FRAUG: FREQUENCY DOMAIN AUGMENTATION FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Data augmentation (DA) has become a *de facto* solution to expand training data size for deep learning. With the proliferation of deep models for time series analysis, various time series DA techniques are proposed in the literature, e.g., cropping-, warping-, flipping-, and mixup-based methods. However, these augmentation methods are mainly applicable for time series classification and anomaly detection tasks. In time series forecasting (TSF), we need to model the fine-grained temporal relationship within time series segments so that we could generate faithful forecasting results given data in a look-back window. Existing DA solutions in the time domain would break such relationship, leading to poor forecasting accuracy. To tackle this problem, this paper proposes simple yet effective frequency domain augmentation techniques that ensure the semantic consistency of augmented data-label pairs in forecasting, named *FrAug*. We conduct comprehensive experiments on eight widely-used benchmarks with several state-of-the-art TSF deep models. Our results show that FrAug can boost the forecasting accuracy of existing models in most cases. Moreover, we show that, FrAug enables models trained with 1% of the original training data to achieve similar performance to the ones trained on full training data, which is particularly attractive for cold-start forecasting often occurred in real-life applications.

1 INTRODUCTION

Deep learning is data hungry. Without abundant training data, deep models tend to suffer from poor convergence or overfitting problems. As collecting and labeling real-world data can be costly and time-consuming, data augmentation (DA) has become a *de facto* solution to expand the training dataset size for performance improvement (Cubuk et al., 2019).

Time series related tasks (e.g., classification, forecasting, and anomaly detection (AD)) have a wide range of applications. Depending on the application itself, the collected data could be quite scarce. For example, one particular use case, known as cold start forecasting, performs time series forecasting (TSF) with little or no historical data, e.g., sales prediction for new products. Consequently, with the proliferation of deep models for time series analysis, various time series DA techniques are proposed in the literature (Wen et al., 2020).

When using data augmentation for supervised learning, we create artificial data-label pairs from existing labeled data. It is critical to ensure the semantic consistency of such modified data-label pairs. Otherwise, *augment ambiguity* is introduced, thereby deteriorating the performance of the model instead of improving it, as examined in several previous works in the computer vision (CV) field (Gong et al., 2021; Wei et al., 2020). In fact, DA for image data is less ambiguous when compared to that for time series data, because image data are relatively easy to interpret, enabling us to create semantics-preserving augmentations (e.g., rotation and cropping/masking less relevant regions).

In contrast, time series data are comprised of measurements or events generated from complicated dynamic systems, and we are interested in the temporal relationship among continuous data points in the series. As any perturbations would change such relationship, care must be taken to ensure that the semantics of the data-label pair is preserved, i.e., they are likely to occur according to the behavior of the underlying system. Note that, most existing DA methods for time series focus on the classification and AD tasks, and the data-label pair semantics are preserved to a large extent. For

example, we could perform window cropping (Cui et al., 2016; Le Guennec et al., 2016), window warping (Wen et al., 2020), and noise injection (Wen & Keyes, 2019) on time series without changing the classification labels as long as such manipulations do not yield class changes. Similarly, label expansion (Gao et al., 2020) that manipulates the "blurry" start and end points of sequence anomalies brings performance improvements for the anomaly detection task.

However, time series forecasting is a regression task. It requires modeling the fine-grained temporal relationship within a timing window divided into two parts: the data points in the *look-back window* and those in the *forecasting horizon*, serving as the data and the label when training TSF models, respectively. Aggressive augmentation methods such as cropping or warping would cause missing values or periodicity changes in the series and hence are not applicable for TSF models. In other words, the augmented data-label pairs for TSF are much more stringent compared to those for other time series analysis tasks, which has not been thoroughly investigated in the literature.

In this paper, we argue that augmentation methods for TSF should abandon perturbations in the time domain because such augmented data-label pairs do not conform to the fine-grained temporal relationship within a timing window. For a dynamic system that generates time series data, its forecastable behavior is usually driven by some periodical events[1]. By identifying and manipulating such events in the frequency domain, the generated data-label pairs would still be faithful to the underlying dynamic system. This motivates us to propose two simple yet effective frequency domain augmentation methods for TSF, named *FrAug*. Specifically, FrAug performs *frequency masking* and *frequency mixing*, which randomly eliminate some frequency components of a timing window or mix-up the same frequency components of different timing windows. Experimental results on eight widely-used TSF benchmark datasets show that FrAug improves the forecasting accuracy of various deep models, especially when the size of the training dataset is small.

Specifically, the main contributions of this work include:

- To the best of our knowledge, this is the first work that systematically investigates data augmentation techniques for the TSF task.
- We propose a novel frequency domain augmentation technique named *FrAug*, including two simple yet effective methods (i.e., frequency masking and frequency mixing) that preserve the semantic consistency of augmented data-label pairs in forecasting.
- In our experiments, we show that, FrAug alleviates overfitting problems of state-of-the-art (SOTA) TSF models, thereby improving their forecasting performance. Moreover, FrAug enables models trained with 1% of the original training data to achieve similar performance to the ones trained on full training data in some datasets, which is particularly attractive for cold-start forecasting problems.

2 Related Work and Motivation

Data augmentation methods are task-dependent. In this section, we first survey existing time series DA techniques in Sec. 2.1. Next, in Sec. 2.2, we analyze why existing DA methods are not applicable for the forecasting task. Finally, we discuss the motivation to perform frequency domain augmentation for TSF in Sec. 2.3.

2.1 Data Augmentation Methods for Time Series Analysis

For the time series classification task, many works regard the series as a waveform image and borrow augmentation methods from the CV field, e.g., window cropping (Le Guennec et al., 2016), window flipping (Wen et al., 2020), and Gaussian noise injection (Wen & Keyes, 2019). There are also DA methods that take advantage of specific time series properties, e.g., window warping (Wen et al., 2020), surrogate series (Keylock, 2006; Lee et al., 2019), and time-frequency feature augmentation (Keylock, 2006; Steven Eyobu & Han, 2018; Park et al., 2019; Gao et al., 2020). For the time series AD task, window cropping and window flipping are also often used. In addition, label expansion and amplitude/phase perturbations are introduced in (Gao et al., 2020).

[1]The measurements contributed by random events are not predictable.

There are few DA works for the forecasting task in the literature. (Hu et al., 2020) proposes *DATSING*, a transfer learning-based framework that leverages cross-domain time series latent representations to augment target domain forecasting. (Bandara et al., 2021) introduces two DA methods for forecasting: (i). Average selected with distance (ASD), which generates augmented time series using the weighted sum of multiple time series (Forestier et al., 2017), and the weights are determined by the dynamic time warping (DTW) distance; (ii). Moving block bootstrapping (MBB) generates augmented data by manipulating the residual part of the time series after STL decomposition (Cleveland et al., 1990) and recombining it with the other series.

2.2 Time Series Forecasting

Given the data points in a look-back window $x = \{x_1^t, ..., x_C^t\}_{t=1}^{L}$ of multivariate time series, where L is the look-back window size, C is the number of variates, and x_i^t is the value of the i_{th} variate at the t_{th} time step. The TSF task is to predict the horizon $\hat{x} = \{\hat{x}_1^t, ..., \hat{x}_C^t\}_{t=L+1}^{L+T}$, the values of all variates at future T time steps. When the forecasting horizon T is large, it is referred to as long-term forecasting problem, which has attracted lots of attention in recent research(Zhou et al., 2021; Zeng et al., 2022).

During training, the data points in the look-back window and the data points in the forecasting horizon serve as the data and the label, respectively. Obviously, the augmented data-label pairs for TSF need to be semantically consistent with the behavior of the underlining system. Existing time series DA techniques, however, cannot adhere to this principle since they only apply augmentation to the data.

