The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)

Federated Foundation Models on Heterogeneous Time Series

Shengchao Chen[1], **Guodong Long**[1], **Jing Jiang**[1], **Chengqi Zhang**[2]

[1]Australian Artificial Intelligence Institute, University of Technology Sydney
[2]Department of Data Science and Artificial Intelligence, The Hong Kong Polytechnic University
shengchao.chen.uts@gmail.com, {guodong.long, jing.jiang}@uts.edu.au
chengqi.zhang@polyu.edu.hk

Abstract

Training a general-purpose time series foundation models with robust generalization capabilities across diverse applications from scratch is still an open challenge. Efforts are primarily focused on fusing cross-domain time series datasets to extract shared subsequences as tokens for training models on Transformer architecture. However, due to significant statistical heterogeneity across domains, this cross-domain fusing approach doesn't work effectively as the same as fusing texts and images. To tackle this challenge, this paper proposes a novel federated learning approach to address the heterogeneity in time series foundation models training, namely FFTS. Specifically, each data-holding organization is treated as an independent client in a collaborative learning framework with federated settings, and then many client-specific local models will be trained to preserve the unique characteristics per dataset. Moreover, a new regularization mechanism will be applied to both client-side and server-side, thus to align the shared knowledge across heterogeneous datasets from different domains. Extensive experiments on benchmark datasets demonstrate the effectiveness of the proposed federated learning approach. The newly learned time series foundation models achieve superior generalization capabilities on cross-domain time series analysis tasks.

Code — https://github.com/shengchaochen82/FFTS
Extended version — https://arxiv.org/abs/2412.08906

Introduction

Training time series foundation models (TSFMs) requires access to numerous publicly available datasets and a large number of datasets provided by various organizations. Federated learning for Foundation Models (Zhuang, Chen, and Lyu 2023) is a new pathway to achieve this utilizing both public and private datasets. Existing efforts focus on training TSFMs in a centralized manner, which is unsuitable for accessing datasets in the private domain (Liu et al. 2024c; Goswami et al. 2024; Das et al. 2023; Woo et al. 2024). For example, cross-silo federated learning (McMahan et al. 2017) explores cross-institutional collaborations among hospitals or financial institutions. Moreover, many end-users with wearable devices and self-driving cars are highly concerned about the use of data collected by these devices.

In comparison to text and images, a unique challenge for time series foundation models is the increasing heterogeneity of tokens. The token, a subsequence of a time series, can represent distinct physical meanings in different application scenarios while the words or image objects usually represent similar meanings in different domains. Lack of cross-domain invariant and increasing heterogeneity of time series significantly impact the performance of foundation models.

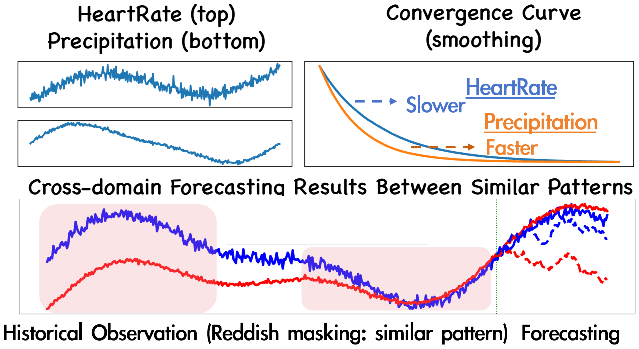

Figure 1: Examples of statistical heterogeneity across time series data. HeartRate: healthcare, Precipitation: weather.

Heterogeneity arises as cross-domain time series often exhibit significant variations in temporal patterns, including trends and timescales. This leads to two main issues: **(1)** inconsistent convergence rates across different domains (refer to **Fig. 1**, top): models trained on complex healthcare data converge more slowly compared to those trained on simpler weather data, despite similar underlying trends; and **(2)** inability of models to leverage heterogeneous data with analogous patterns effectively (refer to **Fig. 1**, bottom): models trained on weather data underperform when applied to healthcare data and vice versa, even though both datasets exhibit similar patterns. Additionally, distinct domain-specific contextual meanings within different timescales often imply unique interpretations, such as fluctuations in weather or health status changes, imply unique interpretations. Despite uniform observation periods, these timescale differences can skew meanings and complicate data integration for training, often causing models to misinterpret underlying patterns when trained on fused cross-domain time series.

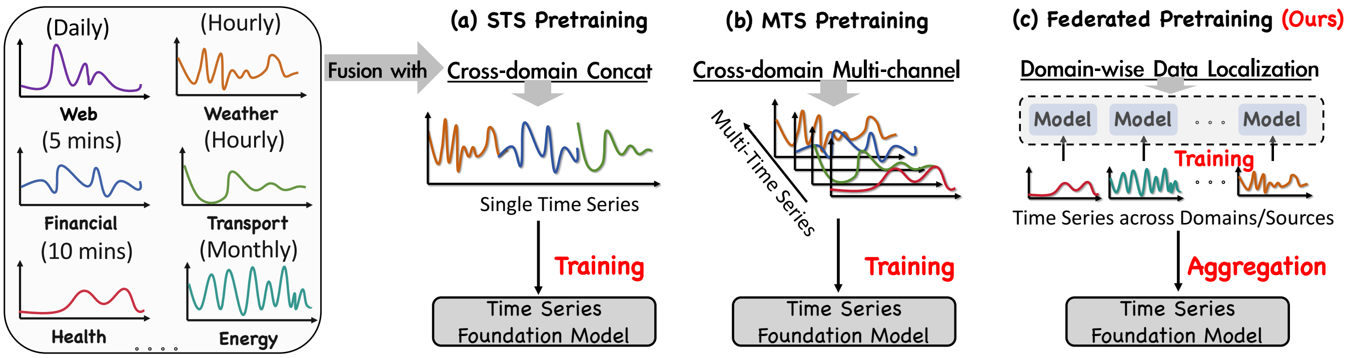

Figure 2: *Overview* of Time Series Foundation Models (TSFMs) that training from scratch. (a) **STS Pretraing:** Training from scratch using a single time series (STS) obtained by fusing data from different domains (Goswami et al. 2024; Liu et al. 2024c; Das et al. 2023). (b) **MTS pretraing:** Training from scratch using multiple time series (MTS) from different domains (Woo et al. 2024). (c) **Federated Pretraining (Ours, this paper):** Instead of merging time series from various domains, separate local models are trained for each source. These models are then aggregated into a global model on the server to form a TSFM.

