Time Series Forecasting with LLMs: Understanding and Enhancing Model Capabilities

Mingyu Jin[1,*], **Hua Tang**[2,*], **Chong Zhang**[3,*], **Qinkai Yu**[3], **Chengzhi Liu**[3], **Suiyuan Zhu**[5], **Yongfeng Zhang**[1], **Mengnan Du**[4]
[1] Rutgers University, [2] Shanghai Jiao Tong University, [3] University of Liverpool,
[4] New Jersey Institute of Technology, [5] New York University

Abstract

Large language models (LLMs) have been applied in many fields with rapid development in recent years. As a classic machine learning task, time series forecasting has recently received a boost from LLMs. However, there is a research gap in the LLMs' preferences in this field. In this paper, by comparing LLMs with traditional models, many properties of LLMs in time series prediction are found. For example, our study shows that LLMs excel in predicting time series with clear patterns and trends but face challenges with datasets lacking periodicity. We explain our findings through designing prompts to require LLMs to tell the period of the datasets. In addition, the input strategy is investigated, and it is found that incorporating external knowledge and adopting natural language paraphrases positively affects the predictive performance of LLMs for time series. Overall, this study contributes to insight into the advantages and limitations of LLMs in time series forecasting under different conditions.

1 Introduction

Recently, large language models (LLMs) have been widely used, achieving promising performance across various domains, such as health management, customer analysis, and text feature mining (Peng et al., 2023; Ledro et al., 2022; Huang et al., 2023). Time series forecasting requires extrapolation from sequential observations. Language models, designed to discern intricate concepts within temporally correlated sequences, intuitively appear well-suited for this task. Hence, LLMs demonstrate proficiency in the domain of time series forecasting (Gruver et al., 2023; Rasul et al., 2023; Sun et al., 2023).

Currently, the application of LLMs for time series prediction is at a nascent stage, with the boundaries of this research area remaining ill-defined. The utilization of proprietary large language models in this domain lacks a clear, established methodology. Our objective is to shed light on this emerging field, offering valuable insights and guidance for future research endeavors. However, the LLMs' preferences for the input time series remain unexplored, such as their proficiency in forecasting both seasonal and trend data. We believe that studying this area clearly will greatly improve the performance of LLM in time series forecasting.

To fill this research gap, in this paper, we focus on the question of what are LLMs' preferences for the input time series in time series forecasting. To answer this, we conduct experiments on both real and synthesized datasets. Our observations reveal that LLMs perform better on time series characterized by higher trend or seasonal strengths. To further discern the LLMs' preferences for the specific segments of the input data, we design counterfactual experiments involving systematic permutations of input sequences. We find that LLMs are sensitive to the segment of input sequences proximate to the output.

Naturally, we are interested in the question: Why do LLMs forecast well on datasets with higher trend or seasonal strengths? To solve this, we design prompts that require LLMs to tell the period of the datasets. Through experiments, we let the large language model tell the period of the dataset several times and take the median. We found that the large language model can accurately pinpoint the periodicity of a dataset. This can explain why large language models can predict data sets with high trends or seasonal intensities well since they already learned this kind of knowledge.

In light of these findings, our focus lies on how to leverage these insights to further improve model performance. To address this, we propose two simple techniques to enhance model performance: incorporating external human knowledge and converting numerical sequences into natural language counterparts. Incorporating supplementary information enables large language models to more ef-

fectively grasp the periodic nature of time series data, moving beyond a mere emphasis on the tail of the time series. Transforming numerical data into a natural language format enhances the model's ability to comprehend and reason, also serving as a beneficial approach. Both approaches improve model performance and contribute to our understanding of LLMs in time series forecasting. The workflow is illustrated in Figure 1.
The key contributions are as follows:

- We investigate the LLMs' preferences for the input sequences in time series forecasting. Based on our analysis, we have observed that LLMs are more effective in datasets that have clear periods and trends. A surprising finding is that LLMs may generate output sequences that exhibit higher trend and seasonal strengths. Besides, our results reveal a limitation in LLMs' performance when confronted with multi-period time series datasets. It indicates that LLMs face challenges in recognizing the distinct periods inherent in such datasets. To further discern the LLMs' preferences for the specific segments of the input data, we add Gaussian to the original time series to create counterfactual instances. Our results indicate that large language models are sensitive to the segment of input sequences proximate to the output. All these findings suggest a potential area for improvement in LLMs' adaptability to complex temporal structures.
- Through prompt design and the use of in-context Learning, we have verified that LLMs make better predictions on datasets that exhibit strong trends or seasonality, showing that these models can precisely identify dataset periodicity.
- We propose two simple techniques to improve model performance and find that incorporating external knowledge in prompts and paraphrasing natural language positively affects the performance of LLMs in time series forecasting.

2 Preliminaries

2.1 Large Language Model

We use LLMs as a zero-shot learner for time series forecasting by treating numerical values as text sequences. The success of LLMs in time series forecasting can significantly depend on correct preprocessing and handling of the data (Gruver et al., 2023). We followed their approach and this process involves a few crucial steps.

In the pre-processing phase for time series forecasting with LLMs, numerical values are transformed into strings, a crucial step that significantly influences the model's comprehension and data processing. For instance, a series like 0.123, 1.23, 12.3, 123.0 is reformatted to "1 2, 1 2 3, 1 2 3 0, 1 2 3 0 0", introducing spaces between digits and commas to delineate time steps, while decimal points are omitted to save token space. Tokenization is equally pivotal, shaping the model's pattern recognition capabilities. Unlike traditional methods like byte-pair encoding (BPE) (Hugging Face, 2023), which may disrupt numerical coherence, spacing digits ensures individual tokenization, enhancing pattern discernment.

Moreover, rescaling is employed to efficiently utilize tokens and manage large inputs by adjusting values so a specific percentile aligns to 1, facilitating the model's exposure to varying digit counts and supporting the generation of larger values, a testament to the nuanced yet critical nature of data preparation in leveraging LLMs for time series analysis.

2.2 Time Series Forecasting

In the context of time-series forecasting, the primary goal is to predict the values for the next H steps based on observed values from the preceding K steps, which is mathematically expressed as:

$$\hat{X}_t, ..., \hat{X}_{t+H-1} = F(X_{t-1}, ..., X_{t-K}; V; \lambda) \quad (1)$$

Here, $\hat{X}_t, ..., \hat{X}_{t+H-1}$ represents the H-step estimation given the previous K-step values $X_{t-1}, ..., X_{t-K}$. λ denotes the trained parameters from the model F, and V denotes the prompt or any other information used for the inference. This paper predominantly focuses on single-variate time series forecasting.

Motivated by the interpretability requirements in real-world scenarios, time series can often be decomposed into the trend component, the seasonal component, and the residual component through the addictive model (Cleveland et al., 1990). The trend component captures the hidden long-term changes in the data, such as the linear or exponential pattern. The seasonal component captures the repeating variation in the data, and the residual component captures the remaining variation in the data after the removal of trend and seasonal components. This decomposition offers a method to quantify the properties of the time series, which is detailed in section 3.2.