Figure 1 visualizes the impact of a few existing DA methods that are originally used for classification and AD tasks, which clearly break the fine-grained temporal relationship between look-back window and forecasting horizon. When adopting them for the forecasting task, as expected, the results are quite poor, as shown in Table 1. Note that, the DA methods dedicated to the forecasting task in (Bandara et al., 2021) have similar problems, and we show their results in Sec. 4.

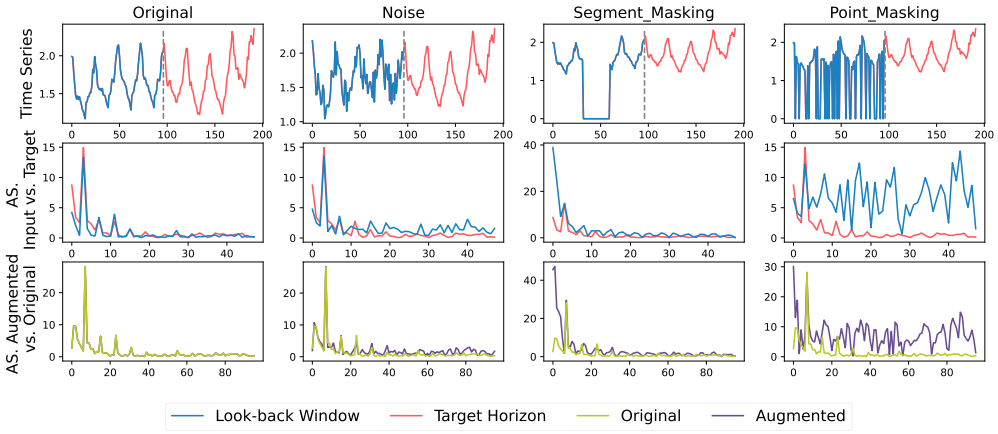


Figure 1: Augmentation result of some previous augmentation on the 1000∼1192 frames of 'OT' channel of ETTh1 dataset. The first row shows the waveform of the augmented time series. The second row shows the amplitude spectrum (AS.) of input and output, where these methods break the consistency of input and target output. The third row shows the amplitude spectrum of augmented time series and original time series, in which we find that these augmentation methods introduce random noise on the entire frequency bands. For better scale, we removed the 0 frequency components from the graphs.

Table 1: Results of different models trained on ETTh1 dataset augmented by existing augmentation methods. We can observe that these methods will degrade performance on all models. The forecasting length is 96. The metric is MSE, the lower the better.

Method	Origin	Noise	Mask-Rand.	Mask-Seg.	Flipping	Warping
DLinear	**0.381**	0.444	0.803	0.448	0.544	0.401
FEDformer	**0.374**	0.380	0.448	0.433	0.420	0.385
Autoformer	**0.449**	0.460	0.608	0.568	0.446	0.465
Informer	0.931	0.936	**0.846**	1.013	0.955	1.265

2.3 WHY FREQUENCY DOMAIN AUGMENTATION?

Time series data are comprised of measurements or events generated from complicated dynamic systems, whose forecastable behavior is usually driven by some periodical events. For example, the hourly sampled power consumption of a house is closely related to the periodical behavior of the house owner. His/her daily routine (e.g., out for work during the day and activities at night) would introduce a daily periodicity, his/her routines between weekdays and weekends would introduce a weekly periodicity, while the yearly atmosphere temperature change would introduce an annual periodicity (e.g., cooling in summer and heating in winter). Such periodical events are decoupled in the frequency domain and can be manipulated independently.

Motivated by the above, we propose to perform frequency domain augmentation for TSF task. By identifying and manipulating events in the frequency domain for data points in both look-back window and forecasting horizon, the resulting augmented data-label pair would largely conform to the behavior of the underlining system.

3 METHODS

In this section, we detail the proposed frequency domain data augmentation methods for time series forecasting, named FrAug.

3.1 THE PIPELINE OF FRAUG

To ensure the semantic consistency of augmented data-label pairs in forecasting, we add frequency domain perturbations on the concatenated time series of look-back window and target horizon, with the help of Fast Fourier transform (FFT) as introduced in Appendix A. We only apply FrAug during the training stage and use original test samples for testing.

As shown in Fig. 2, in the training stage, given a training sample (data points in the look-back window and the forecasting horizon), FrAug (i) concatenates the two parts, (ii) performs frequency domain augmentations, and (iii) splits the concatenated sequence back into lookback window and target horizon in the time domain. The augmentation result of an example time series training sample is shown in Fig. 3.

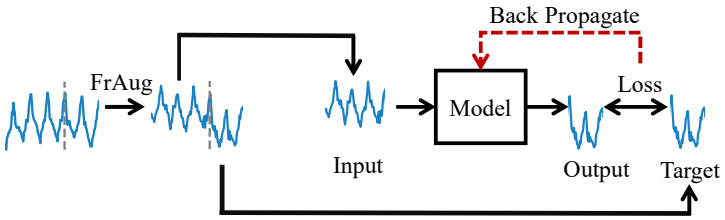


Figure 2: Training with FrAug.

3.2 FREQUENCY MASKING AND FREQUENCY MIXING

We propose two simple yet effective augmentation methods under FrAug framework, namely Frequency Masking and Frequency Mixing. Specifically, frequency masking randomly masks some frequency components, while frequency mixing exchanges some frequency components of two training samples in the dataset.

Frequency masking: The pipeline of frequency masking is shown in Fig. 4(a). For a training sample comprised of data points in the look-back window $x_{t-b:t}$ and the forecasting horizon $x_{t+1:t+h}$, we first concatenate them in the time domain as $s = x_{t-b:t+h}$ and apply real FFT to calculate the frequency domain representation S, which is a tensor composed of complex number. Next, we randomly mask a portion of this complex tensor S as zero and get $\tilde{S}$. Finally, we apply inverse real FFT to project the augmented frequency domain representation back to the time domain $\tilde{s} = \tilde{x}_{t-b:t+h}$. The detailed procedure is shown in Algorithm 1.

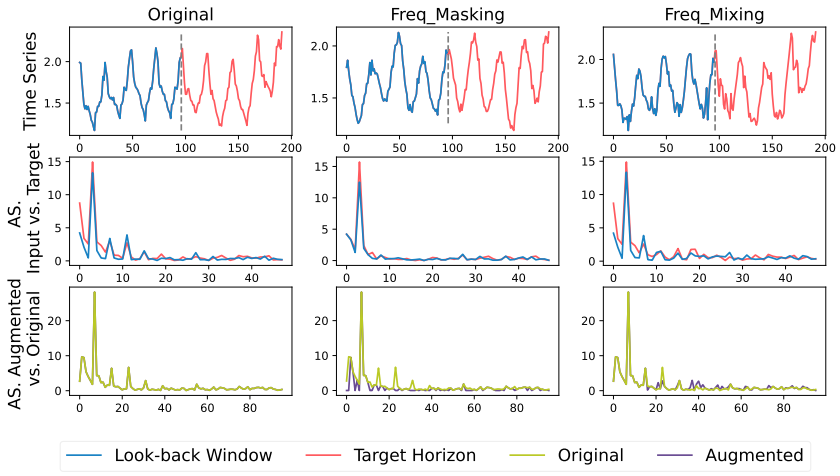
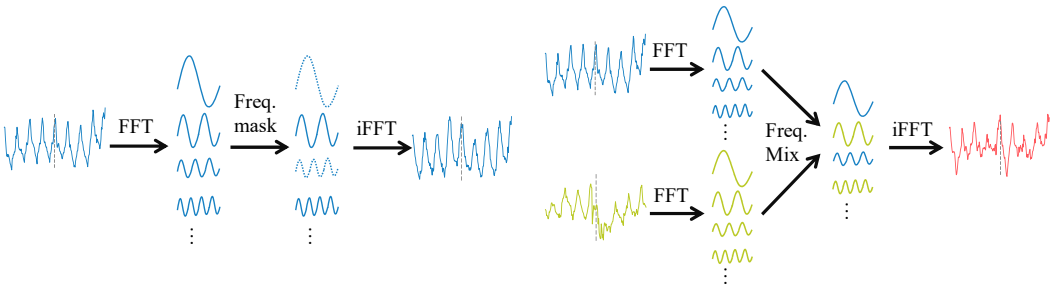


Figure 3: Visualization results of FrAug on 1000∼1192 frames of 'OT' channel of ETTh1 dataset. As shown in the figures of amplitude spectrum (AS.) of look-back window and target horizon (second row), thanks to the unique augmentation pipeline, FrAug largely preserves the semantic consistency between the look-back window and target horizon. Compared with previous DA methods, FrAug also avoids introducing unexpected noise by only performing masking or mixing in the frequency domain, according to the comparison of original and augmented sample amplitude spectrum (third row). For better scale, we removed the 0 frequency components from the graphs.



