MOAT: Motif-guided Debiasing Framework for Time Series Forecasting

Li Zhang[1], **Yifeng Gao**[1] , **Mucun Sun**[2], **Shuochao Yao**[3], **Ashley Gomez**[1], **Jessica Lin** [3]

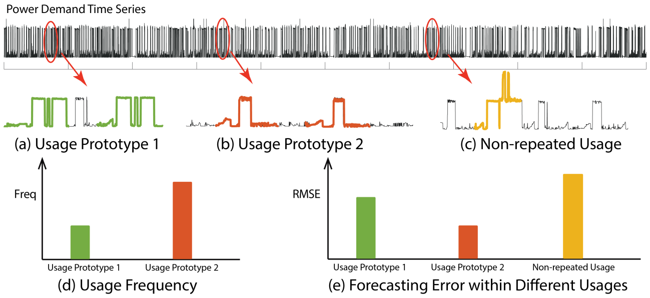
[1]University of Texas Rio Grande Valley
[2]Idaho National Lab
[3]George Mason University
{li.zhang,yifeng.gao, asley.gomez06}@utrgv, mucun.sun@inl.gov, {shuochao,jessica}@gmu.edu

Abstract

Deep learning models have achieved remarkable progress in recent years in the task of long-sequence time series forecasting and have the potential to make a transformative impact on critical applications such as renewable energy and disaster response. However, current deep forecasting methods mostly rely on average loss – which often excels in average performance and fails to deliver reliable predictions in critical regions—periods characterized by less frequent occurrences, yet important and difficult pattern regions. Such discrepancy in performance greatly hurts the reliability and usability of deep learning models in the real world. To address this problem, we propose we propose a new training framework incorporates 1) a new motif-based loss function to debias forecasting model 2) a special-designed robust training framework to adaptively adjust the gradient of poor performing motif region in training and avoid overfitting. The experiment shows that our training framework can mitigate the poor performance region bias on multiple deep forecasting baseline methods compared to the vanilla models and has compatible performance on average loss.

Introduction

Deep learning models have recently achieved notable success [42, 37] in tackling the task of Time series Long-Sequence Forecasting (TLSF), which involves predicting a long sequence of future values based on historical data. Despite the great success, most existing TLSF models are predominantly assessed based on their average performance across all instances by treating all instances equally. However, in practice, not all regions or occurrences are treated equally—systematic failures to predict certain critical event-associated patterns can lead to significant costs, severely impacting the practical utility of these models in real-world applications. Winter storm Uri in 2021 placed immense energy pressure on Texas [40], largely due to the model's failure to predict the atypical power patterns caused by snows.



(a) Usage Prototype 1 (b) Usage Prototype 2 (c) Non-repeated Usage

(d) Usage Frequency (e) Forecasting Error within Different Usages

Figure 1: While forecasting model achieve better result in majority usage pattern, model may not perform well in less frequent pattern. **(a-c)**: Two usage prototypes and one type of non-repeated usage in a power demand time series. **(d)**: occurrence frequency of usage prototype 1 and 2, respectively. **(e)**: Forecasting errors within three shown cases

To demonstrate the discrepancy performance across different underlying events, Figure 1.top shows a dishwasher power demand time series which is well known for consisting of heterogeneous types of power usage cases[10, 13, 22, 2]. Figure 1(a-b) shows two different power usage patterns in this time series, whereas usage prototype 2 (red in (b)) is more common, and usage prototype 1 (green in (a)) has a more complex shape and occurs less frequently (Figure 1(d)). As shown in Figure 1 (e), the per-prototype region RMSE performance has a significant difference. RMSE associated with Usage Prototype 1 prediction is almost as twice as Prototype 2. Mitigating the shown discrepancy has three unique challenges. First, event existed in a sub-region of the forecasting span and typically unrelated to forecasting task and potentially have arbitrary mechanism. Second, the event knowledge will not be available at the forecasting time, when event is not fully developed. Third, the empirically training dynamic of RMSE is significant from cross-entropy based loss. As a result, most existing studies on performance discrepancy on classification tasks [30, 23] cannot be used in this problem. Other work on performance discrepancy on extreme values [6] or distribution shifts on seasonal trends are also hard to apply to this problem since only a small portion of important events are extreme values or seasonal trends shifting, and we aim to focus on a general case of time series events on the semantic level.

To address the challenges above, we propose we propose a new training framework named **MO**tif Debi**A**s **T**raining framework (**MOAT**), which incorporates 1) a new motif-based loss function debiasing forecasting model 2) a special-designed robust training framework to adaptively adjust the gradient of poor performing motif region in training and avoid overfitting. Time series motif is widely used to identify meaningful similar patterns in groups, known as motifs, in large-scale time series in a wide range of domain applications [21, 38, 9]. Our framework first utilizes time series motifs to identify similar pattern regions, then uses them to guide the model training phase. Once the training is done, it does not require extra motif knowledge in the testing phase. The experiment shows that our training framework can mitigate the poor performance region bias on multiple deep forecasting baseline methods compared to the vanilla models under the RMSE-based loss, with compatible performance on average.

