Synth-FAR: A Synthetic Frequency-Autoregressive Driven Framework for Time Series Forecasting

Anonymous authors
Paper under double-blind review

Abstract

Time series forecasting is essential for predicting future values based on observed patterns. Traditional methods perform well in in-domain scenarios with ample data but struggle with scarce data, leading to the rise of zero-shot and few-shot learning. Recent advancements use large-scale models but require extensive data and resources, often learning ineffectively from the available data. This study explores factors influencing effective learning in time series forecasting using Fourier analysis. Findings show that forecasters struggle with data containing multiple frequencies and generalizing to unseen frequencies. To address this, we introduce *Synth-FAR*, a synthetic data generation framework that enhances or replaces real data by creating a mixture of autoregressive and frequency information, improving model robustness in limited data scenarios. Our method outperforms other popular synthetic data techniques, such as Kernel-Synth, in both generation time and performance, and demonstrates the potential for integration into foundation model data pipelines, thereby enhancing their effectiveness.

1 Introduction

Time series forecasting (TSF) plays a critical role in various areas, such as finance, healthcare, and energy, where accurate predictions of future values are essential for decision-making and planning. Traditionally, in-domain learning has been the common setting for developing forecasting models, where a model is trained using data from the same domain it will later be deployed in (Salinas et al., 2020; Zhou et al., 2021). This ensures that the model captures the patterns, seasonality, and trends specific to the target domain, improving its predictive performance. However, a significant challenge arises when there is scarce or no historical information available for training, limiting the ability to apply traditional in-domain learning approaches (Sarmas et al., 2022; Fong et al., 2020). In such cases, the emergence of zero-shot (ZS) and few-shot (FS) learning settings offer potential solutions. Zero-shot learning enables models to generalize to new, unseen domains without requiring domain-specific data by leveraging knowledge transfer from other domains or tasks. Few-shot learning, on the other hand, allows fine-tuning on limited amounts of domain-specific data. In this paper, we focus mostly on the ZS TSF setting.

Zero-shot techniques for TSF are often built upon foundation models, which are pre-trained on vast amounts of diverse data and can generalize to a wide range of tasks (Das et al., 2024; Ansari et al., 2024). However, foundation models (FMs) face several challenges, such as their huge data requirements, high computational costs, difficulties in fine-tuning for specific applications, and the risk of model over-generalization, which can lead to sub-optimal performance on specialized tasks (Ekambaram et al., 2024; Liu et al., 2024b). Moreover, foundation models often struggle to fully exploit the train distribution, limiting their ability to capture domain-specific patterns crucial for accurate zero-shot time series forecasting (see Sec. 4). A potential approach to alleviate data and compute limitations is to train models, including non-foundational ones, on task-specific *synthetic* information (Dooley et al., 2024), thus eliminating real-data requirements and reducing compute time. Unfortunately, the factors that govern effective learning from synthetic data using non-foundation models remain unclear, and our work aims to advance the general understanding of this challenge.

We advocate that *Fourier analysis* is the natural framework for assessing the effectiveness of synthetic data in TSF (Yi et al., 2023). Fourier analysis decomposes a signal into its constituent frequencies, revealing periodic patterns, smoothing out noise, and identifying important frequency-based features (Körner, 2022). This helps understand overfitting and underfitting as forms of *frequency generalization* and *frequency adaptation*, respectively (see Sec. 4). Analyzing synthetic data through Fourier transforms clarifies how well the data captures true underlying patterns, leading to better adaptation and generalization while avoiding over-representation of irrelevant details. By harnessing Fourier analysis in time series forecasting with both non-foundational and foundational models, we identify several shortcomings. First, increasing the available frequencies in the train set while fixing those in the test set leads to inferior test results. Second, test performance improves with better alignment of train and test sets in the frequency space. Finally, foundation models overfit to certain frequencies, underperforming on general frequencies. Based on our findings, we propose that *the train set should span the predominant frequencies of the target domain*. While intuitive, this is often infeasible as the span of target frequencies is unknown. Instead, we design a method that generates lightweight, target-oriented synthetic data using a mixture of frequencies and autoregressive processes, with two configurations: 1) based on the sampling rate of the target domain, and 2) without any information on the target domain. We evaluate our approach, **Synth**etic **F**requency-**AR** (*Synth-FAR*), in zero-shot settings on recent state-of-the-art models, showing its effectiveness compared to other methods. Our main contributions include:

1. We analyze the importance of frequencies in time series models, especially in the context of generalization. We introduce the concepts frequency adaptation and frequency generalization, which facilitate the identification of potential challenges in ZS forecasting.
2. We propose a simple, easy-to-code, multi purpose and efficient time series synthetic generator that is effective for different tasks.
3. Through extensive tests, we demonstrate that our synthetic data achieves better (ZS) results than other synthetic datasets and complements foundation and non-foundation models in areas where they fall short.

2 Related Work

In-domain time series forecasting. For decades, non-deep statistical TSF models held the state-of-the-art (SOTA) status (Makridakis & Hibon, 2000; Makridakis et al., 2008), but in recent years, purely neural network-based SOTA approaches for TSF have emerged (Salinas et al., 2020; Oreshkin et al., 2020). Rapid development has led to various techniques including the usage of trend and seasonality (Zhou et al., 2022), patching time series (Nie et al., 2023), exploiting inter-channel relations (Liu et al., 2024c), and many others (Wu et al., 2021; Zhang & Yan, 2023; Xu et al., 2024). These approaches, however, have been considered in the in-domain setting, where there is an available train set that statistically aligns with the test set.

Zero-shot. While several attempts have been made in utilizing non-foundation models for ZS and FS TSF (Orozco & Roberts, 2020; Oreshkin et al., 2021; Jin et al., 2022), interest has quickly shifted to foundation models. Specifically Large Language Models (LLMs) are commonly considered, using various backbones including GPT-2 (Zhou et al., 2023; Liu et al., 2024b), LLaMA (Jin et al., 2024), and others (Gruver et al., 2024). Additional methods exploit trend-seasonality-residual decompositions (Cao et al., 2024) and Transformer blocks (Goswami et al., 2024). One of the main limitations of foundation models is their dependence on large volumes of data. Thus, recent studies have incorporated synthetic information alongside real data involving seasonal patterns and trends (Das et al., 2024) and Gaussian processes (Ansari et al., 2024). Closely related to our work is ForecastPFN (Dooley et al., 2024), where the authors perform zero-shot time series forecasting by training solely on synthetic data. Still, we argue that the factors determining effective learning in such settings remain unclear, particularly for non-foundation models.

Fourier analysis in time series applications. Fourier analysis and spectral theory are commonly used in various modern machine learning tasks (Yi et al., 2023). Incorporating frequency information has been done

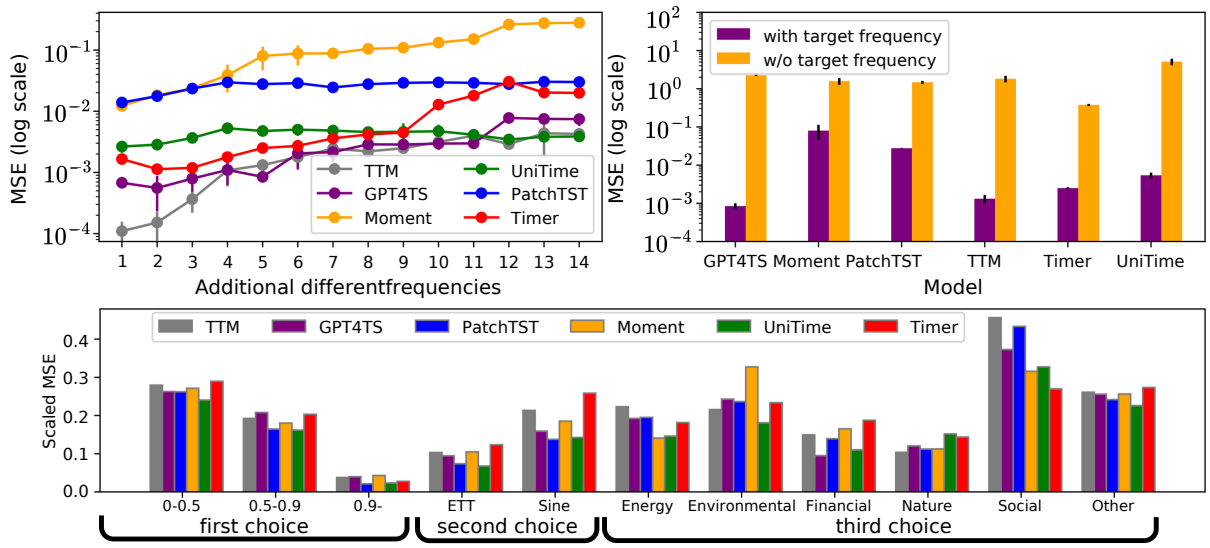

Figure 1: Top: We demonstrate poor frequency adaptation, where adding more frequencies gradually degrades performance (left). We also observe large performance differences when the fundamental target frequency exist vs. absent in the train set, implying poor frequency generalization (right). Bottom: Transfer learning performance bars for various frequency-based alignments between the train and test sets. Ranged values represent the periodogram PCC (left), ETT is a single dataset with different sampling rates (middle), and the remaining are sector based categories (right).

via compression (Rippel et al., 2015), data augmentation (Yang & Hong, 2022), and neural operators (Li et al., 2021). Examples in neural network design use real-valued (Xu et al., 2020) and complex-valued (Cao et al., 2020) representations. Other works span across anomaly detection (Ren et al., 2019), classification (Wang et al., 2018; Zhang et al., 2022), and time series forecasting (Zhang et al., 2017; Woo et al., 2022). Recent works have also introduced model design modification to support periodic pattern embedding (Ekambaram et al., 2024; Liu et al., 2024b; Cao et al., 2024), some of which are employed in our work. Closely related to our work are studies that considered synthetic data including many frequencies (Das et al., 2024; Ansari et al., 2024; Dooley et al., 2024). In this paper, we extend this research direction and harness Fourier analysis to study synthetic data and its effect on zero-shot time series forecasting.

3 Background

To motivate our analysis and approach to zero-shot and few-shot forecasting as discussed in Sec. 4, we present basic results related to Fourier analysis and time series information, see also (Shumway & Stoffer, 2000). It is well-known that for any time series sample $x_1, \ldots, x_n \subset \mathbb{R}^n$ and under carefully chosen coefficients, we have for odd n that

$$x_t = a_0 + \sum_{j=1}^{(n-1)/2} \left[a_j \cos(2\pi t\, j/n) + b_j \sin(2\pi t\, j/n) \right] \ , \tag{1}$$

for $t = 1, \ldots, n$, a_0 is the bias, a_j and b_j are the amplitude coefficients, and $t \in \mathbb{Z}$. The frequencies $\omega_j := j/n$ represent cycles per time unit, where a cycle is a complete period of the cosine or sine, e.g., for $\omega = 0.5$, the series makes two cycles per time unit. We also consider an equivalent form, obtained via a trigonometric identity of Eq. 1 and given by

$$x_t = a_0 + \sum_{j=1}^{(n-1)/2} A_j \cos(2\pi t\, \omega_j + \phi_j) \ , \tag{2}$$

where the amplitude $A_j = \sqrt{a_j^2 + b_j^2}$ and $\phi_j = \tan^{-1}(b_j/a_j)$ is the phase of the jth frequency, express the standard deviation and the cosine function starting point respectively. Notably, dominant periodic components in a signal are associated with larger amplitudes.

Another important concept for our work is the *periodogram* (Schuster, 1898). We define the scaled periodogram, which is closely related to the amplitude A_j, and it is defined via

$$P(\omega_j) = A_j^2 \ , \tag{3}$$

where large values of $P(\omega_j)$ correspond to predominant *fundamental frequencies* j/n, whereas small values of $P(\omega_j)$ can be viewed as noise. In practice, the scaled periodogram can be estimated via the discrete Fourier transform (DFT), which represents a weighted average of the data $d(\omega_j) = n^{-1/2} \sum_{t=1}^{n} x_t \exp(-2\pi i t\, j/n)$, with i the imaginary number. It follows that $P(\omega_j) = \frac{4}{n}|d(\omega_j)|^2$. Finally, *Harmonics* represent frequencies of the form $k\bar{\omega}_j$ for a dominant fundamental frequency $\bar{\omega}_j$, $k \in \mathbb{N}$. They appear in time series when non-sinusoidal components contribute to the structure of the signal. Below, we will show that harmonics, as depicted in the periodogram, are crucial in understanding the effect of data on zero-shot and few-shot learning and information transfer in large time series models.

4 Fourier Analysis for Zero-Shot TSF

Many existing approaches for zero-shot TSF are based on large foundation models (Ansari et al., 2024). These neural networks are computationally demanding and need large volumes of data for training. In this work, we aim to maximize the learning efficiency from data, with the goal of reducing data and compute requirements, especially in the ZS settings, where data is scarce or unavailable. Particularly, we are interested in answering the following overarching question:

What factors govern effective learning in zero-shot time series forecasting?

Understanding such factors better may lead to reducing data requirements by generating compact task-specific synthetic data. Similarly, compute reduction can be achieved by using non-foundation models on that data. Ultimately, if we succeed in answering the above question, we could potentially employ *non-foundation* models for solving zero-shot TSF, training solely on *synthetic data*.

4.1 Frequency adaptation and generalization

Toward uncovering the factors that determine effective learning, we examine time series forecasting through the lens of Fourier analysis. Specifically, to quantify the differences between the train and test sets and their corresponding forecasting errors, we will use the periodogram (see Sec. 3) and the following two new frequency-based concepts.

Definition 4.1 (Frequency Adaptation)**.** The model's ability to perform well in the case where the train set consists of the target frequencies along with other, unrelated, frequencies.

Definition 4.2 (Frequency Generalization)**.** The model's ability to perform well during inference on data with frequencies that were unavailable during training.

In other words, Def. 4.1 describes the model's difficulty in learning from multiple frequencies, where some may be unnecessary. It is closely related to *capacity*, introduced in (Han et al., 2024) to assess data fit, and *domain confusion* (Liu et al., 2024b) related to datasets from different domains. Def. 4.2, on the other hand, deals with the ability of a trained model to obtain consistent performance across learned as well as unseen frequencies. This definition is closely related to *domain generalization* (Wang et al., 2022), where there, the generalization is in the context of performing well on datasets of different domains.

Equipped with these definitions, we consider experiments, aiming to identify whether frequency adaptation and frequency generalization assist in understanding model behavior. For these experiments, we use recent non-foundation and foundation state-of-the-art (SOTA) TSF models including GPT4TS (Zhou et al., 2023), Moment (Goswami et al., 2024), PatchTST (Nie et al., 2023), TTM (Ekambaram et al., 2024), Timer (Liu

et al., 2024f), and UniTime (Liu et al., 2024b). The first experiment trains and infers the above models on a simple sine wave dataset with $\omega = 1/24$, representing an hour to daily based sampling rate. From here and throughout our discussion, we interchangeably use the terms sampling rate and frequency. Then, we incrementally add additional sine waves with various frequencies to the train set, re-train, and measure the prediction error of the $\omega = 1/24$ sine wave only. We plot in Fig. 1 (left) the mean squared error (MSE) of the prediction in log scale vs. the number of additional train frequencies. Notably, all models present an increase in test MSE, even if mild, as more frequencies are added, suggesting that they struggle with frequency adaptation.

In the second experiment, we used the same models. However, now every model is trained twice: on a mixed frequency sine wave train set including the target sine wave, and without it. We plot the test MSE errors in log scale and present the results in Fig. 1 (right). The bar chart shows that in all cases, having access to the target frequency sine wave (purple) leads to significant performance gains in comparison to training without that frequency (orange). This experiment suggests that deep TSF models may struggle to generalize to unseen frequencies, even on simple toy examples.

The third experiment explores the effect of similar and dissimilar frequency spaces in the context of transfer learning. Using the above models, we utilize a pool of datasets, see Fig. 13. We train each model separately on every dataset in the pool, and we use the learned model to infer over all the remaining datasets. To make the test MSE of different datasets comparable, we perform min-max normalization between train datasets per train setup as shown in Fig. 13. We organize the results into clusters based on the following choices: 1) The train and test sets share similar or dissimilar frequencies, as measured by the periodogram. 2) The train and test are sampled from the same dataset (ETT, Sine), but with different sampling rates (frequencies). Specifically, ETTh1 and ETTm1 are sampled at an hourly rate and per 15 minutes, respectively. 3) The train and test are from similar domains or sectors, e.g., energy-related information. We present in Fig. 1 the scaled MSE measures of this experiment for the abovementioned choices. Particularly, the three leftmost bar groups correspond to the Pearson correlation coefficient (PCC) of the periodogram between datasets (Choice 1). Then, ETT is the electricity transformer temperature dataset, and 'Sine' are sines waves of different rates (Choice 2). Finally, the six rightmost bar groups are various sector domains (Choice 3). The results show that while same-sector training may help (e.g., TTM on nature), the general performance is inconsistent. Similarly, using the same data (ETT) lowers the scaled MSE, however, it is still higher than training on datasets whose PCC is correlated in the frequency space (PCC ≥ 0.9).

4.2 AR(p) for complex frequencies

In certain datasets, unless the domain's dynamics are well understood, models may prefer to rely on the most recent values. For example, in datasets with complex frequencies or low frequencies, a limited input size may not be sufficient to capture the underlying frequency. Therefore, focusing on the most recent values of the input may be a better overall choice, which brings us to present the following concept.

