ChatTime: A Unified Multimodal Time Series Foundation Model Bridging Numerical and Textual Data

Chengsen Wang[1]*, Qi Qi[1]*, Jingyu Wang[1,2]†,
Haifeng Sun[1], Zirui Zhuang[1], Jinming Wu[1], Lei Zhang[3], Jianxin Liao[1]
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Pengcheng Laboratory, Shenzhen, China
[3]China Unicom Network Communications Corporation Limited, Beijing, China
cswang@bupt.edu.cn, qiqi8266@bupt.edu.cn, wangjingyu@bupt.edu.cn

Abstract

Human experts typically integrate numerical and textual multimodal information to analyze time series. However, most traditional deep learning predictors rely solely on unimodal numerical data, using a fixed-length window for training and prediction on a single dataset, and cannot adapt to different scenarios. The powered pre-trained large language model has introduced new opportunities for time series analysis. Yet, existing methods are either inefficient in training, incapable of handling textual information, or lack zero-shot forecasting capability. In this paper, we innovatively model time series as a foreign language and construct ChatTime, a unified framework for time series and text processing. As an out-of-the-box multimodal time series foundation model, ChatTime provides zero-shot forecasting capability and supports bimodal input/output for both time series and text. We design a series of experiments to verify the superior performance of ChatTime across multiple tasks and scenarios, and create four multimodal datasets to address data gaps. The experimental results demonstrate the potential and utility of ChatTime.

Code — https://github.com/ForestsKing/ChatTime

1 Introduction

Time series data is common in various fields, and its accurate forecasts are vital for decision support in industries such as finance (He, Siu, and Si 2023), transportation (He et al. 2022), energy (Pinto et al. 2021), healthcare (Puri et al. 2022), and climate (Du et al. 2021). Human experts frequently integrate multimodal information for time series forecasting. For instance, economists combine historical financial series with policy reports to predict future market trends. Due to their remarkable performance, deep learning predictors (Nie et al. 2023; Cai et al. 2024) have become the mainstream method in recent years. However, most current deep paradigms train and predict on a single dataset based on fixed history and prediction windows, lacking adaptability to different scenarios or datasets. Additionally, most existing methods utilize only unimodal numerical data. Recent studies have demonstrated that simple linear models (Zeng et al. 2023; Li et al. 2023b) often rival the performance of state-of-the-art (SOTA) complex models, indicating that current unimodal approaches may be nearing a saturation point.

Method	Zero-Shot Forecast	Missing Support	Training Token	Trainable Parameter
TimesFM	✓	✗	3T	200M
Moirai	✓	✓	150B	300M
TimeGPT	✓	✓	100B	Unknown
MOMENT	✗	✓	100B	300M
Timer	✗	✗	50B	50M
Chronos	✓	✓	25B	700M
ChatTime	✓	✓	1B	350M

Table 1: The comparison between pre-trained time series foundation models.

Meanwhile, the rapid advancement of pre-trained large language models (LLM) has garnered significant attention (Touvron et al. 2023a,b). Through autoregressive pre-training on vast amounts of text, these robust tools are capable of performing a wide array of tasks in a zero-shot learning paradigm. This has spurred interest in incorporating LLMs into time series analysis. Some works (Ansari et al. 2024; Das et al. 2024) have utilized extensive time series data to construct time series foundational models, which can handle the forecasting task across any scenario with a single model. However, the training-from-scratch strategy renders them highly inefficient and forfeits the ability to process textual information. Other research (Jin et al. 2024; Xu et al. 2024) has attempted to integrate the weights of pre-trained LLMs into a new time series forecasting framework. They fine-tune additional input and output layers to consider both time series and textual information. Nevertheless, these additional layers are incapable of zero-shot learning and require re-fine-tuning for each dataset. Furthermore, the inability to output text hindered the aforementioned paradigms in addressing scenarios such as time series question answering and summarization. This motivates the question: *Is it possible to construct a multimodal time series foundation model that allows for zero-shot inference and supports both time series and textual bimodal inputs and outputs?*

Linguistic models for predicting the next word and time series models for predicting the next value fundamentally

*Equal contribution.
†Corresponding author.

model the sequential structure of historical data to predict future patterns. At the core of both is an n-order Markov process (Zhou et al. 2023). In this work, we innovatively conceptualize time series as a foreign language and construct ChatTime, an out-of-the-box multimodal time series foundation model, as a framework for the unified processing of time series and text. ChatTime converts continuous unbounded time series into a finite set of discrete values through normalization and discretization, and then characterizes them as foreign language words by adding mark characters. We employ continuous pre-training and instruction fine-tuning for the pre-trained LLM using the same methodology as vocabulary expansion (Csaki et al. 2024; Kim, Choi, and Jeong 2024), eliminating the need to train from scratch or alter the model architecture. Compared to other foundation models, as shown in Table 1, we not only significantly reduce the training cost but also gain an additional inference capability to process textual information. This simple yet effective approach addresses a wide range of time series problems at minimal cost, paving the way for further leveraging the findings of LLMs and multimodal communities in the future.

To comprehensively evaluate the performance of ChatTime, we design a series of experiments including three main tasks: zero-shot time series forecasting (ZSTSF), context-guided time series forecasting (CGTSF), and time series question answering (TSQA). These tasks examine the modal translation capabilities of the foundational model for time series to time series, text to time series, and time series to text, respectively. Alongside the text-to-text inference capability of the pre-trained LLM itself (OpenAI 2023a), ChatTime achieves seamless input and output of both time series and text modalities. The zero-shot time series forecasting task is evaluated on eight real-world benchmark datasets across four domains, which are commonly used (Wu et al. 2021) for long-term time series forecasting. For the multimodal context-guided time series forecasting task, we collect time series records from three different scenarios, adding and aligning background, weather, and date information without any leakage of future information. Regarding the multimodal time series question answering task, we synthesize a variable-length question answering dataset covering four typical time series features (Fons et al. 2024). The experimental results confirm the superior performance of ChatTime in multiple tasks and scenarios, highlighting its potential as a multimodal time series foundation model.

