A WAVE IS WORTH 100 WORDS: INVESTIGATING CROSS-DOMAIN TRANSFERABILITY IN TIME SERIES

A PREPRINT

Xiangkai Ma[*], **Xiaobin Hong**[*], **Wenzhong Li**[✉], **Sanglu Lu**
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
{xiangkai.ma,xiaobinhong}@smail.nju.edu.cn
{sanglu,lwz}@nju.edu.cn

December 3, 2024

ABSTRACT

Time series analysis is a fundamental data mining task that has made encouraging progress in many real-world scenarios. Supervised training methods based on empirical risk minimization have proven their effectiveness on specific tasks and datasets. However, the acquisition of well-annotated data is costly and a large amount of unlabeled series data is under-utilized. Due to distributional shifts across various domains and different patterns of interest across multiple tasks. The problem of cross-domain multi-task migration of time series remains a significant challenge. To address these problems, this paper proposes a novel cross-domain pretraining method based on Wave Quantization (termed as WQ4TS), which can be combined with any advanced time series model and applied to multiple downstream tasks. Specifically, we transfer the time series data from different domains into a common spectral latent space, and enable the model to learn the temporal pattern knowledge of different domains directly from the common space and utilize it for the inference of downstream tasks, thereby mitigating the challenge of heterogeneous cross-domains migration. The establishment of spectral latent space brings at least three benefits, cross-domain migration capability thus adapting to zero- and few-shot scenarios without relying on priori knowledge of the dataset, general compatible cross-domain migration framework without changing the existing model structure, and robust modeling capability thus achieving SOTA results in multiple downstream tasks. To demonstrate the effectiveness of the proposed approach, we conduct extensive experiments including three important tasks: forecasting, imputation, and classification. And three common real-world data scenarios are simulated: full-data, few-shot, and zero-shot. The proposed WQ4TS achieves the best performance on **87.5%** of all tasks, and the average improvement of the metrics on all the tasks is up to **34.7%**.

1 Introduction

Time series analysis has broad real-world applications, including imputation of missing data Karmitsa et al. [2022], power consumption detection for production equipment Wang et al. [2022b], and weather forecasting Schultz et al. [2021]. Traditional end-to-end deep learning models have made significant progress in time series analysis Zhou et al. [2023], Wang et al. [2022a], Zhou et al. [2022b], Seyfi et al. [2022], Li et al. [2022], Jeon et al. [2022], with many popular deep neural network architectures being applied to time series modeling, such as Linear-based Zeng et al. [2023], Yi et al. [2023b], CNN-based Wu et al. [2023], Wang et al. [2023], RNN-based Shi et al. [2015], Transformer-based Nie et al. [2023], Zhou et al. [2022a, 2021], Liu et al. [2022], and GNN-based models Wu et al. [2020]. In addition, work on signal processing before the backbone based on the numerical characterization of the series has driven the development of time series forecasting, such as Seasonal-Trend Decomposition Zhou et al. [2022a], Wu et al. [2021], Wang et al. [2023], which improves the efficiency of backbone in representation extraction by capturing the complex patterns from original series in advance. Recently, we have witnessed the remarkable success of pre-trained foundation

[*]Equal Contribution

models Radford and Narasimhan [2018], Radford et al. [2019], Brown et al. [2020], Touvron et al. [2023a,b] in Natural Language Processing (NLP), Computer Vision (CV), and Multimedia (MM). Large-scale Language Models (LLM) in particular have demonstrated impressive performance in various areas, thus many works try to build a large time series model on top of LLMs Zhou et al. [2023], Cao et al. [2023], Li et al. [2023], Xue and D.Salim [2022], Chang et al. [2023], Sun et al. [2023], Jin et al. [2023], Liu et al. [2023a].

However, series with different domain information tend to be heterogeneous and their performance deteriorates rapidly when domain shifts occurs, making it difficult to deploy either specialized models trained for specific tasks and datasets Cai et al. [2021], He et al. [2023], Ragab et al. [2022], Jin et al. [2022], Xu et al. [2022], in complex and variable real-world scenarios. When the test set distribution and the training data are not the same, the models often fail to exhibit satisfactory inferential capability. Consider the following case, in the absence of sufficient training data, we would like the models to be pre-trained on heterogeneous datasets first, followed by a small amount of data fine-tuning in the target domain, or even inference directly on the target domain task. To overcome these challenges, there has been work looking to utilize limited data in the target domain to enable the model to adapt to the new target domain, which is referred to as the cross-domain adaptation technique.

For real-world time series analysis tasks, an inference model with strong domain adaptation capabilities is crucial. Recently, more attention has been paid to Time Series Cross Domain Migration Jin et al. [2022], Ragab et al. [2022], He et al. [2023], Cai et al. [2021], Ragab et al. [2021]. however, existing approaches either require additional domain expertise or a unique design of the model structure to accommodate the challenges of domain migration, which makes it difficult to directly apply the existing domain adaptation research to SOTA models designed for specific tasks, which makes it challenging to deploy domain adaptive models in the real world. deployment in the real world remains challenging. Moreover, considering the impressive achievements of previous research for specific time series analysis tasks, we would like to design a general and compatible cross-domain migration framework that enables existing models to overcome the challenge of data distribution drift without changing the model structure and without relying on priori knowledge of the dataset. This is the starting point of this paper, i.e., **"a generic and compatible cross-domain migration framework"**.

Some recent studies Finder et al. [2022], Fang et al. [2023], Zhang et al. [2023], Guo et al. [2023] applying spectral analysis techniques to time series have prompted us to think about the fact that time series in spectral space can be uniformly characterized by a set of filters with continuous frequencies, as opposed to the high degree of perturbation in the time domain space that prevails in the pattern of sequence changes, which encourages us to establish a shared spectral space and embed the time series data from different domain domains into this shared space, thus mitigating the time series domain drift phenomenon between data, and enabling existing models to overcome the challenge of data distribution migration drift without changing the model structure and without relying on priori knowledge of the dataset.

Table 1: To verify the scalability of the existing models on different sampling rates (rate=10), we design two groups of supervised experiments which training on the *Original* dataset (ETTh1) and the *Resample* dataset (ETTh1), respectively, and show the MSE Loss calculated on the same test set.

Model	OneFitsAll	DLinear	PatchTST	FEDformer	TimesNet
Original Supervised	0.352	0.357	0.351	0.448	0.400
Resample Supervised	0.757	0.722	0.758	0.739	1.149
MSE ↑ (%)	115.1	102.2	116.0	65.0	187.3

In time-domain space, the prevalent distributional drift in time series makes cross-domain migration more challenging, and existing studies tend to mitigate the heterogeneity between time series data from different domains by establishing trend-seasonal decomposition Wu et al. [2021], Zhou et al. [2022a], Wang et al. [2023] or multiscale decomposition Shabani et al. [2022]. However, the challenges of cross-domain migration are mainly caused by the differences between data domains: (1) Firstly, time series data with diverse domain background knowledge tend to exhibit different characteristics, e.g., electricity consumption generally exhibits long-term trends, ECG signals have stable horizontal baselines, and oscillate frequently, and establishing multi-domain connectivity and complex sequence patterns among different datasets is difficult. (2) Secondly, even in the same domain background, datasets from different sources may exhibit heterogeneous in their time scales, sampling intervals, periodic patterns, etc., which results in the same data instances can also face the challenge of cross-domain migration due to a priori features such as sampling rate, etc., e.g., Table 1 shows the experimental results. (3) Thirdly, unlike natural language processed by human brain abstraction as a set of ordered values continuously sampled from the real world, the time series itself is a primitive and low-level data form that does not have any universally applicable inherent pattern, encoding the fluctuating changes into rich semantic representation is non-trivial.

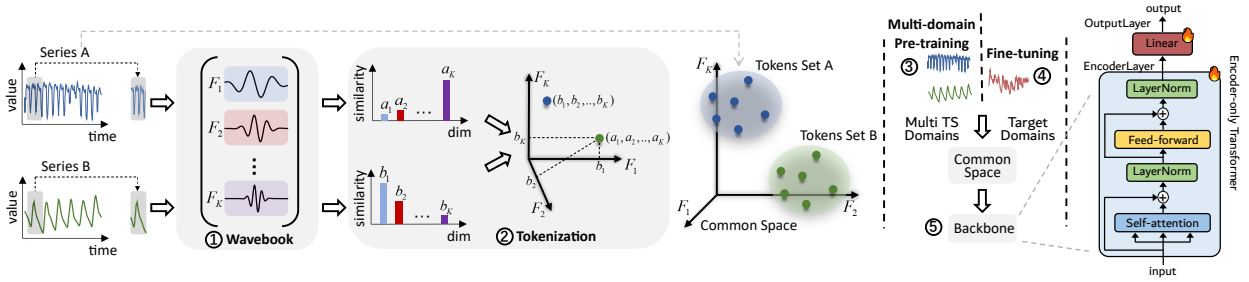

Figure 1: Illustration of the proposed WQ4TS architecture. ① The proposed **Wave Quantize Module** was utilized to establish the common spectral latent space. ② Based on proposed **Tokenization Strategy**, the raw series is bijectively projected to the common spectral latent space. The distance between multiple TS domains is effectively reduced, which activates the generalization and migration capabilities of the model. Meanwhile, the generated tokens as fluctuation pattern similarities contain sufficient semantic information ③ Subsequently, the TS pattern knowledge in common spectral latent space will be utilized for multiple downstream tasks, by **Cross-domain Pre-training**. ④ Due to the Effective feature programming of the embedding technique, we design full parameters **Fine-tuning for Domain migration**. ⑤ We design the encoder-only transformer structure as the backbone, which contains a stack of EncoderLayers and OutputLayer.

To address these challenges, we propose a novel framework to build a pre-trained model based on multi-domain time series data, which is illustrated in Fig. 1. Firstly, we construct the *wavebook* that consists of a set of orthogonal basis functions to form a common latent space from multiple domains. Based on the wavebook, a *tokenization* mechanism bijectively maps the input time series from diverse domains into tokens in the common spectral latent space following the "wave as token" principle, which mitigates the challenge of heterogeneous multiple domains. Subsequently, the comprehensive time series pattern knowledge of different domains will be learned by a backbone (i.e., a vanilla Transformer encoder) from the shared common embedding space with a *multi-domain pre-training* process. The latent representation was formed by calculating the inner product between the original series and each basis function of the wavebook, which implies semantic information of fluctuation pattern similarity of tokens. we can prove that the proposed wavebook and tokenization strategy is interpretable and each token has semantic information, as detailed in the section 3. For different TS domains, the above computations have no specific requirements for hyperparameters, thus addressing challenge of heterogeneous. Therefore, each local segment corresponds to the vector of length λ, where λ is the size of the wavebook. Since the token contains λ item fluctuation pattern similarity as semantic information, thus addresses challenge of information pattern.

Since all tokens come from the unified common space, which alleviates the negative migration phenomenon and thus ensures that the model has a considerable advantage in the cross-domain migration phase. In addition, the proposed WQ4TS reduces the representation distance between the target and source domains by projecting the raw series from the target domain into the common space, and the model also has the advantage of "in-modality", which encouraged us to design full parameters fine-tuning for domain adaptation.

The key contributions of our work are as follows.

- This paper presents the concept of the Wave Quantize in time series for the first time, and analyzes the reasons for the slow progress of unsupervised cross-domain migration research in time series. Finally, it analyzes the differences between TS and NLP in terms of data paradigms, which serve as foundational work on establishing general compatible cross-domain migration framework and effective tokenization strategies.
- In this paper, we propose a novel **Wave Quantize** Strategy that advances research related to ***cross-domain migration*** and common spectral latent space without changing the existing model structure and without relying on priori knowledge of the dataset.
- To verify that wave quantize strategy can mitigate negative migration in time series, the experiments consisted of single-domain training and multi-domain pre-training, and showed encouraging results: multi-domain pre-training would be far superior to single-domain pre-training.
- We performed 3 tasks (forecasting, imputation, and classification) under 3 data-limited scenarios (full data, few-shot, and zero-shot) are experimented to validate the effectiveness of the proposed approach. Specifically, for the forecasting and imputation tasks, extensive experiments on seven mainstream datasets; for the classification task, 35 most representative datasets were chosen from the UCR dataset. In forecasting and imputation tasks,

compared to the existing SOTA, the proposed WQ4TS achieves the best performance on **87.5%** of all tasks, moreover, WQ4TS demonstrates **25.8%** and **44.1%** performance improvement in few-shot and zero-shot settings, respectively. In the few-shot classification task based on cross-domain migration, the proposed WQ4TS improves the average accuracy by **24.9%** in all seven scenarios. These experiments demonstrate the potential of WQ4TS to be the general compatible cross-domain migration framework in time series.

The subsequent structure of this paper is as follows: Section 2 summarizes the related work about the tokenization strategy, spectrum analysis and cross-domain migration on TS. Section 3 details the proposed wavebook and "wave as token" strategy. Section 4 shows the model architecture. Section 5 shows the performance of forecasting, imputation, and classification tasks under three data-limited scenarios: full-data, few-shot, and zero-shot. Finally, experimental results and future research are discussed in Section 6.

2 Related Work

We introduce the related works in terms of LLM for TS, attempts for the unified pre-trained Model, tokenization strategy in TS, and spectrum analysis in TS.

2.1 Cross Domain Migration in TS

Deep neural networks trained on one domain can be poor at generalizing to another domain due to the issue of domain shift, that is, domain shift problem Wang et al. [2020c], Zhao et al. [2020], Zhang et al. [2020], Oza et al. [2021]. Domain adaptation (DA) methods attempt to mitigate the harmful effect of domain shift by aligning features extracted across source and target domains Hong et al. [2020], Wang et al. [2020b]. Existing approaches mainly focus on classification tasks, where a classifier learns a mapping from a learned domain-invariant latent space to a fixed label space using source data. Consequently, the classifier depends only on common features across domains, and can be applied to the target domain Wilson and Cook [2018]. Unlike DA, This paper is devoted to design a novel framework to establish cross-domain connectivity between different TS datasets, enabling the knowledge learned by the backbone network from different complex sequence patterns to be migrated to multiple downstream tasks, which is known as Unsupervised Cross-domain Migration (UCM).

Recently, lots of research efforts Oza et al. [2021], Vibashan et al. [2021] are devoted on unsupervised domain adaptation. This task aims to transfer knowledge learned from a labeled source domain to a different unlabeled target domain, and most approaches focus on aligning distributions of source and target domain and learning domain-invariant feature representations. There are mainly two levels for UDA methods: domain-level Bousmalis et al. [2017], Tzeng et al. [2017] and category-level Kang et al. [2019], Du et al. [2021], Li et al. [2021]. Domain-level UDA mitigates the distribution divergence between the source and target domain by pulling them into the same distribution at different scale levels. Besides, some works Du et al. [2021], Li et al. [2021] focus on the fine-grained category-level label distribution alignment through an adversarial manner between the feature extractor and two domain-specific classifiers. Finally, One prevailing paradigm aims at minimizing statistical distribution measures to mitigate the distribution shift problem between the source and target domains Chen et al. [2019], Wang et al. [2020a], Wu and Ng [2022].

In light of successes in related fields, domain adaptation techniques have been introduced to time series tasks Jin et al. [2022], Ragab et al. [2022], He et al. [2023], Cai et al. [2021], Ragab et al. [2021]. To generate accurate input pairs, CDTrans Xu et al. [2022] designs a two-way center-aware labeling algorithm to produce pseudo labels for target samples. Along with the pseudo labels, a weight-sharing triple-branch transformer framework is proposed to apply self-attention and cross-attention for source/target feature learning and source-target domain alignment, respectively. Recent approach Jin et al. [2022] proposes a shared-attention model with domain-adaptive capabilities that predicts future sequences using local representations over different time periods by extracting domain-invariant features and modeling domain-relevant attributes in conjunction with domain-specific features in order to appropriately approximate the data distribution of the respective domain. AdaTime Ragab et al. [2022] develops a bench marking evaluation suite to systematically and fairly evaluate different domain adaptation methods on time series data. Specifically, we standardize the backbone neural network architectures and bench marking datasets, while also exploring more realistic model selection approaches that can work with no labeled data or just a few labeled samples. RainCoat He et al. [2023] as a model for analyzing both closed-set and generalized-set data for complex time series, addresses feature and label shifts by considering both temporal and frequency features, aligning them across domains, and correcting for misalignments to facilitate the detection of private labels. Existing model Cai et al. [2021] designs intra- and inter-variable sparse attention mechanisms to extract correlation-structured time-series data considering time lags and utilize correlation-structured alignment to guide the transfer of knowledge from the source domain to the target domain. SLARDA Ragab et al. [2021] designs a self-supervised learning module that utilizes forecasting as an auxiliary task to improve the transferability of the source features, and proposes a novel autoregressive domain adaptation technique

that incorporates temporal dependency of both source and target features during domain alignment. However, current DA frameworks are often only oriented to a single downstream task, such as classification DA Xu et al. [2022], He et al. [2023] and prediction DA Jin et al. [2022]. A general DA framework for a wide range of downstream tasks is encouraged to generally improve the cross-domain transfer ability of a wide range of SOTA models. Besides, unlike existing methods that require designing shared/shared feature types for different domains and choosing appropriate architectures for time series forecasting models, our approach provides the first end-to-end Unsupervised Crossdomain Migration (UCM) generalized framework for multiple downstream tasks.

2.2 Tokenization strategy in TS

Motivated by the successful application of transformers in NLP Vaswani et al. [2017] and pre-trained foundation models Radford and Narasimhan [2018], Brown et al. [2020], Touvron et al. [2023a,b], transformer-based models are viewed as equally promising in time series. Due to the characteristics of time series, initial approaches Godfried et al. [2020], Kitaev et al. [2020a], Zhou et al. [2021], Liu et al. [2021], Wu et al. [2021], Woo et al. [2022b], Zhou et al. [2022a] generally followed the ***Point as Token*** design, that is each sampled time point acts as a separate token. There are two limitations of this strategy: firstly, computing global dependencies between any time points leads to high computational complexity; secondly, there is a serious information redundancy in the global dependencies captured by the attention mechanism, considering that the time series is continuously varying, which leads to two neighboring tokens having nearly the same numerical distribution. Therefore, approaches at the initial stage of time series analysis have focused on reducing computational complexity by mitigating information redundancy.

For instance, Reformer Kitaev et al. [2020b] designed the locally sensitive hashing self-attention to reduce computational complexity. Informer Zhou et al. [2021] proposes the ProbSparse self-attention mechanism to efficiently replace the canonical self-attention. Pyraformer Liu et al. [2021] introduces the pyramidal attention module which reduces computational complexity by constraining the maximum length of the signal traversing paths. Autoformer Wu et al. [2021] designs the Auto-Correlation mechanism based on the series periodicity, which conducts the dependencies discovery and representation aggregation at the sub-series level. ETSformer Woo et al. [2022b] proposes the novel exponential smoothing attention and frequency attention to replace the self-attention mechanism in vanilla Transformers, thus improving both accuracy and efficiency. FEDformer Zhou et al. [2022a] avoids the high overhead of computation on the time domain by calculating the dependencies between individual bands in the frequency domain, while Fourier Transform has been used to ensure that individual bands have a global view.

Subsequently, the representative PatchTST Nie et al. [2023] proposed a ***Patch as Token*** strategy and channel-independent design to demonstrate the effectiveness of the transformer in time series. Based on this, the recent iTransformer Liu et al. [2023b] proposes the ***Series as Token*** strategy, that is the whole series is regarded as the token, and the attention mechanism is used to capture the dependencies between the sequences of different channels, which is surprisingly intuitive and effective in data domains with a very large number of channels and complex dependency relationships between them. The above studies have shown that the reasonable tokenization strategy can drastically improve the performance of Transformer-based models compared to the complex and tedious model structure improvement. However, the existing tokenization strategy cannot be well applied to cross-domain migration. Since TS datasets from different domains and sampling settings exhibit diverse periodic patterns, it is difficult to identify a unified sub-series span that matches all TS data domains. existing tokenization strategies are unable to establish a unified framework to adapt to the potential TS data domains, and the pre-training phase is unable to adequately learn cross-domain representational information, which limits the generalization ability and scalability of the model. Therefore, The ideal tokenization should be insensitive to the mathematical characteristics of different data domains, which motivates us to propose ***Wave as Token*** as the general compatible cross-domain migration strategy.

2.3 LLM for TS

The adaptation of pre-trained LLMS for time series analysis has attracted attention, exploiting their superior capabilities in sequence representation learning. OneFitsAll Zhou et al. [2023] first attempted to apply the pre-trained GPT2 Radford et al. [2019] to the time series downstream tasks, and achieved comparable performance to the state-of-the-art methods by freezing the pre-trained self-attention and feed forward structures to maximize the retention of pre-training information, and only fine-tuning the layer norm. In contrast, the recent TimeLLM Jin et al. [2023] completely freezes the model parameters of the pre-trained Llama Touvron et al. [2023a], aligning both time series and natural language modalities by introducing textual prototypes. The task settings and dataset priori information are fed into the pre-trained Llama as hard-prompt by the pre-trained Embedder, thus fully activating the inference capability of the foundation model on the time series forecasting task. TEMPO Cao et al. [2023] designs a prompt pool based on seasonal-trend decomposition to generate specific prompts for each sub-component, and incorporates LoRA Hu et al. [2022] to achieve efficient fine-tuning of LLM. TEST Sun et al. [2023] establishes a TS embedding method applicable to LLMs by using

orthogonal text embedding vectors as prototypes to constrain the TS embedding space, thus activating the feature extraction capability of LLMs in the time series data domain. LLM4TS Chang et al. [2023] proposes a two-stage fine-tuning strategy, which firstly enables supervised fine-tuning of LLMs in the time series modality, and subsequently suggests downstream fine-tuning of LLMs in specific tasks, which unleash the flexibility of pre-trained LLMs.

However, existing approaches have not yet realized and attempted to address the challenge of negative migration in cross-domain migration, where UniTime Liu et al. [2023a] has proposed domain instructions to ensure the model recognizes the differences between multiple data domains, yet the problem of negative migration Hu et al. [2019] still inevitably arises. These difficulties motivate us to propose a novel tokenization strategy to establish potential connections between multiple data domains, thus enabling cross-domain migration capability with the original transformer architecture.

2.4 Spectrum Analysis in TS

Over the past two decades, spectrum analysis techniques based on Fast Fourier Transform (FFT) and Wavelet Transform (WT) has been widely used in diverse model structures for time series analysis Finder et al. [2022], Fang et al. [2023], Zhang et al. [2023], Guo et al. [2023] to improve the performance of learning directly from the time domain. Specifically, transformer-based models use spectral analysis to reduce the complexity of self-attention mechanisms, e.g., Autoformer Wu et al. [2021] captures the periodic pattern information from original series based on FFT and establishes the Auto-Correlation mechanism at the sub-series level for learning dependencies and representation aggregation. FEDformer Zhou et al. [2022a] generates a set of mixed frequency components by Fourier analysis and designs a frequency enhanced attention mechanism, which exploited the sparse representation of the spectrogram and achieved linear complexity. Besides, the MLP-based FreTS Yi et al. [2023b] captures a complete global view of the original signal by operating on the spectral components obtained by frequency-transformation, and overcomes the information bottleneck of MLP on time series by ensuring that MLP focuses on the key part frequency components. In addition, the FourierGNN Yi et al. [2023a] proposes a novel architecture that uniformly captures inter-series (spatial) dynamics and intra-series (temporal) dependencies by performing matrix multiplication in Fourier space. Finally, researchers in representation learning have noted the significance of establishing consistency constraints in the temporal-frequency space. For example, TF-C Zhang et al. [2022b] designs a self-supervised pre-training strategy based on time-frequency consistency, that is temporal- and frequency- representations learned from the same TS samples should be closer in the temporal-frequency space than different TS samples. In addition, CoST Woo et al. [2022a] establishes a more effective connection by mapping the embedded features in the time domain space to the frequency domain. Based on the advantages of descriptive spectrum analysis, this paper designs the Wave Quantize tokenization strategy.

3 Methodology

Establishing the generic and compatible cross-domain migration framework between different time series domains has the following challenges: (1) time series data with diverse domain background knowledge tend to exhibit different characteristics; (2) Datasets from different sources may exhibit variations, even in the same domain background; (3) The same data instances can also face the challenge of cross-domain migration due to a priori features such as sampling rate.

The proposed ***Wave Quantize*** Module solves challenge-1 through the designed common spectral latent space. In addition, the designed strategy ensures that arbitrary series data can be embedded to equal-length groups of tokens (number of tokens equal to series length), thus solving challenge-2. This ensures that no hyperparameter setting tricks are utilized in the model to adapt to potential data domains, even if these domains are completely different from each other in terms of structural features (e.g., channel number, sequence length) are completely different. Finally, we solve challenge-3 by ensuring that each token contains TS pattern information within a localized window through the finite-length basis functions and bridges the difference in the distribution of pattern information from different domains within the λ-dimensional space formed by basis functions, where λ is the size of the proposed wavebook.