An autoregressive process is based on the idea that the current value of the series, x_t can be explained as a function of p past values, where p determines the number of steps into the past needed to forecast the current value. Formally:

$$\begin{aligned} &x_t = w_t + x_{t-1}\gamma_1 + x_{t-2}\gamma_2, ..., x_{t-p}\gamma_p \ , \\ &\text{where,} \quad w_t \sim \mathcal{N}(0, \sigma_w^2) \ , \end{aligned} \tag{4}$$

By selecting an appropriate p, the model can better capture the underlying patterns in the data, leading to improved forecasting performance and most importantly, assist with frequency adaptation. This approach can be particularly beneficial for datasets with complex, low-frequency, and mixed frequency patterns, as it allows the model to adapt to the most relevant recent information, thereby enhancing its predictive accuracy, thus shifting its reliance on fixed frequency patterns.

To summarize, The above analysis reinforces the emergence of Fourier analysis as a key tool for determining the factors that affect effective learning. Moreover, it leads to the following straightforward observation: *training on datasets that share a similar periodogram with the target data improves the results of deep neural networks for TSF, in zero-shot scenarios and more generally.* Unfortunately, this observation is impractical

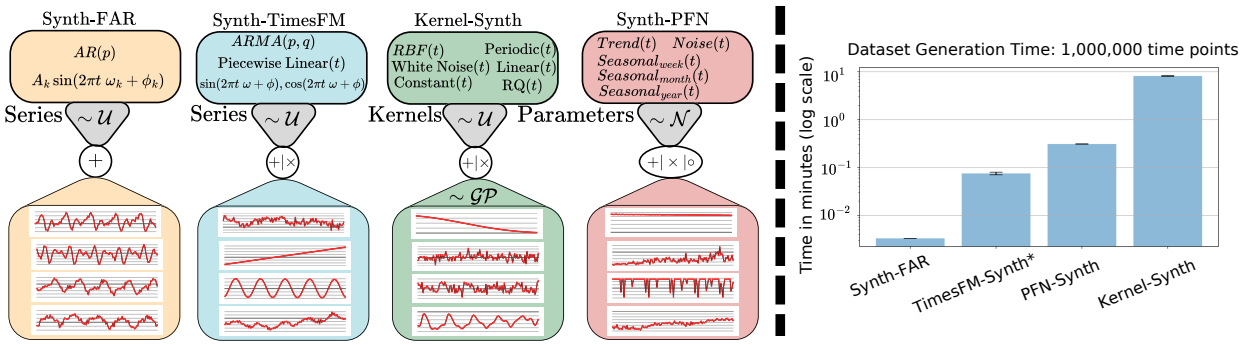

Figure 2: Left: Illustration of the generation frameworks for Synth-FAR, Synth-TimesFM, Kernel-Synth and Synth-PFN (left to right). Synth-FAR produces more distilled, frequency-focused patterns, while the remaining methods generate a broader and potentially overly diverse range of patterns, and requires more operations. Right: Dataset generation time.

to implement, as the full target frequency distribution is unknown, and even if the frequency is known, datasets with complex frequencies pose a challenge. To address this, we propose a synthetic dataset with two configurations: 1) mixture of frequencies and AR processes. 2) target frequency enhanced, with a simple heuristic given the sampling rate mixed with AR processes, which we found to be highly effective.

4.3 Synth-FAR: Synthetic time series based on fundamental frequencies and AR(p)

Following our analysis, we propose a simple yet effective approach to zero-shot TSF, which we refer to as *Synth-FAR*. Namely, we generate task-specific synthetic data and use it to train non-foundation and foundation models. Our approach has the potential to replace standard multi-dataset training of large time series models or serve as a complementary process. To generate data, we assume that the target distribution is mostly dominated by the fundamental target frequency and its harmonics augmented by varying trend and noise components. To create our synthetic dataset, we first generate a pool of sines and AR processes, followed by sampling from that pool and constructing various time series signals. We illustrate this approach in Fig. 2 in comparison to Synth-TimesFM and Kernel-Synth (Das et al., 2024; Ansari et al., 2024).

A pool of sines and AR(p). Given the fundamental target frequency $\bar{\omega}$, we create a pool P consisting of m_s sine waves whose frequencies are the harmonics of $\bar{\omega}$, i.e., where the amplitude is sampled from an exponential distribution $A_k \sim \text{Exp}(A')$, and the phase is drawn from a uniform distribution $\phi_k \sim \mathcal{U}([0, 2\pi])$. The frequencies ω_k are sampled from a uniform distribution over the harmonics. Namely, we have that $\omega_k \sim \mathcal{U}(\Omega)$, where $\Omega := \{\bar{\omega}, 2\bar{\omega}, \ldots, h\bar{\omega}\}$. Next we add to P a total of m_{ar} AR(p) processes, as defined in Eq. 4, where $\sum_{j=1}^{p} \gamma_j = 1$ and $\gamma_j = 1/p$. This formulation offers some interesting properties. If $p = 1$, it trains the model to become *naive*. Conversely, if $p = p_{max}$, which takes the maximal lookback, we can expect a *mean* model that returns the mean of the input. The variables m_s, h, A' and m_{ar}, p are hyper-parameters whose values are detailed and ablated in App. B. The formulation of P is defined as follows:

$$
\begin{aligned}
&P := \{s_s^1, s_s^2, \ldots, s_s^{m_s}, s_{ar}^1, s_{ar}^2, \ldots, s_{ar}^{m_{ar}}\} \ , \\
&\text{where,} \quad (s_s^k)_t := A_k \sin(2\pi t\ \omega_k + \phi_k) \ , \\
&\text{and,} \ (s_{ar}^k)_t = x_t^k := w_t + x_{t-1}^k \gamma_1 + x_{t-2}^k \gamma_2, ..., x_{t-p}^k \gamma_{t-p} \ .
\end{aligned} \tag{5}
$$

Dataset construction. We generate a synthetic multivariate time series $x \in \mathbb{R}^{d \times n}$ of d variates and length n, i.e., $x = (x_t^j)$, for $t = 1, \ldots, n$ and $j = 1, \ldots, d$, by sampling uniformly from P. In particular, we draw l series per variate j and sum them together. Formally,

$$
x^j = \sum_{i=1}^{l} s^i \ , \ \text{where,} \quad s^i \sim \mathcal{U}(P) \ , \ j = 1, \ldots, d \ . \tag{6}
$$

Table 1: Comparison between different synthetic data methods with the known target sampling rate (left block) and without it (right block). The mean over the datasets is given in the last row. Red bold, marks lowest in the entire row, black marks lowest in each block.

	Known Sampling Rate								Unknown Sampling Rate (mix)											
	Synth-FAR		Ker-Synth		FM		PFN		Synth-FAR		Ker-Synth		FM		PFN		Naive		Mean	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.452**	**0.439**	0.506	0.460	0.619	0.529	0.783	0.578	**0.529**	**0.484**	0.555	0.494	1.045	0.664	0.710	0.557	1.294	0.713	0.701	0.558
ETTh2	**0.327**	**0.358**	0.345	0.375	0.330	0.361	0.841	0.543	**0.338**	**0.374**	0.348	0.380	0.436	0.425	0.691	0.495	0.432	0.422	0.353	0.387
ETTm1	**0.397**	**0.393**	0.456	0.403	0.750	0.539	2.392	0.962	**0.576**	**0.480**	0.688	0.523	2.705	0.810	2.378	0.959	1.214	0.665	0.693	0.548
ETTm2	0.209	0.280	0.213	**0.276**	**0.206**	0.287	0.482	0.452	0.235	0.306	**0.226**	**0.302**	0.338	0.361	0.476	0.450	0.266	0.328	0.229	0.307
Electricity	**0.290**	**0.357**	0.318	0.381	0.440	0.478	0.704	0.497	**0.330**	**0.383**	0.471	0.492	1.154	0.840	0.580	0.469	1.588	0.945	0.846	0.762
Traffic	**0.941**	**0.549**	1.022	0.579	1.203	0.667	1.767	0.765	**1.190**	**0.650**	1.262	0.709	1.971	0.971	1.634	0.765	2.714	1.077	1.410	0.805
Weather	0.272	0.265	**0.223**	**0.248**	0.233	0.257	0.573	0.414	0.243	**0.269**	0.243	0.275	0.404	0.332	0.558	0.409	0.259	0.254	**0.215**	0.271
Average	**0.413**	**0.377**	0.440	0.389	0.540	0.445	1.077	0.602	**0.492**	**0.421**	0.542	0.454	1.150	0.629	1.004	0.586	1.110	0.629	0.635	0.520

The hyper-parameter l controls the number of series sampled from P, and it directly influences the correlation factor in that if l is larger, then more variates in the time series are correlated. To create a full dataset, we simply repeat the above process to generate multiple time series. For additional diversity, one can create several datasets with different parameters such as the number of harmonics.

We propose to construct several datasets each associated with a fundamental frequency $\bar{\omega}$, given a mix of natural fundamental frequencies or the fundamental frequency (a scalar) of a target distribution. In the latter case, We derive the fundamental frequency using the sampling rate of the target dataset, which is typically given as a co-variate and exploited by several models (Cao et al., 2024; Liu et al., 2024b; Ekambaram et al., 2024; Das et al., 2024; Liu et al., 2024d). See also App. C.1 for details on the relation between the sampling rate and the fundamental frequency, and ways to estimate it.

5 Experiments and Analysis

In this section, we evaluate Synth-FAR in comparison to recent state-of-the-art (SOTA) forecasting approaches in three settings: Synthetic data comparison (Sec. 5.1), real and synthetic data pre-training evaluations (Sec. 5.2), and lastly a comparison to prominent pre-trained foundation models (Sec. 5.3). We consider the popular long-term time series forecasting (LTSF) benchmark (Zhou et al., 2021; Wu et al., 2021) for the first two settings, and the recently published Time-MMD repository (Liu et al., 2024a) for the last. We observed that foundation models exhibit a tendency to overfit specific frequencies and their associated datasets, particularly in the LTSF benchmark. Frequency over-fit along with other technical limitations, may hinder their performance on other target datasets. More details regarding foundation models' limitations which motivate our approach are described in App. D.3. For all experiments in this work, a lookback of 96 is employed. Additional details including dataset information and Synth-FAR implementations are given in App. D and App. C, respectively.

The implementation details of Synth-FAR are as follows: For each sampling rate, we generated 3 Synth-FAR datasets, each corresponding to a different maximum harmonics parameter h as described in 4.3, specifically $h \in \{1, 2, 3\}$. From each generated dataset, we sample a total maximum of 10,000 sequences of size lookback+horizon, which are used for training or validation. For the known sampling rate setting, we create only 3 Synth-FAR datasets as described above. However, for the mixed setting, we employ 7 common frequencies: 1/12, 1/52, 1/7, 1/30, 1/24, 1/96, and 1/144, corresponding to monthly, weekly, daily-weekly, daily-monthly, hourly, 15 minutes, and 10 minutes, respectively, repeating the process above for each frequency. For AR(p), we use $p = 8$ for all experiments, with all weights set to $\frac{1}{p}$. We provide ablation studies in App. B and additional parameters and details in App. C.

5.1 Comparing synthetic dataset approaches

We evaluate forecasting results using synthetic data, comparing recent approaches such as TimesFM synthetic data (Das et al., 2024), ForecastPFN (Dooley et al., 2024), and Chronos Kernel-Synth (Ansari et al., 2024). TimesFM and Chronos use synthetic data to diversify real data for pre-trained models, while ForecastPFN trains solely on synthetic data. We generate 500 channels, each 1024 in length, for ForecastPFN, TimesFM, and Kernel-Synth to ensure diversity. We compare two setups: 1) Known target sampling rate, and 2)

Table 2: A comparison of zero-shot forecasting when training with real data vs. synthetic data. Including Synth-FAR outperforms real data in the all cases, with a notable advantage to Synth-FAR with the sampling rate denoted (**S**). The Lowest MSE for each model is marked with bold, and second lowest with an underline. The full table is given in Tab. 9.

Model	TTM				Timer				Moment				UniTime			
Data Type	**R**	**R+S(M)**	**S(M)**	**S**	**R**	**R+S(M)**	**S(M)**	**S**	**R**	**R+S(M)**	**S(M)**	**S**	**R**	**R+S(M)**	**S(M)**	**S**
ETTh1	0.636	**0.497**	0.569	0.521	0.722	0.603	0.639	**0.575**	0.68	0.714	0.553	**0.536**	0.95	0.574	0.537	**0.505**
ETTh2	0.438	**0.408**	0.43	0.418	0.601	0.604	0.455	**0.452**	**0.408**	0.422	0.436	0.424	0.562	0.526	**0.407**	0.422
ETTm1	1.066	**0.468**	0.472	0.477	1.268	1.17	0.681	**0.486**	0.842	0.718	0.676	**0.487**	1.133	1.114	0.601	**0.46**
ETTm2	0.337	**0.307**	0.313	0.313	0.414	0.414	0.354	**0.33**	0.326	**0.321**	0.336	0.33	0.407	0.382	0.317	**0.308**
Exchange	0.382	0.376	**0.358**	0.359	**0.35**	0.361	0.423	0.371	0.412	0.422	0.419	**0.382**	0.366	**0.359**	0.416	0.36
Electricity	0.499	**0.283**	0.321	0.304	0.516	**0.305**	0.344	0.32	0.768	0.862	0.38	**0.321**	0.573	**0.298**	0.31	0.303
Traffic	1.105	**0.845**	1.035	0.919	0.961	**0.913**	1.284	1.03	1.331	1.426	1.035	**0.931**	0.985	0.905	1.003	**0.872**
Weather	0.342	**0.31**	0.322	0.333	0.339	**0.337**	**0.337**	0.374	**0.284**	0.293	0.316	0.348	0.304	**0.296**	0.3	0.332
Average	0.601	**0.437**	0.478	0.456	0.646	0.588	0.565	**0.492**	0.631	0.647	0.519	**0.47**	0.66	0.557	0.486	**0.445**

Model	TimeMixer				GPT4TS				PatchTST				DLinear			
Data Type	**R**	**R+S(M)**	**S(M)**	**S**	**R**	**R+S(M)**	**S(M)**	**S**	**R**	**R+S(M)**	**S(M)**	**S**	**R**	**R+S(M)**	**S(M)**	**S**
ETTh1	0.763	0.573	0.59	**0.53**	0.615	0.592	0.635	**0.462**	1.958	1.119	0.538	**0.493**	0.621	0.549	0.704	**0.529**
ETTh2	0.624	0.57	0.428	**0.424**	0.525	0.488	0.44	**0.397**	0.617	0.586	0.416	**0.407**	1.01	0.609	0.493	**0.457**
ETTm1	1.335	1.069	0.659	**0.505**	1.188	0.79	0.74	**0.426**	1.243	1.114	0.572	**0.445**	0.943	0.782	0.682	**0.518**
ETTm2	0.417	0.401	0.341	**0.324**	0.393	0.356	0.342	**0.302**	0.448	0.443	0.32	**0.3**	0.964	0.54	0.386	**0.36**
Exchange	0.367	0.37	0.422	**0.361**	0.392	0.4	0.423	**0.367**	0.362	**0.359**	0.413	0.361	0.786	0.375	0.276	**0.236**
Electricity	0.396	**0.295**	0.329	0.308	0.442	0.316	0.366	**0.29**	0.649	**0.302**	0.323	0.304	0.413	0.34	0.388	**0.309**
Traffic	0.897	**0.884**	1.153	0.937	0.986	0.92	1.24	**0.824**	0.954	0.887	1.072	**0.87**	0.982	1.004	1.53	**0.93**
Weather	0.295	**0.294**	0.322	0.369	0.311	**0.309**	0.32	0.334	0.313	0.324	**0.305**	0.321	0.343	**0.294**	0.319	0.444
Average	0.637	0.557	0.53	**0.47**	0.606	0.521	0.563	**0.425**	0.818	0.642	0.495	**0.438**	0.758	0.562	0.597	**0.473**

Unknown target sampling rate, assuming no prior knowledge of the target domain. For the latter, we use a mixed variant of Synth-FAR, including datasets with common natural frequencies (e.g., 15 minutes, hourly, daily, and weekly) and their harmonics. For the synthetic datasets, we use their original implementation in Setup 2, and in Setup 1, we narrow their frequency space to the relevant fundamental frequency based on our heuristic given the sampling rate.

We detail in Tab. 1 the results, with the left and right blocks corresponding to known and unknown sampling rates, respectively. The MSE and MAE measures are averaged on a forecasting horizon of 96 across seven models, namely, TTM (Ekambaram et al., 2024), UniTime (Liu et al., 2024b), Moment (Goswami et al., 2024), Timer (Liu et al., 2024f), GPT4TS (Zhou et al., 2023), PatchTST (Nie et al., 2023), and TimeMixer Wang et al. (2024). As in Tab. 2. We also use the naive and mean forecasters as baselines for comparison. Notably, a large performance difference is observed between the left and right blocks, in favor of the known sampling rate case. Thus, utilizing the target sampling rate or the fundamental frequency facilitates the alleviation of poor frequency adaptation. When analyzing each block separately, we find Synth-FAR to be superior to the other baselines, presenting an average MSE reduction of **6.1%** vs. Kernel-Synth, and **23.5%** vs. (Times)FM in the left block, and an **9.2%** reduction vs. Kernel-Synth in the right block. Remarkably, while Synth-FAR trains on a approximately 50% of the data ForecastPFN, TimesFM, and Kernel-Synth use, it consistently obtains better error measures.

