

BALANCING MSE AGAINST ABRUPT CHANGES FOR TIME-SERIES FORECASTING

Anonymous authors

Paper under double-blind review

ABSTRACT

Time-series forecasting models often encounter drastic changes in a given period of time (*i.e.*, abrupt changes) which generally occur due to unexpected or unknown events. Despite their scarce occurrences in the training set, abrupt changes incur loss (*i.e.*, MSE) that significantly contributes to the total loss. Therefore, they act as noisy training samples and prevent the model from learning generalizable sequence patterns, namely the normal states. Based on such an intuition, we propose a reweighting framework that down-weights the losses incurred by abrupt changes and up-weights those by normal states. For the reweighting framework, we first define a measurement termed *Local Discrepancy (LD)* which measures the degree of abruptness of a change in a given period of time. Then, we calculate how frequently the temporal changes appear in the training set based on LD (*i.e.*, estimated LD density). Since normal states generally appear frequently compared to the abrupt changes, they achieve higher LD density. Using such a property, we reweight the losses proportionally to the estimated LD density. Our reweighting framework is applicable to existing time-series forecasting models regardless of the architectures. Through extensive experiments on 12 time-series forecasting models over eight datasets with various in-output sequence lengths, we demonstrate that applying our reweighting framework reduces MSE by 10.1% on average and by up to 18.6% in the state-of-the-art model.

1 INTRODUCTION

As vast records are collected over time in diverse fields, the demand to predict the future based on the previous sequential data has led to efforts to solve the time-series forecasting problem in various applications such as energy (Ahmad et al., 2014), economics (Granger & Newbold, 2014), traffic (Vlahogianni et al., 2014), weather (Salman et al., 2015), environment pollution (De Vito et al., 2008) and mechanical system monitoring (Zhou et al., 2021). Previous studies focused on addressing the well-known challenges of time-series forecasting such as finding reliable dependencies from intricate and entangled temporal patterns (Wu et al., 2021; Oreshkin et al., 2019) or extending the forecasting time (*i.e.*, long-term forecasting) (Zhou et al., 2022; Liu et al., 2022; Wu et al., 2021; Zhou et al., 2021). For example, recent studies focused on improving the Transformer-based (Vaswani et al., 2017) models to address the long-term forecasting by taking the advantage of the long-term capacity of the self-attention mechanism and reducing quadratic computational costs (Li et al., 2019; Zhou et al., 2021; Wu et al., 2021; Liu et al., 2022).

Despite the remarkable improvements of the previous studies, even the state-of-the-art models take little account of the *abrupt changes* in time-series data. Abrupt change refers to the drastic change of target values (either increase or decrease) beyond the extent of the changes observed in the recent past. These abrupt changes are challenging, if not impossible to predict based solely on previous observations of the target variable, as they are generally caused by unexpected and external events (*e.g.*, natural disaster and war). Such changes break the auto-correlation structures, the periodic relationships between target variables, which are essential for a time-series forecaster to predict futures. One straightforward remedy is to laboriously collect external variables (*e.g.*, annotations of external events) and enforce a model to learn the relationship between the collected variables and the target variables (*i.e.*, cross-correlation). However, utilizing additional variables without thorough verification causes the model to learn a spurious correlation between variables, which

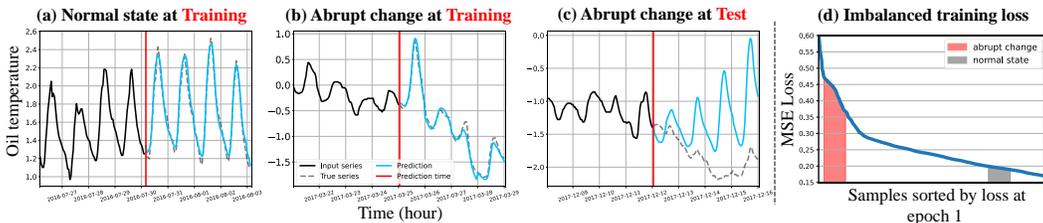


Figure 1: We observe that the state-of-the-art forecaster correctly predicts the target values during the training phase over both (a) normal states and (b) abrupt changes, respectively. However, (c) illustrates that the model fails to correctly predict the abrupt change during the test phase. (d) shows imbalanced loss when the training samples are sorted by MSE loss of each sample in the early training phase. Our important finding is that the training samples with abrupt change (b) occupy the large portion of total loss. On the other hand, training samples within the normal states (a) have a relatively small loss. This leads the model to focus less on the normal states during training.

worsens the generalization ability. Moreover, some abrupt changes have unknown causes (*e.g.*, sensor malfunction), which cannot be addressed by simply collecting external variables.

While forecasting abrupt changes is known to be challenging (Moniz et al., 2017; Hou et al., 2021), even worse, another significant issue of abrupt changes is that they limit the generalization performance of forecasting models during the test phase. Deep learning models are known to correctly predict all training samples regardless of the noisy labels by simply memorizing them (*i.e.*, overfitting) (Zhang et al., 2016). Our finding is that recent time-series forecasting models can easily memorize even abrupt changes in which the output sequence shows the different temporal characteristics (*e.g.*, mean, variance, and periodic structure) with the input sequence. To be more specific, Figure 1(a) and (b) show that the model correctly predicts the target values during the training phase in both normal states (*i.e.*, trend or periodicity of input sequence maintained in the output sequence) and abrupt changes, respectively. However, Figure 1(c) illustrates that the model fails to correctly predict the abrupt change during the test phase. The main reason is that the model is heavily overfitted to the abrupt changes since they take a significant portion of the total loss value compared to the ones in normal states (Figure 1(d)).

Based on such an intuition, we propose a simple yet effective reweighting framework that encourages the model to balance the imbalanced loss between abrupt changes and normal states as shown in Figure 2. Generally, time-series datasets do not provide explicit labels as to when the abrupt changes occur. Moreover, explicitly bisecting time stamps into abrupt changes and normal states is challenging since the definition of abrupt change may be vague depending on perspectives. Thus, we define a measurement termed *Local Discrepancy* (LD) which is used to determine how much a change in a given period of time is abrupt. By sliding a fixed-size window over the training time-series data, we compute the statistical difference between the in-output sequences as LD. Then, based on the observation that abrupt changes rarely appear in the training samples while normal states comprise the majority of the training set, we count the frequency of temporal changes based on LD. We divide the LD values into a predefined number of bins which are smoothed by kernel density estimation (*i.e.*, estimated LD density). By obtaining low LD density for the abrupt changes and high ones for the normal states, we *reweight* loss values proportional to the estimated **LD Density**, which we term our method as *ReLD*. This enables to emphasize the normal states which are the ones a model should learn for enhancing the forecasting capability.

In summary, the main contributions of our work are as follows:

- We reveal that the abrupt changes significantly degrade the time-series forecasting performance by taking most of the loss values.
- We propose a simple yet effective reweighting framework that adjusts the balance of MSE loss based on LD density, namely ReLD.
- Our reweighting framework consistently improves the performance of twelve existing time-series models on eight datasets, which reduces MSE by 10.1% on average and up to 18.6% when applied to the state-of-the-art model.
- ReLD also outperforms methods addressing the noisy samples such as smoothing, outlier filtering, and error-based baselines with a significant margin.

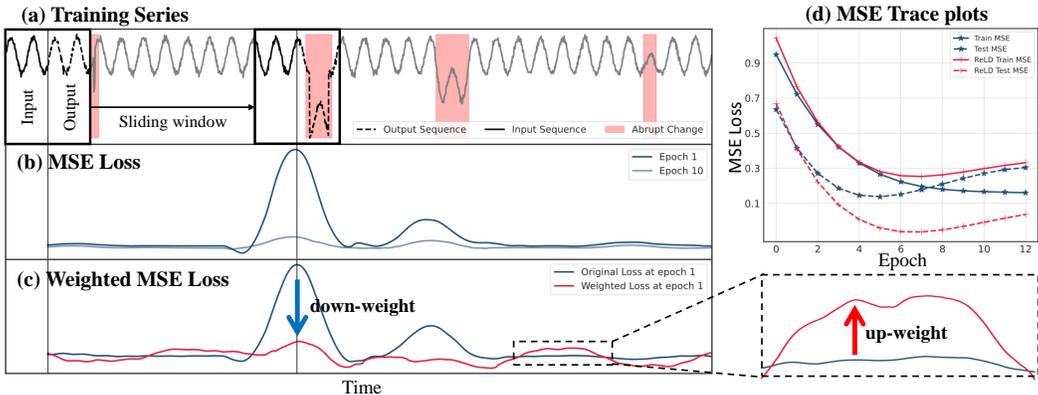


Figure 2: (a) We trained a model with a training series including four abrupt changes (red-shaded regions). (b) While the losses caused by the abrupt changes are considerably high in the early training phase, they are reduced significantly after several epochs of training. (d) After the losses by abrupt changes are decreased, however, we observe that the test losses rather increase, implying the degraded generalization capability. (c) We mitigate such an issue by proposing a reweighting framework that down-weights the losses of samples containing the abrupt changes (blue arrow) and up-weights normal samples (red arrow). (d) The model trained with our proposed reweighting framework achieves lower test MSE compared to that of the model without our framework.

2 RELATED WORK

2.1 DEEP LEARNING MODELS FOR TIME-SERIES FORECASTING

Deep learning-based models that have shown successful results in various domains have been actively applied to the time-series forecasting problem, which was originally dominated by classic statistical-based models (Anderson, 1976). Typically, previous time-series forecasting models leverage RNNs to learn the temporal dependencies inherent in time series (Bahdanau et al., 2014). In addition, models using CNNs or those combined with RNNs have appeared to focus on the local characteristics of time series (Bai et al., 2018). Recent studies focused on addressing two main challenges: 1) extending forecasting time (Zhou et al., 2022; Liu et al., 2022; Wu et al., 2021; Zhou et al., 2021) and 2) facilitating learning of entangled temporal dependencies (Wu et al., 2021; Oreshkin et al., 2019). As the demand for long-term planning and early warning in the real-world applications has increased, long-term forecasting has become essential. Thus, transformer-based forecasting models, which are known to effectively learn global temporal patterns, have emerged. These studies proposed sparse attention mechanisms to reduce the computational cost of the canonical attention mechanism when processing long sequences. Another trend is to apply the time-series decomposition techniques to a deep learning model in an end-to-end training manner. While it was originally used for the time-series analysis, a model trained in such an approach is effective to capture the temporal dependencies which are considered as compositions of multiple signals. The previous studies have demonstrated their effectiveness on various time-series datasets across multiple domains. However, they do not deal with how the locally appearing anomalous patterns (*i.e.*, abrupt changes) of time series affect the generalization capability of models.

2.2 ROBUSTNESS AGAINST NOISY SAMPLES AND DATA IMBALANCE

As aforementioned, deep learning models perfectly classify samples even with wrong annotations (*i.e.*, noisy samples) by simply memorizing them during the training phase (Zhang et al., 2016). Similarly, the abrupt changes in time-series forecasting are generally occurred by unexpected or unknown events, making them challenging to forecast correctly solely based on the previous time series. Due to this fact, perfectly forecasting them during the training phase indicates that the models simply memorized them which are in fact noisy samples in the time-series data. Addressing such noisy samples has been explored widely in image classification (Han et al., 2018; Li et al., 2020a; Song et al., 2019; Zhang & Sabuncu, 2018), especially utilizing the fact that data samples with small loss are likely to be annotated with clean labels (Arpit et al., 2017). For example, Generalized Cross-Entropy assigns less emphasis on the gradients of samples which show weak agreement be-

tween predicted labels and the ground truth labels since they are likely to be noisy samples (Zhang & Sabuncu, 2018). While reweighting approaches are effective to address noisy samples, such aforementioned studies are not originally designed to mitigate the data imbalance. In time-series forecasting, the number of abrupt changes is excessively scarce compared to that of normal states, necessitating to consider the data imbalance in addition to the noisy samples.

Addressing the data imbalance has been studied extensively in diverse fields such as image classification (Cao et al., 2019; Kang et al., 2020; Liu et al., 2019; Duggal et al., 2021), regression (Ren et al., 2022; Yang et al., 2021), object detection (Lin et al., 2017; Oksuz et al., 2020; Li et al., 2020b), and semantic segmentation (Jung et al., 2021; Choi et al., 2020). Their main intuition is to emphasize the training of the minor samples based on the frequency of each class. For example, Yang et al. (Yang et al., 2021) proposed the label distribution smoothing method that addresses the data imbalance in the image regression task. To tackle such data imbalance in time-series forecasting due to the scarce temporal patterns, few studies proposed an augmentation approach (Moniz et al., 2017) or modified model architectures (Hou et al., 2021). However, when addressing the data imbalance, they did not take account of models being overfitted to the scarce abrupt changes during the training phase. In this regard, we propose a reweighting framework that takes both issues into account: 1) abrupt changes work as noisy samples, and 2) they cause the data imbalance.