Specifically, the basic idea is illustrated in Fig. 1. Motivated by VQVAE van den Oord et al. [2017], we design a fine-grained tokenization strategy to standardize the original series before the backbone, project input series from diverse domains into a common latent space, and generate a set of tokens. Subsequently, the TS pattern knowledge of different domains will be learned directly by backbone from the common spectral latent space, and used for multiple downstream tasks. Specifically, a set of orthogonal basis functions is designed to form the latent space, where each basis function $\{A_i|1 \leq i \leq \lambda\}$ is a finite-length waveform with attenuation, thus ensuring that the energies of the basis functions are confined to a local window. We define this set of orthogonal basis functions as the *"Wavebook"* in Fig. 1①. We perform the inner product operation between the original series and basis function by continuously sliding the window with the step equal to 1, thus calculating the fluctuation pattern similarity between the basis function and each segment of the input series in the local window. Repeating for all λ basis functions so that each segment of the local

window corresponds to a feature vector of length λ. In addition, considering the continuity of the basis functions, when the window length is set as small as possible, each time point is embedded to a vector of length λ, which contains the pattern information in λ dimensions for the local window in which the time point is located, as shown in Fig. 1②. Moreover, when a fixed generating function is chosen and a set of basis functions with different frequencies is obtained by resampling the generating function continuously. Then, the vector will represent the energy distribution of the corresponding time point on λ frequency bands. This makes the proposed tokenization strategy interpretable and each token has semantic information.

3.1 Framework Overview

We propose a framework called WQ4TS to build a cross-domain migration time series general model, which is illustrated in Fig. 1. It consists of four parts: wave quantize module, time series tokenization, cross-domain pre-training, and fine-tuning for domain migration. Wave quantize module construction extracts a set of orthogonal basis functions from multiple TS domains to form the common spectral latent space. Based on the wavebook, a tokenization mechanism projects the input time series from diverse domains into tokens in the common spectral latent space to mitigate their heterogeneity. Subsequently, in the cross-domain pre-training phase, the comprehensive time series pattern knowledge of different domains is extracted from the tokens to train a backbone Transformer encoder to form a time series unified framework.

To guarantee that the model can simultaneously learn latent representations from diverse TS data domains with different statistical features and temporal pattern, we adopt two promising designs: 1) Encoder-only design. Since TS data often exhibit the complexity of multiple patterns superimposed Wang et al. [2023], Cao et al. [2023], we adopt an encoder-only design with excellent generalization capabilities for representation learning, which has been proven competent by the SOTA Transformer-based models Nie et al. [2023], Liu et al. [2023b]. 2) channel independent. The multivariate time series sample with n channels was regarded as n separate univariate series, which is utilized as the exemplary case to simplify the methodology and was shown effective in the literature Zhou et al. [2023], Jin et al. [2023]. Besides, the specific description of the important symbols involved in the method is shown in Table 2.

The overview of the proposed WQ4TS architecture is illustrated in Fig. 1. The time series from multiple TS domains is represented as $X = (x_1, ..., x_l) \in \mathbb{R}^l$ with l time steps, and WQ4TS is utlized to predict future series $Y = (x_{l+1}, ..., x_{l+c}) \in \mathbb{R}^c$ with c time steps. The common space V^λ will be formed by the proposed λ-dimensional **Wave Quantize Module**, which comprises a set of orthogonal basis functions. Subsequently, in **Time Series Tokenization**, according to proposition 2, any sub-series from the original series can be transformed into a group of coordinates in V^λ, and the bijection relation is satisfied between the sub-series and the coordinates. Furthermore, proposition 3 gives the sufficient-necessary condition and construction method for orthogonal wavelet bases. In **Cross-domain Pre-training** phase, the sub-series at the timestep-j are converted to $token_j = (p_{1,j}, p_{2,j}, ..., p_{\lambda,j}) \in \mathbb{R}^\lambda$, from diverse TS domains. Subsequently, simple full-parameter **Fine-tuning for Domain migration** is designed since the common spectral latent space reduces the distance between the source and target domains.

3.2 Wave Quantize Module

This section presents the theoretical foundations of the wave quantize module, which is the fine-grained tokenization strategy that makes the establishment of the general compatible cross-domain migration framework possible. Among it, proposition 2 theoretically proves the bijective projection relation established by the wavebook, and proposition 3 proposes the sufficient-necessary condition for the construction of the wavebook. According to these propositions, we construct the wavebook and eventually form the common spectral latent space. For all propositions described in the current section, the thorough proof procedure and the background of the wavebook will be provided.

We introduce the following notations: $F(t)$ is the specific basis function, $H(\omega)$ and $G(\omega)$ refer as the low-pass filter and band-pass filter of $F(t)$, respectively.

Definition 1. *We define the wavebook as a set consisting of several basis functions, as* $\{F_{j,k}(t) = 2^{j/2}F(2^j t - k), (j,k) \in \mathbb{Z}^2\}$*, which constitutes a set of standard orthonormal basis (O.N.B) and forms the finite series space* $L^2(\mathbb{R})$ *if and only if* $F(t)$ *is the orthogonal wavelet.*

Definition 2. *Based on the proposed wavebook, there must exist the unique coefficients sequence* $\{c_{j,k}; (j,k) \in \mathbb{Z}^2\} \in l^2(\mathbb{Z})$*. For* $\forall f(t) \in L^2(\mathbb{R})$*, we have* $f(t) = \sum_{j\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} c_{j,k} \cdot F_{j,k}(t)$*, where* $f(t)$ *denotes the univariate time series and* $\{c_{j,k}\}$ *contains the coordinates of* $f(t)$ *in space* $\mathbb{Z}^2$*. Specifically, the bijection is satisfied between variables* $L^2(\mathbb{R})$ *and* $l^2(\mathbb{Z})$ *in two spaces* $f(t)$ *and* $\{c_{j,k}\}$*.*

Table 2: The specific description of the symbols involved in the method. (Due to space constraints, only the symbols appearing in Eqs.(11)-(15) are shown in this table, considering that symbols in the section 3.2 are not necessary for understanding the framework process.)

Symbol formula	Definition
$X \in \mathbb{R}^l$	the history series, where l as the lookback length
$Y, \bar{Y} \in \mathbb{R}^c$	the future and prediction series, where c is the forecast length
m	the precision of the discrete sequence describing the information in A
$A \in \mathbb{R}^{2^m}$	the amplitude sequence, which discretely inscribe the orthogonal wavelet
f_c	the central frequency of the orthogonal wavelet
λ	the size of the wavebook
$S_i \in \mathbb{R}, i \in [1, .., \lambda]$	the set of scale factors utilized to generate the wavebook
$W_i \in \mathbb{R}^{m \cdot s_i}$	the downsampling coordinate
$A_i = A[W_i] \in \mathbb{R}^{m \cdot s_i}$	the basis function in wavebook
V^λ	the shared embedding space, where λ is the size of wavebook
$p_{i,j}$	the pattern similarity between the segment of X at timestep$-j$ and A_i
λ, l	the dimension and number of embedded token vector
L	the number of stacked of EncoderLayers in Encoder
$token_j \in \mathbb{R}^\lambda$	the projection coordinates in the λ-dimensional common space V^λ
$T_{pos} \in \mathbb{R}^{\lambda \times l}$	the learnable position encoding
$T^0, T^L \in \mathbb{R}^{\lambda \times l}$	time series tokens that utilized as the input and output to the Encoder
d_k	the dimension of latent space
$T_{pos} \in \mathbb{R}^{\lambda \times l}$	the learnable position encoding tensor
$W^Q, W^K \in \mathbb{R}^{\lambda \times d_k}, W^V \in \mathbb{R}^{\lambda \times \lambda}$	the trainable Linear Layer of self-attention
$Q_L, K_L \in \mathbb{R}^{l \times d_k}, V_L \in \mathbb{R}^{l \times \lambda}$	Query, Key and Value latent-variable

Proof of Proposition 2. Based on the Definition 1, we have $f(t) = \sum_{j\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} c_{j,k} \cdot F_{j,k}(t)$, where $\left\{F_{j,k}(t), (j,k) \in \mathbb{Z}^2\right\}$ constitutes a standard orthonormal basis for $L^2(\mathbb{R})$, and $\{c_{j,k}\}$ represents coefficients. Besides, since $\{F_{j,k}(t), (j,k) \in \mathbb{Z}^2\}$ constitutes a set of orthogonal bases satisfying $\forall (j_1, k_1) \neq (j_2, k_2)$, there has $\langle F_{j_1,k_1}, F_{j_2,k_2} \rangle = 0$. Further, we have

$$\begin{aligned} \langle f(t), F_{m,n}(t) \rangle &= \left\langle \sum_{j\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} c_{j,k} \cdot F_{j,k}(t), F_{m,n}(t) \right\rangle, \\ &= c_{m,n} \cdot \langle F_{m,n}(t), F_{m,n}(t) \rangle, \\ &= c_{m,n} \cdot |F_{m,n}(t)|^2 . \end{aligned} \tag{1}$$

Thus, the coefficients sequence can be uniquely determined by $c_{m,n} = \langle f(t), F_{m,n}(t) \rangle \,/\, |F_{m,n}(t)|^2$, and the bijection relation is satisfied between the variables $f(t)$ and $\{h_n; n \in \mathbb{Z}\}$, which completes the proof of the Proposition 2. □

Definition 3. *Let* $M(\omega) = \begin{pmatrix} H(\omega) & G(\omega) \\ H(\omega+\pi) & G(\omega+\pi) \end{pmatrix}$*, and the matrix* $M^H(\omega)$ *is the conjugate transpose matrix of* $M(\omega)$*. The sufficient-necessary condition for* $F(t)$ *as the orthogonal wavelet is that* $M(\omega)$ *is the Unitary Matrix:* $M^H(\omega)M(\omega) = M(\omega)M^H(\omega) = I, a.e. \omega \in \mathbb{R}$.

Proof of Proposition 3. Firstly, we introduce and briefly prove the Lemma on the sufficiently-necessary condition for orthonormal system: Defining the function $F(t) \in L^2(\mathbb{R})$, then the set $\{F_{0,n} = 2^{0/2} F(2^0 t - n) = F(t-n)\}$ forms the orthonormal system of $L^2(\mathbb{R})$, that is $\langle F(t-n), F(t-l) \rangle = \delta(n-l)$ is the sufficiently-necessary for

$\sum_{k\in\mathbb{Z}}\left|\hat{F}(\omega+2k\pi)\right|^2=1$, where $\delta(n-l)$ represents $\frac{1}{2\pi}\int_0^{2\pi}e^{-i(n-l)\omega}d\omega$. Lemma is proved due to

$$\begin{aligned}\langle F(t-n),F(t-l)\rangle&=\frac{1}{2\pi}\int_{\mathbb{R}}\hat{F}(\omega)e^{-in\omega}\cdot\overline{\left(\hat{F}(\omega)e^{-il\omega}\right)}d\omega,\\&=\frac{1}{2\pi}\int_0^{2\pi}\sum_{k\in\mathbb{Z}}\left|\hat{F}(\omega+2k\pi)\right|^2e^{-i(n-l)\omega}d\omega.\end{aligned}\tag{2}$$

In the Shannon Sampling Theorem, for any signal $f(t)$ defined on $L^2(\mathbb{R})$, if the frequency domain form $\hat{f}(\omega)$ of that signal has a truncation frequency B, then that signal can be reconstructed by equally spaced discrete sampling. This sampling interval can be at most π/B. If the function $f(t)$ satisfies the following conditions $\forall f(t)\in L^2(\mathbb{R}),\hat{f}(\omega)=\int_{-\infty}^{+\infty}f(t)e^{-i\omega t}dt,|\omega|>B$, we have:

$$f(t)=\sum_{n\in\mathbb{Z}}f(n\Delta)\frac{sin(\pi/\Delta)(t-n\Delta)}{(\pi/\Delta)(t-n\Delta)},0<\Delta\leq\frac{\pi}{B}.\tag{3}$$

Considering first the case $B=\pi$, and the scale function is defined as $\phi(t)=\frac{\sin(\pi t)}{\pi t}$, we have $f(t)=\sum_{n\in\mathbb{Z}}f(n)\phi(t-n)$. Further, defining the space $V_j=\{f(t);\hat{f}(\omega)=0,|\omega|>2^j\pi\}$ with truncation frequency $B=2^j\pi$ and taking the sampling interval $\Delta=2^{-j}$, we have $f(t)=\sum_{n\in\mathbb{Z}}2^{-j/2}f(2^{-j}n)\phi_{j,n}(t)$, where the scale function $\phi_{j,n}$ as:

$$\phi_{j,n}(t)=2^{j/2}\phi(2^jt-n)=\frac{sin\pi(2^jt-n)}{\pi(2^jt-n)}.\tag{4}$$

Based on the wavelet function $F_{j,k}$, we define the close-span space $W_j=closespan\{F_{j,k}(t)=2^{j/2}F(2^jt-k),(j,k)\in\mathbb{Z}^2\}$, and the close-span space has the following characteristic: 1) Spatial orthogonality $W_j\perp W_{j+1}$; 2) Spatial approximability $L^2(\mathbb{R})=\bigoplus_{j=-\infty}^{+\infty}W_j$. if $F(t-k)$ is the set of the orthonormal basis of W_0, then for any j there has the orthonormal basis $\{2^{j/2}F(2^jt-k),(j,k)\in\mathbb{Z}^2\}$ of W_j. Moreover, it is easy to verify: $W_j\perp V_j,V_{j+1}=W_j\oplus V_j$, the construction of orthogonal wavelets is equivalent to finding a set of standard orthogonal bases for W_0.

Since the scale function $\phi(t)\subseteq V_1$, and there exists a set of orthonormal basis $\{\sqrt{2}\phi(2t-n);n\in\mathbb{Z}\}$ for V_1, there must exist a unique sequence of coefficients $\{h_n;n\in\mathbb{Z}\}$ such that $\phi(t)=\sqrt{2}\sum_{n\in\mathbb{Z}}h_n\phi(2t-n)$, which is regarded as the scale equation, and since $\phi(2t-n)$ is mutually orthogonal to each other for different n, the coefficients are calculated as follows $h_n=\left\langle\phi(t),\sqrt{2}\phi(2t-n)\right\rangle=\sqrt{2}\int_R\phi(t)\overline{\phi}(2t-n)dt$.

In addition, the scale equation are converted to frequency domain form by Fourier transforms

$$\hat{\phi}(\omega)=H(\omega/2)\hat{\phi}(\omega/2),\quad H(\omega)=\frac{1}{\sqrt{2}}\sum_{n\in\mathbb{Z}}h_ne^{-in\omega},\tag{5}$$

where $H(\omega)$ is referred to as the low-pass filter and hence h_n is also referred to as the low-pass filter coefficients.

For the wavelet function $F(t)\subseteq V_1$, there exists $\{g_n;n\in\mathbb{Z}\}$ such that $F(t)=\sqrt{2}\sum_{n\in\mathbb{Z}}g_n\phi(2t-n)$, which is regarded as the wavelet equation, and since $\phi(2t-n)$ is orthogonal for different n, the coefficients are calculated as $g_n=\left\langle F(t),\sqrt{2}\phi(2t-n)\right\rangle=\sqrt{2}\int_RF(t)\overline{\phi}(2t-n)dt$.

In addition, the wavelet equations can be obtained in frequency domain form by Fourier transformation

$$\hat{F(\omega)}=G(\omega/2)\hat{\phi}(\omega/2),\quad G(\omega)=\frac{1}{\sqrt{2}}\sum_{n\in\mathbb{Z}}g_ne^{-in\omega},\tag{6}$$

where $G(\omega)$ is referred to as the bandpass filter and g_n is also referred to as the impulse response coefficient.

We define the functions $H(\omega)$ and $G(\omega)$ refer as the low-pass filter and band-pass filter based on $F(t)\in L^2(\mathbb{R})$, which determined by (5) and (6), respectively, and introduce the matrix $M(\omega)$.

The function group $\{F_{0,k}=2^{0/2}F(2^0t-k)=F(t-k),k\in\mathbb{Z}\}$ forms the orthonormal basis of W_0, that is, the sufficient-necessary condition for $F(t)$ as the orthogonal wavelet is that $M(\omega)$ is the Unitary Matrix: $M^H(\omega)M(\omega)=M(\omega)M^H(\omega)=I,a.e.\omega\in\mathbb{R}$, where $M^H(\omega)$ is defined to be the conjugate transpose matrix of $M(\omega)$.

By the definition of the Unitary Matrix, $M(\omega)$ is the Unitary Matrix equivalent to

$$\begin{aligned} |H(\omega)|^2 + |H(\omega+\pi)|^2 &= 1, a.e.\omega \in \mathbb{R}, \\ |G(\omega)|^2 + |G(\omega+\pi)|^2 &= 1, a.e.\omega \in \mathbb{R}, \\ H(\omega)\overline{G}(\omega) + H(\omega+\pi)\overline{G}(\omega+\pi) &= 0, a.e.\omega \in \mathbb{R}. \end{aligned} \tag{7}$$

We define the spaces V_0 and W_0 as follows

$$\begin{aligned} V_0 &= closespan\left\{\phi(t-n), n \in \mathbb{Z}\right\}, \\ W_0 &= closespan\left\{F(t-n), n \in \mathbb{Z}\right\}. \end{aligned} \tag{8}$$

By the definitions of $H(\omega)$ and $G(\omega)$, Equation 7 is equal to

$$\begin{aligned} \sum_{k\in\mathbb{Z}} \left|\hat{\phi}(\omega+2k\pi)\right|^2 &= 1, a.e.\omega \in \mathbb{R}, \\ \sum_{k\in\mathbb{Z}} \left|\hat{F}(\omega+2k\pi)\right|^2 &= 1, a.e.\omega \in \mathbb{R}, \\ V_0 \perp W_0, a.e.\omega &\in \mathbb{R}. \end{aligned} \tag{9}$$

Because of Lemma, the first two conditions are equivalent to

$$\begin{aligned} \langle\phi(t-n), \phi(t-l)\rangle &= \delta(n-l), \\ \langle F(t-n), F(t-l)\rangle &= \delta(n-l). \end{aligned} \tag{10}$$

In summary, the sufficient-necessary condition for $M(\omega)$ as the Unitary Matrix is that $\phi(t-n)$ and $F(t-n)$ form the standard orthogonal system of $L^2(\mathbb{R})$, respectively, and the two spaces formed by $\phi(t-n)$ and $F(t-n)$ are orthogonal. Considering the properties of V_j and W_j, $\{F_{j,k}(t) = 2^{j/2}F(2^j t-k), (j,k) \in \mathbb{Z}^2\}$ constitutes the orthonormal basis for $L^2(\mathbb{R})$, and hence $F(t)$ is an orthogonal wavelet.

According to Proposition 3, by designing a specific functional relationship between $H(\omega)$ and $G(\omega)$, we can guarantee that $M(\omega)$ is the Unitary Matrix, and thus that $F(t)$ is the orthogonal wavelet. For instance, when $G(\omega) = e^{-i\omega}\overline{H}(\omega+\pi)$, it is easy to verify that $M(\omega)$ is the Unitary Matrix, when the frequency domain form of the wavelet function $F(t)$ is that $\hat{F}(\omega) = e^{-i\omega/2}\overline{H}(\pi+\omega/2)\cdot\hat{\phi}(\omega/2)$, where the corresponding impulse response coefficient is $g_n = (-1)^{n-1}\overline{h}_{1-n}, n \in \mathbb{Z}$. Therefore, the time domain form of the wavelet function is $F(t) = \sqrt{2}\sum_{n\in\mathbb{Z}}(-1)^{n-1}\overline{h}_{1-n}\phi(2t-n)$. □

3.3 Time Series Tokenization

Tokenizing multiple time series domains and establishing their cross-domain connectivities is a challenging task. We propose a time series tokenization approach to map the input time series from diverse domains into tokens in the common spectral latent space following the "wave as token" principle, which is described as follows.

Based on proposition 3, the amplitude sequence $A \in \mathbb{R}^{2^m}$ is utilized to discretely inscribe the orthogonal wavelet. Specifically, A defines the amplitude values of 2^m points that contain information about the waveforms of the specified orthogonal wavelet function, where m serves as the precision of the discrete sequence describing the information in A, and the spacing between any two neighboring points is denoted as $step$ which equals to $m/(2^m-1)$

We denote the central frequency of the orthogonal wavelet as f_c and the size of the wavebook as λ. The set of scale factors is calculated by:

$$S = \left\{S_i = (2\cdot f_c\cdot\lambda)/i, i \in [1,2,\ldots,\lambda]\right\}, \tag{11}$$

where $(m\cdot S_i) \in \mathbb{Z}$. For each scale factor S_i, the downsampling coordinate W_i and basis function $A_i \in \mathbb{R}^{m\cdot S_i}$ in wavebook can be sampled by:

$$\begin{aligned} W_i &= \left\{w_{i,j} = \frac{(2^m-1)\cdot j}{m\cdot s_i}, j \in [1,...,m\cdot s_i]\right\}, \\ A_i &= A[W_i] = A[w_{i,1},...,w_{i,m\cdot s_i}] \in \mathbb{R}^{m\cdot s_i}, \end{aligned} \tag{12}$$

where each A_i contains waveform information of the specified basis function, which is obtained by the scaling transform on the orthogonal wavelet with the scale factor S_i. For convenience, we subsequently refer to A as the orthogonal wavelet and A_i as the basis function that makes up the wavebook.

For the time series $X = (x_1, ..., x_l)$ of length l and the basis function $A_i = (a_{i,1}, ..., a_{i,n})$, where $n = m \cdot s_i$ and we define the following operation:

$$\begin{aligned} Convolve(X, A_i) &= (c_{i,1}, c_{i,2}, ..., c_{i,l+n-1}), \\ Difference(X, A_i) &= (d_{i,1}, d_{i,2}, ..., d_{i,l+n-2}), \\ Recode_{A_i}(X) &= (p_{i,1}, p_{i,2}, ..., p_{i,l}), \end{aligned} \tag{13}$$

where, $p_{i,j}$, $d_{i,j}$ and $c_{i,j}$ are calculated sequentially based on a given set of x_k and $a_{i,j}$ as follows:

$$\begin{aligned} c_{i,j} &= \sum_{k=1}^{j} x_k \cdot a_{i,j+1-k}, \quad \Delta a_{i,j} = a_{i,j+1} - a_{i,j}, \\ d_{i,j} &= -\sqrt{s_i} \cdot (c_{i,j+1} - c_{i,j}) = -\sqrt{s_i} \cdot \sum_{k=1}^{j+1} x_k \Delta a_{i,j+1-k}, \\ p_{i,j} &= d_{i,j+\frac{n}{2}-1} = -\sqrt{s_i} \cdot \sum_{k=1}^{j+\frac{n}{2}} x_k \Delta a_{i,j+\frac{n}{2}-k}. \end{aligned} \tag{14}$$

Considering the property defined by the basis function A_i, that is the fluctuation of A_i is concentrated in a finite region (assumed to be the intermediate region). In Eq. 13, when $k \to j$, there is $\left|a_{i,j+(n/2)-k}\right|, \left|a_{i,j+(n/2)-k-1}\right| \to \left|a_{i,(n/2)}\right|$, and $|a_{i,1}|, |a_{i,n}| \to 0$. Therefore, $p_{i,j}$ is regarded as the fluctuation pattern similarity between the segment of time series X at timestep-j and the basis function A_i. Iterating over all basis functions $\{A_1, A_2, ..., A_\lambda\}$, the set of pattern similarity $token_j = (p_{1,j}, p_{2,j}, ..., p_{\lambda,j})$ can be calculated for any timestep among time series.

Further, the set of orthogonal basis functions can be utilized to form the λ-dimensional embedding space V^λ. With the $token$ described above, the neighborhood segment at timestep-j can be projected as the separate point in V^λ, and $token_j$=$(p_{1,j}, p_{2,j}, ..., p_{\lambda,j}) \in \mathbb{R}^\lambda$ is regarded as the projection coordinates of the segment in the λ-dimensional common space V^λ, where the length of the segment is adapted by the attenuation of the fluctuation distribution of the basis functions. Besides, each timestep of the original series X is traversed, and all the projected coordinates are concatenated together as T^0=$(token_1, token_2, ..., token_l)$+$T_{pos}$, where $T^0, T_{pos} \in \mathbb{R}^{\lambda \times l}$ and the learnable position encoding T_{pos} indicates the temporal order between sub-series.

3.4 Cross-domain Pre-training

We describe the fine-grained cross-domain pre-training strategy as follows. In each epoch, variables from multiple domains are randomly shuffled with the batch to ensure that the unified general cross-domain model can simultaneously exhibit satisfactory generalization ability across diverse time series domains. Specifically, the backbone encoder accepts the tokens from diverse time series domains as input and learns the dependencies between segments via the attention mechanism. Besides, each token vector serves as the fluctuation pattern similarity, thus containing sufficient semantic information. Notably, since all tokens come from the unified common embedding space, it alleviates the heterogeneity of TS multi-domains. These advantages ensure that the proposed model is advantageous in multi-domain pre-training phase.