5.2 Real and synth data zero-shot forecasting

In this evaluation, we compare forecasting performances obtained by four settings. 1) **R**: Real data, 2) **R+S(M)**: Real data with Synth-FAR with mixed frequencies, 3) **S(M)**: Synth-FAR with mixed frequencies, and lastly 4) **S**: Synth-FAR with the sampling rate. In this experiment, we include the same models from the previous subsection also considering DLinear (Zeng et al., 2023). These models are either foundation models, designed for the zero-shot forecasting setting, or non-foundation models that marked significant milestones in developing such approaches. Inspired by TTM, we employ a slightly similar setup and train these baselines in the real data case on a subset of datasets from PEMS (Liu et al., 2022), and a subset of Monash (Godahewa et al., 2021), considering only datasets with a minimum length of approximately 1,000 timesteps. These datasets span a large range of sampling rates, including 4 seconds, 10 minutes, 1 hour, and more. Importantly, all the sampling rates of the evaluation data are also contained in this large train set, except for 15 minutes, which is the sampling rate of the ETTm datasets. We show in Tab. 2 the zero-shot forecasting results on several datasets (rows) as obtained by various methods (columns). We report the mean squared error (MSE),

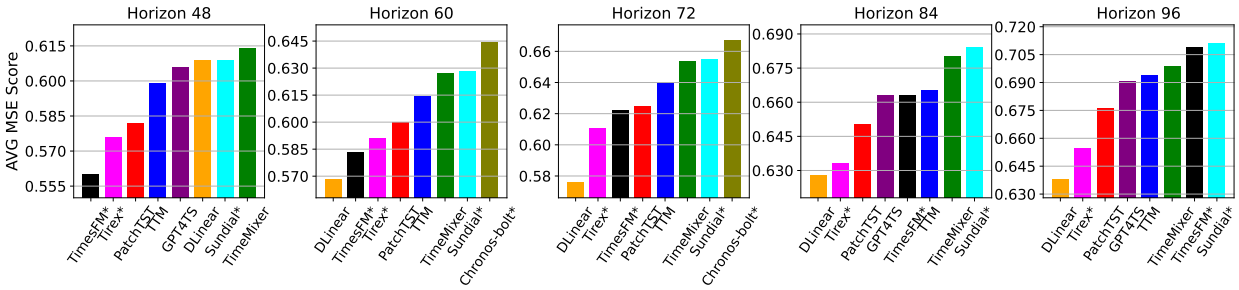

Figure 3: The average MSE performance across 10 Time-MMD datasets. The plots from left to right represent the performance with respect to the horizons 48, 60, 72, 84, and 96. Only the eight best-performing models are given in each plot.

where each result represents the average MSE over the forecast horizons $96, 192, 336, 720$ and three random seeds. The bottom row lists the average errors across all datasets. In App. 9, we provide the full table.

Notably, in most cases, performance gradually improves from **R** to **S**. This improvement is due to poor frequency generalization in **R**, as seen with ETTm1, which shows significant enhancements when Synth-FAR is included (e.g., 1.188 vs. 0.79 when selecting GPT4TS **R+S(M)**). Secondly, poor frequency adaptation is prevalent in different synthetic and real data settings, as more frequencies require attention, with an exception for the TTM model, which demonstrates superior performance in the **R+S(M)** setting. This can be attributed to the model's effectiveness in handling various data types and frequencies, highlighting the importance of using Synth-FAR to complement real data and improve performance. The majority of the lowest MSE scores and the lowest average MSE are observed in the **S** column, which suffers less from poor generalization and adaptation. This experiment is valuable as it underscores the pitfalls of using arbitrary datasets for training a large and general model for TSF. Specifically, we use a lookback of 96, which is the original standard lookback for the LTSF benchmarks (Zhou et al., 2021), and is not commonly used by pre-trained TSF foundation models for training and evaluation, where a lookback of 512 and above is standard but not always available. In the next subsection, however, we extend the topic of pre-trained foundation models.

5.3 Pre-trained foundation models

To demonstrate the effectiveness of Synth-FAR in areas where pre-trained foundation models may fall short, we employ a new benchmark comprised of the recently introduced Time-MMD dataset repository (Liu et al., 2024a). In this experiment, we use Synth-FAR with the sampling rate and the in-domain forecasting models DLinear, GPT4TS, TimeMixer, and PatchTST. The model TTM is also included as we find it notably effective. All mentioned models are trained solely on Synth-FAR. Additionally, we include the performance of prominent pre-trained models available online, namely Chronos (Ansari et al., 2024), MOIRAI (Woo et al., 2024), Timer (XL) (Liu et al., 2024e;f), and TimesFM (Das et al., 2024). Unfortunately, the pre-trained version of TTM is not suitable for a lookback of 96, highlighting some limitations of foundation models.

The results, shown in Fig. 3, exhibit the average MSE score across the 10 datasets for horizons 48, 60, 72, 84, and 96, for the 8 best-performing models. Detailed results are provided in App. 8. To differentiate between pre-trained foundation models and non-foundation models, we add the "*" symbol to mark pre-trained models. The results suggest that, in most cases, all synthetic-trained models perform consistently better on average than Timer*, Chronos*, and MOIRAI*, which are not shown as they are not in the best-performing group. Notably, DLinear outperforms TimesFM* for all horizons except for 48. This experiment suggests that although foundation models in TSF have made significant improvements in recent years, there is still room for improvement, and their performance in other domains remains unclear.

5.4 Data Generation time

Generating synthetic data introduces overhead to the computation pipeline. In what follows, we compare the generation time of different synthetic approaches. We test this by generating a single dataset with one million time points comprised of 1000 variates each of length 1000 for each of the methods, Synth-FAR, TimesFM, ForecastPFN, and Kernel-Synth. The time required by each method is 0.2 seconds for Synth-FAR, 4.5 seconds for TimesFM, 18.6 seconds for ForecastPFN, and 8 minutes for Kernel-Synth, presenting a significant advantage to Synth-FAR (see inset). For each reported result, we calculated the average creation time of three datasets. In addition to the generation time, we also note that Synth-FAR is easy-to-code, requiring only a few lines of code as presented in the code snippet in App. C.3.

6 Conclusion

Deep foundation models are often used for zero- and few-shot time series forecasting. However, our findings reveal key challenges in handling complex frequency patterns and limited data. Using Fourier analysis, we show that both foundation and non-foundation models struggle with learning from multiple frequencies (poor frequency adaptation) and generalizing to unseen ones (poor frequency generalization). To tackle these issues, we propose *Synth-FAR*, a synthetic data framework that enriches or replaces real data. Experiments show that Synth-FAR boosts performance and robustness in zero-shot settings for both model types. This work advances the understanding of frequency learning and provides a practical solution for low-data scenarios. It also opens avenues for improving models with Synth-FAR and similar techniques.

References

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. Tirex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. *arXiv preprint arXiv:2505.23719*, 2025.

Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33:17766–17778, 2020.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. TEMPO: Prompt-based generative pre-trained transformer for time series forecasting. *The Twelfth International Conference on Learning Representations, ICLR*, 2024.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning, ICML*, 2024.

Berken Utku Demirel and Christian Holz. An unsupervised approach for periodic source detection in time series. In *Forty-first International Conference on Machine Learning, ICML*, 2024.

Samuel Dooley, Gurnoor Singh Khurana, Chirag Mohapatra, Siddartha V Naidu, and Colin White. ForecastPFN: Synthetically-trained zero-shot forecasting. *Advances in Neural Information Processing Systems*, 36, 2024.

Vijay Ekambaram, Arindam Jati, Nam H Nguyen, Pankaj Dayama, Chandra Reddy, Wesley M Gifford, and Jayant Kalagnanam. TTMs: Fast multi-level tiny time mixers for improved zero-shot and few-shot forecasting of multivariate time series. *arXiv preprint arXiv:2401.03955*, 2024.

Simon James Fong, Gloria Li, Nilanjan Dey, Rubén González Crespo, and Enrique Herrera-Viedma. Finding an accurate early forecasting model from small dataset: A case of 2019-nCoV novel coronavirus outbreak. *Int. J. Interact. Multim. Artif. Intell.*, 6(1):132–140, 2020.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. In *Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Forty-first International Conference on Machine Learning, ICML*, 2024.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.

Lu Han, Han-Jia Ye, and De-Chuan Zhan. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations, ICLR*, 2024.

Xiaoyong Jin, Youngsuk Park, Danielle Maddix, Hao Wang, and Yuyang Wang. Domain adaptation for time series forecasting via attention sharing. In *International Conference on Machine Learning*, pp. 10280–10297. PMLR, 2022.

Thomas William Körner. *Fourier analysis*. Cambridge university press, 2022.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *9th International Conference on Learning Representations, ICLR*, 2021.

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

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. Sparsetsf: Modeling long-term time series forecasting with 1k parameters. *arXiv preprint arXiv:2405.00946*, 2024b.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, et al. Time-mmd: Multi-domain multimodal dataset for time series analysis. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024a.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. SCINet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022.

Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. UniTime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM on Web Conference 2024*, pp. 4095–4106, 2024b.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations, ICLR*, 2024c.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *arXiv preprint arXiv:2402.02370*, 2024d.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting. *arXiv preprint arXiv:2410.04803*, 2024e.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Transformers for time series analysis at scale. *arXiv preprint arXiv:2402.02368*, 2024f.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. *arXiv preprint arXiv:2502.00816*, 2025.

Spyros Makridakis and Michele Hibon. The m3-competition: results, conclusions and implications. *International journal of forecasting*, 16(4):451–476, 2000.

Spyros Makridakis, Steven C Wheelwright, and Rob J Hyndman. *Forecasting methods and applications*. John wiley & sons, 2008.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations, ICLR*, 2023.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: neural basis expansion analysis for interpretable time series forecasting. In *8th International Conference on Learning Representations, ICLR*, 2020.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. Meta-learning framework with applications to zero-shot time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9242–9250, 2021.

Bernardo Pérez Orozco and Stephen J. Roberts. Zero-shot and few-shot time series forecasting with ordinal regression recurrent neural networks. In *28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN*, pp. 503–508, 2020.

Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. Time-series anomaly detection service at Microsoft. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3009–3017, 2019.

Oren Rippel, Jasper Snoek, and Ryan P Adams. Spectral representations for convolutional neural networks. *Advances in neural information processing systems*, 28, 2015.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.

Elissaios Sarmas, Nikos Dimitropoulos, Vangelis Marinakis, Zoi Mylona, and Haris Doukas. Transfer learning strategies for solar power forecasting under data scarcity. *Scientific Reports*, 12(1):1–13, 2022.

Arthur Schuster. On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena. *Terrestrial Magnetism*, 3(1):13–41, 1898.

Robert H Shumway and David S Stoffer. *Time series analysis and its applications*, volume 3. Springer, 2000.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and S Yu Philip. Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering*, 35(8):8052–8072, 2022.

Jingyuan Wang, Ze Wang, Jianfeng Li, and Junjie Wu. Multilevel wavelet decomposition network for interpretable time series analysis. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2437–2446, 2018.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024.

Xiaozhe Wang, Kate Smith, and Rob Hyndman. Characteristic-based clustering for time series data. *Data mining and knowledge Discovery*, 13:335–364, 2006.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven C. H. Hoi. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *The Tenth International Conference on Learning Representations, ICLR*, 2022.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. *Forty-first International Conference on Machine Learning, ICML 2024*, 2024.

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.

Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1740–1749, 2020.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: modeling time series with 10k parameters. In *The Twelfth International Conference on Learning Representations, ICLR*, 2024.

Ling Yang and Shenda Hong. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International conference on machine learning*, pp. 25038–25054. PMLR, 2022.

Kun Yi, Qi Zhang, Longbing Cao, Shoujin Wang, Guodong Long, Liang Hu, Hui He, Zhendong Niu, Wei Fan, and Hui Xiong. A survey on deep learning based time series analysis with frequency transformation. *arXiv preprint arXiv:2302.02173*, 2023.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, pp. 11121–11128, 2023.

Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 2141–2149, 2017.

Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems*, 35:3988–4003, 2022.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations, ICLR*, 2023.

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

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

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

A Appendix

In this appendix, we provide additional information and details to supplement the main body of the paper. This includes ablation studies in App. B, supplementary method details in App. C, prominent foundation models limitations in D.3, datasets and models details in App. D, and extended tables and results in App. F that were not included in the main text. The purpose of this appendix is to provide readers with a more comprehensive understanding of the research methodology and results, as well as to offer further insights into the experimental procedures and analysis.

B Ablation and Additional Experiments

B.1 Ablation: AR(p) processes

To evaluate the effectiveness of the AR(p) model, we conducted an ablation study to compare the impact of different values of m_{ar} on the results. Specifically, we tested $m_{ar} = 0$, $m_{ar} = 10$, and $m_{ar} = 100$ on TTM, UniTime, Timer, and Moment with a forecast horizon of 96. The results are presented in Fig. 4. In the mixed frequency setting, for all datasets, the AR(p) processes demonstrate a positive effect, with a noticeable gradual decrease in MSE. This indicates that AR(p) acts as a conflicting frequency mitigator that enhances frequency adaptation. In the fixed sampling rate setup, where the sampling rate is predetermined, it is observed that AR(p) processes may sometimes degrade performance, as seen in the Electricity, ETTh1, and Traffic datasets. However, for the Weather dataset, increasing AR(p) improves performance in both setups. This improvement may be attributed to the limitations of the sampling rate heuristic or the presence of complex frequencies in the Weather dataset.

All datasets and configurations include AR(p) using the same m_{ar} parameters. Unfortunately, since we are working with zero-shot cases, we cannot alter these parameters for specific cases. However, if permitted, it is recommended to do so, as datasets with very strong seasonal components may suffer slightly due to the added random process.

We find that AR(p) is essential, based on the observations made in Section 4.2, if the frequency cannot be easily extracted by the model due to lacking frequencies, complex frequencies, or low frequencies that are not easily captured, then the other dominant factors are the last values of the input, which are generally correlated with the forecast output. Based on this observation, we found that AR(p) supports models with:

1. **Better Adaptation**: In a mixed frequency setting, AR(p) improves performance (see Fig 4).
2. **Forecast Stability**: In a mixed frequency setting, AR(p) lowers the output variance (see Fig 4 error bars).
3. **Regularization**: If the period estimation is not accurate, or the dataset has only low or complex frequencies, AR(p) aids in minimizing the error (see Fig 5. 4 Weather dataset). Additionally, AR(p) avoids over-fitting to specific frequencies.

B.2 Ablation: Harmonics

We conducted experiments to explore the impact of different harmonics on performance, as shown in Fig. 5. In this experiment, we created five configurations. Each configuration, except for the last, contains a training set consisting of three Synth-FAR generated datasets. The first four configurations consider only $h = 1$, $h = 2$, $h = 3$, and $h = 4$, respectively. The last configuration contains a mix of harmonics $h \in 1, 2, 3$, one for each generated dataset X. This configuration resembles the one used in this work.

If the sampling rate is known, considering $h > 1$ improves performance in all datasets except for the Weather dataset, most notably in ETTh1, Traffic, and Electricity. This suggests that Synth-FAR better assists the model in learning the structure of finer periodic signals, as shown in App. 12. The Weather dataset, on the other hand, sees performance degradation, which could be explained by the presence of complex signals,

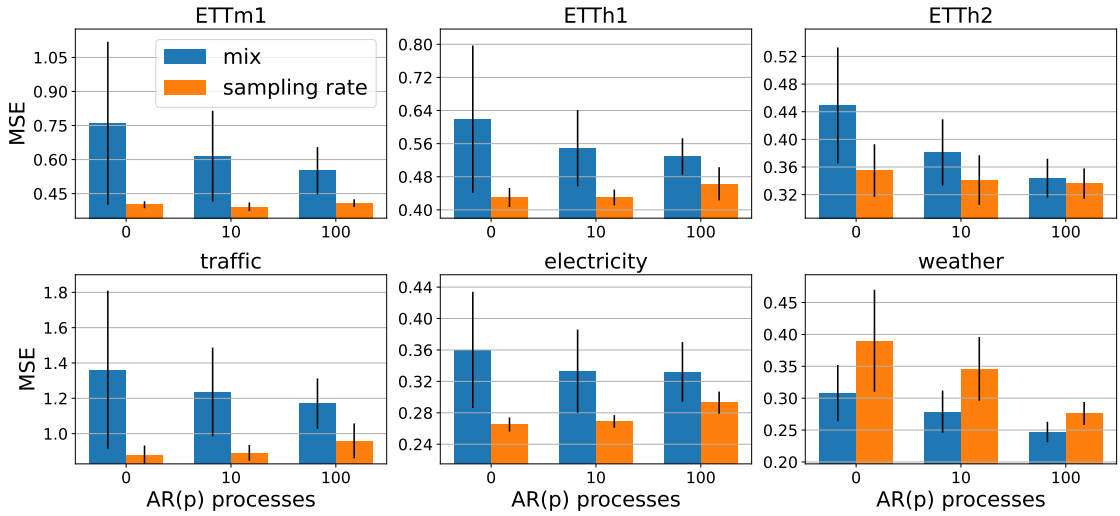

Figure 4: The influence of the number of series with AR(p) on the ZS performance per dataset, where each result reports the average MSE for TTM, UniTime, Timer, and Momentt for a forecast horizon 96.

low frequencies, or failure to approximate the fundamental frequency. In such cases, a mixed approach is better, as it may include the relevant frequency of interest and its harmonics, avoiding overfitting to a specific frequency.