Related Work

Time Series Forecasting

Deep neural networks in recent studies [28, 29, 31, 26] have outperformed traditional statistical models such as ARIMA [3] and achieved great success in time series forecasting. Recently, leveraging the popular Transformer model from text machine translation [34] structure, Zhou et al. [42] proposed a sparse Transformer forecasting model that achieves superior performance on TSLF task. Several other work [17, 37] focus on improving the self-attention modules to improve memory requirement and efficiency. Recent work based on MLP or transformer [24, 18, 39, 43] have boosted performance, but mostly focus on improving the model structure and all of them are using the average loss of RMSE. By taking a model-agnostic approach by proposing a new training framework, our framework could potentially be used to improve on top of existing work to mitigate the weakness of their forecasting performance.

Model Debiasing

To the best of our knowledge, none of the existing work discusses pattern-level discrepancy on time series forecasting. Most of the existing work focus on proposed solutions for extreme-value [6][7][8] to improve the performance on the extreme point and block-wise maximum values, which is a special case for extreme events. Instead, our training framework focuses on more general rare events associated with long-sequence forecasting problems and aims to perform well across different types of events ahead of time.

Although numerous recent research show that over-parameterized deep learning models could consistently fail on minority and atypical examples [4, 25, 16, 30], they are mostly on image domain tasks such as image classification and object detection. Thus, they often rely on some image-specific features such as grey level or background scene to extract unbiased features and train the model [35, 1, 5], and thus are not designed to handle rolling-based time series forecasting. Pezeshki [27] proposed a gradient regularization approach to penalize the gradient of the imbalanced samples on classification models but could result in overfitting when coming across noise in data.

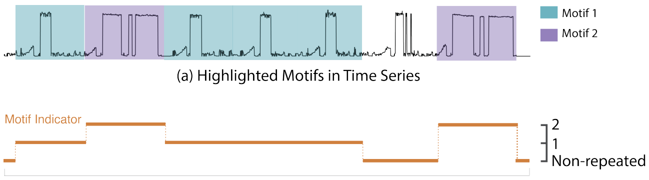
Time Series Motif

The time series motif is one of the most widely used primitives in time series data mining. Motifs can detect the insightful pattern-level structure of the time series. Numerous previous works have shown that such detected information can advance our understanding of the hidden mechanism behind time series datasets and have been widely used in various data mining tasks such as classification, clustering, rule discovery, and anomaly detection [15, 32, 41, 33, 13, 19, 20, 12, 9, 11, 10, 21]. Despite the advantage, motif has not been used to evaluate or debiasing time series forecasting models. To the best of our knowledge, this is the first work making connection of time series motif and TSLF problem to debiasing on deep forecasting models based on motif.

Notations and Background

In this section, we introduce the background and definitions related to our problems.

Time Series Motif



(a) Highlighted Motifs in Time Series

(b) Motif Indicator Series

Figure 2: Example of Time series with Motif Indicator.

Time series motifs [14] have been used to identify meaningful non-overlapping sets of segments. Instances within each motif set are similar, and instances belonging to different motif sets are dissimilar. Fig. 2 (a) illustrates a time series that contains two motifs (highlighted in blue and purple respectively). In this work, we follow the set motif definition [14] to form motif indicators for our training framework (For detailed implementation, please see supplement material).

Motif Indicator series: A global Motif Indicator Series $M = m_{i}{}_{i=1}^{n}$ is a meta time series of length n. Where I_i stores the motif indicator that covers the time stamp i. Specifically, $M_i = 0$ indicates non-repeat region. Fig. 2(b) shows a the corresponding motif indicator series for the time series in Fig. 2(a). A motif indicator M will be a vector consisting of $\{0, 1, 2\}$ denoting non-motif region, Motif 1 and 2 respectively.

Since $\{\mathcal{Y}\}$ is defined on segment level. So, we defined segment-level motif indicator $\mathcal{M} \in N^{|\mathcal{Y}| \times L_{pred}}$ as the motif Indicator segments extracted from sliding window of length L_{pred} (i.e. $M_i = [m_{i+1} \cdots m_{i+L_{pred}-1}] \in \mathcal{M}$).

Note that a time series motif discovery algorithms [44, 38, 21] can easily identify such motifs. We omit the discussion on how to obtain the motif information since this is an orthogonal problem beyond the scope of this paper.

Problem Setting

We introduce problem setting: Given time series $T = \{t_i\}_{i=1}^{n}$ and two fixed-length sliding window L_{input} and L_{pred}, the task aims to learn a model with parameter $f_\Theta : \mathcal{X} \rightarrow \mathcal{Y}$ which maps every subsequence of length L_{input}: $X_i = [t_{i-L_{in}+1}, \cdots, t_i] \in \mathcal{X}$ to its next subsequence of