To fully stimulate the inference capability of WQ4TS in downstream tasks, we design forecasting and classification as multi-tasks for different downstream tasks. Besides, in the cross-domain pre-training phase, a set of adaptive dynamic weights was designed to balance the gradients from different time series domains, considering differences in generalization across time series domains and the dataset size. Specifically, we define the aggregate loss function as $Loss = \sum_{i \in \mathbb{Z}} (\alpha_i \cdot Loss_i)$, where $Loss_i$ is defined as the loss value of specific $domain_i$, and multi-task weightings $\{\alpha_i; i \in \mathbb{Z}\}$ are automatically calculated by the learnable strategy Kendall et al. [2018] which considering the homoscedastic uncertainty of each task.

3.5 Cross-domain Migration

To validate the domain migration capability, WQ4TS is pre-trained in multiple TS source domains, it contains two natural advantages: (1) the knowledge learned by the model in the pre-training phase has the same modality information

as the target domain; (2) the proposed wave quantize module uniformly represents the source and target domains in the common space. These advantages encourage us to adopt the simple fine-tuning strategy of **allowing all parameters to fine-tune in the target domain**. The TS samples in the source and target domain are directly projected to the common spectral latent space formed by the wave quantize module. Then it achieves alignment between target and source domains at the representation level and stimulates the cross-domain migration capability of the downstream backbone. Specifically, WQ4TS is fine-tuned on partial samples of the target domain, finally predicted on the target domain, in the few-shot classification task, as shown in the table 10; This implementation ensures that all performance improvements come from the proposed **Wave Quantize** module, and there is still a huge room for improvement in WQ4TS , based on the existing Parameter-Efficient Fine-Tuning approaches Mangrulkar et al. [2022].

4 Architecture of WQ4TS

The architecture of WQ4TS consists of a **Tokenization** operation (Section. 3.3), an encoder-only Transformer, and an **OutputLayer**, as illustrated in Fig. 1. Specifically, the OutputLayer adaptively selects the model architecture according to different downstream tasks. For example, for forecasting and impuation, the OutputLayer only contains a single linear layer, for classification task, OutputLayer consists of linear layer and Softmax. The backbone network is a Transformer encoder consisting of stack of L **EncoderLayers**. The overall architecture is as follows:

$$\begin{aligned} T^0 &= Tokenization(X), \\ T^{k+1} &= EncoderLayer(T^k), k = 0, ..., L-1, \\ \overline{Y} &= OutputLayer(T^L). \end{aligned} \tag{15}$$

EncoderLayer Based on section 3.3, the time series tokens $T^k \in \mathbb{R}^{\lambda \times l}$ are utilized as the input to the multi-head attention mechanism, where T^k contains l λ-dimensional embedded token vectors.

Specifically, the dependencies between the generated tokens in the common spectral latent space V^λ can be calculated by the equation $\left(\hat{T}^k\right)^\top = Softmax\left(Q_L \cdot (K_L)^\top / \sqrt{d_k}\right) \cdot V_L$, where $Q_L, K_L, V_L = \left(T^k\right)^\top \cdot \left[W^Q, W^K, W^V\right]$ are served as query, key, and value latent-variable in attention.

Concretely, $W^Q, W^K \in \mathbb{R}^{\lambda \times d_k}$ and $W^V \in \mathbb{R}^{\lambda \times \lambda}$ denote the trainable linear layer, which projects the $token_j$ into the d_k-dimensional latent space. Finally, $\hat{T}^k \in \mathbb{R}^{\lambda \times l}$ is the output of the attention mechanism. Besides, each **EncoderLayer** is also composed of the feed-forward network and layer normalization with residual connections as shown in Fig. 1, and generates the latent representation $T^{k+1} \in \mathbb{R}^{\lambda \times l}$ as the input for the next **EncoderLayer**.

OutputLayer The output of the last EncoderLayer is essentially the set of tokens. To make it possible to match the predicted time series format, tokens are first processed by flattening to obtain a 1D series, which is subsequently projected to a specified prediction length via the linear layer. Specifically, the **OutputLayer** contains two components, **Flatten**: $\mathbb{R}^{\lambda \times l} \mapsto \mathbb{R}^{\lambda l}$ and **Linear**: $\mathbb{R}^{\lambda l} \mapsto \mathbb{R}^c$, where the output of the OutputLayer is represented as $\overline{Y} \in \mathbb{R}^c$. The detailed process of the master training stage is in Algorithm 1.

5 Experiments

In this section, we first introduce the benchmark and baseline which will be utilized in the subsequent experiments, followed by three tasks in each of the three subsections, where each task consists of three different settings: 1) **Full-data**: models are trained and predicted on the target domain; 2) **Few-shot**: In forecasting and imputation tasks, models are trained and predicted on the target domain, where the dataset is only partially available in the training phase; In the classification task, models are first pre-trained on the source domain, subsequently, are fine-tuned on partial samples of target domain, finally predicted on the target domain; These designs were utilizaed to demonstrate the data efficiency and cross-domain adaptability of the proposed model; 3) **Zero-shot**: models are predicted in the target domain directly after pre-training in the single or multiple source domain;

5.1 Configurations

We provide the experiment configuration in Table 3. All experiments are repeated three times, implemented in PyTorch and conducted on a single Tesla V100 SXM2 32GB GPU. Our method is trained with the L2 Loss, using the ADAM optimizer with an initial learning rate of 10^{-4}, and Batch size is set in $16 \rightarrow 64$. The training process is early stopped after three epochs (patience=3) if there is no loss degradation on the valid set. The mean square error (MSE) and mean

Algorithm 1: Master Training Stage of WQ4TS

Input: Lookback series $X = (x_1, ..., x_l) \in \mathbb{R}^l$.
Output: Forecasting series $Y = (x_{l+1}, ..., x_{l+c}) \in \mathbb{R}^c$.

Based on the sufficient-necessary condition introduced in Proposition 3, the orthogonal wavelet F(t) is first designed to adapte the characteristic of the data domain;
Subsequently, the amplitude sequence $A \in \mathbb{R}^{2^m}$ is utilized to discretely inscribe the orthogonal wavelet;
Finally, a set A_i will be sampled from A, where each A_i contains waveform information of the specified basis function, which is obtained by the scaling transform on the orthogonal wavelet with the scale factor S_i, as shown in (11)-(12);
Randomly initialize the parameter set θ_1;
for *iteration = 1,2,3, ...* **do**
    Based on the design of the set $\{A_1, ..., A_\lambda\}$, each timestep of the original series X is traversed by Proposition 2, and all the projected coordinates are concatenated together as T^0=$(token_1, ..., token_l)$+T_{pos}, where $token_j$ contains the fluctuation pattern similarity between time series X and basis function $\{A_i\}$, by (13)-(14);
    WQ4TS uses the Encoder obtained by stacking L EncoderLayers as the model backbone. Subsequently the time series tokens $T^0 \in \mathbb{R}^{\lambda \times l}$ are utilized as the input to the multi-head attention mechanism of EncoderLayer, where λ and l indicate the dimension and number of embedded token vectors, as follows (15);
    The output $T^L \in \mathbb{R}^{\lambda \times l}$ of the last EncoderLayer is essentially a set of tokens that are first processed by flattening to obtain a 1D series, and then projected to a specified prediction length via the Linear Layer. Ultimately, the output of the output layer is represented as $\overline{Y} \in \mathbb{R}^c$;
    Get the forecasting loss $\mathcal{L}_{mse}$ between prediction $\overline{Y}$ and ground truth Y, subsequently update parameters θ_1 according to the gradient of $\mathcal{L}_{mse}$;
return θ_1

Table 3: Experiment configuration of WQ4TS .

Tasks	Model Hyper-parameter				Training Process			
	Layers	d_{min} †	d_{max} †	λ°	LR*	Loss	Batch Size	Epochs
Forecasting	10	32	512	100	10^{-4}	MSE	32	10
Imputation	10	64	128	100	10^{-4}	MSE	32	10
Classification	5	64	128	100	10^{-3}	MSE	64	30

† $d_{\text{model}} = \min\{\max\{2^{\lceil \log C \rceil}, d_{\text{min}}\}, d_{\text{max}}\}$, where C is input series dimension.
∗ LR means the initial learning rate.
∘ λ means the dimension of shared embedding space, that is the size of the wavebook.

absolute error (MAE) are used as metrics in forecasting and imputation tasks. Besides, the accuracy (Acc), precision (Pre), recall (Rec), and f1-score (F1) are used as metrics in the classification task. By default, the proposed **WQ4TS** contains $5 \rightarrow 10$ **EncoderLayers**. All the baselines that we reproduced are implemented based on configurations of the original paper or their official code. For a fair comparison, we design the same input embedding and final prediction layer for all base models.

5.2 Benchmarks

To evaluate the performance of the proposed method, we extensively experiment with the mainstream time series analysis tasks including long-term forecasting, imputation (i.e., predicting the missing data in a time series), and classification. The long-term forecasting, imputation and classification are evaluated with several popular real-world datasets, including: **ETT (ETTh1, ETTh2, ETTm1, and ETTm2)**[2] Zhou et al. [2021] contains six power load features and oil temperature used for monitoring electricity transformers. ETT involves four subsets. ETTm1 and ETTm2 are recorded at 15-minute intervals, while ETTh1 and ETTh2 are recorded hourly. **Exchange**[3] Lai et al. [2017] records daily exchange rates of eight different countries ranging from 1990 to 2016. **Weather**[4] contains 21 meteorological indicators, such as temperature, humidity, and precipitation, which are recorded every 10 minutes in the year 2020.

[2]https://github.com/zhouhaoyi/Informer2020
[3]https://github.com/laiguokun/multivariate-time-series-data
[4]https://www.bgc-jena.mpg.de/wetter/

Electricity[5] comprises hourly power consumption of 321 clients from 2012 to 2014. **Traffic**[6] reports the number of vehicles loaded on all 862 roads at each moment in time. **Sunspot**[7] records observations of sunspots for long-term monitoring, consisting of 73924 timesteps. **River Flow**[8] reports th daily river flow, consisting of 23741 timesteps. **Solar Power**[9] contains a single long daily time series representing the wind power production in MW recorded every 4 seconds starting from 2019. **UCR archive**[10] as the well-known time series classification repository, where representative **35 datasets** are selected to validate the performance of the proposed approach.

5.3 Baselines

We compare the proposed WQ4TS model with the well-acknowledged and advanced models, which include the CNN-based Models: **TimesNet** Wu et al. [2023] and **MICN** Wang et al. [2023]; the MLP-based model: **DLinear** Zeng et al. [2023]; the Transformer-based models: **Informer** Zhou et al. [2021], **ETSformer** Woo et al. [2022b], **Stationary** Liu et al. [2022], **Autoformer** Wu et al. [2021], **FEDformer** Zhou et al. [2022a], and **PatchTST** Nie et al. [2023]; and a LLM-empowered model: **OneFitsAll** Zhou et al. [2023]. In the **forecasting** task, to indicate the generalization capability on different prediction scales, we fixed the lookback length as 336, and the prediction lengths including $\{96, 192, 336, 720\}$. In the **imputation** task, to compare the performance under different proportions of missing data, we randomly mask the time points with a ratio of $\{12.5\%, 25\%, 37.5\%, 50\%\}$, and the lookback length is fixed as 96. Besides, the **classification** task is shown only for accuracy and F1 score, while the full-data task on all 35 datasets from UCR, and the few-shot task on multiple scenarios.

5.4 Long term forecasting

Table 4: Comparison of the averaged performance from diverse prediction lengths ($\{96, 192, 336, 720\}$) on **full-data forecasting** task, where ETTh1,h2,m1,m2 are from the same dataset.

Models	**WQ4TS (Ours)**		OneFitsAll 2023		DLinear 2023		PatchTST 2023		TimesNet 2023		FEDformer 2022a		Autoformer 2022		Stationary 2021		ETSformer 2021		Informer 2021	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	0.368	**0.369**	0.352	0.383	0.357	0.378	**0.351**	0.380	0.400	0.406	0.448	0.452	0.588	0.517	0.481	0.456	0.429	0.425	0.961	0.734
ETTm2	**0.247**	**0.295**	0.266	0.326	0.267	0.333	0.255	0.315	0.291	0.333	0.305	0.349	0.327	0.371	0.306	0.347	0.293	0.342	1.410	0.810
ETTh1	**0.407**	**0.418**	0.427	0.426	0.422	0.437	0.413	0.430	0.458	0.450	0.440	0.460	0.496	0.487	0.570	0.537	0.542	0.510	1.040	0.795
ETTh2	0.347	**0.373**	0.354	0.394	0.431	0.446	**0.330**	0.379	0.414	0.427	0.437	0.449	0.450	0.459	0.526	0.516	0.439	0.452	4.431	1.729
Electricity	**0.154**	**0.247**	0.167	0.263	0.166	0.263	0.161	0.252	0.192	0.295	0.214	0.295	0.227	0.327	0.193	0.338	0.208	0.296	0.311	0.397
Traffic	**0.379**	0.286	0.414	0.294	0.433	0.295	0.390	**0.263**	0.620	0.336	0.610	0.376	0.628	0.379	0.624	0.340	0.621	0.396	0.764	0.416
Weather	**0.216**	**0.246**	0.237	0.270	0.248	0.300	0.225	0.264	0.259	0.287	0.309	0.360	0.338	0.382	0.288	0.314	0.271	0.334	0.634	0.548
Sunspot	**0.395**	**0.442**	0.445	0.477	0.526	0.554	0.446	0.476	0.450	0.478	0.477	0.498	0.458	0.488	0.462	0.496	0.481	0.533	0.559	0.604
RiverFlow	**1.004**	**0.497**	1.218	0.551	1.146	0.605	1.233	0.658	1.247	0.651	1.139	0.596	1.246	0.651	1.137	0.574	1.250	0.659	1.312	0.737
SolarPower	**0.031**	**0.063**	0.036	0.072	0.046	0.093	0.043	0.086	0.082	0.149	0.046	0.094	0.093	0.167	0.064	0.120	0.076	0.149	0.078	0.147

Experimental setups: First, to verify that the proposed wave quantize module and tokenization strategy can fully stimulate the learning ability of the transformer-based backbone as an effective feature program, Table 4 and Table 5 show the results of the long-time forecasting task under full-data and zero-shot setting. Subsequently, to validate that the proposed approach can learn key information from multiple time series domains and efficiently migrate it to previously unseen target domains, we designed diverse adaption approaches for zero-shot learning. Specifically, Table 6 shows the performance of the models in the target domain test set directly after pre-training in the single source domain and unified pre-training in multiple source domains, respectively. Finally, to illustrate the superior data efficiency of the model, Table 7 shows the results of the models under the few-shot 5% settings, respectively.

Analysis of results: In the full-data forecasting task shown in Table 4, the proposed WQ4TS exhibits the best performance in **85%** of the metrics. In the zero-shot forecasting task shown in Table 5, the average MSE of the proposed

[5]https://archive.ics.uci.edu/dataset/321/electricity

[6]http://pems.dot.ca.gov

[7]https://www.sidc.be/SILSO/newdataset

[8]http://www.jenvstat.org/v04/i11

[9]https://zenodo.org/records/4656032

[10]https://www.cs.ucr.edu/~eamonn/time_series_data_2018

Table 5: Comparison of the averaged performance from diverse prediction lengths ($\{96, 192, 336, 720\}$) on **zero-shot forecasting** task. Where $Source \rightarrow Traget$ indicates that the model is first pre-trained on the single train set of the *SourceDomain*, subsequently, the model parameters are frozen and predicted on the test set of the *TargetDomain*.

Scenarios	**WQ4TS**		OneFitsAll		DLinear		PatchTST		TimesNet		FEDformer		Autoformer	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm2→ETTm1	**0.434**	**0.429**	0.790	0.579	0.516	0.473	0.596	0.508	0.857	0.599	0.718	0.564	0.722	0.566
ETTm1→ETTm2	**0.293**	**0.326**	0.342	0.369	0.360	0.410	0.325	0.361	0.357	0.384	0.321	0.360	0.325	0.365
ETTh2→ETTh1	**0.512**	**0.493**	0.780	0.604	0.609	0.532	0.616	0.537	0.920	0.635	0.746	0.598	0.735	0.593
ETTh1→ETTh2	**0.385**	**0.405**	0.420	0.430	0.478	0.483	0.416	0.444	0.443	0.442	0.444	0.463	0.445	0.459
RiverFlow→Exchange	**0.381**	**0.424**	0.464	0.491	0.585	0.537	0.421	0.458	0.497	0.508	0.942	0.765	0.845	0.739
Sunspot→Weather	**0.254**	**0.286**	0.264	0.297	0.263	0.310	0.263	0.297	0.311	0.325	0.705	0.634	0.509	0.501

Table 6: Comparison of the averaged performance from diverse prediction lengths on **zero-shot forecasting** task, where $Source \rightarrow Traget$ indicates that the model is first pre-trained uniformly on all train sets from multiple *SourceDomains*, subsequently, the model parameters are frozen and predicted on the test set of the *TargetDomain*.

Scenarios	**Zero-shot**								**Full-data**				**Few-shot**			
Models	**WQ4TS**								**OneFitsAll**		**PatchTST**		**OneFitsAll**		**PatchTST**	
Metric	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
SourceData	ETT{m2,h1,h2}		ETTm2		ETTh1		ETTh2		ETTm1		ETTm1		ETTm1		ETTm1	
SourceData→ETTm1	**0.411**	**0.416**	0.434	0.429	0.682	0.742	0.742	0.802	0.352	0.383	**0.351**	**0.380**	0.472	0.450	0.526	0.476
SourceData	ETT{m1,h1,h2}		ETTm1		ETTh1		ETTh2		ETTm2		ETTm2		ETTm2		ETTm2	
SourceData→ETTm2	**0.280**	**0.315**	0.292	0.326	0.316	0.359	0.316	0.360	0.266	0.326	**0.255**	**0.315**	0.308	0.346	0.314	0.352
SourceData	ETT{m1,m2,h2}		ETTm1		ETTm2		ETTh2		ETTh1		ETTh1		ETTh1		ETTh1	
SourceData→ETTh1	**0.461**	**0.449**	0.512	0.493	0.536	0.499	0.578	0.521	0.427	**0.426**	**0.413**	0.430	0.693	0.568	0.712	0.580
SourceData	ETT{m1,m2,h1}		ETTm1		ETTm2		ETTh1		ETTh2		ETTh2		ETTh2		ETTh2	
SourceData→ETTh2	**0.371**	**0.384**	0.430	0.433	0.408	0.422	0.385	0.405	0.354	0.394	**0.330**	**0.379**	0.413	0.441	0.449	0.456

Table 7: Comparison of the averaged performance from diverse prediction lengths ($\{96, 192, 336, 720\}$) on **few-shot forecasting** task, where all samples of trainset are only partially available (5%) in the training phase.

Models	**WQ4TS**		OneFitsAll		DLinear		PatchTST		TimesNet		FEDformer		Autoformer	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.394**	**0.381**	0.472	0.450	0.400	0.417	0.526	0.476	0.717	0.561	0.730	0.592	0.796	0.620
ETTm2	**0.267**	**0.306**	0.308	0.346	0.399	0.426	0.314	0.352	0.344	0.372	0.381	0.404	0.388	0.433
ETTh1	**0.521**	**0.449**	0.681	0.560	0.750	0.611	0.694	0.569	0.925	0.647	0.658	0.562	0.722	0.598
ETTh2	**0.373**	**0.381**	0.400	0.433	0.827	0.615	0.439	0.448	0.463	0.454	0.441	0.457	0.470	0.489
Weather	**0.233**	**0.262**	0.263	0.301	0.263	0.308	0.269	0.303	0.298	0.318	0.309	0.353	0.310	0.353

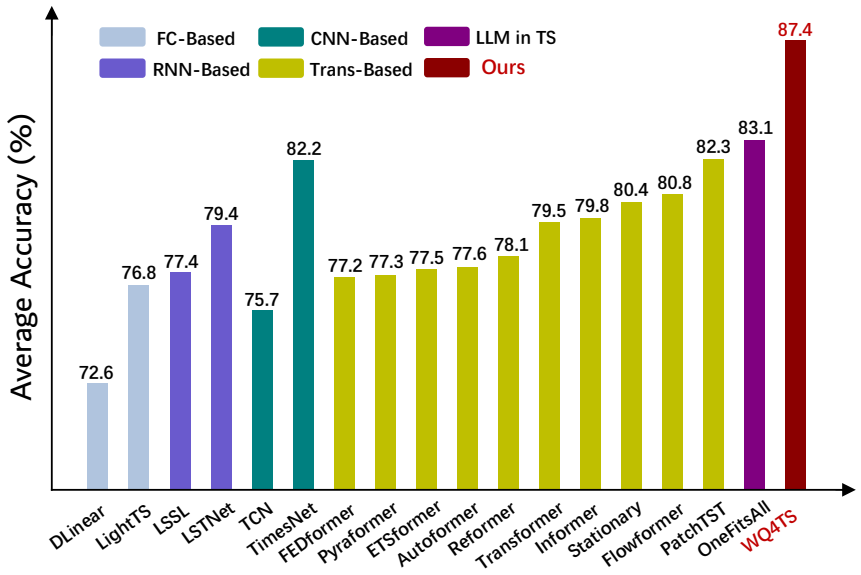

Figure 2: Model comparison in classification. The results are averaged from 35 subsets of UCR. The proposed WQ4TS achieves the best performance on the classification task under the full-data setting.

WQ4TS is reduced by **26.2%**, **19.6%**, **14.3%**, and **33.4%** compared to the existing OneFitsAll, DLinear, PatchTST, and TimesNet, respectively. As shown in Table 6, the performance of general unsupervised cross-domain migration on the zero-shot forecasting task would be far superior to that of single-domain pre-training, which indicates that our proposed wave quantize strategy could alleviate the negative migration on the time series. Besides, the performance of the proposed WQ4TS on the zero-shot task is much superior over the few-shot task of the existing SOTA models (Average MSE reduced by **31.4%**), and archives comparable performance to the full-data results of the existing SOTA models (Average MSE difference is only **8.1%**). In the few-shot (5%) forecasting task shown in Table 7, the average MSE of the proposed WQ4TS is reduced by **15.8%**, **32.2%**, **20.2%**, and **34.9%** compared to the existing OneFitsAll, DLinear, PatchTST, and TimesNet, respectively.

5.5 Imputation

Table 8: Comparison of the averaged performance from mask ratios ($\{12.5\%, 25\%, 37.5\%, 50\%\}$) on **full-data imputation** task.

Models	**WQ4TS (Ours)**		OneFitsAll [2023]		TimesNet [2023]		PatchTST [2023]		ETSformer [2023]		LightTS [2022a]		DLinear [2023]		FEDformer [2022a]		Stationary [2022]		Autoformer [2021]	
MaskRatio	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.026**	**0.099**	0.028	0.105	0.027	0.107	0.047	0.140	0.120	0.253	0.104	0.218	0.093	0.206	0.062	0.177	0.036	0.126	0.051	0.150
ETTm2	**0.020**	0.085	0.021	**0.084**	0.022	0.088	0.029	0.102	0.208	0.327	0.046	0.151	0.096	0.208	0.101	0.215	0.026	0.099	0.029	0.105
ETTh1	**0.064**	**0.167**	0.069	0.173	0.078	0.187	0.115	0.224	0.202	0.329	0.284	0.373	0.201	0.306	0.117	0.246	0.094	0.201	0.103	0.214
ETTh2	**0.047**	**0.138**	0.048	0.141	0.049	0.146	0.065	0.163	0.367	0.436	0.119	0.250	0.142	0.259	0.163	0.279	0.053	0.152	0.055	0.156
Electricity	**0.053**	**0.147**	0.090	0.207	0.092	0.210	0.072	0.183	0.214	0.339	0.131	0.262	0.132	0.260	0.130	0.259	0.100	0.218	0.101	0.225
Weather	**0.028**	**0.046**	0.031	0.056	0.030	0.054	0.060	0.144	0.076	0.171	0.055	0.117	0.052	0.110	0.099	0.203	0.032	0.059	0.031	0.057

Experimental setups: The imputation task, that is predicting the masked portion of the original series based on the unmasked portion. Similarly, to the forecasting task, Table 8 and Table 9 show the experimental results for the imputation task under the full-data and zero-shot settings, respectively.

Table 9: Comparison of the averaged performance from diverse mask ratios ($\{12.5\%, 25\%, 37.5\%, 50\%\}$) on **zero-shot imputation** task. Where $Source \rightarrow Traget$ indicates that the model is first pre-trained on the train set of the *SourceDomain*, subsequently, the model parameters are frozen and predicted on the test set of the *TargetDomain*.