3 METHOD

3.1 PRELIMINARY

We first describe the forecasting task in a rolling window setting (Li et al., 2019; Zhou et al., 2021; Wu et al., 2021; Liu et al., 2022), which covers all possible in-output sequence pairs of the entire time series $\mathcal{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_T \mid \mathbf{s}_t \in \mathbb{R}^m\}$, where T is the length of observed series and m denotes the number of variables at time t . Univariate and multivariate time-series forecasting addresses time-series data with $m = 1$ and $m > 1$, respectively. By sliding a fixed-size window on \mathcal{S} , we obtain the windows $\mathcal{D} = \{(\mathcal{X}_t, \mathcal{Y}_t)\}_{t=1}^N$, which are divided into two parts: input sequence $\mathcal{X}_t = \{\mathbf{s}_{t-I}, \dots, \mathbf{s}_{t-1}\}$ with given length I and output sequence $\mathcal{Y}_t = \{\mathbf{s}_t, \dots, \mathbf{s}_{t+O-1}\}$ with length O to predict. A forecaster f predicts the most probable length- O sequence in the future given the past length- I sequence by learning temporal dependencies in \mathcal{S} . We mainly address the loss imbalance caused by the in-output sequence pairs which include a large discrepancy between adjacent \mathcal{X}_a and \mathcal{Y}_a compared to other \mathcal{X}_t and \mathcal{Y}_t pairs where a is the time stamp with an abrupt change. However, since most time-series datasets do not provide a label for the abrupt change, we propose a training framework in an unsupervised setting.

3.2 LOCAL DISCREPANCY

We propose the *Local Discrepancy* (LD) based on a statistical difference in order to measure how two adjacent in-output sequences, \mathcal{X}_t and \mathcal{Y}_t , are different from each other. We define LD as follows:

$$\text{LocalDis}(\mathcal{X}_t, \mathcal{Y}_t) = \frac{\bar{\mathcal{X}}_t - \bar{\mathcal{Y}}_t}{\sqrt{\frac{s_{\bar{\mathcal{X}}_t}^2}{I} + \frac{s_{\bar{\mathcal{Y}}_t}^2}{O} + \varepsilon}} := v_t, \quad (1)$$

where $\bar{\mathcal{X}}_t$ is the sample mean and $s_{\bar{\mathcal{X}}_t}$ is the sample standard deviation of \mathcal{X}_t . Statistical tests are generally used to determine whether means of two samples (*i.e.*, groups of data points in a sequence) are identical or not (Welch, 1938). In this regard, we leverage t-statistic¹, a scalar value, as normalized discrepancy to measure how much two adjacent groups of samples are distinct. Figure 3 describes how LD reflects the different types of local temporal changes in time-series data (e.g., (a) normal changes, (b) fluke point, (c) frequency change, and (d) trend shift). The LD values of normal states oscillate within a certain range (see (a) red line), but the LD values of abrupt changes is beyond the range of normal LD. The periodicity and boundedness of LD in normal periodic series are theoretically discussed in Appendix A.

As aforementioned, the definition of abrupt change may be vague depending on perspectives. Thus, rather than bisecting the time stamps into abrupt changes and normal states, we utilize LD values

¹Other statistics such as KPSS and t -squared can be used as LD. However, when we conduct a preliminary experiment, the t -statistic measures better than others. We further discuss the details in Section 5.

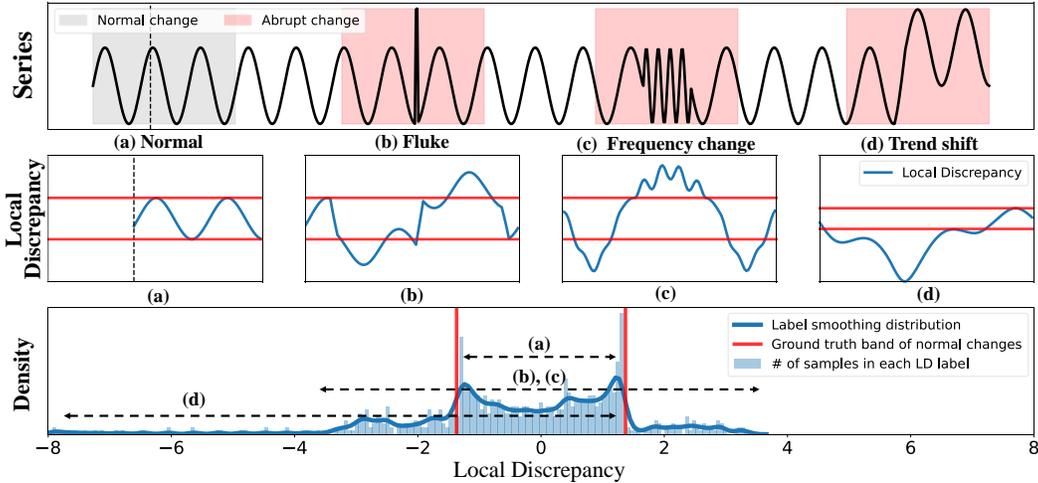


Figure 3: The four examples of temporal changes locally seen in time series data: (a) normal, (b) fluke, (c) frequency change, and (d) trend shift. Local discrepancy computed by the sliding window captures the three abrupt changes beyond the bounds (red line) seen in normal states. In the estimated LD density distribution, training samples with abrupt changes are visibly fewer than training samples with normal state and are sparsely distributed with large absolute local discrepancy.

as weights of reweighting framework to mitigate the impact of abrupt changes in training phase. In other words, losses of training samples which have large absolute v_t values will be down-weighted since we consider them to be close to the abrupt change. By computing LD over the training dataset \mathcal{D}_{train} and each of m dimensions, we obtain the dataset $\mathcal{D}_{train} = \{(\mathcal{X}_t, \mathcal{Y}_t, v_t)\}_{t=1}^N$ containing local discrepancy $v_t \in \mathbb{R}^m$ for prediction time t and for each of m dimensions. We then assign the weight $w_t = \frac{c}{|v_t|+1} \propto \frac{1}{|v_t|+1} \in \mathbb{R}^m$ to each training sample inversely to LD value of sample in \mathcal{D}_{train} with constant c as scaling factor. Finally, we calculate the reweighted MSE loss \mathcal{L}_w as follows:

$$\mathcal{L}_w(\mathcal{Y}_t, \hat{\mathcal{Y}}_t) = \frac{1}{m \cdot O} \cdot \sum_{j=1}^m w_t^j \sum_{i=0}^{O-1} \cdot (s_{t+i}^j - \hat{s}_{t+i}^j)^2 \quad (2)$$

where $\hat{\mathcal{Y}}_t$ is forecasting results of f conditioned on \mathcal{X}_t . Through this simple reweighting framework which assigns weight inversely to LD values, namely *invLD*, we down-weight the loss of abrupt changes (large absolute LD) and up-weight the loss of normal states (small absolute LD), following the observation that the original MSE loss in the presence of abrupt changes is much larger than the loss at the normal state. Reweighting MSE only based on LD, however, does not take into account the property that normal states frequently appear while the abrupt changes are rarely included in the time-series data. We further improve our reweighting framework by considering such frequency differences between abrupt changes and normal states.

3.3 DENSITY-BASED REWEIGHTING FOR TIME-SERIES FORECASTING

Time series often exhibit both short-term and long-term repeating patterns (Lai et al., 2018) by periodicity, and taking them into account is crucial for making accurate predictions. Let’s consider a time series which has large shifts in a short period, but repeated. We can assume such large shifts are part of the normal states considering their frequent occurrences. However, this temporal pattern is down-weighted because of their large LD values regardless of the number of occurrences. In other words, *invLD* based on the inverse of LD (*i.e.*, $w_t \propto \frac{1}{|v_t|}$) will not only down-weight the loss values of abrupt changes but also those of normal states, which the model should learn to properly forecast. Therefore, we improve the time-series forecasting by considering the frequency of temporal changes (*i.e.*, LD density) when reweighting loss values in time-series forecasting.

Inspired by deep imbalanced regression (Yang et al., 2021), we use the kernel density estimation to address the missing regions between continuous LD spaces. Through the estimated density of LD

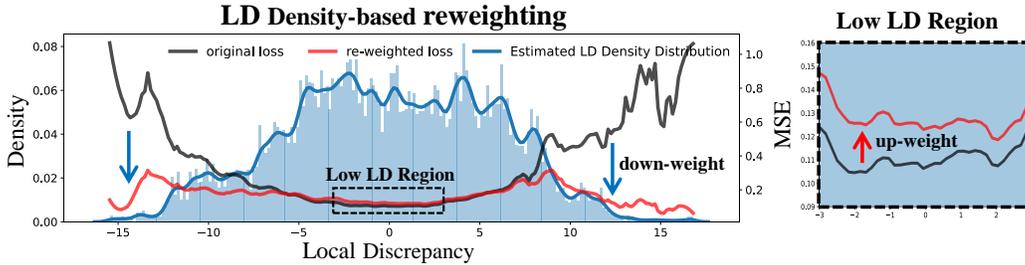


Figure 4: For the real-world dataset (ETTh1), we visualize the estimated LD density distribution and the averaged MSE loss of samples in each LD bin after training a forecaster for one epoch. Our density-based re-weighting framework effectively down-weights (blue arrow) the losses on abrupt changes (low density and large LD) and up-weights (red arrow) those on normal states (high density and small LD).

$\tilde{p}(v)$, we assign the weight $w_t = c \cdot \tilde{p}(v_t) \propto \tilde{p}(v_t)$ and use these weights to train a model as \mathcal{L}_w described in Equation 2. Figure 4 demonstrates that our final reweighting framework based on LD density, ReLD, mitigates the imbalanced loss problem in a real-world dataset. The procedure of our framework is described in Appendix D.

4 EXPERIMENTS

This section demonstrates that our proposed framework consistently improves existing time-series forecasting models regardless of the architectures. We apply our framework to 12 forecasting models and evaluate them on eight datasets which include four real-world forecasting applications: energy consumption, mechanical systems, weather, and pollution. Dataset analysis shows that our proposed framework brings larger performance gains as the number of abrupt changes in a given dataset increases. We also provide other experiments in the Appendix, which include results on synthetic series (Appendix B.2), computational cost of methods (Appendix B.4), qualitative results (Appendix E) and details for reproducibility (Appendix D).

4.1 EXPERIMENT SETTING

Dataset descriptions As mainstream benchmarks, ETT are widely used to evaluate long-term forecasting methods (Zhou et al., 2021; Wu et al., 2021; Liu et al., 2022; Zhou et al., 2022) ETT contains the crucial indicators (e.g., oil temperature, load, etc) collected from the electricity transformers over two years, and are categorized into four datasets depending on the location (ETT1 and ETT2) and interval (15 minutes and one hour). Electricity (ECL) dataset contains the hourly electricity consumption of 321 customers from 2012 to 2014. Weather dataset (Wea) is recorded every 10 minutes for a year, which contains 21 meteorological indicators (e.g., air temperature, humidity, etc). Pump dataset (Pum) is collected from 52 sensors monitoring the water pump. AirQuality dataset (De Vito et al., 2008), taken from the UCI repository, contains hourly averaged responses obtained from five metal oxide chemical sensors of an air quality chemical multi-sensor device.

Comparison methods To verify that our reweighting framework works consistently in existing forecasting models, we applied it to 12 forecasting models and reported the reduced forecasting errors by applying ReLD. The baselines are roughly categorized into three groups: Transformer-based (Zhou et al., 2022; Liu et al., 2022; Wu et al., 2021; Zhou et al., 2021; Kitaev et al., 2019; Li et al., 2019; Vaswani et al., 2017), CNN-based (Bai et al., 2018), and RNN-based (Bahdanau et al., 2014; Lai et al., 2018) models. We also include two univariate forecasting models: DeepAR (Salinas et al., 2020) and N-BEATS (Oreshkin et al., 2019). We applied our ReLD to the baselines without additional external data indicating abrupt changes or modifications of architectures.

Experiment details The forecasting performance is measured using the mean square error (MSE). Note that we follow the evaluation protocol proposed by previous work (Zhou et al., 2021; Wu et al., 2021), including normalization through the mean and standard deviation of the train dataset. We conducted experiments on five different lengths of time series. All reported values are averaged MSE from five independent experiments with different random initializations. Due to the space limit, the

Table 1: Multivariate results with different input length I and prediction lengths O . A lower MSE indicates a better prediction and the best results in each row are bolded. Imp. means averaged MSE reduction rate for a given model and dataset. Total denotes the averaged MSE reduction rate of a given dataset across all baselines models. The full results, which include other ETT datasets and confidence interval, are available in Appendix F.