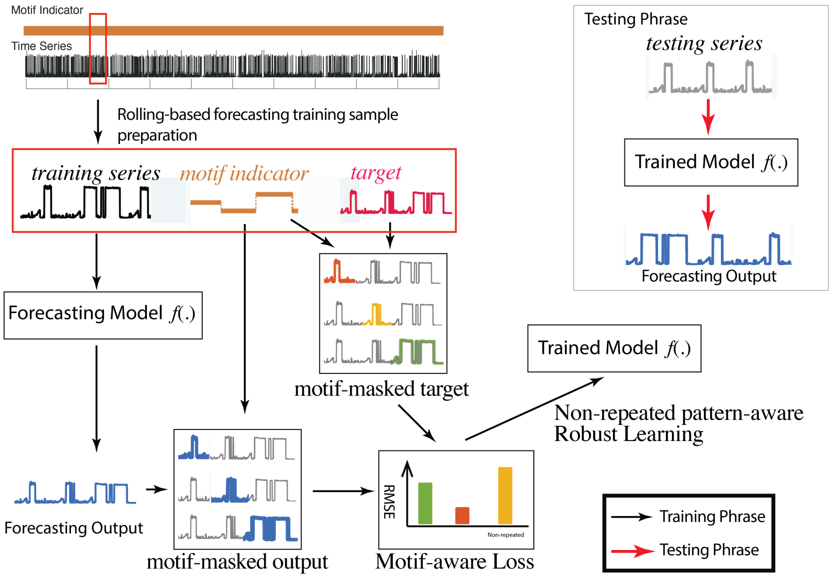


Figure 3: Overall Proposed Framework - MOAT

length L_{pred}: $Y_i = [t_{i+1}, \cdots, t_{i+L_{pred}-1}] \in \mathcal{Y}$ (i.e. $Y_i = f(X_i)$) in a rolling forcasting fashion. During the training and testing phases, each pair of (X_i, Y_i) is called a *input sequence* and a *target sequence*, respectively.

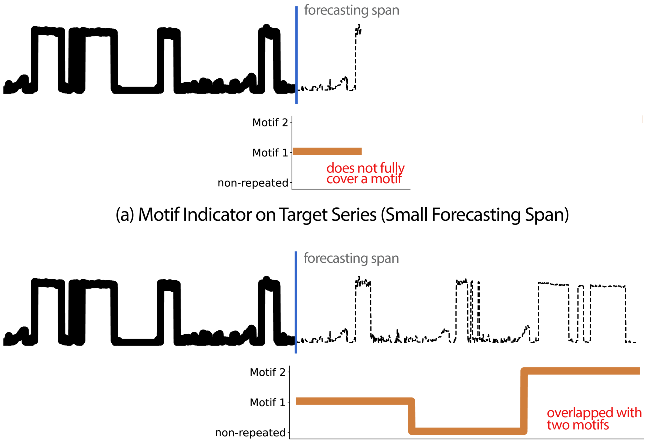
In addition, we consider the time series T contains multiple types of latent events $E = \{E_1, E_2, ..., E_k\}$ associated with different types of shapes of patterns. When we obtained the prediction, we care more about the performance of forecasting associated with any latent potential type of events E_j (i.e. $Y_i \cap E_j$). However, we don't know E_j at the time of predicting nor have the knowledge of the location of events E_j in testing data T_{test}. We aim to answer the following research questions:

[RQ1] How can we optimize and evaluate the performance of motifs region while allowing forecasting sequence to be arbitrary length?

[RQ2] How can we mitigate the poor performance region bias while maintaining or exceeding the overall performance across all motif groups?

Proposed Method

The overall framework of proposed MOAT training framework is shown in Figure 3. Instead of optimizing the loss computed via average performance, we propose to distributedly optimize per-motif group loss and reduce the potential performance disparity across different motifs. Specifically, given training set $\{\mathcal{X}, \mathcal{Y}, \mathcal{M}\}$, the framework first convert the target sequences Y_i and model forecasting $\hat{Y}_i = f(X)$ to masked series Y_i^m and $\hat{Y}_i^m$ via the motif indicator M_i. Then motif-aware forecasting loss that contains per-motif-group loss along with the non-repeat region is computed by aggregated forecasting errors in different motif regions. Finally, the model is trained on a proposed non-repeat pattern-aware distributed robust learning framework. Note that in the proposed framework only assume that the motifs information is available in the training data. The model $f(X)$ does not rely on motif indicator to perform forecasting. So the framework can adapt to any existing forecasting models.



(a) Motif Indicator on Target Series (Small Forecasting Span)

(b) Motif Indicator on Target Series (Long Forecasting Span)

Figure 4: Target series may contain a partial motif or multiple motif instances.

Motif-guided Masking

To address **RQ1**, we propose to compute motif-wise loss value with respect to a given motif region, which requires modification of classical L_{RMSE} loss used in the TLSF work.

Figure 4(a-b) illustrates two different scenarios of target sequences along with their motif indicator vectors. We can observe that when the forecasting span is small, the target sequence may only contains a part of a motif 1. On the other hand, when the forecasting span is large, the target sequence may contain more than one motif instances. In this example (Figure 4(b)), the target sequence across the region contains motif 1, 2, and non-repeated region. This observation is widespread because the pattern existed in the data is independent with the aimed forecasting span. It cause a unique challenge in assigning the correct motif type for each forecasting sample (X_i, Y_i) — all the data are not purely belong to one motif.

Based on the observation above, we define motif guided masking series operation by:

$$Mask(Y, M, k) = Y \bigoplus 1\{M = k\} \tag{1}$$

where $\bigoplus$ indicates element-wise addition. $1\{M = k\}$ is a indicator function which return 1 if $m_i = k$, otherwise, return 0. We denote the masked sequence set as: $\mathcal{Y}^M \in R^{|\mathcal{Y}| \times K \times L_{pred}}$ where:

$$Y^{(m)}_{i,k} = Mask(Y_i, M_i, k) \tag{2}$$

and K means the number of motifs. Similarly, given the model forecasting result $\hat{Y}_i = f(X_i)$, we compute the motif-guided masking forecasting output:

$$\hat{Y}^{(m)}_{i,k} = Mask(\hat{Y}_i, M_i, k) \tag{3}$$

where $\hat{Y}^{(m)}_{i,k} \in \hat{\mathcal{Y}}^M$. In the next step, $Y^{(m)}$ and $\hat{Y}^{(m)}$ are used to compute the motif-aware loss functions that used to train the model.