In summary, we present the following contributions:

- We construct ChatTime, a multimodal time series foundation model, by conceptualizing time series as a foreign language. It allows for zero-shot inference and supports both time series and textual bimodal inputs and outputs.
- We establish three context-guided time series forecasting datasets and a time series question answering dataset to fill gaps in related multimodal domains, offering valuable resources for future research.
- We demonstrate the considerable advantages of ChatTime across multiple time series tasks through comprehensive experiments, offering innovative perspectives and solutions for time series analysis.

2 Related Work

2.1 Long-Term Time Series Forecasting

As a significant real-world challenge, time series forecasting has garnered considerable attention. Initially, ARIMA (Box and Jenkins 1968) performs forecasts in a moving average manner. However, the complex real world often renders such statistical methods challenging to adapt. With the development of deep learning, neural network-based methods have become increasingly important. Recurrent neural networks (Hochreiter and Schmidhuber 1997; Flunkert, Salinas, and Gasthaus 2017) dynamically capture temporal dependencies within a sequential structure. Unfortunately, this architecture suffers from gradient vanishing/exploding and information forgetting. To further improve prediction performance, convolutional networks (Wang et al. 2023; Wu et al. 2023) and self-attention mechanisms (Zhou et al. 2021; Liu et al. 2024a) have been introduced to capture long-range dependencies. Despite achieving impressive performance, most current deep paradigms lack adaptability to different scenarios and utilize only unimodal numerical data.

2.2 LLM-Based Time Series Analysis

The rise of pre-trained LLMs has introduced new opportunities for time series analysis. Based on the dependence on pre-training weights, these works can be broadly categorized into the following three paradigms.

The first category of work relies entirely on pre-trained weights. They (Gruver et al. 2023) employ LLMs directly for time series forecasting via prompts. Due to the lack of understanding (Fons et al. 2024) about time series features, their prediction accuracy is typically too low (Merrill et al. 2024). These methods also have low token utilization due to the bit-by-bit tokenization. Instruction fine-tuning has improved accuracy in some cases (Guo et al. 2024), but these improvements do not address high inference costs.

The second category of work integrates pre-training weights into new frameworks. Additional neural layers will be fine-tuned to adapt for the time series. Some studies (Zhou et al. 2023; Jin et al. 2024) use pre-trained weights as the backbone and incorporate extra input and output layers, significantly enhancing prediction performance. Others (Xu et al. 2024; Jia et al. 2024) utilize pre-trained weights as an embedding module to enable the reception of context. However, most of them cannot perform zero-shot inference.

The third category of work uses the architecture of pre-trained LLMs but does not utilize the weights. They (Garza and Canseco 2023; Ansari et al. 2024; Das et al. 2024; Woo et al. 2024; Goswami et al. 2024; Liu et al. 2024b) employ vast amounts of time series data to construct new foundation models. While yielding promising results, training from scratch is highly inefficient, and most of these models support only unimodal numerical data.

Some studies have explored the multimodal time series pre-training within limited domains and tasks (King, Yang, and Mortazavi 2023; Li et al. 2023a). Plotting time series into charts (Meng et al. 2024; Masry et al. 2024) is also viable. However, they do not support fine-grained time series forecasting, the most crucial task of time series analysis.

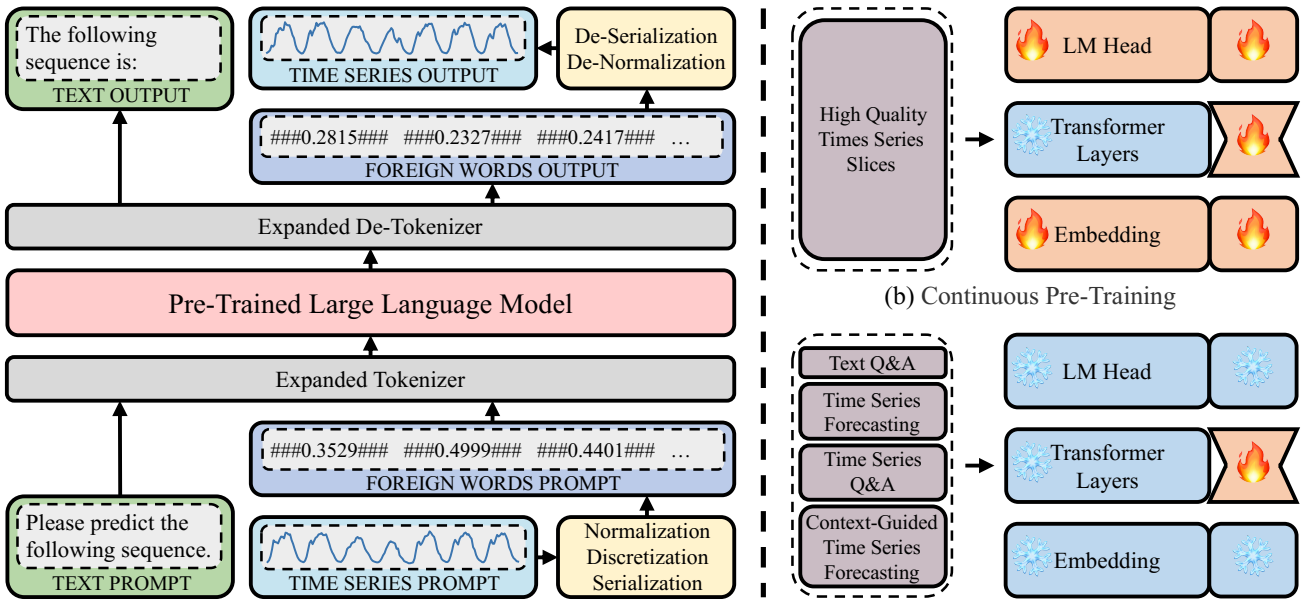


Figure 1: The overview of ChatTime. (a) illustrates the overall architecture, introducing the yellow plug-ins that enable the intertranslation of time-series real values and foreign language. The vocabulary of the grey tokenizer is also extended to accommodate the time series language. We further pre-train (b) and fine-tune (c) existing LLMs using the same methodology as vocabulary expansion, eliminating the need to train from scratch or alter the model architecture.