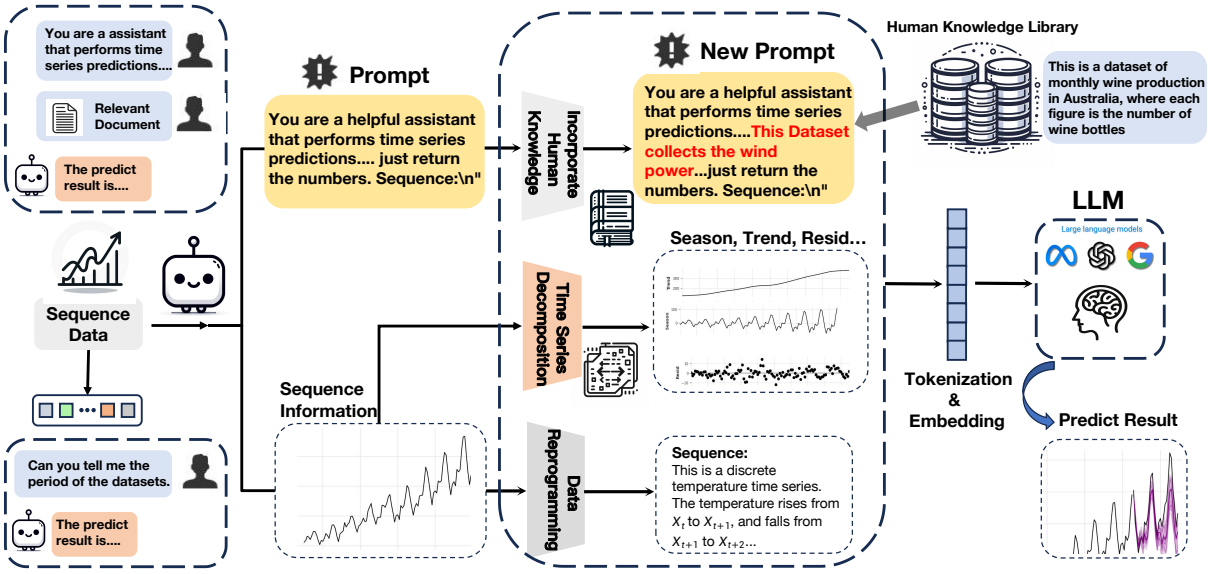


Figure 1: The workflow of our analysis process.

3 What are LLMs' preferences in time series forecasting?

To explore the preference in the large language model, we first quantify the properties of the input time series to investigate the LLMs' preferences for time series. Then, to further emphasize our findings, we evaluate the importance of different segments of the input sequence by adding Gaussian noise to the original time series.

3.1 Analyzing Method

To understand the preferences of the LLMs in time series forecasting, we compare LLMTime using various foundational models like GPT-4 and GPT-3.5-turbo traditional methods on various datasets. We also design experiments on synthesized datasets to validate our findings and analyze the impact of the multiple periods. To quantify the LLMs' preferences towards time series, following (Wang et al., 2006), we define the strength of the trend and the seasonality as follows:

$$Q_T = 1 - \frac{\mathrm{Var}(X_R)}{\mathrm{Var}(X_T + X_R)}, \quad Q_S = 1 - \frac{\mathrm{Var}(X_R)}{\mathrm{Var}(X_S + X_R)} \tag{2}$$

where $X_K \in R^K$, $X_S \in R^K$ and $X_R \in R^K$ denote the trend component, the seasonal component and the residual component respectively. The presented indices serve as indicators of the strength of the trend and seasonality, providing a measure ranging up to 1. It is easy to find that a higher value indicates a stronger trend or seasonality within the time series.

To further discern the LLMs' preferences for the specific segments of the input data, we add Gaussian to the original time series to create counterfactual examples. We initiate by defining a sliding window that constitutes 10% of the total length of the time series. Given the variability in the length of time sequences, we first scale the sequence before incorporating the noise into the original sequence. This method allows us to assess the impact of individual segments and thereby infer the interpretability of the time series segments that LLMs predominantly focus on.

3.2 Preferences for the time series Experiment

In this subsection, we delve into the input sequence preferences for time series forecasting with LLMs. Our experiments are conducted on both real datasets and synthetic datasets. Throughout this subsection, we use GPT-3.5-turbo-instruct and GPT-4 for time series forecasting and measure model performance through R^2 and Mean Absolute Percentage Error (MAPE).

3.2.1 Implementation Details

Real Datasets: We conduct experiments on ten real datasets, as enumerated in Appendix A.2. we apply the Seasonal-Trend decomposition using the LOESS (STL) technique(Cleveland et al., 1990), to decompose the original time series into trend, seasonal, and residual components. Subsequently, we computed the strengths of the trend strength Q_T and seasonal strength Q_S. To further discern the LLMs' preferences for the specific segments of the

input data, we conduct the counterfactual analysis with a systematic permutation to the input time series. We initiate by defining a sliding window that constitutes 10% of the total length of the time series and add Gaussian noise into the data within this window data. Given the variability in the length of time sequences, we first scale the sequence through max-min normalization before incorporating it into the original sequence. Refer to A.2.1 for detailed information.

Synthesized Datasets: To further validate our findings, we conduct experiments on synthesized datasets. We formulate the dataset through the expression $y = \alpha * x + \beta * cos(2\pi f * x) + \epsilon$. To investigate the influence of the number of periods on model performance, we generated a dataset using the function $y = \alpha * x + \beta_1 * cos(2\pi f_1 * x) + \beta_2 * cos(2\pi f_2 * x) + \epsilon$. x ranges from 0 to 20 and ϵ follows the normal distribution $\mathcal{N}(0, 1)$. Refer to A.2.1 for detailed information.

3.2.2 Key Findings

After computing the Pearson correlation coefficients (PCC), we observed a nearly strong correlation between the strengths and model performance, signifying that LLMs perform better when the input time series has a higher trend and seasonal strength. Notably, GPT-4 achieved a higher PCC compared to GPT-3.5-turbo-instruct. It may be attributed to human feedback during GPT-4 training, as individuals may be more aware of seasonal and trend data. Interestingly, there is an increase in the Q_S of the output generated by GPT-4, compared to the original test sequences. It indicates that GPT-4 tends to forecast time series with high seasonal strength, which may provide insights for further research. In the context of multi-period time series, the model performance decreases as the number of periods increases. It indicates that LLMs may have trouble recognizing the multiple periods inherent in such datasets, which is common in reality. For the counterfactual analysis, as shown in Figure 6, there is a noticeable decrease in R^2 values when Gaussian noise is added to the latter segments. Our findings reveal that LLMs are more sensitive to the end of input time series data when it acts as a time series predictor.

4 Why do LLMs forecast well on datasets with higher trend or seasonal strengths?

Our findings indicate that LLMs exhibit enhanced performance on time series data characterized by pronounced trends or significant seasonal variations. This raises the question: Why do LLMs excel in forecasting datasets with marked trend or seasonal patterns? To explore this phenomenon, we crafted prompts that necessitate LLMs to acknowledge the temporal span of the dataset.

This approach is grounded in the hypothesis that LLMs' proficiency in handling datasets with distinct trend or seasonal attributes in-context learningvanced pattern recognition capabilities. By explicitly prompting LLMs to consider the dataset's period, we aim to leverage their inherent ability to discern and extrapolate from complex patterns, thereby shedding light on the mechanisms that underpin their superior performance in such contexts.

4.1 Implementation Details

In order to explore the phenomenon that LLMs forecast well on datasets with higher trend or seasonal strengths, we design experiments to verify this phenomenon. We design prompts to let the LLMs output the predicted value after each sequence prediction. The target model of the experiment is GPT-3.5-Turbo (gpt-3.5-turbo-1106), and the main role of our prompt is to perform in-context learning and require output cycles, as detailed in the Appendix A.3. We select eight time series data sets such as AirPassengersDataset, count the period value after ten times of prediction, and compare the median of these ten results with the real period. The results are shown in Table 3.

4.2 Key Findings

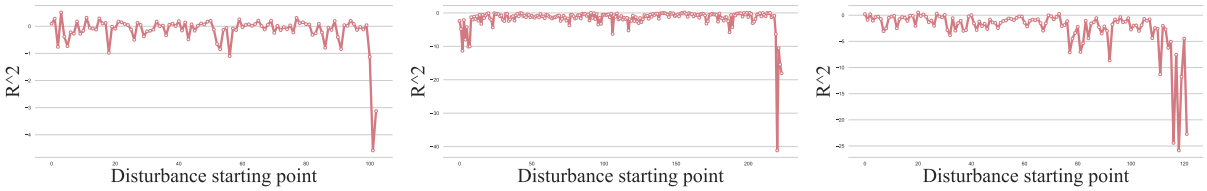
According to the results, we find that large language models can determine the periodicity of a dataset to some extent. Although the fluctuation of each time series prediction is large, the prediction on AirPassenger, Sunspots and Wooly datasets is relatively accurate, and the predicted values on other datasets are also close to our real cycle value. We speculate that LLMs can predict datasets with high trend or seasonal intensity well because they already possess some knowledge about the scene and content of the dataset.