Motif-Aware Loss (MAL)

To address **RQ2**, we proposed to optimize model parameters with distributed robust optimization under per-motif group losses to avoid performance disparity between each motif region. We first introduce the proposed motif-aware loss $L^{\mathcal{M}}$ as a set of loss functioned defined as:

$$L_{M_i} = \sum_i ||Y^{(m)}_{i,k} - \hat{Y}^{(m)}_{i,k}||^2_2. \tag{4}$$

In the proposed optimization method. We aims to distributed optimizing the largest L_{M_i} (corresponding to the worst forecasting performance on a motif region).

Non-repeated Pattern-aware Distributed Robust Learning

The proposed distributed robust learning contains two objective functions.

First, given per-motif region loss set $\mathcal{L}_M = \{L_{M_i}\}^K_{i=1}$, we aim to minimize the *worst motif loss*:

$$L_{worst} = \max_{L_{M_k} \in \mathcal{L}_M} L_{M_k}(X, Y, \theta) \tag{5}$$

Second, we also aim to maintain an overall group-wise performance that includes the non-motif region loss (denoted by L_{M_0}). Because L_{M_0} is often noisy and more unpredictable than regular motif region, which will greatly impact the performance if take L_{M_0} into consideration in Eq. 4.5.

Therefore, the overall objective loss is defined as:

$$\min_\theta \alpha \underbrace{L_{worst}(X, Y, \theta)}_{worst \quad motif \quad region} + (1 - \alpha) \underbrace{L_{M_0}(X, Y, \theta)}_{non-motif \quad region} , \tag{6}$$

where α is hyper-parameter that determines the importance of non-motif region.

Note that the proposed problem setting utilizes the motif information only during training and that the training process is *model agnostic*. All existing deep-learning forecasting models can be trained in this framework to reduce the motif prediction bias without any modification, and such motif information is no longer needed during the testing phase. In the rest of this section, we will introduce the optimization approach and the proposed Motif-aware loss in the rest of the subsections.

Relaxed Min-Max Problem from Eq. 4.6

Directly optimizing this min-max problem in Eq.4.6 will result in an unstable gradient descent process [30]. Therefore, we transform Eq. 4.8 to a relaxed version:

$$\min_{\theta \in \Theta} \left[\max_w \sum_{i=1}^{M} \alpha w_i L_{M_i}(X, Y, \Theta)] \right] + (1-\alpha) L_{M_0}(X, Y, \Theta). \tag{7}$$

where $w = [w_1, w_2, \cdots, w_M]$ is the weight vector the motif sets, and $\sum_{i=1}^{M} w_i = 1$ where M is total number of motifs we considered. We convert the proposed objective function to a saddle point optimization loss function.

Without loss of generality, we modify $w' = \alpha w$ and thus we have:

$$\min_{\theta \in \Theta} \left[\max_{w'} \sum_{i=1}^{k} w'_i L_{M_i}(X, Y, \Theta)] \right] + (1-\alpha) L_{M_0}(X, Y, \Theta). \tag{8}$$

where $\sum_{i=1}^{k} w'_i = \alpha$.

Optimizating Eq. 4.8

The overall algorithm is illustrated in Algorithm 1. Intuitively, the algorithm performs a modified min-max optimization. The algorithm first initializes model parameter θ and mixture weight parameter w (Line 3-5). Then it uses exponential gradient ascent to update the mixture weights to maximize the loss (Line 10), and then it fixes w to perform the gradient descent on model parameters θ (Line 16).

Adaptively Adjustment of α

Empirically, we observe that the non-motif region potentially can be hard to predict. As a result, wrongly assigned an α could significantly impact the performance and lead to over-fitting by paying too much attention to the non-motif region. To address this issue, we further proposed a loss value-driven adaptive α adjustment strategy to reduce the weight of α.

Concretely, instead of use a fixed α, we allow adaptive α' value to be equal to:

Algorithm 1: Proposed Algorithm

```
Input: $X$, $Y$, $M$, step size $\eta_w$, $\eta_\theta$
Output: $\Theta$
$\Theta^{(0)} = \mathbf{0}$
blue /* initialize $w$ and $\Theta$ */
$w^{(0)} = init()$;$\Theta^{(0)} = init()$
for $k = 1, \cdots, iter\_max$ do
    for $j = 0, \cdots, K$ do
        $x, y, p = extractMotifData(X, Y, j)$
        blue /* Gradient accent to update mixture weights
        $w$ max-problem in Eqn 8*/
        $w^{(k)} = w^{(k-1)} \exp(\eta_w L_{M_k}(x, y, \Theta^{(k-1)}))$
    end for
    blue /* update $\alpha$ */
    Update $\alpha'$ based on Eq. 6
    blue /* normalize $w^{(k)}$ */
    for $j = 0, \cdots, K$ do
        $w^{(k)} = \frac{w^{(k)}}{\sum_i^K w^{(i)}}$
    end for
    blue /* Update $\Theta$ through Gradient descent*/
    $\Theta^{(k)} = \Theta^{(k-1)} - \eta_\Theta w_i^{(k)} \nabla L_{M_g}(x, y, \Theta^{(k-1)}))$
end for
return $\Theta$
```

$$\alpha' = \min(\alpha, \frac{L_{M_0}(X, Y, \Theta)}{\sum_{i=1}^{M} L_{M_i}(X, Y, \Theta) + L_{M_0}(X, Y, \Theta)}) \quad (9)$$

Intuitively, the α is treated as the maximal weight that the non-motif region instead of a fixed threshold, which could make the parameter easy to tune.

