FAT: Frequency-Aware Pretraining for Enhanced Time-Series Representation Learning

Rui Cheng
School of Finance
Southwestern University of Finance and Economics
Chengdu, China

Xiangfei Jia
School of Computing and Artificial Intelligence
Southwestern University of Finance and Economics
Chengdu, China

Qing Li*
Research Institute for Digital Economy and Interdisciplinary Sciences
Southwestern University of Finance and Economics
Chengdu, China

Rong Xing
School of Computing and Artificial Intelligence
Southwestern University of Finance and Economics
Chengdu, China

Jiwen Huang
School of Finance
Southwestern University of Finance and Economics
Chengdu, China

Yu Zheng
School of Finance
Southwestern University of Finance and Economics
Chengdu, China

Zhilong Xie
School of Management Science and Engineering
Southwestern University of Finance and Economics
Chengdu, China

Abstract

Recent advancements in time-series forecasting have highlighted the importance of frequency-domain modeling. However, deep learning models primarily operate in the time domain, limiting their ability to capture frequency-based patterns. Existing approaches normally introduce novel neural network architectures tailored to task-specific frequency properties, yet they often lack generalization and require extensive domain-specific adaptations. In this paper, we propose **FAT**, a novel pretraining framework that learns generalizable **F**requency-**A**ware **T**ime-series representations through self-supervised learning. The key idea of FAT is to pretrain any backbone model to directly extract generalizable frequency patterns from time-domain signals and encode them into robust representations—eliminating the need for architectural modifications or additional modules during inference. This is achieved via a frequency reformer that amplifies critical frequency components learned through self-supervision and enforces similarity between the original and frequency-reformed time-series representations produced by the encoder. In addition, recognizing that semantically equivalent time-series can exhibit different frequency expressions—analogous to how the same phrase is pronounced differently by different speakers—FAT introduces a Knowledge-Guided Frequency Reformer that unifies the expression of frequency patterns with the same underlying semantics and extends similarity constraints to frequency-invariant augmented samples to enhance robustness of learned representation. Experiments on 14 benchmark datasets across regression and classification tasks show that FAT consistently achieves state-of-the-art performance while maintaining robustness across diverse backbone models, significantly outperforming existing pretraining methods. Our code is available at https://github.com/JiaXiangfei/FAT.

CCS Concepts

• **Computing methodologies** → **Learning latent representations**; • **Computer systems organization** → **Neural networks**; • **Mathematics of computing** → **Time series analysis**.

Keywords

Time Series, Representation Learning, Self-supervised Pretraining, Frequency-domain Modeling

ACM Reference Format:
Rui Cheng, Xiangfei Jia, Qing Li, Rong Xing, Jiwen Huang, Yu Zheng, and Zhilong Xie. 2025. FAT: Frequency-Aware Pretraining for Enhanced Time-Series Representation Learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3736952

*Qing Li is the corresponding author (liq_t@swufe.edu.cn).

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3736952

1 Introduction

Frequency-domain information is fundamental to time-series analysis across various domains, including finance [2] and healthcare [14]. In these fields, key patterns emerge at different frequency ranges, carrying essential predictive signals. For instance, low-frequency trends dominate long-term economic behaviors in financial markets, whereas high-frequency components capture short-term fluctuations. Similarly, in physiological signals such as ECG, different frequency bands correspond to distinct diagnostic markers [26]. These variations highlight the necessity of frequency-aware modeling for improving forecasting accuracy and robustness.

Despite the success of Transformer-based architectures [19, 27, 30] in time-series forecasting, they struggle to capture frequency patterns due to their reliance on time-domain operations. To address this, recent works have incorporated frequency-related modules into backbone models (e.g., Transformer architectures) to learn frequency patterns based on supervised signal [31, 44]. While the adaptions on backbone models improve their accuracy in downstream tasks, they suffer from limited generalization, as the modules are typically designed to emphasize either high- or low-frequency components based on specific task requirements [31, 37]. Since different tasks rely on frequency information at varying spectral ranges, these task-specific designs prevent the models from generalizing across diverse datasets [18].

Recently, self-supervised pretraining has emerged as a promising approach for time-series modeling. These methods design a set of pretraining tasks that enable any given backbone model to learn generalizable representations, which can be effectively adapted to diverse downstream tasks through fine-tuning [1, 27, 40]. A widely adopted strategy in time-series pretraining is masked sequence modeling, where randomly masked segments are reconstructed to enforce temporal dependencies [6, 27]. However, recent studies have identified a fundamental limitation: masked modeling makes models highly sensitive to local fluctuations and ineffective at capturing periodic information [6, 21]. A natural solution is to attach frequency-aware modules before the encoder. However, these modules often lack generalization, tending to prioritize either high- or low-frequency components, and they introduce additional computational complexity during inference [38].

To address these challenges, we propose FAT, a novel self-supervised pretraining framework that integrates **F**requency-**A**ware **T**ime-series learning into the existing pretraining paradigm. FAT enables widely used backbone models to generate frequency-aware representations directly from time-domain signals during inference, eliminating the need for architectural modifications or additional modules. This is accomplished by incorporating three key components into the pretraining phase:

Frequency Reformers: Instead of manually defining frequency properties for downstream tasks, FAT employs a self-supervised approach to identify key frequency components that generalize across diverse tasks. Specifically, the frequency reformer learns to adjust the amplitude and phase of frequency components to reconstruct the original sequence, enabling the model to capture structured frequency representations without relying on predefined task priors. Furthermore, recognizing that semantically similar sequences may exhibit different frequency compositions—analogous to how the same phrase can be pronounced with varying tones by different speakers—FAT introduces a **Knowledge-guided Frequency Reformer** that unifies the expression of frequency patterns with the same underlying semantics and adaptively maps time-series to this shared representation, enhancing the consistency and generalization of learned representations.

Frequency-similarity Constraints: To ensure that the encoder learns the frequency reforming process and generates equivalent representations during inference without the Frequency Reformer, FAT imposes similarity constraints between the original time-series and its frequency-reformed counterparts. This encourages the encoder to extract frequency features directly through time-domain operations, bridging the gap between time- and frequency-domain information and eliminating the need for Frequency Reformers during the inference.

Frequency-invariant Augmentations: Existing backbone models primarily operate in the time domain and are highly sensitive to local fluctuations, leading to inconsistencies in the learned representations. To address this, FAT applies Frequency-invariant Augmentations that introduce controlled variations in the time domain while preserving key frequency properties. This prevents the encoder from overfitting to random fluctuations, generating consistent representations across both time and frequency domains for downstream tasks.

Experiments on 14 benchmark datasets across regression and classification tasks show that FAT consistently achieves state-of-the-art performance, surpassing existing pretraining methods. Additionally, FAT demonstrates strong generalization across various backbone architectures, establishing it as a robust framework for self-supervised time-series representation learning.

2 Related Work

Self-supervised pretraining has gained increasing attention in time-series learning [5, 6, 13, 27], inspired by its success in language models [3, 29]. A key technique in this domain is masked modeling [5, 20, 40, 41], where segments or individual points are randomly masked, and the model is trained to reconstruct the missing values. Variants of this approach include tokenizing time-series data for discrete modeling [27, 38] and refining reconstruction tasks to better capture local temporal variations [20, 40].

To address the issue of temporal semantic misalignment in masked modeling, where small local temporal variations lead to divergent representations, researchers have introduced manifold learning [6, 24] and contrastive learning [1, 33, 35, 41]. Manifold learning enforces sequence-wise similarity constraints, whereas contrastive learning explicitly defines positive and negative sample pairs to enhance structural consistency. Despite their success, these efforts have focused solely on modeling variations in the time domain, overlooking the importance of frequency patterns for downstream tasks. In addition, encoders operating in the time domain inherently struggle to capture frequency characteristics effectively. The most relevant work that considers frequency-domain feature is TF-C [41], which attempts to align frequency- and time-domain representations of the same instance. However, it fails to identify key frequency patterns and generate consistent representations for

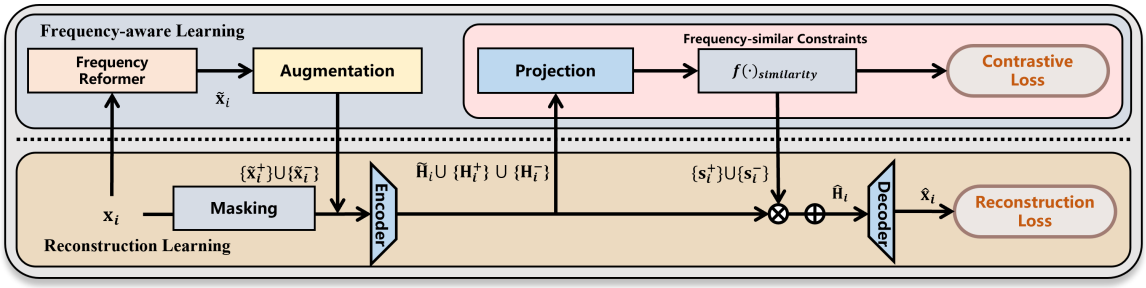

Figure 1: Overview of the FAT pretraining framework, which integrates frequency-aware learning into the conventional masked modeling paradigm. Bottom: Standard masked modeling process. Top: Frequency-aware learning process.

semantically similar time-series with different frequency expressions, as achieved by FAT.

