LeRet: Language-Empowered Retentive Network for Time Series Forecasting

Qihe Huang[1] , **Zhengyang Zhou**[1,2,3,*] , **Kuo Yang** [1] , **Gengyu Lin** [1] , **Zhongchao Yi** [1] , **Yang Wang** [1,2,*]

[1] University of Science and Technology of China (USTC), Hefei, China
[2] Suzhou Institute for Advanced Research, USTC, Suzhou, China
[3] State Key Laboratory of Resources and Environmental Information System
{hqh, yangkuo, lingengyu, zhongchaoyi}@mail.ustc.edu.cn, {zzy0929, angyan}@ustc.edu.cn

Abstract

Time series forecasting (TSF) plays a pivotal role in many real-world applications. Recently, the utilization of Large Language Models (LLM) in TSF has demonstrated exceptional predictive performance, surpassing most task-specific forecasting models. The success of LLM-based forecasting methods underscores the importance of causal dependence modeling and pre-trained knowledge transfer. However, challenges persist in directly applying LLM to TSF, i.e., the unacceptable parameter scales for resource-intensive model optimization, and the significant gap of feature space between structural numerical time series and natural language. To this end, we propose LeRet, a Language-empowered Retentive network for TSF. Technically, inspired by the causal extraction in LLM, we propose a causal dependence learner, enhanced by a patch-level pre-training task, to capture sequential causal evolution. To minimize the gap between numeric and language, we initialize a language description protocol for time series and design a TS-related language knowledge extractor to learn from language description, avoiding training with large-scale parameters. Finally, we dedicatedly achieve a Language-TS Modality Integrator for the fusion of two types data, and enable language-empowered sequence forecasting. Extensive evaluations demonstrate the effectiveness of our LeRet, especially reveal superiority on few-shot, and zero-shot forecasting tasks. Code is available at https://github.com/hqh0728/LeRet.

1 Introduction

Time series forecasting (TSF) is fundamental in many real-world applications [Zhou *et al.*, 2020b; Wang *et al.*, 2021a; Zhang *et al.*, 2022; Wang *et al.*, 2024; Wang *et al.*, 2023d], including weather forecasting [Wu *et al.*, 2021], traffic planning [Tian *et al.*, 2021; Wang *et al.*, 2023b; Zhou *et al.*, 2020a; Zhang *et al.*, 2024] and electricity scheduling [Zhou *et al.*, 2022]. In the past decade, various deep learning models have been applied to TSF, such as convolutional neural networks (CNN) [Luo and Wang, 2024; Wang *et al.*, 2023c; Wu *et al.*, 2023], recurrent neural networks (RNN) [Lin *et al.*, 2023b], graph neural networks (GNN) [Wang *et al.*, 2023a; Huang *et al.*, 2024b; Zhao *et al.*, 2024a; Zhou *et al.*, 2023b; Yang *et al.*, 2023; Zhao *et al.*, 2024b], and Transformers [Zhou *et al.*, 2021; Nie *et al.*, 2023; Zhang and Yan, 2023; Shabani *et al.*, 2023; Liu *et al.*, 2021; Lin *et al.*, 2023a], achieving excellent predictive performance. Despite the excellent performance, these task-specific models are confined to single time series modality, concentrating on the modeling of either intra-sequence dependence [Wu *et al.*, 2021; Zhou *et al.*, 2022], or inter-sequence dependence [Zhang and Yan, 2023; Huang *et al.*, 2024a]. At a cost of focusing on time series modeling, the task-specific models are only valid for data with limited size and thus lacking generalization, posing challenges to achieve few-shot or zero-shot forecasting.

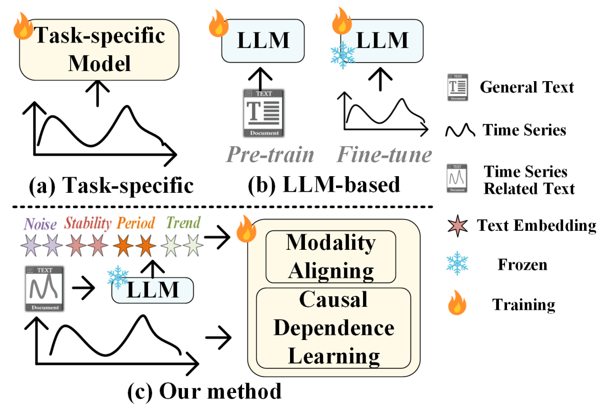

Figure 1: Illustration of the differences between LeRet and the other methods.

To gain the generalization capacity and general cross-domain forecasting, numerous Large Language Models (LLM) based TSF methods [Zhou *et al.*, 2023a; Cao *et al.*, 2023; Gruver *et al.*, 2023; Jin *et al.*, 2023; Chang *et al.*, 2023; Yu *et al.*, 2023] have emerged. These models fine-tune pre-trained LLM (e.g., GPT [Radford *et al.*, 2018], LLaMa [Touvron *et al.*, 2023]) to embed extensive language knowledge into time series, transferring pre-trained domain knowledge

* Yang Wang and Zhengyang Zhou are corresponding authors.

to temporal data. Generally, the success of LLM-based methods stem from two aspects, **i) Language knowledge empowering** [Zhou *et al.*, 2023a]. LLM is provided with abundant language knowledge, including a nuanced understanding of time series, notably enhancing the model to recognize sparse and complex temporal data. With the well understanding of series patterns, many studies [Gruver *et al.*, 2023; Yu *et al.*, 2023] further reveal the exceptional capability of LLM in few-shot and even zero-shot forecasting scenes. **ii) Causal dependence learning** [Chang *et al.*, 2023]. In contrast to existing bidirectional attention models [Nie *et al.*, 2023; Zhang and Yan, 2023; Shabani *et al.*, 2023; Liu *et al.*, 2021; Lin *et al.*, 2023a], LLM is based on causal attention where the hidden state of a token is only related to itself and preceding tokens, following inherent sequential causality in time series. Such encoding strategy enhances the comprehension of the temporal evolution process and has recently been effectively exploited in following studies [Lin *et al.*, 2023b].