Experimental Evaluation

In this section, we demonstrate that the proposed loss can learn an unbiased model with respect to the patterns while maintaining similar average performance on both real-world and synthetic data. We use Multi-layer Perception, Informer[42], and FEDFormer [43] as the backbone models to test performance against different losses on two real-world datasets ETTH and Italian Power Demand, as well as two pattern-based synthetic datasets.

Performance Evaluation

We evaluate the performance of our proposed training framework over the data based on the following criterion:

- **C1** The average performance should be comparable with using vanilla model under RMSE.
- **C2** The poor performance pattern region should be improved compare with the vanilla model under RMSE.

To evaluate **C1**, measure the utility (Eq. 10) and to evaluate **C2**, we measure the bias metric (Eq. 11).

$$utility = E_{M_i \in \mathcal{M}}[L_M(X, Y, \theta)]. \quad (10)$$

$$bias = \max_{L_{M_k} \in \mathcal{L}_M} L_{M_k}(X, Y, \theta) \quad (11)$$

Baselines

To the best of our knowledge, this is the first work to point out and evaluate the model forecasting bias against the occurrence of motifs in time series. We compared with Vanilla RMSE based solution and Spectrum Decoupling[27], a regularization-based debasing approach:

- **Vanilla Solution**: The original designed deep learning model which trained by minimizing the expectation of the MSE over the entire data distribution. We applied it in all the backbone networks.
- **Spectrum Decoupling** [27]: A recently proposed gradient regularization-based approach. We applied the proposed regularization in the backbone networks.

For all the experiments, we tested our proposed approach with three backbone deep neural networks: 1) **MLP**: the classical multi-layer perception, 2) **Informer** [42], a well-known state-of-the-art long sequence forecasting model. 3) **FEDformer** [43]: a recent proposed frequency-based long sequence forecasting model. We select the wavelet version due to its versatility. Unless otherwise stated, the parameter α in the model is set to be $\frac{1}{k+1}$, where k is the number of motifs.

In each experiment, following the experiment conducted in [43], we repeat the experiment three times. For each dataset, we perform three forecasting tasks with different forecasting lengths. For ETTh2 data, we evaluate the forecasting task on three different horizons: one day, two days, and three days. For the power demand, we evaluate the forecasting task on three different horizons: half a day, one and a half days, and two days. For both TwoPattern datasets, we evaluate the performance on forecasting 64, 96, and 128 sample points.

Comparison Experimental Results

The performance of the proposed method against all compared baselines is shown in Table 1. According to the bias performance evaluated based on bias metric, our method wins 34 out of 36 comparison experiment tests. At the same time, the proposed method achieves similar or slightly higher utility. This may be due to its ability to avoid fitting the noise region that is highly likely located in the non-motif region. Especially in the two TwoPattern datasets, where the data contains large sections of unpredictable data, we find that the performance using patterns is much better than Vanilla. The critical difference diagram for debias and utility performance is illustrated in Figure 5 and Figure 6 respectively. According to Figure 5, the proposed method has significant better debiasing performance against Vanilla and SD, whereas SD has similar performance as the Vanilla version. According to Figure 6, the proposed method has similar utility performance compared with Vanilla approach but is significant better than SD. Overall, the result shows that the proposed model has competitive performance against both baselines.

Ablation Testing: without $L_{M_0}(X, Y, \Theta)$

We next evaluate the impact of the parameter in the proposed algorithm: α. We first compared the proposed algo-