Other studies have proposed new backbone architectures tailored to learning frequency-aware representations [9, 36–38, 43, 44]. Notably, these frequency-aware models can outperform existing pretraining approaches under certain conditions—even without pretraining—highlighting the limitations of current pretraining methods in capturing frequency-domain patterns in time-series representation learning [31, 38, 41]. However, beyond the added computational complexity introduced during inference, these architectures often exhibit specific frequency biases tailored to particular datasets. Some are designed to emphasize high-frequency components [28], while others prefer low-frequency signals [37], resulting in limited generalization across diverse datasets and tasks.

Motivated by advances in both self-supervised learning and frequency-domain feature learning architectures, we propose FAT—a pretraining framework that enables any encoder to explicitly generate frequency-aware time-series representations, free from frequency bias across the entire spectrum and without introducing additional computational complexity during inference.

3 Methods

3.1 Overall Architecture

FAT learns consistent and generalizable frequency patterns from time-domain signals and encodes them into representations, eliminating the need for architectural adaptations or additional modules during inference. This is achieved through three key components: the Knowledge-guided Frequency Reformer, Frequency-invariant Augmentations, and frequency-similarity constraints. An overview of the FAT framework is presented in Figure 1 and introduced as follows.

3.1.1 Frequency Reformer. The Frequency Reformer in previous study is applied to select important frequency components of the input time-series that facilitated downstream tasks. Specifically, given a time-series of length T, denoted as $\mathbf{x}_i \in \mathbb{R}^T$, Frequency Reformer first transforms inputs into the frequency domain by applying Discrete Fourier Transform [7], denoted as $\mathcal{F}(\cdot)$:

$$\mathbf{x}_i^{\mathcal{F}} = \mathcal{F}(x_i) \tag{1}$$

where $x_i^{\mathcal{F}} \in \mathbb{C}^T$ is the frequency representation of x_i. Then a frequency reformer operator, denoted as $\mathbf{M}_{\mathcal{F}}(\cdot)$, is applied to alter the amplitude and phase of frequency components in its spectrum:

$$\tilde{\mathbf{x}}_i^{\mathcal{F}} = \mathbf{M}_{\mathcal{F}}(\mathcal{F}(\mathbf{x}_i)), \tag{2}$$

An inverted Discrete Fourier Transform $\mathcal{F}^{-1}(\cdot)$ is then applied to map the reformed frequency signal $\tilde{\mathbf{x}}_i^{\mathcal{F}}$ back into the time domain, obtaining the reshaped output signal $\tilde{\mathbf{x}}_i$:

$$\tilde{\mathbf{x}}_i = \mathcal{F}^{-1}(\tilde{\mathbf{x}}_i^{\mathcal{F}}) \tag{3}$$

Conventional frequency filters that are predefined for specific purposes are typically employed as the frequency reformer operator $\mathbf{M}_{\mathcal{F}}(\cdot)$. These include denoising methods that retain only the top-k frequency components with the highest magnitudes and selective filtering approaches that preserve only specific spectral regions, such as high-pass or low-pass filters [37].

Instead of relying on predefined filters, the frequency reformer should be capable of learning a robust frequency representation that adaptively transforms input frequency components to generalize across various tasks. To achieve this, we propose the **Knowledge-guided Frequency Reformer**, a data-dependent frequency reformer that identifies important frequency components and standardizes the expression of frequency patterns with the same underlying meaning by adaptively transforming them into a unified representation. The details of this approach are provided in Section 3.2.

3.1.2 Augmentation. Given a time-series x_i, the augmentation process aims to generate a set of candidate variants of x_i, containing both relevant and irrelevant information from the original input:

$$\{\tilde{\mathbf{x}}_i^+\}_{j=1}^K, \{\tilde{\mathbf{x}}_i^-\}_{j=1}^K = \text{Aug}(\tilde{\mathbf{x}}_i), \tag{4}$$

where $\{\tilde{\mathbf{x}}_i^+\}_{j=1}^K$ and $\{\tilde{\mathbf{x}}_i^-\}_{j=1}^K$ represent K positive (relevant) and negative (irrelevant) augmented samples, respectively.

A common approach in previous studies for generating positive samples is to apply a random masking strategy to the original time-series x_i, encouraging the encoder to learn temporal dependencies through sequence reconstruction. Instead of applying augmentation strategies directly to $\mathbf{x}_i$, FAT applies a set of **Frequency-invariant Augmentations** to the reformed input $\tilde{\mathbf{x}}_i$, introducing controlled variations in the time domain while preserving key frequency properties, as detailed in Section 3.3. These augmented samples serve two purposes: (1) guiding the encoder to generate similar embeddings for the original and its frequency-reformed

counterpart through frequency-similarity constraints, and (2) bridging the gap between time- and frequency-domain information via reconstruction.

3.1.3 Encoder. The encoder$(\cdot)$ projects the input time-series into deep representations. Specifically, given an input time-series $\mathbf{x}_i \in \mathbb{R}^T$, the encoder transforms the signal into a latent representation $\mathbf{h}_i \in \mathbb{R}^D$:

$$\mathbf{h}_i = \text{Encoder}(\mathbf{x}_i), \tag{5}$$

where D represents the dimension of the latent space.

Existing time-series forecasting models often employ task-specific architectures tailored to capture specific frequency characteristics. However, these approaches tend to be highly complex and are not inherently designed to support a general representation learning framework. To improve training and inference efficiency while ensuring fair comparisons with existing methods, FAT adopts the vanilla Transformer [30] encoder with patched input[27], for regression, and the 1D-ResNet [10] for classification [12]. Experimental results demonstrate that FAT consistently achieves state-of-the-art (SOTA) performance across various backbone models.

3.1.4 Frequency-Similar Constraints. As previously discussed, a key property of FAT is to to directly extract consistent and generalizable frequency patterns from time-domain signals and encode them into representations, eliminating the need for architectural adaptations or additional modules for capturing frequency-domain feature during inference.

To achieve this, we impose a similarity constraint between the deep representations of the original time-series and its frequency-reformed counterpart. Specifically, we first project their representations into a space for similarity comparison using a projector and compute their similarity as::

$$\mathbf{z}_i = \text{Projector}_z(\mathbf{h}_i), \quad s_i = \text{sim}(\mathbf{z}_i, \tilde{\mathbf{z}}_i), \tag{6}$$

where Projector$(\cdot)$ is a simple MLP layer [11], and similarity is measured using the cosine similarity.

To ensure that the frequency similarity measure remains invariant to random distortions in the time domain, we extend the similarity constraint to all augmented samples of $\tilde{\mathbf{x}}_i$, i.e., Aug$(\tilde{\mathbf{x}}_i)$:

$$\begin{aligned} \text{Positive pairs} &: (\mathbf{z}_i, \tilde{\mathbf{z}}_i^+), \quad \tilde{\mathbf{z}}_i^+ \in \{\tilde{\mathbf{z}}_i^+\}_{j=1}^K, \\ \text{Negative pairs} &: (\mathbf{z}_i, \tilde{\mathbf{z}}_i^-), \quad \tilde{\mathbf{z}}_i^- \in \{\tilde{\mathbf{z}}_i^-\}_{j=1}^K. \end{aligned} \tag{7}$$

The softmax function is applied to their similarity scores to obtain the normalized similarity distribution:

$$a_{i,j}^{\mathcal{T}} = \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2K} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}, \tag{8}$$

where τ is the temperature parameter controlling the sharpness of the similarity distribution.

The frequency-similarity constraints is defined as the contrastive loss between the representations of the original time-series and its augmented samples:

$$\mathcal{L}_{\text{contrastive}} = -\sum_{k \in K^+} \log a_{i,k}^{\mathcal{T}}. \tag{9}$$

By enforcing frequency-similarity constraints, the encoder is directly trained to capture frequency-aware representations from time-domain signals, thereby eliminating the need for frequency-specific architectures (e.g., explicit frequency reforming processes in Section 3.1.1) during inference.