(a) Frequency masking (b) Frequency mixing

Figure 4: Illustration of the proposed augmentation methods.

Frequency Masking corresponds to removing some events in the underlining system. For example, considering the household power consumption time series, removing the weekly frequency components would create an augmented time series that belongs to a house owner that has similar activities on the weekdays and weekends.

Algorithm 1 Frequency Masking

Input: Look-back window x, target horizon y, mask rate μ
Output: Augmented Look-back window $\tilde{x}$, augmented target horizon $\tilde{y}$

1: $s = x||y$; {Concatenate x and y}
2: $S = rFFT(s)$; {Calculate the frequency representation S. S is composed of complex numbers and have length of $(b+h)//2+1$}
3: $m = CreateRandomMask(len(S), \mu)$ {Create random mask for frequency representation with mask rate μ}
4: $\tilde{S} = Masking(S, m)$;
5: $\tilde{s} = irFFT(\tilde{S})$;
6: $\tilde{x}, \tilde{y} = s[0:b], s[b:b+t]$; {Split the augmented training sample}

Frequency mixing: The pipeline of frequency mixing is shown in Fig. 4, wherein we randomly replace the frequency components in one training sample with the same frequency components of another training sample in the dataset. The details of this procedure is presented in Algorithm 2.

Algorithm 2 Frequency Mixing

Input: Look-back window $x1$, target horizon $y1$, another training sample pair $x2$, $y2$, mix rate μ
Output: Augmented look-back window $\tilde{x}$, augmented target horizon $\tilde{y}$

$s1 = x1||y1, s2 = x2||y2$; {Concatenate x and y}
$S1 = rFFT(s1), S2 = rFFT(s2)$; {Calculate the frequency representation S. S is composed of complex numbers and have length of $(b + h)//2 + 1$}
$m1 = CreateRandomMask(len(S), \mu)$ {Create random mask for frequency representation with mix rate μ no more than 0.5}
$m2 = BitwiseNOT(m1)$ {Create inverted mask for training sample 2}
$\tilde{S} = Masking(S1, m1) + Masking(S2, m2)$;
$\tilde{s} = irFFT(\tilde{S})$;
$\tilde{x}, \tilde{y} = s[0 : b], s[b : b + t]$; {Split the augmented training sample}

Similarly, frequency mixing can be viewed as exchanging events between two samples. For the earlier example on household power consumption, the augmented time series could be one owner's weekly routine replaced by another's, and hence the augmented data-label pair largely preserves semantical consistency for forecasting.

Note that, frequency masking and frequency mixing only utilize information from the original dataset, thereby avoiding the introduction of unexpected noises compared to those dataset expansion techniques based on synthetic generation (Esteban et al., 2017; Yoon et al., 2019). Moreover, as the number of combinations of training samples and their frequencies components in a dataset is extremely large, FrAug can generate nearly infinite reasonable samples.

4 EXPERIMENTS

In this section, we first compare the results of different augmentation methods for the long-term time series forecasting task[2]. To further demonstrate the effectiveness of our methods, we conduct experiments on cold-start forecasting.

4.1 EXPERIMENTAL SETUP

Table 2: The statistics of the eight datasets.

Datasets	Exchange-Rate	Traffic	Electricity	Weather	ETTh1&ETTh2	ETTm1 &ETTm2
Variates	8	862	321	21	7	7
Frequency	1day	1hour	1hour	10min	1hour	5min
Total Timesteps	7,588	17,544	26,304	52,696	17,420	69,680

Dataset. All dataset used in our experiments are widely-used and publicly available real-world datasets, including Exchange-Rate Lai et al. (2017), Traffic, Electricity, Weather, ETT Zhou et al. (2021). We summarize the characteristics of these datasets in Table 2.

Baselines. We compare FrAug, including Frequency Masking (FreqMask) and Frequency Mixing (FreqMix), with existing time-series augmentation techniques for forecasting, including ASD Forestier et al. (2017); Bandara et al. (2021) and MBB Bandara et al. (2021); Bergmeir et al. (2016).

Deep Models. We include four state-of-the-art models for long-term forecasting, including Informer Zhou et al. (2021), Autoformer Wu et al. (2021), FEDformer Zhou et al. (2022) and DLinear Zeng et al. (2022). The effectiveness of augmentations methods are evaluated by comparing the performance of the same model trained with different augmentations methods.

Evaluation metrics. Following previous works Zhou et al. (2021); Wu et al. (2021); Zhou et al. (2022); Zeng et al. (2022), we use Mean Squared Error (MSE) as the core metrics to compare performance.

[2]FrAug is also effective for short-term forecasting task. We present the results in Appendix C.

Table 3: Comparison of different augmentation methods on ETT benchmarks under four forecasting lengths. Performances are measured by MSE. The **best results** are highlighted in **bold**.

Model	Method	ETTh1				ETTh2				ETTm1				ETTm2			
		96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
DLinear	Origin	0.381	0.405	0.439	0.514	0.295	0.378	0.421	0.696	0.300	0.335	**0.368**	**0.425**	0.171	0.235	0.305	0.412
	FreqMask	**0.379**	**0.403**	**0.435**	0.472	**0.277**	**0.338**	0.432	**0.588**	**0.295**	**0.331**	**0.368**	0.426	**0.166**	**0.227**	**0.281**	0.399
	FreqMix	0.380	0.409	0.438	0.478	**0.277**	0.342	0.449	0.636	0.296	**0.331**	0.372	0.428	0.168	**0.227**	0.286	**0.398**
	ASD	0.387	0.554	0.445	**0.467**	0.302	0.363	**0.411**	0.677	0.311	0.343	0.377	0.430	0.188	0.237	0.297	0.400
	MBB	0.389	0.423	0.508	0.521	0.313	0.391	0.433	0.651	0.307	0.339	0.373	0.428	0.177	0.242	0.323	0.430
FEDformer	Origin	0.374	0.425	**0.456**	0.485	0.339	0.430	0.519	0.474	0.364	0.406	**0.446**	0.533	0.189	0.253	0.327	0.438
	FreqMask	0.372	0.417	0.457	**0.474**	**0.323**	**0.409**	**0.462**	**0.440**	**0.352**	0.400	0.450	**0.507**	**0.183**	**0.249**	**0.315**	**0.423**
	FreqMix	**0.371**	**0.416**	0.459	0.478	0.327	0.421	0.502	0.456	0.360	**0.399**	0.447	0.515	0.184	0.250	0.317	0.432
	ASD	0.429	0.455	0.561	0.582	0.339	0.429	0.501	0.454	0.390	0.430	0.514	0.585	0.200	0.264	0.345	0.460
	MBB	0.412	0.460	0.501	0.514	0.356	0.455	0.526	0.484	0.385	0.427	0.477	0.548	0.211	0.270	0.340	0.439
Autoformer	Origin	0.449	0.463	0.495	0.535	0.432	0.430	0.482	0.471	0.552	0.559	0.605	0.755	0.288	0.274	0.335	0.437
	FreqMask	0.419	**0.426**	0.476	0.501	**0.346**	**0.422**	**0.447**	0.462	0.419	**0.513**	**0.472**	0.595	0.213	**0.264**	**0.325**	**0.421**
	FreqMix	**0.401**	0.454	**0.471**	0.517	0.351	0.423	0.455	**0.449**	**0.410**	0.542	0.497	0.529	**0.211**	0.265	**0.325**	0.433
	ASD	0.486	0.497	0.530	**0.499**	0.362	0.442	0.477	0.523	0.561	0.532	0.518	0.616	0.233	0.276	0.331	0.444
	MBB	0.479	0.526	0.592	0.602	0.363	0.431	0.472	0.547	0.535	0.652	0.704	**0.522**	0.239	0.283	0.334	0.454
Informer	Origin	0.931	1.010	1.036	1.159	2.843	6.236	5.418	3.962	0.626	0.730	1.037	0.972	0.389	0.813	1.429	3.863
	FreqMask	**0.621**	**0.788**	**0.851**	**1.009**	2.295	**3.983**	3.724	**2.561**	**0.425**	**0.512**	**0.761**	**0.778**	0.368	**0.463**	**0.984**	3.173
	FreqMix	0.675	0.905	1.044	1.095	2.774	5.917	**3.700**	3.385	0.587	0.636	0.867	0.906	**0.341**	0.557	1.111	2.939
	ASD	0.853	1.020	1.124	1.226	**2.280**	5.830	4.345	3.886	0.726	0.775	0.921	0.968	0.405	0.874	1.317	**2.585**
	MBB	0.958	1.031	1.049	1.227	3.112	6.398	5.668	4.007	0.630	0.731	0.988	0.961	0.399	0.783	1.476	4.012