Scenarios	**WQ4TS**		OneFitsAll		DLinear		PatchTST		TimesNet		FEDformer		Autoformer	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm2→ETTm1	**0.050**	**0.144**	0.767	0.549	0.203	0.295	0.099	0.191	0.118	0.205	0.762	0.655	0.507	0.501
ETTm1→ETTm2	**0.029**	**0.098**	0.145	0.256	0.114	0.224	0.058	0.149	0.093	0.216	2.140	1.113	1.342	0.842
ETTm1→ETTh1	**0.176**	**0.274**	0.854	0.602	0.397	0.424	0.313	0.366	0.327	0.395	1.074	0.787	0.956	0.725
ETTm1→ETTh2	**0.064**	**0.160**	0.245	0.333	0.160	0.277	0.079	0.184	0.109	0.238	2.796	1.266	2.473	1.206
ETTm1→Exchange	**0.003**	**0.031**	0.027	0.117	0.358	0.437	0.006	0.044	0.045	0.150	3.107	1.440	2.904	1.382
ETTm1→Weather	**0.030**	**0.043**	0.103	0.160	0.174	0.283	0.065	0.099	0.132	0.188	0.999	0.779	1.000	0.788

Analysis of results: In the full-data imputation task shown in Table 8, the proposed WQ4TS exhibits the best performance in **91.6%** of the metrics. In the zero-shot imputation task shown in Table 9, the average MSE values of the proposed WQ4TS are reduced by **83.6%**, **74.9%**, **43.2%**, and **57.3%** compared to the existing OneFitsAll, DLinear, PatchTST, and TimesNet, respectively.

5.6 Classification

Table 10: Comparison of the accuracy and F1-score on **few-shot classification** task. Where *Scenario-i: Source-i→Traget-i* ($i \in [0, .., 7]$) indicates that the model is first pre-trained in the *SourceDomain*, subsequently, the parameters are fine-tuned in partial (5%/10%) samples of the *TargetDomain* and finally predicted on the *TargetDomain*.

Scenarios		Scenario-1		Scenario-2		Scenario-3		Scenario-4		Scenario-5		Scenario-6		Scenario-7	
Task	Models	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
Full Data	TimesNet	**85.22**	**82.99**	**97.96**	**97.64**	76.37	76.61	78.10	75.26	**97.18**	**97.18**	95.00	95.00	86.34	71.11
	PatchTST	78.69	75.39	95.63	94.80	69.55	72.08	83.13	78.10	96.60	96.61	**98.89**	**98.89**	**89.27**	**84.07**
	OneFitsAll	79.73	77.20	96.79	96.32	**79.37**	**80.62**	83.03	79.50	96.99	96.99	98.33	98.34	88.78	83.12
Few-shot (10%)	Random init.	75.26	70.60	66.79	66.36	58.51	46.53	47.39	29.49	78.92	79.07	50.00	33.33	41.46	18.58
	TimesNet	77.32	74.44	91.25	89.43	64.23	66.07	83.13	66.52	86.59	86.69	83.89	83.83	85.85	72.08
	PatchTST	78.69	74.57	80.76	75.36	63.56	64.40	48.19	45.63	87.37	87.59	90.00	89.96	87.80	80.67
	OneFitsAll	77.66	74.34	78.13	71.24	64.56	64.03	83.13	67.34	94.66	94.68	97.23	97.23	87.32	71.62
	WQ4TS	**86.25**	**84.65**	**94.17**	**93.62**	**91.51**	**91.38**	**93.17**	**91.03**	**96.31**	**96.31**	**98.89**	**98.89**	**89.27**	**84.61**
Few-shot (5%)	Random init.	68.38	48.30	27.41	28.16	47.09	37.94	35.74	24.63	70.86	71.21	50.00	33.33	36.59	16.39
	TimesNet	78.01	73.61	27.41	28.16	60.23	57.26	35.74	26.61	73.28	74.01	78.89	78.85	85.85	63.18
	PatchTST	78.35	74.52	81.05	75.44	57.07	42.98	37.35	30.04	60.93	61.27	82.78	82.73	87.32	79.84
	OneFitsAll	76.98	72.85	27.41	28.16	57.07	42.98	35.74	24.63	83.19	83.65	90.00	89.99	84.39	58.62
	WQ4TS	**84.54**	**82.11**	**91.55**	**89.76**	**85.03**	**86.09**	**83.85**	**81.93**	**94.85**	**94.85**	**95.00**	**95.00**	**87.80**	**82.48**

Table 11: Comparison of the ablation experiment, where *w/o WQ* indicates the model without the proposed *wave quantize* module. To guarantee fairness, *w/o WQ* and *WQ4TS* have the same model structure and parameter size.

Variant	full datal				few shot (10%)				zero shot (single)				zero shot (multi)			
	ETTh1		ETTm1		ETTh1		ETTm1		ETTh1		ETTm1		ETTh1		ETTm1	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/o TP	0.582	0.542	0.513	0.480	0.643	0.581	0.524	0.489	0.704	0.628	0.599	0.548	0.677	0.606	0.623	0.564
WQ4TS	0.407	0.418	0.368	0.369	0.477	0.448	0.383	0.380	0.512	0.493	0.434	0.429	0.461	0.449	0.411	0.416
loss ↓ (%)	42.9	29.6	39.4	30.1	34.8	29.7	36.8	28.7	37.5	27.4	38.0	27.7	46.9	34.9	51.6	35.6

Table 12: The detailed information of Scenarios in Table 10. All of the sub-datasets involved in seven scenarios are derived from the UCR dataset presented in the section 5.2.

Cross-domain	Source domain	Target domain
Scenario-1	DistalPhalanxTW	ProximalPhalanxOutlineCorrect
Scenario-2	SonyAIBORobotSurface2	Chinatown
Scenario-3	SonyAIBORobotSurface2	SonyAIBORobotSurface1
Scenario-4	PigArtPressure	InsectEPGRegularTrain
Scenario-5	Earthquakes	ItalyPowerDemand
Scenario-6	BeetleFly	PowerCons
Scenario-7	EOGHorizontalSignal	ProximalPhalanxOutlineAgeGroup

Experimental setups: First, Fig. 2 shows the average accuracy of the existing baselines and the proposed WQ4TS on all 35 classification datasets, where each coloration represents multiple models established by the specific backbone. In addition, Table 10 demonstrates the cross-domain migration capability of the proposed WQ4TS , where the upper and lower parts represent the accuracy and F1 score of all existing baselines in the full-data and few-shot (5% and 10%) settings, respectively. Specifically, each *scenario-i* represents a specific tuple of source and target domain, where each domain contains trainset and testset. Further, in the full data setting, the model is trained and tested on the target trainset and target testset. In the few-shot setting, the model is first pre-trained and tested on the source trainset and source testset, and the all-parameters will be fine-tuned in portion data (5% and 10%) of the target trainset. The detailed information of the 7 scenarios shown therein is given in the Table 12.

Analysis of results: Fig. 2 demonstrates the average accuracy of the proposed approach and the existing models on all 35 classification datasets, with the proposed WQ4TS showing the best performance. Table 10 demonstrates the unsupervised cross-domain migration capability of the proposed method. Besides, it is noteworthy that the proposed approach archives comparable or superior performance in few-shot settings over state-of-the-art models under full-data settings, in all seven scenarios. Concretely, the average accuracy of the proposed WQ4TS is increased by **23.9%**, **19.6%**, **26.1%**, and **30.0%** compared to the existing OneFitsAll, PatchTST, TimesNet, and MICN respectively.

5.7 Ablation Study

To elaborate on the property of our proposed WQ4TS , we conduct detailed ablations on model architecture. As shown in Table 11, we find that removing the *wave quantize* module in WQ4TS will cause significant performance degradation. These results may come from that the proposed feature program will improve the generalization capability of models to learn representation from complex series and migrate it to never-before-seen domains. Specifically, for fairness purposes, the ablation model *w/o WQ* is designed to have the same Encoder and OutputLayer structure as WQ4TS , using only 1D Convolution in place of the proposed *wave quantize* module, which can be expressed as $Conv : \mathbb{R}^{1\times l} \mapsto \mathbb{R}^{\lambda\times l}$. From Table 11, we can find that the performance of *w/o WQ* degenerates **35.5%** in the full-data task, degenerates **32.5%** in the few-shot task, degenerates **32.7%** and **42.3%** in the zero-shot task under single-domain and multi-domain respectively. Similar results are found in other datasets, which indicate the advantages of our design.

5.8 Tokenization Paradigm Strategy Analysis

In this subsection, we investigate the generalizability of the tokenization paradigm strategy of **Wave Quantize** by plugging it into other different kinds of models. **Model selection and experimental setting.** To achieve this objective, we conduct experiments across a spectrum of representative time series forecasting model structures, including (1) Transformer-based methods: PatchTST Nie et al. [2023], FEDformer Zhou et al. [2022a] and Autoformer Wu et al. [2021]; (2) Linear-based methods: DLinear Zeng et al. [2023] (3) TCN-based methods: TimesNet Wu et al. [2023]; (4) LLM-based methods: OneFitsAll Zhou et al. [2023].

We standardize the input length to 336, and similarly, the prediction length is uniformly set to 336. Subsequently, comparative experiments were conducted on five datasets: ETTh1, ETTh2, ETTm1, ETTm2, RiverFlow, Exchange, Sunspot, and Weather. Specifically, we sequentially replaced the wave quantize components with each model. The comparative analysis was performed to assess the predictive performance before and after the incorporation of **Wave Quantize**, comparing the original models with the augmented counterparts.

In Table 13, it is apparent that the incorporation of the Wave Quantize structure leads to a notably substantial enhancement in the predictive performance of various models, even with the introduction of only a single pre-standardized layer.

Table 13: Improvements of **Wave Quantize Module (WQ)** over different models with prediction lengths $F = 336$, and fixed lookback length $T = 336$.

	Models	OneFitsAll		DLinear		PatchTST		TimesNet		FEDformer		Autoformer	
	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm2 ↓ ETTm1	Original	0.790	0.579	0.516	0.473	0.596	0.508	0.857	0.599	0.718	0.564	0.722	0.566
	+ **WQ**	**0.621**	**0.497**	**0.425**	**0.419**	**0.490**	**0.461**	**0.583**	**0.490**	**0.526**	**0.470**	**0.518**	**0.469**
	Improve	**21.39%**	**14.16%**	**17.64%**	**11.42%**	**17.79%**	**9.25%**	**31.97%**	**18.21%**	**26.74%**	**16.57%**	**28.26%**	**17.14%**
ETTm1 ↓ ETTm2	Original	0.342	0.369	0.360	0.410	0.325	0.361	0.357	0.384	0.321	0.360	0.325	0.365
	+ **WQ**	**0.335**	**0.352**	**0.316**	**0.335**	**0.287**	**0.319**	**0.321**	**0.342**	**0.289**	**0.315**	**0.293**	**0.316**
	Improve	**2.05%**	**4.61%**	**12.30%**	**18.29%**	**11.69%**	**11.63%**	**10.08%**	**10.94%**	**9.97%**	**12.50%**	**9.85%**	**13.42%**
ETTh2 ↓ ETTh1	Original	0.780	0.604	0.609	0.532	0.616	0.537	0.920	0.635	0.746	0.598	0.735	0.593
	+ **WQ**	**0.679**	**0.541**	**0.483**	**0.469**	**0.517**	**0.487**	**0.649**	**0.525**	**0.534**	**0.520**	**0.515**	**0.508**
	Improve	**12.94%**	**10.43%**	**20.69%**	**11.84%**	**16.07%**	**9.31%**	**29.46%**	**17.32%**	**28.42%**	**13.04%**	**29.93%**	**14.33%**
ETTh1 ↓ ETTh2	Original	0.420	0.430	0.478	0.483	0.416	0.444	0.443	0.442	0.444	0.463	0.445	0.459
	+ **WQ**	**0.408**	**0.412**	**0.419**	**0.428**	**0.382**	**0.408**	**0.413**	**0.420**	**0.421**	**0.437**	**0.425**	**0.439**
	Improve	**2.86%**	**4.19%**	**12.34%**	**11.39%**	**8.17%**	**8.10%**	**6.77%**	**4.98%**	**5.18%**	**5.62%**	**4.49%**	**4.36%**
RiverFlow ↓ Exchange	Original	0.464	0.491	0.585	0.537	0.421	0.458	0.497	0.508	0.942	0.765	0.845	0.739
	+ **WQ**	**0.451**	**0.476**	**0.431**	**0.460**	**0.405**	**0.427**	**0.477**	**0.491**	**0.460**	**0.481**	**0.496**	**0.509**
	Improve	**2.86%**	**3.05%**	**26.32%**	**14.34%**	**3.80%**	**6.71%**	**4.02%**	**3.35%**	**51.17%**	**37.12%**	**41.30%**	**31.20%**

Specifically, DLinear demonstrates an average MSE reduction of **17.79%** across five datasets, other models are OneFitsAll: **8.41%**, PatchTST: **11.65%**, TimesNet: **16.72%**, FEDformer: **24.64%**, and Autoformer: **22.67%**. Particularly noteworthy is the performance enhancement observed in the classical FEDformer model, where the MSE experiences a remarkable decrease of **51.17%** and **26.74%** on *RiverFlow→Exchange* and *ETTm2→ETTm1*, respectively, a result that is profoundly surprising. This unequivocally substantiates the generality of the Wave Quantize structure.

6 Conclusion and future work

This paper presents the concept of the tokenization paradigm in time series for the first time and proposes the novel wavebook tokenization that advances research related to multi-domain unified pre-train and cross-domain adaptation without changing the model structure, which will lay the groundwork for the large time series model. To verify that wavebook strategy can mitigate negative migration, the experiments showed encouraging results: unified multi-domain pre-training would be far superior to single-domain pre-training. In addition, modeling multi-period features from time series will be an important research direction, especially the period features across channels. For example, capturing correlation representations between multiple underlying periodic patterns across channels is essential for multivariate time series tasks.

References

K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 95–104, 2017.

T. B. Brown, B. Mann, N. Ryder, and M. S. et al. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

R. Cai, J. Chen, Z. Li, W. Chen, K. Zhang, J. Ye, Z. Li, X. Yang, and Z. Zhang. Time series domain adaptation via sparse associative structure alignment. In *AAAI Conference on Artificial Intelligence*, 2021.

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

C. Chang, W. Peng, and T.-F. Chen. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. In *ArXiv*, 2023.

C. Chen, Z. Fu, Z. Chen, S. Jin, Z. Cheng, X. Jin, and X. Hua. Homm: Higher-order moment matching for unsupervised domain adaptation. In *AAAI Conference on Artificial Intelligence*, 2019.

H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, and Hexagon-ML. The ucr time series classification archive, October 2018. `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`.

Z. Du, J. Li, H. Su, L. Zhu, and K. Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3936–3945, 2021.

Y. Fang, Y. Qin, H. Luo, F. Zhao, B. Xu, L. Zeng, and C. Wang. When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks. *ICDE*, pages 517–529, 2023.

S. E. Finder, Y. Zohav, M. Ashkenazi, and E. Treister. Wavelet feature maps compression for image-to-image cnns. *ArXiv*, abs/2205.12268, 2022.

I. Godfried, K. Mahajan, M. Wang, K. Li, and P. Tiwari. Flowdb a large scale precipitation, river, and flash flood dataset, 2020.

S. Guo, Y. Lin, L. Gong, C. Wang, Z. Zhou, Z. Shen, Y. Huang, and H. Wan. Self-supervised spatial-temporal bottleneck attentive network for efficient long-term traffic forecasting. *ICDE*, pages 1585–1596, 2023.

H. He, O. Queen, T. Koker, C. Cuevas, T. Tsiligkaridis, and M. Zitnik. Domain adaptation for time series under feature and label shifts. In *International Conference on Machine Learning, ICML*, 2023.

Y. Hong, L. Niu, J. Zhang, and L. Zhang. Matchinggan: Matching-based few-shot image generation. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. S. Pande, and J. Leskovec. Strategies for pre-training graph neural networks. *arXiv: Learning*, 2019.

J. Jeon, J. KIM, H. Song, S. Cho, and N. Park. Gt-gan: General purpose time series synthesis with generative adversarial networks. In *NeurIPS*, 2022.

M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. L. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen. Time-llm: Time series forecasting by reprogramming large language models. *ArXiv*, abs/2310.01728, 2023.

X. Jin, Y. Park, D. C. Maddix, B. Wang, and X. Yan. Domain adaptation for time series forecasting via attention sharing. In *International Conference on Machine Learning, ICML*, 2022.

G. Kang, L. Jiang, Y. Yang, and A. Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4888–4897, 2019.

N. Karmitsa, S. Taheri, A. M. Bagirov, and P. Mäkinen. Missing value imputation via clusterwise linear regression. *IEEE Transactions on Knowledge and Data Engineering*, 34:1889–1901, 2022.

A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018.

N. Kitaev, L. Kaiser, and A. Levskaya. Reformer: The efficient transformer. *ArXiv*, abs/2001.04451, 2020a.

N. Kitaev, L. Kaiser, and A. Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020b.

T. Kudo and J. Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Conference on Empirical Methods in Natural Language Processing*, 2018.

G. Lai, W.-C. Chang, Y. Yang, and H. Liu. Modeling long- and short-term temporal patterns with deep neural networks. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2017.

J. Li, G. Li, Y. Shi, and Y. Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2505–2514, 2021.

J. Y. Li, C. Liu, S. Cheng, R. Arcucci, and linda Qiao. Frozen language model helps ecg zero-shot learning. *ArXiv*, abs/2303.12311, 2023.

Y. Li, X. Lu, Y. Wang, and D. Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. In *NeurIPS*, 2022.

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

X. Liu, J. Hu, Y. Li, S. Diao, Y. Liang, B. Hooi, and R. Zimmermann. Unitime: A language-empowered unified model for cross-domain time series forecasting. *ArXiv*, abs/2310.09751, 2023a.

Y. Liu, H. Wu, J. Wang, and M. Long. Non-stationary transformers: Rethinking the stationarity in time series forecasting. In *NeurIPS*, 2022.

Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long. itransformer: Inverted transformers are effective for time series forecasting, 2023b.

S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, and B. Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *ArXiv*, abs/2211.14730, 2023.

P. Oza, V. A. Sindagi, V. Vs, and V. M. Patel. Unsupervised domain adaptation of object detectors: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46:4018–4040, 2021.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, and T. Killeen. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

A. Radford and K. Narasimhan. Improving language understanding by generative pre-training. In *ArXiv*, 2018.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. In *ArXiv*, 2019.

M. Ragab, E. Eldele, Z. Chen, M. Wu, C. Kwoh, and X. Li. Self-supervised autoregressive domain adaptation for time series data. *IEEE Transactions on Neural Networks and Learning Systems*, 35:1341–1351, 2021.

M. Ragab, E. Eldele, W. L. Tan, C.-S. Foo, Z. Chen, M. Wu, C. Kwoh, and X. Li. Adatime: A benchmarking suite for domain adaptation on time series data. *ACM Transactions on Knowledge Discovery from Data*, 17:1–18, 2022.

M. G. Schultz, C. Betancourt, B. Gong, F. Kleinert, M. Langguth, L. H. Leufen, A. Mozaffari, and S. Stadtler. Can deep learning beat numerical weather prediction? *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 379, 2021.

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. *ArXiv*, abs/1508.07909, 2015.

A. Seyfi, J.-F. Rajotte, and R. T. Ng. Generating multivariate time series with common source coordinated GAN (COSCI-GAN). In *NeurIPS*, 2022.

A. Shabani, A. H. S. Abdi, L. Meng, and T. Sylvain. Scaleformer: Iterative multi-scale refining transformers for time series forecasting. *ArXiv*, 2022.

X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W.-K. Wong, and W. chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015.

C. Sun, Y. Li, H. Li, and linda Qiao. Test: Text prototype aligned embedding to activate llm's ability for time series. *ArXiv*, abs/2308.08241, 2023.

H. Touvron, T. Lavril, G. Izacard, and X. M. et al. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023a.

H. Touvron, L. Martin, K. R. Stone, and e. a. Peter Albert. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023b.

E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017.

A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In *NIPS*, volume abs/1711.00937, 2017.

A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

V. Vibashan, V. Gupta, P. Oza, V. A. Sindagi, and V. M. Patel. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4514–4524, 2021.

C. Wang, C. Xu, and D. Tao. Self-supervised pose adaptation for cross-domain image animation. *IEEE Transactions on Artificial Intelligence*, 1:34–46, 2020a.

H. Wang, H. He, and D. Katabi. Continuously indexed domain adaptation. In *ICML*, 2020b.

H. Wang, J. Peng, F. Huang, J. Wang, J. Chen, and Y. Xiao. MICN: Multi-scale local and global context modeling for long-term series forecasting. In *ICLR*, 2023.

R. Wang, D. C. Maddix, C. Faloutsos, B. Wang, and R. Yu. Bridging physics-based and data-driven modeling for learning dynamical systems. *ArXiv*, abs/2011.10616, 2020c.

Z. Wang, X. Xu, W. Zhang, G. Trajcevski, T. Zhong, and F. Zhou. Learning latent seasonal-trend representations for time series forecasting. In *NeurIPS*, 2022a.

Z. Wang, Y. Zhou, R. Wang, T.-Y. Lin, A. Shah, and S. N. Lim. Few-shot fast-adaptive anomaly detection. In *Neural Information Processing Systems (NeurIPS)*, 2022b.

G. Wilson and D. J. Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11:1–46, 2018.

G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi. CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *ICLR*, 2022a.

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

H. Wu and M. K. Ng. Multiple graphs and low-rank embedding for multi-source heterogeneous domain adaptation. *ACM Trans. Knowl. Discov. Data*, 16:77:1–77:25, 2022.

H. Wu, J. Xu, J. Wang, and M. Long. Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting. In *NeurIPS*, 2021.

H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *ICLR*, 2023.

Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

T. Xu, W. Chen, P. Wang, F. Wang, H. Li, and R. Jin. Cdtrans: Cross-domain transformer for unsupervised domain adaptation. In *ICLR*, 2022.

H. Xue and F. D.Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

K. Yi, Q. Zhang, W. Fan, H. He, L. Hu, P. Wang, N. An, L. Cao, and Z. Niu. FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective. In *NeurIPS*, 2023a.

K. Yi, Q. Zhang, W. Fan, S. Wang, P. Wang, H. He, N. An, D. Lian, L. Cao, and Z. Niu. Frequency-domain MLPs are more effective learners in time series forecasting. In *NeurIPS*, 2023b.

A. Zeng, M. Chen, L. Zhang, and Q. Xu. Are transformers effective for time series forecasting? In *AAAI*, 2023.

Q. Zhang, D. Guo, X. Zhao, L. Yuan, and L. Luo. Discovering frequency bursting patterns in temporal graphs. *ICDE*, pages 599–611, 2023.

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

X. Zhang, Z. Zhao, T. Tsiligkaridis, and M. Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. In *NeurIPS*, 2022b.

Y. Zhang, B. Deng, H. Tang, L. Zhang, and K. Jia. Unsupervised multi-class domain adaptation: Theory, algorithms, and practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:2775–2792, 2020.

S. Zhao, X. Yue, S. Zhang, B. Li, H. Zhao, B. Wu, R. Krishna, J. Gonzalez, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and K. Keutzer. A review of single-source deep unsupervised visual domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 33:473–493, 2020.

H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, 2021.

T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML*, 2022a.

T. Zhou, Z. Ma, xue wang, Q. Wen, L. Sun, T. Yao, W. Yin, and R. Jin. Film: Frequency improved legendre memory model for long-term time series forecasting. In *NeurIPS*, 2022b.

T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin. One fits all: Power general time series analysis by pretrained LM. In *NIPS*, 2023.

A Background

Function space We define the space $L^2(\mathbb{R})$ to describe all functions defined on $x \in [-\infty, +\infty]$ with finite energy

$$L^2(\mathbb{R}) = \left\{ f(t) \colon \int_{-\infty}^{+\infty} |f(t)|^2 < +\infty \right\}. \tag{16}$$

Wavelet generating function and wavelet function If $\psi(t) \in L^2(\mathbb{R})$ satisfies

$$C_\psi = \int_{-\infty}^{+\infty} \frac{\left|\hat{\psi}(\omega)\right|^2}{|\omega|} d\omega < +\infty, \tag{17}$$

then $\psi(t)$ is called the wavelet generating function, where C_ψ is called the tolerance parameter. For any $\forall a \neq 0, b \in \mathbb{R}$, we define the continuous wavelet function as follows:

$$\psi_{(a,b)}(t) = \frac{1}{\sqrt{|a|}} \psi(\frac{t-b}{a}). \tag{18}$$

In addition, the energy of the wavelet generating function A will only be distributed over the finite interval, the distribution of the energy decaying rapidly to converge to zero as time approaches infinity, and there exists a horizontal line such that the wavelet function integrates up and down the area of the line for this line. Thus wavelets have two characteristics, the attenuation feature and the fluctuation feature:

$$\begin{gathered} \int_{-\infty}^{+\infty} |\psi(t)|^2 \, dt < +\infty, \int_{-\infty}^{+\infty} \left|\psi_{(a,b)}(t)\right|^2 dt < +\infty, \\ \int_{-\infty}^{+\infty} \psi(t) dt = 0, \int_{-\infty}^{+\infty} \psi_{(a,b)}(t) dt = 0. \end{gathered} \tag{19}$$

Wavelet transform For $\forall f(t) \in L^2(\mathbb{R})$, we have

$$W_f(a,b) = \int_{-\infty}^{+\infty} f(t) \overline{\psi}_{(a,b)}(t) dt, \tag{20}$$

which is considered to be the wavelet transform of the signal $f(t)$.