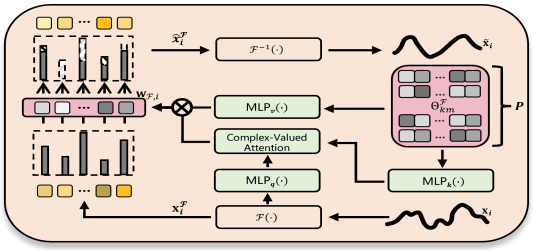

Figure 2: Overview of the Knowledge-guided Frequency Reformer, where frequency components are transformed into a unified frequency representation by retrieving reforming patterns stored in the knowledge memory.

3.1.5 Reconstruction. Beyond learning temporal dependencies, FAT encourages the encoder to extract key features from time-series with similar frequency components and transfer this knowledge for reconstruction. Furthermore, it aims to help the encoder bridge the frequency and time domains by leveraging frequency-domain information to reconstruct time-domain signals.

Therefore, similar to previous studies, we also include a reconstruction task. However, instead of using the unmasked parts of the original sequence, we rely on augmented samples that retain key frequency components while introducing random distortions in the time domain.

Specifically, FAT first aggregates information from multiple augmented time-series derived from its frequency-reformed counterparts, where subsequences of varying lengths have been distorted:

$$\hat{\mathbf{h}}_i = \text{Agg}(\mathcal{H}_i, \mathbf{a}_i) = \sum_{k=1}^{2K} a_{i,k}^{\mathcal{T}} \tilde{\mathbf{h}}_k. \tag{10}$$

It then reconstructs the original time-series using the aggregated information:

$$\hat{\mathbf{x}}_i = \text{Decoder}(\hat{\mathbf{h}}_i). \tag{11}$$

Following masked modeling paradigm, the reconstruction loss that supervises the pretraining process is defined as:

$$\mathcal{L}_{\text{reconstruction}} = ||\mathbf{x}_i - \hat{\mathbf{x}}_i||_2^2. \tag{12}$$

The overall optimization objective of FAT is formulated as:

$$\min_{\Theta} \quad \mathcal{L}_{\text{reconstruction}} + \lambda \mathcal{L}_{\text{contrastive}}, \tag{13}$$

where the hyperparameter λ controls the relative weight of the contrastive loss. To dynamically balance the loss terms, we adopt the adaptive weighting strategy proposed in [15].

3.2 Knowledge-guided Frequency Reformer

As mentioned earlier, the Frequency Reformer is designed to extract key frequency components that facilitate downstream tasks. Instead of using predefined filters such as high-pass or low-pass filters, FAT applies a learnable parameter $\mathbf{w}_{\mathcal{F}} \in \mathbb{C}^T$, which element-wise multiplies the frequency representation of $\mathbf{x}_i$ ($\mathcal{F}(\mathbf{x}_i)$) to select

relevant frequency components based on supervised signals:

$$\tilde{\mathbf{x}}_i^{\mathcal{F}} = \mathbf{M}_{\mathcal{F}}(\mathcal{F}(\mathbf{x}_i)) = \mathbf{w}_{\mathcal{F}} \odot \mathcal{F}(\mathbf{x}_i). \tag{14}$$

However, time-series with similar semantics can be expressed in different frequency forms, analogous to how the same phrase is pronounced differently by different speakers. The inconsistency among semantically similar time-series prevents frequency reformers from effectively identifying robust frequency features.

Inspired by how humans recognize and map different pronunciations of the same word—despite variations in tone, speed, or accent—to a consistent phonetic structure, speech recognition systems do not treat every variation as entirely distinct but instead align them based on shared phonetic patterns. Similarly, time-series variations are not arbitrary but follow structured and predictable variations. Therefore, we propose Knowledge-guided Frequency Reformer, a data-dependent approach that unifies the expression of frequency patterns with the same underlying semantics.

Specifically, as in Figure 2, FAT adaptively reforms time-series of various frequency expressions to this shared representation via mapping patterns stored in a parameterized knowledge memory, $\Theta_{km}^{\mathcal{F}} \in \mathbb{C}^{P \times T}$, where P represents the number of patterns. Given the frequency-domain representation of the input time-series, $\mathcal{F}(\mathbf{x}_i) \in \mathbb{C}^T$, we implement the Complex-Valued Attention Mechanism [8] to retrieve reweighting coefficients from the parameterized knowledge memory, $\Theta_{km}^{\mathcal{F}}$. The matching score between the input time series i and the reweighting knowledge entry j is computed as:

$$s_{i,j}^{\mathcal{F}} = \langle \mathbf{q}_i, \mathbf{k}_j \rangle = |\mathbf{q}_i||\mathbf{k}_j| \exp(i(\phi_{\mathbf{q}_i} - \phi_{\mathbf{k}_j})), \tag{15}$$

where the query vector $\mathbf{q}_i = \mathrm{MLP}_q(\mathcal{F}(\mathbf{x}_i))$ is obtained by applying an MLP to the input frequency representation, while the keys $\mathbf{k}_j \in \mathbb{C}^T$ and values $\mathbf{v}_j \in \mathbb{C}^T$ are derived from the parameterized knowledge memory $\Theta_{km}^{\mathcal{F}}$ via two additional MLPs, denoted as $\mathrm{MLP}_k(\cdot)$ and $\mathrm{MLP}_v(\cdot)$. The terms $\phi_{\mathbf{q}_i}, \phi_{\mathbf{k}_j} \in \mathbb{R}^T$ represent the phase components of $\mathbf{q}_i$ and $\mathbf{k}_j$, respectively. The real part $\mathcal{R}(\langle \mathbf{q}_i, \mathbf{k}_j \rangle) = |\mathbf{q}_i||\mathbf{k}_j| \cos(\phi_{\mathbf{q}_i} - \phi_{\mathbf{k}_j})$ is taken to ensure a real-valued similarity measure that maintains symmetry and rotational invariance. The attention weights ($a_j^{\mathcal{F}}$) are computed by applying a softmax over all P knowledge entries. The frequency-domain reweighting vector ($\mathbf{w}_{\mathcal{F},i}$) is then obtained by aggregating the reweighted knowledge representations based on these attention weights.:

$$a_j^{\mathcal{F}} = \mathrm{softmax}\left(\frac{\mathcal{R}(s_{i,j}^{\mathcal{F}})}{\sum_{k=1}^{P} s_{i,k}^{\mathcal{F}}}\right), \quad \mathbf{w}_{\mathcal{F},i} = \sum_{j=1}^{P} a_j^{\mathcal{F}} \mathbf{v}_j. \tag{16}$$

The vector $\mathbf{w}_{\mathcal{F},i}$ then replaces $\mathbf{w}_{\mathcal{F}}$ in Eq. 14 to adaptively reform the frequency components of $\mathbf{x}_i$.

3.3 Frequency-invariant Augmentations

These augmented samples serve two purposes: (1) guiding the encoder to generate similar embeddings for the original and its frequency-reformed counterpart through frequency-similarity constraints, and (2) bridging the gap between time- and frequency-domain information via reconstruction.

Therefore, instead of applying a random masking strategy to the original time-series, we distort variable-length continuous segments in the frequency-reformed time-series. The augmentation process consists of two steps: (1) Generate continuous indexes. (2) Apply frequency-invariant augmentation strategies.

3.3.1 Generate Continuous Indexes. Recent studies have shown that short masked sequences can often be trivially reconstructed by replicating or averaging adjacent values, making the learned representation overly sensitive to local variations. To mitigate this, we generate continuous index sets of variant lengths.

Unlike conventional Bernoulli-based masking, which generates discrete indexes, we apply a Markov process-based structured masking strategy: we apply a Markov process-based structured masking strategy:

$$\mathbf{M}_{\mathcal{T}} \sim \mathrm{Markov}(p_m, p_{um}, r), \tag{17}$$

where $\mathbf{M}_{\mathcal{T}} \in \{0,1\}^T$ is a binary masking vector with a masking ratio r, $p_m = \frac{1}{l_m}$ and $p_{um} = \frac{1}{l_u}$ are transition probabilities for masked and unmasked states, ensuring that masked (l_m) and unmasked (l_u) segment lengths satisfy, $l_u = \frac{1-r}{r} l_m$. This approach enforces a geometric distribution for masked segments with an expected length of l_m, compelling the model to capture both long- and short-term dependencies [40].