Table 4: Comparison of different augmentation methods on four datasets under four forecasting lengths. Performances are measured by MSE. The **best results** are highlighted in **bold**.

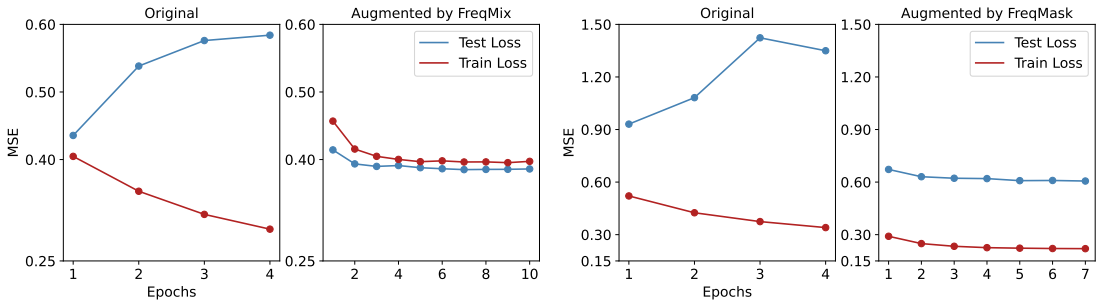
Model	Method	Exchange Rate				Electricity				Traffic				Weather			
		96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
DLinear	Origin	**0.079**	0.205	0.309	1.029	**0.140**	**0.154**	**0.169**	**0.204**	**0.410**	**0.423**	0.436	**0.466**	0.175	0.217	0.265	0.324
	FreqMask	0.095	**0.177**	**0.263**	0.842	**0.140**	**0.154**	**0.169**	**0.204**	0.411	**0.423**	**0.435**	**0.466**	**0.174**	0.217	0.263	**0.323**
	FreqMix	**0.079**	0.180	0.274	0.796	**0.140**	**0.154**	**0.169**	**0.204**	0.412	**0.423**	**0.435**	0.467	**0.174**	**0.216**	0.264	0.324
	ASD	0.102	0.273	0.294	**0.787**	0.163	0.175	0.189	0.222	0.437	0.450	0.463	0.493	0.195	0.230	0.275	0.329
	MBB	0.080	0.204	0.308	1.021	0.145	0.157	0.172	0.206	0.420	0.430	0.441	0.467	0.176	0.217	**0.262**	0.324
FEDformer	Origin	0.135	0.271	0.454	1.140	0.188	0.196	0.212	0.250	0.574	0.611	0.623	0.631	0.250	0.266	0.368	0.397
	FreqMask	**0.122**	**0.232**	**0.422**	**1.058**	**0.176**	0.188	**0.201**	**0.220**	0.569	**0.586**	0.607	0.630	**0.180**	**0.240**	**0.308**	**0.371**
	FreqMix	0.129	0.240	0.444	1.129	**0.176**	**0.187**	0.204	0.225	**0.564**	**0.586**	**0.604**	0.629	0.200	0.245	0.317	0.377
	ASD	0.149	0.265	0.441	1.128	0.192	0.205	0.214	0.243	0.573	0.601	0.608	**0.613**	0.700	0.513	0.623	0.649
	MBB	0.145	0.257	0.456	1.142	0.202	0.223	0.240	0.297	0.601	0.613	0.630	0.647	0.309	0.280	0.352	0.392
Autoformer	Origin	0.145	0.385	**0.453**	1.087	0.203	0.231	0.247	0.276	0.624	0.619	0.604	0.703	0.271	0.315	0.345	0.452
	FreqMask	**0.139**	0.279	0.485	**0.806**	0.170	0.193	0.208	**0.234**	0.564	**0.578**	0.595	0.654	**0.210**	**0.257**	**0.315**	**0.388**
	FreqMix	**0.139**	0.281	0.497	1.050	**0.162**	**0.189**	**0.204**	0.242	**0.559**	0.603	**0.581**	**0.640**	0.252	0.288	0.334	**0.388**
	ASD	0.147	0.312	1.344	1.152	0.248	0.223	0.268	0.254	0.608	0.616	0.603	0.694	1.015	0.574	0.584	0.874
	MBB	0.152	**0.273**	0.472	1.641	0.231	0.317	0.269	0.272	0.628	0.650	0.658	0.664	0.237	0.349	0.376	0.451
Informer	Origin	0.879	1.147	1.562	2.919	0.305	0.349	0.349	0.391	0.736	0.770	0.861	0.995	0.452	0.466	0.499	1.260
	FreqMask	**0.534**	**0.764**	**1.074**	**1.102**	**0.262**	0.281	**0.284**	**0.299**	**0.673**	0.679	**0.715**	**0.797**	**0.199**	**0.293**	**0.356**	**0.487**
	FreqMix	0.962	1.156	1.498	2.689	0.266	**0.276**	0.287	0.300	0.674	**0.677**	0.720	0.805	0.216	0.402	0.459	0.666
	ASD	0.994	1.132	1.669	1.924	0.317	0.331	0.334	0.348	0.812	0.747	0.805	0.900	0.342	0.452	0.529	0.644
	MBB	0.859	1.136	1.549	2.874	0.354	0.389	0.397	0.451	0.752	0.768	0.892	1.057	0.544	0.425	0.601	1.221

Implementation. FreqMask and FreqMix only have one hyper-parameter, which is the mask-rate/mix-rate, respectively. In all the experiments, the rate is selected from {0.1, 0.2, 0.3, 0.4, 0.5}. In most cases, a large mask/mix rate, i.e, 0.5, is better than smaller rates. More implementation details are presented in the Appendix B.

4.2 Main Results

Table 3 and Table 4 show the comparisons of different augmentation methods that double the size of the training dataset. As can be observed, FreqMask and FreqMix improve the performance of the original model in most cases. However, the performances of ASD and MBB are often inferior to the original model.

Notably, FreqMask improves DLinear's performance by 16% in ETTh2 when the predict length is 192, and it improves FEDformer's performance by 28% and Informer's performance by 56% for the Weather dataset when the predict length is 96. Similarly, FreqMix improves the performance of Autoformer by 27% for ETTm2 with a predict length of 96 and the performance of Informer by 35% for ETT2 with predict length of 720. These results indicate that FrAug is an effective DA solution for long-term forecasting, significantly boosting SOTA performance in many cases.