5 How to leverage these insights to improve model's performance?

Based on these findings, our focus is on how to leverage these findings to further improve model performance. Because our ultimate goal is to improve the time series prediction performance of the

Table 1: Model performance in the analysis of LLMs' preferences

Dataset Name	GPT4-MAPE	GPT4-R^2	GPT3.5-MAPE	GPT3.5-R^2	Trend Strength	Seasonal Strength
AirPassengersDataset	6.80	0.79	9.98	0.32	1.00	0.98
AusBeerDataset	3.69	0.78	5.12	0.57	0.99	0.96
MonthlyMilkDataset	5.12	0.38	6.25	−0.34	1.00	0.99
SunspotsDataset	334.30	−0.43	194.29	−1.21	0.81	0.28
WineDataset	10.90	0.49	14.98	0.11	0.67	0.92
WoolyDataset	20.41	−1.74	19.26	−1.42	0.96	0.82
IstanbulTrafficGPT	47.29	−1.96	60.11	−1.75	0.31	0.72
GasRateCO2Dataset	4.21	−0.05	5.97	−1.47	0.65	0.50
HeartRateDataset	7.90	−0.85	6.75	−0.40	0.42	0.49
TurkeyPower	3.36	0.71	3.52	0.76	0.90	0.88



(a) AirPassengers (b) GasRateCO2 (c) MonthlyMilk

Figure 2: Experiments of Sequence Focused Attention Through Counterfactual Explanation

model. Therefore, we retain the original structure of the time series data but enhance the model's input with prompts infused with additional knowledge. These prompts are crafted with seasonal trends and cyclical behaviours in the dataset to endow the model with a richer contextual understanding. In another way, we transform time series data into formats that resemble natural language sequences, enabling large language models to leverage their superior inference skills more effectively in the realm of time series prediction.

5.1 External Knowledge Enhancing Time Series Forecasting

We introduce a novel method to improve the performance and stability of large language models for time series forecasting. The core idea of this part is to use the knowledge obtained from pretraining of large language models to help predict. We will give the large language model some basic information about the current dataset such as the name of the dataset in the prompt, and this process does not involve data leakage (it does not tell the large language model about the period or predicted value, etc.) Assuming we define initial prompt V_s as the original time sequence, the extra information defined as z, the new prompt V_e can be written as: $V_e = z + V_s$

5.1.1 Implementation Details

We input the dataset's external knowledge through prompts before the sequence's input. The external knowledge of each dataset is presented in Appendix A.1. The external knowledge includes the description of dataset features and scenarios and the dataset's data properties. The results are shown in Table 5.

5.1.2 Key Findings

In External Knowledge Enhancement, GPT-4 generally performs better than GPT-3.5 on MSE, MAE and MAPE, especially on AirPassengers, AusBeer and other datasets. Llama-2 significantly outperforms GPT-3.5 and GPT-4 in terms of MSE and MAE metrics on some datasets (e.g., Wooly, ETTh1, ETTm2), indicating that it can capture data features more accurately. R-squared values are shown on some datasets such as ETTh1, ETTm2, and Turkey Power, all models can provide relatively accurate predictions with R-squared values close to 1.

In the LLMTime Prediction, GPT-4 and Llama-2 perform relatively well on AirPassengers and AusBeer datasets with R^2 values close to or above 0.5. GPT-3.5 achieves very high MSE on Sunspots and Wine datasets, which may be due to the complexity of the dataset or the poor adaptation of the model

for these specific tasks. The Turkey Power dataset has a high R^2 value on the time prediction task for all models, indicating a good model fit for this task.

5.2 Natural Language Paraphrasing of Sequences

We will be creating experiments related to natural language paraphrasing of sequences. This will be based on the fact that LLMs are not affected by the order of magnitude and size of digits (Shah et al., 2023).

We have created a method to convert sequences into natural language paraphrases, the main process is using traditional natural language processing techniques to preprocess the series. For instance, if we have a discrete time series that represents temperature changes, we can denote it as X_t, where $X_t = [X_1, X_2, X_3, \ldots, X_n]$. To do this, we extract a natural language description of the sequence based on two data points and a trend from X_t to X_{t+1}, "This is a discrete temperature time series. The temperature rises from X_t to X_{t+1}, and falls from X_{t+1} to X_{t+2}...". The string we get is our natural language paraphrasing sequence.

The sequence generated by the natural language paraphrasing of the text in X_t is represented by P_t. Finally, we paraphrase the sequence from X_t to P_t through the large language model; we let the LLM prediction of P_t get $\hat{P}_t$, after Extraction and Processing, get $\hat{X}_t$, then the above completed our prediction task.

We load in the sequence and determine the movement of the sequence at each step. Then add "from xx increasing to xx," or "from xx decreasing to xx." Complete the process of modifying from time series to natural language. To convert a text sequence into a discrete sequence, we use the Reverse function, which converts the natural language series into a regular expression format through hard code.

5.2.1 Implementation Details

We use the LLMTime model with GPT-3.5-Turbo, GPT-4-turbo and Llama-2 as the core to forecast the built-in time series data set, the training set and test set ratio is 4:1. We will get the time series prediction results and the original series MSE, MAE, MAPE and R^2. The result is present in Table 4 compared with LLMTime forecasting results.

5.2.2 Key Findings

According to the experimental data obtained in the above table, we find that enhancing LLM through natural language paraphrasing has a certain improvement effect on time series analysis and prediction on some datasets. From the perspective of model performance comparison, the R^2 value of GPT-3.5-Turbo in Natural Language Paraphrasing and LLMTime Prediction method is negative in some cases, indicating that the model performs poorly. GPT-4-Turbo performs better on most datasets compared to other magic methods, especially on Natural Language Paraphrasing methods, MSE, MAE, and MAPE usually decrease, while R^2 improves. Llama-2 shows better performance on some datasets, such as lower MSE and MAE and better MAPE and R^2 in the Natural Language Paraphrasing method of the AusBeer dataset.

6 Related Works

6.1 Time Series Prediction

Time series prediction is an important field in the field of artificial intelligence. The unified idea of time series prediction is to analyze past and future series trends. It is used in finance and business to predict the trend of indexes and stocks, in the field of intelligent manufacturing to predict anomaly detection and power load, and in the field of meteorology, agriculture, and navigation to predict temperature, humidity, and climate (Singh and Malhotra, 2023; Munir et al., 2018; Shen et al., 2020; Braei and Wagner, 2020). We categorize related research within this domain into three categories: traditional time series prediction, time series prediction based on machine learning, and deep learning.

Two commonly used methods for traditional time series analysis are the ARIMA method (Box and Pierce, 1970) and the exponential smoothing method (Gardner Jr, 2006). The ARIMA model is a classic forecasting method that breaks down a time series into auto-regressive (AR), difference (I), and moving average (MA) components to make predictions. On the other hand, exponential smoothing is a straightforward yet effective technique that forecasts future values by taking a weighted average of past observations. The ARIMA model requires testing the stationarity of data and selecting the right order. On the other hand, the exponential smoothing method is not affected by outliers, it is only suitable for stationary time series, and its accuracy in predicting future values is lower compared to the ARIMA model.

Time series analysis based on linear regression, decision tree random forest, and other methods

Table 2: Correlation matrix between the strengths of the input time series and the model performance.