3.3.2 Apply Augmentation Strategy. Given an input sequence $\tilde{\mathbf{x}}_i$, we define operators $\mathcal{T}_+(\cdot)$ and $\mathcal{T}_-(\cdot)$ to generate K positive ($\{\tilde{\mathbf{x}}_i^+\}_{k=1}^K$) and negative ($\{\tilde{\mathbf{x}}_i^-\}_{k=1}^K$) augmented samples, respectively:

$$\mathcal{T}_+(\tilde{\mathbf{x}}_i) \to \{\tilde{\mathbf{x}}_i^+\}_{k=1}^K, \quad \mathcal{T}_-(\tilde{\mathbf{x}}_i) \to \{\tilde{\mathbf{x}}_i^-\}_{k=1}^K. \tag{18}$$

Positive Sample Augmentation. Following prior work, we adopt the masking operation ($\mathcal{T}_+^{\mathcal{M}}(\cdot)$), where a fraction r of the time-series is randomly masked to simulate missing data:

$$\mathcal{T}_+^{\mathcal{M}}(\mathbf{x}_i, \mathbf{M}_{\mathcal{T}}) = \mathbf{x}_i \odot \mathbf{M}_{\mathcal{T}}. \tag{19}$$

While reconstructing long sequences captures both short- and long-term dependencies, masking excessively long segments can make the task overly difficult, degrading downstream performance (Section 4.3.4). To mitigate this, we propose two augmentation strategies that generate time-series with preserved frequency patterns but distorted in the time domain to guide long-sequence reconstruction:

- Amplitude Scaling ($\mathcal{T}_+^{\mathcal{A}}(\cdot)$): This augmentation scales the masked regions of the time-series to simulate variations in signal strength:

$$\mathcal{T}_+^{\mathcal{A}}(\mathbf{x}_i, \mathbf{M}_{\mathcal{T}}, \mathbf{a}) = \mathbf{x}_i \odot (1 + (\mathbf{a} - 1) \odot \mathbf{M}_{\mathcal{T}}), \tag{20}$$

where $\mathbf{a} \in \mathbb{R}^T \sim \mathrm{Uniform}(0.5, 2)$ is the amplitude scaling factor.

- Noise Injection ($\mathcal{T}_+^{\mathcal{N}}(\cdot)$): This augmentation injects t-distributed noise into the masked regions of the time-series to simulate real-world fluctuations:

$$\mathcal{T}_+^{\mathcal{N}}(\mathbf{x}_i, \mathbf{M}_{\mathcal{T}}, \mathbf{u}) = \mathbf{x}_i + \mathbf{u} \odot \mathbf{M}_{\mathcal{T}}, \tag{21}$$

where $\mathbf{u} \in \mathbb{R}^T \sim T(v)$ represents noise sampled from a standard Student's t-distribution with degrees of freedom $v = 5$.

These augmentation strategies not only help encode consistent frequency representations under time-domain distortions but also encourage the encoder to leverage and align information from both the frequency and time domains.

Negative Sample Augmentation. To prevent over-fitting to easy negatives and reduce computational cost, we select hard negatives for each input time-series. Specifically, the negative sample for an input sequence is chosen as the most similar sequence within the

reformed time-series in the training batch $\{\mathbf{x}_i\}_{i=1}^{N}$. The similarity is measured in the frequency domain and is defined in Eq. 15.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets. We use 14 well-established datasets, covering two primary tasks in time-series analysis: regression and classification. For regression, we evaluate our framework on 8 widely used benchmark datasets: Weather, Traffic, Electricity [34], Exchange [17], and the ETT datasets (ETTH1, ETTH2, ETTM1, ETTM2) [42]. For classification, we pretrain our model on the SleepEEG dataset [41], a widely used benchmark for transfer learning with abundant samples [6, 41]. The downstream evaluation is conducted on 6 classification datasets, including Gesture, FD-B, EMG, EPI, HAR, and a collection of 128 datasets from the UCR archive [41]. All datasets are preprocessed following the methods outlined in [23]. Detailed descriptions are provided in Appendix A.

4.1.2 Baseline Settings. We compare our framework with 9 most advanced time-series pre-training frameworks. These include masked modeling methods: SimMTM [6], Timesiam [4], and TST [40]. Additionally, we evaluate contrastive learning methods, including TF-C [41], LaST [32], PatchTST [27], TS2Vec [39], TimesURL [22], and InfoTS [25].

For a fair comparison, all baselines use the Transformer with patched time-series input [27] as the backbone for regression and 1D-ResNet [10] for classification, following previous work. The hyper-parameters in the backbones were either searched within the same range as or set according to their original configurations.

The Adam optimizer [16] was employed with a learning rate ranging from {0.01 to 0.0001}. During pretraining, the maximum number of training steps was capped at 50, while in fine-tuning, early stopping with a patience of 10 epochs was applied to ensure optimal performance.

4.2 Main Results

We summarize the results for both regression and classification tasks in Figure 3. Each experiment was repeated five times with different random seeds, and the reported results are averaged. FAT's average improvement over all baselines is statistically significant at the 95% confidence level. As demonstrated, FAT exhibits significant improvement over the state-of-the-art (SOTA) across diverse datasets and backbone models.

4.2.1 Regression. We first compares our framework to representative baselines across eight benchmarks with various prediction lengths under in-domain settings, where encoders are pre-trained and fine-tuned on the same dataset [6]. As illustrated in Table 2, our model consistently outperforms the baseline models across all tests, showing the effectiveness of the proposed method. Specifically, FAT improves the average performance by 6.9% over random initialization and by 1.4% over the second-best baseline model in terms of Mean Squared Error (MSE).

While pretraining frameworks for time-series representation outperform training from scratch, their effectiveness remains limited as they operate purely in the time domain. Additionally, their representations are overly sensitive to local variations, resulting in inconsistent embeddings and failing to capture essential frequency semantics.

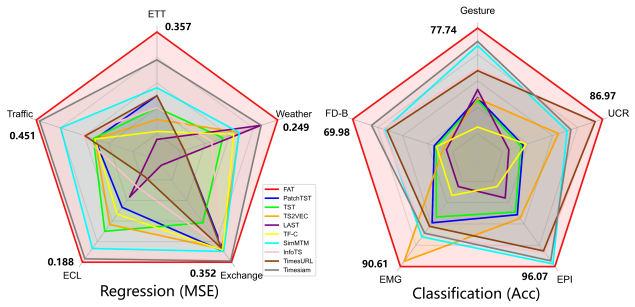

Figure 3: Comparison of the performance of FAT and all baselines on the regression and classification task.

In contrast, FAT overcomes the limitations of existing pretraining frameworks by enabling the encoder to directly extract frequency patterns from time-domain signals without requiring explicit frequency transformations during inference. The Knowledge-guided Frequency Reformer unifies representations of important frequency components across semantically similar time-series, addressing inconsistencies in prior methods. Meanwhile, frequency-similarity constraints allow direct frequency pattern extraction in the time domain, while Frequency-invariant Augmentations mitigate sensitivity to local variations.

4.2.2 Classification. Following previous works in transfer learning [6], we conduct cross-domain experiments by pretraining the model on the SleepEEG dataset and fine-tuning it on diverse classification tasks where in-domain data is scarce for effective training. This cross-domain evaluation introduces significant challenges in handling mismatched data distributions.

The results consistently demonstrate that FAT outperforms random initialization across all regression benchmarks and surpasses other baselines in both regression and classification tasks. Specifically, FAT improves the average performance by 117.4% over random initialization and by 7.8% over the second-best baseline model in terms of accuracy. This highlights FAT's robustness across various downstream tasks and different backbone models, particularly in adapting to mismatched data distributions.

We attribute this to the unified frequency expressions and reforming patterns learned by the Knowledge-guided Frequency Reformer, which captures multiple reforming rules to align data with mismatched distributions by mapping them to a unified structure, effectively enhancing generalization.

4.3 Model Analysis

4.3.1 Ablation Study. To further investigate the effectiveness of each mechanism in FAT, we conduct an ablation analysis by systematically removing key components and evaluating their impact. Specifically, we consider three ablation models:

• *w/o p-reform*: The frequency reformer is removed during the **pretraining stage**, and augmentations are applied directly to the original inputs.

Table 1: Full results for the in-domain setting of regression. Pre-training and fine-tuning are performed on the same datasets. The standard deviations are within 0.005 for MSE and within 0.004 for MAE.