Table 1: Experiment Result

Dataset	Length	Method	Base Models											
2*			MLP				FEDFormer				Informer			
			bias ↓	utility ↓	improved bias ↑ (%)	improved utility ↑ (%)	bias ↓	utility ↓	improved bias ↑ (%)	improved utility ↑ (%)	bias ↓	utility ↓	improved bias ↑ (%)	improved utility ↑ (%)
9*ETTH2	L=24	Proposed	**0.07869**	0.09881	11.73	-1.85	**0.25245**	0.23163	3.23	0.02	**0.0829**	0.095	44.73	0
		Vanila	0.08915	0.09702	0	0	0.26088	0.23167	0	0	0.15	0.095	0	0
		SD	0.15123	0.15218	-69.64	-56.85	0.28194	0.24196	-8.07	-4.44	0.15123	0.15	-0.82	-57.89
	L=48	Proposed	**0.11799**	0.13251	7.18	-0.52	**0.27215**	0.24355	4.18	2.21	**0.13325**	0.15115	8.73	3.11
		Vanila	0.12712	0.13182	0	0	0.28402	0.24906	0	0	0.146	0.156	0	0
		SD	0.24416	0.236	-92.07	-79.03	0.30232	0.2565	-6.44	-2.99	0.248	0.247	-69.86	-58.33
	L=96	Proposed	**0.18401**	0.18008	4.41	-1.07	**0.28141**	0.25179	3.4	1.68	**0.2045**	0.211	9.91	3.21
		Vanila	0.1925	0.17817	0	0	0.29131	0.25609	0	0	0.227	0.218	0	0
		SD	0.41965	0.39465	-118	-121.5	0.31952	0.26969	-9.68	-5.31	0.361	0.344	-59.03	-57.8
9*PowerDemand	L=48	Proposed	**0.238**	0.138	11.85	1.43	0.89763	0.75374	-2.1	-2.81	**0.27**	0.188	20.12	1.57
		Vanila	0.27	0.14	0	0	**0.87914**	0.73316	8.11	11.1	0.338	0.191	0	0
		SD	**0.238**	0.2	11.85	-42.86	0.95669	0.82469	-16.87	-3.09	0.254	0.178	24.85	6.81
	L=96	Proposed	**0.282**	0.169	6	-7.64	**0.8186**	0.8	11.02	3.61	**0.3885**	0.234	5.01	0.85
		Vanila	0.3	0.157	0	0	0.92	0.83	0	0	0.409	0.236	0	0
		SD	0.32	0.299	-6.67	-90.45	1.13	0.933	-22.83	-12.41	0.399	0.239	2.44	-1.27
	L=192	Proposed	0.386	0.22	-2.66	-10	**0.942**	0.86	11.13	3.37	**0.399**	0.235	7.21	0.84
		Vanila	**0.376**	0.2	0	0	1.06	0.89	0	0	0.43	0.237	0	0
		SD	0.6	0.5	-59.57	-150	1.12	0.916	-5.66	-2.92	1.06303	0.356	-147.22	-50.21
9*TwoPattern10	L=64	Proposed	**0.67733**	1.30872	48.73	15.83	**0.95918**	1.79206	11.93	0.6	**0.877**	1.002	36.22	35.56
		Vanila	1.32118	1.55485	0	0	1.08915	1.80285	0	0	1.375	1.55485	0	0
		SD	0.99382	1.35392	24.78	12.92	0.97622	1.60533	10.37	10.96	1.03	1.35392	25.09	12.92
	L=96	Proposed	**0.96985**	1.31167	25.08	9.27	**0.92499**	1.77354	8.68	-7.07	**0.77**	1.04	40.95	28.07
		Vanila	1.29453	1.44576	0	0	1.01292	1.65646	0	0	1.304	1.44576	0	0
		SD	1.06303	1.2784	17.88	11.58	0.99334	1.45744	1.93	12.01	1.07	1.2784	17.94	11.58
	L=128	Proposed	1.08828	1.26564	13.92	5.72	**0.891**	1.46	7.23	1.13	**0.905**	1.025	25.82	0.49
		Vanila	1.26432	1.34246	0	0	0.96044	1.47676	0	0	1.22	1.03	0	0
		SD	**1.04834**	1.181	17.08	12.03	0.97628	1.31255	-1.65	11.12	1.13	1.181	7.38	-14.66
9*TwoPattern5	L=64	Proposed	**0.42666**	1.35032	38.29	9.1	**0.574**	1.792	28.23	-1.79	**0.7675**	0.8475	32.82	26.94
		Vanila	0.6914	1.48547	0	0	0.79976	1.76046	0	0	1.1425	1.16	0	0
		SD	0.65901	1.37387	4.69	7.51	0.70245	1.61965	12.17	8	0.806	1.25	29.45	-7.76
	L=96	Proposed	**0.61961**	1.38122	20.06	6.75	**0.594**	1.80337	15.95	-6.34	**0.74525**	0.9745	34.48	26.45
		Vanila	0.77508	1.48117	0	0	0.70673	1.69583	0	0	1.1375	1.325	0	0
		SD	0.7792	1.34203	-0.53	9.39	0.71594	1.72	-1.3	-1.42	1.03	1.31	9.45	1.13
	L=128	Proposed	**0.71703**	1.3818	8.75	3.1	**0.706**	1.73	-5.22	-1.99	**0.9575**	1.0275	14.7	25.81
		Vanila	0.7858	1.42606	0	0	0.671	1.69631	0	0	1.1225	1.385	0	0
		SD	0.87153	1.3038	-10.91	8.57	0.76611	1.56247	-14.18	7.89	1.08	1.386	3.79	-0.07

Table 2: # of Win vs. Baselines

Method / Backbone	MLP	FEDFormer	Informer
Proposed	**10**	**11**	**12**
Vanila	1	1	0
SD	2	0	0

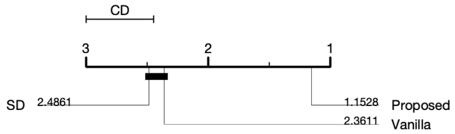


Figure 5: Bias Critical Difference Diagram.

rithm without $L_{M_0}(X, Y, \Theta)$ term. In this case, the algorithm downgrades to classical group distributed robust optimization objective function. The comparsion result in all tested data is illustrated in Table 2. From the table, the proposed approach achieve better worst pattern loss L_{Worst} in 11 out of 12 testing cases with only one testing case with slightly worse performance (0.306 vs. 0.297). We further conduct Wilcoxon signed-rank test [36] on the reported L_{Worst} result. The p-value compared with vanilla RMSE training and our loss without Alpha is 4.88×10^{-4} and 0.0034 respectively. Both values are smaller than 0.05 significant level. The result indicates that our proposed method is significantly better than our method without Alpha.