While fine-tuning LLM for TSF leverages the cross-domain transferability from NLP pre-training, it suffers from two serious issues from the practicality and interpretability perspectives. **Regarding practicality**, although a large number of parameters can enhance the fitting capacity, directly optimizing such large network, even with techniques like LoRA [Hu *et al.*, 2022] and SoRA [Ding *et al.*, 2023], is resource-intensive. Besides, such large size of parameters hinders the feasibility for lightweight applications at the edge devices, limiting the accommodation of LLM-based methods to real-world forecasting scenarios. **Regarding interpretability**, LLM is pre-trained on discrete token-based text, while time series data accounts for numerical continuous data, resulting in a significant gap between natural language and numerics in representation space. This gap poses difficulties for LLM in achieving interpretable forecasting.

Hence, efficiently and effectively leveraging time series related knowledge from LLM to enhance the generalization and forecastability of TSF remains challenging. Fortunately, the success of LLM has prompted us to simultaneously consider the superiority of model structure design, the inherent sequential causality in time series modeling, and the task-oriented advantage, the excellent knowledge representation of natural languages with the capacity of series-level pattern transfer.

In this work, we propose a LeRet, a Language-empowered Retentive network for Time Series Forecasting. The differences between our LeRet and other TSF solutions are shown in Figure 1. **In terms of causal dependence**, we introduce a Causal Dependence Learner for time series, emphasizing the informativeness of historical information to capture causal-related temporal features. We further design a patch-level autoregressive forecasting task with a pre-training objective for the learner, to enhance the nuanced understanding of the causal evolution of series. **For language knowledge**, to efficiently utilize LLM, we propose a TS-Related Language Knowledge Extractor to extract general time series characteristics from pre-trained LLM. With Language-TS Modality Integrator, we map text embedding to time series feature space, then empower time series feature by actively receiving extracted language knowledge, and project the language-empowered representation to sequence-level forecasting. Our comprehensive evaluations demonstrate that LeRet is a robust time series learner that outperforms state-of-the-art forecasting models. As to generalization, with language model empowered, LeRet also excels in both few-shot and zero-shot learning scenarios. The main contributions of this work can be summarized as follows,

- We comprehensively dissect the advantages and limitations of LLM-based TSF methods, highlighting the strengths of model structure in sequential causality extraction and the language-oriented knowledge representation, and pointing out its shortcomings on the aspects of practicality and interpretability.
- We propose a novel framework that leverages the strengths of both LLM-based methods and task-specific models. Our LeRet especially gains the zero-shot and few-shot learning ability, from pre-trained language knowledge and our design of modality alignment.
- LeRet demonstrates outstanding predictive performance across various TSF tasks, including long-term, short-term, few-shot, and zero-shot forecasting. Quantitatively, LeRet outperforms 8 state-of-the-art models for long-term forecasting, achieving top-1 performance in 57 settings and top-2 in 5 settings out of a total of 64 settings.

2 Related Work

2.1 Task-sepcific Forecasting Methods

Benefiting from the advancements in deep learning, various models, including CNN-, GNN-, RNN-, and Transformer-based architectures, have been designed for task-specific forecasting [Liu *et al.*, 2021; Miao *et al.*, 2024; Wang *et al.*, 2021b; Lin *et al.*, 2024]. Notably, Transformer-based models have gained widespread acknowledgment for their global modeling capacity, allowing capturing long-term temporal dependencies through self-attention. Autoformer [Wu *et al.*, 2021] presents a series decomposition block based on a moving average to decompose complex temporal data into seasonal and trend components. FEDformer [Zhou *et al.*, 2022] leverages a frequency enhanced decomposition mechanism to obtain more efficient forecasting. Crossformer [Zhang and Yan, 2023] advocates for not only focusing on temporal dependencies but also considering relationships among variables. It introduces a routing mechanism to efficiently model cross-variable dependencies. PatchTST [Nie *et al.*, 2023] introduces patching and channel independence for TSF, reducing model complexity while dramatically enhancing forecasting performance. However, these models focus solely on a single time series modality, lacking strong predictive generalization. Additionally, their bidirectional attention mechanisms overlook the causal evolution inherent in time series. In response to this, in LeRet, we refine such causal features through the Causal Dependence Learner and extract language knowledge by TS-Related Language Knowledge Extractor to expand forecasting generalization.

2.2 LLM-based Forecasting Methods

The recent emergence of Large Language Models (LLMs) has introduced new possibilities for time series modeling, leading to a growing interest in the application of LLM to Time Series Forecasting (TSF). GPT4TS [Zhou *et al.*, 2023a] utilizes pre-trained language model without updating its self-attention and feedforward layers. The model undergoes fine-tuning and evaluation across various time series analysis tasks, demonstrating comparable or state-of-the-art performance by leveraging knowledge transfer from natural language pre-training. LLM4TS [Chang *et al.*, 2023] adopts a two-stage fine-tuning approach on the LLM to fully leverage time series data. Tempo [Cao *et al.*, 2023] decomposes the trend, seasonality, and residual components of time series, and dynamically selects prompts to ease the comprehension challenges for LLM. However, these LLM-based methods directly feed time series data into language pre-trained LLM, lacking interpretability. Additionally, the large parameter sizes limit their application scenarios. Therefore, we design the TS-Related Language Knowledge Extractor to extract lightweight language knowledge from the extensive LLM, and Language-TS Modality Integrator for multi-modality fusion, significantly improving the interpretability and practicality of LLM-based models.

