yisong Fu, Zezhi Shao, Chengqing Yu, Yujie Li, Zhulin An, Qi Wang, Yongjun Xu, and Fei Wang. Selective learning for deep time series forecasting. *arXiv preprint arXiv:2510.25207*, 2025.

Ian Goodfellow. Deep learning, 2016.

Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

Sasikiran Kandula and Jeffrey Shaman. Near-term forecasts of influenza-like illness: An evaluation of autoregressive time series approaches. *Epidemics*, 27:41–51, 2019.

Dilfira Kudrat, Zongxia Xie, Yanru Sun, Tianyu Jia, and Qinghua Hu. Patch-wise structural loss for time series forecasting. *arXiv preprint arXiv:2503.00877*, 2025.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

Hyunwook Lee, Chunggi Lee, Hongkyu Lim, and Sungahn Ko. Tilde-q: a transformation invariant loss function for time-series forecasting. *arXiv preprint arXiv:2210.15050*, 2022.

Qi Li, Zhenyu Zhang, Lei Yao, Zhaoxia Li, Tianyi Zhong, and Yong Zhang. Diffusion-based decoupled deterministic and uncertain framework for probabilistic multivariate time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025.

Renzhao Liang, Sizhe Xu, Chenggang Xie, Jingru Chen, Feiyang Ren, Shu Yang, and Takahiro Yabe. Abstain mask retain core: Time series prediction by adaptive masking loss with representation consistency. *arXiv preprint arXiv:2510.19980*, 2025.

Weifeng Liu, Puskal P Pokharel, and Jose C Principe. Correntropy: Properties and applications in non-gaussian signal processing. *IEEE Transactions on signal processing*, 55(11):5286–5298, 2007.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023a.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. *Advances in neural information processing systems*, 36:12271–12290, 2023b.

Chelsea S Lutz, Mimi P Huynh, Monica Schroeder, Sophia Anyatonwu, F Scott Dahlgren, Gregory Danyluk, Danielle Fernandez, Sharon K Greene, Nodar Kipshidze, Leann Liu, et al. Applying infectious disease forecasting to public health: a path forward using influenza forecasting examples. *BMC Public Health*, 19(1):1659, 2019.

Ziqing Ma, Wenwei Wang, Tian Zhou, Chao Chen, Bingqing Peng, Liang Sun, and Rong Jin. Fusionsf: Fuse heterogeneous modalities in a vector quantized framework for robust solar power forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5532–5543, 2024.

Robert M May. Simple mathematical models with very complicated dynamics. *Nature*, 261(5560): 459–467, 1976.

Mohammad Amin Morid, Olivia R Liu Sheng, and Joseph Dunbar. Time series prediction using deep learning methods in healthcare. *ACM Transactions on Management Information Systems*, 14(1):1–29, 2023.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. 2012.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Jose C Principe. *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer Science & Business Media, 2010.

Xiangfei Qiu, Xingjian Wu, Hanyin Cheng, Xvyuan Liu, Chenjuan Guo, Jilin Hu, and Bin Yang. Dbloss: Decomposition-based loss function for time series forecasting. *arXiv preprint arXiv:2510.23672*, 2025.

Stephan Rasp, Peter D Dueben, Sebastian Scher, Jonathan A Weyn, Soukayna Mouatadid, and Nils Thuerey. Weatherbench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020.

Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering (studies in nonlinearity)*, volume 1.

Hao Wang, Licheng Pan, Zhichao Chen, Degui Yang, Sen Zhang, Yifei Yang, Xinggao Liu, Haoxuan Li, and Dacheng Tao. Fredf: Learning to forecast in the frequency domain. *arXiv preprint arXiv:2402.02399*, 2024.

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. *arXiv preprint arXiv:2505.17847*, 2025.

Zhiyuan Wang, Xovee Xu, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fan Zhou. Pref: Probabilistic electricity forecasting via copula-augmented state space model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 12200–12207, 2022.

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. *arXiv preprint arXiv:2210.02186*, 2022.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. *arXiv preprint arXiv:2307.03756*, 2023.

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.

A Additional Experimental Details

A.1 Dataset Details

To comprehensively evaluate the effectiveness of BKLoss under different real-world time series forecasting scenarios, we adopt eight widely used multivariate time series forecasting benchmarks, including ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, Solar, and ILI. These datasets cover diverse domains such as electricity, environment, energy, and public health, and exhibit different sampling frequencies, sequence lengths, and variable dimensions. Specifically, the ETT datasets record operational indicators related to electricity transformers; the Weather dataset contains multiple meteorological variables; the Electricity dataset records electricity consumption from multiple meters or clients; the Solar dataset tracks solar power production from photovoltaic plants; and the ILI dataset reflects public-health indicators related to influenza-like illness. Detailed statistics of these datasets are summarized in Table 5.

Table 5: Statistics of the real-world multivariate time series datasets used in our experiments.

Dataset	Domain	Frequency	Lengths	Dim	Split	Description
ETTh1	Electricity	1 hour	14,400	7	6:2:2	Electricity transformer temperature dataset for one transformer, with 7 load/oil-related indicators
ETTh2	Electricity	1 hour	14,400	7	6:2:2	Electricity transformer temperature dataset for another transformer, with 7 load/oil-related indicators
ETTm1	Electricity	15 mins	57,600	7	6:2:2	Same benchmark as ETTh1 at 15-minute resolution
ETTm2	Electricity	15 mins	57,600	7	6:2:2	Same benchmark as ETTh2 at 15-minute resolution
Weather	Environment	10 mins	52,696	21	7:1:2	Meteorological dataset with 21 weather variables, sampled every 10 minutes
Electricity	Electricity	1 hour	26,304	321	7:1:2	Hourly electricity consumption dataset with 321 variables/meters
Solar	Energy	10 mins	52,560	137	7:1:2	Solar power production dataset collected from 137 PV plants
ILI	Public Health	1 week	966	7	7:1:2	Weekly influenza-like illness dataset with 7 health-related indicators

A.2 Settings for Different Loss Functions

To ensure a fair and reproducible comparison, we keep the hyperparameters of all training objectives fixed across different datasets, prediction lengths, and backbone models. In particular, **BKLoss** adopts the same hyperparameter configuration for all tasks, without dataset-specific, horizon-specific, or backbone-specific tuning. For **FreDF**, we set $\lambda_{\mathrm{rec}} = 1.0$ and $\lambda_{\mathrm{freq}} = 0.5$, and compute the frequency-domain objective using an rFFT-based loss with mean absolute spectral discrepancy. For **DBLoss**, we set the EMA smoothing factor to $\alpha = 0.05$ and the balancing coefficient to $\beta = 0.5$. For **AMRC**, we set $\lambda_{\mathrm{AML}} = 1.0$ and $\lambda_{\mathrm{ESP}} = 1.0$, and use $m = 12$ candidate prefix mask lengths for each batch. For **SL**, we set the uncertainty masking ratio to $r_u = 0.1$ and the anomaly masking ratio to $r_a = 0.05$, and employ a pre-trained DLinear model as the estimator.

A.3 Synthetic Data Generation

To provide a controlled analysis under different uncertainty settings, we construct three clean synthetic multivariate time series datasets and one noisy variant. In all cases, the original temporal indices are preserved only for time feature encoding, while the original observed values are replaced by synthetic signals with the same sequence length and number of variables. The scaler is fitted on the clean training split before optional noise injection, so that the noisy setting only changes the training observations rather than the normalization statistics.

General protocol. Let L denote the total number of time steps and C denote the number of variables. For each synthetic dataset, we generate a multivariate sequence

$$\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^{L} \in \mathbb{R}^{L \times C}, \tag{5}$$

where each channel is generated independently with channel-specific phase shifts. After generation, each channel is normalized by its maximum absolute value:

$$x_i(t) \leftarrow \frac{x_i(t)}{\max_t |x_i(t)|}, \tag{6}$$

so that the clean synthetic signal is approximately bounded within $[-1, 1]$.

Short-period seasonal dataset. The *Short-Period Seasonal* dataset is designed to represent a clean and highly observable seasonal setting. Each variable is generated by a weighted combination of

multiple short-period sinusoidal components:

$$x_i(t) = \sum_{j=1}^{4} \tilde{w}_j \sin\left(\frac{2\pi t}{T_j} + \phi_{ij}\right), \tag{7}$$

where $\tilde{w}_j$ denotes the normalized weight of the j-th component, and the phase shift is defined as

$$\phi_{ij} = i \cdot \alpha + j \cdot 0.4. \tag{8}$$

In our experiments, we set

$$(T_1, T_2, T_3, T_4) = (12, 24, 48, 96), \tag{9}$$

$$(w_1, w_2, w_3, w_4) = (0.2, 0.3, 0.3, 0.2), \tag{10}$$

and $\alpha = 0.5$. Since these periods are short, the input window can cover one or more complete cycles for most components. Therefore, this dataset provides a clean setting where the dominant seasonal patterns are explicit and sufficiently observable.

Long-period trend dataset. The *Long-Period Trend* dataset is generated using the same multi-period sinusoidal formulation as the short-period dataset, but with much longer periods:

$$x_i(t) = \sum_{j=1}^{4} \tilde{w}_j \sin\left(\frac{2\pi t}{T_j} + \phi_{ij}\right), \tag{11}$$

where

$$(T_1, T_2, T_3, T_4) = (168, 336, 720, 1440), \tag{12}$$

$$(w_1, w_2, w_3, w_4) = (0.3, 0.3, 0.2, 0.2), \tag{13}$$

and $\alpha = 0.5$.