Metrics	GPT4-MAPE	GPT4-R^2	GPT3.5-MAPE	GPT3.5-R^2	Trend Strength Q_T	Seasonal Strength Q_S
GPT4-MAPE	1.00	−0.19	0.99	−0.36	−0.02	−0.68
GPT4-R^2	−0.19	1.00	−0.28	0.83	0.58	0.43
GPT3.5-MAPE	0.99	−0.28	1.00	−0.43	−0.12	−0.67
GPT3.5-R^2	−0.36	0.83	−0.43	1.00	0.49	0.60
Trend Strength Q_T	−0.02	0.58	−0.12	0.49	1.00	0.51
Seasonal Strength Q_S	−0.68	0.43	−0.67	0.60	0.51	1.00

Table 3: Median values of datasets

Dataset	Period										Real	Median
AirPassengersDataset	1	12	18	21	7	6	18	18	4	6	12	12
AusBeerDataset	24	10	6	6	1	3	6	7	3	21	4	6
GasRateCO2Dataset	7	36	24	72	28	34	20	7	16	14	39	20
MonthlyMilkDataset	6	6	25	11	6	2	7	34	8	7	12	7
SunspotsDataset	1	12	18	21	7	6	18	18	4	6	12	12
WineDataset	1	11	12	11	12	7	6	8	10	12	12	10
WoolyDataset	10	5	6	7	16	11	5	11	3	4	4	5
HeartRateDataset	13	16	6	12	13	11	12	32	32	64	144	12

can make predictions for future series through the analysis and learning of existing series, as more complex time series prediction methods based on deep learning, RNN, CNN, and LSTM have played a huge role in many applications. Especially in the field of sequence prediction, many variants of LSTM have emerged, such as bidirectional LSTM (Graves et al., 2005), gated recurrent unit (Chung et al., 2014), long short-term memory network (Malhotra et al., 2015), and deep recurrent neural network (Pascanu et al., 2013). This method improves prediction accuracy but takes longer to train and is less prone to overfitting than the original LSTM.

6.2 LLMs for Time Series Prediction

Given the notable achievements of LLMs across various domains, it is prudent to incorporate time series prediction tasks with LLMs. Similar to (Sun et al., 2023), we categorize contemporary research within this domain into two categories: LLM-for-Time Series and Time Series-for-LLM.

LLM-for-Time Series involves either pre-training a foundational large language model or fine-tuning existing LLMs by leveraging extensive time-series data (Rasul et al., 2023; Garza and Mergenthaler-Canseco, 2023; Das et al., 2023; Cao et al., 2023). For instance, (Rasul et al., 2023) aimed to build the foundational models for time series and investigate its scaling behavior. (Chang et al., 2024) proposed a two-stage fine-tuning strategy for handling multivariate time-series forecasting. Although these studies contribute significantly to understanding foundational models, they demand considerable computing resources and expertise in fine-tuning procedures. Moreover, the details of the model may not be disclosed for commercial purposes (Garza and Mergenthaler-Canseco, 2023), which impedes future research. Additionally, in scenarios with limited data available, there is insufficient information for training or fine-tuning. Therefore, in this paper, we concentrate on the Time Series-foR-LLM, harnessing fixed LLMs for the time-series forecasting task.

Time Series-for-LLM centres around using existing LLMs and designing mechanisms, such as crafting appropriate prompts or reprogramming inputs, to handle time series data effectively (Gruver et al., 2023; Sun et al., 2023; Jin et al., 2023; Xue and Salim, 2023). (Sun et al., 2023) tokenized the time series and managed to embed those tokens, and (Jin et al., 2023) reprogrammed the time series data with text prototypes before feeding them to the LLMs. These studies illuminated the characteristics of time series data and devised methods to align them with LLMs. However, they lack an analysis of the ability and bias in forecasting time series. The most related work to us is (Gruver et al., 2023), though, it lacks a quantitative analysis of the preference for the time series in LLMs, and it failed to explore the impact of input forms and prompt contents, such as converting the numerical time series into the natural language sequences and incorporating the background information into the prompt. Our work fills the gap, and we expect our work to be the benchmark for time-series analysis

Table 4: The results of natural language paraphrasing of sequences.

Models	Datasets	Natural Language Paraphrasing			LLMTime Prediction		
		MSE	MAE	MAPE	MSE	MAE	MAPE
GPT-3.5-Turbo	AirPassengers	**267.66**	**3.66**	**0.99**	6244.07	61.39	14.43
	AusBeer	**598.45**	**5.81**	**1.36**	841.68	23.59	5.62
	GasRateCO2	**3.16**	**0.46**	**0.85**	10.88	2.66	4.73
	MonthlyMilk	**968.69**	**8.61**	**1.02**	7507.13	66.28	112.77
	Sunspots	**251.61**	**4.27**	**20.42**	6556.55	58.95	217.94
	Wine	**11403.89**	**96.95**	**37.04**	30488.60	388.28	15.83
	Wooly	**12110.16**	**33.23**	**4.07**	526903.08	574.58	12.00
	HeartRate	**4.38**	**0.55**	**0.57**	76.83	7.15	7.42
GPT-4-Turbo	AirPassengers	**133.10**	**2.87**	**0.80**	1286.25	28.04	6.07
	AusBeer	661.80	**7.24**	**1.63**	**513.49**	18.57	4.28
	GasRateCO2	**2.28**	**0.41**	**0.75**	7.27	2.32	4.18
	MonthlyMilk	**413.63**	**4.94**	**0.57**	4442.18	50.75	172.82
	Sunspots	**194.52**	**5.30**	**16.10**	3374.70	41.87	321.11
	Wine	56138.87	**54.67**	23.63	**22488.17**	253.08	**9.98**
	Wooly	**18063.64**	**11.06**	25.06	942987.19	871.64	**18.55**
	HeartRate	**11.64**	**1.21**	**1.30**	988.14	26.57	29.22
Llama-2	AirPassengers	**751.34**	**6.77**	**1.53**	1286.25	28.04	6.07
	AusBeer	**591.75**	23.25	5.41	644.82	**17.88**	**4.08**
	GasRateCO2	**10.16**	**2.89**	**5.16**	12.78	2.97	5.47
	MonthlyMilk	**851.17**	84.83	**9.46**	3410.20	**41.40**	240.25
	Sunspots	**1483.29**	**33.27**	**17.79**	4467.67	48.95	91.79
	Wine	**102434.52**	852.97	34.72	951194.94	**240.08**	**9.45**
	Wooly	**12180.05**	**83.99**	16.92	675062.52	736.04	**15.83**
	HeartRate	**49.8**	**5.84**	**6.53**	75.58	7.11	7.94

and provide insights for subsequent research.

7 Conclusions

Our research has unearthed key preferences of large language models (LLMs) in the domain of time series forecasting, revealing a proclivity for data with distinct trends and seasonal patterns. Through a blend of real and synthetic datasets, coupled with counterfactual experiments, we have demonstrated LLMs' improved forecasting performance with time series that exhibit clear periodicity. Our proposed strategies incorporating external knowledge and transforming numbers into natural language have both enhanced LLMs' predictive accuracy. Despite encountering challenges with multi-period time series, our work provides a pivotal foundation for future advancements, offering actionable insights and methods to refine the application of LLMs in time series analysis.

Limitation

This study may be limited in the following ways. First, limitations in the scope of the dataset and large language models may not capture the full variability of the results with a wider array. In addition, some experimental sessions lack a comparison with hard-coded solutions, and there is a gap in understanding the performance of LLMs compared to traditional programming methods. Furthermore, the inability to categorize datasets by type and conduct specific types of experiments limits insight into the performance of the model in different data domains. These limitations suggest that the results could benefit from more extensive experiments and more nuanced analyses, which underscores the need to expand future research.

References

George EP Box and David A Pierce. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526.

Mohammad Braei and Sebastian Wagner. 2020. Anomaly detection in univariate time-series: A

survey on the state-of-the-art. *arXiv preprint arXiv:2004.00433*.

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

Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. 2024. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. 1990. Stl: A seasonal-trend decomposition. *J. Off. Stat*, 6(1):3–73.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2023. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*.

Everette S Gardner Jr. 2006. Exponential smoothing: The state of the art—part ii. *International journal of forecasting*, 22(4):637–666.

Azul Garza and Max Mergenthaler-Canseco. 2023. Timegpt-1. *arXiv preprint arXiv:2310.03589*.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *International conference on artificial neural networks*, pages 799–804. Springer.

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

Allen H Huang, Hui Wang, and Yi Yang. 2023. Finbert: A large language model for extracting information from financial text. *Contemporary Accounting Research*, 40(2):806–841.

Hugging Face. 2023. Chapter 6.5 of nlp course. https://huggingface.co/learn/nlp-course/chapter6/5. Accessed: 2023-02-10.

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. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.

Cristina Ledro, Anna Nosella, and Andrea Vinelli. 2022. Artificial intelligence in customer relationship management: literature review and future research directions. *Journal of Business & Industrial Marketing*, 37(13):48–63.