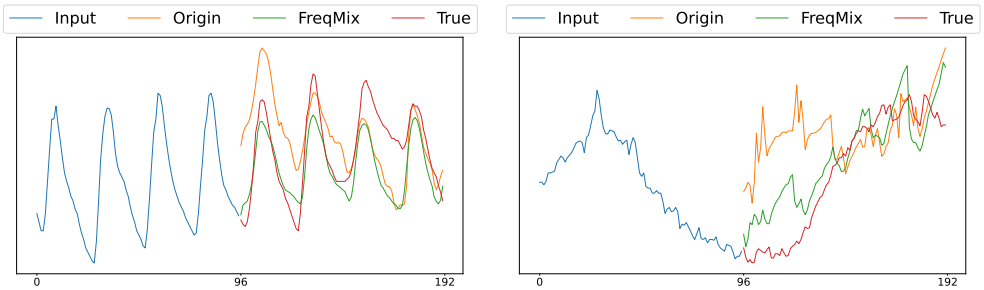


(a) Autoformer (b) Informer

Figure 5: The over-fitting problem of recent SOTA models. We plot the training and testing curve of Autoformer and Informer in ETTh1 dataset, predict length 96. The testing curve are much more flatten after applying FreqMask and FreqMix.

We attribute the performance improvements brought by FrAug to the fact that it alleviates the over-fitting issues of the original model. Generally speaking, a large gap between the training loss and the test loss, the so-called generalization gap, is an indicator of over-fitting. Fig 5 demonstrates training loss and test error curves from deep models Autoformer and Informer. Without FrAug, the training loss of Autoformer and Informer decease with more training epochs, but the test errors increase. In contrast, when FrAug is applied to include more training samples, the test loss can decrease steadily until it is stable. This result clearly shows the benefits of FrAug.

In Fig 6, we visualize some prediction results with/without FrAug. Without FrAug, the model can hardly capture the scale of data, and the predictions show little correlation to the look-back window, i.e, there is a large gap between the last value of the look-back window and the first value of predicted horizon. This indicates that the models are over-fitting. In contrast, with FrAug, the prediction is much more reasonable. More visualization results are presented in Appendix D.



(a) Autoformer, ETTh2 289-6 (b) Informer, Weather 2790-6

Figure 6: Visualization of models' predictions. (a) shows the predictions of Autoformer trained on original dataset and dataset augmented by FreqMask. (b) shows the predictions of Informer trained on original dataset and dataset augmented by FreqMix. With our augmentations, models are less likely to overfit.

4.3 COLD-START FORECASTING

Apart from improving SOTA performances, another important application of FrAug is its capability for cold-start forecasting, where only very few training samples are available.

To simulate this scenario, we reduce the number of training samples of each dataset to the last 1% of the original size. For example, we only use the 8366^{th}-8449^{th} training samples (83 in total) of ETTh2 dataset for training. Then, we use FrAug to generate augmented data to enrich the dataset. In this experiment, we only consider two extreme cases: enlarging the dataset size to 2x or 50x, and a better result is presented. Models are evaluated on the same test dataset as the original long-term forecasting tasks.

Table 5 shows the results of two forecasting models: DLinear and Autoformer on 4 datasets. The other results are shown in Appendix E. We also include the results of models trained with the full dataset for comparison. As can be observed, FrAug maintains the overall performances of the models

Table 5: Performance of models trained with the last 1% training samples compared with that trained with full training set. Performances are measured by MSE. The **best results** are highlighted in **bold** (row full data is not included).

Model	Method	ETTh2				Exchange Rate				Electricity				Traffic			
		96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
DLinear	Full Data	0.295	0.378	0.421	0.696	0.079	0.205	0.309	1.029	0.140	0.154	0.169	0.204	0.410	0.423	0.436	0.466
	1% Data	0.572	0.704	0.628	0.662	0.280	0.549	1.711	0.897	0.196	0.205	0.218	0.280	0.764	0.658	0.825	0.908
	FreqMask	**0.351**	**0.467**	**0.541**	**0.640**	0.174	0.258	**0.777**	**0.820**	**0.172**	**0.183**	**0.197**	**0.257**	**0.466**	**0.484**	**0.509**	**0.539**
	FreqMix	0.464	0.531	0.590	1.005	**0.139**	**0.252**	1.205	0.897	0.176	0.185	0.200	**0.257**	0.486	0.510	0.515	0.578
Autoformer	Full Data	0.432	0.430	0.482	0.471	0.145	0.385	0.453	1.087	0.203	0.231	0.247	0.276	0.624	0.619	0.604	0.703
	1% Data	0.490	0.546	0.498	0.479	0.247	0.487	**0.579**	1.114	0.462	0.488	0.518	0.541	1.238	1.366	1.299	1.325
	FreqMask	**0.409**	**0.496**	**0.476**	**0.471**	**0.166**	**0.304**	0.791	**1.085**	**0.326**	0.331	0.410	0.483	0.761	**0.832**	**0.706**	**0.786**
	FreqMix	0.414	0.502	0.484	**0.471**	0.208	0.369	0.832	1.091	0.347	**0.326**	**0.381**	**0.479**	**0.746**	0.844	0.715	0.795

to a large extent. For Traffic dataset, compared with the one trained on full training data, the overall performance drop of DLinear is 13% with FrAug, while the drop is 45% without FrAug. Surprisingly, sometimes the model performance is even better than those trained with the full dataset. For example, the performance of Autoformer is 5.4% better than the one trained with the full dataset for ETTh2 when the predict length is 96. The performance of DLinear is 20% better than the one trained with full dataset for Exchange Rate when the predict length is 720. We attribute it to the distribution shift in the full training dataset, which could deteriorate the model performance.

4.4 Discussions

Combining FreqMask and FreqMix: FreqMask and FreqMix do not conflict with each other. Therefore, it is possible to combine them for data augmentation. Table 6 shows the results of such combinations. However, we cannot observe much improvements. We hypothesize that either FreqMask or FreqMix has already alleviated the overfitting issue of the original model with the augmented samples.

Table 6: Performance of combination of FreqMask and FreqMix in ETTh2. Performances are measured by MSE. The **best results** are highlighted in **bold**.

Model	DLinear				Autoformer			
Method	Origin	FreqMask	FreqMix	Combine	Origin	FreqMask	FreqMix	Combine
96	0.295	**0.277**	**0.277**	**0.277**	0.432	**0.346**	0.351	0.349
192	0.378	0.338	0.342	**0.334**	0.430	0.422	0.423	**0.419**
336	**0.421**	0.432	0.449	0.428	0.482	**0.447**	0.455	0.458
720	0.696	0.588	0.636	**0.546**	0.471	0.462	**0.449**	0.455

Selection of mask/mix rate: In our experiment, a large mask/mix rate, i.e, 0.5, is usually better than smaller rates. On the other hand, we also observe that small rate sometimes achieve better results for datasets whose sizes are relatively large (e.g., ETTm1 and ETTm2) or datasets with clean patterns (e.g., electricity). In Appendix F, we present the mask/mix rate used in Sec. 4.2.

5 Conclusion

This work explores effective data augmentation techniques for the time series forecasting task. By systematically analyzing existing augmentation methods for time series, we first show they are not applicable for TSF as the augmented data-label pairs cannot meet the semantical consistency requirements in forecasting. Then, we propose FrAug, a novel frequency domain augmentation technique. The proposed frequency masking and frequency mixing strategies not only effectively expand training data size, but also are easy to implement. Comprehensive experiments on widely-used datasets validate that FrAug alleviates the overfitting problems of state-of-the-art (SOTA) TSF models, thereby improving their forecasting performance. In particular, FrAug significantly improves the model performance under cold-start forecasting, which occurs frequently in practical applications.

REFERENCES

Kasun Bandara, Hansika Hewamalage, Yuan-Hao Liu, Yanfei Kang, and Christoph Bergmeir. Improving the accuracy of global forecasting models using time series data augmentation. *Pattern Recognition*, 120:108148, 2021.