Methods		Random init.		PatchTST		TST		TS2Vec		LaST		TFC		SimMTM		InfoTS		TimesURL		Timesiam		**FAT**	
Metric		mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae
ETTh1	96	0.420	0.423	0.369	0.391	0.377	0.401	0.381	0.400	0.396	0.413	0.399	0.420	0.367	0.389	0.376	0.395	0.372	0.392	0.378	0.401	0.368	0.392
	192	0.465	0.449	0.426	0.425	0.432	0.436	0.421	0.427	0.457	0.451	0.444	0.449	0.424	0.423	0.410	0.423	0.417	0.420	0.422	0.430	0.411	0.419
	336	0.504	0.470	0.475	0.458	0.475	0.461	0.468	0.452	0.507	0.478	0.479	0.467	0.473	0.456	0.457	0.440	0.455	0.441	0.459	0.452	0.452	0.443
	720	0.502	0.492	0.496	0.495	0.525	0.500	0.553	0.507	0.516	0.508	0.491	0.490	0.494	0.493	0.493	0.470	0.477	0.468	0.459	0.466	0.454	0.458
	Avg	0.473	0.459	0.442	0.442	0.452	0.450	0.456	0.447	0.469	0.463	0.453	0.457	0.440	0.440	0.434	0.432	0.430	0.430	0.430	0.437	**0.421**	**0.428**
ETTh2	96	0.297	0.345	0.301	0.355	0.304	0.358	0.297	0.343	0.294	0.345	0.302	0.345	0.299	0.352	0.293	0.348	0.297	0.346	0.293	0.345	0.283	0.337
	192	0.388	0.400	0.382	0.401	0.379	0.403	0.366	0.392	0.379	0.395	0.369	0.392	0.380	0.398	0.370	0.392	0.372	0.391	0.370	0.392	0.356	0.385
	336	0.426	0.434	0.420	0.429	0.412	0.432	0.416	0.430	0.423	0.436	0.412	0.428	0.422	0.432	0.421	0.422	0.416	0.427	0.410	0.424	0.401	0.419
	720	0.431	0.446	0.426	0.446	0.438	0.457	0.424	0.447	0.445	0.460	0.428	0.446	0.428	0.449	0.440	0.452	0.437	0.447	0.418	0.440	0.419	0.433
	Avg	0.386	0.406	0.382	0.408	0.383	0.413	0.376	0.403	0.385	0.409	0.378	0.403	0.382	0.408	0.381	0.404	0.381	0.403	0.373	0.400	**0.365**	**0.394**
ETTm1	96	0.330	0.368	0.322	0.362	0.319	0.360	0.325	0.364	0.345	0.381	0.353	0.378	0.317	0.356	0.329	0.368	0.320	0.357	0.319	0.360	0.313	0.355
	192	0.369	0.385	0.368	0.393	0.360	0.387	0.370	0.389	0.372	0.391	0.361	0.384	0.362	0.387	0.373	0.391	0.364	0.377	0.353	0.379	0.347	0.374
	336	0.400	0.407	0.393	0.411	0.391	0.408	0.405	0.415	0.412	0.420	0.392	0.406	0.387	0.405	0.407	0.413	0.398	0.400	0.383	0.402	0.375	0.396
	720	0.460	0.439	0.450	0.445	0.449	0.445	0.471	0.452	0.462	0.448	0.448	0.440	0.443	0.438	0.466	0.448	0.455	0.434	0.440	0.436	0.432	0.430
	Avg	0.390	0.400	0.383	0.403	0.380	0.400	0.393	0.405	0.398	0.410	0.389	0.402	0.377	0.397	0.394	0.405	0.384	0.392	0.374	0.394	**0.367**	**0.389**
ETTm2	96	0.175	0.258	0.177	0.265	0.181	0.265	0.174	0.261	0.177	0.258	0.281	0.327	0.175	0.262	0.198	0.277	0.200	0.287	0.175	0.261	0.171	0.257
	192	0.247	0.307	0.246	0.310	0.247	0.309	0.247	0.306	0.252	0.309	0.241	0.302	0.244	0.307	0.255	0.320	0.266	0.329	0.241	0.303	0.236	0.299
	336	0.309	0.345	0.310	0.348	0.314	0.354	0.306	0.345	0.307	0.344	0.304	0.343	0.312	0.351	0.305	0.341	0.308	0.350	0.300	0.341	0.292	0.333
	720	0.408	0.403	0.408	0.405	0.408	0.407	0.427	0.415	0.404	0.402	0.404	0.403	0.410	0.408	0.399	0.397	0.407	0.405	0.399	0.398	0.391	0.389
	Avg	0.285	0.328	0.285	0.332	0.288	0.334	0.289	0.332	0.285	0.328	0.308	0.344	0.285	0.332	0.289	0.334	0.295	0.343	0.279	0.326	**0.273**	**0.320**
Weather	96	0.177	0.218	0.191	0.229	0.177	0.221	0.174	0.216	0.170	0.212	0.177	0.218	0.184	0.220	0.185	0.223	0.189	0.225	0.171	0.213	0.170	0.211
	192	0.225	0.259	0.225	0.265	0.223	0.260	0.220	0.257	0.215	0.253	0.222	0.257	0.217	0.255	0.229	0.258	0.233	0.261	0.217	0.253	0.215	0.249
	336	0.278	0.297	0.284	0.308	0.279	0.301	0.276	0.297	0.272	0.295	0.277	0.296	0.273	0.296	0.282	0.295	0.286	0.299	0.272	0.293	0.269	0.291
	720	0.354	0.348	0.362	0.357	0.355	0.350	0.352	0.346	0.349	0.344	0.353	0.346	0.348	0.344	0.349	0.337	0.356	0.344	0.348	0.343	0.343	0.341
	Avg	0.259	0.281	0.266	0.290	0.259	0.283	0.256	0.279	0.252	0.276	0.257	0.279	0.256	0.279	0.261	0.278	0.266	0.282	0.252	0.276	**0.249**	**0.273**
Exchange	96	0.084	0.201	0.087	0.206	0.098	0.218	0.084	0.201	0.096	0.220	0.083	0.201	0.083	0.202	0.087	0.205	0.084	0.195	0.084	0.203	0.082	0.200
	192	0.187	0.307	0.186	0.307	0.187	0.308	0.185	0.306	0.190	0.313	0.173	0.296	0.182	0.303	0.182	0.304	0.185	0.305	0.176	0.300	0.175	0.297
	336	0.337	0.422	0.342	0.423	0.330	0.418	0.328	0.415	0.409	0.455	0.332	0.418	0.346	0.427	0.332	0.420	0.328	0.417	0.310	0.404	0.311	0.408
	720	0.858	0.695	0.827	0.685	0.925	0.731	0.856	0.696	1.035	0.749	0.860	0.698	0.831	0.689	0.860	0.701	0.857	0.686	0.842	0.690	0.838	0.687
	Avg	0.367	0.406	0.361	0.405	0.385	0.419	0.363	0.405	0.433	0.434	0.362	0.403	0.361	0.405	0.365	0.408	0.364	0.401	0.353	0.399	**0.352**	**0.398**
ECL	96	0.193	0.291	0.174	0.271	0.171	0.267	0.175	0.268	0.183	0.275	0.171	0.263	0.164	0.255	0.188	0.274	0.178	0.264	0.164	0.245	0.162	0.249
	192	0.199	0.297	0.189	0.285	0.181	0.276	0.183	0.275	0.190	0.281	0.188	0.277	0.178	0.268	0.182	0.270	0.198	0.281	0.173	0.256	0.174	0.261
	336	0.216	0.312	0.202	0.298	0.197	0.291	0.199	0.292	0.205	0.296	0.205	0.291	0.190	0.280	0.191	0.286	0.220	0.291	0.189	0.275	0.187	0.273
	720	0.257	0.345	0.250	0.338	0.237	0.325	0.240	0.324	0.248	0.330	0.244	0.322	0.235	0.318	0.231	0.321	0.249	0.336	0.229	0.310	0.228	0.306
	Avg	0.216	0.311	0.204	0.298	0.197	0.290	0.199	0.290	0.207	0.296	0.202	0.288	0.192	0.280	0.196	0.285	0.213	0.297	0.189	0.272	**0.188**	**0.272**
Traffic	96	0.472	0.305	0.462	0.298	0.478	0.292	0.469	0.291	0.506	0.330	0.465	0.301	0.442	0.285	0.435	0.472	0.437	0.474	0.429	0.279	0.425	0.273
	192	0.474	0.304	0.472	0.319	0.469	0.316	0.477	0.293	0.503	0.326	0.470	0.311	0.452	0.305	0.489	0.305	0.491	0.306	0.442	0.282	0.440	0.283
	336	0.491	0.331	0.494	0.337	0.482	0.323	0.494	0.301	0.517	0.332	0.498	0.320	0.473	0.322	0.498	0.332	0.500	0.334	0.456	0.288	0.455	0.292
	720	0.523	0.327	0.519	0.346	0.516	0.327	0.508	0.319	0.552	0.349	0.514	0.326	0.497	0.331	0.495	0.340	0.497	0.342	0.486	0.307	0.484	0.313
	Avg	0.490	0.317	0.487	0.325	0.486	0.315	0.487	0.301	0.520	0.334	0.487	0.315	0.466	0.311	0.479	0.362	0.481	0.364	0.453	0.289	**0.451**	**0.290**

Table 2: Results of cross-domain classification, where encoders are pre-trained on the SleepEEG dataset and fine-tuned on other datasets. Accuracy(%) is reported, and full metrics are presented in Appendix C.