It should be noted that this dataset does not contain an explicit linear trend term. Instead, its trend-like property comes from the fact that the underlying periods are much longer than the input window. As a result, only a local segment of each long-period component can be observed within one input window, and such local segments usually appear as smooth increasing or decreasing trends. This setting is therefore used to evaluate whether a model can forecast low-frequency structures when the complete global cycle is not directly visible from the input window.

Chaotic periodic dataset. The *Chaotic Periodic* dataset is designed to combine a learnable periodic structure with a deterministic but difficult-to-predict chaotic component. For each variable, we first generate a periodic component:

$$p_i(t) = \frac{1}{3} \sum_{j=1}^{3} \sin\left(\frac{2\pi t}{T_j} + \phi_i + j \cdot 0.4\right), \tag{14}$$

where

$$(T_1, T_2, T_3) = (24, 48, 96), \tag{15}$$

and $\phi_i = i \cdot \alpha$ with $\alpha = 0.5$.

The chaotic component is generated by the logistic map:

$$z_i(t+1) = r z_i(t)(1 - z_i(t)), \tag{16}$$

where $r = 3.99$. The initial value is channel-dependent:

$$z_i(0) = 0.3 + 0.07i + \delta_i, \tag{17}$$

where δ_i is a small random perturbation. The chaotic sequence is then linearly rescaled to approximately $[-1, 1]$.

Finally, the periodic and chaotic components are mixed as

$$x_i(t) = (1 - \lambda) p_i(t) + \lambda c_i(t), \tag{18}$$

where $c_i(t)$ denotes the rescaled chaotic sequence and $\lambda = 0.45$. Therefore, the final signal contains both a stable periodic backbone and a strong chaotic component:

$$x_i(t) = 0.55 p_i(t) + 0.45 c_i(t). \tag{19}$$

This dataset is used to evaluate model behavior when the target sequence contains both learnable regular patterns and deterministic but highly sensitive nonlinear dynamics.

Training-set spike noise. To evaluate robustness under aleatoric uncertainty, we further construct a noisy version of the chaotic periodic dataset by injecting spike noise into the training split only. The validation and test splits remain clean. For each training observation, the spike noise is defined as

$$\epsilon_i(t) = m_i(t) \cdot s_i(t) \cdot A, \tag{20}$$

where $m_i(t) \sim \text{Bernoulli}(p)$ determines whether a spike is injected, $s_i(t)$ is randomly sampled from $\{-1, +1\}$, and A denotes the spike amplitude. In our experiments, we use

$$p = 0.15, \quad A = 6.0 \times 0.5 = 3.0. \tag{21}$$

Thus, the noise added to each training point belongs to

$$\epsilon_i(t) \in \{-3.0, 0, +3.0\}. \tag{22}$$

The noisy training signal is then given by

$$\tilde{x}_i(t) = x_i(t) + \epsilon_i(t). \tag{23}$$

Since the clean synthetic signals are normalized to approximately $[-1, 1]$, the injected spikes have substantially larger magnitudes than the normal signal range. This setting simulates sparse but strong training-set corruption. Therefore, the noisy synthetic experiment evaluates whether a loss function can prevent the model from overfitting extreme corrupted observations and still generalize to clean validation and test targets.

B More Results and Analysis

B.1 Main Results with Standard Deviations

Tables 6–9 give the five-seed results corresponding to Table 1. Each entry reports mean $\pm$ standard deviation; the better mean in each original-versus-BKLoss pair is bold. Standard deviations are rounded to three decimal places.

Table 6: Five-seed forecasting results for iTransformer (mean ± standard deviation).

Dataset	Len	Original		BKLoss	
		MSE	MAE	MSE	MAE
ETTh1	96	0.394 ± 0.002	0.409 ± 0.001	$\mathbf{0.391} \pm 0.001$	$\mathbf{0.404} \pm 0.001$
	192	0.447 ± 0.002	0.440 ± 0.001	$\mathbf{0.444} \pm 0.002$	$\mathbf{0.437} \pm 0.002$
	336	0.491 ± 0.003	0.464 ± 0.002	$\mathbf{0.483} \pm 0.002$	$\mathbf{0.458} \pm 0.001$
	720	0.517 ± 0.003	0.501 ± 0.002	$\mathbf{0.501} \pm 0.003$	$\mathbf{0.491} \pm 0.002$
	Avg	0.462 ± 0.002	0.454 ± 0.001	$\mathbf{0.455} \pm 0.001$	$\mathbf{0.447} \pm 0.000$
ETTh2	96	0.300 ± 0.000	0.350 ± 0.001	$\mathbf{0.296} \pm 0.001$	$\mathbf{0.342} \pm 0.000$
	192	0.380 ± 0.001	0.399 ± 0.001	$\mathbf{0.375} \pm 0.002$	$\mathbf{0.392} \pm 0.001$
	336	0.423 ± 0.001	0.432 ± 0.001	$\mathbf{0.422} \pm 0.004$	$\mathbf{0.428} \pm 0.002$
	720	0.428 ± 0.002	0.446 ± 0.001	$\mathbf{0.422} \pm 0.002$	$\mathbf{0.440} \pm 0.001$
	Avg	0.383 ± 0.001	0.407 ± 0.000	$\mathbf{0.379} \pm 0.001$	$\mathbf{0.401} \pm 0.000$
ETTm1	96	0.342 ± 0.001	0.376 ± 0.000	$\mathbf{0.325} \pm 0.001$	$\mathbf{0.353} \pm 0.001$
	192	0.382 ± 0.002	0.395 ± 0.001	$\mathbf{0.377} \pm 0.001$	$\mathbf{0.380} \pm 0.001$
	336	0.418 ± 0.001	0.418 ± 0.001	$\mathbf{0.413} \pm 0.001$	$\mathbf{0.405} \pm 0.001$
	720	0.489 ± 0.004	0.457 ± 0.002	$\mathbf{0.479} \pm 0.001$	$\mathbf{0.444} \pm 0.001$
	Avg	0.408 ± 0.001	0.412 ± 0.001	$\mathbf{0.399} \pm 0.001$	$\mathbf{0.396} \pm 0.000$
ETTm2	96	0.185 ± 0.001	0.270 ± 0.002	$\mathbf{0.177} \pm 0.001$	$\mathbf{0.255} \pm 0.001$
	192	0.253 ± 0.001	0.314 ± 0.001	$\mathbf{0.244} \pm 0.001$	$\mathbf{0.300} \pm 0.000$
	336	0.315 ± 0.001	0.352 ± 0.001	$\mathbf{0.307} \pm 0.001$	$\mathbf{0.340} \pm 0.000$
	720	0.413 ± 0.001	0.407 ± 0.001	$\mathbf{0.408} \pm 0.002$	$\mathbf{0.400} \pm 0.001$
	Avg	0.292 ± 0.001	0.336 ± 0.001	$\mathbf{0.284} \pm 0.001$	$\mathbf{0.324} \pm 0.000$
Electricity	96	$\mathbf{0.148} \pm 0.000$	0.240 ± 0.000	0.149 ± 0.000	$\mathbf{0.235} \pm 0.000$
	192	$\mathbf{0.165} \pm 0.001$	0.256 ± 0.001	0.166 ± 0.001	$\mathbf{0.251} \pm 0.001$
	336	$\mathbf{0.179} \pm 0.001$	0.272 ± 0.001	0.180 ± 0.000	$\mathbf{0.266} \pm 0.000$
	720	0.213 ± 0.008	0.301 ± 0.006	$\mathbf{0.208} \pm 0.001$	$\mathbf{0.292} \pm 0.002$
	Avg	$\mathbf{0.176} \pm 0.002$	0.267 ± 0.001	$\mathbf{0.176} \pm 0.000$	$\mathbf{0.261} \pm 0.001$
Weather	96	0.175 ± 0.001	0.215 ± 0.001	$\mathbf{0.172} \pm 0.001$	$\mathbf{0.207} \pm 0.001$
	192	0.225 ± 0.002	0.258 ± 0.001	$\mathbf{0.220} \pm 0.000$	$\mathbf{0.252} \pm 0.001$
	336	$\mathbf{0.281} \pm 0.001$	0.299 ± 0.001	$\mathbf{0.281} \pm 0.002$	$\mathbf{0.296} \pm 0.001$
	720	0.359 ± 0.002	0.351 ± 0.001	$\mathbf{0.357} \pm 0.002$	$\mathbf{0.347} \pm 0.001$
	Avg	0.260 ± 0.001	0.281 ± 0.001	$\mathbf{0.258} \pm 0.001$	$\mathbf{0.275} \pm 0.000$
ILI	24	3.068 ± 0.115	1.183 ± 0.030	$\mathbf{2.993} \pm 0.097$	$\mathbf{1.150} \pm 0.025$
	36	2.901 ± 0.095	1.148 ± 0.027	$\mathbf{2.760} \pm 0.078$	$\mathbf{1.104} \pm 0.022$
	48	2.863 ± 0.077	1.143 ± 0.021	$\mathbf{2.740} \pm 0.066$	$\mathbf{1.102} \pm 0.018$
	60	2.958 ± 0.089	1.170 ± 0.023	$\mathbf{2.830} \pm 0.082$	$\mathbf{1.130} \pm 0.021$
	Avg	2.948 ± 0.086	1.161 ± 0.023	$\mathbf{2.831} \pm 0.075$	$\mathbf{1.122} \pm 0.020$
Solar	96	0.201 ± 0.004	0.236 ± 0.002	$\mathbf{0.196} \pm 0.003$	$\mathbf{0.218} \pm 0.003$
	192	0.236 ± 0.001	0.262 ± 0.001	$\mathbf{0.234} \pm 0.003$	$\mathbf{0.244} \pm 0.004$
	336	$\mathbf{0.250} \pm 0.002$	0.275 ± 0.001	0.253 ± 0.003	$\mathbf{0.259} \pm 0.002$
	720	$\mathbf{0.250} \pm 0.000$	0.276 ± 0.001	0.255 ± 0.000	$\mathbf{0.256} \pm 0.001$
	Avg	0.235 ± 0.001	0.262 ± 0.001	$\mathbf{0.234} \pm 0.001$	$\mathbf{0.245} \pm 0.001$

Table 7: Five-seed forecasting results for DLinear (mean ± standard deviation).