3 Methodology

3.1 Overview

As illustrated in Figure 1(a), ChatTime initially encodes time series into a foreign language through normalization, discretization, and the incorporation of mark characters. The expanded tokenizer then transforms text and foreign words into token indexes. After processing by the LLM, the de-tokenizer translates the token indexes back into text and foreign words. Finally, the foreign words are re-decoded into time series by removing mark characters and applying inverse normalization. As depicted in Figures 1(b) and 1(c), the training process is divided into two phases: continuous pre-training and instruction fine-tuning. Both phases utilize 4-bit quantized models with LoRA (Hu et al. 2022).

3.2 Model Architecture

By conceptualizing it as a foreign language, ChatTime enables pre-trained LLMs to process time series through vocabulary expansion. As illustrated in Figure 1(a), ChatTime implements two critical modifications: first, it introduces a yellow plug-in that supports the interconversion between real values of time series and foreign language; second, it extends the vocabulary of grey tokenizer to accommodate time series language.

Unlike natural language derived from a finite dictionary, time series are typically real-valued data within unbounded continuous domains. Consider the time series $\mathbf{x}_{1:C+H} = \{x_1, \ldots, x_{1:C+H}\}$, where the initial C time steps constitute the history series, and the subsequent H time steps form the prediction series. ChatTime employs min-max scaling to map unbounded real values into a bounded range of -1 to 1. Given that the prediction series is unknown during the actual inference process, we scale based solely on the history series. Acknowledging that the prediction series may surpass the range of the history series, we scale the history series into the range of -0.5 to 0.5, reserving the remaining interval as a buffer for the prediction series. The scaling process is described as follows:

$$\tilde{\mathbf{x}}_{1:C+H} = \frac{\mathbf{x}_{1:C+H} - \min\left(\mathbf{x}_{1:C}\right)}{\max\left(\mathbf{x}_{1:C}\right) - \min\left(\mathbf{x}_{1:C}\right)} - 0.5 \qquad (1)$$

The scaled time series remain continuous real values that cannot be directly converted into a finite dictionary. We employ a binning technique to quantize these real values into discrete tokens. Specifically, we uniformly partition the interval from -1 to 1 into 10K bins. Each scaled real value is mapped to the corresponding bin, and the center value of the bin is used as the quantized lossy discrete value.

Next, we fix the precision of the discretized time series to 4 like LLMTIME (Gruver et al. 2023). As illustrated in Table 2, LLMTIME presents two methods for GPT and LLaMA tokenizing time series bit-by-bit. However, this method consumes a substantial number of tokens, leading to large computational costs. To address this issue, we introduce the mark characters *"###"* at the beginning and end of the discretized time series to form foreign language words. By extending the vocabulary of the tokenizer, only one token is needed for each value, regardless of its precision. Moreover, not only do we add the foreign words derived from the center of the 10K bins into the vocabulary, but also include an additional *"###Nan###"* to manage missing values.

Time Series
[0.2835, 0.2285, 0.1587, 0.4001]
GPT (34 tokens)
"2 8 3 5 , 2 2 8 5 , 1 5 8 7 , 4 0 0 1"
['2', '_', '8', '_', '3', '_', '5', '_', '_', '2', '_', '2', '_', '8', '_', '5', '_', '_', '1', '_', '5', '_', '8', '_', '7', '_', '_', '4', '_', '0', '_', '0', '_', '1']
LLaMA (22 tokens)
"2835, 2285, 1587, 4001"
['2', '8', '3', '5', ',', '_', '2', '2', '8', '5', ',', '_', '1', '5', '8', '7', ',', '_', '4', '0', '0', '1']
ChatTime (7 tokens)
"###0.2835### ###0.2285### ###0.1587### ###0.4001###"
['###0.2835###', '_', '###0.2285###', '_', '###0.1587###', '_', '###0.4001###']

Table 2: The comparison of token consumption between LLMTIME and ChatTime.

3.3 Continuous Pre-Training

Continuous pre-training is frequently employed to enhance the comprehension of LLMs in specialized domains. Grasping the fundamental principles of time series is essential for executing downstream tasks. As depicted in Figure 1(b), during the continuous pre-training stage, 1M high quality time series slices are used to pre-train LLaMA-2-7B-Base (Touvron et al. 2023b), resulting in ChatTime-1-7B-Base. We employ autoregressive forecasting on extensive time series data as a pre-training task. As the vocabulary of the tokenizer is expanded, the embedding layer and output header also require training alongside the Transformer layer.

The data for continuous pre-training is sourced from two extensive open-source time series repositories, Monash (Godahewa et al. 2021) and TFB (Qiu et al. 2024), encompassing approximately 100 sub-datasets. Notably, the 11 sub-datasets for evaluating ZSTSF and CGTSF tasks in Section 4.2 and 4.3 have been excluded to prevent information leakage. The autoregressive forecasting strategy enables ChatTime to support history and prediction windows of any size. We apply sliding slices to the original time series using five distinct window and step sizes, as illustrated in Table 3. We prioritize slicing the original time series into larger segments. Given the numerous repeating patterns and the limited computational resources, we perform K-means (Pedregosa et al. 2011) on 10M original time series slices. We categorize them into 1M and 25K groups, randomly selecting one sample from each group to serve as a representative. Consequently, we create a high-quality dataset for continuous pre-training (1M) and instruction fine-tuning (25K).

3.4 Instruction Fine-Tuning

As shown in Figure 1(c), during the instruction fine-tuning phase, four task datasets are used to fine-tune ChatTime-1-7B-Base, yielding the final ChatTime-1-7B-Chat. 25K samples are extracted for each task, totaling 100K instances of fine-tuned data. We only fine-tune the Transformer layer during this phase.