Orthogonal wavelet Taking $a = 2^{-j}, b = 2^{-j}k$ in (18), we get a set of wavelets

$$\left\{ \psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k), (j,k) \in \mathbb{Z}^2 \right\}, \tag{21}$$

which forms the orthonormal basis (O.N.B) of $L^2(\mathbb{R})$, that is $\psi(t)$ as the orthogonal wavelet. Thus, any signal $f(t)$ defined in $L^2(\mathbb{R})$ can be described as a linear representation of this set of orthonormal basis:

$$f(t) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} \alpha_{j,k} \cdot \psi_{j,k}(t). \tag{22}$$

Since $\left\{ \psi_{j,k}(t), (j,k) \in \mathbb{Z}^2 \right\}$ are the mutually orthonormal basis, the coefficients of the above linear representation can be calculated by

$$\alpha_{j,k} = \int_{-\infty}^{+\infty} f(t) \overline{\psi}_{j,k}(t) dt = W_f(2^{-j}, 2^{-j}k), \tag{23}$$

and the signal $f(t)$ has the following complete representation

$$f(t) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} W_f(2^{-j}, 2^{-j}k) \cdot \psi_{j,k}(t). \tag{24}$$

To satisfy the efficient computation of GPUs in the Pytorch environment, we would like to discretize some values to carve a continuous piece of function. Based on Shannon Sampling Theorem, substituting (24) into (20) yields

$$W_f(a,b) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} W_f(2^{-j}, 2^{-j}k) \times \int_R \psi_{j,k}(t) \overline{\psi}_{(a,b)}(t) dt. \tag{25}$$

Since the part within the integral is only related to a, b, j, k, the process of (25) can be calculated by discrete sampling. This means that when doing the wavelet transform on a continuous signal, the information has coalesced into discrete sampling points, which provides rationalization for subsequent calculations.

Scale function In the Shannon Sampling Theorem, for any signal $f(t)$ defined on $L^2(\mathbb{R})$, if the frequency domain form $\hat{f}(\omega)$ of that signal has a truncation frequency B, then that signal can be reconstructed by equally spaced discrete sampling. This sampling interval can be at most $\frac{\pi}{B}$. If the function $f(t)$ satisfies the following conditions

$$\begin{aligned} &\forall f(t) \in L^2(\mathbb{R}), \\ &\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i\omega t} dt = 0, |\omega| > B, \end{aligned} \tag{26}$$

then we have

$$f(t) = \sum_{n \in \mathbb{Z}} f(n\Delta) \frac{sin(\pi/\Delta)(t - n\Delta)}{(\pi/\Delta)(t - n\Delta)}, 0 < \Delta \leq \frac{\pi}{B}. \tag{27}$$

Considering first the case $B = \pi$ in (27), we have

$$f(t) = \sum_{n \in \mathbb{Z}} f(n) \frac{sin(\pi(t-n))}{\pi(t-n)} = \sum_{n=-\infty}^{+\infty} f(n)\phi(t-n), \tag{28}$$

where the scale function is defined as $\phi(t) = \frac{\sin(\pi t)}{\pi t}$.

Further, define the space $V_0 = \left\{ f(t); \hat{f}(\omega) = 0, |\omega| > \pi \right\}$ with truncation frequency $B = \pi$i, where $\left\{ \phi_{0,n} = 2^{0/2}\phi(2^0 t - n) = \phi(t-n); n \in \mathbb{Z} \right\}$ constitutes a set of orthonormal basis (O.N.B) which forms V_0. Specifically, we have

$$\begin{aligned} \langle \phi(t-n), \phi(t-m) \rangle &= \int_{-\infty}^{+\infty} \phi(t-n)\overline{\phi}(t-m) dt, \\ &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} (\hat{\phi}(\omega) e^{-in\omega})(\hat{\phi}(\omega) e^{-im\omega}) d\omega, \\ &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left| \hat{\phi}(\omega) \right|^2 e^{-i(n-m)\omega} d\omega, \\ &= \frac{1}{2\pi} \int_{-\pi}^{+\pi} e^{-i(n-m)\omega} d\omega. \end{aligned} \tag{29}$$

We introduce the function δ for refined expression, that is

$$\forall m, n \in \mathbb{Z}, \langle \phi(t-n), \phi(t-m) \rangle = \delta(n-m). \tag{30}$$

Defining the space $V_j = \left\{ f(t); \hat{f}(\omega) = 0, |\omega| > 2^j \pi \right\}$ with truncation frequency $B = 2^j \pi$ and taking the sampling interval $\Delta = 2^{-j}$, we have

$$\begin{aligned} f(t) &= \sum_{n=-\infty}^{+\infty} f(2^{-j}n) \frac{sin\pi(2^j t - n)}{\pi(2^j t - n)}, \\ &= \sum_{n=-\infty}^{+\infty} 2^{-j/2} f(2^{-j}n) \cdot 2^{j/2} \phi(2^j t - n), \\ &= \sum_{n=-\infty}^{+\infty} 2^{-j/2} f(2^{-j}n) \phi_{j,n}(t), \end{aligned} \tag{31}$$

where

$$\phi_{j,n}(t) = 2^{j/2}\phi(2^j t - n) = \frac{sin\pi(2^j t - n)}{\pi(2^j t - n)}. \tag{32}$$

Besides, we have

$$\begin{aligned}\langle \phi_{j,n}(t), \phi_{j,m}(t) \rangle &= \int_{-\infty}^{+\infty} \phi_{j,n}(t)\overline{\phi}_{j,m}(t)dt, \\ &= \frac{1}{2\pi \cdot 2^j} \int_{-\infty}^{+\infty} \left|\hat{\phi}(2^{-j}\omega)\right|^2 e^{-i(n-m)\omega} d\omega, \\ &= \frac{1}{2\pi \cdot 2^j} \int_{-2^j\pi}^{2^j\pi} e^{-i(n-m)\omega} d\omega, \\ &= \delta(n-m).\end{aligned} \tag{33}$$

Therefore, $\phi_{j,n}$ constitutes a set of orthonormal basis of V_j, i.e., V_j is a linear subspace tensored by $\phi_{j,n}$.

Orthogonal wavelet and close-span space Based on the wavelet function $\psi_{j,k}$, we define the close-span space as:

$$W_j = closespan\left\{\psi_{j,k}(t)2^{j/2}\psi(2^j t - k), (j,k) \in \mathbb{Z}^2\right\}, \tag{34}$$

and the close-span space has the following characteristics: 1) Spatial orthogonality $W_j \perp W_{j+1}$, since $\psi_{j,k}$ and $\psi_{j+1,k}$ are mutually orthogonal to each other; 2) Spatial approximability $L^2(\mathbb{R}) = \bigoplus_{j=-\infty}^{+\infty} W_j$. Since orthogonal wavelets form a standard set of orthogonal bases of space $L^2(\mathbb{R})$, and the close-span space will approximate space $L^2(\mathbb{R})$ by the direct-sum operation; 3) The transfer relation of neighboring spaces: $g(t) \in W_j \Leftrightarrow g(2t) \in W_{j+1}$, which is known by the definition of close-span space. Based on the above characteristics, by calculating the basis functions of W_0, a set of orthogonal wavelets can be obtained, where each wavelet $\psi_{j,k}$ can be formed into a close-span space W_j and the direct-sum of all the closure spaces can be approximated to $L^2(\mathbb{R})$.

The formulaic description of the above idea is that, if $\psi(t-k)$ is the set of the orthonormal basis of W_0, then for any j there has $\left\{2^{j/2}\psi(2^j t - k), (j,k) \in \mathbb{Z}^2\right\}$ as the orthonormal basis of W_j. Moreover, it is easy to verify:

$$W_j \perp V_j, V_{j+1} = W_j \oplus V_j. \tag{35}$$

Thus, the construction of orthogonal wavelets is equivalent to finding a set of standard orthogonal bases for W_0.

Scale equation and low-pass filter Since the scale function $\phi(x) \in V_0 \subseteq V_1$, and there exists a set of orthonormal basis $\left\{\sqrt{2}\phi(2t-n); n \in \mathbb{Z}\right\}$ for V_1, there must exist a unique sequence of coefficients $\{h_n; n \in \mathbb{Z}\} \in l^2(\mathbb{Z})$ such that

$$\phi(t) = \sqrt{2}\sum_{n\in\mathbb{Z}} h_n \phi(2t-n), \tag{36}$$

which is regarded as the scale equation, and since $\phi(2t-n)$ is mutually orthogonal to each other for different n, the coefficients are calculated as follows

$$h_n = \left\langle \phi(t), \sqrt{2}\phi(2t-n) \right\rangle = \sqrt{2}\int_R \phi(t)\overline{\phi}(2t-n)dt. \tag{37}$$

In addition, the scale equation are converted to frequency domain form by Fourier transforms

$$\hat{\phi}(\omega) = H(\omega/2)\hat{\phi}(\omega/2), H(\omega) = \frac{1}{\sqrt{2}}\sum_{n\in\mathbb{Z}} h_n e^{-in\omega}, \tag{38}$$

where $H(\omega)$ is referred to as the low-pass filter and hence h_n is also referred to as the low-pass filter coefficients.

Wavelet equation and band-pass filter For the wavelet function $\psi(x) \in W_0 \subseteq V_1$, there exists $\{g_n; n \in \mathbb{Z}\} \in l^2$ such that

$$\psi(t) = \sqrt{2}\sum_{n\in\mathbb{Z}} g_n \phi(2t-n), \tag{39}$$

which is regarded as the wavelet equation, and since $\phi(2t-n)$ is orthogonal for different n, the coefficients are calculated as follows

$$g_n = \left\langle \psi(t), \sqrt{2}\phi(2t-n) \right\rangle = \sqrt{2}\int_R \psi(t)\overline{\phi}(2t-n)dt. \tag{40}$$

In addition, the wavelet equations can be obtained in frequency domain form by Fourier transformation

$$\hat{\psi(\omega)} = G(\omega/2)\hat{\phi}(\omega/2), G(\omega) = \frac{1}{\sqrt{2}}\sum_{n\in\mathbb{Z}} g_n e^{-in\omega}, \tag{41}$$

where $G(\omega)$ is referred to as the bandpass filter and g_n is also referred to as the impulse response coefficient.

B Proof of results in Section 3.2

Proof of Proposition 2. If $\psi(t)$ is the orthogonal wavelet, we have that $\left\{\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k), (j,k) \in \mathbb{Z}^2\right\}$ constitutes a standard orthonormal basis for $L^2(\mathbb{R})$. Then there must exist the unique coefficients sequence $\left\{c_{j,k}; (j,k) \in \mathbb{Z}^2\right\} \in l^2(\mathbb{Z})$. For $\forall f(t) \in L^2(\mathbb{R})$, we have

$$f(t) = \sum_{j\in\mathbb{Z}}\sum_{k\in\mathbb{Z}} c_{j,k} \cdot \psi_{j,k}(t). \tag{42}$$

Specifically, the bijection relation is satisfied between variables $L^2(\mathbb{R})$ and $l^2(\mathbb{Z})$ in two spaces $f(t)$ and $\{h_n; n \in \mathbb{Z}\}$. Since $\left\{\psi_{j,k}(t), (j,k) \in \mathbb{Z}^2\right\}$ constitutes a set of orthogonal bases satisfying

$$\forall (j_1,k_1) \neq (j_2,k_2), \langle \psi_{j_1,k_1}, \psi_{j_2,k_2}\rangle = 0. \tag{43}$$

Further, we have

$$\begin{aligned}
\langle f(t), \psi_{m,n}(t)\rangle &= \left\langle \sum_{j\in\mathbb{Z}}\sum_{k\in\mathbb{Z}} c_{j,k}\cdot\psi_{j,k}(t), \psi_{m,n}(t)\right\rangle, \\
&= \sum_{j\in\mathbb{Z}}\sum_{k\in\mathbb{Z}} c_{j,k}\cdot\langle\psi_{j,k}(t), \psi_{m,n}(t)\rangle, \\
&= c_{m,n}\cdot\langle\psi_{m,n}(t), \psi_{m,n}(t)\rangle, \\
&= c_{m,n}\cdot|\psi_{m,n}(t)|^2.
\end{aligned} \tag{44}$$

Thus, the coefficients sequence can be uniquely determined by $c_{m,n} = \langle f(t), \psi_{m,n}(t)\rangle \,/\, |\psi_{m,n}(t)|^2$, and the bijection relation is satisfied between the variables $f(t)$ and $\{h_n; n \in \mathbb{Z}\}$, which completes the proof. □

Lemma B.1. *The sufficiently-necessary condition for orthonormal system: Defining the function $f(x) \in L^2(\mathbb{R})$, then the set*

$$\left\{f_{0,n} = 2^{0/2} f(2^0 t - n) = f(t-n); n \in \mathbb{Z}\right\} \tag{45}$$

forms the orthonormal system of $L^2(\mathbb{R})$, that is

$$\langle f(t-n), f(t-l)\rangle = \delta(n-l), \tag{46}$$

is sufficiently-necessary for

$$\sum_{k\in\mathbb{Z}}\left|\hat{f}(\omega + 2k\pi)\right|^2 = 1 \quad a.e.\omega \in \mathbb{R}. \tag{47}$$

In fact, Lemma B.1 is proved due to

$$\begin{aligned}
\langle f(t-n), f(t-l)\rangle &= \frac{1}{2\pi}\int_{\mathbb{R}} \hat{f}(\omega)e^{-in\omega}\cdot\overline{\left(\hat{f}(\omega)e^{-il\omega}\right)}d\omega, \\
&= \frac{1}{2\pi}\int_0^{2\pi}\sum_{k\in\mathbb{Z}}\left|\hat{f}(\omega+2k\pi)\right|^2\cdot e^{-i(n-l)\omega}d\omega.
\end{aligned} \tag{48}$$

Proof of Proposition 3. We define the functions $H(\omega)$ and $G(\omega)$ refer as the low-pass filter and band-pass filter based on $\psi(t) \in L^2(\mathbb{R})$, which determined by (38) and (41), respectively, and introduce the definition of matrix $M(\omega)$ as follows

$$M(\omega) = \begin{pmatrix} H(\omega) & G(\omega) \\ H(\omega+\pi) & G(\omega+\pi)\end{pmatrix}. \tag{49}$$

The function group $\left\{\psi_{0,k} = 2^{0/2}\psi(2^0 t - k) = \psi(t-k), k \in \mathbb{Z}\right\}$ forms the orthonormal basis of W_0, that is, the sufficient-necessary condition for $\psi(x)$ as the orthogonal wavelet is that $M(\omega)$ is the Unitary Matrix: $M^H(\omega)M(\omega) = M(\omega)M^H(\omega) = I, a.e.\omega \in \mathbb{R}$, where $M^H(\omega)$ is defined to be the conjugate transpose matrix of $M(\omega)$.

By the definition of the Unitary Matrix, $M(\omega)$ is the Unitary Matrix equivalent to

$$\begin{aligned}
|H(\omega)|^2 + |H(\omega+\pi)|^2 &= 1, a.e.\omega \in \mathbb{R}, \\
|G(\omega)|^2 + |G(\omega+\pi)|^2 &= 1, a.e.\omega \in \mathbb{R}, \\
H(\omega)\overline{G}(\omega) + H(\omega+\pi)\overline{G}(\omega+\pi) &= 0, a.e.\omega \in \mathbb{R}.
\end{aligned} \tag{50}$$

We define the spaces V_0 and W_0 as follows

$$\begin{aligned} V_0 &= closespan\left\{\phi(t-n), n \in \mathbb{Z}\right\}, \\ W_0 &= closespan\left\{\psi(t-n), n \in \mathbb{Z}\right\}. \end{aligned} \tag{51}$$

By the definitions of $H(\omega)$ and $G(\omega)$, Equation 50 is equivalent to

$$\begin{aligned} \sum_{k\in\mathbb{Z}} \left|\hat{\phi}(\omega + 2k\pi)\right|^2 &= 1, a.e. \omega \in \mathbb{R}, \\ \sum_{k\in\mathbb{Z}} \left|\hat{\psi}(\omega + 2k\pi)\right|^2 &= 1, a.e. \omega \in \mathbb{R}, \\ V_0 \perp W_0, a.e. \omega &\in \mathbb{R}. \end{aligned} \tag{52}$$

Because of Lemma B.1, the first two conditions are equivalent to

$$\begin{aligned} \langle \phi(t-n), \phi(t-l)\rangle &= \delta(n-l), \\ \langle \psi(t-n), \psi(t-l)\rangle &= \delta(n-l). \end{aligned} \tag{53}$$

In summary, the sufficient-necessary condition for $M(\omega)$ as the Unitary Matrix is that $\phi(t-n)$ and $\psi(t-n)$ form the standard orthogonal system of $L^2(\mathbb{R})$, respectively, and the two spaces formed by $\phi(t-n)$ and $\psi(t-n)$ are orthogonal. Considering the properties of V_j and W_j in section A, $\left\{\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k), (j,k) \in \mathbb{Z}^2\right\}$ constitutes the orthonormal basis for $L^2(\mathbb{R})$, and hence $\psi(t)$ is an orthogonal wavelet.

According to Proposition 3, by designing a specific functional relationship between $H(\omega)$ and $G(\omega)$, we can guarantee that $M(\omega)$ is the Unitary Matrix, and thus that $\psi(t)$ is the orthogonal wavelet. For instance, when $G(\omega) = e^{-i\omega}\overline{H}(\omega + \pi)$, it is easy to verify that $M(\omega)$ is the Unitary Matrix, when the frequency domain form of the wavelet function $\psi(t)$ is:

$$\hat{\psi}(\omega) = e^{-i\omega/2}\overline{H}(\pi + \omega/2) \cdot \hat{\phi}(\omega/2), \tag{54}$$

where the corresponding impulse response coefficient is

$$g_n = (-1)^{n-1}\overline{h}_{1-n}, n \in \mathbb{Z}, \tag{55}$$

Therefore, the time domain form of the wavelet function $\psi(t)$ is

$$\psi(t) = \sqrt{2}\sum_{n\in\mathbb{Z}} (-1)^{n-1}\overline{h}_{1-n}\phi(2t-n). \tag{56}$$

□

C Related Work of Tokenization

In the initial phase, the transformer-based model utlized the "point as token" tokenization strategy, which caused to two limitations: high computational complexity and serious information redundancy, Therefore, existing approaches have focused on reducing computational complexity by mitigating information redundancy.

For instance, Reformer Kitaev et al. [2020b] designed the locally sensitive hashing self-attention to reduce computational complexity. Informer Zhou et al. [2021] proposes the ProbSparse self-attention mechanism to efficiently replace the canonical self-attention. Pyraformer Liu et al. [2021] introduces the pyramidal attention module which reduces computational complexity by constraining the maximum length of the signal traversing paths. Autoformer Wu et al. [2021] designs the Auto-Correlation mechanism based on the series periodicity, which conducts the dependencies discovery and representation aggregation at the sub-series level. ETSformer Woo et al. [2022b] proposes the novel exponential smoothing attention and frequency attention to replace the self-attention mechanism in vanilla Transformers, thus improving both accuracy and efficiency. FEDformer Zhou et al. [2022a] avoids the high overhead of computation on the time domain by calculating the dependencies between individual bands in the frequency domain, while Fourier Transform has been used to ensure that individual bands have a global view.

D More analysis of Tokenization Strategy

D.1 Ideal tokenization generation strategy

Since TS datasets from different domains and sampling settings exhibit diverse periodic patterns, it is difficult to identify a unified sub-series span that matches all TS data domains. existing tokenization strategies are unable to establish

a unified framework to adapt to the potential TS data domains, and the pre-training phase is unable to adequately learn cross-domain representational information, which limits the generalization ability and scalability of the model. Therefore, The ideal tokenization generation strategies should be insensitive to the mathematical characteristics of different data domains. However, the patch span depends on the periodic pattern of the raw series

Thinking about the approach in NLP: the word is a high-level input data abstracted and generalized by the human brain, and existing subword algorithms Kudo and Richardson [2018], Sennrich et al. [2015] can disassemble the complex and lengthy word into a set of mutually independent and complementary units, which ensures that each token input to attention contains similar-size semantic information. This ensures that semantic information, which is essential for inference, is evenly distributed across subwords. However, the distribution of pattern information in the TS data is completely uncertain (e.g., randomly occurring anomalous oscillatory waveforms contain more information), which increases the inference difficulty of attention under the ***Point as token*** strategy. Therefore, the ideal tokenization strategy should play a role similar to feature programming, such that the generated tokens contain some pattern information from the original series (as semantic information in NLP). At the same time, it needs to ensure that the pattern information obtained from different TS domains is essentially isomorphic or similarly distributed, which is regarded as the key to activate the model cross-domains adaptability.

D.2 Challenges addressed by wave tokenization strategy

Establishing the unified Tokenization Paradigm between different TS domains has the following challenges: (1) TS data with diverse domain background knowledge tend to exhibit different characteristics; (2) Datasets from different sources may exhibit variations, even in the same domain background; (3) The same data instances can also face the challenge of cross-domain adaptation due to a priori features such as sampling rate.

The proposed ***Wave as Token*** strategy solves *challenge-1* through the designed shared embedding space. In addition, the designed strategy ensures that arbitrary series data can be recoded to equal-length groups of tokens (number of tokens equal to series length), thus solving *challenge-2*. This ensures that no hyperparameter setting tricks are utilized in the model to adapt to potential data domains, even if these domains are completely different from each other in terms of structural features (e.g., channel number, sequence length) are completely different. Finally, we solve *challenge-3* by ensuring that each token contains TS pattern information within a localized window through the finite-length basis functions and bridges the difference in the distribution of pattern information from different domains within the K-dimensional space formed by basis functions, where K is the size of the proposed wavebook.

E Implementation Details

Table 14: Description of datasets in forecasting and imputation tasks. The dataset size is organized in (Train, Validation, Test).

Tasks	Dataset	Dim	Series Length	Dataset Size	Information (Frequency)
Forecasting	ETTm1, ETTm2	7	{96, 192, 336, 720}	(34465, 11521, 11521)	Electricity (15 mins)
	ETTh1, ETTh2	7	{96, 192, 336, 720}	(8545, 2881, 2881)	Electricity (15 mins)
	Electricity	321	{96, 192, 336, 720}	(18317, 2633, 5261)	Electricity (Hourly)
	Traffic	862	{96, 192, 336, 720}	(12185, 1757, 3509)	Transportation (Hourly)
	Weather	21	{96, 192, 336, 720}	(36792, 5271, 10540)	Weather (10 mins)
	Exchange	8	{96, 192, 336, 720}	(5120, 665, 1422)	Exchange rate (Daily)
	Sunspot	1	{96, 192, 336, 720}	(44354, 14785, 14785)	Nature (Daily)
	RiverFlow	1	{96, 192, 336, 720}	(14244, 4748, 4748)	Nature (Daily)
	SolarPower	1	{96, 192, 336, 720}	(4438333, 1479444, 1479444)	Energy (4 seconds)
Imputation	ETTm1, ETTm2	7	96	(34465, 11521, 11521)	Electricity (15 mins)
	ETTh1, ETTh2	7	96	(8545, 2881, 2881)	Electricity (15 mins)
	Electricity	321	96	(18317, 2633, 5261)	Electricity (15 mins)
	Weather	21	96	(36792, 5271, 10540)	Weather (10 mins)

Table 15: Description of datasets in the classification task. The dataset size is organized in (Train, Validation, Test).