This paper presents FFTS, a novel federated learning approach to address the heterogeneity in time series foundation models training. As shown in **Fig. 2(c)**, FFTS treats each data-holding organization as an independent client within a collaborative framework. Each client trains a local model to preserve unique dataset characteristics, while a server aggregates these to form a TSFM. FFTS provides the insights for training TSFM on heterogeneous time series in two ways: model architecture and optimization. We adopt a standard encoder-only Transformer architecture and introduce an adaptive trend awareness module to identify similar patterns among heterogeneous sequences. For optimization, we use a uniform masking strategy to prevent local models from memorizing domain-specific knowledge and introduce a heterogeneous knowledge alignment strategy to reduce bias in the global model. A unified adaptation architecture supports diverse downstream tasks. FFTS shows robust generalization and outperforms existing task-specific models. Key contributions include:

- This is the first exploration of the potential for using FL to train time series foundation models for generalization across various downstream application and tasks.
- We present an empirical qualitative analysis demonstrating how statistical heterogeneity across cross-domain time series can negatively affect centralised training.
- We introduce FFTS, a FL approach to tackle statistical heterogeneity in training time series foundation models. FFTS provides insights from model architecture and optimization to enhance training.
- Extensive experiments on real-world time series datasets demonstrate our FFTS exhibits robust zero/few-shot generalization in long-term forecasting and outperform state-of-the-art task-specific models in forecasting, imputation, and anomaly detection.

Related Work

Time Series Foundation Models Pre-trained models have evolved into large foundation models, effectively handling diverse data across domains and tasks with notable advancements in few/zero-shot generalization (Chen et al. 2023a). However, Time Series Foundation Models (TSFMs) are still underdeveloped due to data heterogeneity and scarcity. Research on TSFMs is emerging and broadly falls into two categories: (1) Adapting existing LLMs to TSFMs, which involves enhancing their forecasting accuracy through fine-tuning (Zhou et al. 2023; Chang, Peng, and Chen 2023) or using multimodal prompts (Jin et al. 2023; Liu et al. 2024b; Cao et al. 2023). These approaches depend significantly on the quality of the LLM backbone and effective cross-modal alignment. (2) Training TSFMs from scratch using extensive time series datasets, either real-world (Goswami et al. 2024; Liu et al. 2024c; Garza and Mergenthaler-Canseco 2023) or synthetic (Dooley et al. 2024). This method, although robust for generalization across various domains, is resource-intensive and faces challenges due to the heterogeneous nature of time series data and privacy concerns related to centralized data training. Our FFTS can train a TSFM from scratch ***without*** (1) fusing large-scale datasets on the server, and (2) centralized training raises privacy concerns.

Federated Learning in Time Series Federated Learning (FL) is a distributed learning paradigm that enables collaborative model training while preserving data privacy (McMahan et al. 2017; Chen et al. 2024b). It has been effectively applied in real-world time series analyses across various domains including climate change (Chen et al. 2023b,c), energy (Sun et al. 2024) and finance (Nevrataki et al. 2023). Motivated by the success of foundation models and heightened privacy concerns (Zhuang, Chen, and Lyu 2023), FL is increasingly used to fine-tune pre-trained LLMs, allowing for personalized training that addresses data heterogeneity among clients in distributed time series analysis (Chen et al. 2024a; Liu et al. 2024a). However, the challenge of training a time series foundation model from scratch (instead of fine-tuning pre-trained Large Language Models or other sequence models) in a heterogeneous data environment using federated learning remains a open challenge.

Preliminary

Problem Definition This paper explores training a TSFM from scratch on heterogeneous time series using FL. **Fig. 2** illustrates the primary strategies for training TSFMs. These strategies are categorized into three approaches based on data fusion methods: (1) Single Time Series (STS) pre-training involves concatenating multiple datasets along the time dimension to train a model for a single time series, as shwon in **Fig. 2(a)**. (2) Multiple Time Series (MTS) pre-training combines multiple datasets along the channel dimension to train a model for a multi-channel time series, as shown in **Fig. 2(b)**. (3) Our proposed Federated Pre-training approach treats each time series as an independent client, training a separate model for each client and then aggregating them globally to generate a TSFM, as shown in **Fig. 2(c)**. We define an observation of a multi-channel time series from a given domain $d \in \mathcal{D}$ is as $\boldsymbol{X}_d \in \mathbb{R}^{L \times C}$, where C and L represent the number of channels and the observation length, respectively. Each domain (data-holder), viewed as an independent client, updates its own model, $F(\cdot)$ parameterized by θ, to minimize the distance between output and ground truth. The global optimization objective is to create a uniform model by integrating each client's knowledge as:

$$F(\theta) \colon = \arg\min \sum_{i=1}^{N} \frac{n_k}{n} F_i(\theta_i; \{D_i\}), \qquad (1)$$

where n_i and n is the number of samples held by the k-th client and all clients, respectively, and $F(\theta; \{D\})$ denotes the local objective function, such as mean square error (MSE). This paper proposes FFTS, offering insights into training federated foundation models on heterogeneous time series from both model architecture and optimization perspectives. We will elaborate on these aspects.

FFTS: Model Architecture

As shown in **Fig. 2(c)**, our FFTS treats each data-holder as an independent client, training their own local model and uploading the local model's parameters to the server to generate a global model. The architecture of the local model on each client is depicted in **Fig. 3**, which is based on a standard encoder-only Transformer (**Fig. 3a**) to ensure FFTS's flexibility and versatility. Specifically, we adopt patch embedding to enhance the representation of time series. Additionally, we introduce adaptive trend-awareness module to discover complex temporal patterns in time series and mitigate the influence of heterogeneity across cross-domain data for the global model. Further details are described below.