Window Size	History Length	Prediction Length	Sliding Step
576	512	64	32
288	256	32	16
144	128	16	8
72	64	8	4
36	32	4	2

Table 3: The setting of sliding windows when constructing continuous pre-training dataset.

We introduce the text question answering task to retain the textual inference capabilities of the LLMs. We randomly select 25K samples from the widely used Alpaca (Taori et al. 2023) dataset for this task. For the unimodal time series forecasting task, we utilize 25K high quality time series slices from Section 3.3. Moreover, context-guided forecasting and time series question answering tasks involve the interconversion of time series and text modalities, where related datasets are lacking. Therefore, we collect three CGTSF datasets and synthesize a TSQA dataset to address this gap and offer a valuable resource for future research.

The context-guided forecasting task is supported by three multimodal datasets: Melbourne Solar Power Generation (MSPG), London Electricity Usage (LEU), and Paris Traffic Flow (PTF). Only background, weather (forecast from Open-Meteo (Open-Meteo 2021)), and date are included as textual auxiliary information to prevent future information leakage. Detailed dataset information is provided in Appendix B.2. To avoid information leakage during the evaluation phase in Section 4.3, each dataset is chronologically split into training, validation, and test sets with a ratio of 6:2:2. A sample of 25K data points is randomly selected from the training sets of these three datasets.

For the time series question answering task, we employ the KernelSynth (Ansari et al. 2024) to generate a variable-length multimodal question answering dataset based on four generic typical time series features (Fons et al. 2024). Detailed dataset information is provided in Appendix B.3. We randomly select 25K data entries from this dataset for instruction fine-tuning. By aligning time series features with textual representations, this task can also improve the performance of ChatTime in context-guided forecasting.

4 Experiment

4.1 Implementation Setting

The training process of ChatTime is divided into continuous pre-training and instruction fine-tuning. Both phases utilize 4-bit quantized models with LoRA. In the LoRA, the rank and alpha are set to 8 and 16, respectively. The batch size is 8 with a gradient accumulation of 32, resulting in a global batch size of 256. The number of epochs for pre-training is set to 2, spanning 8K steps, with a visualization of the losses shown in Figure 3(a). The number of epochs for fine-tuning is set to 4, spanning 1.6K steps, with a visualization of the losses depicted in Figure 3(b). Owing to Unsloth (AI 2023), the entire train process can be executed on an Ubuntu server equipped with a single NVIDIA GeForce RTX 4090 graph-

Dataset	Hist	Pred	Full-Shot Forecast				Zero-Shot Forecast				
			DLinear	iTransformer	GPT4TS	TimeLLM	TimeGPT	Moirai	TimesFM	Chronos	ChatTime
ETTh1	48	24	0.1462	0.1650	**0.1389**	0.1467	**0.1604**	0.1694	0.2021	0.1634	0.1698
	72	24	**0.1358**	0.1852	0.1469	0.1439	0.1603	0.1796	0.1599	**0.1372**	0.1403
	96	24	**0.1398**	0.1964	0.1447	0.1473	0.1577	0.1433	0.1454	**0.1374**	0.1374
	120	24	**0.1371**	0.1971	0.1414	0.1513	0.1594	0.1492	0.1502	**0.1348**	0.1431
ETTh2	48	24	**0.2724**	0.2937	0.2742	0.2758	**0.2874**	0.2963	0.3360	0.3128	0.2906
	72	24	0.2756	0.3118	**0.2717**	0.2972	0.2888	0.3109	**0.2880**	0.3045	0.3092
	96	24	**0.2831**	0.3417	0.2900	0.2864	**0.2902**	0.3139	0.3144	0.3158	0.2917
	120	24	0.2863	0.3299	**0.2854**	0.3175	0.3026	**0.2905**	0.3311	0.3150	0.3124
ETTm1	192	96	0.1479	0.1608	**0.1384**	0.1503	0.1921	0.1608	0.1719	0.1604	**0.1442**
	288	96	0.1400	0.1813	**0.1345**	0.1425	0.1715	0.1848	0.1650	**0.1452**	0.1587
	384	96	**0.1428**	0.1680	0.1518	0.1452	0.1616	0.1619	0.1584	0.1463	**0.1393**
	480	96	**0.1406**	0.2001	0.1472	0.1527	0.1570	0.1703	0.1582	**0.1401**	0.1802
ETTm2	192	96	0.2793	0.3397	**0.2792**	0.2918	0.4294	0.4206	0.3405	0.3759	**0.3135**
	288	96	**0.2881**	0.3623	0.2918	0.2904	0.3625	0.3882	**0.3277**	0.3472	0.3340
	384	96	0.2947	**0.2880**	0.3089	0.3003	**0.3389**	0.3742	0.3562	0.3589	0.3434
	480	96	0.3014	0.3725	**0.2945**	0.3054	**0.3242**	0.3597	0.3679	0.3353	0.4213
Electric	48	24	0.5719	0.5951	**0.5008**	0.5733	**0.5276**	0.6617	0.6005	0.6098	0.6083
	72	24	0.5486	0.5619	**0.4896**	0.4989	**0.4953**	0.6018	0.5454	0.5914	0.6238
	96	24	0.5536	0.5290	**0.4432**	0.4816	0.4971	0.5260	0.5276	0.5139	**0.4951**
	120	24	0.4714	0.5622	**0.4540**	0.4848	0.5196	0.4963	**0.4900**	0.5031	0.5101
Exchange	14	7	0.0543	**0.0526**	0.0533	0.0531	0.0620	0.0784	0.0647	0.0555	**0.0540**
	21	7	0.0571	0.0547	**0.0505**	0.0505	0.0599	0.0812	0.0743	0.0635	**0.0556**
	28	7	0.0595	0.0581	**0.0508**	0.0511	0.0610	0.0844	0.0652	0.0595	**0.0559**
	35	7	0.0615	0.0607	**0.0493**	0.0524	0.0629	0.0677	0.0632	0.0598	**0.0558**
Traffic	48	24	0.4662	0.5000	0.4557	**0.4473**	0.4668	0.4887	0.4483	0.4718	**0.4220**
	72	24	0.4475	0.4443	**0.4116**	0.4252	0.4635	0.4581	0.4196	**0.3725**	0.3873
	96	24	0.4438	0.4348	0.4190	**0.4064**	0.4332	0.4082	**0.3714**	0.3787	0.4074
	120	24	0.4190	0.4149	**0.3416**	0.4279	0.4161	**0.3539**	0.3542	0.3908	0.4125
Weather	288	144	**0.0339**	0.0367	0.0364	0.0352	0.0331	**0.0305**	0.0354	0.0343	0.0352
	432	144	**0.0366**	0.0404	0.0401	0.0395	0.0321	0.0302	**0.0298**	0.0346	0.0356
	576	144	**0.0364**	0.0379	0.0399	0.0377	0.0328	0.0331	0.0321	0.0349	**0.0284**
	720	144	**0.0371**	0.0395	0.0392	0.0392	**0.0323**	0.0353	0.0369	0.0335	0.0332
Avg. MAE			0.2409	0.2661	**0.2286**	0.2390	0.2544	0.2659	0.2541	**0.2512**	0.2515
Avg. Rank			3.7500	6.9688	**3.0000**	3.9688	5.5625	6.5000	5.7500	4.8438	**4.4688**