3 Method

3.1 Problem Definition

The objective of time series forecasting (TSF) task is to predict the future values based on historical observations. Given historical L-steps input $X_{\text{input}} = [x_1, x_2, ..., x_L] \in \mathbb{R}^L$. We aim to learn a function $\mathbb{F}(\cdot)$ to accurately forecast $\hat{X}$, where $\hat{X} = [\hat{x}_{L+1}, \hat{x}_{L+2}, \ldots, \hat{x}_{L+T}] \in \mathbb{R}^T$ represents predicted T-steps future values. The optimization objective is to minimize the discrepancy between predicted values and actual future values over time.

3.2 Overall Architecture

The overall framework of LeRet is illustrated in Figure 2. Initially, a multivariate time series is decomposed into multiple univariate time series, which are then treated independently [Nie *et al.*, 2023]. This transforms the task of multivariate time series forecasting into a set of univariate time series forecasting tasks.

Subsequently, for a time series $X_{\text{input}} \in \mathbb{R}^L$, We partition it into non-overlapping patches of length P, resulting in a total of $N = \left\lfloor \frac{L}{P} \right\rfloor + 1$ input patches $X_{\text{patch}} \in \mathbb{R}^{N \times P}$. These patches are embedded as $X_{\text{pe}} \in \mathbb{R}^{N \times d_p}$ using a simple linear layer:

$$X_{\text{pe}} = \text{Linear}(\text{Reshape}(X_{\text{input}})). \tag{1}$$

Based on X_{pe}, we firsly apply Retentive Network (RetNet) [Sun *et al.*,] as a Causal Dependence Learner (CDL) to encode features of the patched time series, remaining temporal causal nature and obtaining features $H \in R^{N \times d_m}$. It can be given by:

$$H = \text{CausalLearner}(X_{\text{pe}}). \tag{2}$$

In the causal dependence representation, previous patch features is independent of next patch. Thus, conducting a patch-level autoregressive task as pre-training helps the model understand causal growth patterns in the time series by predicting the next patch's values based on the preceding patch.

Moreover, in TS-Related Language Knowledge Extractor, we input K text descriptions of fundamental time series features TDs, such as ***trend, period, stability, and noise level***, into a pre-trained Large Language Model (LLM) to extract language knowledge related to the time series, denoted as:

$$\delta = \text{LLM-Extractor}(TDs), \tag{3}$$

where $\delta = \{S_1, S_2, ..., S_K\} \in \mathbb{R}^{K \times d_s}$ is the extracted time series related text embedding and K is the number of texts.

Subsequently, through the Language-TS Modality Integrator (LTMI), text embedding are firstly mapped to the time series space, and the fusion of language knowledge and time series produces language-empowered sequential features $Z \in \mathbb{R}^{N \times d_m}$:

$$Z = \text{Integrator}(\delta, H). \tag{4}$$

LeRet employs sequence forecasting with patch-level enhanced. (a) Patch-level Pre-training. LeRet predict the temporal values of the next patch based on the causal temporal features $H \in \mathbb{R}^{N \times d_m}$ to pre-train Causal Dependent Learner; (b) Sequence-level Forecasting. The language-empowered sequential features $Z \in \mathbb{R}^{N \times d_m}$ is finally used to predict T future time steps.

3.3 Causal Dependence Learner

To preserve the inherent causal dependence of time series and improve computational efficiency, LeRet utilizes the Retentive Network (RetNet) [Sun *et al.*,] as the backbone of Causal Dependence Learner.

Retention Mechanism The input of model is $X_{\text{pe}} = \{x_{\text{pe}}^1, x_{\text{pe}}^2, ..., x_{\text{pe}}^N\}$. Without employing bidirectional attention, RetNet utilizes a retention mechanism for sequence modeling. We use $q_n = W_q x_{\text{pe}}^n \in$, $k_m = W_k x_{\text{pe}}^m$ and $v_m = W_v x_{\text{pe}}^m$ as query, key and value of corresponding patch embedding, respectively. Denote o_n as the output feature of x_{pe}^n by rentention metchanism. It can be expressed as:

$$o_n = \sum_{m=1}^{n} \left(q_n \left(\gamma e^{i\theta}\right)^n\right) \left(k_m \left(\gamma e^{i\theta}\right)^{-m}\right)^\top v_m, \tag{5}$$

where $q_n \left(\gamma e^{i\theta}\right)^n, k_m \left(\gamma e^{i\theta}\right)^{-m}$ is known as xPos [Sun *et al.*, 2022], i.e., a relative position embedding proposed for Transformer. We further simplify γ as a pr-defined scalar to formulate Eq(5) becomes:

$$o_n = \sum_{m=1}^{n} \gamma^{n-m} \left(q_n e^{in\theta}\right) \left(k_m e^{im\theta}\right)^\dagger v_m, \tag{6}$$

where γ serves as a pre-defined decay factor, replacing the initial calculation of the attention map, and † denotes the conjugate transpose. The formulation is easily parallelizable within training instances and can be represented as:

$$\begin{gathered} Q = (X_{\text{pe}} W_q) \odot \Theta, \quad K = (X_{\text{pe}} W_k) \odot \bar{\Theta}, \quad V = X_{\text{pe}} W_v \\ \Theta_n = e^{in\theta}, \quad D_{nm} = \begin{cases} \gamma^{n-m}, & n \geq m \\ 0, & n < m \end{cases} \\ \text{Retention}\left(X_{\text{pe}}\right) = \left(QK^\top \odot D\right) V, \end{gathered} \tag{7}$$