Patch Embedding To address the lack of semantic information in individual time points of time series data, we utilize a patching strategy that groups neighboring time points into discrete tokens (Nie et al. 2022), achieving two main objectives: (1) capturing local semantic patterns, and (2) enhancing computational efficiency for long sequences. Initially, each time series $\mathbf{X}$ is normalized to zero mean and unit variance via Reverse Instance Normalization (Kim et al. 2021) to counter distribution shifts. Then, $\mathbf{X}$ is divided into overlapping patches of length L_p. The total count of patches, P, is calculated as $P = [(T - L_p)/S] + 2$, with S as the sliding stride. These patches, $\mathbf{X}_P \in \mathbb{R}^{L_p \times P}$, are subsequently transformed into a higher-dimensional space, $\hat{\mathbf{X}}_P \in \mathbb{R}^{P \times d_m}$, using a linear transformation where d_m represents the embedding dimension.

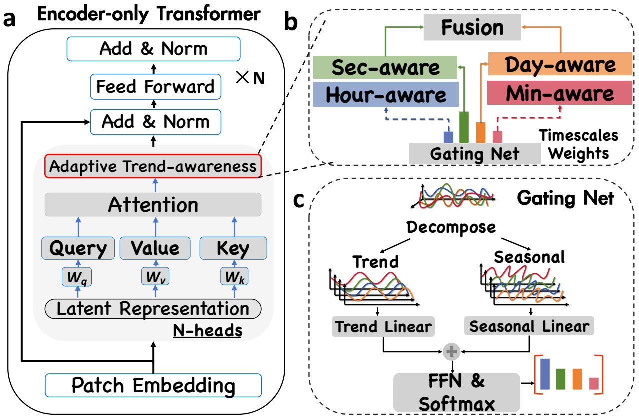

Figure 3: *Architecture of model.* **a** Structure of local model. **b** Architecture of Adaptive Trend-awareness Module, which consists of four independent experts for extracting trends at different timescales based on the representation. Structurally inspired by the Mixture of Experts (MoE) (Fedus, Zoph, and Shazeer 2022). **c** Architecture of the Gating Network.

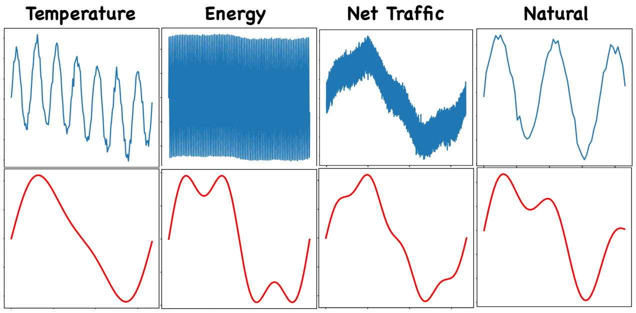

Figure 4: Cross-domain trend similarity within historical observations. **Upper:** Weather (1-hour resolution), Energy (5-minute resolution), Network (30-second resolution), Natural (1-day resolution), **Bottom:** Corresponding trend.

Adaptive Trend-awareness Module Heterogeneous time series can exhibit similar temporal patterns across different timescales, as shown in **Fig. 4**, yet these cross-domain similarities do not typically benefit model training (as discussed in **Fig. 1**). This is means that such time series often contain complex, intertwined temporal patterns. To address this, we propose decoupling individual time series into multiple timescale representations and then applying targeted processing to enhance the model's comprehension of these patterns. This approach aims to improve local model updates and facilitate the sharing of common knowledge during global optimization. Specifically, we introduce Adaptive Trend-awareness Module (ATM) (**Fig. 3b**), which approaches pattern extraction as a multi-task problem, employing independent operations to mine representations across various timescales. Structurally, ATM comprise a gating network (**Fig. 3c**), four independent timescale-aware layers—each serving as an expert in extracting representations according to common timescale criteria (e.g., second, minute, hour, day)—and a fusion layer for integrating these representations. The gating network initially decomposes the training sample's representation $\bar{\boldsymbol{X}}$ into trend component $\bar{\boldsymbol{X}}_{\text{trend}}$ and seasonal component $\bar{\boldsymbol{X}}_{\text{seasonal}}$ to form intermediate representations $\boldsymbol{X}_{mid}$. Subsequently, it calculates the timescale weight W for expert computation. This process can be described mathematically as follows:

$$\begin{aligned}\boldsymbol{X}_{mid} = \boldsymbol{W}_t \cdot \bar{\boldsymbol{X}}_{\text{trend}} + \boldsymbol{W}_s \cdot \bar{\boldsymbol{X}}_{\text{seasonal}},\\ W = \texttt{Softmax}(\text{FFN}(\boldsymbol{X}_{mid})).\end{aligned} \tag{2}$$

The top-k weighted experts process the training sample to minimize computational resource consumption. Their predictions are then combined, using these weights, to generate the final output, which is expressed mathematically as:

$$\begin{aligned}\hat{\boldsymbol{X}} = \text{FFN}\left[\sum_i W_i \cdot f_i(\boldsymbol{X}_{rep})\right]\\ \text{where } i \in \{second, hour, minute, day\}\end{aligned} \tag{3}$$

where $\mathbf{X}_{\text{rep}}$ and $\hat{\mathbf{X}}$ signify the latent and predicted representations, respectively. $f(\cdot)$ represents the linear transformation for each timescale, and FFN is the fusion layer. The underlying motivations are: (1) improving the local model updating via treating time series sample modeling into learning multiple potential representation across time scales and (2) facilitating global common knowledge sharing by mining multiple temporal patterns in heterogeneous time series.

FFTS: Optimization

This section introduces the optimization process of FFTS. Specifically, we implement a unified masking strategy in the local training process to learn sequential patterns of time series and a heterogeneous knowledge alignment strategy in the update process of local and global models to mitigate the impact of heterogeneity on the global model. Furthermore, we apply the global model obtained by FFTS to various time series analysis tasks through a unified downstream adaptation structure. We will elaborate on the details of each component in the following sections.