Table 4: The evaluation result in the traditional unimodal time series forecasting task. The lower values for all metrics represent the better performance. The best results among full-shot and zero-shot forecasting methods are highlighted in bold, respectively.

ics card. All source code, data, and weight will be made publicly accessible upon the publication of the paper.

4.2 Zero-Shot Time Series Forecasting

For the regular unimodal time series forecasting task, we conduct experiments on eight datasets across four domains: Electric, Exchange, Traffic, and Weather, in addition to four ETT datasets. These datasets, widely used for benchmarking, are publicly available (Wu et al. 2021). Detailed information is provided in Appendix B.1. Notably, we have excluded these datasets during the training of ChatTime to prevent information leakage. Each dataset is chronologically divided into training, validation, and test sets with a ratio of 6:2:2. We determine a priori period of each dataset based on its collection granularity and use it as the prediction length. The history length is set to be $\{2,3,4,5\}$ times the prediction length, ensuring that the history window of the zero-shot models contains at least two complete periods. We report the Mean Absolute Error (MAE) as the evaluation metric, where lower values mean better performance.

The baselines are broadly categorized into two groups. The first group consists of models trained and predicted on a single dataset with fixed history and prediction lengths, including DLinear (Zeng et al. 2023), iTransformer (Liu et al. 2024a), GPT4TS (Zhou et al. 2023), and TimeLLM (Jin et al. 2024). GPT4TS and TimeLLM both utilize pretrained LLMs as their backbone. The second group comprises foundational models capable of zero-shot forecasting, such as TimeGPT (Garza and Canseco 2023), Moirai (Woo et al. 2024), TimesFM (Das et al. 2024), and Chronos (Ansari et al. 2024). For the foundational models available in different sizes, we use their most powerful versions. All

Dataset	Hist	Pred	Dataset-Specific Forecast				Dataset-Shared Forecast				
			DLinear	GPT4TS	TimeLLM	TGForecaster	Moirai	TimesFM	Chronos	ChatTime-	ChatTime
MSPG	192	96	**0.7136**	0.7558	0.7697	0.7595	0.8108	0.8362	0.7427	0.7606	**0.7346**
	288	96	**0.7083**	0.7464	0.7959	0.7610	0.7849	0.7896	0.7408	0.7606	**0.7353**
	384	96	**0.7014**	0.7388	0.7672	0.7638	0.7749	0.7811	0.7352	0.7607	**0.7330**
	480	96	**0.7018**	0.7311	0.7632	0.7695	0.7664	0.7667	0.7344	0.7607	**0.7292**
LEU	96	48	0.6676	0.6697	0.6531	**0.6181**	**0.6228**	0.6670	0.6571	0.6496	0.6305
	144	48	0.6495	0.6567	0.6474	**0.6355**	**0.6085**	0.6475	0.6597	0.6506	0.6231
	192	48	0.6407	0.6771	**0.6329**	0.6458	**0.6008**	0.6490	0.6645	0.6407	0.6111
	240	48	**0.6316**	0.6383	0.6356	0.6329	**0.5968**	0.6333	0.6631	0.6377	0.6085
PTF	48	24	0.5204	0.4373	**0.4211**	0.4411	0.5981	0.4851	**0.4813**	0.5155	0.4849
	72	24	0.5075	0.4253	0.4031	**0.3943**	0.5776	**0.4258**	0.4276	0.4436	0.4307
	96	24	0.4965	0.3921	0.4392	**0.3653**	0.5179	0.4054	0.4336	0.4172	**0.3920**
	120	24	0.4796	0.3713	0.3594	**0.3594**	0.5245	0.3807	0.3902	0.3943	**0.3480**
Avg. MAE			0.6182	0.6033	0.6073	**0.5955**	0.6487	0.6223	0.6109	0.6160	**0.5884**
Avg. Rank			4.7500	5.0833	4.9167	**3.9167**	5.9167	6.4167	5.5000	5.7500	**2.5833**

Table 5: The evaluation result in the context-guided time series forecasting task. The lower values for all metrics represent the better performance. The best results among dataset-specific and dataset-shared methods are highlighted in bold, respectively.

baselines are evaluated based on our runs using the same hardware as ChatTime, except for the closed-source model TimeGPT, which requires official API calls. We use official implementations from GitHub and follow the hyperparameter configurations recommended in their papers. The prompt templates for ChatTime are provided in Appendix A.1.