Dataset	Len	Original		BKLoss	
		MSE	MAE	MSE	MAE
ETTh1	96	0.396 ± 0.001	0.411 ± 0.001	$\mathbf{0.390} \pm 0.000$	$\mathbf{0.403} \pm 0.001$
	192	0.446 ± 0.001	0.441 ± 0.001	$\mathbf{0.439} \pm 0.001$	$\mathbf{0.432} \pm 0.001$
	336	0.489 ± 0.004	0.467 ± 0.004	$\mathbf{0.480} \pm 0.001$	$\mathbf{0.457} \pm 0.001$
	720	0.513 ± 0.003	0.511 ± 0.002	$\mathbf{0.498} \pm 0.002$	$\mathbf{0.498} \pm 0.002$
	Avg	0.461 ± 0.001	0.457 ± 0.001	$\mathbf{0.452} \pm 0.000$	$\mathbf{0.448} \pm 0.000$
ETTh2	96	0.347 ± 0.005	0.400 ± 0.004	$\mathbf{0.306} \pm 0.001$	$\mathbf{0.359} \pm 0.001$
	192	0.476 ± 0.008	0.476 ± 0.005	$\mathbf{0.401} \pm 0.003$	$\mathbf{0.421} \pm 0.002$
	336	0.587 ± 0.011	0.539 ± 0.005	$\mathbf{0.472} \pm 0.005$	$\mathbf{0.470} \pm 0.003$
	720	0.831 ± 0.011	0.658 ± 0.005	$\mathbf{0.678} \pm 0.015$	$\mathbf{0.583} \pm 0.007$
	Avg	0.561 ± 0.004	0.518 ± 0.002	$\mathbf{0.464} \pm 0.005$	$\mathbf{0.458} \pm 0.003$
ETTm1	96	0.345 ± 0.000	0.372 ± 0.001	$\mathbf{0.332} \pm 0.000$	$\mathbf{0.354} \pm 0.000$
	192	0.382 ± 0.000	0.391 ± 0.001	$\mathbf{0.375} \pm 0.000$	$\mathbf{0.377} \pm 0.001$
	336	0.414 ± 0.001	0.414 ± 0.001	$\mathbf{0.407} \pm 0.002$	$\mathbf{0.401} \pm 0.002$
	720	0.473 ± 0.001	0.450 ± 0.001	$\mathbf{0.467} \pm 0.000$	$\mathbf{0.437} \pm 0.001$
	Avg	0.403 ± 0.000	0.407 ± 0.001	$\mathbf{0.395} \pm 0.000$	$\mathbf{0.392} \pm 0.001$
ETTm2	96	0.193 ± 0.002	0.292 ± 0.002	$\mathbf{0.182} \pm 0.000$	$\mathbf{0.265} \pm 0.001$
	192	0.282 ± 0.004	0.359 ± 0.004	$\mathbf{0.246} \pm 0.001$	$\mathbf{0.312} \pm 0.002$
	336	0.378 ± 0.006	0.424 ± 0.004	$\mathbf{0.309} \pm 0.002$	$\mathbf{0.357} \pm 0.003$
	720	0.534 ± 0.015	0.512 ± 0.008	$\mathbf{0.417} \pm 0.003$	$\mathbf{0.428} \pm 0.003$
	Avg	0.347 ± 0.005	0.397 ± 0.003	$\mathbf{0.289} \pm 0.001$	$\mathbf{0.340} \pm 0.001$
Electricity	96	0.210 ± 0.000	0.302 ± 0.000	$\mathbf{0.205} \pm 0.000$	$\mathbf{0.288} \pm 0.000$
	192	0.210 ± 0.000	0.305 ± 0.000	$\mathbf{0.204} \pm 0.000$	$\mathbf{0.291} \pm 0.000$
	336	0.223 ± 0.000	0.319 ± 0.000	$\mathbf{0.216} \pm 0.000$	$\mathbf{0.306} \pm 0.000$
	720	0.258 ± 0.000	0.350 ± 0.000	$\mathbf{0.251} \pm 0.000$	$\mathbf{0.337} \pm 0.000$
	Avg	0.225 ± 0.000	0.319 ± 0.000	$\mathbf{0.219} \pm 0.000$	$\mathbf{0.305} \pm 0.000$
Weather	96	$\mathbf{0.196} \pm 0.000$	0.258 ± 0.002	0.200 ± 0.000	$\mathbf{0.240} \pm 0.002$
	192	$\mathbf{0.237} \pm 0.001$	0.296 ± 0.003	$\mathbf{0.237} \pm 0.001$	$\mathbf{0.279} \pm 0.002$
	336	0.282 ± 0.001	0.332 ± 0.001	$\mathbf{0.280} \pm 0.000$	$\mathbf{0.312} \pm 0.001$
	720	0.347 ± 0.002	0.383 ± 0.003	$\mathbf{0.341} \pm 0.000$	$\mathbf{0.363} \pm 0.001$
	Avg	0.266 ± 0.001	0.317 ± 0.001	$\mathbf{0.265} \pm 0.000$	$\mathbf{0.299} \pm 0.001$
ILI	24	2.826 ± 0.025	1.161 ± 0.006	$\mathbf{2.738} \pm 0.022$	$\mathbf{1.112} \pm 0.014$
	36	2.741 ± 0.069	1.132 ± 0.018	$\mathbf{2.495} \pm 0.056$	$\mathbf{1.063} \pm 0.021$
	48	2.752 ± 0.052	1.138 ± 0.015	$\mathbf{2.473} \pm 0.036$	$\mathbf{1.067} \pm 0.013$
	60	2.944 ± 0.020	1.186 ± 0.006	$\mathbf{2.599} \pm 0.052$	$\mathbf{1.108} \pm 0.013$
	Avg	2.816 ± 0.019	1.154 ± 0.004	$\mathbf{2.576} \pm 0.022$	$\mathbf{1.088} \pm 0.009$
Solar	96	0.290 ± 0.000	0.378 ± 0.000	$\mathbf{0.283} \pm 0.001$	$\mathbf{0.336} \pm 0.003$
	192	0.320 ± 0.000	0.398 ± 0.001	$\mathbf{0.313} \pm 0.001$	$\mathbf{0.356} \pm 0.001$
	336	$\mathbf{0.352} \pm 0.000$	0.422 ± 0.016	$\mathbf{0.352} \pm 0.001$	$\mathbf{0.375} \pm 0.002$
	720	$\mathbf{0.357} \pm 0.001$	0.413 ± 0.001	0.359 ± 0.001	$\mathbf{0.370} \pm 0.001$
	Avg	0.330 ± 0.000	0.403 ± 0.004	$\mathbf{0.327} \pm 0.000$	$\mathbf{0.359} \pm 0.001$

Table 8: Five-seed forecasting results for PatchTST (mean ± standard deviation).