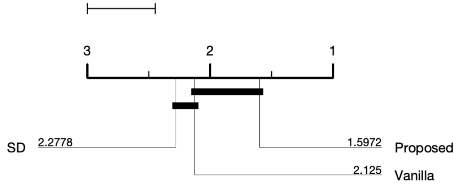


Figure 6: Utility Critical Difference Diagram.

Table 3: Ablation Test: w/o α

3*Dataset	3*Tested Length	3*Method	2*Base Model: Informer	
			bias	utility
6*Etth2	2*L=24	Proposed	**0.0829**	**0.095**
		w/o alpha	0.10125	0.1035
	2*L=48	Proposed	**0.13325**	0.15115
		w/o alpha	0.14	**0.14925**
	2*L=96	Proposed	**0.2045**	**0.211**
		w/o alpha	0.209	0.217
6*PowerDemand	2*L=48	Proposed	0.3065	**0.188**
		w/o alpha	**0.297**	0.194
	2*L=96	Proposed	**0.3885**	**0.234**
		w/o alpha	0.3927	0.239
	2*L=192	Proposed	**0.399**	**0.235**
		w/o alpha	0.4225	0.243
6*TwoPattern5	2*L=64	Proposed	**0.7675**	**0.8475**
		w/o alpha	0.855	0.96
	2*L=96	Proposed	**0.74525**	**0.9745**
		w/o alpha	0.942	1.12
	2*L=128	Proposed	**0.9575**	**1.0275**
		w/o alpha	1.05	1.18
6*TwoPattern10	2*L=64	Proposed	**0.877**	1.0025
		w/o alpha	0.897	**1.02**
	2*L=96	Proposed	**0.77**	1.04
		w/o alpha	0.99375	**1.01**
	2*L=128	Proposed	**0.9025**	**1.025**
		w/o alpha	1.015	1.048
# of Win			**11 out of 12**	**9 out of 12**

Parameter Testing: α

Finally, we next evaluate the impact of α vs. the performance. We tested different α from 0.1 to 1.0 in the TwoPattern5 time series to show the impact of the parameter. The result is shown in Fig. 7. When α is close to 1, the objective function will not have any constraint and the algorithm is similar to the ablation testing case without alpha. When

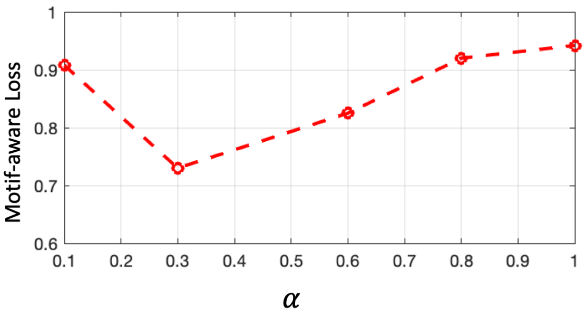


Figure 7: Choosing Different α. (A smaller motif-aware loss means a better performance).

α is too small close to 0.1, the algorithm, however, tends to be close to using RMSE on non-pattern regions, which can harm the overall performance. Overall, we suggest choosing α from 0.2 to 0.6.

Conclusion

In this paper, we take the first step to investigate the model bias issue with respect to patterns. We found that systematic bias could cause serious problems in practice and limit the usability of the forecasting models. To mitigate the bias, we propose a novel loss function named cumulative pattern Loss, which aggregates loss over a set of groups of motifs available in training data, and an optimization strategy to optimize the worst cumulative motif loss. It allows us to optimize the worst-motif loss while maintaining the overall performance. Our training framework is model agnostic, which means any deep TLSF model could be trained on the framework without any modification and reduce the model bias. We test the proposed methods on both real-world and simulated data and demonstrate that our model can significantly reduce the model bias.

Acknowledgment

This work is supported by the NSF under Grant CNS-2431514. The computational resource is supported by Google Cloud Platform (GCP) credit.

References

[1] Alvi, M.; Zisserman, A.; and Nellåker, C. 2018. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 0–0.

[2] Begum, N.; and Keogh, E. 2014. Rare time series motif discovery from unbounded streams. *Proceedings of the VLDB Endowment*, 8(2): 149–160.

[3] Box, G. E.; Jenkins, G. M.; Reinsel, G. C.; and Ljung, G. M. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

[4] Buolamwini, J.; and Gebru, T. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, 77–91. PMLR.

[5] Choi, J.; Gao, C.; Messou, J. C.; and Huang, J.-B. 2019. Why can't i dance in the mall? learning to mitigate scene bias in action recognition. *Advances in Neural Information Processing Systems*, 32.

[6] Ding, D.; Zhang, M.; Pan, X.; Yang, M.; and He, X. 2019. Modeling extreme events in time series prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1114–1122.

[7] Galib, A. H.; McDonald, A.; Wilson, T.; Luo, L.; and Tan, P.-N. 2022. Deepextrema: A deep learning approach for forecasting block maxima in time series data. *arXiv preprint arXiv:2205.02441*.

[8] Galib, A. H.; Tan, P.-N.; and Luo, L. 2023. SimEXT: Self-supervised Representation Learning for Extreme Values in Time Series. In *2023 IEEE International Conference on Data Mining (ICDM)*, 1031–1036. IEEE.

[9] Gao, Y.; and Lin, J. 2017. Efficient discovery of time series motifs with large length range in million scale time series. In *2017 IEEE International Conference on Data Mining (ICDM)*, 1213–1222. IEEE.