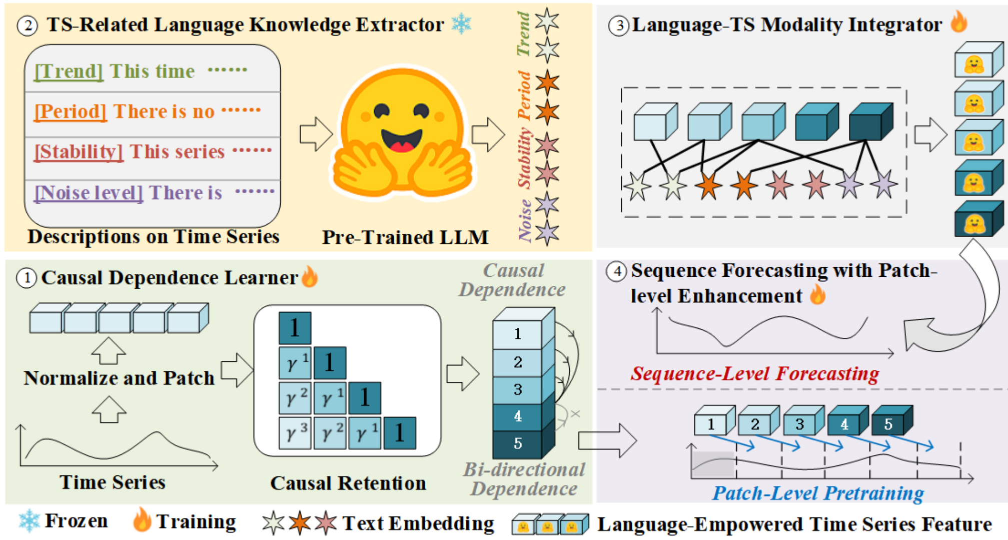

Figure 2: LeRet involves four key steps. ① Divide time series into patches, and a causal dependence learner with ④ patch-level pre-training is applied to obtain causal dependence representation. ② Extract time series related language knowledge from LLM. ③ Language-TS Modality Integrator for alignment between language representations and numerical time series. ④ Project language representation to make sequence forecasting.

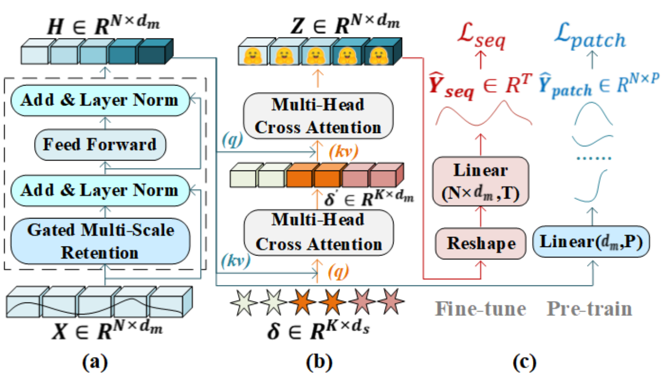

Figure 3: The forward propagation details of LeRet, encompassing time series feature extraction, modality fusion and prediction.

where $\bar{\Theta}$ is the complex conjugate of Θ, and $D \in \mathbb{R}^{N\times N}$ combines causal masking and pre-defined exponential decay. Similar to self-attention, the parallel representation enables us to train the models with GPUs efficiently.

Gated Multi-scale Retention The model employs $h = d_{\mathrm{m}}/d$ retention heads in each layer, where d is the head dimension. Multi-scale retention (MSR) assigns different γ for each head, adding a swish gate to increase non-linearity. The layer of Multi-scale Retention (MSR) is defined as:

$$\gamma = 1 - 2^{-5-\text{arange}(0,h)} \in \mathbb{R}^h \tag{8}$$

$$\text{head}_i = \text{Retention}(X_{\text{pe}}, \gamma_i) \tag{9}$$

$$Y = \text{GN}_h\left(\text{Concat}(\text{head}_1, \dots, \text{head}_h)\right) \tag{10}$$

$$\text{MSR}(X_{\text{pe}}) = \left(\text{SG}(X_{\text{pe}}W_G) \odot Y\right) W_O \tag{11}$$

Here, $W_G, W_O \in \mathbb{R}^{d_{\mathrm{m}}\times d_{\mathrm{m}}}$ are learnable parameters. GN, Concat and SG are group normalization, concatenation and swish gate, respectively.

Causal Dependence Representation The forward propagation process of CDL is illustrated in Figure 3(a), which includes the MSR layer, feed forward layer and residual operation to enhance the fitting ability for extracting time series features. The entire forward process is expressed as,

$$X_{\text{msr}} = \text{LN}(\text{MSR}(X_{\text{pe}})) + X_{\text{pe}}, \tag{12}$$

$$H = \text{LN}(\text{FF}(X_{\text{msr}})) + X_{\text{msr}}, \tag{13}$$

where LN and FF are normalization layer and feed forward layer. $H \in \mathbb{R}^{N\times d_m}$ is the final output of CDL, capturing temporal causal dependencies, where each subsequent time series patch can only attend to the ones preceding it.

3.4 TS-Related Language Knowledge Extractor

The extensive language knowledge in LLM is redundant, and it is difficult to directly form targeted knowledge related to time series based on LLM. Additionally, using LLM during as a part of forecasting consumes significant computational and storage resources in both training and inference phases. Therefore, we propose TS-Related Language Knowledge Extractor, which efficiently extracts language knowledge about time series from LLM.

Language Description Protocol for Time Series As shown in Table 1, we select four significant time series characteristics (i.e., trend, period, stability, and noise level) and form several text descriptions based on these features. Formally, we represent a collection of K text descriptions about time series characteristics as $TD = \{td_1, td_2, ..., td_K\}$, where $td_i (1 \le i \le K)$ is an independent text with a length of θ_i.