Unified Masking Strategy Time series from various domains display significant heterogeneity: simpler patterns can accelerate convergence and simplify representation, whereas complex patterns may slow these processes. This diversity leads clients to learn distinctly different patterns, causing the global model to deviate unpredictably and impeding the development of robust, generalized criteria. To mitigate this, we adopt a uniform masked strategy for each client aimed at learning time series trend representations rather than memorizing domain-specific patterns. Specifically, we mask selected input points in each client's time series and require the model to predict these during training. Binary noise masks $\boldsymbol{M} \in \{0, 1\}$ are generated for each sample and applied by element-wise multiplication with the data. We define key parameters, $[L_{\text{m}}, r_m]$, to control the total length of masking and the masking probability. Following (Zerveas et al. 2021), the state transition probabilities are used so that the length of each masked segment adheres to a geometric distribution with a mean of $L_{\text{seg}} = [(1 - r_m)/r_m \cdot L_{\text{m}}]$. This strategy focuses the model on analyzing the relationships between consecutive segments, enhancing its capability to identify underlying trends through contextual analysis rather than merely memorizing isolated points.

Heterogeneous Knowledge Alignment The primary objective of FFTS is to extract common knowledge from heterogeneous time series, thereby establishing a global TSFM with uniform standards. To mitigate the bias introduced by heterogeneity across domains and improve the model performance, we adopt a heterogeneous knowledge alignment strategy into optimization objectives. This involves incorporating ATM-specific regularization terms during both local updates and global optimization to minimize discrepancies between local and global knowledge at various timescales. The local updating process is an unsupervised masked time points reconstruction task. The local optimization objective include MSE for reconstruction accuracy and an additional regularization term specifically for ATM that used for heterogeneous knowledge alignment, can be formulated as:

$$\mathcal{L} = \frac{1}{|\boldsymbol{M}|}\sum_{t \in \boldsymbol{M}}\left[\hat{\boldsymbol{X}}(t) - \boldsymbol{X}(t)\right]^2 + \lambda||\Theta_T - \hat{\Theta}_T||^2, \tag{4}$$

where $|\boldsymbol{M}|$ is the total number of masked time points, $\hat{\boldsymbol{X}}(t)$ and $\boldsymbol{X}(t)$ represent the prediction of masking points and ground truth, respectively, Θ_T and $\hat{\Theta}_T$ denote the local ATM's parameters and global ATM's parameters. The global optimization objective can be formulated as:

$$F(\theta){:} = \arg\min \sum_{i=1}^{N} \frac{n_k}{n}\mathcal{L}_i(\theta_i; D_i), \tag{5}$$

where n_i and n is the number of samples held by the k-th client and all clients, respectively. The motivation is to enhance insights into local and global shared knowledge, integrating local features while maintaining global consistency to improve the adaptability of the global model.

Unified Architecture FFTS adapts to downstream tasks using a unified architecture (see **Fig. 5**). This architecture consists of a unified adaptation head, which includes a multilayer perceptron and layer normalization, enabling efficient processing of downstream tasks with minimal effort.

Experiments

This section presents our main experimental results, highlighting our FFTS's performance during federated pre-training and its competitive results compared to DL-based time series analysis in different downstream tasks. These tasks include forecasting (long/short-term), interpolation, and anomaly detection. All downstream datasets were excluded from the pre-training phase to prevent data leakage.

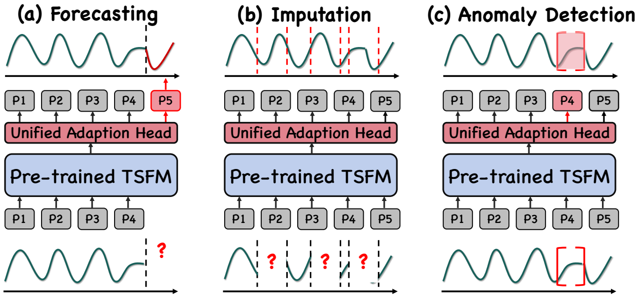

Figure 5: FFTS for downstream adaption. A unified adaptation head facilitates knowledge transfer across tasks. (a) Predicting future from past. (b) Filling gaps in data using related context. (c) Identifying unusual patterns in time series.

Federated Pretraining

Setups Federated pre-training is the most critical step in acquiring TSFM in our work. To evaluate the effectiveness of FFTS, we evaluated its performance on a range of different training settings. Specifically, we empirically selected 18 time series datasets (see **Appendix** for details) from different domains/sources to complete this process and set each dataset to correspond to an independent client. We adopted a uniform pre-training setting of $L = 512$ for each client, and $L_m \in \{8, 16, 24\}$ and $r_m \in \{15\%, 25\%, 50\%\}$ and $k = 3$ for evaluate its performance across different setting. Moreover, to further evaluate the performance of FFTS in FL perspective, we compare it with FL baseline such as FedAvg (McMahan et al. 2017), FedProx (Li et al. 2020), and pFedMe (T Dinh, Tran, and Nguyen 2020). We also introduced a centralized training strategy, FFTS-Cen, using the same model to demonstrate the effectiveness of FFTS in pretraining for TSFM. During the evaluation phase, we used a uniform configuration with $[L_m = 16, r_m = 35\%]$. More about pretraining can be found at **Appendix B**.

Main Results The main results of pre-training are shown in **Table 1**. For sub-tables (fixed $r_m = 25\%$): **(a)** FFTS outperforms baselines and indicate that heterogeneous FL algorithms are not effective as on natural image tasks when dealing with heterogeneous time series. **(b)** the performance of FFTS is affected by k in the ATM and optimal at $k = 3$, which implies that these are indeed multiple cross-timescale trend similarities between heterogeneous time series and that FFTS is able to handle effectively. **(c)** the effect of client participation rate (PRTP), empirical results show that the PRTP is proportional to the performance. **(d)** the effect of the regularization weights λ associated with ATM on performance. **(e)** for different timescales layer, the ablation of the trend awareness layer proves the effectiveness of the ATM.

Framework Ablation Results We conducted additional ablation experiments to analyze the impact of specific components. These included FFTS without ATM (FFTS-A, equivalent to vanilla FedAvg since ATM is a prior of Heterogeneous Knowledge Alignment) and FFTS without Heterogeneous Knowledge Alignment (FFTS-B). The results shown in **Table 2** indicate the effectiveness of the proposed ATM and heterogeneous knowledge alignment strategy.