Dataset	Len	Original		BKLoss	
		MSE	MAE	MSE	MAE
ETTh1	96	0.382 ± 0.003	0.400 ± 0.002	$\mathbf{0.380} \pm 0.003$	$\mathbf{0.391} \pm 0.002$
	192	$\mathbf{0.428} \pm 0.002$	0.432 ± 0.001	0.433 ± 0.007	$\mathbf{0.424} \pm 0.004$
	336	0.469 ± 0.001	0.457 ± 0.001	$\mathbf{0.468} \pm 0.004$	$\mathbf{0.445} \pm 0.003$
	720	0.518 ± 0.006	0.503 ± 0.004	$\mathbf{0.506} \pm 0.007$	$\mathbf{0.488} \pm 0.003$
	Avg	0.449 ± 0.002	0.448 ± 0.002	$\mathbf{0.447} \pm 0.004$	$\mathbf{0.437} \pm 0.002$
ETTh2	96	0.300 ± 0.003	0.350 ± 0.003	$\mathbf{0.293} \pm 0.003$	$\mathbf{0.336} \pm 0.003$
	192	0.375 ± 0.006	0.400 ± 0.002	$\mathbf{0.364} \pm 0.003$	$\mathbf{0.387} \pm 0.002$
	336	0.427 ± 0.007	0.439 ± 0.004	$\mathbf{0.416} \pm 0.008$	$\mathbf{0.425} \pm 0.003$
	720	0.442 ± 0.007	0.457 ± 0.002	$\mathbf{0.431} \pm 0.009$	$\mathbf{0.447} \pm 0.005$
	Avg	0.386 ± 0.004	0.411 ± 0.002	$\mathbf{0.376} \pm 0.004$	$\mathbf{0.398} \pm 0.002$
ETTm1	96	0.326 ± 0.002	0.365 ± 0.001	$\mathbf{0.313} \pm 0.001$	$\mathbf{0.345} \pm 0.001$
	192	0.369 ± 0.002	0.391 ± 0.001	$\mathbf{0.366} \pm 0.002$	$\mathbf{0.374} \pm 0.001$
	336	0.398 ± 0.001	0.408 ± 0.001	$\mathbf{0.396} \pm 0.001$	$\mathbf{0.396} \pm 0.001$
	720	$\mathbf{0.460} \pm 0.003$	0.445 ± 0.001	0.461 ± 0.002	$\mathbf{0.434} \pm 0.001$
	Avg	0.388 ± 0.001	0.402 ± 0.001	$\mathbf{0.384} \pm 0.001$	$\mathbf{0.387} \pm 0.001$
ETTm2	96	0.184 ± 0.002	0.267 ± 0.002	$\mathbf{0.178} \pm 0.002$	$\mathbf{0.258} \pm 0.002$
	192	0.249 ± 0.002	0.309 ± 0.002	$\mathbf{0.244} \pm 0.001$	$\mathbf{0.301} \pm 0.002$
	336	0.313 ± 0.004	0.351 ± 0.003	$\mathbf{0.304} \pm 0.001$	$\mathbf{0.340} \pm 0.001$
	720	0.416 ± 0.006	0.411 ± 0.004	$\mathbf{0.409} \pm 0.001$	$\mathbf{0.403} \pm 0.001$
	Avg	0.290 ± 0.002	0.334 ± 0.002	$\mathbf{0.284} \pm 0.000$	$\mathbf{0.326} \pm 0.000$
Electricity	96	$\mathbf{0.180} \pm 0.000$	0.272 ± 0.001	0.183 ± 0.000	$\mathbf{0.268} \pm 0.001$
	192	$\mathbf{0.187} \pm 0.000$	0.280 ± 0.000	0.189 ± 0.000	$\mathbf{0.275} \pm 0.000$
	336	$\mathbf{0.204} \pm 0.000$	0.296 ± 0.000	0.205 ± 0.000	$\mathbf{0.291} \pm 0.000$
	720	$\mathbf{0.246} \pm 0.001$	0.328 ± 0.001	$\mathbf{0.246} \pm 0.001$	$\mathbf{0.324} \pm 0.001$
	Avg	$\mathbf{0.204} \pm 0.000$	0.294 ± 0.000	0.206 ± 0.000	$\mathbf{0.289} \pm 0.000$
Weather	96	0.176 ± 0.002	0.217 ± 0.002	$\mathbf{0.175} \pm 0.001$	$\mathbf{0.211} \pm 0.001$
	192	0.222 ± 0.001	0.259 ± 0.002	$\mathbf{0.221} \pm 0.001$	$\mathbf{0.253} \pm 0.002$
	336	0.280 ± 0.001	0.298 ± 0.001	$\mathbf{0.277} \pm 0.001$	$\mathbf{0.293} \pm 0.001$
	720	0.356 ± 0.002	0.348 ± 0.000	$\mathbf{0.354} \pm 0.000$	$\mathbf{0.343} \pm 0.000$
	Avg	0.259 ± 0.001	0.280 ± 0.001	$\mathbf{0.257} \pm 0.000$	$\mathbf{0.275} \pm 0.000$
ILI	24	$\mathbf{2.670} \pm 0.028$	$\mathbf{1.084} \pm 0.007$	2.720 ± 0.067	1.088 ± 0.023
	36	2.722 ± 0.029	1.085 ± 0.009	$\mathbf{2.650} \pm 0.032$	$\mathbf{1.055} \pm 0.007$
	48	2.749 ± 0.040	1.098 ± 0.020	$\mathbf{2.694} \pm 0.066$	$\mathbf{1.081} \pm 0.027$
	60	2.719 ± 0.043	1.089 ± 0.010	$\mathbf{2.651} \pm 0.046$	$\mathbf{1.066} \pm 0.010$
	Avg	2.715 ± 0.018	1.089 ± 0.004	$\mathbf{2.679} \pm 0.035$	$\mathbf{1.072} \pm 0.012$
Solar	96	0.225 ± 0.003	0.281 ± 0.004	$\mathbf{0.221} \pm 0.003$	$\mathbf{0.269} \pm 0.003$
	192	0.257 ± 0.006	0.299 ± 0.003	$\mathbf{0.256} \pm 0.007$	$\mathbf{0.289} \pm 0.003$
	336	0.275 ± 0.005	0.309 ± 0.003	$\mathbf{0.273} \pm 0.001$	$\mathbf{0.298} \pm 0.002$
	720	0.277 ± 0.007	0.309 ± 0.004	$\mathbf{0.272} \pm 0.002$	$\mathbf{0.295} \pm 0.001$
	Avg	0.259 ± 0.003	0.299 ± 0.002	$\mathbf{0.255} \pm 0.002$	$\mathbf{0.288} \pm 0.001$

Table 9: Five-seed forecasting results for FITS (mean ± standard deviation).

Dataset	Len	Original		BKLoss	
		MSE	MAE	MSE	MAE
ETTh1	96	0.489 ± 0.024	0.462 ± 0.012	$\mathbf{0.430} \pm 0.015$	$\mathbf{0.428} \pm 0.009$
	192	0.545 ± 0.030	0.494 ± 0.014	$\mathbf{0.466} \pm 0.014$	$\mathbf{0.449} \pm 0.008$
	336	0.612 ± 0.026	0.530 ± 0.013	$\mathbf{0.513} \pm 0.025$	$\mathbf{0.477} \pm 0.016$
	720	0.688 ± 0.021	0.582 ± 0.009	$\mathbf{0.509} \pm 0.021$	$\mathbf{0.497} \pm 0.012$
	Avg	0.583 ± 0.007	0.517 ± 0.004	$\mathbf{0.479} \pm 0.008$	$\mathbf{0.463} \pm 0.005$
ETTh2	96	0.314 ± 0.004	0.360 ± 0.003	$\mathbf{0.305} \pm 0.003$	$\mathbf{0.351} \pm 0.003$
	192	0.404 ± 0.006	0.412 ± 0.004	$\mathbf{0.390} \pm 0.004$	$\mathbf{0.401} \pm 0.002$
	336	0.450 ± 0.012	0.450 ± 0.008	$\mathbf{0.432} \pm 0.007$	$\mathbf{0.436} \pm 0.005$
	720	0.478 ± 0.021	0.476 ± 0.012	$\mathbf{0.434} \pm 0.009$	$\mathbf{0.448} \pm 0.006$
	Avg	0.412 ± 0.010	0.425 ± 0.007	$\mathbf{0.390} \pm 0.005$	$\mathbf{0.409} \pm 0.003$
ETTm1	96	0.355 ± 0.000	0.376 ± 0.000	$\mathbf{0.339} \pm 0.000$	$\mathbf{0.358} \pm 0.000$
	192	0.393 ± 0.001	0.394 ± 0.001	$\mathbf{0.384} \pm 0.000$	$\mathbf{0.380} \pm 0.000$
	336	0.426 ± 0.001	0.416 ± 0.001	$\mathbf{0.416} \pm 0.000$	$\mathbf{0.402} \pm 0.000$
	720	0.487 ± 0.000	0.449 ± 0.001	$\mathbf{0.481} \pm 0.000$	$\mathbf{0.439} \pm 0.000$
	Avg	0.415 ± 0.000	0.409 ± 0.000	$\mathbf{0.405} \pm 0.000$	$\mathbf{0.395} \pm 0.000$
ETTm2	96	$\mathbf{0.183} \pm 0.000$	0.267 ± 0.000	$\mathbf{0.183} \pm 0.000$	$\mathbf{0.260} \pm 0.000$
	192	0.248 ± 0.000	0.305 ± 0.000	$\mathbf{0.247} \pm 0.000$	$\mathbf{0.300} \pm 0.000$
	336	0.308 ± 0.000	0.342 ± 0.000	$\mathbf{0.307} \pm 0.000$	$\mathbf{0.338} \pm 0.000$
	720	$\mathbf{0.408} \pm 0.000$	0.398 ± 0.000	$\mathbf{0.408} \pm 0.000$	$\mathbf{0.395} \pm 0.000$
	Avg	0.287 ± 0.000	0.328 ± 0.000	$\mathbf{0.286} \pm 0.000$	$\mathbf{0.323} \pm 0.000$
Electricity	96	0.214 ± 0.010	0.305 ± 0.016	$\mathbf{0.202} \pm 0.001$	$\mathbf{0.275} \pm 0.002$
	192	0.215 ± 0.008	0.309 ± 0.012	$\mathbf{0.200} \pm 0.000$	$\mathbf{0.277} \pm 0.001$
	336	0.238 ± 0.016	0.335 ± 0.021	$\mathbf{0.214} \pm 0.001$	$\mathbf{0.292} \pm 0.002$
	720	0.311 ± 0.030	0.400 ± 0.031	$\mathbf{0.257} \pm 0.002$	$\mathbf{0.329} \pm 0.005$
	Avg	0.245 ± 0.014	0.337 ± 0.017	$\mathbf{0.218} \pm 0.001$	$\mathbf{0.293} \pm 0.002$
Weather	96	$\mathbf{0.170} \pm 0.001$	0.218 ± 0.001	0.183 ± 0.002	$\mathbf{0.216} \pm 0.001$
	192	$\mathbf{0.218} \pm 0.001$	0.260 ± 0.001	0.226 ± 0.001	$\mathbf{0.258} \pm 0.001$
	336	$\mathbf{0.275} \pm 0.001$	0.300 ± 0.001	0.279 ± 0.001	$\mathbf{0.298} \pm 0.001$
	720	$\mathbf{0.353} \pm 0.001$	0.349 ± 0.001	0.356 ± 0.001	$\mathbf{0.348} \pm 0.001$
	Avg	$\mathbf{0.254} \pm 0.001$	0.282 ± 0.001	0.261 ± 0.000	$\mathbf{0.280} \pm 0.001$
ILI	24	5.206 ± 0.578	1.663 ± 0.104	$\mathbf{4.797} \pm 0.515$	$\mathbf{1.587} \pm 0.100$
	36	5.424 ± 0.672	1.667 ± 0.113	$\mathbf{4.895} \pm 0.714$	$\mathbf{1.569} \pm 0.127$
	48	5.555 ± 0.898	1.702 ± 0.157	$\mathbf{5.016} \pm 1.009$	$\mathbf{1.601} \pm 0.182$
	60	5.854 ± 0.341	1.762 ± 0.056	$\mathbf{5.278} \pm 0.401$	$\mathbf{1.659} \pm 0.068$
	Avg	5.510 ± 0.434	1.698 ± 0.074	$\mathbf{4.997} \pm 0.523$	$\mathbf{1.604} \pm 0.094$
Solar	96	0.343 ± 0.004	0.370 ± 0.001	$\mathbf{0.336} \pm 0.004$	$\mathbf{0.354} \pm 0.003$
	192	0.384 ± 0.003	0.396 ± 0.002	$\mathbf{0.380} \pm 0.002$	$\mathbf{0.385} \pm 0.001$
	336	0.424 ± 0.001	0.409 ± 0.003	$\mathbf{0.423} \pm 0.001$	$\mathbf{0.399} \pm 0.001$
	720	$\mathbf{0.422} \pm 0.001$	0.399 ± 0.002	0.423 ± 0.001	$\mathbf{0.387} \pm 0.001$
	Avg	0.393 ± 0.002	0.394 ± 0.001	$\mathbf{0.390} \pm 0.002$	$\mathbf{0.381} \pm 0.001$