Series Knowledge Representation from Natural Language For the description td_i, we first input it into the LLM for feature encoding, obtaining the text embedding

Characteristics	Text Descriptions
Trend	① This time series exhibits an overall declining trend . ② This time series shows an overall upward trend .
Period	③ There is no apparent periodicity in this time series . ④ This time series displays clear periodicity .
Stability	⑤ The time series remains relatively stable with minimal fluctuations . ⑥ The time series undergoes significant instability over a period .
Noise	⑦ This time series is subject to strong noise interference . ⑧ This time series is not influenced by any noise interference .

Table 1: Time Series Related Text Descriptions

$te_i \in \mathbb{R}^{(\theta_i+2)\times d_s}$, where d_s is the feature dimension of each token, and $\theta_i + 2$ is the number of tokens (adding start token [BOS] and end token [EOS] to the original segmentation). Since we choose LLaMa as the LLM, which is a decoder-only architecture, under this causal encoding, each token can only perceive itself and the tokens before it. As only the last state can store all the information of the sentence, we select the embedding of the [EOS] token of each text embedding as the extracted time series related language knowledge, denoted as $\delta = \{S_1, S_2, ..., S_K\} \in \mathbb{R}^{K\times d_s}$.

3.5 Language-TS Modality Integrator

As language knowledge γ and time series features H belong to two different and distinct feature spaces, directly feeding language knowledge into the time series forecasting model is not feasible. This would make it challenging for the model to understand the originally captured temporal patterns, increasing the difficulty in fitting forecasting task. In the Language-TS Modality Integrator, we design a two-stage modality fusion to achieve language knowledge empowered time series as shown in Figure 3(b).

In the first stage, we need to map language knowledge to the time series feature space. An easy-to-implement method is cross-attention, allowing language knowledge to adaptively aggregate time series features, forming language knowledge expressed in the time series feature:

$$\delta' = \mathrm{CrossAttention}\left(Q_\delta, K_H, V_H\right), \tag{14}$$

where $Q_\delta = \delta \cdot W_\delta^Q$, $K_H = H \cdot W_H^K$, $V_H = H \cdot W_H^V$, are language and time series features linearly mapped, and $\delta' \in \mathbb{R}^{K\times d_m}$ is the mapping of language features into the time series feature space.

In the second stage, we need to integrate this aligned language knowledge δ' with time series features H:

$$Z = \mathrm{CrossAttention}\left(Q_H, K_{\delta'}, V_{\delta'}\right) \tag{15}$$

where $Q_H = \delta \cdot W_\delta^H$, $K_{\delta'} = \delta' \cdot W_{\delta'}^K$, $V_{\delta'} = \delta' \cdot W_{\delta'}^V$, are language and time series features linearly mapped, $Z \in \mathbb{R}^{K\times d_m}$ represents the language-empowered time series features.

3.6 Sequence Forecasting with Patch-level Enhancement

Patch-Level Pre-training To enhance the model in understanding the causal evolution of time series, we devise a patch-level forecasting as a pre-training task to warm up Causal Dependence Learner. For example, given an input sequence of patches such as the 1st patch, 2nd patch, 3rd patch, this task is expected to generate outputs corresponding to the 2nd patch, 3rd patch, 4th patch based on the preceding patch. Since the language-empowered Z obtained after modality fusion may disrupt the causality of time series features, we use causal temporal features H to make patch-level forecasting :

$$Y_{patch} = \mathrm{Head_P}(H), \tag{16}$$

where $Y_{patch} \in \mathbb{R}^{N\times P}$ represents the patch-level predicted time series values, and $\mathrm{Head_P}$ is a linear layer as patch-level pre-training head.

Sequence-Level Forecasting For sequence-level forecasting, we use language-empowered time series features Z to make predictions:

$$Y_{seq} = \mathrm{Head_S}(Z), \tag{17}$$

where $Y_{seq} \in \mathbb{R}^T$ is the forecasting results for the future L steps, and $\mathrm{Head_S}$ is the prediction head consisting of a reshape block and a linear layer.

4 Experiments

4.1 Datasets and Experimental Setups

We evaluate performance of long-term forecasting on **Weather**, **Traffic**, **Solar**, **Electricity** and four **ETT** datasets (i.e., ETTh1, ETTh2, ETTm1, and ETTm2), which have been extensively adopted for benchmarking long-term forecasting models. The input time series length L is set as 336 for all baselines, and we use four different prediction horizons $T \in \{96, 192, 336, 720\}$. For short-term forecasting, we adopt the **PeMS** which contains four public traffic network datasets (PEMS03, PEMS04, PEMS07, PEMS08). All the models are following the same experimental setup with input length $L = 96$ and prediction length $T = 12$.

4.2 Main Results

Long-term Forecasting

Our results are presented in Table 2, where LeRet demonstrates superior performance across different prediction length again all baselines. Quantitatively, LeRet achieves 57 first-place and 5 second-place rankings out of 64 settings. In contrast to the effective linear model DLinear, LeRet