In the mixed Synth-FAR setting, a gradual improvement with larger h is observed only for ETTm1, ETTh1, and Weather. The mixed frequency setting introduces more frequencies, leading to poorer frequency adaptation. In the original setting used in this paper, we mix harmonics parameters $h \in 1, 2, 3$ to allow different representations. This approach is shown to be better than the worst case for each dataset and relatively consistent in performance across different configurations.

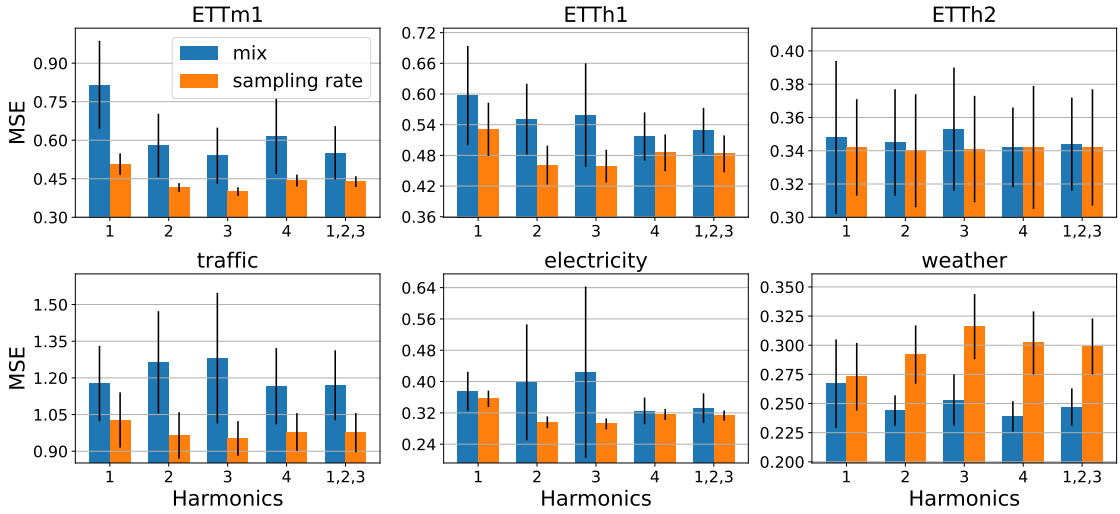

Figure 5: The influence of the number of harmonics on the ZS performance per dataset, where each result reports the average MSE for TTM, Timer, Moment, UniTime, and Moment for a forecast horizon 96.

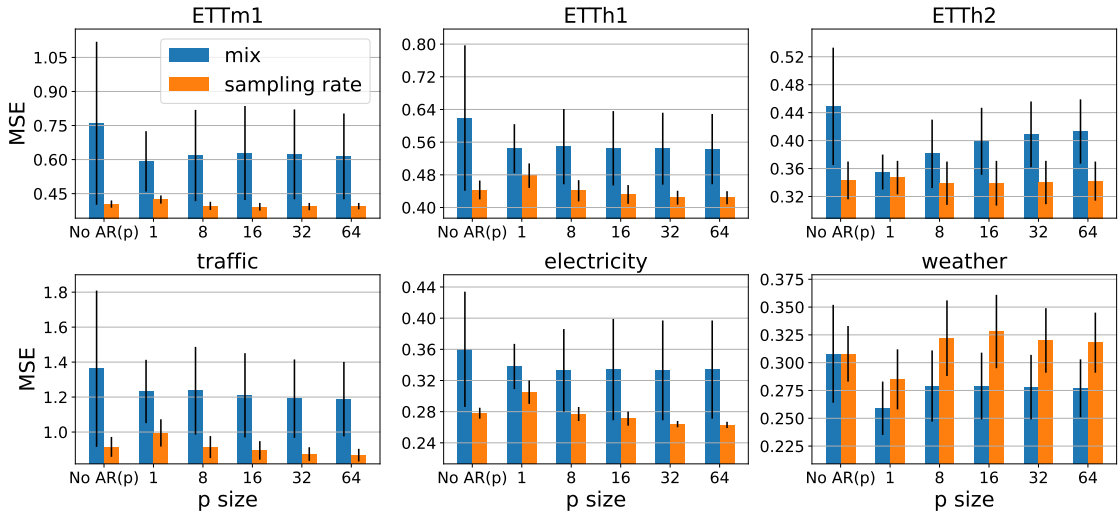

Figure 6: The influence of the p in AR(p) per dataset, where each result reports the average MSE for TTM, Timer, Moment, UniTime, and Moment for a forecast horizon 96.

B.3 Ablation: AR p size

In this subsection, we conduct experiments to test different values for p, which determines the size of the lookback to consider for the autoregressive process, as defined in 4. In this work, we select $p = 8$. In the sampling rate setting, not including AR at all exhibits inferior performance in most cases, with the exception of the Weather dataset. This inferiority becomes more apparent with the mixed frequency setting, where any form of $p > 1$ shows improvement. Similar to the other ablations, we do not find an optimal parameter p. For instance, $p = 1$ is preferable for Weather in both the mix and sampling rate settings. However, for Electricity and ETTh1, $p = 1$ is the worst option, even worse than not having AR(p) in the sampling rate setting.

B.4 Varying Lookbacks

We conducted additional experiments for the models TTM, Moment, and Timer using commonly employed lookbacks for evaluations: 192, 336, and 512, in addition to 96, as shown in Table 2 of the main text. For the models: TTM and Moment, their capacity increases as follows: 0.15M, 0.37M, 0.87M, and 7.18M for TTM, and 39.76M, 44.18M, 50.82M, and 58.92M for Moment, corresponding to lookbacks of 96, 192, 336, and 512, respectively. The results strongly suggest that injecting Synth-FAR with real data (**R+S(M)**) can significantly enhance performance for Timer and TTM, regardless of the lookback size or capacity. For Moment, the model appears to utilize the synthetic data with the sampling rate noted as **S** more effectively than other settings. In conclusion, Synth-FAR is shown to be effective for both larger capacities and lookbacks.

B.5 Embedding Frequency Knowledge in Time Series Models

To further evaluate the effectiveness of Synth-FAR, additional experiments were conducted comparing models trained exclusively on Synth-FAR—specifically DLinear (S) and TTM (S)—against SparseTSF, CycleNet, and TTM (Lin et al., 2024b;a; Ekambaram et al., 2024). These models incorporate architectural components designed to explicitly capture seasonal patterns, leveraging input frequency or sampling rate to enhance zero-shot performance. DLinear was selected as a baseline due to its structural similarity to SparseTSF and CycleNet, which also utilize linear transformations as a core modeling element.

The results, summarized in Table. 4, demonstrate that TTM (S) and DLinear (S) achieve lower mean squared error (MSE) in 6 out of 7 and 5 out of 7 cases, respectively. These findings suggest that Synth-FAR is competitive, and in many cases superior, even when compared to models that explicitly encode periodic structure.

Table 3: The MSE performance of utilizing S(**M**) for mixed Synth-FAR and **S** for frequency-specific Synth-FAR with varying lookback cases: 96, 192, 336, and 512. All settings employ a 96 steps horizon forecast. The model capacity of TTM and Moment increases as the lookback grows.

Model	TTM				Timer				Moment				Lookback
Data Type	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S	
ETTh1	0.592	**0.438**	0.522	0.459	0.66	**0.521**	0.576	**0.521**	0.64	0.701	0.547	**0.434**	96
ETTh2	0.357	0.324	0.337	**0.322**	0.52	0.531	**0.344**	0.368	**0.32**	0.353	0.382	0.323	
ETTm1	1.105	0.404	0.408	**0.398**	1.239	1.165	0.62	**0.422**	0.922	0.693	0.637	**0.422**	
ETTm2	0.236	**0.2**	0.207	0.203	0.314	0.316	0.25	**0.223**	0.232	**0.229**	0.261	0.232	
Electricity	0.484	**0.269**	0.31	0.289	0.438	**0.283**	0.336	0.313	0.687	0.846	0.384	**0.279**	
Traffic	1.098	**0.88**	1.087	0.961	0.963	**0.95**	1.378	1.097	1.251	1.41	1.149	**0.869**	
Weather	0.258	**0.242**	0.252	0.257	0.263	0.259	**0.254**	0.291	**0.197**	0.216	0.26	0.292	
Average	0.59	**0.394**	0.446	0.413	0.628	0.575	0.537	**0.462**	0.607	0.635	0.517	**0.407**	
Model	TTM				Timer				Moment				Lookback
Data Type	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S	
ETTh1	0.499	**0.436**	0.537	0.474	0.539	**0.464**	0.554	0.53	0.607	0.601	0.61	**0.459**	192
ETTh2	0.349	**0.32**	0.339	0.321	0.379	0.375	0.369	**0.367**	0.359	0.335	0.508	**0.331**	
ETTm1	1.026	0.404	0.411	**0.4**	1.369	0.652	0.484	**0.417**	1.327	0.79	0.53	**0.41**	
ETTm2	0.24	**0.184**	0.192	**0.184**	0.298	0.26	0.219	**0.212**	0.297	0.238	0.34	**0.213**	
Electricity	0.31	**0.24**	0.304	0.28	0.46	**0.261**	0.307	0.318	0.595	0.635	0.371	**0.305**	
Traffic	0.823	**0.763**	0.95	0.808	0.813	**0.78**	1.05	0.978	1.141	1.189	1.071	**0.793**	
Weather	0.279	**0.207**	0.215	0.215	0.228	**0.203**	0.205	0.224	0.219	**0.214**	0.311	0.227	
Average	0.504	**0.365**	0.421	0.383	0.584	**0.428**	0.455	0.435	0.649	0.572	0.534	**0.391**	
Model	TTM				Timer				Moment				Lookback
Data Type	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S	
ETTh1	0.435	**0.415**	0.53	0.497	0.504	**0.479**	0.525	0.512	0.59	0.553	0.763	**0.439**	336
ETTh2	0.312	**0.299**	0.322	0.318	0.391	0.396	0.393	**0.389**	0.382	0.364	0.546	**0.326**	
ETTm1	0.578	**0.399**	0.418	0.412	0.764	0.492	0.493	**0.434**	0.546	0.464	0.655	**0.381**	
ETTm2	0.208	**0.175**	0.185	0.182	0.26	0.234	0.219	**0.217**	0.265	0.233	0.331	**0.206**	
Electricity	0.251	**0.234**	0.306	0.289	0.293	**0.266**	0.317	0.334	0.553	0.509	0.576	**0.281**	
Traffic	**0.707**	0.714	0.952	0.88	**0.69**	0.743	0.897	0.901	1.083	1.051	1.418	**0.744**	
Weather	0.235	0.187	**0.183**	0.184	0.216	**0.2**	0.226	0.208	0.231	0.211	0.276	**0.203**	
Average	0.389	**0.346**	0.414	0.395	0.445	**0.401**	0.439	0.428	0.521	0.484	0.652	**0.369**	
Model	TTM				Timer				Moment				Lookback
Data Type	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S	
ETTh1	0.445	**0.424**	0.52	0.491	0.481	**0.465**	0.524	0.481	0.556	0.499	0.715	**0.412**	512
ETTh2	0.293	**0.281**	0.324	0.317	0.389	**0.378**	0.386	**0.378**	0.353	0.353	0.507	**0.328**	
ETTm1	0.703	**0.413**	0.44	0.434	0.734	0.476	0.516	**0.456**	0.565	0.443	0.668	**0.398**	
ETTm2	0.248	**0.174**	0.184	0.181	0.252	**0.219**	0.22	0.222	0.256	0.228	0.327	**0.204**	
Electricity	0.3	**0.253**	0.306	0.293	0.289	**0.251**	0.307	0.317	0.491	0.419	0.64	**0.265**	
Traffic	0.789	**0.762**	0.916	0.906	**0.716**	0.743	0.9	0.872	1.01	0.927	1.34	**0.726**	
Weather	0.203	**0.163**	0.176	0.176	0.219	**0.196**	0.233	0.218	0.218	0.201	0.272	**0.19**	
Average	0.426	**0.353**	0.409	0.4	0.44	**0.39**	0.441	0.421	0.493	0.439	0.638	**0.36**	

Table 4: The MSE performance of comparison of setups with sampling rate information. The models CycleNet (R), SparseTSF (R), and TTM (R) are trained on real data only, and offer architectural components that support periodic forecasting given the sampling rate. The models PatchTST (S) and TTM (S) are trained on synthetic data only. A lookback of 96 and a forecast horizon of 96 was used in this experiment

Dataset	CycleNet (R)	SparseTSF (R)	TTM (R)	PatchTST (S)	TTM (S)
ETTh1	0.571	0.506	0.592	**0.433**	0.459
ETTh2	0.320	0.329	0.357	**0.312**	0.322
ETTm1	1.112	1.101	1.105	**0.373**	0.398
ETTm2	0.253	0.253	0.236	**0.193**	0.203
electricity	0.422	0.405	0.484	**0.287**	0.289
traffic	1.124	0.955	1.098	**0.901**	0.961
weather	0.242	**0.209**	0.258	0.394	0.257

It is important to highlight that TTM is used extensively in this study, always with its resolution prefix tuning (RPT) mechanism enabled to incorporate frequency-specific information. Similarly, in subsection 5.3 and Figure 3 of the main text, the TimesFM model (Das et al., 2024) also adopts a comparable frequency-aware strategy. These results underscore the robustness of Synth-FAR, showing that its benefits persist even when evaluated alongside architectures designed for seasonal pattern recognition.

C Supplementary method details

C.1 Fundamental Frequency Estimation

The proposed Synth-FAR relies on prior information to overcome potential frequency confusion and improve frequency generalization. In this section we list different methods for obtaining this information required by our method.

Table 5: Synth-FAR hyper-parameters.

Notation	Selected Value	Description
A'	5	The expected amplitude for A_k generation.
$\bar{\omega}$	Depends on dataset, see C.1	The estimated fundamental frequency
$m_s,\ m_{ar}$	100	Number of sine waves and AR(p) processes for the pool P
h	$1, 2, 3$	Determines the maximum number of harmonics
l	$0.1 * \lvert P \rvert$	The number of sine waves from P used for the creation of x^j
n	$10,000$, 1024[1]	A fixed length for the all sine waves, AR(p) processes, and x
d	50	the number of variates for x
	$10,000$	number of examples sampled from X for training/validating

[1] For Time-MMD we use $n = 1024$.

Sampling rate information provides us with insight into the length between each collected time-step. This information can be further utilized to estimate the underlying fundamental frequency of the signal by bridging it to a common frequency. Common frequencies act as anchors, and attract a lot of valuable information due to their associations with strong natural or behavioral periods, namely, days, weeks, months, and years. For example, a behavioral signal could be linked to weekly or monthly patterns, as consumer behavior may vary on weekends or during specific months. This brings us to estimate the frequency based on the closest common frequency. For instance the sampling rates 5m, 10m, 15m, 30m, 1h are cycles of daily periods in which case the corresponding frequencies would be 1/288, 1/144, 1/96, 1/60, 1/48 and 1/24, respectively. A daily sampling rate is linked to a weekly 1/7 or monthly 1/30 (in our experiments we use a weekly rate). This continues for as long as a strong common frequency engulfs the sampling rate. This method is not free of challenges, for example the sampling rate may not be linked to the fundamental in cases where the fundamental frequency is unnatural, e.g. 1/100. Nevertheless, it may serve as a useful tool for frequency estimation.

Periodogram is a useful tool to estimate the spectral density of a signal (Schuster, 1898), given a sample series or a handful of samples one can utilize FFT to obtain the spectral density estimation. Given the spectral density estimation, the fundamental frequency is the lowest dominant frequency which is accompanied by harmonics (higher frequencies that are a positive integer multiple of the fundamental frequency), hence the periodogram has the potential to provide us with harmonics information as well as the fundamental frequency.

Prior frequency information, although not always intuitive, is another way of determining the fundamental frequency, this could be domain knowledge in a certain field as well as thorough analysis of the spectral density.

C.2 Synth-FAR hyper-parameters and implementation details

In what follows, we complement the description in Sec. 4 and provide details and descriptions of the parameters and their selected values presented in Tab. 5.

To implement Synth-FAR, we follow the next steps: 1) Create three datasets, each corresponding to $h = 1, 2, 3$ with the given parameters in Tab. 5. To create each dataset, using the code in App. C.3 with the relevant parameters is recommended. 2) Each channel is standardized, according to the LTSF protocol (Nie et al., 2023; Wu et al., 2021; Zhou et al., 2021), however, this may not be required depending on the use case or the model, since many models include instance normalization (Nie et al., 2023) as part of their architectural pipeline. 3) From each of the three comprised datasets, we draw $10,000$ samples only each of the lookback and horizon of interest, instead of using the entire dataset ($n{*}d$). Therefore, in practice we use only $10,000$ series from a single Synth-FAR datase, hence the final timestep size for training per-dataset is $(10,000 * (lookback + horizon))$. The reason we employ sampling is due to the same pattern repetition along the entire signal, concluding that the entire signal's length is not necessary.