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Puneet Agarwal, et al. 2015. Long short term memory networks for anomaly detection in time series. In *Esann*, volume 2015, page 89.

Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. 2018. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *Ieee Access*, 7:1991–2005.

Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2013. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*.

Cheng Peng, Xi Yang, Aokun Chen, Kaleb E Smith, Nima PourNejatian, Anthony B Costa, Cheryl Martin, Mona G Flores, Ying Zhang, Tanja Magoc, et al. 2023. A study of generative large language model for medical research and healthcare. *arXiv preprint arXiv:2305.13523*.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Vincent Hassen, Anderson Schneider, et al. 2023. Lag-llama: Towards foundation models for time series forecasting. *arXiv preprint arXiv:2310.08278*.

Raj Shah, Vijay Marupudi, Reba Koenen, Khushi Bhardwaj, and Sashank Varma. 2023. Numeric magnitude comparison effects in large language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6147–6161, Toronto, Canada. Association for Computational Linguistics.

Lifeng Shen, Zhuocong Li, and James Kwok. 2020. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in Neural Information Processing Systems*, 33:13016–13026.

Harmanjeet Singh and Manisha Malhotra. 2023. A time series analysis-based stock price prediction framework using artificial intelligence. In *International Conference on Artificial Intelligence of Things*, pages 280–289. Springer.

Chenxi Sun, Yaliang Li, Hongyan Li, and Shenda Hong. 2023. Test: Text prototype aligned embedding to activate llm's ability for time series. *arXiv preprint arXiv:2308.08241*.

Xiaozhe Wang, Kate Smith, and Rob Hyndman. 2006. Characteristic-based clustering for time series data. *Data mining and knowledge Discovery*, 13:335–364.

Hao Xue and Flora D Salim. 2023. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*.

A Appendix

A.1 The External Knowledge incorporated in the Prompt

(1) **AirPassengersDataset**: This is a series of monthly passenger numbers for international flights, where each value is in thousands of passengers for that month.

(2) **AusBeerDataset**: This is a quarterly series of beer production, with each value representing the kilolitres of beer produced in that quarter.

(3) **GasRateCO2Dataset**: This time series dataset describes monthly carbon dioxide emissions.

(4) **MonthlyMilkDataset**: This is a time-series data set describing monthly milk production. Each is the average number of tons of milk each cow produces during the month.

(5) **SunspotsDataset**: This is a dataset that records the number of sunspots in each month, where each data is the number of sunspots in that month.

(6) **WineDataset**: This is a dataset of monthly wine production in Australia, where each figure is the number of wine bottles produced in that month.

(7) **WoolyDataset**: This is an Australian yarn production for each quarter, where each value is how many tons of yarn were produced in that quarter.

(8) **HeartRateDataset**: The series contains 1800 uniformly spaced instantaneous heart rate measurements from a single subject.

(9) **Weather**: This dataset contains weather-related data such as temperature, precipitation, humidity, etc. It is commonly used in weather forecasting and climate research.

(10) **COVID Deaths**: This is a collection of data documenting related deaths during the COVID-19 pandemic, including information on the number of deaths, infection rates, etc., for use in outbreak analysis and health policy development.

(11) **Solar Weekly**: This dataset contains weekly data on solar power generation, which can be used to analyze and forecast renewable energy production.

(12) **Tourism Monthly**: This dataset records monthly tourism-related data such as the number of tourists, tourism revenue, etc., for tourism trend analysis and prediction.

(13) **Australian Electricity Demand**: This deals with electricity demand data in Australia, which can be used for electricity market analysis and grid management.

(14) **Pedestrian Counts**: Pedestrian count data are recorded for specific areas, such as city streets, for urban planning and traffic management.

(15) **Traffic Hourly**: Provides hourly traffic flow data for traffic forecasting and urban planning.

(16) **Hospital**: This dataset may contain hospital-related data such as admission rates, bed occupancy rates, etc., for medical resource planning and public health research.

(17) **Fred MD**: It is probably a collection of economic data, such as gross domestic product (GDP), unemployment rate, etc., used for macroeconomic analysis and forecasting.

(18) **Tourism Yearly**: Similar to "Tourism Monthly", but provides annual tourism data for longer-term analysis of tourism trends.

(19) **Tourism Quarterly**: Provides quarterly tourism data between monthly and annual data for medium-term tourism trend analysis.

(20) **US Births**: A dataset that records birth rates in the United States, including information such as the number of births, for use in population research and social policy making.

(21) **NN5 Weekly**: This may be a weekly financial or economic dataset, including stock market data, economic indicators, and so on.

(22) **Traffic Weekly**: Provides weekly traffic flow data for long-term traffic pattern analysis and planning.

(23) **Saugeenday**: This may be a daily degree dataset for a specific region, such as the Saugeen region, and the data involved may include environmental, meteorological, or socioeconomic data.

(24) **CIF 2016**: This may be a specific economic or financial time series dataset collected in 2016 for economic forecasting and analysis.

(25) **Bitcoin**: This dataset deals with the price and transaction data of Bitcoin and is used for cryptocurrency market analysis and forecasting.

(26) **Sunspot**: Data recording sunspot activity for astronomical research and prediction of the solar cycle.

(27) **NN5 Daily**: This is similar to NN5 Weekly, but provides daily data that may refer to financial markets or economic indicators.

A.2 Details of the experiments on the LLMs' Preferences

In this subsection, we provide a comprehensive overview of the experiments conducted to investigate the preferences of LLMs for input time series data. We first describe both the real and synthesized datasets we use, and then detail the methods we use to investigate the preferences of LLMs.

A.2.1 Real Datasets

We conducted experiments on ten commonly used datasets: HeartRateDataset, GasRateCO2Dataset, AirPassengersDataset, AusBeerDataset, MonthlyMilkDataset, SunspotsDataset, WineDataset, WoolyDataset, IstanbulTraffic and TurkeyPower, as detailed in A.1. We apply the Seasonal-Trend decomposition using the LOESS (STL) technique, to decompose the original time series into trend, seasonal, and residual components. In those datasets, we obtain the periods through the nature of the data. For instance, the number of passengers is collected every month in AirPassengersDataset, and it's natural to obtain that the period is 12. For the datasets without explicit periods, such as the IstanbulTraffic, the period was determined through the periodogram, a widely used tool in signal processing assisting the identification of the time series period. The strengths and the model performance can be seen in Table 1.

Subsequently, we compute trend strength Q_T and seasonal strength Q_S to measure all of those components. We use R^2 and MAPE to compute the Pearson correlation coefficients (PCC) across every two indexes and observe a relatively strong correlation between the strengths and model performance, signifying that LLMs perform better when the input time series possesses higher trend and seasonal strength (Shown in Table 2). Notably, GPT-4 achieved a higher absolute PCC compared to GPT-3.5-turbo-instruct. It may be attributed to human feedback during GPT-4 training, as individuals may be more aware of seasonal and trend data. Interestingly, upon calculating the strengths of the output sequences, there is an increase in the seasonal component of the output generated by GPT-4, in contrast to the original test sequences. This may provide some insights for further research into the characteristics of the LLMs.

Following this, we conduct the counterfactual analysis with a systematic permutation to the input time series. We initiate by defining a sliding window that constitutes 10% of the total length of the time seriesm and add Gaussian noise into the data within this window data. Given the variability in the length of time sequences, we first scale the sequence through max-min normalization before incorporating it into the original sequence. The size of the sliding window, W, is defined as: $W = 0.1 \times N$, where N is the total length of the time series. We denote the start point of the window as S and the Gaussian noise as η, to the data within the sliding window, the modified data point, x_i', is given by:

$$x_i' = x_i + \eta \cdot x_i \quad \text{for} \quad i \in [S, S+W], \tag{3}$$

where $\eta \sim \mathcal{N}(\mu, \sigma^2)$, and x_i is the data point after normalization. The change in Mean Squared Error (ΔMSE) is defined as the metric to measure the sensitivity of the LLMs:

$$\Delta\text{MSE} = \text{MSE}_{\text{orig}} - \text{MSE}_{\text{perturbed}} \tag{4}$$

where as $S \to N$, ΔMSE reaches its maximum.