Tasks	Dataset	Class Number	Series Length	Dataset Size	Information (Sample Rate)
Classification	BeetleFly	2	512	(20, 20, 20)	Image
	BME	3	128	(30, 150, 150)	Simulated
	CBF	3	128	(30, 900, 900)	Simulated
	Chinatown	2	24	(20, 343, 343)	Traffic
	ChlorineConcentration	3	166	(467, 3840, 3840)	Sensor
	DistalPhalanxTW	6	80	(400, 139, 139)	Image
	ECG200	2	96	(100, 100, 100)	ECG
	ECG5000	5	140	(500, 4500, 4500)	ECG
	ElectricDevices	7	96	(8926, 7711, 7711)	Device
	FaceAll	14	131	(560, 1690, 1690)	Image
	FaceFour	4	350	(24, 88, 88)	Image
	FacesUCR	14	131	(200, 2050, 2050)	Image
	FiftyWords	50	270	(450, 455, 455)	Image
	GunPointAgeSpan	2	150	(135, 316, 316)	Motion
	GunPointMaleVersusFemale	2	150	(135, 316, 316)	Motion
	GunPointOldVersusYoung	2	150	(135, 316, 316)	Motion
	GunPoint	2	150	(50, 150, 150)	Motion
	InsectEPGSmallTrain	3	601	(17, 249, 249)	Sensor
	InsectWingbeatSound	11	256	(220, 1980, 1980)	Sensor
	ItalyPowerDemand	2	24	(67, 1029, 1029)	Sensor
	MedicalImages	10	99	(381, 760, 760)	Image
	MiddlePhalanxTW	6	80	(399, 154, 154)	Image
	MoteStrain	2	84	(20, 1252, 1252)	Sensor
	Plane	7	144	(105, 105, 105)	Sensor
	ProximalPhalanxTW	6	80	(400, 205, 205)	Image
	SonyAIBORobotSurface1	2	70	(20, 601, 601)	Sensor
	SonyAIBORobotSurface2	2	65	(27, 953, 953)	Sensor
	SwedishLeaf	15	128	(500, 625, 625)	Image
	SyntheticControl	6	60	(300, 300, 300)	Simulated
	ToeSegmentation2	2	343	(36, 130, 130)	Motion
	Trace	4	275	(100, 100, 100)	Sensor
	UMD	3	150	(36, 144, 144)	Simulated
	UWaveGestureLibraryY	8	315	(896, 3582, 3582)	Motion
	Wafer	2	152	(1000, 6164, 6164)	Sensor
	WordSynonyms	25	270	(267, 638, 638)	Image

We provide the dataset descriptions and experiment configurations in Table 14, Table 15 and Table 3. Besides, The specific description of the symbols involved in the method is shown in Table 2. All experiments are repeated three times, implemented in PyTorch Paszke et al. [2019] and conducted on a single Tesla V100 SXM2 32GB GPU.

Our method is trained with the L2 Loss, using the ADAM optimizer with an initial learning rate of 10^{-4}, and Batch size is set in $16 \rightarrow 64$. The training process is early stopped after three epochs (patience=3) if there is no loss degradation on the valid set. The mean square error (MSE) and mean absolute error (MAE) are used as metrics in forecasting and imputation tasks. Besides, the accuracy (Acc), precision (Pre), recall (Rec), and f1-score (F1) are used as metrics in the classification task. For a fair comparison, we fix the input length to 336 for all datasets. By default, the proposed **WQ4TS** contains $5 \rightarrow 10$ **EncoderLayers**. All the baselines that we reproduced are implemented based on configurations of the original paper or their official code. For a fair comparison, we design the same input embedding and final prediction layer for all base models.

Besides, the datasets that were utilized in the forecasting, imputation, and classification tasks are described in detail below: (1) **ETT (ETTh1, ETTh2, ETTm1, and ETTm2)**[11] Zhou et al. [2021] contains six power load features and oil temperature used for monitoring electricity transformers. ETT involves four subsets. ETTm1 and ETTm2 are recorded at 15-minute intervals, while ETTh1 and ETTh2 are recorded hourly. (2) **Weather**[12] contains 21 meteorological indicators, such as temperature, humidity, and precipitation, which are recorded every 10 minutes in the year 2020. (3) **Electricity**[13] comprises hourly power consumption of 321 clients from 2012 to 2014. (4) **Traffic**[14] reports the number of vehicles loaded on all 862 roads at each moment in time. (5) **Sunspot**[15] records observations of sunspots for long-term monitoring, consisting of 73924 timesteps. (6) **River Flow**[16] reports th daily river flow, consisting of 23741 timesteps. (7) **Solar Power**[17] contains a single long daily time series representing the wind power production in MW recorded every 4 seconds starting from 2019. (8) **UCR archive**[18] as the well-known time series classification repository, where representative **35 datasets** are selected to validate the performance of the proposed approach.

F Full experiment results

F.1 Forecasting task

In this section, to verify that the proposed WQ4TS has the potential to be a foundational model in time series, we first conduct sufficient experiments under the condition of **full-data**, as shown in Table 16.

Besides, to demonstrate that the model has excellent data efficiency and powerful cross-domain adaptability, the forecasting performance under the **few-shot** and **single-domain zero-shot** conditions are shown in Table 17-18 and Table 19, respectively.

Notably, to show that the proposed ***wave as token*** strategy can establish underlying connections between diverse data domains and thus activate the generalization capability of the backbone network, Table 20 compares the performance of **single-domain adapting** and **multi-domain adapting** on the zero-shot task. The results show that the proposed strategy can alleviate the negative migration phenomenon in the time series domain.

F.2 Imputation

Same as the forecasting task, the performance of the imputation task under **full-data** and **zero-shot** conditions are shown in Table 21 and Table 22, respectively.

F.3 Classification

To validate the performance of the **wavebook** strategy on the classification task, complete experimental results contain the three most representative approaches (OneFitsAll, PatchTST, and TimesNet) and proposed WQ4TS on all 35 classification datasets of UCR archive Dau et al. [2018], as shown in Table 23.

[11] https://github.com/zhouhaoyi/Informer2020
[12] https://www.bgc-jena.mpg.de/wetter/
[13] https://archive.ics.uci.edu/dataset/321/electricity
[14] http://pems.dot.ca.gov
[15] https://www.sidc.be/SILSO/newdataset
[16] http://www.jenvstat.org/v04/i11
[17] https://zenodo.org/records/4656032
[18] https://www.cs.ucr.edu/~eamonn/time_series_data_2018

Table 16: Comparison of the complete performance with diverse prediction lengths ($\{96, 192, 336, 720\}$) on **full-data forecasting** task, where *Sunspot*, *RiverFlow* and *SolarPower* are novel datasets that we introduced.

Models		**WQ4TS (Ours)**		GPT4TS 2023		DLinear 2023		PatchTST 2023		TimesNet 2023		FEDformer 2022a		Autoformer 2022		Stationary 2021		ETSformer 2021		Informer 2021	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.324	**0.342**	0.292	0.346	0.299	0.343	**0.290**	0.342	0.338	0.375	0.379	0.419	0.505	0.475	0.386	0.398	0.375	0.398	0.672	0.571
	192	0.349	**0.357**	0.332	0.372	0.335	0.365	**0.332**	0.369	0.374	0.387	0.426	0.441	0.553	0.496	0.459	0.444	0.408	0.410	0.795	0.669
	336	0.385	**0.375**	0.366	0.394	0.369	0.386	**0.366**	0.392	0.410	0.411	0.445	0.459	0.621	0.537	0.495	0.464	0.435	0.428	1.212	0.871
	720	**0.412**	**0.402**	0.417	0.421	0.425	0.421	0.416	0.420	0.478	0.450	0.543	0.490	0.671	0.561	0.585	0.516	0.499	0.462	1.166	0.823
	Avg	0.368	**0.369**	0.352	0.383	0.357	0.378	**0.351**	0.380	0.400	0.406	0.448	0.452	0.588	0.517	0.481	0.456	0.429	0.425	0.961	0.734
ETTm2	96	**0.154**	**0.239**	0.173	0.262	0.167	0.269	0.165	0.255	0.187	0.267	0.203	0.287	0.255	0.339	0.192	0.274	0.189	0.280	0.365	0.453
	192	**0.212**	**0.274**	0.229	0.301	0.224	0.303	0.220	0.292	0.249	0.309	0.269	0.328	0.281	0.340	0.280	0.339	0.253	0.319	0.533	0.563
	336	**0.267**	**0.309**	0.286	0.341	0.281	0.342	0.274	0.329	0.321	0.351	0.325	0.366	0.339	0.372	0.334	0.361	0.314	0.357	1.363	0.887
	720	**0.357**	**0.359**	0.378	0.401	0.397	0.421	0.362	0.385	0.408	0.403	0.421	0.415	0.433	0.432	0.417	0.413	0.414	0.413	3.379	1.338
	Avg	**0.247**	**0.295**	0.266	0.326	0.267	0.333	0.255	0.315	0.291	0.333	0.305	0.349	0.327	0.371	0.306	0.347	0.293	0.342	1.410	0.810
ETTh1	96	**0.363**	**0.388**	0.376	0.397	0.375	0.399	0.370	0.399	0.384	0.402	0.376	0.419	0.449	0.459	0.513	0.491	0.494	0.479	0.865	0.713
	192	**0.386**	**0.409**	0.416	0.418	0.405	0.416	0.413	0.421	0.436	0.429	0.420	0.448	0.500	0.482	0.534	0.504	0.538	0.504	1.008	0.482
	336	**0.414**	**0.413**	0.442	0.433	0.439	0.443	0.422	0.436	0.491	0.469	0.459	0.465	0.521	0.496	0.588	0.535	0.574	0.521	1.107	0.809
	720	0.467	0.460	0.477	**0.456**	0.472	0.490	**0.447**	0.466	0.521	0.500	0.506	0.507	0.514	0.512	0.643	0.616	0.562	0.535	1.181	0.865
	Avg	**0.407**	**0.418**	0.427	0.426	0.422	0.437	0.413	0.430	0.458	0.450	0.440	0.460	0.496	0.487	0.570	0.537	0.542	0.510	1.040	0.795
ETTh2	96	0.278	**0.326**	0.285	0.342	0.289	0.353	**0.274**	0.336	0.340	0.374	0.358	0.397	0.346	0.388	0.476	0.458	0.340	0.391	3.755	1.525
	192	0.358	**0.373**	0.354	0.389	0.383	0.418	**0.339**	0.379	0.402	0.414	0.429	0.439	0.456	0.452	0.512	0.493	0.430	0.439	5.602	1.931
	336	0.372	0.395	0.373	0.407	0.448	0.465	**0.329**	**0.380**	0.452	0.452	0.496	0.487	0.482	0.486	0.552	0.551	0.485	0.479	4.721	1.835
	720	0.381	**0.398**	0.406	0.441	0.605	0.551	**0.379**	0.422	0.462	0.468	0.463	0.474	0.515	0.511	0.562	0.560	0.500	0.497	3.647	1.625
	Avg	0.347	**0.373**	0.354	0.394	0.431	0.446	**0.330**	0.379	0.414	0.427	0.437	0.449	0.450	0.459	0.526	0.516	0.439	0.452	4.431	1.729
Electricity	96	0.132	**0.205**	0.139	0.238	0.140	0.237	**0.129**	0.222	0.168	0.272	0.193	0.308	0.201	0.317	0.169	0.273	0.187	0.304	0.274	0.368
	192	**0.142**	**0.231**	0.153	0.251	0.153	0.249	0.157	0.240	0.184	0.289	0.201	0.315	0.222	0.334	0.182	0.286	0.199	0.315	0.296	0.386
	336	**0.156**	0.263	0.169	0.266	0.169	0.267	0.163	**0.259**	0.198	0.300	0.214	0.329	0.231	0.338	0.200	0.304	0.212	0.329	0.300	0.394
	720	**0.187**	**0.289**	0.206	0.297	0.203	0.301	0.197	0.290	0.220	0.320	0.246	0.355	0.254	0.361	0.222	0.321	0.233	0.345	0.373	0.439
	Avg	**0.154**	**0.247**	0.167	0.263	0.166	0.263	0.161	0.252	0.192	0.295	0.214	0.295	0.227	0.327	0.193	0.338	0.208	0.296	0.311	0.397
Traffic	96	**0.349**	0.264	0.388	0.282	0.410	0.282	0.360	**0.249**	0.593	0.321	0.587	0.366	0.613	0.388	0.612	0.338	0.607	0.392	0.719	0.391
	192	**0.373**	0.280	0.407	0.290	0.423	0.287	0.379	**0.256**	0.617	0.336	0.604	0.373	0.616	0.382	0.613	0.340	0.621	0.399	0.696	0.379
	336	**0.379**	0.291	0.412	0.294	0.436	0.296	0.392	**0.264**	0.629	0.336	0.621	0.383	0.622	0.337	0.618	0.328	0.622	0.396	0.777	0.420
	720	**0.415**	0.303	0.450	0.312	0.466	0.315	0.432	**0.286**	0.640	0.350	0.626	0.382	0.660	0.408	0.653	0.355	0.632	0.396	0.864	0.472
	Avg	**0.379**	0.286	0.414	0.294	0.433	0.295	0.390	**0.263**	0.620	0.336	0.610	0.376	0.628	0.379	0.624	0.340	0.621	0.396	0.764	0.416
Weather	96	**0.146**	**0.197**	0.162	0.212	0.176	0.237	0.149	0.198	0.172	0.220	0.217	0.296	0.266	0.336	0.173	0.223	0.197	0.281	0.300	0.384
	192	**0.188**	**0.227**	0.204	0.248	0.220	0.282	0.194	0.241	0.219	0.261	0.276	0.336	0.307	0.367	0.245	0.285	0.237	0.312	0.598	0.544
	336	**0.240**	**0.256**	0.254	0.286	0.265	0.319	0.245	0.282	0.280	0.306	0.339	0.280	0.259	0.395	0.321	0.338	0.298	0.353	0.578	0.523
	720	**0.289**	**0.301**	0.326	0.337	0.333	0.362	0.314	0.334	0.365	0.359	0.403	0.428	0.419	0.428	0.414	0.410	0.352	0.288	1.059	0.741
	Avg	**0.216**	**0.246**	0.237	0.270	0.248	0.300	0.225	0.264	0.259	0.287	0.309	0.360	0.338	0.382	0.288	0.314	0.271	0.334	0.634	0.548
Sunspot	96	**0.298**	**0.379**	0.329	0.411	0.354	0.433	0.321	0.401	0.324	0.402	0.332	0.419	0.333	0.414	0.329	0.410	0.362	0.449	0.431	0.512
	192	**0.338**	0.407	0.368	0.438	0.410	0.473	0.362	0.430	0.371	0.437	0.385	0.455	0.384	0.449	0.386	0.456	0.407	0.479	0.459	0.534
	336	**0.401**	**0.447**	0.425	0.470	0.585	0.586	0.432	0.477	0.433	0.475	0.467	0.495	0.436	0.484	0.451	0.491	0.472	0.537	0.561	0.612
	720	**0.543**	**0.534**	0.657	0.589	0.755	0.726	0.665	0.595	0.673	0.599	0.725	0.624	0.678	0.605	0.681	0.626	0.682	0.667	0.784	0.759
	Avg	**0.395**	**0.442**	0.445	0.477	0.526	0.554	0.446	0.476	0.450	0.478	0.477	0.498	0.458	0.488	0.462	0.496	0.481	0.533	0.559	0.604
RiverFlow	96	**0.958**	**0.477**	1.194	0.539	1.110	0.570	1.200	0.611	1.197	0.606	1.133	0.598	1.183	0.602	1.120	0.525	1.206	0.649	1.246	0.701
	192	**0.979**	**0.496**	1.292	0.569	1.142	0.601	1.304	0.687	1.334	0.688	1.143	0.611	1.349	0.711	1.125	0.556	1.263	0.673	1.358	0.783
	336	**0.998**	**0.501**	1.201	0.551	1.168	0.654	1.207	0.662	1.220	0.655	1.163	0.614	1.217	0.645	1.147	0.634	1.251	0.654	1.319	0.723
	720	**1.082**	**0.513**	1.187	0.544	1.165	0.593	1.219	0.673	1.238	0.656	1.116	0.561	1.235	0.647	1.156	0.581	1.279	0.659	1.325	0.739
	Avg	**1.004**	**0.497**	1.218	0.551	1.146	0.605	1.233	0.658	1.247	0.651	1.139	0.596	1.246	0.651	1.137	0.574	1.250	0.659	1.312	0.737
SolarPower	96	**0.010**	**0.031**	0.012	0.034	0.015	0.047	0.014	0.042	0.027	0.069	0.015	0.048	0.037	0.081	0.023	0.057	0.023	0.079	0.025	0.068
	192	**0.020**	**0.048**	0.023	0.054	0.030	0.069	0.033	0.072	0.056	0.095	0.032	0.071	0.065	0.107	0.041	0.083	0.045	0.102	0.052	0.091
	336	**0.032**	**0.067**	0.037	0.074	0.046	0.098	0.039	0.087	0.082	0.176	0.047	0.102	0.097	0.198	0.063	0.140	0.083	0.176	0.076	0.164
	720	**0.063**	**0.104**	0.070	0.126	0.091	0.157	0.086	0.143	0.163	0.257	0.090	0.153	0.175	0.281	0.127	0.198	0.156	0.242	0.160	0.259
	Avg	**0.031**	**0.063**	0.036	0.072	0.046	0.093	0.043	0.086	0.082	0.149	0.046	0.094	0.093	0.167	0.064	0.120	0.076	0.149	0.078	0.147
1st Count		**78**		1		0		21		0		0		0		0		0		0	

Table 17: Comparison of the complete performance with diverse prediction lengths ($\{96, 192, 336, 720\}$) on **few-shot forecasting** task, where all samples of trainset are only partially available (5%) in the training phase, where '-' denotes that the limited length of series cannot constitute a complete training set.

Scenarios		**WQ4TS**		GPT4TS		DLinear		PatchTST		TimesNet		FEDformer		Autoformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.357	**0.350**	0.386	0.405	**0.332**	0.374	0.399	0.414	0.606	0.518	0.628	0.544	0.726	0.578
	192	0.363	**0.368**	0.440	0.438	**0.358**	0.390	0.441	0.436	0.681	0.539	0.666	0.566	0.750	0.591
	336	**0.401**	**0.385**	0.485	0.459	0.402	0.416	0.499	0.467	0.786	0.597	0.807	0.628	0.851	0.659
	720	**0.457**	**0.424**	0.577	0.499	0.511	0.489	0.767	0.587	0.796	0.593	0.822	0.633	0.857	0.655
	avg	**0.394**	**0.381**	0.472	0.450	0.400	0.417	0.526	0.476	0.717	0.561	0.730	0.592	0.796	0.620
ETTm2	96	**0.175**	**0.251**	0.199	0.280	0.236	0.326	0.206	0.288	0.220	0.299	0.229	0.320	0.232	0.322
	192	**0.229**	**0.281**	0.256	0.316	0.306	0.373	0.264	0.324	0.311	0.361	0.394	0.361	0.291	0.357
	336	**0.282**	**0.314**	0.318	0.353	0.380	0.423	0.334	0.367	0.338	0.366	0.378	0.427	0.478	0.517
	720	**0.381**	**0.377**	0.460	0.436	0.674	0.583	0.454	0.432	0.509	0.465	0.523	0.510	0.553	0.538
	avg	**0.267**	**0.306**	0.308	0.346	0.399	0.426	0.314	0.352	0.344	0.372	0.381	0.404	0.388	0.433
ETTh1	96	**0.509**	**0.440**	0.543	0.506	0.547	0.503	0.557	0.519	0.892	0.625	0.593	0.529	0.681	0.570
	192	**0.527**	**0.452**	0.748	0.580	0.720	0.604	0.711	0.570	0.940	0.665	0.652	0.563	0.725	0.602
	336	**0.528**	**0.457**	0.754	0.595	0.984	0.727	0.816	0.619	0.945	0.653	0.731	0.594	0.761	0.624
	720	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	avg	**0.521**	**0.449**	0.681	0.560	0.750	0.611	0.694	0.569	0.925	0.647	0.658	0.562	0.722	0.598
ETTh2	96	**0.337**	**0.351**	0.376	0.421	0.442	0.456	0.401	0.421	0.409	0.420	0.390	0.424	0.428	0.468
	192	**0.378**	**0.381**	0.418	0.441	0.617	0.610	0.452	0.455	0.483	0.464	0.457	0.465	0.496	0.504
	336	**0.404**	**0.410**	0.408	0.439	1.424	0.849	0.464	0.469	0.499	0.479	0.477	0.483	0.486	0.496
	720	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	avg	**0.373**	**0.381**	0.400	0.433	0.827	0.615	0.439	0.448	0.463	0.454	0.441	0.457	0.470	0.489
Weather	96	**0.163**	**0.209**	0.175	0.230	0.184	0.242	0.171	0.224	0.207	0.253	0.229	0.309	0.227	0.299
	192	**0.200**	**0.242**	0.227	0.276	0.228	0.283	0.230	0.277	0.272	0.307	0.265	0.317	0.278	0.333
	336	**0.252**	**0.269**	0.286	0.322	0.279	0.322	0.294	0.326	0.313	0.328	0.353	0.392	0.351	0.393
	720	**0.315**	**0.327**	0.366	0.379	0.364	0.388	0.384	0.387	0.400	0.385	0.391	0.394	0.387	0.389
	avg	**0.233**	**0.262**	0.263	0.301	0.263	0.308	0.269	0.303	0.298	0.318	0.309	0.353	0.310	0.353

In the main body of the experiments on **few-shot classification** task, *Scenario-i: Source-i→Traget-i* ($i \in [0, .., 7]$) are utilized to indicates the cross-domain adaptation scenarios between different TS domains, and the details of scenarios are summarized in Table 12. Besides, the completed **few-shot classification** on cross-domain adaptation is demonstrated in Table 24, 25, 26, and 27, which includes four metrics: accuracy (Acc), precision (Pre), recall (Rec), and F1 score (F1).

F.4 Ablation

As shown in table 28, when the *wave as token* strategy is removed from the Transformer-based model (**w/o TP**), there demonstrates a significant performance degradation. Similar results are found for both ETTh1 and ETTm1 datasets, which proves the effectiveness of our approach.

Table 18: Comparison of the complete performance with diverse prediction lengths ($\{96, 192, 336, 720\}$) on **few-shot forecasting** task, where all samples of trainset are only partially available (10%) in the training phase.)

Scenarios		**WQ4TS**		GPT4TS		DLinear		PatchTST		TimesNet		FEDformer		Autoformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.337**	**0.350**	0.390	0.404	0.352	0.392	0.410	0.419	0.583	0.501	0.578	0.518	0.774	0.614
	192	**0.364**	**0.363**	0.429	0.423	0.382	0.412	0.437	0.434	0.630	0.528	0.617	0.546	0.754	0.592
	336	**0.389**	**0.388**	0.469	0.439	0.419	0.434	0.476	0.454	0.725	0.568	0.998	0.775	0.869	0.677
	720	**0.441**	**0.417**	0.569	0.498	0.490	0.477	0.681	0.556	0.769	0.549	0.693	0.579	0.810	0.630
	avg	**0.383**	**0.380**	0.464	0.441	0.411	0.429	0.501	0.466	0.677	0.537	0.722	0.605	0.802	0.628
ETTm2	96	**0.170**	**0.246**	0.188	0.269	0.213	0.303	0.191	0.274	0.212	0.285	0.291	0.399	0.352	0.454
	192	**0.226**	**0.279**	0.251	0.309	0.278	0.345	0.252	0.317	0.270	0.323	0.307	0.379	0.694	0.691
	336	**0.279**	**0.312**	0.307	0.346	0.338	0.385	0.306	0.353	0.323	0.353	0.543	0.559	2.408	1.407
	720	**0.379**	**0.375**	0.426	0.417	0.436	0.440	0.433	0.427	0.474	0.449	0.712	0.614	1.913	1.166
	avg	**0.264**	**0.303**	0.293	0.335	0.316	0.368	0.296	0.343	0.320	0.353	0.463	0.488	1.342	0.930
ETTh1	96	**0.432**	**0.409**	0.458	0.456	0.492	0.495	0.516	0.485	0.861	0.628	0.512	0.499	0.613	0.552
	192	**0.456**	**0.428**	0.570	0.516	0.565	0.538	0.598	0.524	0.797	0.593	0.624	0.555	0.722	0.598
	336	**0.489**	**0.453**	0.608	0.535	0.721	0.622	0.657	0.550	0.941	0.648	0.691	0.574	0.750	0.619
	720	**0.531**	**0.502**	0.725	0.591	0.986	0.743	0.762	0.610	0.877	0.641	0.728	0.614	0.721	0.616
	avg	**0.477**	**0.448**	0.590	0.525	0.691	0.600	0.633	0.542	0.869	0.628	0.639	0.561	0.702	0.696
ETTh2	96	**0.308**	**0.350**	0.331	0.374	0.357	0.411	0.353	0.389	0.378	0.409	0.382	0.416	0.413	0.451
	192	**0.376**	**0.385**	0.402	0.411	0.569	0.519	0.403	0.414	0.490	0.467	0.478	0.474	0.474	0.477
	336	0.382	**0.406**	**0.406**	0.433	0.671	0.572	0.426	0.441	0.537	0.494	0.504	0.501	0.547	0.543
	720	**0.408**	**0.413**	0.449	0.464	0.824	0.648	0.477	0.480	0.510	0.491	0.499	0.509	0.516	0.523
	avg	**0.369**	**0.389**	0.397	0.421	0.605	0.538	0.415	0.431	0.479	0.465	0.466	0.475	0.488	0.499
Weather	96	**0.155**	**0.205**	0.163	0.215	0.171	0.224	0.165	0.215	0.184	0.230	0.188	0.253	0.221	0.297
	192	**0.194**	**0.237**	0.210	0.254	0.215	0.263	0.210	0.257	0.245	0.283	0.250	0.304	0.270	0.322
	336	**0.249**	**0.261**	0.256	0.292	0.258	0.299	0.259	0.297	0.305	0.321	0.312	0.346	0.320	0.351
	720	**0.310**	**0.325**	0.321	0.339	0.320	0.346	0.332	0.346	0.381	0.371	0.387	0.393	0.390	0.396
	avg	**0.227**	**0.257**	0.238	0.275	0.241	0.283	0.242	0.279	0.279	0.301	0.284	0.324	0.300	0.342

Table 19: Comparison of the complete performance with diverse prediction lengths ($\{96, 192, 336, 720\}$) on **zero-shot forecasting** task. Where $Source \rightarrow Traget$ indicates that the model is first pre-trained on the single train set of the *SourceDomain*, subsequently, the model parameters are frozen and predicted on the test set of the *TargetDomain*.