Christoph Bergmeir, Rob J Hyndman, and José M Benítez. Bagging exponential smoothing methods using stl decomposition and box–cox transformation. *International journal of forecasting*, 32(2): 303–312, 2016.

Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition. *J. Off. Stat*, 6(1):3–73, 1990.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.

Zhicheng Cui, Wenlin Chen, and Yixin Chen. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995*, 2016.

Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

Germain Forestier, François Petitjean, Hoang Anh Dau, Geoffrey I Webb, and Eamonn Keogh. Generating synthetic time series to augment sparse datasets. In *2017 IEEE international conference on data mining (ICDM)*, pp. 865–870. IEEE, 2017.

Jingkun Gao, Xiaomin Song, Qingsong Wen, Pichao Wang, Liang Sun, and Huan Xu. Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks. *arXiv preprint arXiv:2002.09545*, 2020.

Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Keepaugment: A simple information-preserving data augmentation approach. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1055–1064, 2021.

Hailin Hu, MingJian Tang, and Chengcheng Bai. Datsing: Data augmented time series forecasting with adversarial domain adaptation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2061–2064, 2020.

CJ Keylock. Constrained surrogate time series with preservation of the mean and variance structure. *Physical Review E*, 73(3):036707, 2006.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. *international acm sigir conference on research and development in information retrieval*, 2017.

Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*, 2016.

Tracey Eileen KM Lee, YL Kuah, Kee-Hao Leo, Saeid Sanei, Effie Chew, and Ling Zhao. Surrogate rehabilitative time series data for image-based deep learning. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pp. 1–5. IEEE, 2019.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Odongo Steven Eyobu and Dong Seog Han. Feature representation and data augmentation for human activity classification based on wearable imu sensor data using a deep lstm neural network. *Sensors*, 18(9):2892, 2018.

Longhui Wei, An Xiao, Lingxi Xie, Xiaopeng Zhang, Xin Chen, and Qi Tian. Circumventing outliers of autoaugment with knowledge distillation. In *European Conference on Computer Vision*, pp. 608–625. Springer, 2020.

Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

Tailai Wen and Roy Keyes. Time series anomaly detection using convolutional neural networks and transfer learning. *arXiv preprint arXiv:1905.13628*, 2019.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *arXiv preprint arXiv:2205.13504*, 2022.

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

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 2022.

Appendix: FrAug: Frequency Domain Augmentation for Time Series Forecasting

In this appendix, we provide i). a brief introduction to the FFT in Sec. A, ii). more implementation details in Sec. B, iii). an extra study of FrAug in short-term forecasting tasks in Sec. C, iv). more visualizations about how FrAug alleviate over-fitting in Sec. D, v). more experiment results of FrAug in cold-start forecasting problem in Sec. E, vi). hyper-parameters for FrAug in Sec. F.

A Discrete Fourier Transform and Fast Fourier Transform (FFT)

Specifically, DFT converts a finite sequence of equally-spaced samples of a function into a same-length sequence of equally-spaced samples of a complex-valued function of frequency. Given a sequence $x = \{x_n\}$ with $n \in [0, N-1]$, the DFT is defined by:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk}, 0 \leq k \leq N-1$$

The most commonly-used DFT calculating method is Fast Fourier Transform (FFT). However, when dealing with real number input, the positive and negative frequency parts are conjugate with each other. Thus, we can get a more compact one-sided representation where only the positive frequencies are preserved, which have length of $(N+1)//2$. We use *pyTorch* function *torch.fft.rfft* and *torch.fft.irfft* to perform real FFT and inverse real FFT. In the following sections, we refer spectrum as the positive frequency spectrum calculated by real FFT.

B Implementation Details

FrAug. In application, FrAug can be implemented by a few lines of code. In our experiment, when we double the size of the dataset, we apply FrAug in a batch-wise style. For example, when the experiments' default batch size is 32, we reduce it to 16. In the training process, we use FrAug to create an augmented version for each sample in the batch, so that the batch size is back to 32 again. Such procedure can reduce memory costs since we don't need to generate and store all augmented samples at once. Also, such a design enhances the diversity of samples, since the mask/mix rate in FrAug introduces randomness to the augmentation.

Other baselines. In the experiments of the main paper, we compare Frequency Masking and Frequency Mixing with ASD (Forestier et al., 2017; Bandara et al., 2021) and MBB (Bandara et al., 2021; Bergmeir et al., 2016). These two methods are reproduced by us based on the descriptions in their original paper. Specifically, for ASD, we first calculate the pair-wise DTW distances of all training samples. Then, to generate a new sample, we applied an exponentially weighted sum to each sample's top5 closest neighbors. This weighted sum is on both look-back window and horizon. Finally, we combine all new samples with the original dataset and train the model with them. For MBB, we apply a similar batch-wise augmentation procedure as FrAug. For each sample, we use the STL decompose from package *statsmodels* to extract the residual component of each training sample. Then we use the MovingBlockBootstrap from package *arch* to add perturbations on the residual component. Finally, we recombine the residual part with the other components.

C Short-term forecasting tasks

We present the results of FrAug on the short-term forecasting tasks in table7. For FEDformer, Autoformer and Informer, FrAug consistently improves their performances. Notably, FrAug improves the performances of Autoformer by 28% in ETTm1 with a predict length of 3. In exchange rate with a predict length of 3, it improves the performance of Autoformer by 50% and improves the performance of Informer by 79%. However, no significant performance boost is observed in DLinear. We find that the model capacity of DLinear is extremely low in short-term forecasting tasks. For example, when the horizon is 3, the number of parameters in DLinear is 6x look-back window size,

Table 7: Performance of models in short-term forecasting tasks. Performances are measured by MSE. The **best results** are highlighted in **bold**.