Method	Rand. init.	PatchTST	TST	TS2Vec	LaST	TF-C	SimMTM	InfoTS	TimesURL	Timesiam	**FAT**
Gesture	32.50	63.47	62.91	63.33	65.14	57.50	74.17	69.52	69.03	75.02	**77.74**
FD-B	31.39	46.19	45.62	43.59	42.64	45.53	60.74	61.14	59.93	64.47	**69.98**
EMG	46.34	82.88	81.86	89.68	76.50	78.05	85.37	84.87	83.51	84.71	**90.61**
EPI	39.79	77.59	76.64	78.91	71.63	67.56	95.13	91.33	90.49	94.03	**96.07**
UCR	49.03	60.85	60.10	72.50	56.17	61.88	75.34	84.11	84.53	76.56	**86.97**

Table 3: Ablation studies were conducted on FAT.

Input-96		*Rand.init.*		*w/o p-reform*		*w/o contrast*		*w/o contrast + f-reform*		*w/o aug*		**FAT**	
Predict-*O*		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTH1	96	0.420	0.423	0.367	0.389	0.377	0.396	0.382	0.405	0.408	0.371	0.368	0.392
	192	0.465	0.449	0.424	0.423	0.438	0.424	0.427	0.435	0.431	0.442	0.411	0.419
	336	0.504	0.470	0.473	0.456	0.478	0.443	0.464	0.457	0.461	0.480	0.452	0.443
	720	0.502	0.492	0.494	0.493	0.474	0.471	0.468	0.471	0.479	0.495	0.454	0.458
	Avg	0.473	0.459	0.440	0.440	0.442	0.434	0.435	0.442	0.445	0.447	**0.421**	**0.428**
ETTH2	96	0.297	0.345	0.299	0.352	0.294	0.345	0.286	0.340	0.299	0.349	0.283	0.337
	192	0.388	0.400	0.380	0.398	0.369	0.395	0.360	0.375	0.382	0.404	0.356	0.385
	336	0.426	0.434	0.422	0.432	0.420	0.432	0.405	0.413	0.425	0.436	0.401	0.419
	720	0.431	0.446	0.428	0.449	0.430	0.446	0.424	0.440	0.436	0.457	0.419	0.433
	Avg	0.386	0.406	0.382	0.408	0.378	0.405	0.369	0.392	0.386	0.412	**0.365**	**0.394**
ETTM1	96	0.330	0.368	0.317	0.356	0.322	0.361	0.323	0.364	0.325	0.371	0.313	0.355
	192	0.369	0.385	0.362	0.387	0.364	0.383	0.357	0.383	0.366	0.393	0.347	0.374
	336	0.400	0.407	0.387	0.405	0.396	0.406	0.387	0.406	0.391	0.407	0.375	0.396
	720	0.460	0.439	0.443	0.438	0.456	0.444	0.445	0.441	0.459	0.451	0.432	0.430
	Avg	0.390	0.400	0.377	0.397	0.385	0.399	0.378	0.399	0.385	0.406	**0.367**	**0.389**
ETTM2	96	0.175	0.258	0.175	0.262	0.179	0.262	0.177	0.264	0.183	0.266	0.171	0.257
	192	0.247	0.307	0.244	0.307	0.244	0.304	0.244	0.306	0.249	0.310	0.236	0.299
	336	0.309	0.345	0.312	0.351	0.303	0.345	0.303	0.345	0.321	0.355	0.292	0.333
	720	0.408	0.403	0.410	0.408	0.402	0.403	0.403	0.402	0.413	0.411	0.391	0.389
	Avg	0.285	0.328	0.285	0.332	0.282	0.329	0.282	0.329	0.292	0.336	**0.273**	**0.320**

Table 4: Experiments on Variations of Frequency Reformer.

Input-96		$FAT_{Random-k}$		FAT_{Top-k}		FAT_{Shared}		**FAT**	
Predict-O		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTH1	96	0.393	0.417	0.396	0.421	0.387	0.409	0.368	0.392
	192	0.436	0.444	0.440	0.450	0.429	0.439	0.411	0.419
	336	0.474	0.468	0.478	0.474	0.468	0.463	0.452	0.443
	720	0.473	0.484	0.481	0.485	0.469	0.473	0.454	0.458
	Avg	0.444	0.453	0.449	0.457	0.438	0.446	**0.421**	**0.428**
ETTH2	96	0.293	0.346	0.295	0.351	0.290	0.343	0.283	0.337
	192	0.369	0.383	0.372	0.388	0.364	0.380	0.356	0.385
	336	0.415	0.422	0.418	0.426	0.407	0.417	0.401	0.419
	720	0.433	0.452	0.439	0.454	0.429	0.443	0.419	0.433
	Avg	0.378	0.401	0.381	0.405	0.373	0.396	**0.365**	**0.394**
ETTM1	96	0.331	0.374	0.335	0.376	0.325	0.368	0.313	0.355
	192	0.365	0.392	0.370	0.394	0.361	0.385	0.347	0.374
	336	0.396	0.414	0.400	0.420	0.392	0.408	0.375	0.396
	720	0.456	0.451	0.460	0.457	0.448	0.445	0.432	0.430
	Avg	0.387	0.408	0.391	0.412	0.381	0.402	**0.367**	**0.389**
ETTM2	96	0.181	0.271	0.183	0.273	0.178	0.267	0.171	0.257
	192	0.249	0.313	0.252	0.317	0.246	0.308	0.236	0.299
	336	0.310	0.352	0.313	0.357	0.305	0.348	0.292	0.333
	720	0.412	0.413	0.416	0.415	0.406	0.405	0.391	0.389
	Avg	0.288	0.337	0.291	0.340	0.284	0.332	**0.273**	**0.320**

• *w/o contrast*: The similarity constraint between the original time-series and its frequency-reformed counterparts is removed.

• *w/o contrast + f-reform*: The similarity constraint is removed, and the learned Knowledge-guided frequency reformer is applied before the encoder during **fine-tuning** stage.

• *w/o aug*: The augmentation strategy is replaced with random masking tokens, as in previous research.

Table 3 presents the results of our ablation study, highlighting the distinct role of each component:

1. Frequency reformer is crucial for learning frequency-domain features. Removing the frequency reformer from the pre-training stage (*w/o p-reform*) significantly degrades performance, confirming that frequency modeling is essential for time-series representation. Without it, the encoder relies solely on time-domain features, limiting its ability to extract structured frequency patterns.

2. Frequency-similarity constraints are essential for extracting frequency patterns from the time domain. Removing the frequency-similarity constraint (*w/o contrast*) significantly degrades performance. However, when reintroducing the Knowledge-guided Frequency Reformer during fine-tuning (*w/o contrast + f-reform*), the results become comparable to FAT. This indicates that the frequency-similarity constraint effectively transfers the frequency-domain knowledge and transformation process learned by the Knowledge-guided Frequency Reformer to the encoder. As a result, the encoder can directly operate in the time domain without requiring additional modules for specialized frequency processing.

3. Frequency-invariant Augmentation Enhances Robustness to Time-Domain Variations. Without augmentation (*w/o aug*), the model becomes overly sensitive to local perturbations, leading to unstable representations under small temporal distortions. By mitigating overfitting to high-frequency noise, augmentation helps the model recognize frequency-similar sequences despite local variations, ensuring robust and transferable representations.

These findings demonstrate that Knowledge-guided Frequency Reformer, Frequency-similar Constraint, and Frequency-invariant Augmentation are all indispensable for improving the representation power and generalization ability of FAT.

4.3.2 Sensitivity to the Choice of Frequency Reformers. We now evaluate the design choices of the Knowledge-guided Frequency Reformer, which adaptively transforms time-series with diverse frequency expressions into a unified frequency space by mapping patterns stored in a parameterized knowledge memory. Specifically, we consider three variants:

• $FAT_{Random-k}$: The frequency reformer retains k random frequency components as a baseline, where k is set to 50% of the total frequency components.

• FAT_{Top-k}: The frequency reformer retains the top k frequency components with the highest amplitude.

• FAT_{Shared}: The frequency reformer learns a shared transformation parameter across all samples, as in Eq. 14.

As shown in Table 4, FAT consistently achieves the best performance, followed by FAT_{Shared}, with $FAT_{Random-k}$ performing better than FAT_{Top-k}. This result underscores the necessity of learning an adaptive frequency transformation rather than relying on fixed selection rules.

Furthermore, the performance gap between FAT and FAT_{Shared} highlights the importance of learning a data-dependent frequency reforming process. Since time-series sequences with similar semantics can exhibit diverse frequency representations, a static transformation fails to align them effectively, limiting generalization. By contrast, FAT adaptively aligns these variations into a unified frequency expression, ensuring representation consistency while preserving meaningful frequency information.