Scenarios		**WQ4TS**		GPT4TS		DLinear		PatchTST		TimesNet		FEDformer		Autoformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm2 ↓ ETTm1	96	**0.400**	**0.407**	0.706	0.543	0.487	0.455	0.481	0.441	0.749	0.557	0.691	0.548	0.696	0.551
	192	**0.415**	**0.417**	0.753	0.566	0.498	0.461	0.542	0.473	0.908	0.613	0.710	0.558	0.716	0.562
	336	**0.442**	**0.434**	0.787	0.581	0.520	0.477	0.599	0.520	0.901	0.613	0.723	0.567	0.726	0.569
	720	**0.481**	**0.459**	0.914	0.627	0.558	0.501	0.760	0.597	0.871	0.615	0.747	0.581	0.748	0.582
	avg	**0.434**	**0.429**	0.790	0.579	0.516	0.473	0.596	0.508	0.857	0.599	0.718	0.564	0.722	0.566
ETTm1 ↓ ETTm2	96	**0.191**	**0.263**	0.231	0.308	0.260	0.346	0.224	0.299	0.259	0.327	0.225	0.305	0.233	0.315
	192	**0.254**	**0.303**	0.307	0.349	0.309	0.380	0.286	0.338	0.307	0.358	0.285	0.339	0.290	0.348
	336	**0.312**	**0.340**	0.372	0.388	0.378	0.426	0.345	0.375	0.372	0.394	0.339	0.371	0.340	0.374
	720	**0.413**	**0.399**	0.457	0.432	0.491	0.489	0.443	0.430	0.489	0.456	0.437	0.426	0.435	0.423
	avg	**0.293**	**0.326**	0.342	0.369	0.360	0.410	0.325	0.361	0.357	0.384	0.321	0.360	0.325	0.365
ETTh2 ↓ ETTh1	96	**0.466**	**0.460**	0.712	0.562	0.527	0.480	0.545	0.491	0.834	0.608	0.721	0.573	0.709	0.576
	192	**0.505**	**0.483**	0.762	0.595	0.587	0.521	0.642	0.549	0.958	0.637	0.751	0.594	0.749	0.593
	336	**0.535**	**0.501**	0.815	0.623	0.671	0.554	0.630	0.541	0.842	0.606	0.765	0.611	0.739	0.593
	720	**0.543**	**0.529**	0.830	0.637	0.652	0.572	0.646	0.568	1.047	0.689	0.747	0.616	0.742	0.611
	avg	**0.512**	**0.493**	0.780	0.604	0.609	0.532	0.616	0.537	0.920	0.635	0.746	0.598	0.735	0.593
ETTh1 ↓ ETTh2	96	**0.297**	**0.343**	0.357	0.390	0.327	0.387	0.335	0.388	0.388	0.404	0.372	0.415	0.383	0.419
	192	**0.395**	**0.403**	0.424	0.424	0.444	0.459	0.419	0.438	0.437	0.433	0.452	0.462	0.451	0.456
	336	**0.418**	0.429	0.453	0.449	0.513	0.510	0.455	0.471	0.484	0.465	0.481	0.489	0.486	0.489
	720	**0.428**	**0.446**	0.447	0.458	0.626	0.576	0.456	0.481	0.462	0.464	0.471	0.488	0.460	0.471
	avg	**0.385**	**0.405**	0.420	0.430	0.478	0.483	0.416	0.444	0.443	0.442	0.444	0.463	0.445	0.459
RiverFlow ↓ Exchange	96	**0.090**	**0.220**	0.131	0.271	0.278	0.342	0.098	0.232	0.138	0.286	0.520	0.583	0.530	0.571
	192	**0.198**	**0.331**	0.228	0.359	0.435	0.438	0.197	0.334	0.245	0.372	0.668	0.660	0.527	0.587
	336	**0.356**	**0.435**	0.400	0.477	0.533	0.522	0.357	0.444	0.407	0.485	0.898	0.760	0.898	0.774
	720	**0.879**	**0.711**	1.099	0.855	1.092	0.846	1.032	0.822	1.195	0.888	1.683	1.057	1.426	1.022
	avg	**0.381**	**0.424**	0.464	0.491	0.585	0.537	0.421	0.458	0.497	0.508	0.942	0.765	0.845	0.739
Sunspot ↓ Weather	96	**0.181**	**0.235**	0.198	0.253	0.200	0.268	0.186	0.242	0.252	0.281	0.669	0.621	0.589	0.581
	192	**0.226**	**0.269**	0.241	0.283	0.242	0.298	0.235	0.279	0.341	0.343	0.700	0.632	0.543	0.522
	336	**0.275**	**0.301**	0.283	0.310	0.282	0.323	0.280	0.308	0.302	0.321	0.704	0.633	0.467	0.461
	720	**0.326**	**0.341**	0.333	0.343	0.329	0.353	0.352	0.358	0.347	0.354	0.746	0.651	0.435	0.440
	avg	**0.254**	**0.286**	0.264	0.297	0.263	0.310	0.263	0.297	0.311	0.325	0.705	0.634	0.509	0.501

Table 20: Comparison of the complete performance with diverse prediction lengths on **zero-shot forecasting** task, where $Source \rightarrow Traget$ indicates that the model is first pre-trained uniformly on all train sets from multiple *SourceDomains*, subsequently, the model parameters are frozen and predicted on the test set of the *TargetDomain*.

Scenarios		**Zero-shot**								**Full-data**				**Few-shot**			
Models		**WQ4TS**								**OneFitsAll**		**PatchTST**		**OneFitsAll**		**PatchTST**	
SourceData		ETT{m2,h1,h2}		ETTm2		ETTh1		ETTh2		ETTm1		ETTm1		ETTm1		ETTm1	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
SourceData ↓ ETTm1	96	**0.379**	**0.398**	0.400	0.407	0.647	0.512	0.733	0.533	0.292	0.346	**0.290**	**0.342**	0.386	0.405	0.399	0.414
	192	**0.389**	**0.404**	0.415	0.417	0.681	0.531	0.740	0.546	0.332	0.372	**0.332**	**0.369**	0.440	0.438	0.441	0.436
	336	**0.414**	**0.415**	0.442	0.434	0.673	0.535	0.742	0.553	0.366	0.394	**0.366**	**0.392**	0.485	0.459	0.499	0.467
	720	**0.461**	**0.446**	0.481	0.459	0.728	0.543	0.751	0.570	0.417	0.421	**0.416**	**0.420**	0.577	0.499	0.767	0.587
	avg	**0.411**	**0.416**	0.434	0.429	0.682	0.742	0.742	0.802	0.352	0.383	**0.351**	**0.380**	0.472	0.450	0.526	0.476
SourceData		ETT{m1,h1,h2}		ETTm1		ETTh1		ETTh2		ETTm2		ETTm2		ETTm2		ETTm2	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MAE	MSE
SourceData ↓ ETTm2	96	**0.177**	**0.258**	0.191	0.263	0.214	0.299	0.215	0.301	0.173	0.262	**0.165**	**0.255**	0.199	0.280	0.206	0.288
	192	**0.242**	**0.296**	0.254	0.303	0.278	0.338	0.280	0.341	0.229	0.301	**0.220**	**0.292**	0.256	0.316	0.264	0.324
	336	**0.300**	**0.332**	0.312	0.340	0.331	0.368	0.338	0.376	0.286	0.341	**0.274**	**0.329**	0.318	0.353	0.334	0.367
	720	**0.400**	**0.385**	0.413	0.399	0.439	0.430	0.432	0.422	0.378	0.401	**0.362**	**0.385**	0.460	0.436	0.454	0.432
	avg	**0.280**	**0.315**	0.292	0.326	0.316	0.359	0.316	0.360	0.266	0.326	**0.255**	**0.315**	0.308	0.346	0.314	0.352
SourceData		ETT{m1,m2,h2}		ETTm1		ETTm2		ETTh2		ETTh1		ETTh1		ETTh1		ETTh1	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MAE	MSE
SourceData ↓ ETTh1	96	**0.438**	**0.419**	0.466	0.460	0.488	0.469	0.527	0.480	0.376	**0.397**	**0.370**	0.399	0.543	0.506	0.557	0.519
	192	**0.449**	**0.439**	0.505	0.483	0.532	0.492	0.587	0.521	0.416	**0.418**	**0.413**	0.421	0.748	0.580	0.711	0.570
	336	**0.471**	**0.467**	0.535	0.501	0.564	0.511	0.584	0.527	0.442	**0.433**	**0.422**	0.436	0.754	0.595	0.816	0.619
	720	**0.484**	**0.473**	0.543	0.529	0.585	0.527	0.612	0.557	0.477	**0.456**	**0.447**	0.466	0.725	0.591	0.762	0.610
	avg	**0.461**	**0.449**	0.512	0.493	0.536	0.499	0.578	0.521	0.427	**0.426**	**0.413**	0.430	0.693	0.568	0.712	0.580
SourceData		ETT{m1,m2,h1}		ETTm1		ETTm2		ETTh1		ETTh2		ETTh2		ETTh2		ETTh2	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MAE	MSE
SourceData ↓ ETTh2	96	**0.294**	**0.357**	0.344	0.380	0.326	0.370	0.297	0.343	0.285	0.342	**0.274**	**0.336**	0.376	0.421	0.401	0.421
	192	**0.375**	**0.397**	0.430	0.424	0.411	0.415	0.395	0.403	0.354	0.389	**0.339**	**0.379**	0.418	0.441	0.452	0.455
	336	**0.407**	**0.407**	0.460	0.452	0.450	0.448	0.418	0.429	0.373	0.407	**0.329**	**0.380**	0.408	0.439	0.464	0.469
	720	**0.408**	**0.435**	0.487	0.477	0.446	0.455	0.428	0.446	0.406	0.441	**0.379**	**0.422**	0.449	0.464	0.477	0.480
	avg	**0.371**	**0.384**	0.430	0.433	0.408	0.422	0.385	0.405	0.354	0.394	**0.330**	**0.379**	0.413	0.441	0.449	0.456

Table 21: Comparison of the complete performance with diverse mask ratios ($\{12.5\%, 25\%, 37.5\%, 50\%\}$) on **full-data imputation** task.

Models		**WQ4TS (Ours)**		GPT4TS [2023]		TimesNet [2023]		PatchTST [2023]		ETSformer [2023]		LightTS [2022a]		DLinear [2023]		FEDformer [2022a]		Stationary [2022]		Autoformer [2021]	
MaskRatio		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	12.5%	0.019	**0.077**	**0.017**	0.085	0.023	0.101	0.041	0.130	0.096	0.229	0.093	0.206	0.080	0.193	0.052	0.166	0.032	0.119	0.046	0.144
	25%	0.022	**0.092**	**0.022**	0.096	0.023	0.101	0.044	0.135	0.096	0.229	0.093	0.206	0.080	0.193	0.052	0.166	0.032	0.119	0.046	0.144
	37.5%	**0.028**	**0.110**	0.029	0.111	0.029	0.111	0.049	0.143	0.133	0.271	0.113	0.231	0.103	0.219	0.069	0.191	0.039	0.131	0.057	0.161
	50%	**0.035**	**0.117**	0.040	0.128	0.036	0.124	0.055	0.151	0.186	0.323	0.134	0.255	0.132	0.248	0.089	0.218	0.047	0.145	0.067	0.174
	Avg	**0.026**	**0.099**	0.028	0.105	0.027	0.107	0.047	0.140	0.120	0.253	0.104	0.218	0.093	0.206	0.062	0.177	0.036	0.126	0.051	0.150
ETTm2	12.5%	0.018	0.079	**0.017**	**0.076**	0.018	0.080	0.026	0.094	0.108	0.239	0.034	0.127	0.062	0.166	0.056	0.159	0.021	0.088	0.023	0.092
	25%	**0.019**	0.082	0.020	**0.080**	0.020	0.085	0.028	0.099	0.164	0.294	0.042	0.143	0.085	0.196	0.080	0.195	0.024	0.096	0.026	0.101
	37.5%	**0.021**	**0.085**	0.022	0.087	0.023	0.091	0.030	0.104	0.237	0.356	0.051	0.159	0.106	0.222	0.110	0.231	0.027	0.103	0.030	0.108
	50%	**0.024**	**0.094**	0.025	0.095	0.026	0.098	0.034	0.110	0.323	0.421	0.059	0.174	0.131	0.247	0.156	0.276	0.030	0.108	0.035	0.119
	Avg	**0.020**	0.085	0.021	**0.084**	0.022	0.088	0.029	0.102	0.208	0.327	0.046	0.151	0.096	0.208	0.101	0.215	0.026	0.099	0.029	0.105
ETTh1	12.5%	**0.040**	**0.137**	0.043	0.140	0.057	0.159	0.093	0.201	0.126	0.263	0.240	0.345	0.151	0.267	0.070	0.190	0.060	0.165	0.074	0.182
	25%	**0.053**	**0.155**	0.054	0.156	0.069	0.178	0.107	0.217	0.169	0.304	0.265	0.364	0.180	0.292	0.106	0.236	0.080	0.189	0.090	0.203
	37.5%	**0.070**	**0.175**	0.072	0.180	0.084	0.196	0.120	0.230	0.220	0.347	0.296	0.382	0.215	0.318	0.124	0.258	0.102	0.212	0.109	0.222
	50%	**0.093**	**0.202**	0.107	0.216	0.102	0.215	0.141	0.248	0.293	0.402	0.334	0.404	0.257	0.347	0.165	0.299	0.133	0.240	0.137	0.248
	Avg	**0.064**	**0.167**	0.069	0.173	0.078	0.187	0.115	0.224	0.202	0.329	0.284	0.373	0.201	0.306	0.117	0.246	0.094	0.201	0.103	0.214
ETTh2	12.5%	0.040	**0.124**	**0.039**	0.125	0.040	0.130	0.057	0.152	0.187	0.319	0.101	0.231	0.100	0.216	0.095	0.212	0.042	0.133	0.044	0.138
	25%	**0.043**	**0.131**	0.044	0.135	0.046	0.141	0.061	0.158	0.279	0.390	0.115	0.246	0.127	0.247	0.137	0.258	0.049	0.147	0.050	0.149
	37.5%	**0.049**	**0.143**	0.051	0.147	0.052	0.151	0.067	0.166	0.400	0.465	0.126	0.257	0.158	0.276	0.187	0.304	0.056	0.158	0.060	0.163
	50%	**0.053**	**0.155**	0.059	0.158	0.060	0.162	0.073	0.174	0.602	0.572	0.136	0.268	0.183	0.299	0.232	0.341	0.065	0.170	0.068	0.173
	Avg	**0.047**	**0.138**	0.048	0.141	0.049	0.146	0.065	0.163	0.367	0.436	0.119	0.250	0.142	0.259	0.163	0.279	0.053	0.152	0.055	0.156
Electricity	12.5%	**0.043**	**0.129**	0.080	0.194	0.085	0.202	0.055	0.160	0.196	0.321	0.102	0.229	0.092	0.214	0.107	0.237	0.093	0.210	0.089	0.210
	25%	**0.049**	**0.142**	0.087	0.203	0.089	0.206	0.065	0.175	0.207	0.332	0.121	0.252	0.118	0.247	0.120	0.251	0.097	0.214	0.096	0.220
	37.5%	**0.056**	**0.151**	0.094	0.211	0.094	0.213	0.076	0.189	0.219	0.344	0.141	0.273	0.144	0.276	0.136	0.266	0.102	0.220	0.104	0.229
	50%	**0.065**	**0.165**	0.101	0.220	0.100	0.221	0.091	0.208	0.235	0.357	0.160	0.293	0.175	0.305	0.158	0.284	0.108	0.228	0.113	0.239
	Avg	**0.053**	**0.147**	0.090	0.207	0.092	0.210	0.072	0.183	0.214	0.339	0.131	0.262	0.132	0.260	0.130	0.259	0.100	0.218	0.101	0.225
Weather	12.5%	**0.024**	**0.040**	0.026	0.049	0.025	0.045	0.029	0.049	0.057	0.141	0.047	0.101	0.039	0.084	0.041	0.107	0.027	0.051	0.026	0.047
	25%	**0.026**	**0.043**	0.028	0.052	0.029	0.052	0.031	0.053	0.065	0.155	0.052	0.111	0.048	0.103	0.064	0.163	0.029	0.056	0.030	0.054
	37.5%	**0.030**	**0.047**	0.033	0.060	0.031	0.057	0.035	0.058	0.081	0.180	0.058	0.121	0.057	0.117	0.107	0.229	0.033	0.062	0.032	0.060
	50%	**0.033**	**0.052**	0.037	0.065	0.034	0.062	0.038	0.063	0.102	0.207	0.065	0.133	0.066	0.134	0.183	0.312	0.037	0.068	0.037	0.067
	Avg	**0.028**	**0.046**	0.031	0.056	0.030	0.054	0.060	0.144	0.076	0.171	0.055	0.117	0.052	0.110	0.099	0.203	0.032	0.059	0.031	0.057
1^{st} Count		**53**		7		0		0		0		0		0		0		0		0	

Table 22: Comparison of the complete performance with diverse mask ratios ($\{12.5\%, 25\%, 37.5\%, 50\%\}$) on **zero-shot imputation** task. Where $Source \rightarrow Traget$ indicates that the model is first pre-trained on the single train set of the *SourceDomain*, subsequently, the model parameters are frozen and predicted on the test set of the *TargetDomain*.

Scenarios		**WQ4TS**		GPT4TS		DLinear		PatchTST		TimesNet		FEDformer		Autoformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm2 ↓ ETTm1	96	**0.043**	**0.134**	0.759	0.546	0.118	0.231	0.080	0.176	0.092	0.180	0.666	0.607	0.367	0.419
	192	**0.046**	**0.139**	0.767	0.549	0.163	0.270	0.094	0.191	0.110	0.199	0.721	0.638	0.525	0.515
	336	**0.050**	**0.145**	0.770	0.550	0.220	0.313	0.109	0.196	0.125	0.212	0.790	0.671	0.528	0.517
	720	**0.061**	**0.159**	0.772	0.551	0.309	0.368	0.116	0.202	0.146	0.228	0.871	0.706	0.607	0.553
	avg	**0.050**	**0.144**	0.767	0.549	0.203	0.295	0.099	0.191	0.118	0.205	0.762	0.655	0.507	0.501
ETTm1 ↓ ETTm2	96	**0.026**	**0.094**	0.121	0.236	0.067	0.173	0.055	0.142	0.087	0.206	1.676	0.989	0.927	0.708
	192	**0.028**	**0.095**	0.152	0.262	0.100	0.211	0.056	0.145	0.091	0.213	2.019	1.086	1.148	0.782
	336	**0.030**	**0.100**	0.153	0.262	0.131	0.242	0.059	0.151	0.094	0.219	2.309	1.159	1.484	0.890
	720	**0.033**	**0.104**	0.154	0.263	0.160	0.270	0.063	0.157	0.098	0.226	2.558	1.219	1.808	0.986
	avg	**0.029**	**0.098**	0.145	0.256	0.114	0.224	0.058	0.149	0.093	0.216	2.140	1.113	1.342	0.842
ETTm1 ↓ ETTh1	96	**0.128**	**0.241**	0.851	0.601	0.324	0.384	0.291	0.355	0.278	0.361	1.101	0.795	0.881	0.696
	192	**0.148**	**0.257**	0.852	0.602	0.365	0.407	0.301	0.360	0.305	0.381	1.066	0.783	0.970	0.729
	336	**0.183**	**0.283**	0.856	0.602	0.416	0.435	0.317	0.368	0.338	0.403	1.065	0.784	0.962	0.727
	720	**0.244**	**0.317**	0.856	0.602	0.485	0.469	0.342	0.381	0.387	0.434	1.065	0.785	1.011	0.750
	avg	**0.176**	**0.274**	0.854	0.602	0.397	0.424	0.313	0.366	0.327	0.395	1.074	0.787	0.956	0.725
ETTm1 ↓ ETTh2	96	**0.059**	**0.152**	0.232	0.325	0.118	0.238	0.073	0.176	0.098	0.223	2.489	1.194	2.161	1.127
	192	**0.061**	**0.156**	0.249	0.336	0.145	0.265	0.076	0.180	0.106	0.234	2.767	1.260	2.750	1.276
	336	**0.065**	**0.161**	0.249	0.336	0.174	0.291	0.080	0.186	0.113	0.242	2.916	1.294	2.407	1.189
	720	**0.072**	**0.172**	0.249	0.336	0.204	0.316	0.087	0.193	0.120	0.252	3.014	1.316	2.576	1.230
	avg	**0.064**	**0.160**	0.245	0.333	0.160	0.277	0.079	0.184	0.109	0.238	2.796	1.266	2.473	1.206
ETTm1 ↓ Exchange	96	**0.003**	**0.029**	0.027	0.117	0.086	0.223	0.005	0.038	0.045	0.149	3.126	1.443	2.672	1.336
	192	**0.003**	**0.029**	0.027	0.117	0.217	0.361	0.005	0.042	0.047	0.153	3.114	1.437	3.005	1.426
	336	**0.003**	**0.032**	0.027	0.117	0.425	0.508	0.006	0.046	0.046	0.151	3.096	1.438	2.968	1.376
	720	**0.004**	**0.033**	0.027	0.117	0.705	0.656	0.007	0.051	0.044	0.149	3.092	1.441	2.973	1.391
	avg	**0.003**	**0.031**	0.027	0.117	0.358	0.437	0.006	0.044	0.045	0.150	3.107	1.440	2.904	1.382
ETTm1 ↓ Weather	96	**0.027**	**0.041**	0.102	0.159	0.119	0.207	0.060	0.089	0.132	0.188	0.991	0.777	1.002	0.792
	192	**0.029**	**0.041**	0.102	0.160	0.145	0.252	0.064	0.097	0.132	0.188	0.992	0.781	0.950	0.756
	336	**0.031**	**0.044**	0.103	0.160	0.187	0.306	0.067	0.103	0.132	0.187	1.008	0.779	1.039	0.804
	720	**0.034**	**0.046**	0.104	0.160	0.244	0.365	0.071	0.109	0.134	0.187	1.006	0.778	1.010	0.799
	avg	**0.030**	**0.043**	0.103	0.160	0.174	0.283	0.065	0.099	0.132	0.188	0.999	0.779	1.000	0.788

Table 23: Model comparison in classification. The experimental results contain the performance of the three most representative approaches (OneFitsAll, PatchTST, and TimesNet) and proposed WQ4TS on all 35 classification datasets, which includes four metrics: accuracy (Acc), precision (Pre), recall (Rec), and F1 score (F1).