Table 10: Residual distribution statistics of **DLinear** on the full test set at prediction length 96. **Bold** indicates the better value in each BKLoss-MSE pair. For *Mean* and *Median*, better means closer to zero; for the other metrics, better means smaller.

Dataset	**Mean**		**Variance**		**Median**		**RMSE**		$\|r\|$**-Q90**		$\|r\|$**-Q95**	
	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE
ETTh1	**0.0094**	0.0100	**0.3950**	0.3961	**-0.0072**	-0.0075	**0.6286**	0.6294	**0.8943**	0.8970	**1.3316**	1.3337
ETTh2	**-0.1400**	-0.1819	**0.2984**	0.3084	**-0.0879**	-0.1150	**0.5639**	0.5843	**0.8889**	0.9135	**1.2172**	1.2473
ETTm1	**0.0048**	0.0056	**0.3442**	0.3459	-0.0034	**-0.0030**	**0.5867**	0.5882	**0.8068**	0.8107	**1.2116**	1.2170
ETTm2	**-0.0872**	-0.1012	**0.1808**	0.1832	**-0.0656**	-0.0748	**0.4341**	0.4398	**0.6705**	0.6795	**0.9365**	0.9411
ECL	0.0051	**0.0047**	**0.2100**	0.2103	**-0.0129**	-0.0139	**0.4583**	0.4587	**0.6703**	0.6715	**0.9400**	0.9407
Weather	**-0.0318**	-0.0444	0.1956	**0.1942**	**-0.0273**	-0.0429	0.4434	**0.4430**	**0.5852**	0.5914	0.8162	**0.8158**
ILI	0.5273	**0.5034**	**2.4610**	2.4802	0.3774	**0.3426**	1.6550	**1.6534**	**2.5791**	2.5934	**3.6395**	3.6525
Solar	**-0.1358**	-0.1412	0.2701	**0.2699**	**-0.1362**	-0.1430	**0.5371**	0.5384	**0.8810**	0.8829	1.1799	**1.1770**

Table 11: Residual distribution statistics of **iTransformer** on the full test set at prediction length 96. **Bold** indicates the better value in each BKLoss-MSE pair. For *Mean* and *Median*, better means closer to zero; for the other metrics, better means smaller.

Dataset	**Mean**		**Variance**		**Median**		**RMSE**		$\|r\|$**-Q90**		$\|r\|$**-Q95**	
	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE
ETTh1	0.0096	**0.0093**	**0.3942**	0.3949	0.0121	0.0121	**0.6279**	0.6285	**0.9014**	0.9035	**1.3382**	1.3401
ETTh2	**-0.0113**	-0.0150	**0.2989**	0.3002	0.0007	**0.0005**	**0.5468**	0.5481	**0.8901**	0.8912	**1.2254**	1.2264
ETTm1	**-0.0268**	-0.0281	**0.3388**	0.3405	**-0.0149**	-0.0151	**0.5827**	0.5842	**0.8432**	0.8486	**1.1906**	1.1966
ETTm2	-0.0049	**-0.0038**	**0.1820**	0.1838	-0.0003	-0.0003	**0.4266**	0.4287	**0.6510**	0.6566	**0.9425**	0.9467
ECL	**-0.0024**	-0.0037	**0.1479**	0.1487	**-0.0042**	-0.0065	**0.3845**	0.3857	**0.5428**	0.5455	**0.7649**	0.7681
Weather	0.0109	**0.0102**	**0.1709**	0.1753	-0.0001	**-0.0000**	**0.4135**	0.4188	**0.5260**	0.5405	**0.7653**	0.7764
ILI	0.3739	**0.3449**	**2.7143**	2.7740	0.2325	**0.1956**	**1.6894**	1.7009	**2.7556**	2.7812	**3.8523**	3.8717
Solar	**0.0239**	0.0251	**0.1984**	0.2024	**0.0166**	0.0182	**0.4461**	0.4506	**0.7253**	0.7304	**1.0613**	1.0696

B.2 Residual analysis

We provide additional residual distribution statistics for three representative backbone models, including DLinear, iTransformer, and PatchTST. The results are shown in Tables 10, 11, and 12. Overall, BKLoss achieves better residual statistics than MSE in most cases across different datasets and backbone architectures. In particular, the Mean and Median are often closer to zero, suggesting that BKLoss reduces the overall bias of the residual distribution. The smaller Variance and RMSE further indicate that BKLoss tends to produce more concentrated residuals with reduced overall error magnitude. Moreover, BKLoss obtains smaller $|r|$-Q90 and $|r|$-Q95 on most datasets, which suggests that the high-percentile absolute residuals are effectively suppressed. This observation is consistent with the design motivation of BKLoss: instead of uniformly emphasizing all residuals as MSE does, BKLoss softly down-weights large-error samples and prevents them from dominating the optimization process. Therefore, these additional results further demonstrate that BKLoss can improve the residual distribution not only by reducing overall dispersion, but also by alleviating the influence of tail errors.

B.3 Comparison with Other Loss Functions

We further compare BKLoss with several recent and representative loss-function-based methods on DLinear in the appendix. As shown in Table 13, BKLoss exhibits strong competitiveness. Specifically, BKLoss achieves the best performance in 8 out of 16 cases and ranks second best in 4 out of 16 cases. These results further indicate that the effectiveness of BKLoss is not limited to a specific

Table 12: Residual distribution statistics of **PatchTST** on the full test set at prediction length 96. **Bold** indicates the better value in each BKLoss-MSE pair. For *Mean* and *Median*, better means closer to zero; for the other metrics, better means smaller.

Dataset	**Mean**		**Variance**		**Median**		**RMSE**		$\|r\|$**-Q90**		$\|r\|$**-Q95**	
	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE	BKLoss	MSE
ETTh1	0.0088	**-0.0037**	**0.3781**	0.3793	0.0066	**-0.0065**	**0.6149**	0.6159	**0.8816**	0.8889	**1.3316**	1.3389
ETTh2	0.0437	**-0.0093**	0.2986	**0.2921**	0.0362	**0.0017**	0.5482	**0.5405**	0.8954	**0.8757**	1.2204	**1.2029**
ETTm1	**0.0186**	0.0196	**0.3254**	0.3264	**0.0205**	0.0223	**0.5708**	0.5716	**0.8011**	0.8074	**1.1479**	1.1552
ETTm2	-0.0196	**-0.0003**	0.1852	**0.1828**	-0.0022	**-0.0008**	0.4308	**0.4275**	**0.6573**	0.6602	0.9522	**0.9467**
ECL	**0.0028**	0.0029	0.1802	**0.1800**	-0.0188	**-0.0186**	0.4245	**0.4243**	**0.6083**	0.6101	**0.8612**	0.8619
Weather	0.0214	**0.0140**	**0.1730**	0.1771	-0.0002	-0.0002	**0.4165**	0.4211	**0.5465**	0.5583	**0.7814**	0.7936
ILI	**0.3495**	0.3513	**2.5169**	2.6451	**0.1395**	0.1886	**1.6245**	1.6639	**2.6788**	2.7354	**3.7183**	3.8359
Solar	-0.0427	**-0.0401**	**0.2269**	0.2270	-0.0580	**-0.0574**	0.4782	**0.4781**	**0.8074**	0.8095	1.1045	**1.1010**

Table 13: Averaged forecasting results of **DLinear** (Zeng et al., 2023) across four prediction lengths. **Bold** indicates the best performance, and underlined indicates the second best.

Dataset	**BKLoss**		**FreDF**		**DBLoss**		**AMRC**		**SL**		**MSE+MAE**	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.452**	**0.448**	0.460	0.455	0.454	0.450	0.461	0.457	0.459	0.456	0.455	0.451
ETTh2	0.464	0.458	**0.448**	**0.454**	0.459	0.456	0.569	0.522	0.512	0.488	0.514	0.488
ETTm1	**0.395**	**0.392**	0.404	0.403	0.400	0.399	0.405	0.409	0.403	0.406	0.398	0.399
ETTm2	**0.289**	**0.340**	0.292	0.345	0.297	0.348	0.359	0.405	0.307	0.361	0.319	0.370
ECL	0.219	**0.305**	**0.218**	**0.305**	0.227	0.317	0.225	0.319	0.225	0.318	0.223	0.311
Weather	0.265	**0.299**	0.263	0.306	**0.262**	**0.299**	0.265	0.318	0.266	0.317	**0.262**	0.302
ILI	2.576	1.088	**2.484**	**1.064**	2.580	1.092	2.873	1.172	2.614	1.109	2.696	1.119
Solar	0.327	0.359	0.347	0.405	0.326	0.364	0.330	0.402	**0.324**	0.377	0.332	**0.343**

model architecture; even when compared with several recent advanced optimization objectives, it still maintains stable and competitive performance across different models and datasets.