[10] Gao, Y.; and Lin, J. 2018. Exploring variable-length time series motifs in one hundred million length scale. *Data Mining and Knowledge Discovery*, 32(5): 1200–1228.

[11] Gao, Y.; and Lin, J. 2019. Discovering subdimensional motifs of different lengths in large-scale multivariate time series. In *2019 IEEE International Conference on Data Mining (ICDM)*, 220–229. IEEE.

[12] Gao, Y.; and Lin, J. 2019. HIME: discovering variable-length motifs in large-scale time series. *Knowledge and Information Systems*, 61(1): 513–542.

[13] Gao, Y.; Lin, J.; and Brif, C. 2020. Ensemble Grammar Induction For Detecting Anomalies in Time Series. *arXiv preprint arXiv:2001.11102*.

[14] Keogh, E.; and Lin, J. 2005. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems*, 8: 154–177.

[15] Keogh, E.; Lin, J.; and Fu, A. 2005. Hot sax: Efficiently finding the most unusual time series subsequence. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, 8–pp. Ieee.

[16] Kilbertus, N.; Rojas Carulla, M.; Parascandolo, G.; Hardt, M.; Janzing, D.; and Schölkopf, B. 2017. Avoiding discrimination through causal reasoning. *Advances in neural information processing systems*, 30.

[17] Li, S.; Jin, X.; Xuan, Y.; Zhou, X.; Chen, W.; Wang, Y.-X.; and Yan, X. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in Neural Information Processing Systems*, 5244–5254.

[18] Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

[19] Mueen, A. 2014. Time series motif discovery: dimensions and applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(2): 152–159.

[20] Mueen, A.; and Chavoshi, N. 2015. Enumeration of time series motifs of all lengths. *Knowledge and Information Systems*, 45(1): 105–132.

[21] Mueen, A.; Keogh, E. J.; Zhu, Q.; Cash, S.; and Westover, M. B. 2009. Exact Discovery of Time Series Motifs. In *SDM*, 473–484. SIAM.

[22] Murray, D.; Liao, J.; Stankovic, L.; Stankovic, V.; Hauxwell-Baldwin, R.; Wilson, C.; Coleman, M.; Kane, T.; and Firth, S. 2015. A data management platform for personalised real-time energy feedback.

[23] Nam, J.; Cha, H.; Ahn, S.; Lee, J.; and Shin, J. 2020. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33: 20673–20684.

[24] Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

[25] O'neil, C. 2016. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Broadway books.

[26] Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*.

[27] Pezeshki, M.; Kaba, O.; Bengio, Y.; Courville, A. C.; Precup, D.; and Lajoie, G. 2021. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34: 1256–1272.

[28] Qin, Y.; Song, D.; Chen, H.; Cheng, W.; Jiang, G.; and Cottrell, G. 2017. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*.

[29] Rangapuram, S. S.; Seeger, M. W.; Gasthaus, J.; Stella, L.; Wang, Y.; and Januschowski, T. 2018. Deep state space models for time series forecasting. In *Advances in neural information processing systems*, 7785–7794.

[30] Sagawa, S.; Koh, P. W.; Hashimoto, T. B.; and Liang, P. 2019. Distributionally Robust Neural Networks. In *International Conference on Learning Representations*.

[31] Salinas, D.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2019. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*.

[32] Senin, P.; Lin, J.; Wang, X.; Oates, T.; Gandhi, S.; Boedihardjo, A. P.; Chen, C.; and Frankenstein, S. 2015. Time series anomaly discovery with grammar-based compression. In *Edbt*, 481–492.

[33] Shokoohi-Yekta, M.; Chen, Y.; Campana, B.; Hu, B.; Zakaria, J.; and Keogh, E. 2015. Discovery of meaningful rules in time series. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1085–1094.

[34] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

[35] Wang, H.; He, Z.; Lipton, Z. C.; and Xing, E. P. 2018. Learning Robust Representations by Projecting Superficial Statistics Out. In *International Conference on Learning Representations*.

[36] Woolson, R. F. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, 1–3.

[37] Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34: 22419–22430.

[38] Yeh, C.-C. M.; Zhu, Y.; Ulanova, L.; Begum, N.; Ding, Y.; Dau, H. A.; Silva, D. F.; Mueen, A.; and Keogh, E. 2016. Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, 1317–1322. Ieee.

[39] Yi, K.; Zhang, Q.; Fan, W.; Wang, S.; Wang, P.; He, H.; An, N.; Lian, D.; Cao, L.; and Niu, Z. 2024. Frequency-domain MLPs are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems*, 36.

[40] Zhang, G.; Zhong, H.; Tan, Z.; Cheng, T.; Xia, Q.; and Kang, C. 2022. Texas electric power crisis of 2021 warns of a new blackout mechanism. *CSEE journal of Power and Energy Systems*, 8(1): 1–9.

[41] Zhang, L.; Gao, Y.; and Lin, J. 2020. Semantic Discord: Finding unusual local patterns for time series. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, 136–144. SIAM.

[42] Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11106–11115.

[43] Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. *arXiv preprint arXiv:2201.12740*.

[44] Zhu, Y.; Zimmerman, Z.; Senobari, N. S.; Yeh, C.-C. M.; Funning, G.; Mueen, A.; Brisk, P.; and Keogh, E. 2016. Matrix profile ii: Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, 739–748. IEEE.