Models		LeRet		ModernTCN		LLM4TS		GPT4TS		PatchTST		DLinear		Crossformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh2	96	**0.247**	**0.321**	0.263	0.332	0.269	0.332	0.285	0.342	0.274	0.336	0.289	0.353	0.628	0.563
	192	**0.301**	**0.358**	0.320	0.374	0.328	0.377	0.354	0.389	0.339	0.379	0.383	0.418	0.703	0.624
	336	0.320	**0.373**	**0.313**	0.376	0.353	0.396	0.373	0.407	0.331	0.380	0.448	0.465	0.827	0.675
	720	0.384	0.428	0.392	0.433	0.383	0.425	0.406	0.441	**0.379**	**0.422**	0.605	0.551	1.181	0.840
ETTh1	96	**0.355**	**0.382**	0.368	0.394	0.371	0.394	0.376	0.397	0.375	0.399	0.375	0.399	0.386	0.429
	192	**0.375**	**0.394**	0.405	0.413	0.403	0.412	0.416	0.418	0.414	0.421	0.405	0.416	0.419	0.444
	336	**0.381**	**0.404**	0.391	0.412	0.420	0.422	0.442	0.433	0.431	0.436	0.439	0.443	0.440	0.461
	720	**0.420**	**0.438**	0.450	0.461	0.422	0.444	0.477	0.456	0.449	0.466	0.472	0.490	0.519	0.524
ETTm1	96	**0.283**	**0.332**	0.292	0.346	0.285	0.343	0.292	0.346	0.290	0.342	0.299	0.343	0.316	0.373
	192	0.325	**0.357**	0.332	0.368	**0.324**	0.366	0.332	0.372	0.332	0.369	0.335	0.365	0.377	0.411
	336	**0.349**	**0.378**	0.365	0.391	0.353	0.385	0.366	0.394	0.366	0.392	0.369	0.386	0.431	0.442
	720	0.411	**0.411**	0.416	0.417	**0.408**	0.419	0.417	0.421	0.420	0.424	0.425	0.421	0.600	0.547
ETTm2	96	**0.161**	**0.250**	0.166	0.256	0.165	0.254	0.173	0.262	0.165	0.255	0.167	0.260	0.421	0.461
	192	**0.219**	**0.288**	0.222	0.293	0.220	0.292	0.229	0.301	0.220	0.292	0.224	0.303	0.503	0.519
	336	**0.261**	**0.320**	0.272	0.324	0.268	0.326	0.286	0.341	0.278	0.329	0.281	0.342	0.611	0.580
	720	**0.340**	**0.371**	0.351	0.381	0.350	0.380	0.378	0.401	0.367	0.385	0.397	0.421	0.996	0.750
Traffic	96	**0.356**	**0.248**	0.368	0.253	0.372	0.259	0.388	0.282	0.367	0.251	0.410	0.282	0.512	0.290
	192	**0.375**	**0.255**	0.379	0.261	0.391	0.265	0.407	0.290	0.385	0.259	0.423	0.287	0.523	0.297
	336	**0.384**	**0.263**	0.397	0.270	0.405	0.275	0.412	0.294	0.398	0.265	0.436	0.296	0.530	0.300
	720	**0.428**	**0.286**	0.440	0.296	0.437	0.292	0.450	0.312	0.434	0.287	0.466	0.315	0.573	0.313
Electricity	96	0.129	**0.220**	0.129	0.226	**0.128**	0.223	0.139	0.238	0.130	0.222	0.140	0.237	0.187	0.283
	192	**0.141**	**0.238**	0.143	0.239	0.146	0.240	0.153	0.251	0.148	0.240	0.153	0.249	0.258	0.330
	336	**0.160**	**0.255**	0.161	0.259	0.163	0.258	0.169	0.266	0.167	0.261	0.169	0.267	0.323	0.369
	720	**0.188**	0.288	0.191	**0.286**	0.200	0.292	0.206	0.297	0.202	0.291	0.203	0.301	0.404	0.423
Weather	96	**0.144**	**0.185**	0.149	0.200	0.147	0.196	0.162	0.212	0.152	0.199	0.176	0.237	0.153	0.217
	192	**0.189**	**0.228**	0.196	0.245	0.191	0.238	0.204	0.248	0.197	0.243	0.220	0.282	0.197	0.269
	336	**0.225**	**0.261**	0.238	0.277	0.241	0.277	0.254	0.286	0.249	0.283	0.265	0.319	0.252	0.311
	720	**0.300**	**0.313**	0.314	0.334	0.313	0.329	0.326	0.337	0.320	0.335	0.323	0.362	0.318	0.363
Solar	96	**0.175**	**0.231**	0.223	0.285	0.209	0.271	0.215	0.268	0.224	0.278	0.289	0.377	0.181	0.240
	192	**0.195**	**0.248**	0.250	0.294	0.231	0.274	0.250	0.279	0.253	0.298	0.319	0.397	0.196	0.252
	336	**0.196**	**0.238**	0.286	0.288	0.269	0.281	0.262	0.287	0.273	0.306	0.352	0.415	0.216	0.243
	720	**0.201**	**0.255**	0.272	0.294	0.262	0.289	0.264	0.293	0.272	0.298	0.356	0.412	0.220	0.256

Table 2: Long-term forecasting results. Forecasting horizons $T \in \{96, 192, 336, 720\}$, and input length L is set as 336. A lower value indicates better performance. **Red**: the best, Blue: the second best.

achieves performance gains of 21.3% and 18.3% in MSE and MAE metrics. Compared to the state-of-the-art task-specific TSF model ModernTCN, LeRet exhibits a relative reduction of 7.6% and 5.6% in MSE and MAE metrics, respectively. When compared with the cutting-edge LLM-based TSF model LLM4TS, LeRet exhibits superiority in 59 out of 64 experimental settings, with a performance improvement of 7.1% and 5.0% in MSE and MAE metrics, respectively.

Short-term Forecasting

As illustrated in Table 3, for short-term forecasting, LeRet consistently maintains a leading predictive performance. Compared to SOTA short-term forecasting model SCINet, LeRet achieves significant reductions in MAE, MAPE, and RMSE, respectively. The comprehensive experimental results underscore the efficacy of LeRet in short-term forecasting.

Few-shot Learning

In few-shot learning, only 10% of the training data timesteps are utilized, and the outcomes are presented in Table 5. Evidently, LLM-based methods outperform other benchmark TSF models. Quantitativly, LeRet achieves an average 4.4% reduction in MSE and 2.1% reduction in MAE compared to the top-performing LLM4TS.