C.3 Synth-FAR implementation

Here, we provide the python implementation for a single Synth-FAR dataset generation in C.3, covering the two steps described in Sec. 4.3.

```python
import numpy as np

# step 1, create the pool of sine waves
def create_sine_pool(m, n, A_avg, harmonics, w_fund):
    set_fund = [w_fund*i for i in range(1, harmonics+1) if w_fund*i<0.5]
    P_s = []
    t = np.arange(n)
    for k in range(m):
        A = np.random.exponential(scale=A_avg-0.01) + 0.01  # to avoid 0
        w = np.random.choice(set_fund)
        phi = np.random.uniform(0, 2*np.pi)
        s_k = A * np.sin(2*np.pi*t*w + phi)
        P_s.append(s_k)
    return np.array(P_s)

# step 1, create the pool of AR($p$)
def create_ar_pool(m, n, std=1, p=8):
    p_vec = np.ones(p) / p  # AR coefficients 1/p
    P_ar = []
    for k in range(m):
        s_ar = np.zeros(n)
        for j in range(1, n):
            lookback = min(j, p)
            s_ar[j] = np.dot(s_ar[j-lookback:j], p_vec[:lookback])
            s_ar[j] += np.random.normal(0, std)
        P_ar.append(s_ar)
    return np.array(P_ar)

# create the signal step 2
def create_synth(P, var, p_frac):
    m,n = P.shape  # number of series
    l = int(m * p_frac)
    X = []  # Synth-FAR sample dataset
    for i in range(var):
        idx = np.random.choice(m, l)
        s_i = np.sum(P[idx], axis=0)
        X.append(s_i)
    return np.array(X)

A_avg = 5 # average amplitude
w_fund = 1/24 # fundamental frequency
harmonics = 3  # harmonics
m_s = 100 # sine wave pool size

p = 8 # AR order
std = 1 # standard deviation of the noise
m_ar = 100 # AR pool size

var = 50 # number of variates
n = 250 # signal length
p_frac = 0.1 # determines the size of l, as a fraction of the pool size

P_sine = create_sine_pool(m_s, n, A_avg, harmonics, w_fund)
P_ar = create_ar_pool(m_ar, n, std, p)
P = np.vstack((P_sine, P_ar))

X = create_synth(P, var, p_frac)
```

C.4 Synth-FAR limitations

Unfortunately Synth-FAR is not always effective for all zero-shot scenarios. In cases where the there is a wider range of dominant frequencies, often leading to a signal with a higher degree of randomness (Demirel & Holz, 2024), capturing a single or even a handful of fundamental frequencies becomes challenging. In this particular case a single fundamental frequency based approach is not sufficient to represent the signal, and perhaps a mixed Synth-FAR approach is more suitable, as in Tab. 1. A particular case with the Exchange dataset is further discussed in App.F.1.

D Datasets and Models

In this section, we provide additional details regarding the experimental settings, models, and datasets. Each experiment was carried out three times with three different random seeds to ensure robustness and reliability of the results. Our objective was to maintain fidelity to the original parameters of each model, while establishing a unified framework for consistent comparison. Therefore, for each model we employ the original implementation with slight modifications to allow a fair comparison in a unified framework. Throughout the experiments, a lookback of 96 and a train,test and validation fraction split of 0.6,0.2,0.2 respectively for ETT datasets and 0.7,0.2,0.1 for the remaining datasets was employed for training in accordance with the original protocol for the LTSF (Informer) datasets (Zhou et al., 2021; Wu et al., 2021; Liu et al., 2024c). The reported results represent the test fraction of the data. All experiments were conducted with NVIDIA RTX3090 32GB GPU, and each experiment was trained end to end on a single GPU.

D.1 Models

In this work we selected the following models for evaluation:

- **PatchTST** (Nie et al., 2023). An in-domain TSF transformer-based model, introducing instance normalization, patching, a simple vanilla transformer and linear projection. PatchTST is a notable model, as many later released large-scale TSF models employ similar components including instance normalization, linear projection, patching, patch masking and reconstruction.
- **GPT4TS** (Zhou et al., 2023). A unified time-series model designed for a range of tasks including forecasting. GPT4TS uses a pre-trained frozen GPT-2 model, under the assumption that language domain data could be adapted to time-series data. GPT4TS is an important milestone towards foundation models as it showed success with employing a unified pre-trained language transformer for a range of downstream tasks with fine-tuning.
- **TTM** (Ekambaram et al., 2024) is a pre-trained model with a light-weight architecture which utilizes diverse resolution sampling with the implementation of patches of different lengths and resolution prefix tuning, allowing the model to encode sampling rate specific information.
- **Timer** (Liu et al., 2024f) employs a GPT-style architecture, originally designed for a range of tasks such as imputation, anomaly detection, and forecasting. Although originally designed for auto-regressive next token prediction, we employ a non-autoregressive setup in this work, aligning our evaluation with other models.
- **Moment** (Goswami et al., 2024). A transformer-based foundation model for time series, designed for various downstream tasks such as forecasting, classification, imputation, and anomaly detection. Moment utilizes transformers, patching, and learnable mask embedding.
- **TimeMixer** (Wang et al., 2024). A fully MLP-based architecture with Past-Decomposable-Mixing (PDM) and Future-Multipredictor-Mixing (FMM) blocks. PDM decomposes and mixes seasonal and trend components, while FMM combines multiple predictors for enhanced forecasting.
- **UniTime** (Liu et al., 2024b). A cross-domain large forecasting model empowered by a trainable GPT-2 backbone. UniTime also employs masking for generalization and to increase convergence

speed. Language prompts are also utilized for identification information for training. However, we find in our implementation that this contribution hurts performance in ZS, therefore we do not provide "domain-instructions" as suggested in the original paper.

- **Chronos** (Ansari et al., 2024). Chronos is a framework for pretrained probabilistic time series models that tokenize time series values and trains transformer-based language models. It uses the T5 family of models, pre-trained on a mix of public and synthetic datasets, namely Kernel-Synth, to enhance generalization.
- **DLinear** (Zeng et al., 2023). DLinear decomposes raw data into trend and seasonal components using a moving average kernel. It applies one-layer linear models to each component and sums the results for the final prediction, enhancing performance when a clear trend is present.
- **TimesFM** Das et al. (2024). TimesFM for time-series forecasting uses a decoder-only architecture with input patching, breaking time-series into patches to improve performance and inference speed. It employs a large-scale corpus of real-world and synthetic data for pretraining. The model predicts longer output patches than input patches, enhancing accuracy for long-horizon forecasting. Patch masking is used during training to handle various context lengths.
- **MOIRAI** Woo et al. (2024) uses a patch-based approach with a masked encoder architecture for time series modeling, flattening multivariate series into a single sequence. It employs a multi-patch size input projection layer and a multi-patch size output projection layer to decode tokens into mixture distribution parameters. The core Transformer module is an encoder-only architecture with various improvements, including pre-normalization, RMSNorm, query-key normalization, and SwiGLU non-linearity.
- **TiReX** (Auer et al., 2025) is a decoder-only pre-trained time series forecasting model built on xLSTM blocks that enable efficient recurrence and strong state tracking. The model converts normalized time series into non-overlapping patches, embeds them into tokens processed by stacked xLSTM layers, and outputs quantile forecasts for probabilistic prediction. During training, it uses Contiguous Patch Masking (CPM) and data augmentations to improve long-horizon forecasting and robustness, enabling strong zero-shot generalization across unseen datasets.
- **Sundial** (Liu et al., 2025) a decoder-only Transformer-based time series foundation model that operates directly on continuous-valued patch tokens, avoiding discrete tokenization. It models the full predictive distribution of future patches using a generative objective called TimeFlow Loss, which applies flow-matching to transform Gaussian noise into future forecasts conditioned on Transformer representations of the historical context.

Although the given models offer a limited scope with respect to the available models, we selected these baselines for several reasons: 1) code availability for training: an easy access to trainable, original implementations through Hugging Face or github. 2) Performance and time efficiency: these works offer a thorough comparison to other comparable methods and showed better overall performance including faster inference or training time. For example, TTM's superior inference speed compared to other models (Ekambaram et al., 2024). 3) Prominence: for example PatchTST and GPT4TS are important milestones towards foundation models for TSF, due to their popularity and architectural contributions. These reasons eventually guided our decision making toward model selection for evaluation.

D.2 Datasets

In this work, we evaluate the proposed Synth-FAR on the common LTSF (Zhou et al., 2021) benchmark datasets and the recently introduced Time-MMD repository (Liu et al., 2024a). We train the baseline models in Tab. 2 and on a subset of datasets from the Monash repository (Godahewa et al., 2021). Specifically, we select the datasets that have a minimum length of 1,000 timesteps, in order to enable a training configuration of horizon length 720, which requires each example to be 846 timeseps long. The PEMS repository (Liu et al., 2022) is also included for training. This training setup is slightly similar to the one employed in (Ekambaram

et al., 2024). In Tab. 6, we provide details regarding the selected datasets for training and testing. To ensure that certain large datasets do no dominate training, we limit the maximum number of examples per dataset to 500,000 for training and validation. Selecting a subset of the entire training set is also a common practice in large unified training frameworks (Ekambaram et al., 2024; Liu et al., 2024f). In our case, it can prevent large datasets with many examples to engulf the effect of smaller and medium size datasets during training. The given train datasets cover a range of sectors such as nature, energy, traffic and financial and various sampling rates. Most sectors and sampling rates of the evaluation datasets are included in the train data for the experiments in 5.2, except for the sampling rate for ETTm1 and ETTm2. For the Time-MMD datasets used in this work for ZS only, each variate is normalized with standard scaler separately.

In this work we focus on datasets with a variety of seasonal prominence. Not all datasets in our evaluations exhibit purely seasonal patterns; each dataset possesses a different degree of seasonality. This is illustrated in Figure 7, where we have included a *measure of seasonality* (Wang et al., 2006) and sample series from every evaluated dataset used in this work. It is evident that while datasets like Traffic and electricity display strong seasonal patterns, others such as ETTm2 and Climate have weaker seasonal components.

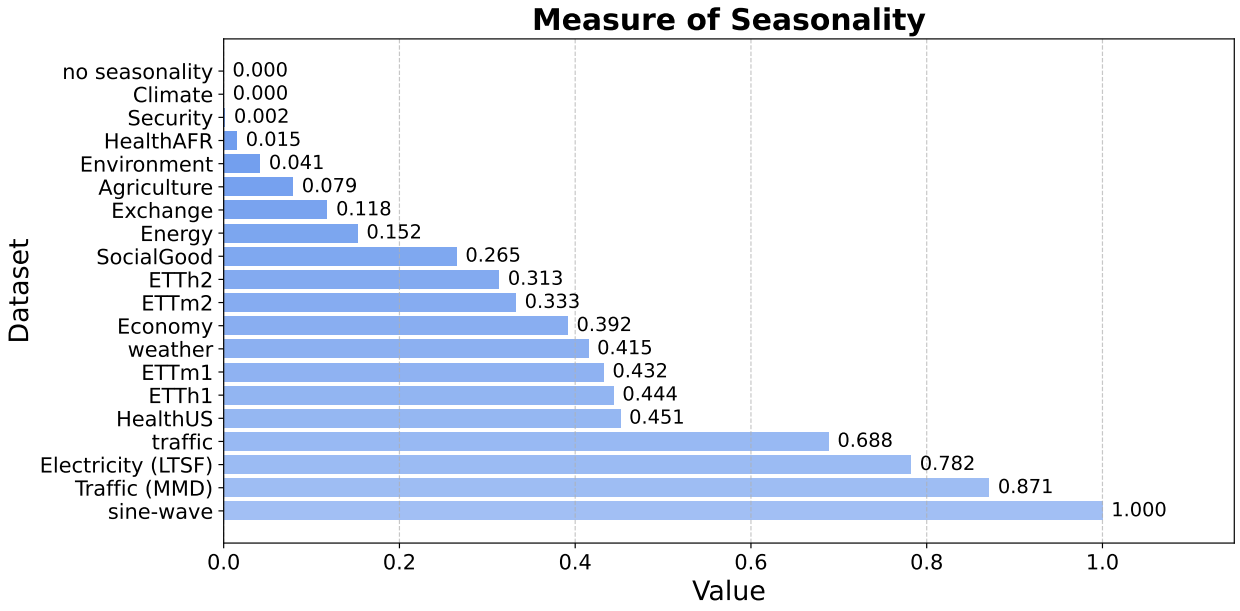

Figure 7: The *Measure of Seasonality* metric of each dataset used in this work. The score 0 represents no seasonality, whereas 1 represents a sine-wave without noise. Formulations: $1 - \frac{VAR(noise)}{VAR(noise+seasonal)}$, where *noise*, *seasonal*, are extracted with STL decomposition, and VAR returns the series variance.

D.3 Synthetic Datasets

In what follows, we provide additional details regarding the synthetic datasets discussed in Sec. 5.1.

- **TimesFM** (Das et al., 2024): Synthetic generated data, where each channel selects up to three possible components that are eventually added together, or multiplied (trend only) among ARMA process, mixture of cosines and sines, and piece-wise linear trends. In this work, we provided results based on our implementation as the original implementation is not available.

- **ForecastPFN** (Dooley et al., 2024): They assume that there exists a shared distribution among real time-series datasets, which can be derived from natural periodic data, trend, global trends and noise. ForecastPFN synthetic data applies multiplication and addition to create signals that meet

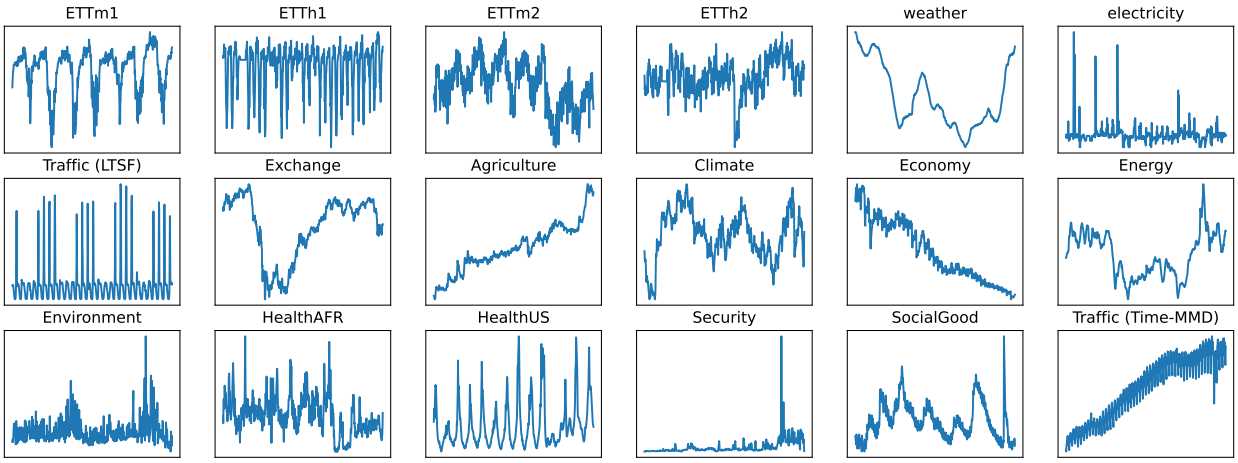

Figure 8: Examples of the different datasets used for evaluation in this work.

Table 6: Details on the considered datasets.

Dataset	Repository	Channels	Min/max channel length	Sampling rate	Sector	Usage
ETTh1	LTSF	7	17,420	hourly	Energy	Evaluation
ETTh2	LTSF	7	17,420	hourly	Energy	Evaluation
ETTm1	LTSF	7	69,680	15 minutes	Energy	Evaluation
ETTm2	LTSF	7	69,680	15 minutes	Energy	Evaluation
Electricity	LTSF	321	26,304	hourly	Energy	Evaluation
Traffic	LTSF	862	17,544	hourly	Transport, Environmental	Evaluation
Weather	LTSF	21	52,696	10 minutes	Nature	Evaluation
Exchange	LTSF	8	7,588	daily	Financial	Evaluation
London Smart Meters	Monash	5,560	288/39,648	30 minutes	Energy	Training
Aus. Electricity Demand	Monash	5	230,736/232,272	30 minutes	Energy, Environmental	Training
Wind Farms	Monash	339	6,345/527,040	minutely	Energy	Training
Bitcoin	Monash	18	2,659/4,581	daily	Financial	Training
KDD Cup 2018	Monash	270	9,504/10,920	hourly	Nature, Environmental	Training
Weather (Monash)	Monash	3,010	1,332/65,981	daily	Nature	Training
Solar	Monash	137	52,560	10 minutes	Nature	Training
Sunspot	Monash	1	23,741	daily	Nature	Training
Us Births	Monash	1	7,305	daily	Nature	Training
Saugeen River Flow	Monash	1	23,741	daily	Nature	Training
Solar Power	Monash	1	7,397,222	4 seconds	Energy	Training
Wind Power	Monash	1	7,397,147	4 seconds	Energy	Training
PEMS03	PEMS	358	25,887	5 minutes	Transport	Training
PEMS04	PEMS	307	16,992	5 minutes	Transport	Training
PEMS07	PEMS	883	28,224	5 minutes	Transport	Training
PEMS08	PEMS	170	17,856	5 minutes	Transport	Training
Agriculture	Time-MMD	1	496	Monthly	Agriculture	Evaluation
Climate	Time-MMD	5	496	Monthly	Nature	Evaluation
Economy	Time-MMD	3	423	Monthly	Financial	Evaluation
Energy	Time-MMD	9	1479	Weekly	Energy	Evaluation
Environment	Time-MMD	4(1)	11102	Daily	Environmental	Evaluation
Health US	Time-MMD	11(7)	1389	Weekly	Social, Healthcare	Evaluation
Health AFR	Time-MMD	1	1459	Weekly	Social, Healthcare	Evaluation
Security	Time-MMD	1	297	Monthly	Social	Evaluation
Social Good	Time-MMD	1	900	Monthly	Social	Evaluation
Traffic	Time-MMD	1	531	Monthly	Transport	Evaluation

* The brackets (x) mark the number of variates used for forecasting in practice.

their prior distributions criteria. To handle extreme scales in their generated data, they introduce robust scaling and outlier removal, which is also employed for ForecastPFN in Tab. 1.