4.3.3 Sensitivity to the Capacity of Knowledge-guided Frequency Reformers. Results in previous section demonstrate that capturing frequency information requires an adaptive, data-driven approach rather than relying on fixed selection rules. However, the optimal number of Frequency Reforming Patterns (P) in the Knowledge-guided Frequency Reformer remains uncertain.

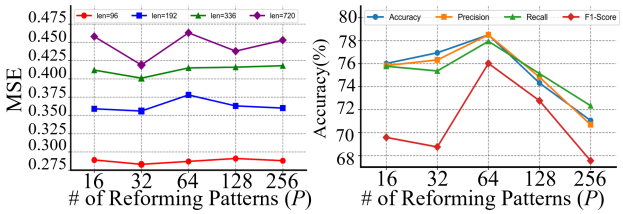

Figure 4: Left: Ablation results (MSE) on the number of rules (P) in $\Theta^{\mathcal{F}}_{km}$, evaluated in the in-domain setting (ETTH2). Right: Ablation results (Accuracy, Precision, Recall, and F1 Score) on the number of rules (P) in $\Theta^{\mathcal{F}}_{km}$, evaluated in the cross-domain setting (SleepEEG → GESTURE).

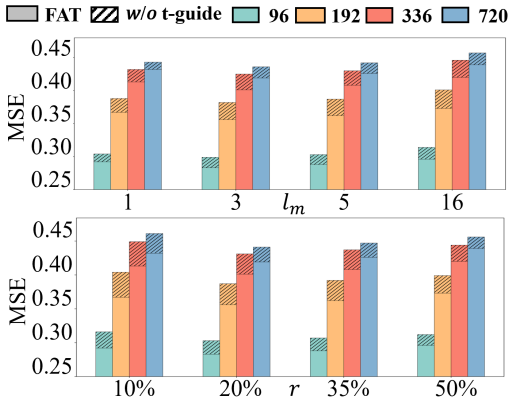

Figure 5: Impact of augmentation hyper-parameters, including masked segment length (l_m) and distortion ratio (r), under settings with and without augmented samples in reconstruction. The experiment was conducted on ETTH2.

To investigate this, we vary the number of stored patterns (8 to 128) under in-domain (ETTH2) and cross-domain (SleepEEG → GESTURE) settings, analyzing their impact on tasks with stable vs. shifting frequency distributions.

As in Figure 4, results follows a peak-then-decline trend in both settings, indicating that while frequency adaptation enhances alignment, excessive patterns introduce redundancy or noise. Notably, the optimal number differs between settings—32 patterns perform best in-domain, whereas 64 patterns yield the highest performance in cross-domain adaptation.

The higher requirement in cross-domain settings suggests that domain shifts introduce additional challenges beyond frequency misalignment, necessitating broader adaptation capabilities. These findings highlight the need for dynamically adjusting the number of adaptation patterns based on task complexity to enhance both robustness and generalization.

4.3.4 Sensitivity to Augmentation Strategy. Hyper-parameters in Frequency-invariant Augmentation, including masked segment length (l_m) and distortion ratio (r), control the difficulty of the pretraining task. Excessive distortions can make the task overly difficult, hindering the learning process, while insufficient distortion leads to trivial replication of adjacent values, failing to capture meaningful temporal patterns.

To examine this trade-off, we vary the difficulty of the pretraining task and analyze its impact. As shown in Figure 5, initially, increasing the distortions ratio improves learning by encouraging the encoder to capture essential patterns. However, when $l_m > 3$ or $r > 50\%$, excessive distortions removes too much information, making reconstruction overly difficult and degrading representation quality.

One of the key motivations for proposing Frequency-invariant Augmentation is to provide useful guidance for reconstructing long sequences while preventing the model from exploiting trivial cues. To further investigate whether augmentation aids pretraining, we replace augmented samples with solely masked original sequences (i.e., w/o *aug* in Section 4.3.1). We observe that under more challenging settings, the model struggles to effectively acquire knowledge from the pretraining task.

These results highlight that a moderate reconstruction difficulty optimally balances learning efficiency and generalization. Furthermore, Frequency-invariant Augmentation extends this threshold, allowing the encoder to learn more complex patterns and better adapt to real-world variations.

5 Conclusion

In this paper, we introduced FAT, a novel self-supervised pretraining framework for time-series representation learning. FAT seamlessly integrates conventional pretraining paradigms with frequency-aware learning, comprising three key innovations: the Knowledge-guided Frequency Reformer, the frequency-similarity constraint, and Frequency-invariant Augmentations. By enabling the encoder to directly extract consistent and generalizable frequency patterns from time-domain signals, FAT eliminates the need for architectural modifications or additional modules during inference, making it adaptable to various backbone models.

Comprehensive experiments across 14 benchmark datasets demonstrate that FAT consistently outperforms existing pretraining frameworks in both classification and regression tasks, exhibiting strong generalization across different architectures and experimental settings. Ablation studies further validate the effectiveness of our proposed modules in jointly capturing essential features across both time and frequency domains. Beyond its empirical success, FAT highlights the potential of learning a unified representation for semantically similar time-series with diverse frequency expressions, without requiring human labeling. This presents a promising step toward a truly universal sequence representation learning architecture.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (NSFC) (62072379, 72401235, 72303183), the Major Program of National Fund of Philosophy and Social Science of China (19ZDA074), the Natural Science Foundation of Sichuan Province (2024NSFSC1061, 2025ZNSFSC0041), Financial Innovation Center of Southwestern University of Finance and Economics (FIC2023E007), and the Key Laboratory of Financial Intelligence and Financial Engineering of Sichuan Province.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning (ICML)*.

[2] Rui Cheng and Qing Li. 2021. Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks. In *Proceedings of the AAAI Conference on artificial intelligence (AAAI)*.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American of the Association for Computational Linguistics (NAACL)*.

[4] Jiaxiang Dong, Haixu Wu, Yuxuan Wang, Yunzhong Qiu, Li Zhang, Jianmin Wang, and Mingsheng Long. 2024. Timesiam: A pre-training framework for siamese time-series modeling. *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

[5] Jiaxiang Dong, Haixu Wu, Yuxuan Wang, Yun-Zhong Qiu, Li Zhang, Jianmin Wang, and Mingsheng Long. 2024. TimeSiam: A Pre-Training Framework for Siamese Time-Series Modeling. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[6] Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. 2024. Simmtm: A simple pre-training framework for masked time-series modeling. *Proceedings of the neural information processing systems (NeurIPS)*.

[7] Yogesh Dwivedi and Suhasini Subba Rao. 2011. A test for second-order stationarity of a time series based on the discrete Fourier transform. *Journal of Time Series Analysis* (2011).

[8] Florian Eilers and Xiaoyi Jiang. 2023. Building blocks for a complex-valued transformer architecture. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[9] Wei Fan, Shun Zheng, Xiaohan Yi, Wei Cao, Yanjie Fu, Jiang Bian, and Tie-Yan Liu. 2022. DEPTS: Deep Expansion Learning for Periodic Time Series Forecasting. In *International Conference on Learning Representations (ICML)*.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

[11] Geoffrey E Hinton. 1989. Connectionist learning procedures. *Artificial Intelligence* 40, 1-3 (1989).

[12] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data mining and knowledge discovery* 33, 4 (2019), 917–963.

[13] Junguang Jiang, Yang Shu, Jianmin Wang, and Mingsheng Long. 2022. Transferability in deep learning: A survey. *arXiv preprint arXiv:2201.05867* (2022).

[14] Shruti Kaushik, Abhinav Choudhury, Pankaj Kumar Sheron, Nataraj Dasgupta, Sayee Natarajan, Larry A Pickett, and Varun Dutt. 2020. AI in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data* 3 (2020), 4.

[15] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

[16] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, Yoshua Bengio and Yann LeCun (Eds.).

[17] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The ACM SIGIR conference on research & development in information retrieval (SIGIR)*.

[18] Ruiqi Li, Maowei Jiang, Quangao Liu, Kai Wang, Kaiduo Feng, Yue Sun, and Xiufang Zhou. 2025. Faith: Frequency-domain attention in two horizons for time series forecasting. *Knowledge-Based Systems* 309 (2025), 112790.

[19] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Proceedings of the neural information processing systems (NeurIPS)*.

[20] Zhe Li, Zhongwen Rao, Lujia Pan, Pengyun Wang, and Zenglin Xu. 2023. Ti-mae: Self-supervised masked time series autoencoders. *arXiv preprint arXiv:2301.08871* (2023).

[21] Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. 2024. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining(SIGKDD)*.