This method allows us to assess the impact of individual segments and thereby infer the interpretability of the time series segments that LLM predominantly focuses on. Our observations suggest that introducing noise towards the end of the time series significantly affects LLM's performance, leading to the inference that LLM tends to give more weight to the latter part of the time series in most instances.

A.2.2 Synthesized Datasets

To further validate our findings, we conducted experiments on synthesized datasets. We formulate the dataset through the expression $y = \alpha * x + \beta * cos(2\pi f * x) + \epsilon$, where x ranges from 0 to 20, and the noise term ϵ follows a normal distribution $\mathcal{N}(0, 1)$. The coefficients α and β were uniformly sampled for 10 samples respectively, with $\beta \in [2, 4)$ and $\alpha \in [0.2, 0.7]$. Hence, both trend and seasonal strengths were carefully confined to $(0.6, 0.9)$,, resulting in more discernible conclusions in the output. The chosen frequency, $f = 1$, was consistent across the experiments. We uniformly sampled 200 data points. Our observations validate that LLMs prefer input sequences with higher strengths and generate time series with higher seasonal strength. Furthermore, we observed that the model performance cannot be entirely explained by the linear combination of the strengths, implying that the influence of these strengths is not independent.

To investigate the influence of the number of periods on model performance, we generated a dataset using the function $y = \alpha * x + \beta_1 * cos(2\pi f_1 * x) + \beta_2 * cos(2\pi f_2 * x) + \epsilon$, where α, β_1, β_2 denote the coefficients of the trend and seasonal components. We set $\beta_1 = 2$, $\beta_2 \in [1, 3)$, and $\alpha \in [0.2, 0.7]$, uniformly sampled for 10 instances each, and $f_1 = 1$ and $f_2 = 3$ to represent the chosen frequencies. Similar to the previous experiments, x ranges from 0 to 20 and ϵ follows the normal distribution $\mathcal{N}(0, 1)$. Our results reveal that LLMs exhibit worse performance when input sequences contain multiple periods, even when the seasonal strength is carefully controlled to be nearly unchanged, as is shown in Figure**??**. This observation may be attributed to the LLMs' challenge in recognizing and adapting to multiple periods, similar to human behavior.

A.3 Detail Prompts

Request output periodic prompts Please output the period size of the predicted sequence as an integer. Output an integer directly.

In-context learning examples For example, the sequence: 0.387952, 8.975192, 5.398713, -6.011139, -9.807413, -0.261663, 9.245404, 5.901009, -6.038525, -9.966693, -0.022652, 9.352823,5.317264, -5.809847, -9.565386, -0.325205, 8.711379, 6.176601, -5.911722, -9.078278.We can get the period number is 5. so, try to find out this period:

A.4 Figures and Tables

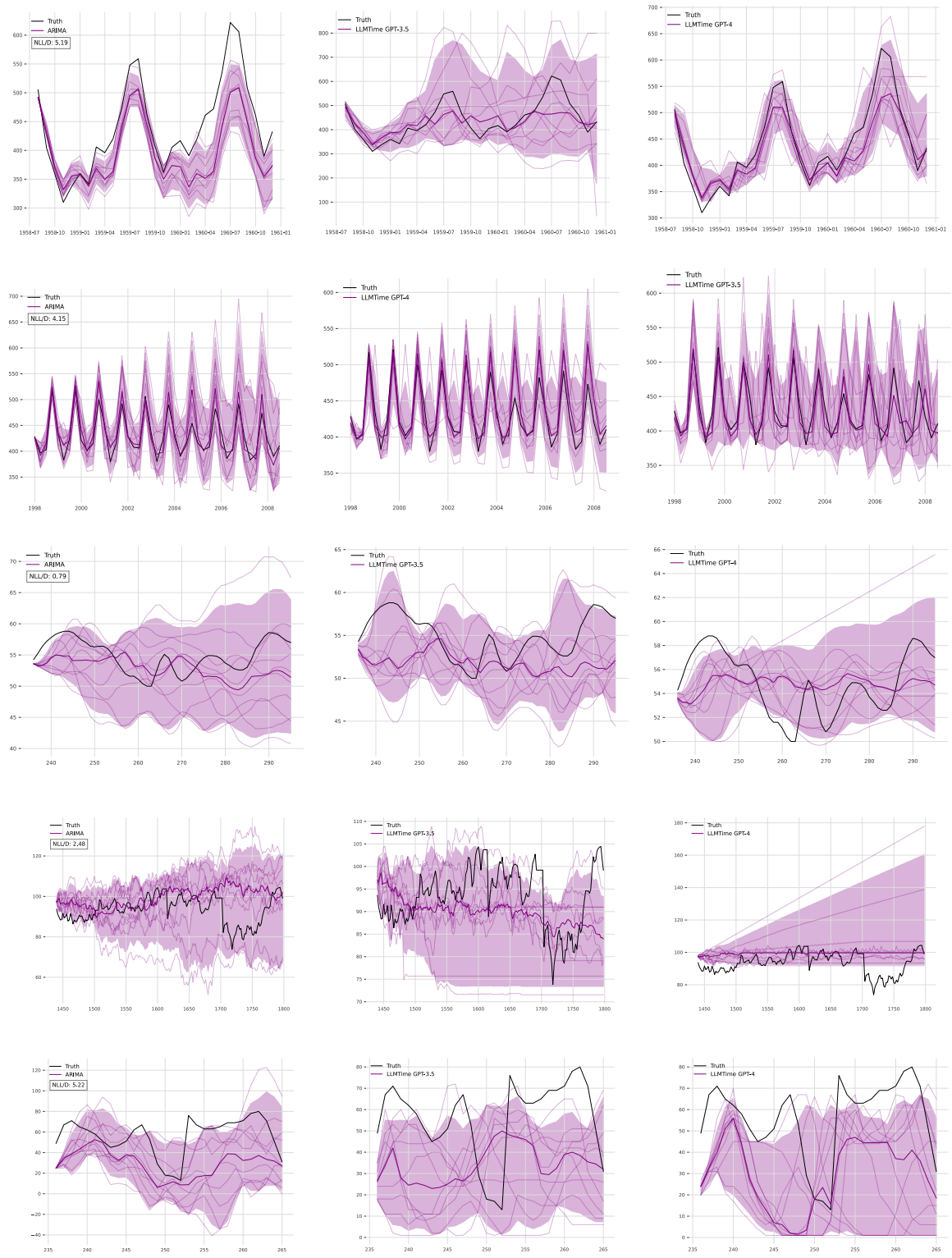


Figure 3: (Left: ARIMA, Center: GPT-3.5, Right: GPT-4)
The predicted results of AirPassengers, AusBeerDataset, GasRateCO2, HeartRate, Istanbul-Traffic datasets.