- **Kernel-Synth** (Ansari et al., 2024): a Gaussian process (GP)-based synthetic time series generation method. Kernels are sampled from a kernel bank and then randomly combined using a binary operator ($\times$ or $+$). The resultant kernel is used in a GP prior to the generation of a synthetic time series.

In each of these methods, the generated channels are independent of the other channels, yet they attain cross-channel relations such as correlations and causality due to the underlying generation process. Synth-FAR on the other hand, supports multivariate channels with a controllable degree of linear similarly (Pearson correlation) through the parameter l. In Tab. 1, each synthetic dataset was standardized per channel except for ForecastPFN which was scaled with a robust scaler in accordance with the original implementation.

E Pre-trained foundation models limitaitons

In this section, we further expand the discussion about frequency generalization and frequency adaptation discussed in Sec. 4. Here, we consider pre-trained models, obtained from the original repositories of TimesFM (Das et al., 2024), Timer (Liu et al., 2024f), TTM (Ekambaram et al., 2024), and MOIRAI (Woo et al., 2024) with lookback values of 512 for TTM, TimesFM, MOIRAI, and 672 for Timer were utilized, to align with the specifications of the original trained models available online. To test whether the given models can generalize well, we evaluated their performance on simple periodic signals, with one to four harmonics of different frequencies. The results are depicted in Fig. 9, where the left plots detail the test MSE in log scale as a function of the frequency, and the right plots show examples of the evaluated signals. We find that models achieve reasonable errors on the 1/24 frequency and its 2-harmonic frequency 1/12, where the red dashed lines are positioned. This could be explained by the amount of pre-training data associated with the 1/24 frequency, which often relates to an hourly sampling rate. For example, hourly sampled data accounts for $> 62\%$ of the pre-training datasets of TimesFM (Das et al., 2024). On the other hand, when evaluated on less common frequencies, we observe a significant performance degradation with MSE values getting closer to 1 (left, top) and even higher. This behavior becomes more apparent when the number of harmonics is greater than 1 (left, bottom). We also show the forecast predictions of individual signals for the over-fitted 1/24 frequency in Fig. 10, and for the under-fitted 1/25 frequency in Fig. 11. This analysis complements our frequency-based analysis above, suggesting that large time series forecasting models suffer from poor frequency adaptation and attain poor frequency generalization.

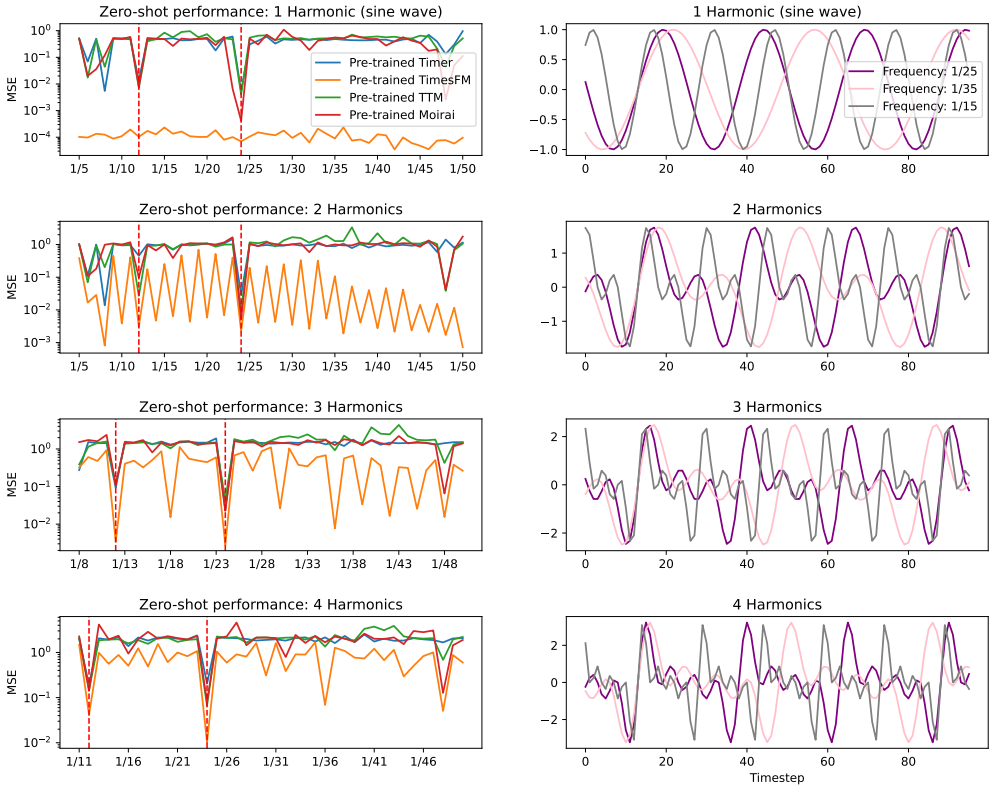

Figure 9: Zero-shot performance of pre-trained models on signals with 1 to 4 harmonics. The models perform well on the 1/24 and 1/12 frequencies, but for the remaining frequencies, the performance decreases significantly.

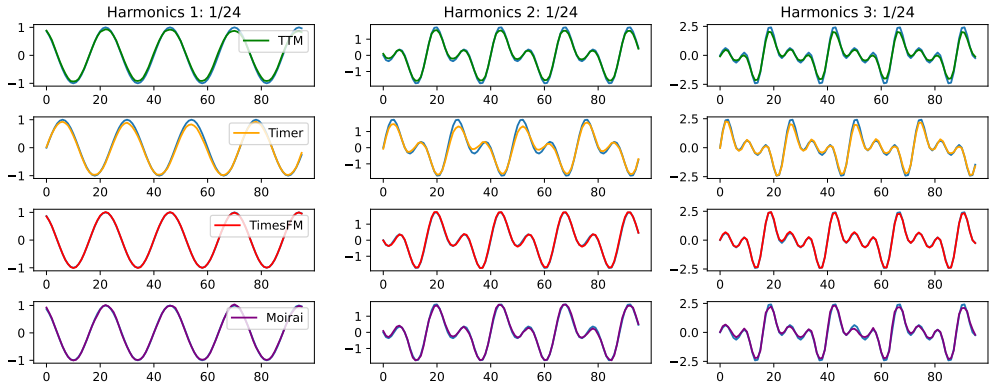

Figure 10: Forecast depiction of a 1/24 periodic series with different harmonics

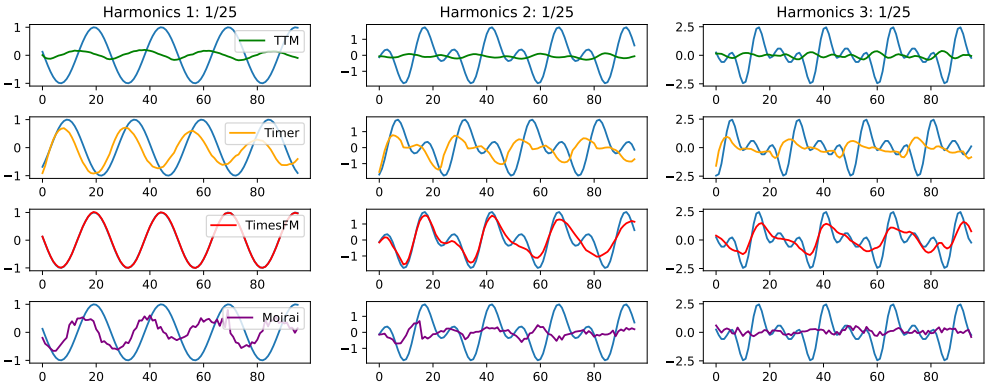

Figure 11: Forecast depiction of a 1/25 periodic series with different harmonics

F Extended experiments and results

In this section, we provide additional depictions and tables that expand the experiments in the main body. The Fig. 13 depicts the Pearson correlation coefficient (PCC) between every pair of datasets included in Fig. 1. Values closer to 1 represent a higher Periodogram similarity.

F.1 Periodogram

The periodogram is often mention in this work as an effective tool for the analysis of a signal to extract the fundamental frequency. In Fig. 12, we provide examples of the periodograms of Exchange, Traffic and Electricity, where it is shown that the periodogram of Traffic and Electricity are very similar, with an identical fundamental frequency 1/24, with apparent harmonics, suggesting that they are potential candidates for successful transfer learning. Traffic and Electricity are both mentioned in the high periodogram correlation category in Fig. 1, and also shown in the periodogram correlation matrix Fig. 13 with PCC > 0.9. On the other hand, Exchange has a wider spread of significant frequencies, e.g., $(0, 0.025]$, hence, being characterized with more randomness without clear dominant fundamental frequencies. Consequently, Synth-FAR fails to perform well on Exchange, since the estimated fundamental frequency does not represent the true spectral span of the data.

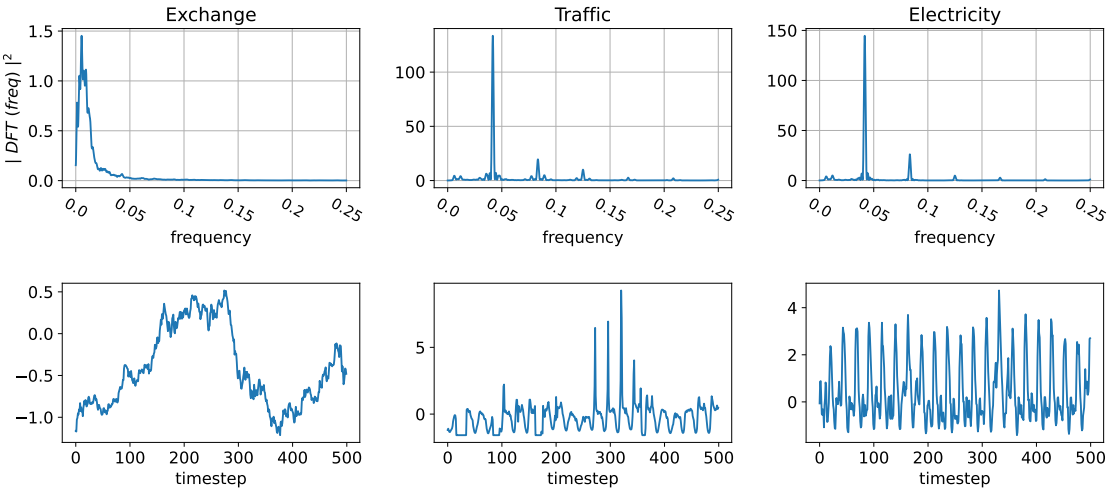

Figure 12: Top: Periodogram of the frequency range $(0, 0.25]$ for the datasets Exchange, Traffic, and Electricity visualized from left to right. Bottom: A random example from each dataset.

F.2 Full experimental tables

In this subsection, the full table results from the experiments section in the main text are given.

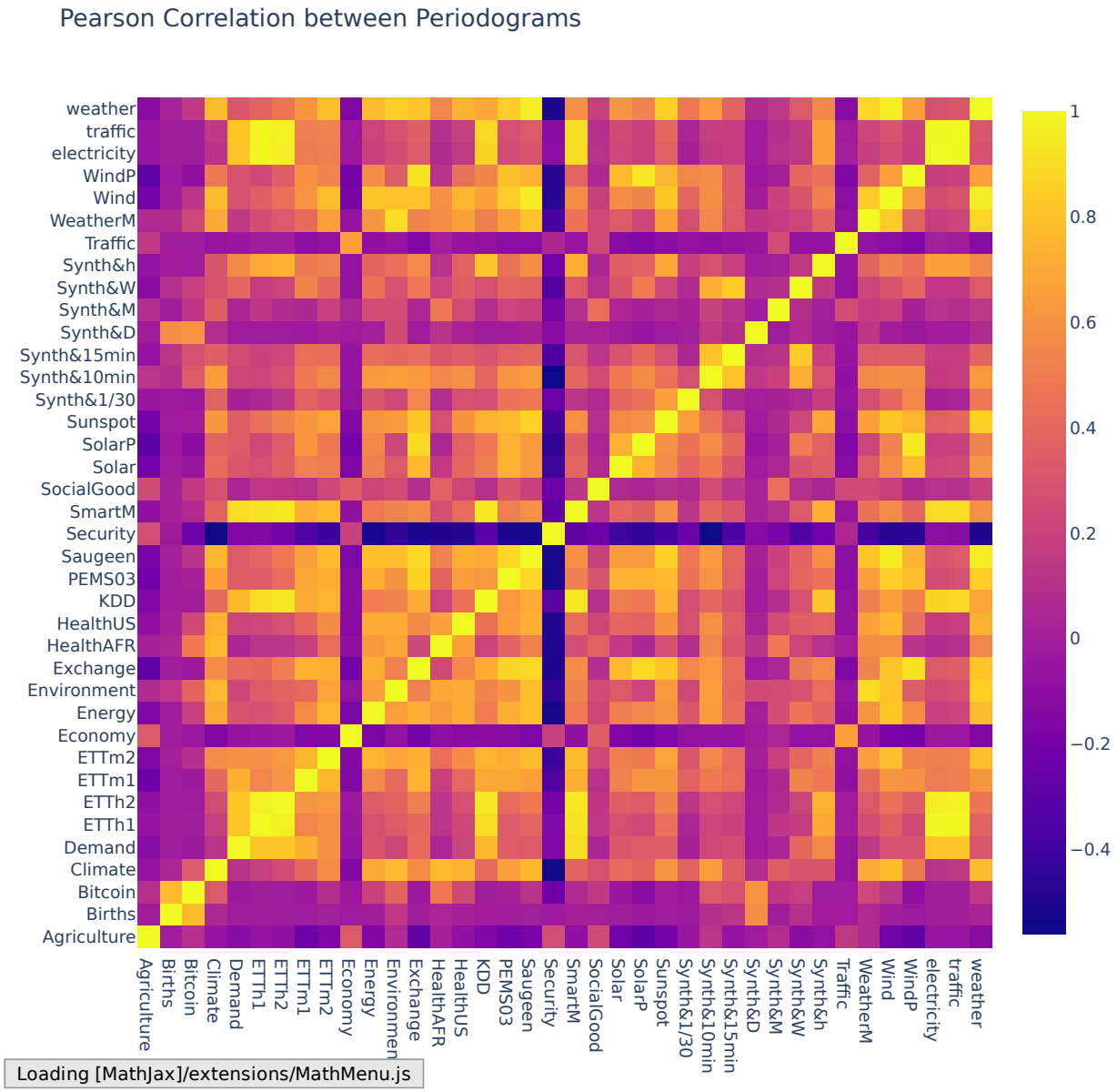

Figure 13: Pearson correlation score between the periodograms of each pair. Before computing the periodogram, a moving average is extracted and removed from the signal. Synth&X denotes a sinewave of a specific frequency X.

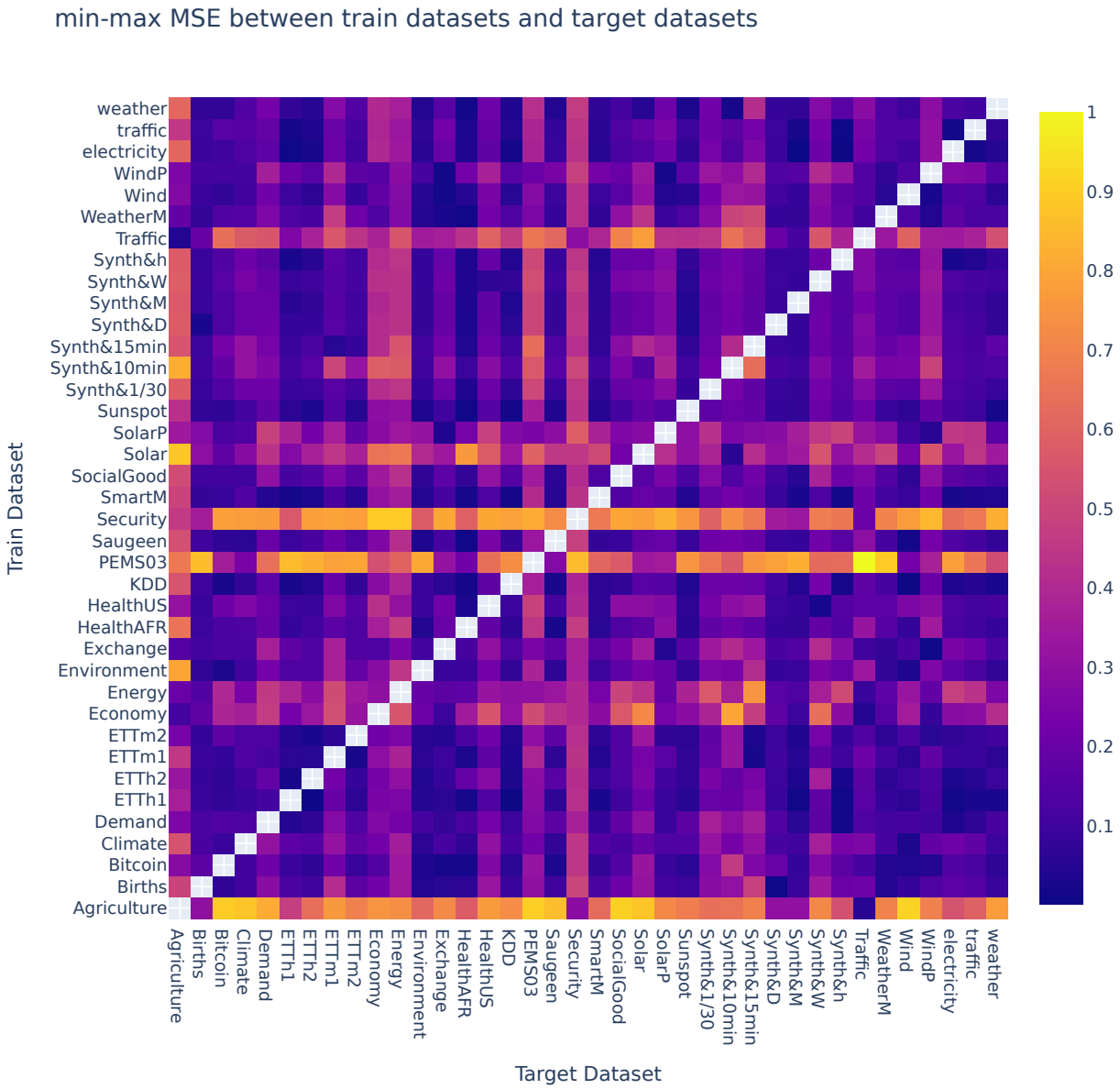

Figure 14: Scaled min-max MSE score between each train dataset and target datasets, the min-max is applied on each column s.t. each train dataset receives a score from 0 to 1. A score of 1 represents the best performer, which is in almost all cases given when the train and target dataset are the same, if the train and target are the same the MSE is removed (grey).