B.4 Comparison with Classical Robust Losses on Synthetic Data

We further compare BKLoss with four classical robust training losses—Huber, Tukey, Cauchy, and Student-t—and the MSE baseline on the two constructed time series datasets, Long-Period Trend and Short-Period Seasonal. For each dataset, we evaluate a clean training condition and six types of training corruption: Cauchy, drift, Gaussian, Laplace, multiplicative Gaussian, and salt-and-pepper. Tables 14 and 15 report test MSE and MAE averaged over four prediction lengths. Across the 14 dataset–condition combinations, BKLoss achieves the lowest test MSE in every case and the lowest test MAE in 13 cases. On Long-Period Trend with salt-and-pepper corruption, Tukey obtains a slightly lower MAE than BKLoss (0.309070 versus 0.309339). These results provide an additional comparison of the training objectives on controlled synthetic signals under the evaluated noise and loss settings.

Table 14: Comparison of training objectives on the Long-Period Trend synthetic dataset with iTransformer. Results are averaged over prediction lengths 96, 192, 336, and 720. For the noisy settings, noise is added during training; MSE and MAE are evaluated on the test set. Bold indicates the lowest value in each row.

Test MSE

Condition	MSE loss	BKLoss	Huber	Tukey	Cauchy	Student-t
Clean	0.042280	**0.038039**	0.042262	0.041000	0.041497	0.041794
Cauchy	0.125470	**0.122634**	0.203077	0.210815	0.201117	0.201797
Drift	0.042376	**0.038134**	0.042347	0.041100	0.041582	0.041874
Gaussian	0.056269	**0.051982**	0.154759	0.158030	0.154712	0.154711
Laplace	0.056211	**0.052082**	0.154400	0.156991	0.154458	0.154388
Multiplicative Gaussian	0.067066	**0.063664**	0.066998	0.066007	0.066475	0.066721
Salt-and-pepper	0.154592	**0.152262**	0.153424	0.152802	0.152951	0.153118

Test MAE

Condition	MSE loss	BKLoss	Huber	Tukey	Cauchy	Student-t
Clean	0.143415	**0.130114**	0.143339	0.137251	0.140534	0.141722
Cauchy	0.281140	**0.277798**	0.366222	0.373488	0.364270	0.364946
Drift	0.143594	**0.130322**	0.143502	0.137456	0.140698	0.141863
Gaussian	0.175822	**0.166921**	0.313305	0.317166	0.313261	0.313261
Laplace	0.175817	**0.167088**	0.313056	0.315466	0.313126	0.313045
Multiplicative Gaussian	0.195754	**0.188886**	0.195678	0.192181	0.194294	0.194935
Salt-and-pepper	0.312770	0.309339	0.311082	**0.309070**	0.310077	0.310445

Table 15: Comparison of training objectives on the Short-Period Seasonal synthetic dataset with iTransformer. Results are averaged over prediction lengths 96, 192, 336, and 720. For the noisy settings, noise is added during training; MSE and MAE are evaluated on the test set. Bold indicates the lowest value in each row.

Test MSE

Condition	MSE loss	BKLoss	Huber	Tukey	Cauchy	Student-t
Clean	0.000315	**0.000095**	0.000311	0.000205	0.000261	0.000281
Cauchy	0.009911	**0.009720**	0.123432	0.144707	0.115180	0.118579
Drift	0.000321	**0.000103**	0.000317	0.000212	0.000267	0.000287
Gaussian	0.000428	**0.000217**	0.024211	0.023997	0.024127	0.024159
Laplace	0.000423	**0.000212**	0.023892	0.024182	0.023949	0.023923
Multiplicative Gaussian	0.000612	**0.000449**	0.000604	0.000508	0.000559	0.000576
Salt-and-pepper	0.024589	**0.019151**	0.022055	0.020366	0.021110	0.021372

Test MAE

Condition	MSE loss	BKLoss	Huber	Tukey	Cauchy	Student-t
Clean	0.013689	**0.007691**	0.013615	0.011105	0.012478	0.012943
Cauchy	0.081796	**0.081042**	0.285604	0.310332	0.275636	0.279768
Drift	0.013873	**0.008067**	0.013798	0.011356	0.012689	0.013142
Gaussian	0.015837	**0.011400**	0.127584	0.126966	0.127354	0.127444
Laplace	0.015756	**0.011315**	0.126928	0.127615	0.127077	0.127015
Multiplicative Gaussian	0.019319	**0.016879**	0.019206	0.017793	0.018558	0.018810
Salt-and-pepper	0.128172	**0.112770**	0.121248	0.116224	0.118518	0.119295

C THEORETICAL ANALYSIS

We first make precise the relationship between BKLoss, MSE, and zero-error kernel density at a fixed bandwidth. We then derive its bounded per-residual score and prove Theorem 1. The latter compares conditional population optima under a specified nonzero-mean training contamination and clean test targets.

C.1 BKLOSS AS A QUANTITATIVE GENERALIZATION OF MSE AND DENSITY MAXIMIZATION

Proof of Proposition 1. Let $e_1, \dots, e_N$ be a fixed, finite set of residuals and write $m_{2k} = N^{-1} \sum_{i=1}^{N} e_i^{2k}$. With the Gaussian kernel used by BKLoss, a Taylor expansion as $\sigma \to \infty$ gives

$$\mathcal{L}_{\mathrm{BKLoss}} = 1 - \frac{1}{N} \sum_{i=1}^{N} \exp\left(-\frac{e_i^2}{2\sigma^2}\right) = \frac{m_2}{2\sigma^2} - \frac{m_4}{8\sigma^4} + O\left(\frac{m_6}{\sigma^6}\right).$$

Here $m_2 = \mathrm{MSE}$, so $2\sigma^2 \mathcal{L}_{\mathrm{BKLoss}} \to \mathrm{MSE}$ as $\sigma \to \infty$. This is a limit of the *rescaled* loss; the unscaled BKLoss itself tends to zero.

For the same fixed σ, the Gaussian kernel density estimate of the empirical residual distribution at zero is

$$\widehat{p}_\sigma(0) = \frac{1}{N\sqrt{2\pi}\sigma} \sum_{i=1}^{N} \exp\left(-\frac{e_i^2}{2\sigma^2}\right) = \frac{1 - \mathcal{L}_{\mathrm{BKLoss}}}{\sqrt{2\pi}\sigma}.$$

Because the normalizing factor is constant when σ is fixed, minimizing BKLoss is then equivalent to maximizing $\widehat{p}_\sigma(0)$. Thus, for a given bandwidth, BKLoss gives a smooth preference for residuals concentrated near zero.

C.2 INFLUENCE FUNCTIONS AND ROBUSTNESS TO EXTREME NOISE

Proof of Proposition 2. For the per-residual squared loss $\ell_{\mathrm{MSE}}(e) = e^2/2$, the score is $\psi_{\mathrm{MSE}}(e) = e$ and is unbounded. For $\ell_{\mathrm{BK}}(e) = 1 - \exp(-e^2/(2\sigma^2))$, holding σ fixed yields

$$\psi_{\mathrm{BK}}(e) = \frac{\partial \ell_{\mathrm{BK}}(e)}{\partial e} = \frac{e}{\sigma^2} \exp\left(-\frac{e^2}{2\sigma^2}\right).$$

Its absolute value is maximal at $|e| = \sigma$, with

$$\sup_{e \in \mathbb{R}} |\psi_{\mathrm{BK}}(e)| = \frac{1}{\sigma\sqrt{\mathrm{e}}}, \qquad \lim_{|e| \to \infty} \psi_{\mathrm{BK}}(e) = 0.$$

We next bound the prediction displacement under contamination directly from the population objective.

C.3 CLEAN-TEST MSE UNDER NONZERO-MEAN CONTAMINATION

Proof of Theorem 1. Fix x and let $t = g - f^*$ denote the prediction's displacement from the clean target. Because the training noise has finite mean $\mathbb{E}[\delta] = \epsilon\mu_{\mathcal{H}}$, the conditional population MSE minimizer is exactly

$$t_{\mathrm{MSE}} = g_{\mathrm{MSE}} - f^* = \epsilon\mu_{\mathcal{H}}.$$

The population Gaussian-kernel similarity at displacement t is

$$Q_\sigma(t) = (1-\epsilon)\mathbb{E}_{Z \sim \mathcal{N}(0,\nu^2)} \exp\left(-\frac{(Z-t)^2}{2\sigma^2}\right) + \epsilon \mathbb{E}_{U \sim \mathcal{H}} \exp\left(-\frac{(U-t)^2}{2\sigma^2}\right).$$

Write $v = \sigma^2 + \nu^2$ and $c = \sigma/\sqrt{v}$. Evaluating the clean Gaussian integral gives

$$Q_\sigma(t) = (1-\epsilon)c \exp\left(-\frac{t^2}{2v}\right) + \epsilon \mathbb{E}_{U \sim \mathcal{H}} \exp\left(-\frac{(U-t)^2}{2\sigma^2}\right).$$