L_m	8			16			24			Avg.
Ratio	15%	25%	50%	15%	25%	50%	15%	25%	50%	-
FedAvg	$\underline{.457}$	$\underline{.449}$	$\underline{.443}$	$\underline{.431}$	$\underline{.448}$	$\underline{.447}$	$\underline{.455}$	$\underline{.435}$	$\underline{.437}$	.448
FedProx	.633	.644	.627	.673	.686	.680	.680	.665	.669	.662
pFedMe	.613	.612	.606	.654	.672	.665	.667	.651	.642	.642
FFTS-Cen	.433	.419	.420	.417	.430	.429	.442	.405	.402	$\underline{.422}$
FFTS (Ours)	**.431**	**.423**	**.421**	**.413**	**.421**	**.425**	**.436**	**.398**	**.399**	**.418**

(a) Results of FFTS and FL baseline in the pretraining process.

k/L_m	8	16	24
1	.427	.432	$\underline{.404}$
2	.424	$\underline{.427}$	.414
3	**.423**	**.421**	**.398**
4	.456	.445	.420

(b) Sensitivity of k.

r_p/L_m	8	16	24
10%	.486	.500	.532
20%	.423	.439	.451
30%	$\underline{.433}$	$\underline{.422}$	$\underline{.429}$
50%	**.423**	**.421**	**.398**

(c) Impact of participation rate.

λ/L_m	8	16	24
$5e^{-2}$	$\underline{.429}$	$\underline{.430}$	$\underline{.412}$
$1e^{-1}$	**.423**	**.421**	**.398**
$2e^{-1}$	.442	.437	.420

(d) Impact of λ.

Reg.$/L_m$	8	16	24
Day	**.426**	$\underline{.427}$	$\underline{.411}$
Minute	$\underline{.427}$	.431	.416
Hour	.429	.430	.414
Second	**.426**	**.425**	**.407**

(e) Ablation on ATM.

Table 1: Main results of federated pre-training process (MSE report). Leading zeros are omitted for values less than one. **Bold**: the best, $\underline{\text{Underline}}$: the second best.

Method$/L_m$	8	16	24	Avg.	Ave. Var.
FFTS-A	.449	.448	.435	.444	↓ 7.25%
FFTS-B	.437	.435	.420	.431	↓ 4.11%
FFTS	.423	.421	.398	.414	-

Table 2: Framework ablation results (MSE report). "Ave. Var.“ denotes the average variation in performance.

Analysis for Effectiveness of ATM To further demonstrate the effectiveness of the proposed ATM, we visualized the timescale weights (**Fig. 6a**) and outputs from the timescale-aware layers (**Fig. 6b**). The varied timescale weights across different clients indicate that the ATM can adapt its strategy to effectively use different layers depending on the time series data. This method efficiently identifies trend similarities in heterogeneous time series, improving the global model's insights. Moreover, regular outputs from the ATM's four timescale-aware layers under sinusoidal functions confirm their capability to detect trends across timescales, validating the ATM's effectiveness.

Downstream Baseline

We evaluated the performance of our FFTS on forecasting, imputation, and anomaly detection, using DL-based time series models following (Zhou et al. 2023; Jin et al. 2023). We compare our FFTS to baselines that includes Log-Transformer (Li et al. 2019), N- BEATS (Oreshkin et al. 2020), Reformer (Kitaev, Kaiser, and Levskaya 2020), Informer (Zhou et al. 2021), LightTS (Zhang et al. 2022), ETSformer (Woo et al. 2022), Stationary (Liu et al. 2022),

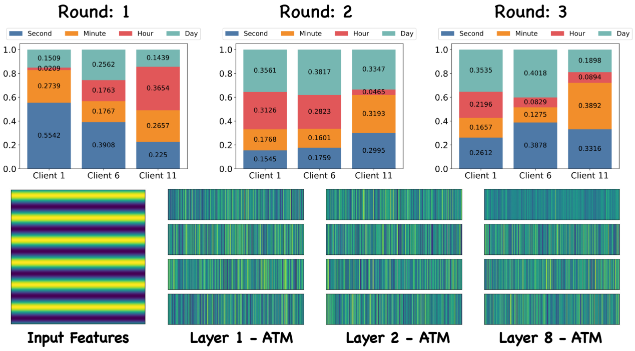

Figure 6: ATM Visualization. Upper: variations in timescale weights for selected clients across various rounds. Bottom: featuring regular sinusoidal patterns for the rightmost input features. Subsequent depict Gating network representation in ATM's first, second, and final layers. From top to bottom, represent second-, minute-, hour-, and day-aware layers.

Autoformer (Wu et al. 2021), FEDformer (Zhou et al. 2022), Pyraformer (Liu et al. 2021), AnomalyTransformer (Xu et al. 2022), TimesNet (Wu et al. 2022), PatchTST (Nie et al. 2022), DLinear (Zeng et al. 2023), N- HiTS (Challu et al. 2022), GPT4TS (Zhou et al. 2023), LLMTime (Gruver et al. 2024), Time- LLM (Jin et al. 2023). Note that the use of our proposed FFTS and FL baselines for downstream tasks is achieved through fine-tuning. More details in **Appendix B**.

Time Series Forecasting

Setups Time series forecasting is essential yet challenging in real-world applications. To evaluate performance, we adopt popular benchmarks and experimental setting following (Jin et al. 2023), including ETT (ETTh1, ETTh2, ETTm1, ETTm2), Weather, and Illness datasets (excluding Traffic and Electricity due to their presence in pretraining), with a unified look-back window length of 512. We conduct three types of downstream experiments: (1) Regular: fine-tuning the pretrained TSFM before evaluation; (2) Few-shot: fine-tuning the pretrained TSFM using 5% or 10% of data; and (3) Zero-shot: evaluating the pretrained TSFM without training on target datasets (only ETT-series dataset).