Models	FEDformer		Pyraformer		Autoformer		Informer		Reformer		LSTNet		LSTMa		TCN		Total	
	I/O	base	ReLD	base	ReLD	base	ReLD	base	ReLD	base	ReLD	base	ReLD	base	ReLD			
ETTM1	96/96	0.359	0.357	0.536	0.471	0.524	0.455	0.640	0.543	0.777	0.641	0.548	0.536	0.705	0.592	0.676	0.594	-12.91%
	336/168	0.385	0.379	0.563	0.506	0.534	0.500	1.224	0.751	0.840	0.689	0.632	0.577	0.871	0.648	0.938	0.913	
	336/336	0.403	0.396	0.697	0.573	0.561	0.514	1.390	1.008	0.987	0.895	0.798	0.686	1.125	0.681	1.148	1.126	
	336/720	0.501	0.480	0.904	0.682	0.560	0.528	1.333	1.078	1.122	1.003	0.925	0.817	0.978	0.828	1.277	1.238	
Imp.		-1.99%		-16.17%		-8.42%		-25.11%		-13.88%		-9.16%		-24.14%		-4.95%		
ETTM2	96/96	0.189	0.184	0.371	0.248	0.293	0.221	0.445	0.286	0.743	0.449	0.443	0.343	0.381	0.280	0.554	0.384	-21.14%
	336/168	0.343	0.379	0.566	0.551	0.309	0.284	2.283	1.453	1.208	0.836	0.950	0.830	1.178	0.601	1.868	1.956	
	336/336	0.338	0.319	1.601	1.330	0.508	0.364	2.479	1.764	2.239	1.425	1.610	1.019	1.479	0.745	2.769	2.773	
	336/720	0.432	0.394	5.476	5.037	0.502	0.411	6.580	5.777	3.068	2.827	6.130	4.449	3.083	2.381	3.204	3.187	
Imp.		-8.83%		-15.15%		-19.85%		-28.29%		-28.65%		-24.85%		-36.96%		-6.57%		
Weather-h	48/48	0.338	0.336	0.292	0.279	0.344	0.343	0.345	0.294	0.343	0.313	0.318	0.310	0.346	0.325	0.348	0.327	-4.41%
	48/96	0.403	0.400	0.393	0.358	0.464	0.446	0.453	0.443	0.526	0.416	0.414	0.386	0.409	0.387	0.450	0.424	
	96/192	0.458	0.447	0.421	0.398	0.516	0.491	0.530	0.498	0.659	0.673	0.464	0.461	0.420	0.416	1.018	1.005	
	168/336	0.510	0.516	0.454	0.440	0.612	0.566	0.592	0.568	0.841	0.782	0.490	0.473	0.473	0.452	1.147	1.209	
Imp.		-0.65%		-5.55%		-4.21%		-6.79%		-8.66%		-3.41%		-4.17%		-1.88%		
AirQuality	96/96	0.825	0.817	1.121	1.112	0.992	0.986	1.353	1.193	1.210	1.196	1.146	1.141	1.145	1.081	1.026	0.992	-3.97%
	336/168	0.811	0.808	1.193	1.115	0.911	0.922	1.796	1.595	1.473	1.345	1.231	1.156	1.644	1.376	1.246	1.163	
	336/336	0.892	0.872	1.224	1.214	0.962	0.933	1.758	1.706	1.473	1.396	1.399	1.388	1.352	1.206	1.301	1.284	
	336/720	0.997	0.953	2.196	1.982	1.129	1.079	2.914	2.985	1.723	1.671	1.826	1.921	2.475	2.333	1.442	1.426	
Imp.		-2.01%		-4.47%		-1.72%		-5.87%		-4.51%		-0.52%		-9.63%		-3.09%		
Pump	96/96	0.520	0.513	0.848	0.796	0.558	0.538	0.831	0.870	0.826	0.760	1.016	1.007	0.813	0.766	1.037	0.970	-7.74%
	336/168	0.550	0.536	0.851	0.843	0.597	0.581	1.705	1.527	1.094	0.856	1.327	1.202	0.909	0.816	1.109	1.077	
	336/336	0.593	0.564	0.922	0.951	0.661	0.621	1.676	1.492	0.966	0.918	1.654	1.292	0.934	0.859	1.521	1.208	
	336/720	0.723	0.580	1.370	1.283	0.707	0.619	1.704	1.699	1.328	1.218	1.608	1.642	1.464	1.244	2.075	1.546	
Imp.		-7.10%		-2.61%		-6.17%		-4.28%		-10.73%		-7.49%		-9.77%		-13.85%		

Table 2: Univariate results with different prediction lengths. The experimental settings are identical to those of multivariate experiments.

Models	FEDformer		Pyraformer		Autoformer		N-BEATS		Informer		LogSparse		Transformer		DeepAR		Total	
	MSE	base	ReLD	base	ReLD	base	ReLD	base	ReLD	base	ReLD	base	ReLD	base	ReLD			
ETTM1	96/96	0.036	0.035	0.082	0.058	0.058	0.052	0.085	0.066	0.105	0.083	0.105	0.058	0.089	0.060	0.179	0.085	-15.11%
	336/168	0.083	0.066	0.129	0.122	0.088	0.072	0.077	0.070	0.181	0.158	0.192	0.161	0.221	0.196	0.200	0.199	
	336/336	0.081	0.076	0.194	0.184	0.090	0.085	0.134	0.125	0.263	0.227	0.367	0.304	0.274	0.247	0.382	0.432	
	336/720	0.109	0.102	0.337	0.275	0.143	0.112	0.144	0.112	0.391	0.349	0.474	0.381	0.479	0.443	0.498	0.418	
Imp.		-8.81%		-14.61%		-13.83%		-15.20%		-14.58%		-24.42%		-15.40%		-14.05%		
ECL	96/96	0.243	0.240	0.239	0.234	0.440	0.424	0.303	0.295	0.269	0.272	0.281	0.282	0.301	0.283	0.917	0.786	-3.87%
	336/168	0.304	0.299	0.255	0.253	0.486	0.482	0.328	0.311	0.709	0.705	0.330	0.335	0.404	0.373	1.018	1.024	
	336/336	0.355	0.345	0.306	0.296	0.463	0.439	0.425	0.386	0.801	0.743	0.399	0.316	0.443	0.424	1.202	1.280	
	336/720	0.469	0.448	0.336	0.321	0.606	0.616	0.544	0.497	0.824	0.781	0.372	0.339	0.442	0.413	1.302	1.331	
Imp.		-2.44%		-2.70%		-2.05%		-6.40%		-3.01%		-6.97%		-6.18%		-1.22%		

main paper only reports the averaged MSE while the Appendix F includes the full benchmarks along with the confidence intervals.

4.2 MAIN RESULTS

As shown in Table 1, applying our reweighting framework reduces the MSE consistently in all existing time-series forecasting models across different datasets and varying length-averaged settings. In addition, the lowest MSE in each setting was generally achieved by the models which applied ReLD. We also observe that the performance improvements vary depending on the datasets. For example, applying ReLD to the baselines achieves an average of 21.14% lower MSE compared to the average of original errors on ETTm2. On the other hand, applying ReLD achieves only 3.97% lower MSE on average with AirQuality dataset. We analyze such an issue in Section 4.4. As for the univariate setting, we list the univariate results of two typical datasets in Table 2. Again, similar to the results observed in multivariate datasets, applying ReLD enhances the forecasting performance consistently regardless of the model architectures compared to baselines without ReLD. Moreover, the lowest error for each setting was again achieved when ReLD was applied.

4.3 COMPARISONS WITH OTHER METHODS

Smoothing and outlier filtering methods Table 3 (a) compares our ReLD with two smoothing and outlier filtering methods. Moving average (MA) and exponential MA (EMA) are widely used smoothing techniques that remove noisiness and reduce values of outliers, allowing meaning-

Table 3: Comparison with other methods which can deal with abrupt changes. We conduct experiments using ETTm2 dataset on two recent state-of-the-art time-forecasting models.

Group	Models I → O	FEDformer			Autoformer			Imp.
		336 → 168	336 → 336	336 → 720	336 → 168	336 → 336	336 → 720	
	Vanilla	0.3429	0.3381	0.4322	0.3095	0.5084	0.5022	-
(a)	Vanilla + MA	0.3552	0.3432	0.4178	0.3132	0.4310	0.5421	-0.72%
	Vanilla + EMA	0.3637	0.3429	0.4039	0.3189	0.5161	0.5492	2.47%
	Vanilla + Outlier (Gaussian)	0.2922	0.3303	0.4054	0.3839	0.4529	0.4675	-2.84%
(b)	Vanilla + Focal-R	0.4028	0.3766	0.5037	0.3151	0.4453	0.5196	6.38%
	Vanilla + flip Focal-R	0.2837	0.3223	0.4049	0.3108	0.3710	0.4704	-10.20%
(c)	Vanilla + invLD	0.2817	0.3259	0.3962	0.2891	0.3431	0.4496	-13.22%
	Vanilla + ReLD	0.2790	0.3191	0.3939	0.2838	0.3642	0.4110	-14.65%

ful temporal patterns to stand out. Similarly, outlier filtering also mitigates the influence of outliers on learning the normal patterns. However, we observe that adopting such methods either shows insignificant performance improvement or rather aggravates the time-series forecasting performance. This result shows that addressing the abrupt change is a challenging task and the performance improvement obtained by our proposed method is not trivial. Details of the methods are described in Appendix C.1.

Error-aware loss We also compare our method with error-based reweighting approaches. Focal-R, the regression version of focal loss (Yang et al., 2021), allows a model to focus on samples with relatively large loss while down-weighting loss on samples with small errors. It works in a way that is contrary to our findings. We modified such an approach by putting negation on the input of Focal-R, termed as flip Focal-R (Details in Appendix C.2). Table 3 (b) shows that the performance of Focal-R is rather degraded while that of flip Focal-R improved. Such a result well demonstrates that our intuition, de-emphasizing the samples with high loss, is valid. Also, we observe that utilizing Focal-R fails to outperform our proposed method. We conjecture such superior performance of ReLD is mainly due to reflecting the temporal changes and periodicity.

Ablation study of our reweighting framework We conduct the ablation study of our proposed method by comparing our full framework ReLD and an approach which considers the LD values only (invLD). Table 3 (c) shows that our full framework is superior to invLD. Additionally, we observe that both approaches outperform the methods in (a) and (b).

4.4 DATASET ANALYSIS

Preserving the robustness on the abrupt changes Our reweighting framework improves time-series forecasting performance by mitigating the loss imbalance mainly occurred due to abrupt changes. Since we impose less emphasis on the abrupt changes during the training phase, utilizing our framework may limit the model’s ability to cope with the abrupt changes in the test phase. However, we confirm that our proposed framework does not degrade the forecasting performance on the abrupt changes. Table 4 reports the MSE of test samples by categorizing them into time series with abrupt changes and those without abrupt changes. For the experiment, we generated synthetic time-series dataset from sinusoidal function of t and injected abrupt changes into the sinusoidal series since the real-world dataset does not have labels for abrupt changes. We conducted the experiments with four recent state-of-the-art time-series forecasting models. As originally intended, applying our framework achieves larger MSE reduction rates (*i.e.*, MSE_N) compared to the ones without ReLD. As for the MSE of abrupt changes (*i.e.*, MSE_A), the MSE_A of three models decreased, and those of Pyraformer show competitive forecasting results. This result shows that our ReLD improves the forecasting performance on normal samples while preserving the robustness on the abrupt changes. This again demonstrates the general applicability of our work since it can be applied to a given dataset regardless of the existence of abrupt changes.

Different performance gains across datasets From the multivariate results (Table 1) and univariate results (Table 2), we found that the reduction rates of MSE vary depending on the datasets. For an in-depth analysis, we present the correlation between the average of reduction rate and the average of LD for each dataset using the scatter plot in Figure 5. We observed that there exists a positive linear correlation between LD and the reduction rate, indicating that we obtain a higher reduction rate of MSE as the average of LD increases in a given dataset. To further demonstrate

Table 4: Forecasting results by categorizing time-series sequences into normal states and abrupt changes. We observe that our ReLD significantly reduces MSE on normal states (MSE_N) while also showing comparable MSE on abrupt changes (MSE_A).

Prediction length		48		96		168		336		720		Averaged Imp.	
Model	Metric	Base	ReLD	Base	ReLD	Base	ReLD	Base	ReLD	Base	ReLD		Total
Pyraformer	MSE_N	0.0702	0.0305	0.0580	0.0232	0.0547	0.0221	0.0449	0.0326	0.0379	0.0247	-47.67%	-27.38%
	MSE_A	0.4289	0.4168	0.5525	0.5905	0.6093	0.5882	0.4300	0.4350	0.2555	0.2665	1.21%	
Autoformer	MSE_N	0.2063	0.1473	0.2560	0.1430	0.2137	0.1261	0.3645	0.1857	0.5099	0.3950	-37.06%	-33.25%
	MSE_A	0.7385	0.6946	0.9878	0.7312	0.8152	0.6541	0.8185	0.6224	0.8230	0.6776	-18.66%	
N-BEATS	MSE_N	0.0472	0.0345	0.0592	0.0401	0.0469	0.0355	0.0646	0.0411	0.0517	0.0394	-28.73%	-17.84%
	MSE_A	0.3331	0.3794	0.6109	0.5375	0.5989	0.5944	0.4597	0.4197	0.2853	0.2909	-1.12%	
Informer	MSE_N	0.1350	0.0538	0.0819	0.0341	0.0762	0.0344	0.2954	0.0492	0.5564	0.1775	-64.96%	-51.84%
	MSE_A	0.5547	0.4746	0.5625	0.6031	0.6011	0.5810	0.7533	0.4694	0.7724	0.3619	-20.28%	

such a finding, we intentionally inject abrupt changes into the Traffic and ECL, the datasets which showed the marginal improvements in the univariate setting. We obtained a larger reduction rate of MSE with both Traffic and ECL including intentional abrupt changes compared to the original datasets. This demonstrates that the marginal performance gain in both Traffic and ECL is due to the few number of abrupt changes in the dataset. Note that datasets without such abrupt changes might be well estimated with existing time-series forecasting models. However, we emphasize that using ReLD does not degrade performance on such datasets, if not marginally improve it, due to a few number of abrupt changes inevitably included in time-series datasets. On the other hand, throughout the paper, we demonstrated that datasets including a non-trivial number of abrupt changes need improvements and using ReLD effectively improves the time-series forecasting performance on such datasets. Abrupt changes in time-series data is unknown beforehand and our proposed framework does not require whether there exists such changes in the dataset. In this perspective, applying ReLD as default on existing models improves the time-series forecasting performance without prior knowledge on the abrupt changes in a given dataset.