The experimental results are summarized in Table 4. To avoid a few datasets dominating the results, we primarily compare the average MAE (the lower, the better) and the average Rank (the smaller, the better) across eight datasets. By fine-tuning an existing pre-trained LLM instead of training it from scratch, ChatTime achieves 99.9% of the zero-shot prediction accuracy of the previous SOTA method, Chronos, using only 4% of the data. Compared to the full-shot forecasting model, ChatTime also attains 90.9% of the prediction accuracy of the previous SOTA method, GPT4TS. Although introducing LLMs brings some performance gains for GPT4TS and TimeLLM, they do not significantly outperform the simple linear model DLinear. This validates that current unimodal methods may be approaching their saturation point. To visually compare the differences between these baselines, we provide a showcase in Appendix C.1, where ChatTime continues to demonstrate its superiority.

4.3 Context-Guided Time Series Forecasting

For the multimodal context-guided time series forecasting task, we conduct experiments on the three datasets collected in Section 3.4. We segment each dataset adhering to the protocols outlined in Section 4.2. The settings for history length, prediction length, and evaluation metric also remain consistent with Section 4.2. Notably, due to the limited multimodal datasets, the instruction fine-tuning phase of ChatTime is performed on partial training sets of these three datasets. Although this deviates from the zero-shot setup, ChatTime still does not require separate training for different scenarios but utilizes shared model weights.

The baselines are similar to Section 4.2, except for including TGForecaster (Xu et al. 2024), which can handle textual information. Moreover, to verify the auxiliary role of context in time series forecasting, we specifically establish a comparison baseline, ChatTime-, which excludes textual input during forecasting. The prompt templates for ChatTime are provided in Appendix A.2.

The experimental results are summarized in Table 5. With the incorporation of textual information, TGForecaster and ChatTime exhibit superior performance compared to other baselines. Owing to the synergistic integration of the two modalities, ChatTime even surpasses TGForecaster, which is trained independently on each dataset. Moreover, ChatTime significantly outperforms ChatTime- using only unimodal values, affirming the effectiveness of contextual assistance. We provide showcases in Appendix C.2 that further validate the substantial potential and utility of ChatTime.

4.4 Time Series Question Answering

For the multimodal time series question answering task, we conduct experiments on the dataset synthesized in Section 3.4. We exclude the 25K samples utilized for the instruction fine-tuning of ChatTime and use the remaining data as the test set. Baselines for comparison are powerful generic pre-trained LLMs: GPT4 (OpenAI 2023a), GPT3.5 (OpenAI 2023b), GLM4 (GLM 2024), and LLaMA3-70B (Meta 2024). For the input formats of the time series, we employ two prompts suggested by LLMTIME (Gruver et al. 2023), as described in Section 3.2. For GLM4, we have tested both prompts and select the format like LLaMA, which yields better results. Except for LLaMA3, which uses API from Alibaba (Alibaba 2024), the remaining baselines all use their official API. The prompt templates for ChatTime are provided in Appendix A.3. Given the nature of feature recognition, we report the accuracy (Acc) as evaluation metric, with higher scores indicating better performance.

Feat	Len	GPT4	GPT3.5	GLM4	LLaMA3	ChatTime
Trend	64	0.6532	0.3507	0.7319	0.6799	**0.9011**
	128	0.7015	0.5846	0.7574	0.5855	**0.9068**
	256	0.7482	0.5028	0.6377	0.6143	**0.8843**
	512	0.6346	0.5903	0.6697	0.6753	**0.8234**
Volatility	64	0.5585	0.5633	0.6797	0.6373	**0.7874**
	128	0.4979	0.3839	0.4770	0.4756	**0.6954**
	256	0.4624	0.4894	0.5418	0.5246	**0.6228**
	512	0.3169	0.3796	0.4549	0.5261	**0.5736**
Season	64	0.3518	0.3428	0.3366	0.3484	**0.6639**
	128	0.3515	0.3952	0.3464	0.3958	**0.6517**
	256	0.5283	0.5089	0.3892	0.4120	**0.6463**
	512	0.4457	0.4889	0.3892	0.4127	**0.6244**
Outlier	64	0.7230	0.4325	0.5359	0.7051	**0.8773**
	128	0.6327	0.5940	0.5298	0.5694	**0.9032**
	256	0.6795	0.4579	0.5019	0.5073	**0.8593**
	512	0.6219	0.4996	0.2822	0.4085	**0.7478**
Avg. Acc		0.5567	0.4728	0.5163	0.5299	**0.7605**
Avg. Rank		3.0625	4.0625	3.6250	3.2500	**1.0000**

Table 6: The evaluation result in the time series question answering task. Higher values mean better performance for all metrics, except Rank, which is better when lower. The best results are highlighted in bold.

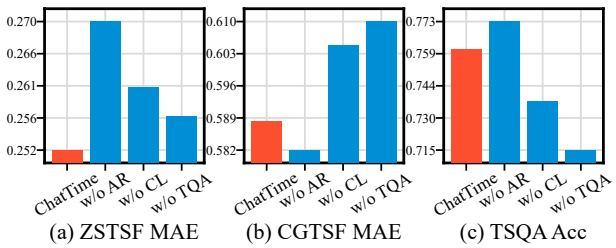
The experimental results are summarised in Table 6. To avoid a few datasets dominating the results, we primarily compare the average Acc (the higher, the better) and the average Rank (the smaller, the better) across four features. Although generic LLMs have shown impressive performance across various text tasks, their efficacy in time series comprehension remains suboptimal. ChatTime, not only preserves the inference capabilities of LLMs but also demonstrates a superior understanding of time series features. We also provide showcases in Appendix C.3.

4.5 Ablation Study

To validate the soundness of each design in ChatTime, we perform an ablation study on the aforementioned three tasks and report their average results across all datasets individually. As depicted in Figure 2, we assess the indispensability of autoregressive continuous pre-training (w/o AR), clustering for time-series slices (w/o CL), and the text question answering in fine-tuning instructions (w/o TQA).