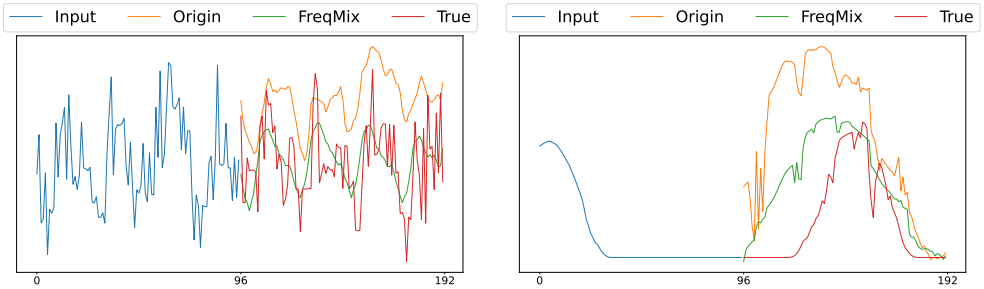
Dataset	PredLen	DLinear			FEDformer			Autoformer			Informer		
		Origin	FreqMask	FreqMix	Origin	FreqMask	FreqMix	Origin	FreqMask	FreqMix	Origin	FreqMask	FreqMix
ETTh1	3	**0.168**	0.170	0.177	0.200	**0.192**	0.198	0.260	0.241	**0.209**	0.264	**0.191**	0.219
	6	0.241	**0.233**	**0.233**	0.250	0.244	**0.242**	0.339	0.329	**0.305**	0.482	**0.311**	0.322
	12	0.307	**0.286**	**0.286**	0.293	0.289	**0.288**	0.380	0.371	**0.331**	0.649	**0.375**	0.377
	24	0.319	**0.317**	0.318	0.312	**0.305**	**0.305**	**0.374**	0.377	0.406	0.690	**0.406**	0.443
ETTh2	3	**0.083**	0.084	0.085	0.151	0.139	**0.136**	0.171	0.146	**0.140**	0.288	0.163	**0.141**
	6	0.104	**0.103**	**0.103**	0.166	0.155	**0.152**	0.231	0.203	**0.177**	0.577	**0.266**	0.292
	12	0.131	**0.130**	**0.130**	0.187	**0.176**	0.177	0.247	0.222	**0.202**	1.078	**0.370**	0.497
	24	0.168	**0.166**	0.167	0.216	0.205	**0.204**	0.298	0.249	**0.246**	1.218	**0.512**	1.127
ETTm1	3	**0.062**	0.063	0.064	0.093	**0.089**	0.090	0.255	0.194	**0.184**	0.091	0.074	**0.072**
	6	**0.088**	**0.088**	0.090	0.120	**0.114**	**0.114**	0.270	0.188	**0.186**	0.130	**0.108**	0.114
	12	0.138	**0.136**	0.137	0.171	**0.168**	**0.168**	0.291	**0.253**	0.262	0.251	**0.175**	0.192
	24	0.211	**0.209**	**0.209**	**0.279**	**0.279**	0.281	0.418	**0.335**	0.346	0.320	**0.277**	0.362
ETTm2	3	**0.044**	**0.044**	**0.044**	0.068	**0.060**	0.061	0.095	0.087	**0.076**	0.071	**0.055**	**0.055**
	6	**0.056**	**0.056**	**0.056**	0.080	**0.074**	**0.074**	0.124	**0.108**	0.110	0.100	**0.077**	0.086
	12	**0.074**	**0.074**	**0.074**	0.096	0.092	**0.091**	0.124	**0.113**	**0.113**	0.142	**0.104**	0.124
	24	**0.098**	**0.098**	0.100	0.115	**0.112**	**0.112**	0.152	**0.131**	0.138	0.235	**0.163**	0.202
Exchange	3	**0.005**	**0.005**	**0.005**	0.031	**0.026**	0.028	0.039	0.024	**0.018**	0.422	**0.088**	0.297
	6	**0.008**	0.009	**0.008**	0.035	**0.030**	**0.030**	0.031	**0.021**	0.023	0.575	**0.125**	0.442
	12	**0.014**	**0.014**	**0.014**	0.044	**0.039**	0.040	0.054	**0.028**	0.030	0.581	**0.132**	0.470
	24	**0.024**	**0.024**	**0.024**	0.054	0.050	**0.049**	0.061	**0.043**	0.049	0.583	**0.255**	0.562
Electricity	3	**0.070**	0.081	0.087	0.142	**0.129**	0.130	0.147	**0.128**	0.129	0.233	**0.182**	0.187
	6	**0.085**	0.090	0.092	0.149	**0.137**	**0.137**	0.152	**0.135**	0.136	0.271	0.211	**0.210**
	12	**0.099**	0.106	0.110	0.157	0.146	**0.145**	0.158	**0.140**	0.141	0.286	**0.217**	0.222
	24	**0.110**	**0.110**	0.111	0.164	**0.153**	**0.153**	0.176	**0.139**	0.141	0.292	**0.229**	0.231
Traffic	3	**0.308**	0.326	0.341	0.537	**0.506**	0.510	0.563	**0.515**	0.519	0.597	**0.580**	0.586
	6	**0.342**	0.347	0.352	0.548	**0.519**	**0.519**	0.564	**0.526**	0.527	0.619	0.609	**0.607**
	12	**0.360**	0.372	0.380	0.547	0.525	**0.521**	0.565	**0.530**	0.532	0.634	**0.608**	0.625
	24	**0.371**	**0.371**	0.372	0.548	0.532	**0.528**	0.560	0.534	**0.527**	0.672	0.632	**0.622**
Weather	3	0.049	**0.047**	**0.047**	0.086	**0.080**	0.081	0.156	**0.115**	0.131	0.068	0.058	**0.056**
	6	**0.061**	**0.061**	**0.061**	0.099	0.093	**0.090**	0.158	**0.123**	0.128	0.074	**0.066**	0.067
	12	**0.079**	**0.079**	**0.079**	0.138	**0.116**	0.122	0.172	**0.139**	0.142	0.111	**0.086**	0.094
	24	**0.104**	0.105	0.105	0.153	0.151	**0.140**	0.182	**0.147**	0.158	0.189	**0.126**	0.161

while other models have more than millions of parameters. Such low model capacity makes DLinear hard to benefit from augmented data.

D MORE VISUALIZATIONS ABOUT OVER-FITTING PROBLEM

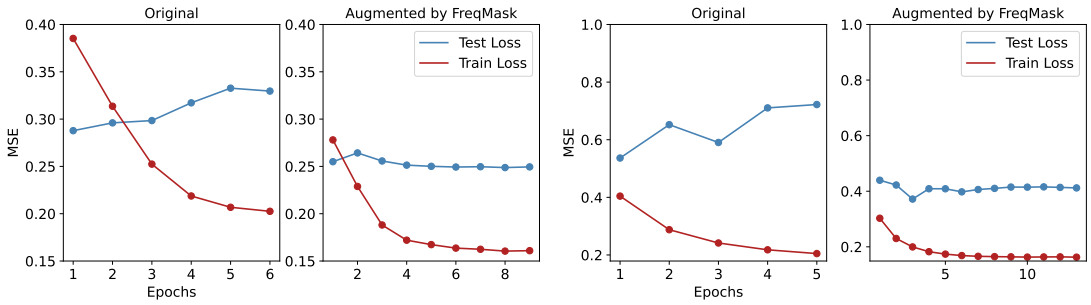
Long-term forecasting. We present more visualization of prediction results in Fig. 7. With FrAug, models are less likely to overfit, therefore they can better capture the information in the look-back window and make reasonable predictions.

Short-term forecasting. We also presents some visualizations of training and testing curve in Fig. 8. FrAug can effectively reduce the generalization gap.



(a) Autoformer, ETTh2 967-4 (b) Informer, Weather 2927-17

Figure 7: Visualization of models' predictions in long-term forecasting tasks. (a) shows the predictions of Autoformer trained on original dataset and dataset augmented by FreqMix. (b) shows the predictions of Informer trained on original dataset and dataset augmented by FreqMask. With our augmentations, models are less likely to overfit.



(a) Autoformer (b) Informer

Figure 8: The over-fitting problem of recent SOTA models in short-term forecasting tasks. We plot the training and testing curve of Autoformer in ETTh2 dataset with predict length 24, and of Informer in ETTh1 with predict length 12. The testing curve are much more flatten after applying FreqMask and FreqMix.

Table 8: Performance of models trained with the last 1% training samples compared with those trained with full training set. Performances are measured by MSE. The **best results** are highlighted in **bold** (row full data is not included).