Table 7: Full table of Tab. 1, the comparison between different synthetic data methods with the known target sampling rate (left block) and without it (right block). The model DLinear is also added in this table for reference, it was not included in Tab. 1

		Known Sampling Rate								Unknown Sampling Rate											
		Synth-FAR		Ker-Synth		FM		PFN		Synth-FAR		Ker-Synth		FM		PFN		Naive		Mean	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Moment	ETTh1	0.434	0.423	0.714	0.567	0.691	0.570	1.117	0.728	0.547	0.501	0.708	0.572	1.797	0.913	1.014	0.697	1.294	0.713	0.701	0.558
	ETTh2	0.323	0.361	0.396	0.412	0.359	0.380	1.033	0.627	0.382	0.400	0.380	0.402	0.626	0.527	1.025	0.626	0.432	0.422	0.353	0.387
	ETTm1	0.422	0.404	0.461	0.437	0.578	0.501	3.778	1.280	0.637	0.524	0.977	0.631	2.781	1.044	3.778	1.280	1.214	0.665	0.693	0.548
	ETTm2	0.232	0.290	0.201	0.276	0.212	0.287	0.780	0.605	0.261	0.324	0.257	0.334	0.372	0.419	0.780	0.605	0.266	0.328	0.229	0.307
	Electricity	0.279	0.357	0.482	0.507	0.541	0.541	1.039	0.716	0.384	0.448	0.753	0.675	1.599	0.992	0.895	0.610	1.588	0.945	0.846	0.762
	Traffic	0.869	0.508	1.421	0.767	1.318	0.752	2.443	1.060	1.149	0.653	1.531	0.840	3.073	1.272	2.	0.924	2.714	1.077	1.410	0.805
	Weather	0.292	0.284	0.195	0.240	0.269	0.288	0.798	0.526	0.260	0.288	0.255	0.287	0.491	0.387	0.798	0.526	0.259	0.254	0.215	0.271
	Average	0.407	0.375	0.553	0.458	0.567	0.474	1.570	0.792	0.517	0.448	0.694	0.534	1.534	0.793	1.470	0.753	1.110	0.629	0.635	0.520
GPT4TS	ETTh1	0.406	0.412	0.451	0.431	0.598	0.520	0.813	0.595	0.580	0.511	0.582	0.511	0.945	0.645	0.852	0.613	1.294	0.713	0.701	0.558
	ETTh2	0.307	0.345	0.325	0.363	0.327	0.358	0.876	0.552	0.344	0.383	0.347	0.381	0.491	0.441	0.900	0.550	0.432	0.422	0.353	0.387
	ETTm1	0.354	0.366	0.464	0.403	0.589	0.494	3.706	1.147	0.708	0.541	0.727	0.541	5.111	0.999	3.706	1.147	1.214	0.665	0.693	0.548
	ETTm2	0.200	0.268	0.208	0.275	0.192	0.274	0.602	0.509	0.248	0.317	0.226	0.304	0.426	0.397	0.602	0.509	0.266	0.328	0.229	0.307
	Electricity	0.278	0.355	0.277	0.347	0.418	0.462	0.706	0.476	0.356	0.404	0.596	0.587	1.067	0.839	0.678	0.495	1.588	0.945	0.846	0.762
	Traffic	0.863	0.511	0.926	0.530	1.143	0.641	1.869	0.767	1.315	0.704	1.311	0.761	1.952	0.952	1.860	0.816	2.714	1.077	1.410	0.805
	Weather	0.262	0.258	0.225	0.250	0.244	0.266	0.796	0.469	0.242	0.270	0.248	0.279	0.752	0.426	0.796	0.469	0.259	0.254	0.215	0.271
	Average	0.381	0.359	0.411	0.371	0.502	0.431	1.338	0.645	0.542	0.447	0.577	0.481	1.535	0.671	1.342	0.657	1.110	0.629	0.635	0.520
PatchTST	ETTh1	0.433	0.430	0.450	0.434	0.569	0.505	0.718	0.546	0.480	0.460	0.507	0.474	0.847	0.605	0.640	0.526	1.294	0.713	0.701	0.558
	ETTh2	0.312	0.346	0.332	0.367	0.310	0.351	0.799	0.521	0.320	0.363	0.330	0.370	0.365	0.398	0.570	0.455	0.432	0.422	0.353	0.387
	ETTm1	0.373	0.384	0.411	0.379	0.666	0.515	1.793	0.862	0.521	0.465	0.571	0.493	1.168	0.625	1.793	0.862	1.214	0.665	0.693	0.548
	ETTm2	0.193	0.268	0.209	0.269	0.194	0.277	0.432	0.428	0.220	0.297	0.212	0.291	0.264	0.324	0.432	0.428	0.266	0.328	0.229	0.307
	Electricity	0.287	0.360	0.288	0.364	0.410	0.468	0.631	0.461	0.311	0.371	0.476	0.506	1.222	0.846	0.488	0.437	1.588	0.945	0.846	0.762
	Traffic	0.901	0.537	0.923	0.539	1.115	0.628	1.750	0.723	1.116	0.629	1.181	0.693	1.746	0.915	1.612	0.745	2.714	1.077	1.410	0.805
	Weather	0.246	0.255	0.221	0.247	0.216	0.247	0.508	0.389	0.228	0.265	0.219	0.264	0.315	0.302	0.508	0.389	0.259	0.254	0.215	0.271
	Average	0.392	0.369	0.405	0.371	0.497	0.427	0.947	0.561	0.457	0.407	0.499	0.442	0.847	0.574	0.863	0.549	1.110	0.629	0.635	0.520
TimeMixer	ETTh1	0.473	0.450	0.505	0.457	0.661	0.541	0.704	0.537	0.527	0.476	0.534	0.477	0.825	0.598	0.638	0.513	1.294	0.713	0.701	0.558
	ETTh2	0.326	0.356	0.341	0.373	0.348	0.368	0.819	0.530	0.324	0.365	0.345	0.376	0.372	0.394	0.544	0.446	0.432	0.422	0.353	0.387
	ETTm1	0.422	0.406	0.428	0.393	1.146	0.638	1.832	0.841	0.595	0.473	0.565	0.475	3.091	0.804	1.842	0.843	1.214	0.665	0.693	0.548
	ETTm2	0.211	0.283	0.211	0.273	0.232	0.310	0.344	0.384	0.242	0.311	0.225	0.297	0.363	0.356	0.345	0.384	0.266	0.328	0.229	0.307
	Electricity	0.294	0.356	0.309	0.372	0.443	0.474	0.644	0.454	0.313	0.359	0.373	0.417	0.930	0.755	0.519	0.436	1.588	0.945	0.846	0.762
	Traffic	0.986	0.572	1.016	0.569	1.314	0.686	1.575	0.690	1.223	0.643	1.247	0.675	1.703	0.886	1.528	0.707	2.714	1.077	1.410	0.805
	Weather	0.295	0.267	0.245	0.258	0.266	0.268	0.421	0.353	0.244	0.266	0.277	0.290	0.338	0.311	0.421	0.353	0.259	0.254	0.215	0.271
	Average	0.430	0.384	0.436	0.385	0.630	0.469	0.906	0.541	0.495	0.413	0.509	0.430	1.089	0.586	0.834	0.526	1.110	0.629	0.635	0.520
UniTime	ETTh1	0.439	0.434	0.455	0.435	0.569	0.508	0.688	0.549	0.472	0.452	0.515	0.478	0.856	0.606	0.560	0.499	1.294	0.713	0.701	0.558
	ETTh2	0.330	0.357	0.331	0.366	0.315	0.353	0.808	0.530	0.314	0.360	0.340	0.374	0.375	0.398	0.654	0.474	0.432	0.422	0.353	0.387
	ETTm1	0.389	0.389	0.502	0.403	0.693	0.525	2.073	0.905	0.540	0.469	0.728	0.537	1.306	0.647	2.073	0.905	1.214	0.665	0.693	0.548
	ETTm2	0.204	0.277	0.225	0.278	0.197	0.280	0.435	0.430	0.217	0.294	0.225	0.303	0.250	0.320	0.435	0.430	0.266	0.328	0.229	0.307
	Electricity	0.291	0.355	0.281	0.356	0.397	0.453	0.597	0.448	0.298	0.363	0.397	0.448	1.188	0.831	0.468	0.421	1.588	0.945	0.846	0.762
	Traffic	0.909	0.534	0.926	0.536	1.087	0.624	1.381	0.678	1.065	0.607	1.198	0.687	1.811	0.940	1.437	0.722	2.714	1.077	1.410	0.805
	Weather	0.263	0.264	0.226	0.251	0.199	0.236	0.532	0.401	0.223	0.262	0.236	0.272	0.264	0.286	0.532	0.401	0.259	0.254	0.215	0.271
	Average	0.404	0.373	0.421	0.375	0.494	0.426	0.931	0.563	0.447	0.401	0.520	0.443	0.864	0.575	0.880	0.550	1.110	0.629	0.635	0.520
TTM	ETTh1	0.459	0.443	0.454	0.437	0.587	0.517	0.634	0.526	0.522	0.483	0.483	0.459	1.184	0.667	0.588	0.512	1.294	0.713	0.701	0.558
	ETTh2	0.322	0.352	0.325	0.365	0.325	0.358	0.740	0.512	0.337	0.367	0.328	0.369	0.428	0.415	0.500	0.438	0.432	0.422	0.353	0.387
	ETTm1	0.398	0.393	0.436	0.393	0.810	0.555	1.834	0.862	0.408	0.404	0.567	0.489	4.137	0.872	1.730	0.840	1.214	0.665	0.693	0.548
	ETTm2	0.203	0.276	0.203	0.271	0.207	0.291	0.383	0.405	0.207	0.282	0.209	0.288	0.405	0.372	0.344	0.391	0.266	0.328	0.229	0.307
	Electricity	0.289	0.351	0.285	0.360	0.419	0.463	0.615	0.448	0.310	0.366	0.353	0.418	1.133	0.848	0.449	0.426	1.588	0.945	0.846	0.762
	Traffic	0.961	0.564	0.918	0.539	1.158	0.650	1.535	0.705	1.087	0.616	1.120	0.641	1.798	0.926	1.295	0.693	2.714	1.077	1.410	0.805
	Weather	0.257	0.257	0.212	0.241	0.224	0.250	0.474	0.379	0.252	0.257	0.216	0.261	0.372	0.312	0.368	0.341	0.259	0.254	0.215	0.271
	Average	0.413	0.377	0.405	0.372	0.533	0.441	0.888	0.548	0.446	0.396	0.468	0.418	1.351	0.630	0.753	0.520	1.110	0.629	0.635	0.520
Timer	ETTh1	0.521	0.480	0.515	0.462	0.658	0.543	0.810	0.568	0.576	0.503	0.554	0.485	0.862	0.614	0.680	0.542	1.294	0.713	0.701	0.558
	ETTh2	0.368	0.390	0.362	0.381	0.329	0.362	0.813	0.527	0.344	0.382	0.363	0.386	0.393	0.403	0.645	0.478	0.432	0.422	0.353	0.387
	ETTm1	0.422	0.408	0.489	0.411	0.771	0.547	1.725	0.837	0.620	0.486	0.684	0.499	1.342	0.677	1.725	0.837	1.214	0.665	0.693	0.548
	ETTm2	0.223	0.295	0.235	0.287	0.206	0.288	0.398	0.406	0.250	0.317	0.228	0.300	0.283	0.337	0.398	0.406	0.266	0.328	0.229	0.307
	Electricity	0.313	0.363	0.303	0.361	0.454	0.482	0.694	0.477	0.336	0.370	0.351	0.395	0.940	0.770	0.565	0.457	1.588	0.945	0.846	0.762
	Traffic	1.097	0.619	1.023	0.571	1.284	0.690	1.816	0.733	1.378	0.694	1.247	0.669	1.713	0.906	1.706	0.748	2.714	1.077	1.410	0.805
	Weather	0.291	0.269	0.234	0.250	0.214	0.243	0.484	0.382	0.254	0.271	0.248	0.276	0.299	0.298	0.484	0.382	0.259	0.254	0.215	0.271
	Average	0.462	0.403	0.452	0.389	0.559	0.451	0.963	0.561	0.537	0.432	0.525	0.430	0.833	0.572	0.886	0.550	1.110	0.629	0.635	0.520
DLinear	ETTh1	0.470	0.449	0.531	0.516	0.674	0.575	0.739	0.555	0.661	0.546	0.673	0.598	1.234	0.803	0.717	0.586	1.294	0.713	0.701	0.558
	ETTh2	0.329	0.360	0.871	0.664	0.588	0.536	0.764	0.531	0.370	0.401	1.130	0.774	2.265	1.076	0.719	0.529	0.432	0.422	0.353	0.387
	ETTm1	0.433	0.414	0.445	0.449	0.611	0.547	3.136	1.141	0.616	0.519	0.796	0.632	1.154	0.768	3.136	1.141	1.214	0.665	0.693	0.548
	ETTm2	0.216	0.294	0.355	0.416	0.524	0.512	0.584	0.524	0.254	0.329	1.062	0.758	2.215	1.054	0.584	0.524	0.266	0.328	0.229	0.307
	Electricity	0.301	0.363	0.304	0.390	0.462	0.490	0.628	0.461	0.381	0.431	0.638	0.639	1.357	0.929	0.594	0.481	1.588	0.945	0.846	0.762
	Traffic	0.981	0.572	0.929	0.536	1.290	0.684	1.801	0.750	1.640	0.783	1.260	0.752	1.976	0.971	1.523	0.775	2.714	1.077	1.410	0.805
	Weather	0.394	0.291	0.203	0.270	0.293	0.320	0.588	0.450	0.253	0.275	0.322	0.396	0.481	0.483	0.588	0.450	0.259	0.254	0.215	0.271
	Average	0.446	0.392	0.520	0.463	0.635	0.523	1.177	0.630	0.596	0.469	0.840	0.650	1.526	0.869	1.123	0.641	1.110	0.629	0.635	0.520

Table 8: The MSE performance across 10 Time-MMD datasets comparing pre-trained foundation models with in-domain models trained solely with Synth-FAR. The horizons 48, 60, 72, 84, and 96 were selected for this comparison.