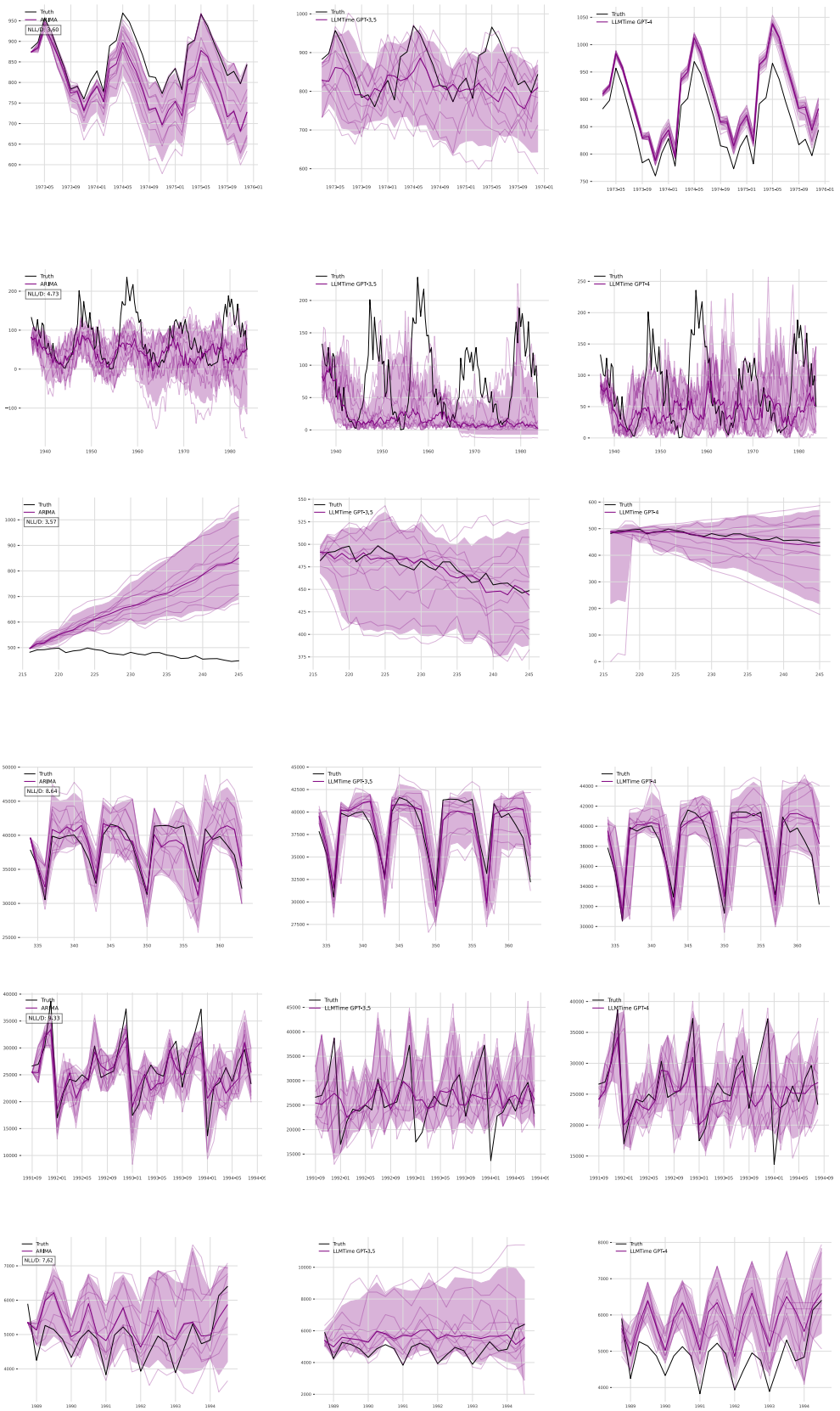
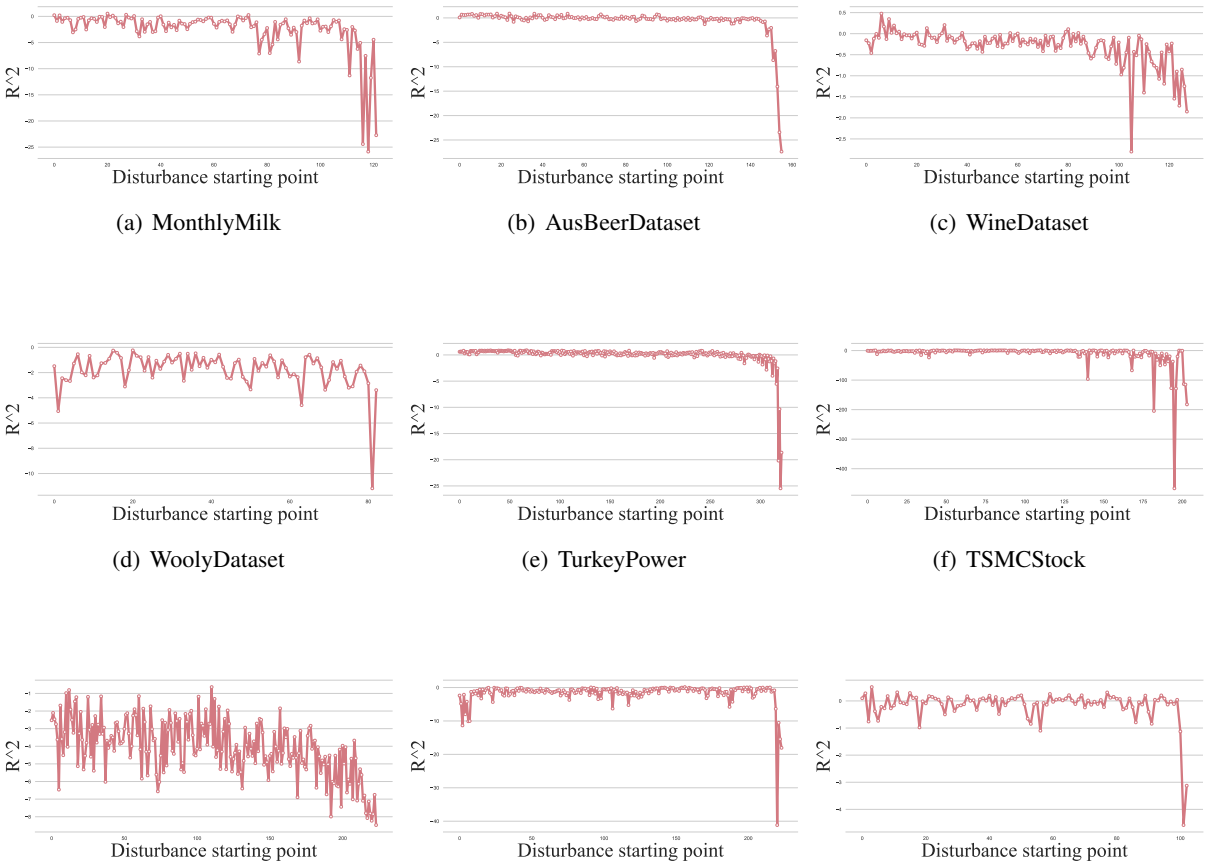


Figure 4: (Left: ARIMA, Center: GPT-3.5, Right: GPT-4)
The predicted results of MonthlyMilk, Sunspots, TSMCStock, TurkeyPower, WineDataset, Wooly datasets.

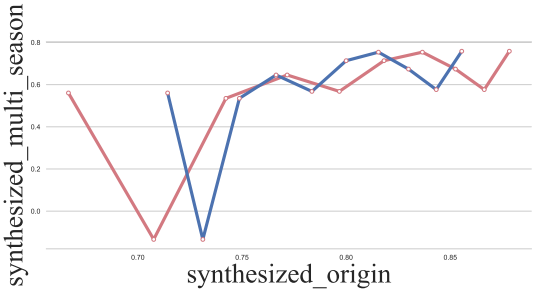


(a) MonthlyMilk (b) AusBeerDataset (c) WineDataset

(d) WoolyDataset (e) TurkeyPower (f) TSMCStock

(g) IstanbulTraffic (h) GasRateCO2 (i) AirPassengers

Figure 5: Results of Counterfactual analysis



(a) MonthlyMilk

Figure 6: Performance of the multi-period time series and single-period time series. x, y represents the strengths and R^2 respectively.

Table 5: The results of external knowledge enhancement.