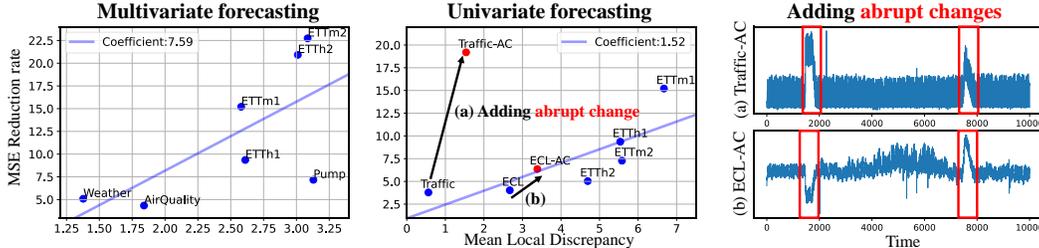


Figure 5: Scatter plots showing the correlation between the averaged LD of each dataset and MSE reduction rates of experiments on the multivariate and univariate settings. We observed that there exists a positive linear correlation between LD and the MSE reduction rate. To further demonstrate such a finding, we intentionally inject the abrupt changes highlighted as the red boxes in (a) and (b) into the traffic and ECL datasets, respectively.

5 DISCUSSION & LIMITATIONS

In this paper, we reveal that abrupt changes between adjacent sequences deteriorate the generalization performance of time-series forecasting models by occupying most of the losses despite their scarce occurrence in the training set. To solve this problem, we propose a simple yet effective reweighting framework that down-weights loss values of abrupt changes and up-weights those of normal states based on LD density. Although our ReLD consistently enhances the performance on real-world datasets, there are two limitations we found. First, we assume that an abrupt change is caused by unobserved external variables. However, if we can have access to those variables, our framework may not show performance improvement from the down-weighted losses of the abrupt changes. We further discuss this limitation in Appendix B.1. Another limitation is that LD may be formulated with other statistics that can capture temporal changes. We report the results of balancing MSE using other statistics (e.g., t -squared for multivariate test and KPSS (Kwiatkowski et al., 1992) for a stationarity test) in Appendix C.3 which shows that utilizing t -statistics (the statistic used in our work) outperforms them. Rather than using predefined statistics to identify abrupt changes, training learnable parameters for detecting them similarly done in time-series anomaly detection (Malhotra et al., 2016; Audibert et al., 2020; Xu et al., 2021) would be another way to balance MSE against abrupt changes.

REFERENCES

- Electricity dataset. <https://archive.ics.uci.edu/ml/datasets/electricityloadaddiagrams20112014>.
- Pump sensor dataset. <https://www.kaggle.com/datasets/nphantawee/pump-sensor-data>.
- Weather dataset. <https://www.bgc-jena.mpg.de/wetter/>.
- Ahmad S Ahmad, Mohammad Y Hassan, Md Pauzi Abdullah, Hasimah A Rahman, F Hussin, Hayati Abdullah, and Rahman Saidur. A review on applications of ann and svm for building electrical energy consumption forecasting. *Renewable and Sustainable Energy Reviews*, 33:102–109, 2014.
- OD Anderson. Time-series. 2nd edn., 1976.
- Devansh Arpit, Stanisław Jastrzundefinedbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proc. the International Conference on Machine Learning (ICML)*. JMLR.org, 2017.
- Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: unsupervised anomaly detection on multivariate time series. In *Proc. the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 3395–3404, 2020.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.
- Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- Sungha Choi, Joanne T. Kim, and Jaegul Choo. Cars can’t fly up in the sky: Improving urban-scene segmentation via height-driven attention networks. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, June 2020.
- Saverio De Vito, Ettore Massera, Marco Piga, Luca Martinotto, and Girolamo Di Francia. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2):750–757, 2008.
- Rahul Duggal, Scott Freitas, Sunny Dhamnani, Duen Horng Chau, and Jimeng Sun. HAR: hardness aware reweighting for imbalanced datasets. In *2021 IEEE International Conference on Big Data (Big Data)*, Orlando, FL, USA, December 15-18, 2021, pp. 735–745. IEEE, 2021.
- Clive William John Granger and Paul Newbold. *Forecasting economic time series*. Academic Press, 2014.
- Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8535–8545, 2018.
- Harold Hotelling. The generalization of student’s ratio. In *Breakthroughs in statistics*, pp. 54–65. Springer, 1992.
- Chenyu Hou, Jiawei Wu, Bin Cao, and Jing Fan. A deep-learning prediction model for imbalanced time series data forecasting. *Big Data Mining and Analytics*, 4(4):266–278, 2021.
- Sanghun Jung, Jungsoo Lee, Daehoon Gwak, Sungha Choi, and Jaegul Choo. Standardized max logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation. In *Proc. of the IEEE international conference on computer vision (ICCV)*, pp. 15425–15434, October 2021.

- Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Proc. the International Conference on Learning Representations (ICLR)*, 2020.
- Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *Proc. the International Conference on Learning Representations (ICLR)*, 2019.
- Denis Kwiatkowski, Peter CB Phillips, Peter Schmidt, and Yongcheol Shin. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics*, 54(1-3):159–178, 1992.
- Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104, 2018.
- Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *Proc. the International Conference on Learning Representations (ICLR)*, 2020a.
- Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyong Zhou, Wenhui Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 10991–11000, 2020b.
- Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017. URL http://openaccess.thecvf.com/content_ICCV_2017/papers/Lin_Focal_Loss_for_ICCV_2017_paper.pdf.
- Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *Proc. the International Conference on Learning Representations (ICLR)*, 2022.
- Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2019.
- Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.
- Nuno Moniz, Paula Branco, and Luís Torgo. Resampling strategies for imbalanced time series forecasting. *International Journal of Data Science and Analytics*, 3(3):161–181, 2017.
- Kemal Oksuz, Baris Can Cam, Sinan Kalkan, and Emre Akbas. Imbalance Problems in Object Detection: A Review. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 1–1, 2020.
- Boris N Oreshkin, Dmitri Carпов, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *Proc. the International Conference on Learning Representations (ICLR)*, 2019.
- Jiawei Ren, Mingyuan Zhang, Cunjun Yu, and Ziwei Liu. Balanced mse for imbalanced visual regression. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2022.
- David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

- Afan Galih Salman, Bayu Kanigoro, and Yaya Heryadi. Weather forecasting using deep learning techniques. In *2015 international conference on advanced computer science and information systems (ICACSIS)*, pp. 281–285. Ieee, 2015.
- Hwanjun Song, Minseok Kim, and Jae-Gil Lee. SELFIE: Refurbishing unclean samples for robust deep learning. In *Proc. the International Conference on Machine Learning (ICML)*, 2019.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias. Short-term traffic forecasting: Where we are and where we’re going. *Transportation Research Part C: Emerging Technologies*, 43: 3–19, 2014.
- Bernard L Welch. The significance of the difference between two means when the population variances are unequal. *Biometrika*, 29(3/4):350–362, 1938.
- Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. *Proc. the International Conference on Learning Representations (ICLR)*, 2021.
- Yuzhe Yang, Kaiwen Zha, Ying-Cong Chen, Hao Wang, and Dina Katabi. Delving into deep imbalanced regression. In *Proc. the International Conference on Machine Learning (ICML)*, 2021.
- Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. 2016. URL <http://arxiv.org/abs/1611.03530>.
- Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proc. the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.
- Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. the International Conference on Machine Learning (ICML)*, 2022.

A THEORETICAL ANALYSIS OF RE LD

As shown in Figure 3 of main paper, we can observe LD values oscillating within a certain range similarly to original time series. We discuss the following points about this observation.

1. When the time series is sampled from a periodic function, LD is also a periodic function.
2. When the time series is sampled from a bounded periodic function, LD is bounded.

First, we can easily prove that LD is also periodic when the time series is sampled from a periodic function, which the model should learn from data (i.e., normal states).

Theorem 1. *If f is a periodic function that satisfies $f(t) = f(t + p)$,*

$$LD(a, a + L) = \frac{m(a) - m(a + L)}{\sqrt{\frac{s(a)}{N} + \frac{s(a+L)}{N}}} \quad (3)$$

is also a periodic function with period p , where $m(a) = \frac{1}{N} \sum_{t \in I(a)} f(t)$, $s(a) = \frac{1}{N} \sum_{t \in I(a)} (f(t) - m(a))^2$, and $I(a) = \{a + \frac{L}{N} \cdot i\}_{i=0}^{N-1}$ for range $[a, a + L]$ and sampling interval L/N .

To prove Theorem 1, we prove and use Proposition 1 and 2 with respect to the mean and the variance of the periodic function.

Proposition 1. *If f is a periodic function that satisfies $f(t) = f(t + p)$, $m(a) = \frac{1}{N} \sum_{t \in I(a)} f(t)$ is also a periodic function with period p where $I(a) = \{a + \frac{L}{N} \cdot i\}_{i=0}^{N-1}$ for range $[a, a + L]$ and sampling interval L/N .*

Proposition 2. *If f is a periodic function that satisfies $f(t) = f(t + p)$, $s(a) = \frac{1}{N} \sum_{x \in I(a)} (f(t) - m(a))^2$ is also a periodic function with period p where $I(a) = \{a + \frac{L}{N} \cdot i\}_{i=0}^{N-1}$ for range $[a, a + L]$ and sampling interval L/N .*

By proposition 1 and 2, we prove Theorem 1 as follows:

$$\begin{aligned} LD(a + p, a + p + L) &= \frac{m(a + p) - m(a + p + L)}{\sqrt{\frac{s(a+p)}{N} + \frac{s(a+p+L)}{N}}} \\ &= \frac{m(a) - m(a + L)}{\sqrt{\frac{s(a)}{N} + \frac{s(a+L)}{N}}} \\ &= LD(a, a + L) \end{aligned}$$

Regarding the bound of LD, if we define LD by ignoring the variances (i.e., $LD(a, a + L) = m(a) - m(a + L)$), we can obtain the bound $2 \cdot B$ given that f is bounded function that satisfies $|f(t)| \leq B$ for all t .

However, since we use the variance of input and output, LD can diverge when both variances of input sequence and output sequence are equal to zero. There are two cases when the variance equals to zero.

1. f is a constant function.
2. The window size L is $N \cdot p$ for data points, which are sampled from a periodic function f with the period p and sampling interval L/N .

In the first case, since the time-series dataset has a constant target value, the prediction also remains as the constant value, leading to a trivial solution. In the second case, the variance is no longer zero if the window size L is adjusted.

In practice, we use epsilon ϵ as a numerical stabilizer to solve the case where variances are zero as shown in Equation 1. Note that we do not use these bounds as thresholds in the proposed method.

Proofs for the Proposition 1 and 2 are as follows.

Proposition 1. *If f is periodic function that satisfy $f(t) = f(t + p)$, $m(a) = \frac{1}{N} \sum_{t \in I(a)} f(t)$ is also periodic function with period p where $I(a) = \{a + \frac{L}{N} \cdot i\}_{i=0}^{N-1}$ for range $[a, a + L]$ and sampling interval L/N .*

Proof:

$$\begin{aligned} m(a+p) &= \frac{1}{N} \sum_{t \in I(a+p)} f(t) \\ &= \frac{1}{N} \left\{ f(a+p) + f\left(a+p + \frac{L}{N}\right) + \cdots + f\left(a+p + L - \frac{L}{N}\right) \right\} \\ &= \frac{1}{N} \left\{ f(a) + f\left(a + \frac{L}{N}\right) + \cdots + f\left(a + L - \frac{L}{N}\right) \right\} \\ &= m(a) \end{aligned}$$

Proposition 2. *If f is periodic function that satisfy $f(t) = f(t + p)$, $s(a) = \frac{1}{N} \sum_{t \in I(a)} (f(t) - m(a))^2$ is also periodic function with period p where $I(a) = \{a + \frac{L}{N} \cdot i\}_{i=0}^{N-1}$ for range $[a, a + L]$ and sampling interval L/N .*

Proof:

$$\begin{aligned} s(a+p) &= \frac{1}{N} \sum_{t \in I(a+p)} \{f(t) - m(a+p)\}^2 \\ &= \frac{1}{N} \left\{ (f(a+p) - m(a+p))^2 + \left(f\left(a+p + \frac{L}{N}\right) - m(a+p)\right)^2 \right. \\ &\quad \left. + \cdots + \left(f\left(a+p + L - \frac{L}{N}\right) - m(a+p)\right)^2 \right\} \\ &= \frac{1}{N} \left\{ (f(a) - m(a))^2 + \left(f\left(a + \frac{L}{N}\right) - m(a)\right)^2 + \cdots + \left(f\left(a + L - \frac{L}{N}\right) - m(a)\right)^2 \right\} \\ &= s(a) \end{aligned}$$

B IN-DEPTH ANALYSIS ON RELD

This section provides various analysis for our reweighting framework.

B.1 ABRUPT CHANGE WITH EXTERNAL VARIABLES

We assumed that an abrupt change can be caused by unobserved and external events as we mentioned in Section 1 and Section 5. If the abrupt change can be predicted using an external variable, down-weighting the loss of the abrupt change would get in the way of learning such correlation for the model. However, utilizing additional variables without thorough verification causes the model to learn a spurious correlation between variables, which worsens the generalization ability. Moreover, some abrupt changes have unknown causes (e.g., sensor malfunction), which cannot be addressed by simply collecting external variables. In fact, as shown in the Table 5, we observed that training baseline models with external variables (i.e., Multivariate to Univariate denoted as Mul2Uni setting) rather shows lower performance than training those with the target time series only (i.e., Univariate to Univariate denoted as Uni2Uni setting). These results indicate that simply adding covariates does not guarantee performance gains. Note that multivariate forecasting we mentioned in main paper is multivariate to multivariate setting (i.e., Mul2Mul), which is different with Mul2Uni setting. In addition, applying our method on Mul2Uni outperformed the Uni2Uni in several cases (see Autoformer 96/96 and 336/168 of Table 5).

B.2 RELD ON REPEATED CHANGES

To further understand our ReLD, we present a rectangular time series as a special case, which generally includes a large shift during a short period of time and shows increasing amplitude (see 1st row

Table 5: Comparison between Mul2Uni and Uni2Uni forecasting on ETTm1 dataset.