Zero-shot Learning

This task is to evaluate how effectively a model can perform on *target dataset* when it has been trained on *source dataset*, and the results are presented in Table 4. LeRet outperforms all state-of-the-art models, achieving a performance improvement of over 10% compared to other models.

4.3 Ablation Study

To assess the effectiveness of each component in LeRet, we conduct a comprehensive ablation study on Causal Dependence Learner (CDL), Time-Language Modality Integrator (TLMI), Patch-level Pre-training (PLP) and TS-Related Language Knowledge Extractor (TRLE). In the corresponding ablations, CDL is replaced with self-attention, TLMI is substituted with a simple concatenation operation, PLP is re-

Models		LeRet	SCINet	ModernTCN	LLM4TS	GPT4TS	PatchTST	DLinear	Crossformer	MICN	TimesNet
PEMS	MAE	**18.34**	19.12	22.74	22.07	22.46	23.01	23.31	19.23	19.34	20.54
	MAPE	**11.89**	12.24	14.48	14.04	14.67	14.95	14.68	12.22	12.38	12.69
	RMSE	**29.12**	30.12	35.54	35.05	35.46	36.05	37.32	30.17	30.40	33.25

Table 3: Short-term forecasting results. The results are averaged across PEMS03, PEMS04,PEMS07 and PEMS08. All input lengths are 96 and prediction lengths are 12. A lower MAE, MAPE or RMSE indicates a better prediction.

Methods	LeRet		LLM4TS		GPT4TS		DLinear		ModernTCN	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.574**	**0.524**	0.592	0.531	0.590	0.525	0.691	0.600	0.613	0.522
ETTh2	**0.393**	**0.420**	0.402	0.426	0.397	0.421	0.605	0.538	0.410	0.430

Table 4: Zero-shot learning results. The results are averaged on different prediction lengths.

Methods		LeRet		LLM4TS		GPT4TS		DLinear		ModernTCN	
Source	Target	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	ETTh2	**0.304**	**0.358**	0.379	0.400	0.406	0.422	0.493	0.488	0.380	0.405
ETTh2	ETTh1	**0.437**	**0.440**	0.548	0.502	0.757	0.578	0.703	0.574	0.556	0.512

Table 5: Few-shot learning on 10% training data. The results are averaged on different prediction lengths.

CDE	TLMI	PLP	TRLE	Solar MSE	Solar MAE
✓	✓	✓	✓	0.192	0.243
×	✓	×	✓	0.210	0.259
✓	×	✓	✓	0.195	0.251
✓	✓	×	✓	0.198	0.250
✓	×	✓	×	0.201	0.251

Table 6: Ablation study on component variants.

moved, and excluding TRLE prevents the model from receiving knowledge from the language modality. Our observations from Table 6 are as follows: Obs.1) The combination of CDL and PLP is necessary. Removing these two modules results in approximately a 8.6% decrease in performance; Obs.2) Integrating TRLE with TLMI contributes to the model's understanding of temporal features, resulting in an average performance improvement of 5.7%.

5 Discussion

Compared to other LLM-based TSF models, LeRet exhibits significant advantages in interpretability and parameter scales. Actually, we focus on ensuring the interpretability of using LLM (i.e., the interpretability of the model input). Usually, LLM-based TSF models directly fine-tune pre-trained language models with time series as inputs, achieving good predictive performance but lacking explanation for why discrete texts can be arbitrarily replaced by continuous numerical values. In contrast, LeRet enhances the interpretability of LLM utilization by firstly mapping text embeddings to the time series feature space, and then integrating the aligned text features with time series features for language-empowered forecasting. In terms of parameter scales, compared to other LLM-based models like LLM4TS with 7B parameters per forward pass, our LeRet requires only 0.15M parameters during both training and inference. This reduction is achieved by utilizing pre-trained LLM solely for ts-related text embeddings. TS-related text embeddings can be stored in memory for quick retrieval, without involving LLM in training, significantly reducing the parameter scales by several thousand times.

6 Conclusion

We propose a language-empowered time-series learning framework, LeRet, which inherits the structure of sequential causality extraction from LLMs and exploits pre-trained language knowledge for effective and semantic interpretable series forecasting. We introduce a novel paradigm for LLM-based methods. Empirical evaluations reveal the effectiveness of our LeRet, especially show superiority in few- and zero-shot scenarios.

Acknowledgements

This paper is partially supported by the National Natural Science Foundation of China (No.62072427, No.12227901), the Project of Stable Support for Youth Team in Basic Research Field, CAS (No.YSBR-005), Academic Leaders Cultivation Program, USTC, and the grant from State Key Laboratory of Resources and Environmental Information System.

References

[Cao *et al.*, 2023] Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*, 2023.

[Chang *et al.*, 2023] Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *arXiv preprint arXiv:2308.08469*, 2023.

[Ding *et al.*, 2023] Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. Sparse low-rank adaptation of pre-trained language models. *arXiv preprint arXiv:2311.11696*, 2023.

[Gruver *et al.*, 2023] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. *arXiv preprint arXiv:2310.07820*, 2023.

[Hu *et al.*, 2022] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[Huang *et al.*, 2024a] Qihe Huang, Lei Shen, Ruixin Zhang, Jiahuan Cheng, Shouhong Ding, Zhengyang Zhou, and Yang Wang. Hdmixer: Hierarchical dependency with extendable patch for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12608–12616, 2024.

[Huang *et al.*, 2024b] Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. *Advances in Neural Information Processing Systems*, 36, 2024.

[Jin *et al.*, 2023] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

[Lin *et al.*, 2023a] Shengsheng Lin, Weiwei Lin, Wentai Wu, Songbo Wang, and Yongxiang Wang. Petformer: Long-term time series forecasting via placeholder-enhanced transformer. 2023.