Models	Dataset	External Knowledge Enhancing			LLMTime Prediction		
		MSE	MAE	MAPE	MSE	MAE	MAPE
GPT-3.5 (gpt-3.5-turbo-1106)	**AirPassengers**	3713.99	50.37	10.88	6244.07	61.39	14.43
	AusBeer	669.01	21.82	5.12	707.43	20.44	4.80
	GasRateCO2	16.47	3.36	5.97	10.88	2.66	4.73
	HeartRate	59.83	6.44	6.75	3863.65	48.90	5.49
	MonthlyMilk	4781.26	55.45	6.25	7507.13	66.28	112.77
	Sunspots	7072.42	62.61	194.29	33094824.12	4209.81	17.69
	Wine	24925885.81	3548.19	14.98	820422.05	760.03	16.51
	Wooly	955708.49	893.02	19.26	65.95	7.05	7.47
	Istanbul-Traffic	888.31	28.16	60.11	1321.44	48.7	7.47
	TSMC-Stock	73.83	7.31	1.54	298.58	15.44	3.23
	Turkey Power	2613198.17	1301.83	3.52	3882704.14	1315.6	3.58
	ETTh1	2.65	1.01	132.13	5.64	2.71	1.625
	ETTm2	2.00	0.89	201.84	3.46	2.17	1.178
	Wind	36.05	4.99	inf	32.07	3.76	10.41
GPT-4 (gpt-4-turbo-preview)	**AirPassengers**	1262.24	30.54	6.80	1286.25	28.04	6.07
	AusBeer	345.59	15.70	3.69	513.49	18.57	4.28
	GasRateCO2	6.99	2.29	4.21	7.27	2.32	4.18
	HeartRate	78.99	6.96	7.90	1878.24	40.79	4.74
	MonthlyMilk	2209.33	44.02	5.12	4442.18	50.75	172.82
	Sunspots	4571.92	50.24	334.30	10105667.50	2438.74	9.85
	Wine	14426570.88	2734.41	10.90	1211754.67	1017.44	21.85
	Wooly	1078968.96	959.42	20.41	85.22	6.89	7.77
	Istanbul-Traffic	954.88	26.92	47.29	1291.17	32.16	6.46
	TSMC-Stock	104.53	8.46	1.79	74.71	6.60	1.39
	Turkey Power	3090055.89	1223.78	3.36	113873.28	814.46	2.17
	ETTh1	2.70	1.06	129.99	4.73	1.53	3.282
	ETTm2	1.18	0.79	291.67	2.30	1.034	1.607
	Wind	1.82	0.81	82.50	28.83	6.32	2.14
Llama-2 (llama-2-13B)	**AirPassengers**	3713.99	50.37	10.88	1286.25	28.04	6.07
	AusBeer	893.56	21.49	4.87	644.82	17.88	4.08
	GasRateCO2	11.38	3.04	5.49	12.78	2.97	5.47
	HeartRate	112.17	7.86	8.93	3317.83	52.71	6.15
	MonthlyMilk	4722.32	60.36	7.05	3410.20	41.40	240.25
	Sunspots	4000.19	46.45	138.69	11788774.81	2770.67	10.90
	Wine	8286095.02	2261.30	8.97	636637.46	742.67	15.54
	Wooly	389685.08	551.18	11.69	64.92	6.39	7.04
	Istanbul-Traffic	979.15	26.70	45.57	1531.37	34.74	7.42
	TSMC-Stock	52105.36	196.02	42.07	2203.97	27.64	27.39
	Turkey Power	3416162.71	1547.49	4.09	2919773.15	1388.10	3.70
	ETTh1	4.15	1.65	408.11	4.84	1.79	3.178
	ETTm2	3.08	1.47	810.56	3.31	2.07	2.153
	Wind	1.78	0.79	116.31	36.78	4.02	12.38

Table 6: Comparison test of traditional prediction methods(Part I)

Dataset	Method	MSE	MAE	MAPE
AirPassengers	Exponential Smoothing	2007.67	37.91	8.10
	SARIMA	2320.47	39.80	8.46
	Cyclical Regression	2028.37	36.70	8.52
	AutoARIMA	8702.09	68.52	13.98
	FFT	3274.46	46.38	10.59
	StatsForecastAutoARIMA	2952.52	45.41	9.71
	Naive Mean	47703.65	204.25	44.61
	Naive Seasonal	6032.80	62.87	14.18
	Naive Drift	6505.79	72.21	17.50
	Naive Moving Average	6032.80	62.87	14.18
	LLMTime with GPT-3.5-Turbo	267.66	3.66	0.99
	LLMTime with GPT-4-Turbo	**133.10**	**2.87**	**0.80**
AusBeer	Exponential Smoothing	703.26	22.80	5.44
	SARIMA	**475.53**	19.07	4.49
	Cyclical Regression	989.31	26.29	6.13
	AutoARIMA	550.05	18.84	4.41
	FFT	7682.56	73.74	17.44
	StatsForecastAutoARIMA	559.46	20.56	4.86
	Naive Mean	1885.72	30.66	6.68
	Naive Seasonal	10828.02	96.35	23.39
	Naive Drift	18507.61	128.23	30.91
	Naive Moving Average	10828.02	96.35	23.39
	LLMTime with GPT-3.5-Turbo	598.45	**5.81**	**1.36**
	LLMTime with GPT-4-Turbo	661.80	7.24	1.63
MonthlyMilk	Exponential Smoothing	564.94	20.23	2.41
	SARIMA	1289.76	32.78	3.87
	Cyclical Regression	3631.53	56.15	6.60
	AutoARIMA	2682.67	42.82	5.20
	FFT	3453.96	45.62	5.48
	StatsForecastAutoARIMA	**186.14**	10.64	1.28
	Naive Mean	19893.07	127.33	14.46
	Naive Seasonal	4870.40	56.00	6.31
	Naive Drift	3998.11	56.06	6.52
	Naive Moving Average	4870.40	56.00	6.31
	LLMTime with GPT-3.5-Turbo	968.69	8.61	1.02
	LLMTime with GPT-4-Turbo	413.63	**4.94**	**0.57**
Sunspots	Moving Average	326750.49	499.78	3129.63
	Exponential Smoothing	326750.49	499.78	3129.63
	SARIMA	2902.72	45.75	466.99
	Cyclical Regression	3917.76	47.84	274.31
	AutoARIMA	4695.67	58.47	709.23
	FFT	3784.56	49.81	150.32
	StatsForecastAutoARIMA	8406.55	72.99	95.18
	Naive Mean	4120.40	49.84	267.22
	Naive Seasonal	4440.63	56.78	688.58
	Naive Drift	5032.77	60.40	724.88
	Naive Moving Average	4440.63	56.78	688.58
	LLMTime with GPT-3.5-Turbo	251.61	**4.27**	20.42
	LLMTime with GPT-4-Turbo	**194.52**	5.30	**16.10**

Table 7: Comparison between the output and test set on trend strength and seasonal strength with GPT-3.5

Dataset Name	**Average Trend Strength** Q_T^{avg}	**Average Seasonal Strength** Q_S^{avg}	**Median Trend Strength** Q_T^{med}	**Median Seasonal Strength** Q_S^{med}	**Test Set Trend Strength** Q_T^{test}	**Test Set Seasonal Strength** Q_S^{test}
AirPassengersDataset	0.97	0.94	1.00	0.99	0.99	0.99
AusBeerDataset	0.73	0.94	0.86	1.00	0.84	0.96
GasRateCO2Dataset	0.80	0.76	0.61	0.87	0.77	0.91
MonthlyMilkDataset	0.93	0.98	0.99	1.00	0.96	0.99
SunspotsDataset	0.73	0.34	0.79	0.37	0.85	0.22
WineDataset	0.46	0.69	0.42	0.77	0.13	0.92
WoolyDataset	0.58	0.73	0.25	0.93	0.92	0.91
HeartRateDataset	0.71	0.76	0.26	0.82	0.63	0.91
IstanbulTraffic	1.00	1.00	1.00	1.00	1.00	1.00
TurkeyPower	0.90	0.99	0.98	1.00	0.76	0.97

Table 8: Comparison of Different Models

Dataset Name	**Average Trend Strength** Q_T^{avg}	**Average Seasonal Strength** Q_S^{avg}	**Median Trend Strength** Q_T^{med}	**Median Seasonal Strength** Q_S^{med}	**Test Set Trend Strength** Q_T^{test}	**Test Set Seasonal Strength** Q_S^{test}
AirPassengersDataset	0.61	0.64	0.62	0.53	0.99	0.99
AusBeerDataset	0.74	0.91	0.91	0.99	0.84	0.96
GasRateCO2Dataset	0.64	0.76	0.06	0.81	0.77	0.91
MonthlyMilkDataset	0.52	0.73	0.58	0.88	0.96	0.99
SunspotsDataset	0.88	0.42	0.93	0.65	0.85	0.22
WineDataset	0.16	0.53	0.12	0.47	0.13	0.92
WoolyDataset	0.52	0.42	0.59	0.54	0.92	0.91
HeartRateDataset	0.80	0.86	0.96	0.83	0.63	0.91
IstanbulTraffic	1.00	1.00	1.00	1.00	1.00	1.00
TurkeyPower	0.61	0.94	0.88	1.00	0.76	0.97