Dataset	PredLen	DLinear				FEDformer				Autoformer				Informer			
		1% Data	FreqMask	FreqMix	Full Data	1% Data	FreqMask	FreqMix	Full Data	1% Data	FreqMask	FreqMix	Full Data	1% Data	FreqMask	FreqMix	Full Data
ETTh1	96	0.468	**0.467**	0.484	0.381	0.636	**0.457**	0.469	0.374	0.693	**0.485**	0.512	0.449	1.542	**0.767**	1.518	0.931
	192	0.666	**0.546**	0.549	0.405	0.659	**0.549**	0.617	0.425	0.753	0.608	**0.594**	0.463	1.508	**1.131**	1.518	1.010
	336	0.901	**0.594**	0.617	0.439	0.665	**0.589**	0.662	0.456	0.661	0.583	**0.580**	0.495	1.509	**1.042**	1.579	1.036
	720	0.761	**0.731**	0.742	0.514	0.648	**0.605**	**0.605**	0.485	0.709	0.589	**0.587**	0.535	1.432	**1.200**	1.514	1.159
ETTh2	96	0.572	**0.351**	0.464	0.295	0.398	**0.391**	0.394	0.339	0.490	**0.409**	0.414	0.432	3.115	**2.276**	3.002	2.843
	192	0.704	**0.467**	0.531	0.378	**0.453**	0.466	0.470	0.430	0.546	**0.496**	0.502	0.430	2.882	**1.969**	2.775	6.236
	336	0.628	**0.541**	0.590	0.421	**0.472**	0.479	0.483	0.519	0.498	**0.476**	0.484	0.482	3.082	**1.815**	2.709	5.418
	720	0.662	**0.640**	1.005	0.696	0.457	**0.455**	0.456	0.474	0.479	**0.471**	**0.471**	0.471	3.017	**1.814**	2.953	3.962
ETTm1	96	0.392	**0.371**	0.374	0.300	0.743	**0.546**	0.616	0.364	0.692	**0.630**	0.656	0.552	1.652	**0.692**	1.943	0.626
	192	0.407	**0.388**	0.389	0.335	0.745	**0.533**	0.541	0.406	0.665	**0.649**	0.652	0.559	1.653	**0.755**	1.904	0.730
	336	0.432	**0.416**	0.418	0.368	0.750	**0.616**	0.709	0.446	**0.625**	0.632	0.626	0.605	1.720	**0.801**	1.812	1.037
	720	0.490	**0.471**	0.472	0.425	0.743	**0.660**	0.710	0.533	0.713	**0.697**	0.699	0.755	1.914	**0.919**	1.979	0.972
ETTm2	96	0.396	**0.202**	0.317	0.171	0.293	**0.255**	0.275	0.189	0.294	**0.258**	0.289	0.288	2.548	**2.206**	2.640	0.389
	192	0.795	**0.254**	0.650	0.235	0.342	**0.330**	0.333	0.253	0.395	**0.355**	0.426	0.274	2.703	**2.287**	2.675	0.813
	336	0.412	**0.381**	0.420	0.305	0.400	**0.389**	0.393	0.327	**0.397**	0.399	0.407	0.335	3.633	**2.382**	3.055	1.429
	720	0.837	**0.721**	0.929	0.412	0.477	**0.474**	0.481	0.438	**0.481**	0.529	0.562	0.437	2.812	**2.074**	2.529	3.863
Exchange	96	0.280	0.174	**0.139**	0.079	**0.160**	0.166	0.166	0.135	0.247	**0.166**	0.208	0.145	1.674	**0.384**	1.220	0.879
	192	0.549	0.258	**0.252**	0.205	**0.266**	0.278	0.288	0.271	0.487	**0.304**	0.369	0.385	1.651	**0.420**	1.313	1.147
	336	1.711	**0.777**	1.205	0.309	**0.430**	0.520	0.523	0.454	**0.579**	0.791	0.832	0.453	1.849	**0.798**	1.330	1.562
	720	0.897	**0.820**	0.897	1.029	**0.927**	0.942	0.942	1.140	1.114	**1.085**	1.091	1.087	1.827	**1.027**	1.597	2.919
Electricity	96	0.196	**0.172**	0.176	0.140	0.537	0.314	**0.302**	0.188	0.462	**0.326**	0.347	0.203	1.238	**0.744**	1.118	0.305
	192	0.205	**0.183**	0.185	0.154	0.530	0.321	**0.310**	0.196	0.488	0.331	**0.326**	0.231	1.230	**0.681**	0.989	0.349
	336	0.218	**0.197**	0.200	0.169	0.533	0.337	**0.335**	0.212	0.518	0.410	**0.381**	0.247	1.216	**0.767**	1.025	0.349
	720	0.280	**0.257**	**0.257**	0.204	0.563	**0.500**	0.500	0.250	0.541	0.483	**0.479**	0.276	1.219	**0.832**	1.070	0.391
Traffic	96	0.764	**0.466**	0.486	0.410	1.360	0.791	**0.787**	0.574	1.238	0.761	**0.746**	0.624	1.613	**1.066**	1.262	0.736
	192	0.658	**0.484**	0.510	0.423	1.365	0.788	**0.742**	0.611	1.366	**0.832**	0.844	0.619	1.615	**1.028**	1.136	0.770
	336	0.825	**0.509**	0.515	0.436	1.376	0.794	**0.740**	0.623	1.299	**0.706**	0.715	0.604	1.624	**1.306**	1.389	0.861
	720	0.908	**0.539**	0.578	0.466	1.400	0.904	**0.761**	0.631	1.325	**0.786**	0.795	0.703	1.638	**1.475**	1.541	0.995
Weather	96	0.245	**0.212**	0.214	0.175	0.295	**0.273**	0.280	0.250	**0.293**	0.299	0.318	0.271	1.735	**0.671**	1.601	0.452
	192	0.264	0.241	**0.240**	0.217	0.318	**0.307**	0.319	0.266	0.361	**0.310**	0.341	0.315	1.950	**0.517**	1.874	0.466
	336	0.294	0.284	**0.283**	0.265	**0.367**	0.372	0.390	0.368	**0.384**	0.400	0.431	0.345	1.608	**0.536**	1.546	0.499
	720	0.374	**0.366**	0.372	0.324	0.428	**0.419**	0.426	0.397	**0.458**	0.472	0.485	0.452	1.234	**0.642**	1.256	1.260

E ALL RESULTS OF COLD-START FORECASTING

We simulate the cold-start forecasting tasks by reducing the training samples of each dataset to the last 1%. For example, when the look-back window and horizon are both 96, the 8640 data points in the training set of ETTh1 can form 8448(8640 - 96 - 96) training samples. We use the last 1% of training samples (the 8364^{th}-8448^{th}) for model training. In total, we only use 279(84 + 96 + 96) continuous data points. This is similar to the situation where we train a model to predict the sale curve of a new product based on just a few days' sale data.

In the main paper, we only present part of the results of FrAug in cold-start forecasting. Here we present all the results in the table 8. We can observe that FrAug consistently improves the performances of the model in cold-start forecasting by a large margin. In some datasets, i.e, exchange rate, models trained with FrAug can achieve comparable to those trained on the full dataset. Surprisingly, the performances of the models are sometimes better than those trained on the full dataset, i.e, Informer in Exchange rate and ETTh2. This indicates that FrAug is an effective tool to enlarge the dataset in cold-start forecasting.

Table 9: Best hyper-parameters for FrAug on all the datasets.

Model	PredLen	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
	Dataset	ETTh1				ETTh2				ETTm1				ETTm2			
Dlinear	Mask Rate	0.5	0.4	0.5	0.4	0.5	0.5	0.3	0.5	0.3	0.2	0.1	0.4	0.5	0.4	0.5	0.5
	Mix Rate	0.5	0.3	0.5	0.4	0.5	0.4	0.1	0.4	0.2	0.2	0.1	0.1	0.3	0.5	0.5	0.2
FEDformer	Mask Rate	0.4	0.2	0.2	0.5	0.5	0.5	0.4	0.5	0.5	0.1	0.5	0.3	0.5	0.5	0.5	0.5
	Mix Rate	0.5	0.5	0.1	0.5	0.5	0.5	0.4	0.5	0.5	0.1	0.5	0.5	0.5	0.4	0.5	0.4
	Dataset	exchange				electricity				traffic				weather			
Dlinear	Mask Rate	0.1	0.1	0.5	0.5	0.1	0.1	0.2	0.1	0.1	0.1	0.5	0.1	0.1	0.1	0.4	0.5
	Mix Rate	0.1	0.1	0.5	0.5	0.1	0.1	0.1	0.1	0.1	0.1	0.5	0.1	0.2	0.1	0.2	0.1
FEDformer	Mask Rate	0.4	0.5	0.5	0.5	0.1	0.2	0.2	0.2	0.4	0.5	0.5	0.5	0.3	0.5	0.4	0.5
	Mix Rate	0.4	0.3	0.2	0.2	0.1	0.1	0.1	0.5	0.3	0.5	0.2	0.1	0.3	0.5	0.4	0.5

F HYPER-PARAMETERS FOR FRAUG

FreqMask and FreqMix only have one hyper-parameter, which is the mask-rate/mix-rate, respectively. In all the experiments, the rate is selected from {0.1, 0.2, 0.3, 0.4, 0.5}. The best parameters for two models are shown in Table 9.