Main Results **Table 3** demonstrates the TSFM obtained by our FFTS can achieve best long-term forecasting performance, surpassing the SOTA method Time-LLM by averaged **9.52%** across six datasets. Notably, the TSFM obtained by vanilla FedAvd can also achieve very competitive performance compared to Time-LLM, with a difference of only **1.7%**. These indicate that (1) FL can be used as an effective pre-training strategy for unimodal TSFM, and (2) our FFTS is more effective in dealing with heterogeneous temporal pre-training compared to comparable baselines.

Few-/Zero-shot Results Few/zero-shot generalization are crucial capabilities for TSFMs. **Table 4** illustrates the few-shot generalization capability of FFTS, which outperforms baseline models. FFTS achieves optimal performance, surpassing Time-LLM by an average of **4.1%** with limited training data. Notably, the TSFM obtained through FedAvg also exhibits superior performance, showing an average **1.1%** improvement over Time-LLM. Despite its comparatively lower pretraining performance, achieves competitive results against Time-LLM. **Table 5** presents zero-shot results on the ETT dataset, where the TSFM obtained by our FFTS outperforms Time-LLM by an average of **3.87%**. Most significant, FFTS outperforms the centralised training baseline (FFTS-Cen) on the same architecture and pretraining datasets in most case. These results underscore the comparable performance of our model to pre-trained LLMs, even with limited training datasets, and underscore the effectiveness and superiority of the proposed FFTS.

Model	ETTh1	ETTh2	ETTm1	ETTm2	Weather	ILI	Avg
Reformer	1.029	6.736	0.799	1.479	0.803	4.724	2.595
Informer	1.040	4.431	0.961	1.410	0.634	5.137	2.269
LightTS	0.491	0.602	0.435	0.409	0.261	7.382	1.597
ETSformer	0.542	0.439	0.429	0.293	0.271	2.497	0.745
Stationary	0.570	0.526	0.481	0.306	0.288	2.077	0.708
Autoformer	0.496	0.450	0.588	0.327	0.338	3.006	0.868
FEDformer	0.440	0.437	0.448	0.305	0.309	2.847	0.798
TimesNet	0.458	0.414	0.400	0.291	0.259	2.139	0.660
PatchTST	0.413	0.330	0.351	0.255	0.225	1.443	0.503
Dlinear	0.422	0.431	0.357	0.267	0.248	2.169	0.649
GPT4TS	0.465	0.381	0.388	0.284	0.237	1.925	0.613
Time-LLM	0.408	0.334	0.329	0.251	0.225	1.435	0.497
FedAvg	0.400	0.342	0.335	0.251	0.218	1.390	0.489
FedProx	0.425	0.354	0.370	0.309	0.263	1.550	0.545
pFedMe	0.425	0.353	0.372	0.302	0.260	1.553	0.544
FFTS-Cen	0.392	0.331	**0.314**	0.245	0.217	1.353	0.475
FFTS (Ours)	**0.389**	**0.329**	0.315	**0.244**	**0.215**	**1.348**	**0.473**

Table 3: Long-term forecasting results average across prediction horizons $\{96, 192, 336, 720\}$ and $\{24, 36, 48, 60\}$ for ILI dataset. **Bold**: the best, Underline: the second best. **Appendix D** shows the full results.

Method	ETTh1	ETTh2	ETTm1	ETTm2	Weather
Ratio	10% / 5%	10% / 5%	10% / 5%	10% / 5%	10% / 5%
Reformer	1.249 / 1.241	3.485 / 3.527	1.426 / 1.264	3.978 / 3.581	0.546 / 0.447
Informer	1.199 / 1.225	3.872 / 3.922	1.192 / 1.163	3.37 / 3.658	0.597 / 0.584
LightTS	1.375 / 1.451	2.655 / 3.206	0.971 / 1.123	0.987 / 1.415	0.289 / 0.305
ETSformer	1.180 / 1.189	0.894 / 0.809	0.980 / 1.125	0.447 / 0.534	0.318 / 0.333
Stationary	0.915 / 0.943	0.462 / 0.470	0.797 / 0.857	0.332 / 0.341	0.318 / 0.327
Autoformer	0.702 / 0.722	0.488 / 0.441	0.802 / 0.796	1.342 / 0.388	0.300 / 0.310
FEDformer	0.639 / 0.658	0.466 / 0.463	0.722 / 0.730	0.463 / 0.381	0.284 / 0.309
TimesNet	0.869 / 0.925	0.479 / 0.439	0.677 / 0.717	0.320 / 0.344	0.279 / 0.298
PatchTST	0.633 / 0.694	0.415 / 0.827	0.501 / 0.526	0.296 / 0.314	0.242 / 0.269
DLinear	0.691 / 0.750	0.605 / 0.694	0.411 / 0.400	0.316 / 0.399	0.241 / 0.263
GPT4TS	0.590 / 0.681	0.397 / 0.400	0.464 / 0.472	0.293 / 0.308	0.238 / 0.263
Time-LLM	0.556 / 0.627	0.370 / 0.382	0.404 / 0.425	0.277 / 0.274	0.234 / 0.260
FedAvg	0.562 / 0.618	0.369 / 0.371	0.398 / 0.420	0.275 / 0.278	0.227 / 0.256
FedProx	0.577 / 0.630	0.382 / 0.384	0.414 / 0.431	0.285 / 0.299	0.233 / 0.269
pFedMe	0.575 / 0.629	0.374 / 0.380	0.421 / 0.433	0.290 / 0.295	0.230 / 0.262
FFTS-Cen	0.549 / 0.609	0.363 / 0.365	0.387 / 0.417	0.270 / 0.263	0.219 / 0.247
FFTS (Ours)	**0.546 / 0.606**	**0.362 / 0.363**	**0.386 / 0.411**	**0.266 / 0.259**	**0.216 / 0.245**

Table 4: Few-shot results on long-term forecasting average across prediction horizons of $\{96, 192, 336, 720\}$ with ratio $\{10\%, 5\%\}$ of training dataset. **Bold**: the best, Underline: the second best. Full results are in **Appendix D**.