Let $a = (1-\epsilon)c$. Since the kernel is positive, $Q_\sigma(0) > a$. The contaminated component of $Q_\sigma(t)$ is at most ϵ, irrespective of the magnitude or shape of $\mathcal{H}$. Consequently, for $|t| \geq B_{\epsilon,\sigma,\nu}$,

$$Q_\sigma(t) \leq a \exp\left(-\frac{B^2_{\epsilon,\sigma,\nu}}{2v}\right) + \epsilon = a < Q_\sigma(0),$$

where the equality uses the definition of $B_{\epsilon,\sigma,\nu}$ and the condition $a > \epsilon$. By dominated convergence, $Q_\sigma(t)$ is continuous and tends to zero as $|t| \to \infty$, so it attains a global maximum. Therefore every global maximizer of Q_σ, equivalently every conditional population BKLoss minimizer, obeys

$$|t_{\mathrm{BK},\sigma}| < B_{\epsilon,\sigma,\nu}.$$

This bound does not depend on the size of the contamination mean. Under the theorem's sufficient condition $\epsilon|\mu_{\mathcal{H}}| > B_{\epsilon,\sigma,\nu}$, we obtain

$$|t_{\mathrm{BK},\sigma}| < B_{\epsilon,\sigma,\nu} < |t_{\mathrm{MSE}}|.$$

Finally, let the clean test target be $Y_{\mathrm{te}} = f^* + \xi$, where $\mathbb{E}[\xi \mid x] = 0$. For any prediction g,

$$\mathcal{R}_{\mathrm{clean}}(g \mid x) = \mathbb{E}[(Y_{\mathrm{te}} - g)^2 \mid x] = \mathrm{Var}(\xi \mid x) + (g - f^*)^2.$$

Hence

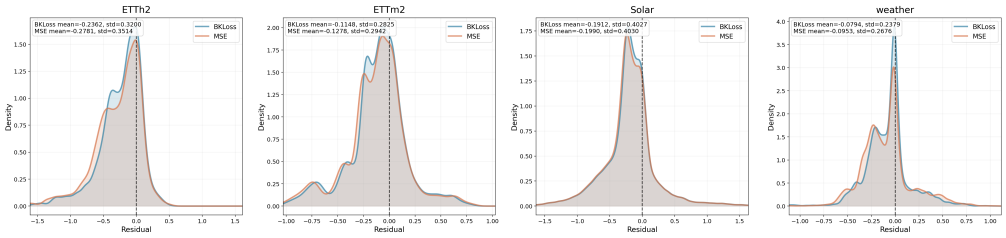
$$\mathcal{R}_{\mathrm{clean}}(g_{\mathrm{MSE}} \mid x) - \mathcal{R}_{\mathrm{clean}}(g_{\mathrm{BK},\sigma} \mid x) = \epsilon^2\mu^2_{\mathcal{H}} - t^2_{\mathrm{BK},\sigma} > 0.$$

This proves the stated improvement in conditional population clean-test MSE. If the theorem's conditions hold for almost every input x, integrating over x gives the corresponding overall population improvement.

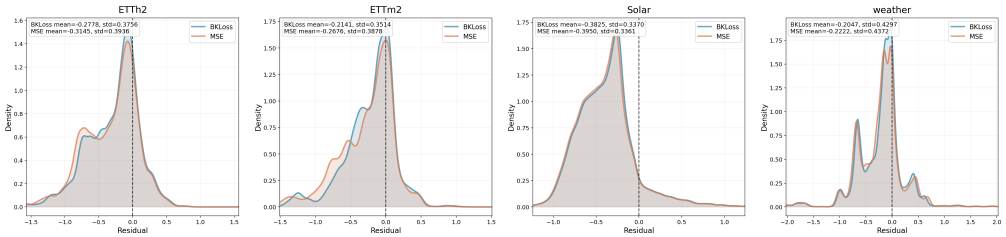
D VISUALIZATION

D.1 RESIDUAL DISTRIBUTION VISUALIZATION

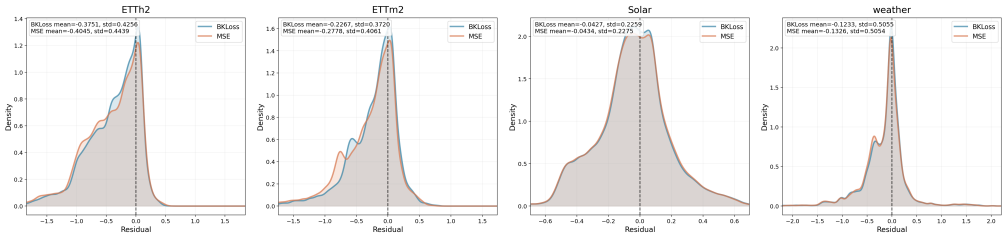
Figure 7 visualizes the residual distributions of DLinear trained with BKLoss and MSE under different prediction lengths. As shown in the figure, BKLoss generally produces more concentrated residual distributions than MSE. Specifically, the residual mean obtained by BKLoss is closer to zero in most cases, while the standard deviation is also smaller, indicating more stable forecasting errors.



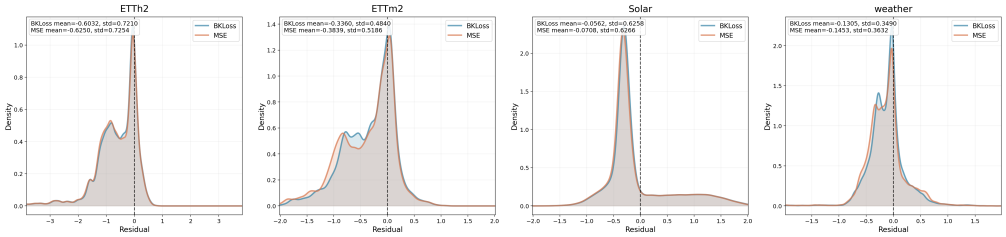
(a) Residual distribution at prediction length 96



(b) Residual distribution at prediction length 192



(c) Residual distribution at prediction length 336

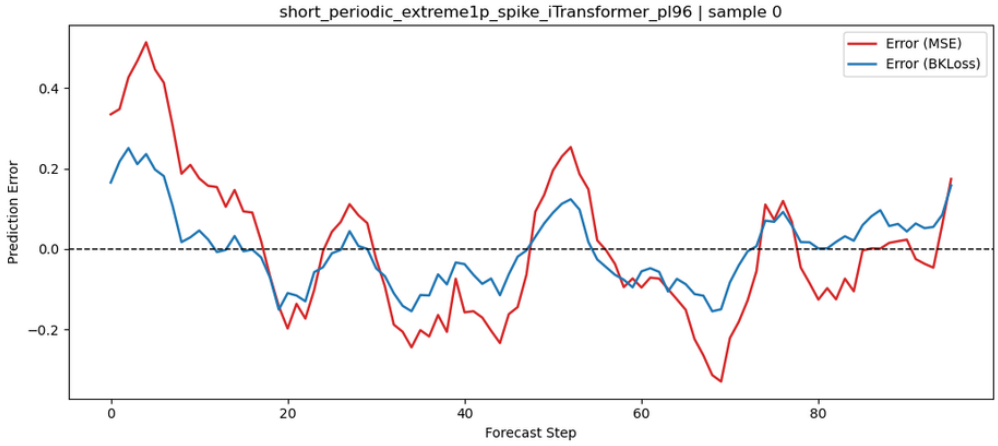


(d) Residual distribution at prediction length 720

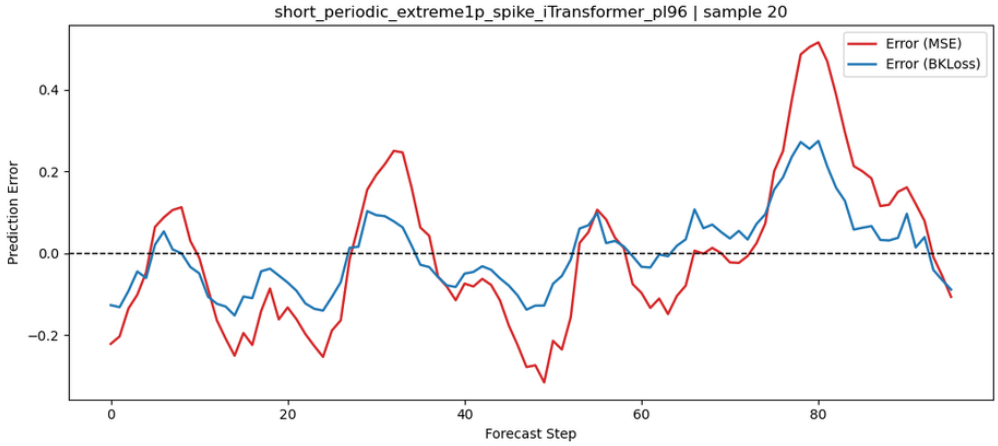
Figure 7: Residual distribution visualization comparing BKLoss and MSE as objective functions on DLinear across different prediction lengths.

D.2 Prediction Errors under Extreme Sparse Spikes

Figures 8–11 visualize signed prediction errors of iTransformer trained with MSE and BKLoss on a short-period synthetic sequence under an extreme sparse-spike setting. Each figure presents two selected test windows, numbered 0 and 20, for one prediction length. The dashed line denotes zero error. Across several peaks and troughs shown here, BKLoss produces smaller deviations from zero; the plots serve as qualitative examples at these test positions.