Setting	Model	Pyraformer			Autoformer			
		I → O	96 → 96	336 → 168	336 → 336	96 → 96	336 → 168	336 → 336
	Uni2Uni		0.0821 ± 0.0289	0.1286 ± 0.0346	0.1941 ± 0.0523	0.0577 ± 0.0081	0.0881 ± 0.0284	0.0903 ± 0.0096
	Uni2Uni + Ours		0.0576 ± 0.0079	0.1218 ± 0.0395	0.1843 ± 0.0507	0.0522 ± 0.0035	0.0723 ± 0.0068	0.0847 ± 0.0084
	Mul2Uni		0.1757 ± 0.0372	0.2926 ± 0.0778	0.5920 ± 0.0591	0.0619 ± 0.0090	0.0799 ± 0.0188	0.1367 ± 0.0392
	Mul2Uni + Ours		0.1137 ± 0.0295	0.2984 ± 0.1246	0.5533 ± 0.0761	0.0496 ± 0.0021	0.0674 ± 0.0071	0.1193 ± 0.0267

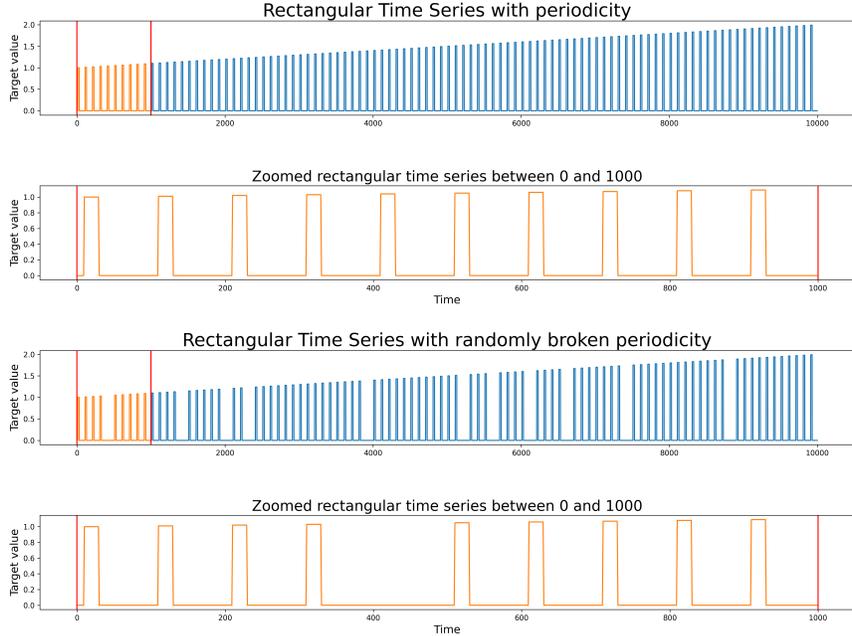


Figure 6: Two rectangular time series which include large shift in a short time. As in the first row, if a rectangular pattern with a large change exists several times, ReLD learns it normally without down-weighting it. However, if the periodicity is broken by sensor malfunctions in the third row, ReLD mitigates impact of anomaly pattern in training phase.

of Figure 6). Although this series includes large shifts, we do not regard those as abrupt changes defined in our paper since rectangular patterns are repeated (i.e., seasonal component). Also, the increasing amplitude (i.e., trend component) is considered one of the trend types. Since we calculate LD by sliding the window, the increasing amplitude does not change the LD values. For example, the LD value of the window, which of size is large enough to cover period, has a value less than 0 (greater than 0 if the amplitude decreases) regardless of time. In this case, since the LD values of all windows are similar, they will be given the same weights. Therefore, even if our method is applied, we would observe more or less the same performance as shown in Rect-Normal dataset of Table 6.

Additionally, we conducted experiments by removing rectangles randomly from the dataset (see 3rd row of Figure 6). This can be considered abrupt changes (e.g., broken periodicity). We observe that our ReLD brings performance gain in such cases (see Rect-Broken of Table 6). This again demonstrates that our proposed method promotes the model to be robust to abrupt changes.

B.3 IMPACT OF THE IN-OUTPUT RATIO

We conducted an experiment by fixing the output length and changing the input length from 48 to 720 to explore the performance change according to the I/O ratio. We conducted experiments on the three datasets, ETTh1, ETTh2, and ETTm1. Applying our method brings consistent performance

Table 6: Rectangular Time Series with the increasing amplitude and the randomly broken periodicity.

Models		Pyraformer		Autoformer		Informer	
MSE		base	our	base	our	base	our
Rect-Normal	96	0.2405 ± 0.0211	0.2468 ± 0.0243	0.9748 ± 0.4805	0.9131 ± 0.4100	0.5409 ± 0.0280	0.5480 ± 0.0316
	168	0.2614 ± 0.0115	0.2652 ± 0.0102	1.4711 ± 0.9902	1.7147 ± 0.7723	1.3573 ± 0.1377	1.3556 ± 0.1068
	336	0.3179 ± 0.0075	0.3193 ± 0.0062	0.5271 ± 0.2047	0.4492 ± 0.1558	1.2945 ± 0.0874	1.2764 ± 0.0891
	720	0.4034 ± 0.0084	0.4088 ± 0.0089	2.7076 ± 0.4390	2.6447 ± 0.6276	1.6796 ± 0.0450	1.7306 ± 0.0614
	Imp.	1.46%		-1.72%		0.71%	
Rect-Broken	96	0.4028 ± 0.0434	0.2343 ± 0.0061	0.9883 ± 0.1790	1.1399 ± 0.4871	0.5781 ± 0.0558	0.3912 ± 0.0463
	168	0.3261 ± 0.0138	0.3020 ± 0.0174	1.7361 ± 0.8130	1.2914 ± 0.3595	0.5622 ± 0.0386	0.5227 ± 0.0178
	336	0.2548 ± 0.0112	0.2579 ± 0.0155	0.7678 ± 0.1225	0.5648 ± 0.1603	0.5144 ± 0.0394	0.4988 ± 0.0494
	720	0.3098 ± 0.0617	0.3037 ± 0.0427	2.0861 ± 0.1504	1.8670 ± 0.3713	0.5936 ± 0.0473	0.5569 ± 0.0390
	Imp.	-12.49%		-11.80%		-12.14%	

Table 7: Impact of the ratio I/O on multivariate time series forecasting.

Models		Pyraformer		Autoformer		Informer		Imp.
Output-96	Input	base	our	base	our	base	our	
ETTh1	48	0.6314 ± 0.0371	0.5166 ± 0.0104	0.4748 ± 0.0328	0.4675 ± 0.0504	1.0632 ± 0.2707	0.8125 ± 0.0679	-23.58%
	96	0.6453 ± 0.0583	0.5345 ± 0.0073	0.4531 ± 0.0282	0.4452 ± 0.0153	0.9075 ± 0.0479	0.8476 ± 0.0532	-6.6%
	168	0.6330 ± 0.0241	0.5604 ± 0.0145	0.4477 ± 0.0247	0.4594 ± 0.0426	0.8997 ± 0.0738	0.7928 ± 0.0698	-11.88%
	336	0.7195 ± 0.0206	0.6310 ± 0.0293	0.4667 ± 0.0240	0.4826 ± 0.0188	1.1695 ± 0.2012	1.0384 ± 0.1830	-11.21%
	720	0.7290 ± 0.0757	0.6540 ± 0.0191	0.6354 ± 0.0386	0.5057 ± 0.0673	1.6608 ± 0.1402	1.3866 ± 0.1341	-16.51%
ETTh2	48	1.5411 ± 0.1880	1.0982 ± 0.1702	0.3637 ± 0.0091	0.3394 ± 0.0051	1.7225 ± 0.1508	1.1461 ± 0.0743	-22.96%
	96	1.6090 ± 0.0866	1.1733 ± 0.2271	0.3731 ± 0.0294	0.3464 ± 0.0102	3.4245 ± 0.4814	2.4505 ± 0.4804	-20.89%
	168	1.7787 ± 0.2003	1.3081 ± 0.2461	0.4414 ± 0.0271	0.3833 ± 0.0069	5.6370 ± 0.8005	2.9705 ± 0.4835	-28.97%
	336	1.7924 ± 0.2872	1.5560 ± 0.1676	0.4897 ± 0.0565	0.4174 ± 0.0517	6.2992 ± 0.9310	3.9496 ± 0.8050	-21.75%
	720	2.0959 ± 0.1960	1.9368 ± 0.2612	0.6769 ± 0.1552	0.4701 ± 0.0869	9.1387 ± 2.0638	6.9792 ± 1.5690	-20.59%
ETTh1	48	0.5559 ± 0.0225	0.4922 ± 0.0134	0.5673 ± 0.0542	0.5182 ± 0.0396	0.6389 ± 0.0270	0.5925 ± 0.0348	-9.13%
	96	0.5364 ± 0.0318	0.4713 ± 0.0299	0.5128 ± 0.0635	0.4545 ± 0.0410	0.6438 ± 0.0596	0.5367 ± 0.0439	-13.38%
	168	0.5015 ± 0.0431	0.4174 ± 0.0176	0.4987 ± 0.0241	0.4603 ± 0.0636	0.6907 ± 0.0578	0.5677 ± 0.0281	-14.09%
	336	0.4876 ± 0.0325	0.4316 ± 0.0171	0.5374 ± 0.0361	0.5053 ± 0.0462	0.8487 ± 0.0578	0.6078 ± 0.0541	-15.28%
	720	0.4841 ± 0.0381	0.4546 ± 0.0224	0.5799 ± 0.0914	0.4988 ± 0.0384	1.0951 ± 0.1741	0.8007 ± 0.1602	-15.65%

Table 8: The processing time of ReLD and training time of Autoformer during 1 epoch.

Dataset	# of Windows	Window size (I + O)	# of Series	ReLD Preprocessing time (a) (seconds)	Training time (b) (seconds per epoch)	Ratio (a) / (a) + (b)
ETTh1	8449	192 (96 + 96)	7	0.18	38.12	0.47%
ETTh2	8449	192 (96 + 96)	7	0.17	39.11	0.43%
ETTM1	34369	192 (96 + 96)	7	0.69	154.68	0.44%
ETTM2	34369	192 (96 + 96)	7	0.68	160.31	0.42%
Weather-hour	5093	1056 (336 + 720)	21	0.63	121.19	0.52%
Pump	9610	672 (336 + 336)	35	1.22	125.30	0.96%
ECL	17741	672 (336 + 336)	1	0.08	94.67	0.08%
Traffic	11225	1056 (336 + 720)	1	0.07	99.05	0.07%

improvements, although there exists different performance gains depending on the input lengths as shown in Table 7.

B.4 COMPUTATIONAL COST OF ReLD

Our reweighting framework requires a marginal amount of additional computational cost of calculating the weights for all input-output sequences before training. As shown in Table 8, the cost of calculating the weights on datasets with multiple settings is less than 1% of the time it takes to train with the dataset during one epoch. The absolute time was mostly less than 1 second.

C COMPARISON WITH OTHER METHODS

C.1 COMPARISON WITH SMOOTHING AND OUTLIER FILTERING

We compared our proposed method with 1) smoothing and 2) outlier filtering which are expected to perform well with drastic changes (e.g., fluke) in time-series datasets. Smoothing techniques are used to remove nosiness and reduce outliers, allowing meaningful temporal patterns to stand out. Conventional methods include moving average (MA) smoothing as follows:

$$s_t = \frac{(x_{t-k+1} + x_{t-k+2} + \dots + x_t)}{k} \quad (4)$$

where s_t is the smoothed observation at t and x_t is the original observation. The other method is exponential (EMA) smoothing calculated by Equation as follows:

$$s_t = \alpha \cdot x_t + (1 - \alpha) \cdot s_{t-1} \quad (5)$$

where $\alpha \in (0, 1)$. We smoothed the training time series and train forecasting models. To use outlier filtering method for forecasting task, a simple way to detect outliers is to assume that the target value follows a Gaussian and remove values that exceed a certain range of values. We train forecasters after removing outliers which exceed a certain value.

C.2 COMPARISON WITH ERROR-BASED REWEIGHTING

As we mentioned in the main paper, we observed that abrupt changes significantly contribute to the total loss in the training phase. In this situation, we can simply reweight a loss of sample that have large error while considering the sample including abrupt change. Reweighting inversely to the error may down-weight the loss of the abrupt change without additional LD calculation. In the main paper, we presented two error-based methods: Focal-R and filp Focal-R. Focal-R loss is calculated as $\sigma(\beta |e_i|)^\gamma L_i$ where e_i is error of i -th sample, L_i is loss of i -th sample, and $\sigma(\cdot)$ is sigmoid function. β and γ are hyperparameters. In case of filp Focal-R, β is negative to flip the sigmoid function along the y axis. Additionally, we provide L2 error-based reweighting results, namely invL2 which is written as $\frac{L_i}{e_i + \epsilon}$. In case of invL2, as the model forecasts accurately and thus the error of the normal states is close to zero, the parameter moves with larger steps by up-weighted loss. Table 9 shows the performance in the case of reweighting inversely to the error of each window.

Table 9: Comparison with error-based reweighting (invL2) in the multivariate forecasting (Top) and in the univariate forecasting (Bottom) using ETTh1 dataset.