Time Series Imputation

Setups Imputation aiming to fill corrupted time series based on partially observed data. We conduct experiment

Methods	FFTS	Time-LLM	LLMTime	GPT4TS	PatchTST	TimesNet
h1⇒h2	**0.339** / 0.343	0.353	0.992	0.406	0.380	0.421
h1⇒m2	**0.266** /0.269	0.273	1.867	0.325	0.314	0.327
h2⇒h1	**0.460** / 0.463	0.479	1.961	0.757	0.565	0.865
h2⇒m2	**0.263** / **0.263**	0.272	1.867	0.335	0.325	0.342
m1⇒h2	0.369 /**0.367**	0.381	0.992	0.433	0.439	0.457
m1⇒m2	**0.249** / 0.254	0.268	1.867	0.313	0.296	0.322
m2⇒h2	**0.339** / 0.345	0.354	0.992	0.435	0.409	0.435
m2⇒h1	**0.403** / 0.405	0.414	1.933	0.769	0.568	0.769
Avg.	**0.336** / 0.339	0.349	1.559	0.472	0.412	0.492

Table 5: Zero-shot results (MSE report). The source dataset is ETT. **Bold**: the best, Underline: the second best. ⇒ means the transfer of the source dataset to the target dataset (ETT series). Full results are in **Appendix D**.

on five popular real-world datasets, including ETT (ETTh1, ETTh2, ETTm1, ETTm2), and Weather, where the data-missing is common. Following the experiment setting of GPT4TS (Zhou et al. 2023), different random mask ratio $\{96, 192, 336, 720\}$ of time points are selected for the evaluation on various proportions of missing data.

Result **Table 6** demonstrates our proposed FFTS can achieve the best performance across different datasets. Compared to the state-of-the-art GPT4TS, the TSFM trained using FFTS demonstrates superior performance, reducing MSE by **14.7%**. Furthermore, the TSFM obtained through vanilla FedAvg surpasses GPT4TS by **2.6%**. FFTS also achieves competitive performance against the centralised strategy. These results validate both the effectiveness of FL in TSFM pre-training and the superiority of our FFTS.

Method	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Avg.
Reformer	0.055	0.157	0.122	0.234	0.038	0.121
Informer	0.071	0.156	0.161	0.337	0.045	0.154
LightTS	0.051	0.029	0.103	0.055	0.031	0.054
ETSformer	0.036	0.026	0.094	0.053	0.032	0.048
Stationary	0.062	0.101	0.117	0.163	0.099	0.108
Autoformer	0.093	0.096	0.201	0.142	0.052	0.117
FEDformer	0.104	0.046	0.284	0.119	0.055	0.122
TimesNet	0.120	0.208	0.202	0.367	0.076	0.195
PatchTST	0.047	0.029	0.115	0.065	0.060	0.063
DLinear	0.027	0.022	0.078	0.049	0.030	0.041
GPT4TS	0.028	0.021	0.069	0.048	0.031	0.039
FedAvg	0.026	0.022	0.063	0.046	0.031	0.038
FedProx	0.030	0.026	0.073	0.047	0.034	0.042
pFedMe	0.031	0.024	0.071	0.049	0.032	0.041
FFTS-Cen	0.024	**0.018**	**0.057**	0.045	**0.029**	0.035
FFTS (Ours)	**0.023**	**0.018**	0.058	**0.044**	**0.029**	**0.034**

Table 6: Imputation. Time points are randomly masked at ratios of $\{12.5\%, 25\%, 37.5\%, 50\%\}$ with an input length of 96. The results presented are averaged across these four different mask ratios. **Bold**: the best, Underline: the second best. Full results are in **Appendix D**.

Time Series Anomaly Detection

Setups We benchmark FFTS against five widely utilized datasets: SMD, MSL, SMAP, SwaT, and PSM, adhering to the experimental protocols of GPT4TS (Zhou et al. 2023) to guarantee a fair comparison. Details in **Appendix B/D**.

Method	SMD	MSL	SMAP	SWaT	PSM	Avg
Transformer	79.56	78.68	69.70	80.37	76.07	76.88
LogTransformer	76.21	79.57	69.97	80.52	76.74	76.60
Autoformer	85.11	79.05	71.12	92.74	93.29	84.26
Pyraformer	83.04	84.86	71.09	91.78	82.08	82.57
Informer	81.65	84.06	69.92	81.43	77.10	78.83
Reformer	75.32	84.40	70.40	82.80	73.61	77.31
ETSformer	83.13	**85.03**	69.50	84.91	91.76	82.87
FEDformer	85.08	78.57	70.76	93.19	97.23	84.97
Stationary	84.62	77.50	71.09	79.88	97.29	82.08
Anomaly	85.49	83.31	71.18	83.10	79.40	80.50
TimesNet	84.61	81.84	69.39	93.02	97.34	85.24
PatchTST	84.62	78.70	68.82	85.72	96.08	82.78
DLinear	77.10	84.88	69.26	87.52	93.55	82.46
GPT4TS	86.89	82.45	72.88	94.23	97.13	86.72
FedAvg	87.21	83.33	74.20	95.00	98.20	87.59
FedProx	85.45	82.21	70.09	92.81	96.59	85.43
pFedMe	85.78	83.04	71.11	91.99	96.20	85.62
FFTS-Cen	87.93	83.56	74.08	94.94	97.95	87.69
FFTS (Ours)	**88.42**	84.01	**74.53**	**95.27**	**98.25**	**88.10**

Table 7: Anomaly detection. We calculate the F1-score for each dataset and statics the average F1-score. **Bold**: the best, Underline: the second best. **Appendix D** shows full results.

Results **Table 7** reveals that the TSFM developed via our FFTS exhibits superior performance, achieving an average F1-score of **88.10%**, which surpasses the previous SOTA GPT4TS by **1.02%**. Additionally, the TSFM trained with FFTS outperforms its counterpart trained via FL baseline by an average of **1%**. Notably, the TSFM trained via FedAvg exceeds the performance of GPT4TS, underscoring the effectiveness of FL in improving TSFM training and demonstrating the efficacy of FFTS. The competitive performance relative to the centralised training baseline (FFTS-Cen) further indicates the superiority of FFTS.