[Lin *et al.*, 2023b] Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. Segrnn: Segment recurrent neural network for long-term time series forecasting. *arXiv preprint arXiv:2308.11200*, 2023.

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

[Liu *et al.*, 2021] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.

[Luo and Wang, 2024] Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *The Twelfth International Conference on Learning Representations*, 2024.

[Miao *et al.*, 2024] Hao Miao, Yan Zhao, Chenjuan Guo, Bin Yang, Zheng Kai, Feiteng Huang, Jiandong Xie, and Christian S Jensen. A unified replay-based continuous learning framework for spatio-temporal prediction on streaming data. In *ICDE*, 2024.

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

[Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[Shabani *et al.*, 2023] Amin Shabani, Amir Abdi, Lili Meng, and Tristan Sylvain. Scaleformer: Iterative multi-scale refining transformers for time series forecasting. *The Eleventh International Conference on Learning Representations*, 2023.

[Sun *et al.*,] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models (2023). *URL http://arxiv.org/abs/2307.08621 v1.*

[Sun *et al.*, 2022] Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer. *arXiv preprint arXiv:2212.10554*, 2022.

[Tian *et al.*, 2021] Hao Tian, Xiaolong Xu, Lianyong Qi, Xuyun Zhang, Wanchun Dou, Shui Yu, and Qiang Ni. Copace: Edge computation offloading and caching for self-driving with deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 70(12):13281–13293, 2021.

[Touvron *et al.*, 2023] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[Wang *et al.*, 2021a] Pengkun Wang, Chuancai Ge, Zhengyang Zhou, Xu Wang, Yuantao Li, and Yang Wang. Joint gated co-attention based multi-modal networks for subregion house price prediction. *IEEE Transactions on Knowledge and Data Engineering*, 35(2):1667–1680, 2021.

[Wang *et al.*, 2021b] Senzhang Wang, Hao Miao, Jiyue Li, and Jiannong Cao. Spatio-temporal knowledge transfer for urban crowd flow prediction via deep attentive adaptation networks. *TITS*, 23(5):4695–4705, 2021.

[Wang *et al.*, 2023a] Binwu Wang, Yudong Zhang, Jiahao Shi, Pengkun Wang, Xu Wang, Lei Bai, and Yang Wang. Knowledge expansion and consolidation for continual traffic prediction with expanding graphs. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[Wang *et al.*, 2023b] Binwu Wang, Yudong Zhang, Xu Wang, Pengkun Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. Pattern expansion and consolidation on evolving graphs for continual traffic prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2223–2232, 2023.

[Wang *et al.*, 2023c] Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023.

[Wang *et al.*, 2023d] Xu Wang, Hongbo Zhang, Pengkun Wang, Yudong Zhang, Binwu Wang, Zhengyang Zhou, and Yang Wang. An observed value consistent diffusion model for imputing missing values in multivariate time series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2409–2418, 2023.

[Wang *et al.*, 2024] Binwu Wang, Pengkun Wang, Yudong Zhang, Xu Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. Towards dynamic spatial-temporal graph learning: A decoupled perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9089–9097, 2024.

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

[Wu *et al.*, 2023] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

[Yang *et al.*, 2023] Kuo Yang, Zhengyang Zhou, Wei Sun, Pengkun Wang, Xu Wang, and Yang Wang. Extract and refine: Finding a support subgraph set for graph representation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2953–2964, 2023.

[Yu *et al.*, 2023] Xinli Yu, Zheng Chen, Yuan Ling, Shujing Dong, Zongyi Liu, and Yanbin Lu. Temporal data meets llm–explainable financial time series forecasting. *arXiv preprint arXiv:2306.11025*, 2023.

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

[Zhang *et al.*, 2022] Yudong Zhang, Binwu Wang, Ziyang Shan, Zhengyang Zhou, and Yang Wang. Cmt-net: A mutual transition aware framework for taxicab pick-ups and drop-offs co-prediction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1406–1414, 2022.

[Zhang *et al.*, 2024] Yudong Zhang, Pengkun Wang, Binwu Wang, Xu Wang, Zhe Zhao, Zhengyang Zhou, Lei Bai, and Yang Wang. Adaptive and interactive multi-level spatio-temporal network for traffic forecasting. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–17, 2024.

[Zhao *et al.*, 2024a] Zhe Zhao, Pengkun Wang, Xu Wang, Haibin Wen, Xiaolong Xie, Zhengyang Zhou, Qingfu Zhang, and Yang Wang. Delayed bottlenecking: Alleviating forgetting in pre-trained graph neural networks. *arXiv preprint arXiv:2404.14941*, 2024.

[Zhao *et al.*, 2024b] Zhe Zhao, Pengkun Wang, Haibin Wen, Yudong Zhang, Zhengyang Zhou, and Yang Wang. A twist for graph classification: Optimizing causal information flow in graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17042–17050, 2024.

[Zhou *et al.*, 2020a] Zhengyang Zhou, Yang Wang, Xike Xie, Lianliang Chen, and Hengchang Liu. Riskoracle: A minute-level citywide traffic accident forecasting framework. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1258–1265, 2020.

[Zhou *et al.*, 2020b] Zhengyang Zhou, Yang Wang, Xike Xie, Lianliang Chen, and Chaochao Zhu. Foresee urban sparse traffic accidents: A spatiotemporal multi-granularity perspective. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3786–3799, 2020.

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

[Zhou *et al.*, 2022] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. 39th International Conference on Machine Learning (ICML 2022)*, 2022.

[Zhou *et al.*, 2023a] Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained lm. 2023.

[Zhou *et al.*, 2023b] Zhengyang Zhou, Qihe Huang, Gengyu Lin, Kuo Yang, LEI BAI, and Yang Wang. GReto: Remedying dynamic graph topology-task discordance via target homophily. In *The Eleventh International Conference on Learning Representations*, 2023.