Multivariate I / O	Pyraformer			Autoformer			Informer		
	base	invL2	ReLD	base	invL2	ReLD	base	invL2	ReLD
96 / 96	0.6453 ± 0.0583	0.6083 ± 0.0149	0.5345 ± 0.0073	0.4422 ± 0.0242	0.4458 ± 0.0212	0.4438 ± 0.0143	0.9084 ± 0.0485	0.8506 ± 0.0280	0.8031 ± 0.0317
336 / 168	0.8644 ± 0.0905	0.7842 ± 0.0250	0.7415 ± 0.0399	0.5042 ± 0.0515	0.4772 ± 0.0144	0.4906 ± 0.0263	1.3720 ± 0.2422	1.2150 ± 0.1333	0.8858 ± 0.0258
336 / 336	0.9328 ± 0.0341	0.9643 ± 0.0404	0.8895 ± 0.0548	0.5694 ± 0.1115	0.5450 ± 0.0886	0.5110 ± 0.0990	1.3425 ± 0.0725	1.2857 ± 0.0710	0.9850 ± 0.0308
336 / 720	0.9843 ± 0.0213	1.0003 ± 0.0228	0.9781 ± 0.0196	0.5348 ± 0.0212	0.5589 ± 0.0613	0.5207 ± 0.0106	1.3933 ± 0.0892	1.3735 ± 0.0386	1.1994 ± 0.0597
Imp.	-	-2.50%	-9.17%	-	-1.08%	-3.81%	-	-5.86%	-21.89%

Univariate I / O	Pyraformer			Autoformer			Informer		
	base	invL2	ReLD	base	invL2	ReLD	base	invL2	ReLD
96 / 96	0.2074 ± 0.0728	0.1928 ± 0.0365	0.1831 ± 0.0533	0.0859 ± 0.0063	0.0861 ± 0.0031	0.0841 ± 0.0067	0.1203 ± 0.0730	0.1132 ± 0.0441	0.1020 ± 0.0472
336 / 168	0.1819 ± 0.0257	0.1750 ± 0.0581	0.1725 ± 0.0406	0.1077 ± 0.0130	0.0949 ± 0.0109	0.0999 ± 0.0072	0.0862 ± 0.0292	0.0946 ± 0.0244	0.0848 ± 0.0297
336 / 336	0.1716 ± 0.0597	0.1853 ± 0.0622	0.1649 ± 0.0426	0.1055 ± 0.0219	0.1135 ± 0.0198	0.1008 ± 0.0157	0.0862 ± 0.0025	0.0870 ± 0.0084	0.0897 ± 0.0165
336 / 720	0.1974 ± 0.0415	0.1746 ± 0.0312	0.1667 ± 0.0298	0.1352 ± 0.0207	0.1251 ± 0.0110	0.1244 ± 0.0270	0.2025 ± 0.0961	0.1800 ± 0.0945	0.1550 ± 0.0237
Imp.	-	-3.60%	-9.09%	-	-2.88%	-5.45%	-	-1.59%	-9.06%

C.3 VARIANTS FOR LOCAL DISCREPANCY

We propose the *Local Discrepancy* (LD) based on the statistics formulated by a statistical test, Welch’s t-test Welch (1938), in order to measure how two adjacent in-output sequences, \mathcal{X}_t and \mathcal{Y}_t , are different from each other. There may exist other metrics to measure the local discrepancy such as multivariate t-statistic Hotelling (1992) (i.e., Hotelling’s *t*-squared statistic) and stationarity tests (e.g., Kwiatkowski–Phillips–Schmidt–Shin (KPSS) tests Kwiatkowski et al. (1992)). We also report the performance of our reweighting framework using a different metric other than t-statistics for measuring the local discrepancy in Table 10. Hotelling’s *t*-squared statistic is a generalization of Student’s *t*-statistic that is used in multivariate hypothesis testing. We can naturally utilize *t*-squared statistic as LD for multivariate forecasting (i.e., $s \in \mathbb{R}^m$ and $m > 1$) as follows:

$$\text{LocalDis}(\mathcal{X}_t, \mathcal{Y}_t) = \frac{I \cdot O}{I + O} (\bar{\mathcal{X}}_t - \bar{\mathcal{Y}}_t)' \hat{\Sigma}^{-1} (\bar{\mathcal{X}}_t - \bar{\mathcal{Y}}_t) := v_t^2 \quad (6)$$

where the mean and covariance are defined as follows:

$$\bar{\mathcal{X}}_t = \frac{1}{I} \sum_{i=1}^I s_{t-i}, \quad \bar{\mathcal{Y}}_t = \frac{1}{O} \sum_{i=0}^{O-1} s_{t+i}, \quad \hat{\Sigma} = \frac{(I-1) \hat{\Sigma}_{\bar{\mathcal{X}}} + (O-1) \hat{\Sigma}_{\bar{\mathcal{Y}}}}{I + O - 2},$$

$$\hat{\Sigma}_{\bar{\mathcal{X}}} = \frac{1}{I-1} \sum_{i=1}^I (s_{t-i} - \bar{\mathcal{X}}_t) (s_{t-i} - \bar{\mathcal{X}}_t)', \quad \hat{\Sigma}_{\bar{\mathcal{Y}}} = \frac{1}{O-1} \sum_{i=1}^{O-1} (s_{t+i} - \bar{\mathcal{Y}}) (s_{t+i} - \bar{\mathcal{Y}})'.$$

We can interpret the time-series data in terms of stochastic processes. KPSS tests are used for testing a null hypothesis that an observable time series is stationary around a deterministic trend (i.e., trend-stationary) against the alternative of a unit root. When the given time series is trend stationary, the KPSS statistic has small value, which is close to zero. Thus, to measure the degree of abruptness of a change in a given period of time, we leverage the KPSS statistic as LD:

$$\text{LocalDis}(\text{concat}(\mathcal{X}_t, \mathcal{Y}_t)) = \frac{1}{(I+O)^2} \cdot \sum_{i=-I}^{O-1} \frac{\mathcal{E}_{t+i}^2}{\hat{\sigma}^2} := v_t \quad (7)$$

where \mathcal{E}_t is partial sum of the residuals and $\hat{\sigma}^2$ is the estimate of the long-run variance of the residuals as follows:

$$\mathcal{E}_k = \sum_{i=1}^k e_i, \quad e = (e_{t-I}, e_{t-I+1}, \dots, e_{O-1})$$

where e means OLS residuals when regressing the concated in-output sequence (i.e., $\text{concat}(\mathcal{X}_t, \mathcal{Y}_t)$). We observe that our reweighting framework consistently outperforms the ones without our framework regardless of the statistics used for measuring the local discrepancy. While we empirically confirmed that using t-statistic is more suitable for LD compared to KPSS or t-Squared statistic, such result demonstrates that our framework can be used with any statistics measure the user deems appropriate.

Table 10: Ablation study on variants of local discrepancy used in our reweighting framework. We compare models which uses 1) KPSS, 2) t -squared, and 3) t -statistic. The t -statistic shows more consistent and superior results compared to other statistics in the multivariate setting.

Model	Dataset Predict-O	ETTh1			ETTh2			ETTm1			Imp.
		96	168	336	96	168	336	96	168	336	
Pyraformer		0.645	0.864	0.933	1.609	5.014	4.356	0.536	0.563	0.697	-
Pyraformer + KPSS		0.554	0.782	0.909	1.482	4.590	5.327	0.470	0.527	0.604	-5.84%
Pyraformer + t -squared		0.640	0.809	0.898	1.440	3.112	3.912	0.490	0.557	0.632	-9.84%
Pyraformer + t -statistic		0.534	0.742	0.889	1.173	3.976	3.281	0.471	0.506	0.573	-16.51%
Autoformer		0.442	0.504	0.569	0.386	0.439	0.494	0.524	0.534	0.561	-
Autoformer + KPSS		0.446	0.528	0.486	0.358	0.436	0.516	0.456	0.538	0.513	-3.68%
Autoformer + t -squared		0.454	0.521	0.515	0.357	0.403	0.436	0.503	0.548	0.512	-4.60%
Autoformer + t -statistic		0.444	0.491	0.511	0.351	0.413	0.424	0.455	0.500	0.514	-7.74%
Informer		0.908	1.372	1.343	3.400	5.796	3.901	0.640	1.224	1.390	-
Informer + KPSS		0.850	1.215	1.215	3.050	5.593	4.202	0.535	0.844	1.087	-11.41%
Informer + t -squared		0.871	1.262	1.234	2.796	4.393	3.419	0.594	0.992	1.195	-12.76%
Informer + t -statistic		0.856	1.113	1.151	2.462	4.723	3.788	0.543	0.751	1.008	-18.81%

D OUR FRAMEWORK DETAILS

D.1 IMPLEMENTATION DETAILS

We include 12 baselines to validate our ReLD. All models were implemented with PyTorch. As for recent models (*i.e.*, FEDformer², Pyraformer³, Autoformer⁴, and Informer⁵), we used the official code released by the original authors, rather than implementing from scratch. For a fair comparison between ReLD and the existing framework, we set the same hyperparameters found in each work. We trained all models from scratch to 10 epochs. To assign weights to all training samples in ReLD, the LD is computed only once before training, and it takes only a negligible amount of time compared to the training time. Most models, which leverage a generative decoding, take an average of less than an hour to train on a TITAN-Xp GPU except for LSTMa which uses auto-regressive decoding. The source code is available in the following address: <https://tinyurl.com/iclr3913>.

D.2 PSEUDO CODE FOR ReLD

Algorithm 1 ReLD: Reweighting framework based on Local Discrepancy Density

Require: Training set $\mathcal{D} = \{(\mathcal{X}_t, \mathcal{Y}_t)\}_{t=1}^N$, bin size Δb , symmetric kernel distribution $k(v, v')$

Compute Local Discrepancy LD $(\mathcal{X}_t, \mathcal{Y}_t) = \frac{\bar{\mathcal{X}}_t - \bar{\mathcal{Y}}_t}{\sqrt{\frac{s_{\mathcal{X}_t}^2}{T} + \frac{s_{\mathcal{Y}_t}^2}{O} + \epsilon}} := v_t$

Compute the empirical label density distribution $p(v)$ based on Δb and \mathcal{D}

Compute the effective label density distribution $\tilde{p}(v') := \int_{\mathcal{Y}} k(v, v') p(v) dv$

for all $(\mathcal{X}_t, \mathcal{Y}_t, v_t) \in \mathcal{D}$ **do**

Assign weight for each sample as $w_t \propto c \cdot \tilde{p}(v_t)$ (constant c as scaling factor)

end for

for all number of training iterations **do**

Sample a mini-batch $\{(\mathcal{X}_b, \mathcal{Y}_b, w_b)\}_{b=1}^B$ from \mathcal{D}

Forward $\{\mathcal{X}_b\}_{b=1}^B$ and get corresponding predictions $\{\hat{\mathcal{Y}}_b\}_{b=1}^B$

Do one training step using the weighted loss $\frac{1}{B} \sum_{b=1}^B \mathcal{L}_{w_b}(\hat{\mathcal{Y}}_b, \mathcal{Y}_b)$

end for

²<https://github.com/MAZiqing/FEDformer>

³<https://github.com/alipay/Pyraformer>

⁴<https://github.com/thuml/Autoformer>

⁵<https://github.com/zhouhaoyi/Informer2020>

Table 11: Performance change according to the number of bins.

Dataset		ETTh1	ETTh2	
Model	# bins	336 → 336	96 → 96	Imp.
Autoformer	-	0.5694 ± 0.1115	0.3859 ± 0.0260	-
Autoformer + ReLD	40	0.5245 ± 0.1543	0.3529 ± 0.0262	-8.55%
	120	0.4907 ± 0.0337	0.3455 ± 0.0229	-10.47%
	200	0.4903 ± 0.0610	0.3501 ± 0.0168	-9.28%
	300	0.4881 ± 0.0413	0.3472 ± 0.0072	-10.03%
	500	0.5130 ± 0.0529	0.3485 ± 0.0142	-9.69%

Table 12: Performance change according to the KDE kernel types.

Dataset		ETTh1	ETTh2	
Model	KDE kernel	336 → 336	96 → 96	Imp.
Autoformer	-	0.5694 ± 0.1115	0.3859 ± 0.0260	-
Autoformer + ReLD	Gaussian	0.4903 ± 0.0610	0.3501 ± 0.0168	-9.28%
	Triangle	0.4792 ± 0.0171	0.3453 ± 0.0232	-10.52%
	Laplace	0.4786 ± 0.0380	0.3496 ± 0.0087	-9.41%

Table 13: Performance changes according to the KDE kernel size.

Dataset		ETTh1	ETTh2	
Model	KDE kernel size	336 → 336	96 → 96	Imp.
Autoformer	-	0.5694 ± 0.1115	0.3859 ± 0.0260	-
Autoformer + ReLD	5	0.4903 ± 0.0610	0.3501 ± 0.0168	-9.28%
	10	0.4896 ± 0.0728	0.3466 ± 0.0018	-10.18%
	15	0.4836 ± 0.0189	0.3567 ± 0.0293	-7.57%
	20	0.4841 ± 0.0444	0.3482 ± 0.0383	-9.77%
	25	0.4835 ± 0.0317	0.3490 ± 0.0166	-9.56%

Table 14: Performance change according to the KDE kernel sigma.

Dataset		ETTh1	ETTh2	
Model	KDE kernel sigma	336 → 336	96 → 96	Imp.
Autoformer	-	0.5694 ± 0.1115	0.3859 ± 0.0260	-
Autoformer + ReLD	1	0.5545 ± 0.1916	0.3521 ± 0.0189	-8.76%
	2	0.4903 ± 0.0610	0.3501 ± 0.0168	-9.28%
	4	0.5218 ± 0.1632	0.3586 ± 0.0567	-7.07%
	8	0.4767 ± 0.0307	0.3435 ± 0.0172	-10.99%
	16	0.4836 ± 0.0318	0.3494 ± 0.0310	-9.46%

We illustrate the pseudo code of the ReLD in Algorithm 1.