Models		WQ4TS (**ours**)				OneFitsAll [2023]					PatchTST [2023]					TimesNet [2023]				
Metric (%)	Acc	Pre	Rec	F1	Avg	Acc	Pre	Rec	F1	Avg	Acc	Pre	Rec	F1	Avg	Acc	Pre	Rec	F1	Avg
BeetleFly	80.00	80.00	80.00	80.00	80.00	**85.00**	**88.46**	**85.00**	**84.65**	**85.78**	80.00	83.46	80.00	79.65	80.78	75.00	83.33	75.00	73.33	76.66
BME	**96.00**	**96.43**	**96.00**	**96.02**	**96.11**	93.33	93.53	93.33	93.35	93.39	88.00	91.18	88.00	88.32	88.88	92.67	93.99	92.67	92.66	93.00
CBF	**97.67**	**97.70**	**97.67**	**97.65**	**97.67**	87.67	89.96	87.66	87.57	88.22	89.78	89.94	89.81	89.84	89.84	93.67	93.68	93.68	93.67	93.67
Chinatown	**98.96**	**98.02**	**98.93**	**98.46**	**98.59**	96.79	94.94	97.46	96.08	96.32	95.63	94.38	94.67	94.52	94.80	97.96	96.53	98.59	97.49	97.64
ChlorineConcentration	53.26	57.75	43.33	33.17	46.88	**60.05**	**65.30**	**46.19**	**46.64**	**54.55**	57.24	51.82	39.09	33.60	45.44	53.75	51.17	34.05	24.64	40.90
DistalPhalanxTW	**81.22**	65.75	**66.69**	**65.02**	**69.67**	71.94	**67.91**	57.61	55.68	63.28	70.50	46.09	50.37	46.70	53.42	73.38	67.09	59.13	59.70	64.83
ECG200	**96.00**	**94.92**	**95.37**	**95.14**	**95.36**	91.00	90.44	89.93	90.17	90.38	86.00	85.12	84.20	84.62	84.98	89.00	88.72	87.15	87.83	88.17
ECG5000	**95.53**	**79.79**	**65.63**	**71.76**	**77.93**	94.56	76.62	56.73	62.01	72.48	94.67	73.41	56.40	59.97	71.11	93.93	77.58	55.31	60.50	71.83
ElectricDevices	**74.24**	**68.21**	**63.80**	**64.86**	**67.78**	62.05	58.84	54.35	55.22	57.62	66.76	63.03	59.51	60.60	62.47	69.23	65.60	63.30	63.79	65.48
FaceAll	**80.00**	**83.12**	**88.58**	**82.70**	**83.60**	75.74	81.52	84.35	78.95	80.14	75.44	77.58	84.92	76.88	78.70	75.09	80.77	85.56	78.92	80.09
FaceFour	84.09	83.21	84.84	83.63	83.94	**90.91**	**91.04**	**91.13**	**90.92**	**91.00**	85.23	86.53	85.98	86.13	85.97	89.77	90.20	90.17	90.11	90.06
FacesUCR	**95.51**	**94.50**	**92.51**	**93.06**	**93.89**	86.15	86.55	82.36	83.67	84.68	84.83	85.28	80.03	81.78	82.98	85.07	85.94	82.21	83.44	84.17
FiftyWords	**80.77**	**66.02**	**66.52**	**64.08**	**69.35**	70.33	60.85	55.43	55.19	60.45	75.16	65.69	60.40	59.12	65.09	66.15	54.22	48.47	47.80	54.16
GunPointAgeSpan	**96.14**	**96.86**	**96.06**	**96.09**	**96.29**	89.56	90.06	89.49	89.51	89.66	89.24	89.80	89.17	89.19	89.35	94.94	94.94	94.94	94.94	94.94
GunPointMaleVersusFemale	**99.68**	**99.70**	**99.67**	**99.68**	**99.68**	94.62	95.00	94.37	93.24	94.31	97.47	97.44	97.49	97.46	97.47	99.05	99.07	99.03	99.05	99.05
GunPointOldVersusYoung	**98.10**	**98.18**	**98.03**	**98.09**	**98.10**	95.05	95.06	95.03	95.05	95.05	93.65	93.77	93.55	93.62	93.65	52.38	26.19	50.00	34.38	40.74
GunPoint	**96.00**	**96.05**	**95.98**	**96.00**	**96.01**	92.67	93.09	92.60	92.64	92.75	76.00	78.47	75.80	75.37	76.41	94.00	94.23	93.95	93.99	94.04
InsectEPGSmallTrain	83.13	57.92	66.67	61.63	67.34	75.17	75.57	70.97	72.70	73.60	**83.53**	**83.97**	**78.86**	**80.78**	**81.78**	83.03	84.46	76.61	73.89	79.50
InsectWingbeatSound	**69.34**	**69.60**	**69.34**	**69.04**	**69.33**	63.03	64.37	63.03	63.18	63.40	65.56	65.37	65.56	64.94	65.36	62.58	62.85	62.58	62.47	62.62
ItalyPowerDemand	96.60	96.60	96.60	96.60	96.60	96.99	96.99	96.99	96.99	96.99	96.60	96.62	96.60	96.60	96.60	**97.18**	**97.18**	**97.18**	**97.18**	**97.18**
MedicalImages	**84.47**	**81.67**	**76.79**	**77.84**	**80.19**	70.53	66.96	66.78	65.50	67.44	68.55	65.34	60.71	61.11	63.93	73.03	68.97	63.52	65.37	67.72
MiddlePhalanxTW	**70.39**	41.28	**51.17**	**45.12**	**51.99**	60.39	42.11	38.75	35.71	44.24	62.34	**48.47**	40.17	37.42	47.10	61.69	41.98	40.95	38.48	45.77
MoteStrain	88.10	88.24	87.82	87.97	88.03	87.62	87.80	87.31	87.47	87.55	85.22	85.73	84.71	84.96	85.16	**89.54**	**89.65**	**89.30**	**89.43**	**89.48**
Plane	**99.05**	99.11	**99.11**	**99.08**	**99.09**	99.05	**99.16**	99.05	99.07	99.08	98.10	98.01	98.15	98.04	98.08	99.05	99.16	99.05	99.07	99.08
ProximalPhalanxTW	**87.44**	**77.46**	**70.73**	**68.10**	**75.93**	80.49	39.21	45.17	41.91	51.70	81.95	56.48	53.84	53.57	61.46	82.93	65.09	62.37	62.22	68.15
SonyAIBORobotSurface1	**84.19**	**84.76**	**85.34**	**84.17**	**84.61**	79.37	82.30	81.49	79.34	80.62	69.55	76.90	72.89	69.00	72.08	76.37	76.62	77.14	76.30	76.61
SonyAIBORobotSurface2	**89.72**	**89.37**	**88.76**	**89.04**	**89.22**	86.15	85.75	84.67	85.13	85.42	85.31	84.35	84.82	84.57	84.76	81.32	80.25	80.19	80.22	80.50
SwedishLeaf	**95.20**	**95.28**	**95.32**	**95.23**	**95.26**	84.80	85.66	84.97	84.56	85.00	91.20	91.33	91.62	91.33	91.37	88.00	88.32	88.43	87.90	88.16
SyntheticControl	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	96.33	96.46	96.33	96.31	96.36	99.00	99.03	99.00	99.00	99.01	99.67	99.67	99.67	99.67	99.67
ToeSegmentation2	**91.54**	70.77	**80.00**	**84.92**	**81.81**	84.62	**75.11**	68.00	70.45	74.55	83.08	72.99	78.34	74.96	77.34	84.62	74.45	72.84	73.59	76.38
Trace	81.00	77.50	75.42	70.87	76.20	84.00	87.47	85.78	83.04	85.07	75.00	85.80	77.68	69.92	77.10	**92.00**	**91.62**	**92.10**	**91.69**	**91.85**
UMD	**99.31**	**99.32**	**99.31**	**99.31**	**99.31**	98.65	98.65	98.65	98.65	98.65	97.92	97.99	97.92	97.90	97.93	92.50	92.12	92.60	92.19	92.35
UWaveGestureLibraryY	**79.65**	**79.44**	**79.75**	**79.24**	**79.52**	67.00	66.65	67.07	66.19	66.73	70.94	70.43	71.02	70.42	70.70	63.51	62.81	63.61	62.82	63.19
Wafer	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	99.58	98.72	99.10	98.91	99.08	99.37	98.83	97.86	98.34	98.60	99.53	99.07	98.48	98.77	98.96
WordSynonyms	56.74	46.59	39.49	39.50	45.58	58.15	47.43	40.03	40.83	46.61	**61.76**	**54.59**	**45.32**	**46.33**	**52.00**	56.74	43.82	38.11	37.75	44.11
1st Count		27				3					1					3				
Average Acc		87.40				83.12					82.28					82.23				

Table 24: Comparison (**Part-1**) of the complete performance on **few-shot classification** task. Where *SourceDomain→TragetDomain* indicates that the model is first pre-trained in the *SourceDomain* train set, subsequently, the parameters are fine-tuned in partial (5%/10%) samples of the *TargetDomain* train set and finally predicted on the *TargetDomain* test set.

Scenarios			Models	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Avg (%)
DistalPhalanxTW → ProximalPhalanxOutlineCorrect		Full Data	MICN	84.88	85.82	**78.42**	**80.73**	82.46
			TimesNet	**85.22**	**88.80**	77.51	80.43	**82.99**
			PatchTST	78.69	75.80	73.03	74.06	75.39
			OneFitsAll	79.73	76.00	76.12	76.35	77.20
		Few-shot (10%)	Random init.	75.26	71.93	67.01	68.21	70.60
			MICN	80.38	79.46	78.00	78.67	79.13
			TimesNet	77.32	73.77	73.19	73.46	74.44
			PatchTST	78.69	81.61	68.06	69.95	74.57
			OneFitsAll	77.66	74.35	72.27	73.09	74.34
			WQ4TS	**86.25**	**84.11**	**84.11**	**84.11**	**84.65**
		Few-shot (5%)	Random init.	68.38	34.19	50.00	40.61	48.30
			MICN	81.79	80.32	75.87	77.42	78.85
			TimesNet	78.01	78.99	67.85	69.60	73.61
			PatchTST	78.35	76.20	71.02	72.52	74.52
			OneFitsAll	76.98	74.22	69.43	70.78	72.85
			WQ4TS	**84.54**	**87.68**	**76.71**	**79.52**	**82.11**
SonyAIBORobotSurface2 → Chinatown		Full Data	MICN	63.27	55.17	55.49	55.27	57.30
			TimesNet	**97.96**	**96.53**	**98.59**	**97.49**	**97.64**
			PatchTST	95.63	94.38	94.67	94.52	94.80
			OneFitsAll	96.79	94.94	97.46	96.08	96.32
		Few-shot (10%)	Random init.	66.79	64.76	67.79	66.11	66.36
			MICN	27.41	13.70	50.00	21.51	28.15
			TimesNet	91.25	89.65	88.02	88.78	89.43
			PatchTST	80.76	87.95	65.22	67.52	75.36
			OneFitsAll	78.13	74.21	65.40	67.21	71.24
			WQ4TS	**94.17**	**91.23**	**95.98**	**93.10**	**93.62**
		Few-shot (5%)	Random init.	27.41	13.70	50.00	21.51	28.16
			MICN	27.41	13.70	50.00	21.51	28.15
			TimesNet	27.41	13.70	50.00	21.51	28.15
			PatchTST	81.05	84.63	66.75	69.33	75.44
			OneFitsAll	27.41	13.70	50.00	21.51	28.15
			WQ4TS	**91.55**	**90.16**	**88.22**	**89.12**	**89.76**
Trace → DistalPhalanxOutlineCorrect		Full Data	MICN	**80.07**	**81.39**	**77.70**	**78.48**	**79.41**
			TimesNet	76.45	75.78	75.96	75.86	76.01
			PatchTST	71.74	72.44	68.57	68.85	70.40
			OneFitsAll	74.28	77.38	70.50	70.82	73.25
		Few-shot (10%)	Random init.	58.33	29.17	50.00	36.84	43.59
			MICN	75.00	75.48	72.48	73.01	73.99
			TimesNet	71.74	72.24	68.70	69.00	70.42
			PatchTST	68.48	71.72	63.66	62.65	66.62
			OneFitsAll	71.74	70.97	70.19	70.43	70.83
			WQ4TS	**75.36**	**76.68**	**74.42**	**74.98**	**75.36**
		Few-shot (5%)	Random init.	58.33	29.17	50.00	36.84	43.59
			MICN	64.49	64.14	60.25	59.37	62.06
			TimesNet	67.75	77.69	61.68	58.74	66.47
			PatchTST	64.13	69.63	57.83	53.70	61.32
			OneFitsAll	63.04	69.96	56.27	50.72	59.99
			WQ4TS	**71.01**	**70.90**	**68.32**	**68.64**	**69.72**

Table 25: Comparison (**Part-2**) of the complete performance on **few-shot classification** task. Where *SourceDomain→TragetDomain* indicates that the model is first pre-trained in the *SourceDomain* train set, subsequently, the parameters are fine-tuned in partial (5%/10%) samples of the *TargetDomain* train set and finally predicted on the *TargetDomain* test set.

Scenarios		Models	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Avg (%)
SonyAIBORobotSurface2 → SonyAIBORobotSurface1	Full Data	MICN	74.38	80.85	77.45	74.07	76.69
		TimesNet	76.37	76.62	77.14	76.30	76.61
		PatchTST	69.55	76.90	72.89	69.00	72.08
		OneFitsAll	**79.37**	**82.30**	**81.49**	**79.34**	**80.62**
	Few-shot (10%)	Random init.	58.51	54.26	52.51	40.88	46.53
		MICN	62.40	61.80	61.92	61.83	61.99
		TimesNet	64.23	69.14	67.07	63.82	66.07
		PatchTST	63.56	65.35	65.15	63.55	64.40
		OneFitsAll	64.56	63.85	63.86	63.85	64.03
		WQ4TS	**91.51**	**91.36**	**91.32**	**91.34**	**91.38**
	Few-shot (5%)	Random init.	47.09	28.55	42.45	33.61	37.94
		MICN	56.91	52.55	50.38	39.85	49.92
		TimesNet	60.23	68.97	54.02	45.82	57.26
		PatchTST	57.07	28.54	50.00	36.33	42.98
		OneFitsAll	57.07	28.54	50.00	36.33	42.98
		WQ4TS	**85.03**	**86.45**	**87.90**	**85.01**	**86.09**
HouseTwenty → GunPointMaleVersusFemale	Full Data	MICN	98.73	98.77	98.70	98.73	98.73
		TimesNet	**99.05**	**99.07**	**99.03**	**99.05**	**99.05**
		PatchTST	97.47	97.44	97.49	97.46	97.47
		OneFitsAll	94.62	95.00	94.37	93.24	94.31
	Few-shot (10%)	Random init.	87.66	89.45	88.22	87.60	88.23
		MICN	75.63	78.43	76.39	75.33	76.44
		TimesNet	81.96	82.73	81.55	81.68	81.98
		PatchTST	96.84	96.80	96.89	96.83	96.84
		OneFitsAll	76.27	78.42	76.93	76.06	76.92
		WQ4TS	**98.10**	**98.26**	**98.00**	**98.09**	**98.11**
	Few-shot (5%)	Random init.	66.77	69.86	67.67	66.08	67.59
		MICN	69.30	72.46	70.17	68.72	70.16
		TimesNet	52.53	51.64	51.16	47.92	50.81
		PatchTST	84.18	86.17	84.78	84.09	84.81
		OneFitsAll	68.99	73.31	70.00	68.12	70.11
		WQ4TS	**85.76**	**85.79**	**85.64**	**85.70**	**85.72**
ProximalPhalanxOutlineCorrect → HouseTwenty	Full Data	MICN	75.63	75.00	75.13	75.06	75.21
		TimesNet	63.87	64.97	58.93	56.84	61.15
		PatchTST	**85.71**	**85.29**	**85.48**	**85.38**	**85.47**
		OneFitsAll	64.71	63.82	61.58	61.36	62.78
	Few-shot (10%)	Random init.	61.34	67.68	54.55	47.57	57.78
		MICN	59.66	57.71	56.68	56.30	57.59
		TimesNet	66.39	65.42	65.23	65.31	65.59
		PatchTST	63.03	61.82	61.51	61.59	61.98
		OneFitsAll	60.50	69.82	53.28	44.43	57.01
		WQ4TS	**76.47**	**82.18**	**72.55**	**72.94**	**76.03**
	Few-shot (5%)	Random init.	54.62	43.44	47.93	39.83	46.45
		MICN	47.06	55.12	52.70	42.93	49.45
		TimesNet	42.02	21.01	50.00	29.59	35.65
		PatchTST	42.02	21.01	50.00	29.59	35.65
		OneFitsAll	42.02	21.01	50.00	29.59	35.65
		WQ4TS	**73.95**	**77.92**	**70.10**	**70.24**	**73.05**

Table 26: Comparison (**Part-3**) of the complete performance on **few-shot classification** task. Where *SourceDomain→TragetDomain* indicates that the model is first pre-trained in the *SourceDomain* train set, subsequently, the parameters are fine-tuned in partial (5%/10%) samples of the *TargetDomain* train set and finally predicted on the *TargetDomain* test set.

Scenarios		Models	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Avg (%)
PigArtPressure → InsectEPGRegularTrain	Full Data	MICN	**86.78**	74.94	**83.13**	**79.83**	**81.17**
		TimesNet	78.10	76.27	74.82	71.85	75.26
		PatchTST	83.13	**84.74**	71.74	72.78	78.10
		OneFitsAll	83.03	84.46	76.61	73.89	79.50
	Few-shot (10%)	Random init.	47.39	15.80	33.33	21.44	29.49
		MICN	35.74	11.91	33.33	17.55	24.63
		TimesNet	83.13	55.98	66.67	60.30	66.52
		PatchTST	48.19	48.67	45.81	39.85	45.63
		OneFitsAll	83.13	57.92	66.67	61.63	67.34
		WQ4TS	**93.17**	**93.74**	**87.53**	**89.68**	**91.03**
	Few-shot (5%)	Random init.	35.74	11.91	33.33	17.55	24.63
		MICN	35.74	11.91	33.33	17.55	24.63
		TimesNet	35.74	15.86	33.33	21.50	26.61
		PatchTST	37.35	20.95	37.35	24.50	30.04
		OneFitsAll	35.74	11.91	33.33	17.55	24.63
		WQ4TS	**83.85**	**84.37**	**78.78**	**80.71**	**81.93**
SonyAIBORobotSurface1 → InsectEPGSmallTrain	Full Data	MICN	75.17	75.57	70.97	72.70	73.60
		TimesNet	83.03	84.46	76.61	73.89	79.50
		PatchTST	**83.53**	**83.97**	**78.86**	**80.78**	**81.78**
		OneFitsAll	75.17	75.57	70.97	72.70	73.60
	Few-shot (10%)	Random init.	47.39	15.80	33.33	21.44	29.49
		MICN	35.74	11.91	33.33	17.55	24.63
		TimesNet	38.55	55.98	35.31	30.70	40.14
		PatchTST	39.76	46.58	39.93	31.79	39.51
		OneFitsAll	35.74	11.91	33.33	17.55	24.63
		WQ4TS	**83.13**	**57.92**	**66.67**	**61.63**	**67.34**
	Few-shot (5%)	Random init.	47.39	15.80	33.33	21.44	29.49
		MICN	35.74	11.91	33.33	17.55	24.63
		TimesNet	26.98	39.18	24.72	21.49	28.09
		PatchTST	27.71	45.16	36.68	26.44	33.99
		OneFitsAll	35.74	11.91	33.33	17.55	24.63
		WQ4TS	**83.13**	**55.16**	**66.67**	**60.32**	**66.32**
Earthquakes → ItalyPowerDemand	Full Data	MICN	95.63	95.63	95.63	95.63	95.63
		TimesNet	**97.18**	**97.18**	**97.18**	**97.18**	**97.18**
		PatchTST	96.60	96.62	96.60	96.60	96.61
		OneFitsAll	96.99	96.99	96.99	96.99	96.99
	Few-shot (10%)	Random init.	78.92	79.58	78.90	78.87	79.07
		MICN	90.28	91.24	90.26	90.22	90.50
		TimesNet	86.59	87.02	86.60	86.55	86.69
		PatchTST	87.37	88.40	87.34	87.28	87.59
		OneFitsAll	94.66	94.75	94.65	94.65	94.68
		WQ4TS	**96.31**	**96.34**	**96.30**	**96.31**	**96.31**
	Few-shot (5%)	Random init.	70.86	72.47	70.89	70.63	71.21
		MICN	70.94	77.75	71.01	69.09	72.19
		TimesNet	73.28	77.08	73.33	72.33	74.01
		PatchTST	60.93	64.96	61.01	58.20	61.27
		OneFitsAll	83.19	85.25	83.22	82.95	83.65
		WQ4TS	**94.85**	**94.85**	**94.85**	**94.85**	**94.85**

Table 27: Comparison (**Part-4**) of the complete performance on **few-shot classification** task. Where *SourceDomain→TragetDomain* indicates that the model is first pre-trained in the *SourceDomain* train set, subsequently, the parameters are fine-tuned in partial (5%/10%) samples of the *TargetDomain* train set and finally predicted on the *TargetDomain* test set.

Scenarios		Models	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Avg (%)
DistalPhalanxOutlineAgeGroup → MiddlePhalanxOutlineAgeGroup	Full Data	MICN	64.94	66.29	48.22	49.50	57.24
		TimesNet	**66.23**	**75.40**	48.98	51.02	**60.41**
		PatchTST	65.58	72.43	**49.12**	**51.37**	59.62
		OneFitsAll	64.29	72.26	47.59	49.46	58.40
	Few-shot (10%)	Random init.	19.22	7.21	30.13	11.62	17.61
		MICN	57.14	51.64	**49.92**	**50.35**	52.26
		TimesNet	48.70	48.76	49.42	46.54	48.36
		PatchTST	45.45	46.03	42.77	40.81	43.76
		OneFitsAll	47.40	28.88	39.98	33.06	37.33
		WQ4TS	**60.39**	**59.20**	45.05	46.23	**52.72**
	Few-shot (5%)	Random init.	18.83	6.28	33.33	10.56	17.25
		MICN	46.10	42.96	45.04	42.89	44.25
		TimesNet	47.40	44.67	47.32	45.21	46.15
		PatchTST	40.26	43.72	47.03	40.26	42.82
		OneFitsAll	35.71	34.12	37.32	32.00	34.79
		WQ4TS	**51.95**	**49.36**	**48.38**	**46.92**	**49.15**
BeetleFly → PowerCons	Full Data	MICN	95.56	95.65	95.56	95.55	95.58
		TimesNet	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
		PatchTST	98.89	98.91	98.89	98.89	98.89
		OneFitsAll	98.33	98.39	98.33	98.33	98.34
	Few-shot (10%)	Random init.	50.00	25.00	50.00	33.33	39.58
		MICN	83.89	84.10	83.89	83.86	83.94
		TimesNet	83.89	84.40	83.89	83.83	84.00
		PatchTST	90.00	90.72	90.00	89.96	90.17
		OneFitsAll	97.23	97.23	97.23	97.23	97.23
		WQ4TS	**98.89**	**98.89**	**98.89**	**98.89**	**98.89**
	Few-shot (5%)	Random init.	50.00	25.00	50.00	33.33	39.58
		MICN	62.78	66.56	62.78	60.53	63.16
		TimesNet	78.89	79.12	78.89	78.85	78.94
		PatchTST	82.78	83.11	82.78	82.73	82.84
		OneFitsAll	90.00	90.18	90.00	89.99	90.04
		WQ4TS	**95.00**	**95.01**	**95.00**	**95.00**	**95.00**
EOGHorizontalSignal → ProximalPhalanxOutlineAgeGroup	Full Data	MICN	86.34	77.94	80.22	78.94	80.86
		TimesNet	86.34	80.16	69.15	71.11	76.69
		PatchTST	**89.27**	**86.13**	**82.49**	**84.07**	**85.49**
		OneFitsAll	88.78	84.33	82.11	83.12	84.59
	Few-shot (10%)	Random init.	41.46	13.82	28.33	18.58	25.55
		MICN	85.85	77.67	67.19	68.46	74.79
		TimesNet	85.85	76.64	70.35	72.08	76.23
		PatchTST	87.80	81.74	79.77	80.67	82.50
		OneFitsAll	87.32	85.20	69.90	71.62	78.51
		WQ4TS	**89.27**	**91.40**	**84.07**	**84.61**	**87.34**
	Few-shot (5%)	Random init.	36.59	12.19	24.96	16.39	22.54
		MICN	77.07	68.90	76.19	68.56	72.68
		TimesNet	85.85	**90.68**	64.02	63.18	75.93
		PatchTST	87.32	80.31	79.40	79.84	81.72
		OneFitsAll	84.39	56.18	61.30	58.62	65.12
		WQ4TS	**87.80**	80.97	**84.52**	**82.48**	**83.94**

Table 28: Comparison of the ablation experiment, where *w/o TP* indicates the model without the proposed ***wave as token*** strategy. To guarantee fairness and effectiveness, *w/o TP* and *WQ4TS* have the same model structure and parameter size.

Variant		full datal				few shot				zero shot (single)				zero shot (multi)			
		ETTh1		ETTm1		ETTh1		ETTm1		ETTh1		ETTm1		ETTh1		ETTm1	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/o TP	96	0.543	0.515	0.455	0.446	0.613	0.560	0.465	0.455	0.691	0.615	0.555	0.522	0.665	0.593	0.582	0.536
	192	0.577	0.534	0.491	0.468	0.639	0.573	0.501	0.477	0.703	0.622	0.588	0.541	0.677	0.602	0.608	0.553
	336	0.597	0.546	0.525	0.487	0.660	0.587	0.540	0.499	0.711	0.629	0.608	0.555	0.682	0.608	0.631	0.569
	720	0.609	0.571	0.581	0.520	0.660	0.604	0.588	0.524	0.710	0.644	0.646	0.577	0.683	0.619	0.674	0.596
	Avg	0.582	0.542	0.513	0.480	0.643	0.581	0.524	0.489	0.704	0.628	0.599	0.548	0.677	0.606	0.623	0.564
WQ4TS	96	0.363	0.388	0.324	0.342	0.432	0.409	0.337	0.350	0.466	0.460	0.400	0.407	0.438	0.419	0.379	0.398
	192	0.386	0.409	0.349	0.357	0.456	0.428	0.364	0.363	0.505	0.483	0.415	0.417	0.449	0.439	0.389	0.404
	336	0.414	0.413	0.385	0.375	0.489	0.453	0.389	0.388	0.535	0.501	0.442	0.434	0.471	0.467	0.414	0.415
	720	0.467	0.460	0.412	0.402	0.531	0.502	0.441	0.417	0.543	0.529	0.481	0.459	0.484	0.473	0.461	0.446
	Avg	0.407	0.418	0.368	0.369	0.477	0.448	0.383	0.380	0.512	0.493	0.434	0.429	0.461	0.449	0.411	0.416