[22] Jiexi Liu and Songcan Chen. 2024. Timesurl: Self-supervised contrastive learning for universal time series representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

[23] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[24] Dalton Lunga, Saurabh Prasad, Melba M Crawford, and Okan Ersoy. 2013. Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning. *IEEE Signal Processing Magazine* 31, 1 (2013), 55–66.

[25] Dongsheng Luo, Wei Cheng, Yingheng Wang, Dongkuan Xu, Jingchao Ni, Wenchao Yu, Xuchao Zhang, Yanchi Liu, Yuncong Chen, Haifeng Chen, et al. 2023. Time series contrastive learning with information-aware augmentations. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

[26] Murugappan Murugappan, Subbulakshmi Murugappan, and Bong Siao Zheng. 2013. Frequency band analysis of electrocardiogram (ECG) signals for human emotional state classification using discrete wavelet transform (DWT). *Journal of physical therapy science* 25, 7 (2013), 753–759.

[27] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*.

[28] Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. 2024. Fredformer: Frequency Debiased Transformer for Time Series Forecasting. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

[29] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[30] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*.

[31] Hao Wang, Licheng Pan, Zhichao Chen, Degui Yang, Sen Zhang, Yifei Yang, Xinggao Liu, Haoxuan Li, and Dacheng Tao. 2024. FreDF: Learning to Forecast in Frequency Domain. *arXiv preprint arXiv:2402.02399* (2024).

[32] Zhiyuan Wang, Xovee Xu, Weifeng Zhang, Goce Trajcevski, Ting Zhong, and Fan Zhou. 2022. Learning latent seasonal-trend representations for time series forecasting. *Proceedings of the Neural Information Processing Systems (NeurIPS)*.

[33] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. 2022. CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[34] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Proceedings of the neural information processing systems (NeurIPS)*.

[35] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 3733–3742.

[36] Sihan Xu, Ziqiao Ma, Yidong Huang, Honglak Lee, and Joyce Chai. 2023. CycleNet: rethinking cycle consistency in text-guided diffusion for image manipulation. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*.

[37] Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. 2024. FilterNet: Harnessing Frequency Filters for Time Series Forecasting. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*.

[38] Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. 2024. Frequency-domain MLPs are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems (NeurIPS)*.

[39] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. 2022. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

[40] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the ACM SIGKDD conference on knowledge discovery data mining (SIGKDD)*.

[41] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. 2022. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems (NeurIPS)*.

[42] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*.

[43] Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. 2022. Film: Frequency improved legendre memory model for long-term time series forecasting. *Proceedings of the Neural Information Processing Systems (NeurIPS)*.

[44] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the International Conference on Machine Learning (ICML)*.

A Dataset Description

Table 5: Dataset descriptions. *Samples* are organized in (Train/Validation/Test).

Tasks	Datasets	Channels	Series Length	Samples	Classes	Information	Frequency
Forecasting	ETTH1,ETTH2	7	{96,192,336,720}	8,545/2,881/2,881	-	Electricity	Hourly
	ETTM1,ETTM2	7	{96,192,336,720}	34,465/11,521/11,521	-	Electricity	15 Mins
	Weather	21	{96,192,336,720}	36,792/5,271/10,540	-	Weather	10 Mins
	Exchange	8	{96,192,336,720}	5,120/665/1,422	-	Exchange rate	Daily
	Electricity	321	{96,192,336,720}	18,317/2,633/5261	-	Electricity	Hourly
	Traffic	862	{96,192,336,720}	12,185/1,757/3,509	-	Transportation	Hourly
Classification	SleepEEG	-	200	371055	5	EEG	100 Hz
	Gesture	-	178	320	8	Gesture	100 Hz
	FD-B	-	5120	60	3	FD	64K Hz
	EMG	-	1500	122	3	EMG	4K Hz
	EPI	-	178	60	2	EPI	178 Hz
	128 UCR	-	15~2,844	16~8,926	2~60	-	-

B Hyperparamter Setting

We have followed and compared the official implementations of all baseline models with our approach. To ensure a fair comparison, we have maintained the original configurations outlined in these papers. Unless otherwise specified, the key hyperparameters are shown as follows:

Table 6: The main hyperparameter settings.

	Hyperparameter	Value
Backbone	*Hidden Size*	[16,32,64,128,256]
	Number of Transformer Layers	[2,3,4]
	Feed-Forward Layer Dimension	[64,128,256]
	Number of Attention Heads	[4,8,16,32]
Pretrain and Finetune	*Optimizer*	Adam
	Dropout Rate	{0.1,0.5}
	Batch Size	[8,16,32,64]
	Learning Rate	0.0001,0.01
	Number of Rules (P)	[8,16,32,64,128]
	Mean Masking Length	[3,5,7,11,15,20]
	Temperature	0.2
	Number of Positive Samples	3
	Masking Ratio	{0.1,0.5}

C Full Results of Classification

Table 7: The full results of baseline models on 5 classification datasets.

Datasets	Methods	Accuracy(%)	Precision(%)	Recall(%)	F1 Score(%)	Mean(%)
Gesture	Rand. Init.	32.50	32.50	21.56	25.92	28.12
	PatchTST	63.47	63.47	60.53	61.97	62.36
	TST	62.91	62.91	60.00	61.42	61.81
	TS2Vec	63.33	63.33	60.40	61.83	62.22
	LAST	65.14	65.14	62.12	63.59	64.00
	TF-C	57.50	57.50	54.75	56.09	56.46
	SimMTM	74.17	74.17	71.68	72.90	73.23
	InfoTS	69.52	69.52	66.31	67.88	68.31
	TimesURL	69.03	69.03	65.86	67.41	67.83
	Timesiam	75.02	75.02	71.56	73.25	73.71
	FAT	**77.74**	**77.74**	**77.16**	**77.45**	**77.52**
FD-B	Rand. Init.	31.39	45.10	30.68	36.52	39.95
	PatchTST	46.19	53.85	44.22	48.56	48.21
	TST	45.62	53.43	43.61	48.02	47.67
	TS2Vec	43.59	51.91	38.79	44.40	44.67
	LAST	42.64	51.24	40.43	45.20	44.88
	TF-C	45.53	53.36	43.51	47.93	47.58
	SimMTM	60.74	69.98	57.96	63.41	63.02
	InfoTS	61.14	64.79	55.11	59.56	60.15
	TimesURL	59.93	63.90	56.84	60.16	60.21
	Timesiam	64.47	67.22	60.67	63.78	64.03
	FAT	**69.98**	**77.98**	**70.32**	**73.95**	**73.06**
EMG	Rand. Init.	46.34	33.33	55.45	41.63	44.19
	PatchTST	82.88	71.96	88.18	79.25	80.57
	TST	81.86	71.07	87.09	78.27	79.57
	TS2Vec	89.68	84.71	95.45	89.76	89.9
	LAST	76.50	66.42	81.40	73.15	74.37
	TF-C	78.05	68.44	74.49	71.34	73.08
	SimMTM	85.37	74.12	90.83	81.63	82.99
	InfoTS	84.87	73.70	90.31	81.16	82.51
	TimesURL	83.51	72.50	88.85	79.85	81.18
	Timesiam	84.71	73.53	90.11	80.98	82.33
	FAT	**90.61**	**89.34**	**89.74**	**89.54**	**89.81**
EPI	Rand. Init.	39.79	47.72	29.89	36.76	38.54
	PatchTST	77.59	77.09	80.08	78.56	78.33
	TST	76.64	76.14	79.09	77.59	77.36
	TS2Vec	78.91	78.40	81.44	79.89	79.66
	LAST	71.63	71.16	73.92	72.51	72.31
	TF-C	67.56	66.14	80.49	72.61	71.70
	SimMTM	95.13	94.33	95.05	94.69	94.80
	InfoTS	91.33	90.77	94.30	92.50	92.23
	TimesURL	90.49	89.87	93.35	91.58	91.32
	Timesiam	94.03	93.39	97.02	95.17	94.90
	FAT	**96.07**	**93.60**	**93.77**	**93.69**	**94.28**
128 UCR	Rand. Init.	49.03	45.07	36.93	40.60	42.91
	PatchTST	60.85	57.70	62.50	60.00	60.26
	TST	60.10	56.99	61.73	59.27	59.52
	TS2Vec	72.50	69.31	71.15	70.22	70.79
	LAST	56.17	53.26	57.69	55.39	55.63
	TF-C	61.88	58.68	63.56	61.02	61.29
	SimMTM	75.34	73.37	74.96	74.16	74.46
	InfoTS	84.11	79.77	86.41	82.96	83.31
	TimesURL	84.53	80.17	86.84	83.37	83.73
	Timesiam	76.56	72.61	78.65	75.51	75.83
	FAT	**86.97**	**85.21**	**87.44**	**86.31**	**86.48**