D.3 HYPERPARAMETER SENSITIVITY

We used KDE to smooth the LD distribution. Related parameters include the bin size that determines how many sections continuous LD is divided into, KDE’s kernel type, kernel size and kernel sigma. In our experiment, we set the bin size to 200, kernel type to Gaussian, and kernel size and sigma to 5 and 2, respectively, as default parameters. We conducted experiments on ETTh1 and ETTh2 to observe the variance of performance according to each parameter. As shown in Table 11, Table 12,

Table 13, and Table 14, we observe that our proposed method is robust to the hyper-parameters while showing consistent performance improvements.

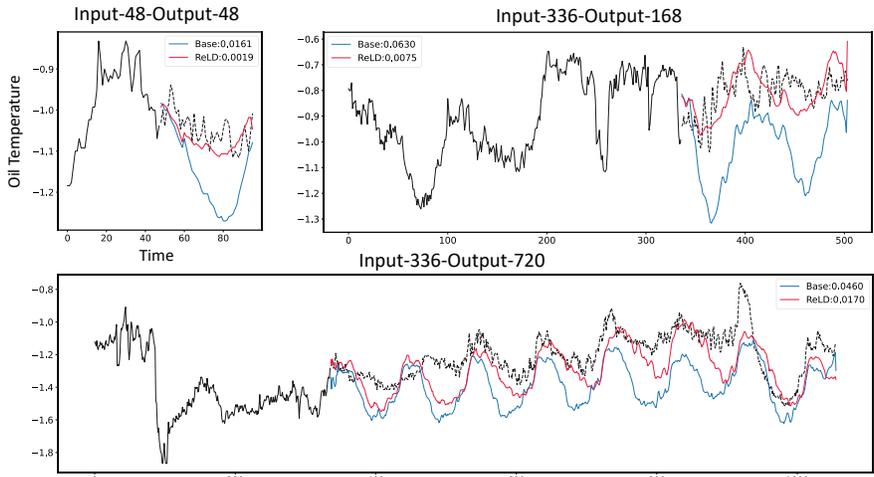


Figure 7: Forecasting results of Autoformer trained on ETTm1 with three different length settings: Input-48-Output-48, Input-336-Output-168, and Input-336-Output-720. The blue line indicates the forecasting results of the baselines without our ReLD and the red line indicates those with our ReLD.

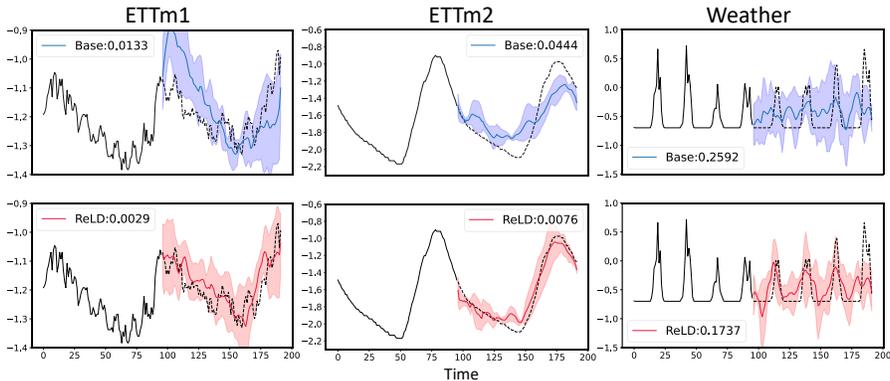


Figure 8: Forecasting results of Autoformer on three datasets: ETTm1, ETTm2, and Weather. The first row shows the forecasting results of the baseline without our ReLD and the second row shows those with our ReLD.

E QUALITATIVE RESULTS

This section visualizes the forecasting results using three criteria: in-output length (Figure 7), dataset (Figure 8), and model architecture (Figure 9). All samples are from the test set of each dataset. The solid black line denotes the input series and the dotted black line denotes the ground truth series that a model should predict. For a reliable comparison, we plot the averaged forecasting results of the independent models trained from different random initializations. The shaded part of the forecasting result indicates the forecasting variation at a given time stamp. In Figure 7, we only report the mean of forecasting results without the forecasting variation for better clarity.

As shown in Figure 7, our ReLD demonstrated enhanced forecasting results in both short-term and long-term settings. We observe that applying ReLD significantly reduces the MSE loss regardless of datasets (see Figure 8) and model architectures (Figure 9). For example, by applying ReLD on Weather dataset (see Figure 8), the prediction variations (red-shaded regions) are fitted to the fluctuations of the target times series which was underfitted without applying ReLD (blue-shaded regions).

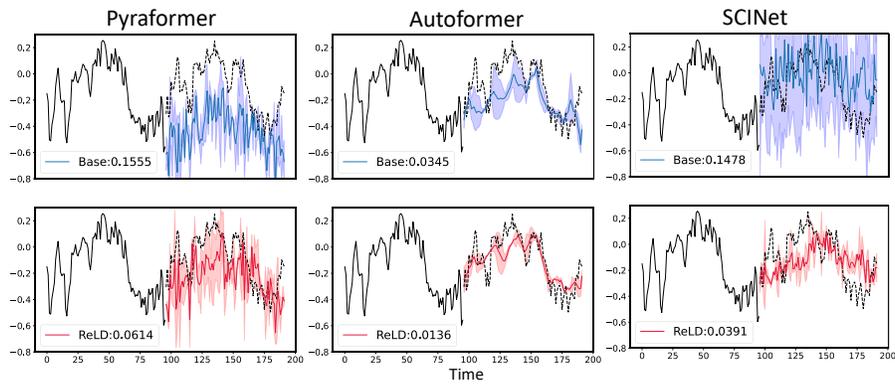


Figure 9: Forecasting results of the recent three models on the same sample in HULL series of ETTm2. The first row shows the forecasting results of the baselines without our ReLD and the second row shows those with our ReLD.

F FULL BENCHMARK ON THE REAL-WORLD DATASETS

Models	FEDformer		Pyraformer		Autoformer		Informer		LogTrans		Reformer		Transformer		LSTNet		LSTMa		TCN		
	base	our	base	our	base	our	base	our	base	our	base	our	base	our	base	our	base	our	base	our	
MSE																					
ETTm1	48	0.3479	0.3483	0.5354	0.4438	0.4160	0.3842	0.5734	0.7247	0.5902	0.7393	0.6409	0.6615	0.5443	0.6272	0.5586	0.6938	0.5809	0.5655	0.4862	
		± 0.0017	± 0.0017	± 0.0157	± 0.0139	± 0.0216	± 0.0181	± 0.0409	± 0.0764	± 0.0473	± 0.0260	± 0.0173	± 0.0767	± 0.0370	± 0.0115	± 0.0069	± 0.0447	± 0.0334	± 0.0287	± 0.0151	
	96	0.3774	0.3743	0.6453	0.5345	0.4422	0.4438	0.9084	0.6584	0.5772	0.8031	0.7306	0.8652	0.7160	0.7605	0.6816	0.8290	0.6613	0.7628	0.6794	
		± 0.0018	± 0.0016	± 0.0584	± 0.0074	± 0.0243	± 0.0143	± 0.0486	± 0.0522	± 0.0307	± 0.0317	± 0.0216	± 0.1242	± 0.0289	± 0.0354	± 0.01048	± 0.0354	± 0.1048	± 0.0551	± 0.0429	± 0.0152
	168	0.4446	0.4367	0.8644	0.7415	0.5042	0.4906	1.3720	1.1125	1.0497	0.9355	0.8858	0.7677	1.0911	0.8628	0.9276	0.7904	1.0522	0.8622	0.9406	0.9319
	± 0.0164	± 0.0153	± 0.0905	± 0.0399	± 0.0515	± 0.0264	± 0.2422	± 0.0813	± 0.1452	± 0.0647	± 0.0259	± 0.0172	± 0.1563	± 0.0823	± 0.0890	± 0.0403	± 0.0564	± 0.0960	± 0.0313	± 0.0327	
336	0.4479	0.4453	0.9328	0.8895	0.5694	0.5110	1.3425	1.1507	1.0618	0.9874	0.9850	0.8481	1.1712	1.0429	1.0125	0.9586	1.1677	0.9672	1.1093	1.1120	
	± 0.0075	± 0.0059	± 0.0341	± 0.0548	± 0.1115	± 0.0990	± 0.0725	± 0.0305	± 0.2052	± 0.1435	± 0.0308	± 0.0130	± 0.0721	± 0.0560	± 0.0738	± 0.0449	± 0.0875	± 0.1548	± 0.0555	± 0.0625	
720	0.5088	0.5064	0.9843	0.9781	0.5348	0.5207	1.3933	1.3912	1.0621	1.1995	1.1994	1.0481	1.1264	1.0834	1.1154	1.0965	1.4609	1.3927	1.1604	1.1484	
	± 0.0131	± 0.0192	± 0.0214	± 0.0196	± 0.0213	± 0.0107	± 0.0893	± 0.0444	± 0.1705	± 0.2349	± 0.0597	± 0.0449	± 0.0765	± 0.0754	± 0.0656	± 0.0391	± 0.1134	± 0.1274	± 0.0653	± 0.0246	
Imp.	-0.71%		-10.75%		-4.57%		-11.13%		-7.17%		-12.44%		-14.13%		-8.62%		-15.28%		-5.33%		
ETTm2	48	0.2539	0.2569	0.7756	0.6186	0.2984	0.2815	0.9560	1.1329	0.6747	0.8518	0.6777	1.1950	0.6794	0.8930	0.6791	0.8816	0.5750	0.9044	0.6387	
		± 0.0004	± 0.0010	± 0.0910	± 0.0369	± 0.0072	± 0.0061	± 0.1512	± 0.0916	± 0.0825	± 0.1191	± 0.0350	± 0.0400	± 0.3054	± 0.0608	± 0.0805	± 0.0832	± 0.0213	± 0.1185	± 0.1442	
	96	0.3420	0.3304	1.6090	1.1733	0.3859	0.3514	3.4000	2.4616	2.7501	2.1171	1.8490	1.3124	2.0744	1.7618	1.4426	1.1167	1.6695	0.9905	1.7404	1.5293
		± 0.0024	± 0.0033	± 0.0866	± 0.2271	± 0.0261	± 0.0124	± 0.5346	± 0.4278	± 0.6731	± 0.3310	± 0.2863	± 0.1231	± 0.1893	± 0.3022	± 0.0179	± 0.0990	± 0.5567	± 0.1973	± 0.4478	± 0.1764
	168	0.4021	0.3837	5.0138	3.9761	0.4394	0.4131	5.7957	4.7229	2.2557	2.2054	3.1301	2.2042	4.0993	3.3118	2.8871	3.9065	2.5133	4.321	2.4058	
	± 0.0096	± 0.0069	± 1.0231	± 0.6641	± 0.0927	± 0.0343	± 0.8701	± 0.5132	± 0.3390	± 0.5046	± 0.3670	± 0.2261	± 0.8010	± 0.8544	± 0.2903	± 0.2880	± 1.1970	± 0.8563	± 0.1782	± 0.1341	
336	0.4006	0.3842	4.3559	3.2814	0.4942	0.4239	3.9011	3.7882	4.4043	3.6032	2.9343	2.2374	2.8095	2.9328	3.9730	3.0537	2.9757	2.1797	2.9652	2.9037	
	± 0.0108	± 0.0071	± 0.3957	± 0.5145	± 0.0307	± 0.0338	± 0.6699	± 0.7034	± 0.5353	± 0.2468	± 0.3673	± 0.1497	± 0.5978	± 0.3583	± 0.3591	± 0.2037	± 0.4939	± 0.3570	± 0.0794	± 0.1516	
720	0.4520	0.4429	4.2013	3.4771	0.4502	0.4209	4.0175	3.9606	2.4391	2.7142	2.6307	2.6276	2.6471	3.0069	3.7246	2.9905	3.2982	2.6797	3.1513	3.1424	
	± 0.0363	± 0.0315	± 0.5338	± 0.2490	± 0.0414	± 0.0232	± 0.5374	± 0.3356	± 0.2176	± 0.5075	± 0.2001	± 0.2424	± 0.5341	± 0.4458	± 0.1502	± 0.3913	± 0.4699	± 0.2820	± 0.0452	± 0.0870	
Imp.	-2.58%		-21.98%		-8.26%		-16.62%		-14.52%		-20.58%		-7.28%		-20.44%		-31.32%		-8.99%		
ETTm1	48	0.5146	0.5190	0.5632	0.5564	0.5487	0.4924	0.6332	0.5984	0.5800	0.5696	0.7703	0.7276	0.6136	0.6415	0.6136	0.6744	0.6478	0.6768	0.6478	
		± 0.0091	± 0.0124	± 0.0285	± 0.0443	± 0.0865	± 0.0329	± 0.0150	± 0.0266	± 0.0225	± 0.0186	± 0.0320	± 0.0077	± 0.0164	± 0.0463	± 0.0256	± 0.0253	± 0.0336	± 0.0620	± 0.0606	
	96	0.3594	0.3580	0.5364	0.4713	0.5239	0.4547	0.6404	0.5429	0.7070	0.5969	0.7773	0.6409	0.5787	0.5157	0.5484	0.5365	0.7054	0.5920	0.6758	0.5936
		± 0.0035	± 0.0033	± 0.0319	± 0.0300	± 0.0851	± 0.0476	± 0.0570	± 0.0368	± 0.0470	± 0.0310	± 0.0698	± 0.0582	± 0.0293	± 0.0672	± 0.0358	± 0.0251	± 0.0793	± 0.0838	± 0.0333	± 0.0259
	168	0.3850	0.3790	0.5631	0.5056	0.5341	0.5001	1.2238	0.7507	0.8866	0.7534	0.8398	0.6889	0.7928	0.7268	0.6323	0.5774	0.8710	0.6477	0.9382	0.9128
	± 0.0259	± 0.0138	± 0.0737	± 0.0346	± 0.0366	± 0.0300	± 0.1911	± 0.0776	± 0.1458	± 0.0856	± 0.0346	± 0.0220	± 0.0628	± 0.0320	± 0.0282	± 0.0338	± 0.0883	± 0.0661	± 0.0412	± 0.0288	
336	0.4033	0.3955	0.6970	0.5729	0.5613	0.5145	1.3896	1.0085	0.9685	0.8368	0.9874	0.8946	0.9579	0.9262	0.7985	0.6858	1.1254	0.6810	1.1476	1.1258	
	± 0.0107	± 0.0090	± 0.0855	± 0.0707	± 0.0217	± 0.0214	± 0.1481	± 0.0877	± 0.0582	± 0.0570	± 0.0519	± 0.0184	± 0.1637	± 0.1313	± 0.1058	± 0.0390	± 0.1963	± 0.0222	± 0.0318	± 0.0166	
720	0.5010	0.4805	0.9037	0.6822	0.5601	0.5278	1.3329	1.0780	1.1222	0.9226	1.1216	1.0027	1.0858	1.0393	0.9248	0.8168	0.9782	0.8280	1.2768	1.2381	
	± 0.0456	± 0.0297	± 0.1019	± 0.0624	± 0.0578	± 0.0301	± 0.1013	± 0.0966	± 0.1009	± 0.0750	± 0.0227	± 0.0421	± 0.1066	± 0.0785	± 0.0516	± 0.0886	± 0.1870	± 0.1288	± 0.2643	± 0.0397	
Imp.	-1.42%		-13.17%		-8.79%		-2.19%		-1.75%		-12.21%		-6.19%		-8.20%		-20.10%		-4.82%		
ETTm2	48	0.1421	0.1386	0.2806	0.2200	0.1708	0.1689	0.2953	0.2143	0.3191	0.2394	0.4635	0.3293	0.3317	0.2923	0.2826	0.2736	0.2494	0.3619	0.2535	
		± 0.0019	± 0.0003	± 0.0215	± 0.0283	± 0.0051	± 0.0023	± 0.0350	± 0.0188	± 0.0620	± 0.0261	± 0.0969	± 0.0322	± 0.1093	± 0.0328	± 0.0174	± 0.0374	± 0.0509	± 0.0255	± 0.0233	
	96	0.1886	0.1844	0.3709	0.2480	0.2934	0.2213	0.4451	0.2860	0.5480	0.3844	0.7433	0.4491	0.4636	0.4430	0.3426	0.3807	0.2798	0.5537	0.3839	0.3839
		± 0.0008	± 0.0012	± 0.1396	± 0.0164	± 0.0691	± 0.0126	± 0.0519	± 0.0221	± 0.0984	± 0.0594	± 0.0529	± 0.0230	± 0.0384	± 0.0845	± 0.0899	± 0.0391	± 0.0377	± 0.0165	± 0.0965	± 0.0607
	168	0.3429	0.2791	0.5655	0.5515	0.3095	0.2838	2.2831	1.4531	1.4953	1.1491	1.2078	0.8359	1.7806	1.3650	0.9504	0.8304	1.1780	0.6010	1.8678	1.9559
	± 0.0240	± 0.0017	± 0.0840	± 0.0586	± 0.0130	± 0.0087	± 0.1434	± 0.1662	± 0.2619	± 0.1603	± 0.1790	± 0.0428	± 0.2038	± 0.0878	± 0.0953	± 0.0878	± 0.2799	± 0.0853	± 0.1214	± 0.0890	
336	0.3381	0.3191	1.6013	1.3296	0.5084	0.3642	2.4795	1.7639	3.4383												