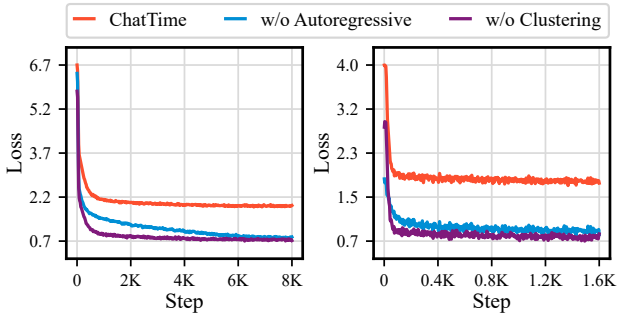
In w/o AR, we substitute the 1M continuous pre-training dataset with the 100K instruction fine-tuning dataset. We increase the epoch by ten times to maintain consistent parameter iterations. This increase leads to a slight improvement in CGTSF and TSQA. However, removing the pre-training dataset causes a struggle to grasp the fundamental time series features, significantly reducing the zero-shot inference capability and practical value. The loss depicted in Figure 3 also illustrates the overfitting in ChatTime after replacing time series pre-training data.

In the w/o CL, we substitute the high-quality time series slices obtained from clustering with low-quality data



(a) ZSTSF MAE (b) CGTSF MAE (c) TSQA Acc

Figure 2: The evaluation result between ChatTime and variants. Lower values are better for ZSTSF and CGTSF, while higher values are better for TSQA.



(a) Pre-Training (b) Fine-Tuning

Figure 3: The loss between ChatTime and its variants during continuous pre-training and instruction fine-tuning.

randomly sampled from the 10M original slices. The findings indicate that ChatTime lacks sufficient comprehension of time series after replacement. There are various degradations across the three tasks. The loss observed in Figure 3 also confirms that randomly sampled data is less challenging to model, making ChatTime prone to overfitting.

In the w/o TQA, we exclude the text question answering dataset in the instruction fine-tuning phase. The findings indicate that omitting this task hampers the inference capability, resulting in performance degradation across all three tasks, particularly in the multimodal CGTSF and TSQA.

5 Conclusion

In this study, we concentrate on the efficient construction of a multimodal time series foundation model that allows for zero-shot inference and supports both time series and textual bimodal inputs and outputs. By innovatively characterizing time series as a foreign language, we introduce ChatTime, a framework for the unified processing of time series and text. To validate the superior performance of ChatTime, we have meticulously designed a series of experiments and constructed four multimodal datasets to fill relevant data gaps. The experimental results demonstrate the significant potential and utility of ChatTime, offering novel perspectives and solutions for time series analysis tasks. Due to resource constraints, ChatTime has not yet reached saturation. In future work, we plan to use more data and computational resources to further extend its applicable tasks, such as anomaly detection, classification, or summarization.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants (62171057, 62201072, 62471055, U23B2001, 62321001, 62101064), the Ministry of Education and China Mobile Joint Fund (MCM20200202, MCM20180101), the Fundamental Research Funds for the Central Universities (2024PTB-004), and the BUPT Excellent Ph.D. Students Foundation (CX20241016).

References

AI, U. 2023. Unsloth AI — Finetune Llama 3 & Mistral LLMs. https://unsloth.ai. Accessed: 2024-05-01.

Alibaba. 2024. DashScope. https://dashscope.console.aliyun.com. Accessed: 2024-05-01.

Ansari, A. F.; Stella, L.; Türkmen, A. C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S. S.; Pineda-Arango, S.; Kapoor, S.; Zschiegner, J.; Maddix, D. C.; Mahoney, M. W.; Torkkola, K.; Wilson, A. G.; Bohlke-Schneider, M.; and Wang, Y. 2024. Chronos: Learning the Language of Time Series. arXiv:2403.07815.

Box, G. E. P.; and Jenkins, G. M. 1968. Some Recent Advances in Forecasting and Control. *Journal of the Royal Statistical Society*, 17.

Cai, W.; Liang, Y.; Liu, X.; Feng, J.; and Wu, Y. 2024. MSGNet: Learning Multi-Scale Inter-Series Correlations for Multivariate Time Series Forecasting. In *AAAI Conference on Artificial Intelligence*.

Csaki, Z.; Li, B.; Li, J.; Xu, Q.; Pawakapan, P.; Zhang, L.; Du, Y.; Zhao, H.; Hu, C.; and Thakker, U. 2024. SambaLingo: Teaching Large Language Models New Languages. arXiv:2404.05829.

Das, A.; Kong, W.; Sen, R.; and Zhou, Y. 2024. A Decoder-Only Foundation Model for Time-Series Forecasting. In *International Conference on Machine Learning*.

Du, S.; Li, T.; Yang, Y.; and Horng, S.-J. 2021. Deep Air Quality Forecasting Using Hybrid Deep Learning Framework. *IEEE Transactions on Knowledge and Data Engineering*, 33.

Flunkert, V.; Salinas, D.; and Gasthaus, J. 2017. DeepAR: Probabilistic Forecasting With Autoregressive Recurrent Networks. arXiv:2201.00382.

Fons, E.; Kaur, R.; Palande, S.; Zeng, Z.; Vyetrenko, S.; and Balch, T. 2024. Evaluating Large Language Models on Time Series Feature Understanding: A Comprehensive Taxonomy and Benchmark. arXiv:2404.16563.

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

GLM, T. 2024. ChatGLM: A Family of Large Language Models From Glm-130B to Glm-4 All Tools. arXiv:2406.12793.

Godahewa, R.; Bergmeir, C.; Webb, G. I.; Hyndman, R. J.; and Montero-Manso, P. 2021. Monash Time Series Forecasting Archive. In *Neural Information Processing Systems Track on Datasets and Benchmarks*.

Goswami, M.; Szafer, K.; Choudhry, A.; Cai, Y.; Li, S.; and Dubrawski, A. 2024. MOMENT: A Family of Open Time-Series Foundation Models. In *International Conference on Machine Learning*.