		Synth-FAR Pretrain					Pretrained Foundation Models				Benchmark	
Horizon	Dataset	GPT4TS	PatchTST	DLinear	TimeMixer	TTM	MOIRAI*	Chronos*	TimesFM*	Timer*	Mean	Naive
	Agriculture	0.155	0.147	0.156	0.147	0.146	0.165	0.145	0.145	0.176	0.249	**0.136**
	Climate	0.932	0.874	**0.763**	0.888	0.892	2.282	0.791	1.153	1.003	1.302	0.855
	Economy	0.19	0.17	0.149	0.167	0.158	0.18	0.186	0.172	0.296	0.396	0.159
	Energy	0.227	0.246	0.23	0.243	0.254	0.45	**0.217**	0.29	0.238	0.335	0.223
	Environment	0.786	0.802	**0.782**	0.816	0.826	1.386	0.942	0.87	0.9	0.928	1.214
48	HealthAFR	0.759	0.67	1.08	0.913	0.735	1.287	0.602	**0.409**	0.463	0.467	1.425
	HealthUS	0.464	**0.433**	0.529	0.482	0.472	1.287	0.601	0.505	0.587	0.614	1.031
	Security	1.474	1.463	1.432	1.464	1.459	**1.425**	1.619	1.542	1.519	1.524	1.483
	SocialGood	0.996	0.937	0.896	0.959	0.982	1.147	1.195	**0.483**	1.014	1.242	1.081
	Traffic	0.082	0.073	0.069	0.062	0.066	0.034	0.039	**0.031**	0.107	0.225	0.152
	Average	0.606	0.582	0.609	0.614	0.599	0.964	0.634	**0.56**	0.63	0.728	0.776
	Agriculture	0.158	0.151	0.155	0.15	0.149	0.192	0.16	0.148	0.19	0.265	0.139
	Climate	1.043	0.996	**0.832**	0.999	1.01	2.315	0.904	1.418	1.153	1.386	0.988
	Economy	0.243	0.222	**0.178**	0.218	0.206	0.272	0.24	0.233	0.386	0.465	0.198
	Energy	0.264	0.271	0.254	0.271	0.284	0.916	**0.247**	0.349	0.274	0.363	0.251
	Environment	0.831	0.847	**0.812**	0.872	0.874	1.47	0.98	0.909	0.938	0.962	1.257
60	HealthAFR	1.011	0.651	0.68	0.863	0.681	1.394	0.497	**0.34**	0.399	0.395	1.372
	HealthUS	0.538	**0.441**	0.461	0.484	0.478	1.461	0.582	0.52	0.595	0.623	0.946
	Security	1.305	1.299	**1.26**	1.296	1.294	1.285	1.419	1.385	1.35	1.344	1.348
	SocialGood	1.077	1.027	0.967	1.039	1.078	1.385	1.251	**0.497**	1.134	1.277	1.168
	Traffic	0.101	0.091	0.082	0.08	0.081	0.043	0.05	**0.033**	0.147	0.256	0.163
	Average	0.657	0.6	**0.568**	0.627	0.614	1.073	0.633	0.583	0.657	0.734	0.783
	Agriculture	0.168	0.159	0.16	0.157	0.155	0.164	0.189	0.157	0.208	0.288	0.144
	Climate	1.145	1.105	**0.887**	1.113	1.118	5.115	0.984	1.701	1.268	1.456	1.111
	Economy	0.304	0.284	**0.216**	0.271	0.262	0.284	0.297	0.299	0.474	0.542	0.242
	Energy	0.302	0.303	0.285	0.3	0.317	0.547	**0.281**	0.407	0.308	0.39	0.285
	Environment	0.876	0.893	**0.842**	0.919	0.937	3.077	1.02	0.943	0.972	0.992	1.307
72	HealthAFR	0.961	0.616	0.629	0.837	0.635	1.911	0.452	**0.293**	0.358	0.347	1.345
	HealthUS	0.56	**0.461**	0.473	0.5	0.495	1.459	0.592	0.543	0.604	0.632	0.97
	Security	1.213	1.212	**1.166**	1.209	1.213	1.173	1.29	1.293	1.244	1.238	1.26
	SocialGood	1.147	1.106	0.998	1.135	1.168	3.188	1.381	**0.545**	1.247	1.306	1.273
	Traffic	0.124	0.111	0.1	0.097	0.101	0.054	0.071	**0.036**	0.183	0.29	0.178
	Average	0.68	0.625	**0.576**	0.654	0.64	1.697	0.656	0.622	0.687	0.748	0.812
	Agriculture	0.177	0.168	0.164	0.166	0.163	0.199	0.199	0.168	0.235	0.317	0.15
	Climate	1.245	1.205	**0.931**	1.216	1.219	57.706	1.046	2.0	1.354	1.514	1.229
	Economy	0.369	0.349	**0.256**	0.334	0.324	0.36	0.366	0.37	0.566	0.621	0.292
	Energy	0.315	0.334	0.339	0.325	0.348	0.661	**0.314**	0.452	0.34	0.415	0.323
	Environment	0.92	0.94	**0.866**	0.947	0.989	3.539	1.055	0.971	1.001	1.018	1.354
84	HealthAFR	0.619	0.565	0.916	0.813	0.6	0.67	0.48	**0.26**	0.351	0.313	1.332
	HealthUS	0.486	**0.472**	0.573	0.515	0.501	1.57	0.6	0.557	0.607	0.622	0.968
	Security	1.142	1.149	1.098	1.149	1.147	**1.088**	1.212	1.228	1.174	1.171	1.222
	SocialGood	1.208	1.183	1.018	1.208	1.235	2.026	1.473	**0.584**	1.325	1.322	1.362
	Traffic	0.149	0.138	0.121	0.123	0.126	0.077	0.089	**0.041**	0.228	0.327	0.197
	Average	0.663	0.65	**0.628**	0.68	0.665	6.79	0.683	0.663	0.718	0.764	0.843
	Agriculture	0.198	0.179	0.17	0.18	0.183	0.252	0.212	0.18	0.268	0.353	**0.158**
	Climate	1.338	1.314	**0.978**	1.317	1.313	3.969	1.103	2.335	1.418	1.564	1.348
	Economy	0.442	0.414	**0.293**	0.406	0.4	0.414	0.447	0.454	0.658	0.705	0.349
	Energy	**0.342**	0.361	0.384	0.359	0.378	0.749	0.345	0.489	0.37	0.439	0.354
	Environment	0.958	0.971	**0.891**	0.992	1.043	2.06	1.083	0.994	1.025	1.041	1.399
96	HealthAFR	0.61	0.527	0.857	0.726	0.576	2.795	0.377	**0.236**	0.327	0.288	1.325
	HealthUS	0.485	**0.467**	0.573	0.496	0.51	2.345	0.59	0.553	0.597	0.608	0.965
	Security	1.103	1.101	1.047	1.1	1.101	**0.999**	1.173	1.192	1.136	1.137	1.16
	SocialGood	1.256	1.26	1.049	1.264	1.28	1.844	1.555	**0.61**	1.353	1.324	1.439
	Traffic	0.179	0.164	0.14	0.148	0.159	0.059	0.122	**0.046**	0.265	0.368	0.219
	Average	0.691	0.676	**0.638**	0.699	0.694	1.549	0.701	0.709	0.742	0.783	0.872

Table 9: A full comparison of zero-shot forecasting when training with real data vs. synthetic data. Including Synth-FAR outperforms real data in most cases, with a notable advantage to Synth-FAR with the sampling rate denoted (**S**). The Lowest MSE is each block is marked with bold, and second lowest with an underline.

	Model	TTM				Timer				Moment				UniTime			
	Data Type	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S
ETTh1	96	0.592	**0.438**	0.522	0.459	0.66	**0.521**	0.576	**0.521**	0.64	0.701	0.547	**0.434**	0.599	0.464	0.472	**0.439**
	192	0.644	**0.488**	0.559	0.513	0.744	0.583	0.634	**0.572**	0.675	0.718	**0.554**	0.582	0.733	0.545	0.534	**0.499**
	336	0.631	**0.529**	0.605	0.562	0.75	0.644	0.682	**0.616**	0.692	0.723	**0.581**	0.595	0.827	0.609	0.57	**0.541**
	720	0.675	**0.532**	0.589	0.548	0.736	0.664	0.662	**0.591**	0.712	0.712	**0.531**	0.534	1.641	0.676	0.572	**0.54**
ETTh2	96	0.357	0.324	0.337	**0.322**	0.52	0.531	**0.344**	0.368	**0.32**	0.353	0.382	0.323	0.493	0.435	**0.314**	0.33
	192	0.449	**0.407**	0.427	0.414	0.648	0.642	**0.451**	0.457	**0.413**	0.428	0.437	0.442	0.6	0.528	**0.41**	0.424
	336	0.474	**0.452**	0.482	0.471	0.602	0.6	0.513	**0.503**	**0.45**	0.454	0.471	0.48	0.563	0.555	**0.451**	0.471
	720	0.473	**0.451**	0.475	0.465	0.635	0.644	0.512	**0.48**	**0.451**	**0.451**	0.455	**0.451**	0.594	0.588	**0.454**	0.464
ETTm1	96	1.105	0.404	0.408	**0.398**	1.239	1.165	0.62	**0.422**	0.922	0.693	0.637	**0.422**	1.074	1.022	0.54	**0.389**
	192	1.014	0.455	**0.45**	0.456	1.301	1.199	0.686	**0.475**	0.907	0.71	0.685	**0.444**	1.174	1.159	0.61	**0.438**
	336	0.993	**0.477**	0.484	0.491	1.24	1.153	0.686	**0.495**	0.793	0.722	0.685	**0.514**	1.136	1.118	0.594	**0.472**
	720	1.152	**0.537**	0.546	0.563	1.291	1.162	0.733	**0.551**	0.746	0.746	0.698	**0.569**	1.149	1.158	0.659	**0.539**
ETTm2	96	0.236	**0.2**	0.207	0.203	0.314	0.316	0.25	**0.223**	0.232	**0.229**	0.261	0.232	0.313	0.283	0.217	**0.204**
	192	0.296	**0.268**	0.273	0.272	0.397	0.392	0.314	**0.29**	0.297	**0.284**	0.297	0.287	0.384	0.345	0.282	**0.269**
	336	0.349	**0.332**	0.337	0.338	0.42	0.419	0.376	**0.353**	0.344	**0.338**	0.35	0.356	0.41	0.399	0.333	**0.33**
	720	0.468	**0.428**	0.435	0.439	0.527	0.527	0.476	**0.456**	**0.433**	**0.433**	0.438	0.446	0.52	0.5	0.436	**0.43**
Exchange	96	**0.084**	**0.084**	0.089	0.089	0.096	0.103	0.151	**0.094**	**0.114**	0.139	0.163	0.121	0.099	0.094	0.144	**0.092**
	192	0.196	0.191	**0.179**	**0.179**	**0.185**	0.199	0.23	0.187	0.23	0.235	0.223	**0.19**	0.204	0.189	0.229	**0.18**
	336	0.352	0.353	**0.321**	0.322	**0.308**	0.316	0.382	0.333	0.374	0.383	0.376	**0.337**	0.334	**0.316**	0.374	0.324
	720	0.897	0.877	**0.844**	0.847	**0.81**	0.825	0.929	0.869	0.931	0.931	0.913	**0.882**	**0.829**	0.837	0.918	0.842
Electricity	96	0.484	**0.269**	0.31	0.289	0.438	**0.283**	0.336	0.313	0.687	0.846	0.384	**0.279**	0.412	**0.273**	0.298	0.291
	192	0.525	**0.267**	0.3	0.285	0.492	**0.288**	0.324	0.306	0.73	0.849	**0.347**	0.349	0.46	**0.277**	0.292	0.286
	336	0.444	**0.28**	0.322	0.304	0.476	**0.301**	0.343	0.319	0.764	0.861	0.369	**0.327**	0.577	**0.295**	0.309	0.298
	720	0.543	**0.316**	0.353	0.337	0.66	0.348	0.374	**0.342**	0.892	0.892	0.422	**0.33**	0.844	0.345	0.343	**0.337**
Traffic	96	1.098	**0.88**	1.087	0.961	0.963	**0.95**	1.378	1.097	1.251	1.41	1.149	**0.869**	0.976	0.921	1.065	**0.909**
	192	1.142	**0.822**	0.998	0.889	0.929	**0.88**	1.283	1.014	1.293	1.413	1.007	**0.991**	0.957	0.872	0.997	**0.844**
	336	1.046	**0.828**	1.016	0.898	0.957	**0.895**	1.232	1.009	1.328	1.429	0.994	**0.948**	0.979	0.89	0.959	**0.849**
	720	1.135	**0.85**	1.04	0.927	0.996	**0.926**	1.241	0.999	1.451	1.451	0.992	**0.916**	1.029	0.935	0.99	**0.886**
Weather	96	0.258	**0.242**	0.252	0.257	0.263	0.259	**0.254**	0.291	**0.197**	0.216	0.26	0.292	0.219	**0.209**	0.223	0.263
	192	0.3	**0.28**	0.292	0.304	0.314	0.311	**0.303**	0.357	**0.255**	0.264	0.283	0.324	0.274	**0.263**	0.27	0.302
	336	0.369	**0.328**	0.342	0.353	**0.346**	**0.346**	0.358	0.399	**0.303**	0.312	0.329	0.361	0.323	0.317	**0.316**	0.349
	720	0.439	**0.392**	0.403	0.418	**0.432**	**0.432**	**0.432**	0.451	**0.381**	**0.381**	0.392	0.413	0.399	0.396	**0.392**	0.413

	Model	TimeMixer				GPT4TS				PatchTST				DLinear			
	Data Type	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S	R	R+S(M)	S(M)	S
ETTh1	96	0.681	0.504	0.527	**0.473**	0.548	0.536	0.58	**0.406**	0.728	0.567	0.48	**0.433**	0.592	0.502	0.661	**0.47**
	192	0.709	0.579	0.585	**0.524**	0.649	0.609	0.637	**0.456**	1.154	0.803	0.537	**0.487**	0.63	0.543	0.732	**0.522**
	336	0.771	0.589	0.632	**0.571**	0.628	0.61	0.67	**0.501**	2.845	1.275	0.577	**0.533**	0.628	0.57	0.735	**0.564**
	720	0.89	0.62	0.618	**0.552**	0.635	0.614	0.652	**0.484**	3.103	1.83	0.556	**0.517**	0.635	0.582	0.689	**0.561**
ETTh2	96	0.562	0.501	**0.324**	0.326	0.449	0.409	0.344	**0.307**	0.537	0.521	0.32	**0.312**	0.949	0.479	0.37	**0.329**
	192	0.648	0.608	0.425	**0.422**	0.544	0.518	0.443	**0.396**	0.653	0.631	0.419	**0.405**	1.066	0.583	0.463	**0.422**
	336	0.614	0.572	0.482	**0.48**	0.531	0.5	0.489	**0.445**	0.61	0.576	0.466	**0.459**	0.933	0.602	0.514	**0.498**
	720	0.67	0.6	0.481	**0.469**	0.575	0.524	0.485	**0.44**	0.667	0.615	0.461	**0.452**	1.092	0.774	0.625	**0.578**
ETTm1	96	1.361	1.017	0.595	**0.422**	1.257	0.824	0.708	**0.354**	1.142	1.092	0.521	**0.373**	0.998	0.788	0.616	**0.433**
	192	1.391	1.141	0.663	**0.483**	1.237	0.819	0.772	**0.4**	1.209	1.143	0.565	**0.424**	1.017	0.828	0.709	**0.497**
	336	1.235	1.038	0.666	**0.521**	1.091	0.719	0.719	**0.439**	1.27	1.098	0.573	**0.457**	0.86	0.737	0.688	**0.54**
	720	1.353	1.079	0.712	**0.595**	1.169	0.799	0.76	**0.509**	1.351	1.123	0.628	**0.527**	0.898	0.774	0.715	**0.601**
ETTm2	96	0.336	0.315	0.242	**0.211**	0.304	0.26	0.248	**0.2**	0.357	0.358	0.22	**0.193**	0.908	0.411	0.254	**0.216**
	192	0.395	0.386	0.303	**0.282**	0.372	0.334	0.305	**0.262**	0.434	0.424	0.281	**0.26**	1.007	0.495	0.322	**0.292**
	336	0.417	0.401	0.361	**0.351**	0.394	0.362	0.36	**0.323**	0.44	0.433	0.341	**0.323**	0.855	0.505	0.386	**0.379**
	720	0.521	0.501	0.457	**0.453**	0.503	0.466	0.456	**0.423**	0.563	0.557	0.439	**0.426**	1.087	0.751	0.58	**0.553**
Exchange	96	0.096	0.095	0.152	**0.089**	0.104	0.109	0.159	**0.094**	0.094	0.098	0.142	**0.092**	0.742	0.251	0.127	**0.078**
	192	0.205	0.208	0.226	**0.179**	0.23	0.231	0.233	**0.186**	0.189	0.189	0.221	**0.183**	0.811	0.323	0.194	**0.152**
	336	0.335	0.332	0.382	**0.324**	0.354	0.363	0.379	**0.329**	0.319	**0.312**	0.371	0.323	0.704	0.364	0.29	**0.247**
	720	**0.833**	0.847	0.926	0.853	0.882	0.899	0.922	**0.858**	0.848	**0.838**	0.918	0.847	0.886	0.563	0.495	**0.468**
Electricity	96	0.359	**0.284**	0.313	0.294	0.411	0.306	0.356	**0.278**	0.421	**0.28**	0.311	0.287	0.396	0.321	0.381	**0.301**
	192	0.383	**0.28**	0.305	0.289	0.451	0.306	0.352	**0.271**	0.577	**0.284**	0.305	0.285	0.419	0.341	0.391	**0.294**
	336	0.382	**0.288**	0.331	0.308	0.432	0.307	0.362	**0.288**	0.678	**0.297**	0.322	0.305	0.407	0.332	0.385	**0.309**
	720	0.461	**0.329**	0.367	0.341	0.476	0.346	0.392	**0.323**	0.919	0.349	0.355	**0.337**	0.431	0.364	0.395	**0.332**
Traffic	96	**0.903**	0.929	1.223	0.986	0.994	0.975	1.315	**0.863**	0.93	0.917	1.116	**0.901**	1.005	1.059	1.64	**0.981**
	192	0.882	**0.844**	1.138	0.912	1.01	0.915	1.253	**0.792**	0.912	0.851	1.066	**0.845**	0.988	1.014	1.655	**0.907**
	336	0.866	**0.854**	1.104	0.915	0.954	0.881	1.19	**0.806**	0.939	**0.856**	1.04	0.857	0.957	0.969	1.498	**0.913**
	720	0.938	**0.91**	1.148	0.936	0.985	0.911	1.203	**0.834**	1.034	0.923	1.067	**0.878**	0.977	0.975	1.325	**0.917**
Weather	96	0.217	**0.213**	0.244	0.295	0.221	**0.22**	0.242	0.262	**0.227**	0.235	0.228	0.246	0.282	**0.222**	0.253	0.394
	192	**0.263**	0.268	0.288	0.344	**0.289**	0.291	0.29	0.308	0.285	0.3	**0.272**	0.292	0.332	**0.27**	0.299	0.433
	336	0.313	**0.312**	0.341	0.391	0.328	**0.325**	0.339	0.352	0.33	0.343	**0.324**	0.34	0.349	**0.308**	0.336	0.464
	720	0.388	**0.385**	0.414	0.445	0.406	**0.4**	0.41	0.412	0.411	0.42	**0.396**	0.405	0.408	**0.374**	0.388	0.483