Conclusion

This paper demonstrates the potential of FL for training foundation models on heterogeneous time series datasets. We introduce FFTS, a novel FL approach designed to address heterogeneity in time series foundation model training. FFTS considers each data-holding organization as an independent client within a collaborative framework. Each client trains a local model to preserve the unique characteristics of its dataset, while a server aggregates these models to form a time series foundation model. FFTS enhances training through model architecture and optimization, introducing an adaptive trend-awareness module, a uniform masking strategy, and a heterogeneous knowledge alignment strategy. A unified adaptation architecture supports various downstream tasks. Extensive experiments on real-world time series datasets demonstrate FFTS's robust generalization capabilities in forecasting, imputation, and anomaly detection.

References

Cao, D.; Jia, F.; Arik, S. O.; Pfister, T.; Zheng, Y.; Ye, W.; and Liu, Y. 2023. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*.

Challu, C.; Olivares, K. G.; Oreshkin, B. N.; Garza, F.; Mergenthaler-Canseco, M.; and Dubrawski, A. 2022. N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting. arXiv:2201.12886.

Chang, C.; Peng, W.-C.; and Chen, T.-F. 2023. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *arXiv preprint arXiv:2308.08469*.

Chen, S.; Long, G.; Jiang, J.; Liu, D.; and Zhang, C. 2023a. Foundation models for weather and climate data understanding: A comprehensive survey. *arXiv preprint arXiv:2312.03014*.

Chen, S.; Long, G.; Jiang, J.; and Zhang, C. 2024a. Personalized Adapter for Large Meteorology Model on Devices: Towards Weather Foundation Models. *arXiv preprint arXiv:2405.20348*.

Chen, S.; Long, G.; Shen, T.; and Jiang, J. 2023b. Prompt federated learning for weather forecasting: Toward foundation models on meteorological data. *arXiv preprint arXiv:2301.09152*.

Chen, S.; Long, G.; Shen, T.; Jiang, J.; and Zhang, C. 2023c. Federated Prompt Learning for Weather Foundation Models on Devices. *arXiv preprint arXiv:2305.14244*.

Chen, S.; Shu, T.; Zhao, H.; Wang, J.; Ren, S.; and Yang, L. 2024b. Free lunch for federated remote sensing target fine-grained classification: A parameter-efficient framework. *Knowledge-Based Systems*, 294: 111694.

Das, A.; Kong, W.; Sen, R.; and Zhou, Y. 2023. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*.

Dooley, S.; Khurana, G. S.; Mohapatra, C.; Naidu, S. V.; and White, C. 2024. Forecastpfn: Synthetically-trained zero-shot forecasting. *Advances in Neural Information Processing Systems*, 36.

Fedus, W.; Zoph, B.; and Shazeer, N. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120): 1–39.

Garza, A.; and Mergenthaler-Canseco, M. 2023. TimeGPT-1. *arXiv preprint arXiv:2310.03589*.

Goswami, M.; Szafer, K.; Choudhry, A.; Cai, Y.; Li, S.; and Dubrawski, A. 2024. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*.

Gruver, N.; Finzi, M.; Qiu, S.; and Wilson, A. G. 2024. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36.

Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J. Y.; Shi, X.; Chen, P.-Y.; Liang, Y.; Li, Y.-F.; Pan, S.; et al. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.

Kim, T.; Kim, J.; Tae, Y.; Park, C.; Choi, J.-H.; and Choo, J. 2021. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*.

Kitaev, N.; Kaiser, L.; and Levskaya, A. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Li, S.; Jin, X.; Xuan, Y.; Zhou, X.; Chen, W.; Wang, Y.-X.; and Yan, X. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2: 429–450.

Liu, Q.; Liu, X.; Liu, C.; Wen, Q.; and Liang, Y. 2024a. Time-FFM: Towards LM-Empowered Federated Foundation Model for Time Series Forecasting. *arXiv preprint arXiv:2405.14252*.

Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A. X.; and Dustdar, S. 2021. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*.

Liu, X.; Hu, J.; Li, Y.; Diao, S.; Liang, Y.; Hooi, B.; and Zimmermann, R. 2024b. Unitime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM on Web Conference 2024*, 4095–4106.

Liu, Y.; Wu, H.; Wang, J.; and Long, M. 2022. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35: 9881–9893.

Liu, Y.; Zhang, H.; Li, C.; Huang, X.; Wang, J.; and Long, M. 2024c. Timer: Transformers for time series analysis at scale. *arXiv preprint arXiv:2402.02368*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Nevrataki, T.; Iliadou, A.; Ntolkeras, G.; Sfakianakis, I.; Lazaridis, L.; Maraslidis, G.; Asimopoulos, N.; and Fragulis, G. F. 2023. A survey on federated learning applications in healthcare, finance, and data privacy/data security. In *AIP Conference Proceedings*, volume 2909. AIP Publishing.

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

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2020. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. arXiv:1905.10437.

Sun, H.; Tang, X.; Yang, C.; Yu, Z.; Wang, X.; Ding, Q.; Li, Z.; and Yu, H. 2024. HiFi-Gas: Hierarchical Federated Learning Incentive Mechanism Enhanced Gas Usage Estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 22824–22832.

T Dinh, C.; Tran, N.; and Nguyen, J. 2020. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33: 21394–21405.

Woo, G.; Liu, C.; Kumar, A.; Xiong, C.; Savarese, S.; and Sahoo, D. 2024. Unified Training of Universal Time Series Forecasting Transformers. arXiv:2402.02592.

Woo, G.; Liu, C.; Sahoo, D.; Kumar, A.; and Hoi, S. 2022. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34: 22419–22430.

Xu, J.; Wu, H.; Wang, J.; and Long, M. 2022. Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. arXiv:2110.02642.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

Zerveas, G.; Jayaraman, S.; Patel, D.; Bhamidipaty, A.; and Eickhoff, C. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2114–2124.

Zhang, T.; Zhang, Y.; Cao, W.; Bian, J.; Yi, X.; Zheng, S.; and Li, J. 2022. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186*.

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

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, 27268–27286. PMLR.

Zhou, T.; Niu, P.; Sun, L.; Jin, R.; et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36: 43322–43355.

Zhuang, W.; Chen, C.; and Lyu, L. 2023. When foundation model meets federated learning: Motivations, challenges, and future directions. *arXiv preprint arXiv:2306.15546*.