Models	FEDformer		Pyraformer		Autoformer		Informer		LogTrans		Reformer		Transformer		LSTNet		LSTMa		TCN		
	base	our	base	our	base	our	base	our	base	our	base	our	base	our	base	our	base	our	base	our	
Pump	MSE																				
	48	0.5246 ±0.0024	0.5227 ±0.0012	0.7242 ±0.0184	0.7151 ±0.0153	0.5767 ±0.0364	0.5439 ±0.0042	0.8648 ±0.0294	0.7934 ±0.1065	0.7886 ±0.0569	0.7129 ±0.0673	0.8083 ±0.0318	0.7282 ±0.0155	0.8434 ±0.0664	0.6645 ±0.0697	0.8234 ±0.0376	0.7343 ±0.0110	0.7540 ±0.0238	0.7155 ±0.0295	0.8649 ±0.0920	0.8310 ±0.0964
	96	0.5197 ±0.0026	0.5135 ±0.0021	0.8484 ±0.0309	0.7958 ±0.0284	0.5576 ±0.0098	0.5381 ±0.0192	0.8314 ±0.0732	0.8699 ±0.0987	0.9948 ±0.1275	0.9839 ±0.1926	0.8261 ±0.0619	0.7605 ±0.0332	0.9108 ±0.0997	0.8816 ±0.1224	1.0155 ±0.1262	1.0068 ±0.1740	0.8126 ±0.0676	0.7660 ±0.0374	1.0367 ±0.1635	0.9699 ±0.0383
	168	0.5502 ±0.0061	0.5363 ±0.0057	0.8512 ±0.0470	0.8431 ±0.0320	0.5974 ±0.0252	0.5814 ±0.0194	1.7047 ±0.1863	1.5268 ±0.1589	1.2820 ±0.1141	1.3845 ±0.2210	1.0941 ±0.2937	0.8555 ±0.0902	1.1326 ±0.1874	1.1846 ±0.1399	1.3271 ±0.2779	1.2025 ±0.2055	0.9088 ±0.1720	0.8161 ±0.0700	1.1093 ±0.0529	1.0771 ±0.0788
	336	0.5931 ±0.0525	0.5638 ±0.0109	0.9225 ±0.0405	0.9508 ±0.0520	0.6615 ±0.0248	0.6209 ±0.0144	1.6763 ±0.3531	1.4923 ±0.2763	0.9181 ±0.0463	0.9100 ±0.0816	0.9658 ±0.0598	0.9183 ±0.0385	1.1494 ±0.3206	1.1369 ±0.2823	1.6537 ±0.4863	1.2921 ±0.3871	0.9340 ±0.1100	0.8587 ±0.0966	1.5206 ±0.0914	1.2081 ±0.1237
720	0.7230 ±0.2988	0.5802 ±0.0318	1.3697 ±0.2434	1.2829 ±0.1263	0.7066 ±0.0144	0.6191 ±0.0203	1.7041 ±0.2126	1.6986 ±0.0835	0.9746 ±0.0806	0.9082 ±0.0435	1.5309 ±0.3202	1.2181 ±0.0630	1.7408 ±0.3202	1.7408 ±0.3202	1.6076 ±0.3964	1.6421 ±0.1623	1.4645 ±0.4705	1.2439 ±0.5097	2.0748 ±0.4642	1.5455 ±0.1738	
Imp.	-5.75%	-2.34%	-6.08%	-5.07%	-2.08%	-1.44%	-10.57%	-1.44%	-2.08%	-1.44%	-10.57%	-1.44%	-1.44%	-1.44%	-8.16%	-8.83%	-8.83%	-8.83%	-11.87%	-11.87%	
AirQuality	MSE																				
	48	0.7306 ±0.0131	0.7447 ±0.0254	0.8780 ±0.0675	0.8579 ±0.0712	0.8606 ±0.0191	0.8520 ±0.0425	0.9094 ±0.0579	0.8682 ±0.0610	0.9281 ±0.1007	0.8010 ±0.0443	1.1148 ±0.0509	0.9913 ±0.0463	0.9718 ±0.0409	0.8877 ±0.0707	1.0227 ±0.0654	0.9738 ±0.0522	0.9273 ±0.1077	0.8540 ±0.0242	0.8829 ±0.1588	0.8358 ±0.1005
	96	0.8251 ±0.0025	0.8166 ±0.0027	1.1207 ±0.1378	1.1122 ±0.1346	0.9976 ±0.0667	0.9271 ±0.0638	1.3526 ±0.0255	1.1934 ±0.0788	1.0586 ±0.0867	1.0653 ±0.0535	1.2099 ±0.0452	1.1961 ±0.0546	1.1338 ±0.1105	1.0816 ±0.1563	1.1462 ±0.0491	1.1413 ±0.0279	1.1454 ±0.1172	1.0810 ±0.1114	1.0260 ±0.0392	0.9915 ±0.0360
	168	0.8108 ±0.0141	0.8080 ±0.0164	1.1930 ±0.1415	1.1149 ±0.0557	0.9141 ±0.0214	0.9256 ±0.0844	1.7961 ±0.1993	1.5947 ±0.1155	1.8682 ±0.2617	1.9269 ±0.4315	1.4728 ±0.1751	1.3455 ±0.1253	1.8278 ±0.4175	1.7274 ±0.2465	1.2313 ±0.0753	1.1560 ±0.0128	1.6440 ±0.2611	1.3756 ±0.2291	1.2456 ±0.0523	1.1629 ±0.0954
	336	0.8918 ±0.0066	0.8724 ±0.0112	1.2240 ±0.0475	1.2140 ±0.0475	0.9525 ±0.0683	0.9295 ±0.0525	1.7579 ±0.0597	1.7060 ±0.0608	1.6436 ±0.2307	1.4867 ±0.1592	1.4731 ±0.1082	1.3958 ±0.0760	1.4636 ±0.1249	1.4199 ±0.0744	1.3990 ±0.0595	1.3881 ±0.0429	1.3524 ±0.2118	1.2055 ±0.1609	1.3006 ±0.0564	1.2844 ±0.0660
720	0.9974 ±0.0302	0.9527 ±0.0261	2.1962 ±0.1120	1.9817 ±0.1065	1.1018 ±0.0378	1.0593 ±0.0508	2.9136 ±0.1185	2.9846 ±0.2300	3.3683 ±0.3599	2.8386 ±0.5806	1.7234 ±0.0728	1.6715 ±0.0258	2.4630 ±0.1439	2.4953 ±0.2564	1.8259 ±0.0761	1.9214 ±0.0689	2.3104 ±0.2172	2.2325 ±0.1665	1.4419 ±0.0792	1.4256 ±0.0703	
Imp.	-1.22%	-4.04%	-2.62%	-5.61%	-7.04%	-4.09%	-5.82%	-4.09%	-7.04%	-4.09%	-5.82%	-4.09%	-4.09%	-4.09%	-1.37%	-8.82%	-8.82%	-8.82%	-3.54%	-3.54%	
Weather-h	MSE																				
	48	0.3376 ±0.0026	0.3357 ±0.0023	0.2922 ±0.0047	0.2786 ±0.0057	0.3441 ±0.0075	0.3425 ±0.0064	0.3454 ±0.0149	0.2941 ±0.0131	0.3593 ±0.0263	0.3262 ±0.0147	0.3429 ±0.0237	0.3127 ±0.0113	0.3047 ±0.0040	0.2860 ±0.0095	0.3181 ±0.0078	0.3099 ±0.0060	0.3457 ±0.0158	0.3245 ±0.0071	0.3476 ±0.0133	0.3270 ±0.0091
	96	0.4031 ±0.0057	0.3997 ±0.0198	0.3933 ±0.0095	0.3578 ±0.0074	0.4636 ±0.0277	0.4455 ±0.0215	0.4528 ±0.0551	0.4428 ±0.0291	0.4471 ±0.0229	0.3856 ±0.0217	0.5264 ±0.0448	0.4163 ±0.0186	0.4279 ±0.0372	0.3704 ±0.0214	0.4142 ±0.0118	0.3857 ±0.0115	0.4090 ±0.0158	0.3873 ±0.0131	0.4502 ±0.0416	0.4243 ±0.0279
	168	0.4578 ±0.0256	0.4466 ±0.0154	0.4207 ±0.0160	0.3978 ±0.0105	0.5163 ±0.0232	0.4907 ±0.0190	0.5304 ±0.0582	0.4984 ±0.0593	0.4706 ±0.0369	0.4431 ±0.0555	0.6591 ±0.1473	0.6726 ±0.1001	0.5106 ±0.0384	0.4813 ±0.0218	0.4641 ±0.0388	0.4612 ±0.0245	0.4198 ±0.0168	0.4164 ±0.0187	1.0181 ±0.0500	1.0051 ±0.1028
	336	0.5099 ±0.0250	0.5162 ±0.0197	0.4540 ±0.0269	0.4401 ±0.0201	0.6123 ±0.0374	0.5662 ±0.0339	0.5921 ±0.0244	0.5680 ±0.0204	0.5736 ±0.1297	0.5854 ±0.0899	0.8411 ±0.0633	0.7824 ±0.0867	0.5051 ±0.0455	0.4726 ±0.0457	0.4905 ±0.0198	0.4730 ±0.0133	0.4727 ±0.0159	0.4518 ±0.0212	1.1468 ±0.1591	1.2091 ±0.1111
Imp.	-0.65%	-5.55%	-4.21%	-6.79%	-6.69%	-4.21%	-6.79%	-6.79%	-6.69%	-4.21%	-8.66%	-7.94%	-7.94%	-7.94%	-3.41%	-4.17%	-4.17%	-4.17%	-1.88%	-1.88%	