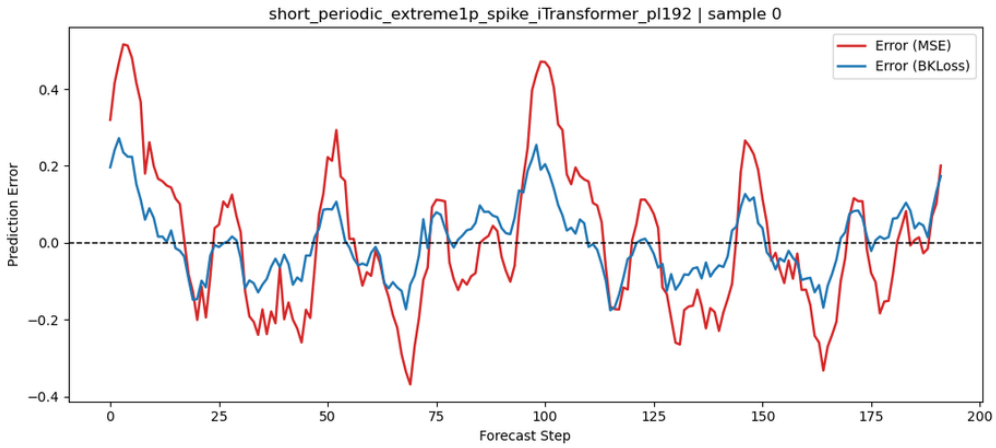


(a) Test-window sample 0

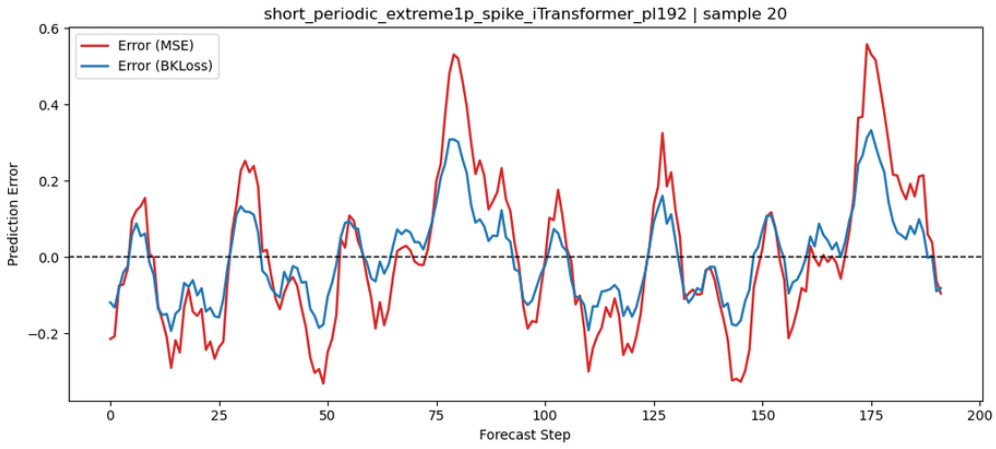


(b) Test-window sample 20

Figure 8: Signed prediction errors of iTransformer trained with MSE (red) and BKLoss (blue) on the Short-Period Seasonal synthetic dataset under extreme sparse spikes, at prediction length 96. Panels (a) and (b) show two selected test windows. The dashed line denotes zero error.

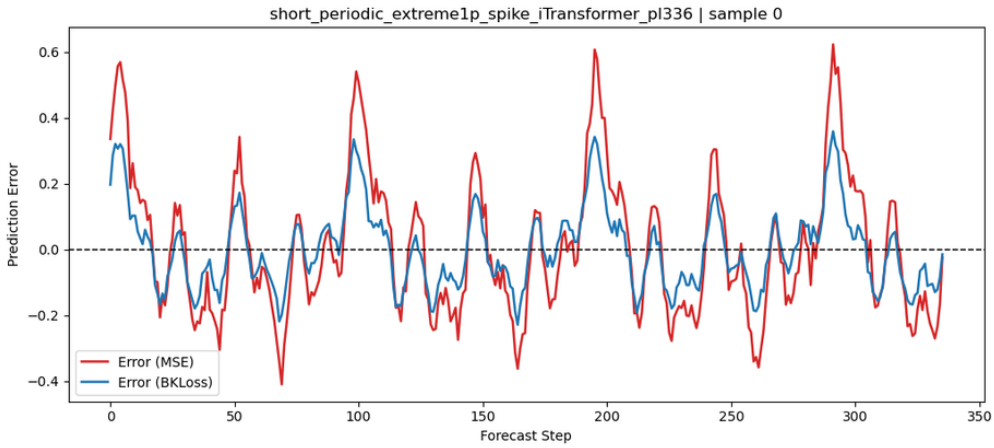


(a) Test-window sample 0

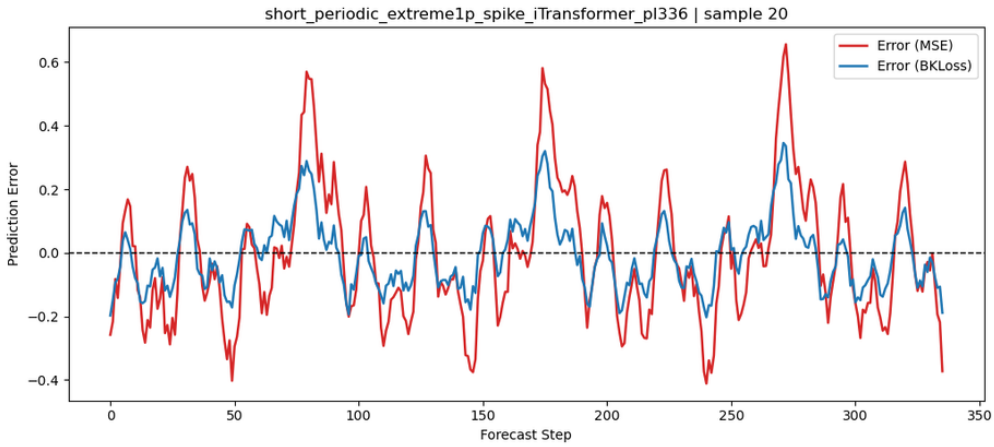


(b) Test-window sample 20

Figure 9: Signed prediction errors of iTransformer trained with MSE (red) and BKLoss (blue) on the Short-Period Seasonal synthetic dataset under extreme sparse spikes, at prediction length 192. Panels (a) and (b) show two selected test windows. The dashed line denotes zero error.

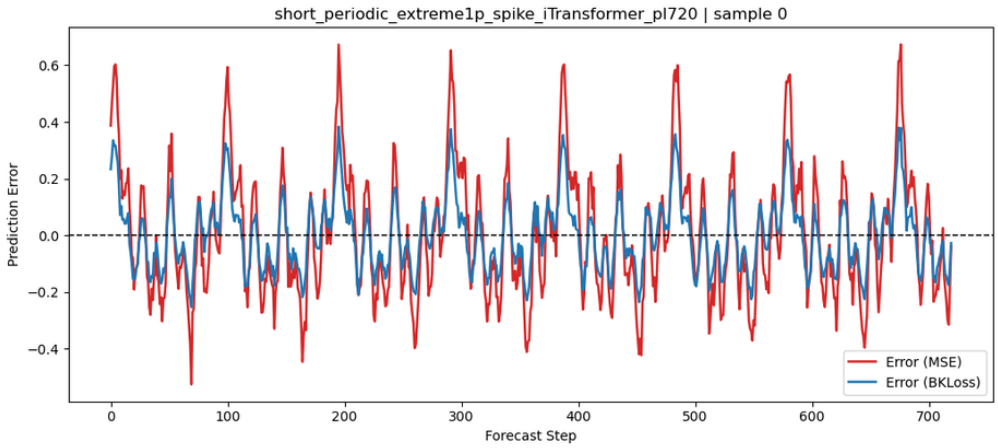


(a) Test-window sample 0

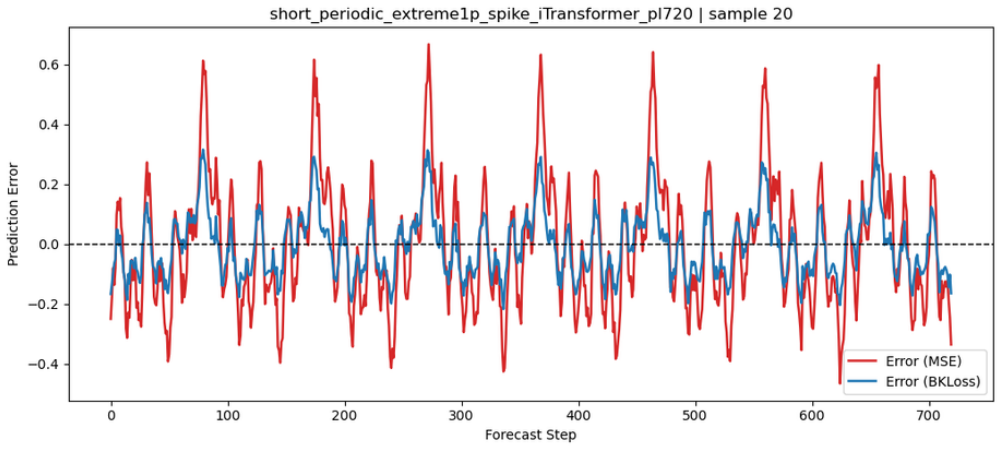


(b) Test-window sample 20

Figure 10: Signed prediction errors of iTransformer trained with MSE (red) and BKLoss (blue) on the Short-Period Seasonal synthetic dataset under extreme sparse spikes, at prediction length 336. Panels (a) and (b) show two selected test windows. The dashed line denotes zero error.



(a) Test-window sample 0



(b) Test-window sample 20

Figure 11: Signed prediction errors of iTransformer trained with MSE (red) and BKLoss (blue) on the Short-Period Seasonal synthetic dataset under extreme sparse spikes, at prediction length 720. Panels (a) and (b) show two selected test windows. The dashed line denotes zero error.