Gruver, N.; Finzi, M.; Qiu, S.; and Wilson, A. G. 2023. Large Language Models Are Zero-Shot Time Series Forecasters. In *Neural Information Processing Systems*.

Guo, X.; Zhang, Q.; Jiang, J.; Peng, M.; Yang, H.; and Zhu, M. 2024. Towards Responsible and Reliable Traffic Flow Prediction With Large Language Models. arXiv:2404.02937.

He, H.; Zhang, Q.; Bai, S.; Yi, K.; and Niu, Z. 2022. CATN: Cross Attentive Tree-Aware Network for Multivariate Time Series Forecasting. In *AAAI Conference on Artificial Intelligence*.

He, Q.-Q.; Siu, S. W. I.; and Si, Y.-W. 2023. Instance-Based Deep Transfer Learning With Attention for Stock Movement Prediction. *Applied Intelligence*, 53.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9.

Hu, E. J.; elong Shen; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Jia, F.; Wang, K.; Zheng, Y.; Cao, D.; and Liu, Y. 2024. GPT4MTS: Prompt-Based Large Language Model for Multimodal Time-Series Forecasting. In *AAAI Conference on Artificial Intelligence*.

Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J. Y.; Shi, X.; Chen, P.-Y.; Liang, Y.; Li, Y.-F.; Pan, S.; and Wen, Q. 2024. TimeLLM: Time Series Forecasting by Reprogramming Large Language Models. In *International Conference on Learning Representations*.

Kim, S.; Choi, S.; and Jeong, M. 2024. Efficient and Effective Vocabulary Expansion Towards Multilingual Large Language Models. arXiv:2402.14714.

King, R.; Yang, T.; and Mortazavi, B. 2023. Multimodal Pretraining of Medical Time Series and Notes. arXiv:2312.06855.

Li, J.; Liu, C.; Cheng, S.; Arcucci, R.; and Hong, S. 2023a. Frozen Language Model Helps Ecg Zero-Shot Learning. In *Medical Imaging with Deep Learning*.

Li, Z.; Qi, S.; Li, Y.; and Xu, Z. 2023b. Revisiting Long-term Time Series Forecasting: An Investigation on Linear Mapping. arXiv:2305.10721.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024a. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations*.

Liu, Y.; Zhang, H.; Li, C.; Huang, X.; Wang, J.; and Long, M. 2024b. Timer: Transformers for Time Series Analysis at Scale. In *International Conference on Machine Learning*.

Masry, A.; Shahmohammadi, M.; Parvez, M. R.; Hoque, E.; and Joty, S. 2024. ChartInstruct: Instruction Tuning for Chart Comprehension and Reasoning. arXiv:2403.09028.

Meng, F.; Shao, W.; Lu, Q.; Gao, P.; Zhang, K.; Qiao, Y.; and Luo, P. 2024. ChartAssisstant: A Universal Chart Multimodal Language Model via Chart-To-Table Pre-training and Multitask Instruction Tuning. arXiv:2401.02384.

Merrill, M. A.; Tan, M.; Gupta, V.; Hartvigsen, T.; and Althoff, T. 2024. Language Models Still Struggle to Zero-Shot Reason About Time Series. arXiv:2404.11757.

Meta. 2024. Meta Llama 3. https://llama.meta.com/llama3. Accessed: 2024-05-01.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series Is Worth 64 Words: Long-Term Forecasting With Transformers. In *International Conference on Learning Representations*.

Open-Meteo. 2021. Open-Meteo: Free Weather API. https://open-meteo.com. Accessed: 2024-05-01.

OpenAI. 2023a. GPT-4 Technical Report. arXiv:2303.08774.

OpenAI. 2023b. OpenAI Platform. https://platform.openai.com/docs/models/gpt-3-5-turbo. Accessed: 2024-05-01.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12.

Pinto, T.; Praça, I.; Vale, Z. A.; and Silva, J. 2021. Ensemble Learning for Electricity Consumption Forecasting in Office Buildings. *Neurocomputing*, 423.

Puri, C.; Kooijman, G.; Vanrumste, B.; and Luca, S. 2022. Forecasting Time Series in Healthcare With Gaussian Processes and Dynamic Time Warping Based Subset Selection. *IEEE Journal of Biomedical and Health Informatics*, 26.

Qiu, X.; Hu, J.; Zhou, L.; Wu, X.; Du, J.; Zhang, B.; Guo, C.; Zhou, A.; Jensen, C. S.; Sheng, Z.; and Yang, B. 2024. TFB: Towards Comprehensive and Fair Benchmarking of Time Series Forecasting Methods. arXiv:2403.20150.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca. Accessed: 2024-05-01.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023a. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Canton-Ferrer, C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; andRobert Stojnic, A. R.; Edunov, S.; and Scialom, T. 2023b. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Wang, H.; Peng, J.; Huang, F.; Wang, J.; Chen, J.; and Xiao, Y. 2023. MICN: Multi-Scale Local and Global Context Modeling for Long-Term Series Forecasting. In *International Conference on Learning Representations*.

Woo, G.; Liu, C.; Kumar, A.; Xiong, C.; Savarese, S.; and Sahoo, D. 2024. Unified Training of Universal Time Series Forecasting Transformers. In *International Conference on Machine Learning*.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition Transformers With Auto-Correlation for Long-Term Series Forecasting. In *Neural Information Processing Systems*.

Xu, Z.; Bian, Y.; Zhong, J.; Wen, X.; and Xu, Q. 2024. Beyond Trend and Periodicity: Guiding Time Series Forecasting With Textual Cues. arXiv:2405.13522.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are Transformers Effective for Time Series Forecasting? In *AAAI Conference on Artificial Intelligence*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *AAAI Conference on Artificial Intelligence*.

Zhou, T.; Niu, P.; Wang, X.; Sun, L.; and Jin, R. 2023. One Fits All: Power General Time Series Analysis by Pretrained LM. In *Neural Information Processing Systems*.