# Time-Prompt: Integrated Heterogeneous Prompts for Unlocking LLMs in Time Series Forecasting

Zesen Wang, Yonggang Li, Lijuan Lan*
Central South University, Changsha, China

***Abstract*—Time series forecasting aims to model temporal dependencies among variables for future state inference, holding significant importance and widespread applications in real-world scenarios. Although deep learning-based methods have achieved remarkable progress, they often struggle with long-term forecasting and few-shot scenarios. Recent research demonstrates that large language models (LLMs) achieve promising performance in time series forecasting, but the full potential of LLMs in understanding time series remains largely untapped. To address this, we propose Time-Prompt, a framework for activating LLMs for time series forecasting. Specifically, we first construct a unified prompt paradigm with learnable soft prompts to guide the LLMs' behavior and textualized hard prompts to enhance the time series representations. Second, to enhance LLMs' comprehensive understanding of the forecasting task, we design a semantic space embedding and cross-modal alignment module to facilitate the fusion of temporal and textual data. Finally, we efficiently fine-tune the LLMs' parameters using time series data. Furthermore, we apply our method to carbon emission forecasting, contributing to the technical advancements aiding global carbon neutrality. Comprehensive evaluations on 6 public datasets and 3 carbon emission datasets demonstrate that Time-Prompt is a powerful framework for time series forecasting. Our code is publicly available at https://github.com/SanMuGuo/Time-Prompt.**

***Index Terms*—time series forecasting, carbon emission forecasting, large language models, prompt engineering, cross-modal alignment**

## I. Introduction

Time series forecasting, which utilizes past observations to predict future data, finds wide applications in domains such as electricity, weather, and carbon emissions [1]. Over the past five years, neural networks with diverse architectures—including linear models [2], [3], convolutional neural networks [4], [5], recurrent neural networks [6], and Transformers [7]—have been applied to time series forecasting. However, existing models face two critical challenges: On one hand, they rely heavily on large-scale labeled samples for parameter optimization, often suffering from limited generalization capability in cross-domain scenarios, which results in performance barriers under few-shot or zero-shot forecasting settings [8], [9]. On the other hand, existing methods struggle to capture long-term temporal dependencies, resulting in suboptimal long-term forecasting performance [10], [11].

This work was supported in part by the National Natural Science Foundation of China under Grant 62394341, Grant 62027811. The authors acknowledge the computational resources provided by the High-Performance Computing Centre of Central South University, China.

Corresponding author: lijuan.lan@csu.edu.cn

Pretrained foundation models, such as Large Language Models (LLMs), have demonstrated strong few-shot and zero-shot learning capabilities in computer vision and natural language processing. As pre-trained models advance, research is increasingly split between harnessing LLMs for time series forecasting [12]–[14] and training time series foundation models [15], [16]. Considering the high computational cost of foundation models, our work focuses on the former. As shown in Figure 1, the most common LLM-based approaches are: (a) Time series-based LLMs: Projecting the time series into the LLM's semantic space via an embedding layer [17]; (b) Prompt-based LLMs: converting temporal data into textual modality through prompt engineering [9]; (c) Multimodal-based LLMs: Processing of time series and text separately [18]; (d) Multimodal-based LLMs: Feeding unified multimodal information into the LLMs [19].

Currently, research primarily focuses on two main types (c) and (d). For example, TimeCMA first encodes the multimodal data, where the time series is processed by an encoder and the text by a frozen LLM. Subsequently, cross-modal alignment is performed, and finally, the prediction result is obtained through a decoder [20]. In contrast, Time-LLM processes both time series and text uniformly through a frozen LLM, which then projects the target value [21].

Despite the state-of-the-art performance of existing methods, Tan et al. [22] argue that LLMs are ill-suited for time series forecasting. However, Li et al. [23] demonstrate that LLMs hold latent capabilities for this task which currently remain dormant. Aligning with this finding, we believe that through effective guidance strategies, the potential of LLMs in time series forecasting can be fully exploited. This is because, for LLMs, textual reasoning is fundamentally a form of discrete-value reasoning. Viewed from this perspective, it is quite similar to the task of time series forecasting, with the main distinction being that time series consists of continuous values. Furthermore, LLMs have exhibited certain mathematical reasoning abilities, which may suggest a latent capacity for handling continuous values.

To unlock the potential of LLMs for time series forecasting, inspired by recent research [20], [21], [24], [25], we propose Time-Prompt, an LLM-empowered framework that activates the potential of LLMs for time series forecasting through time series reprogramming, dual-path prompting, cross-modal alignment, and LoRA. We design a dual-path prompting framework that activates the LLM's specific patterns with soft prompts and enhances its time series representations with

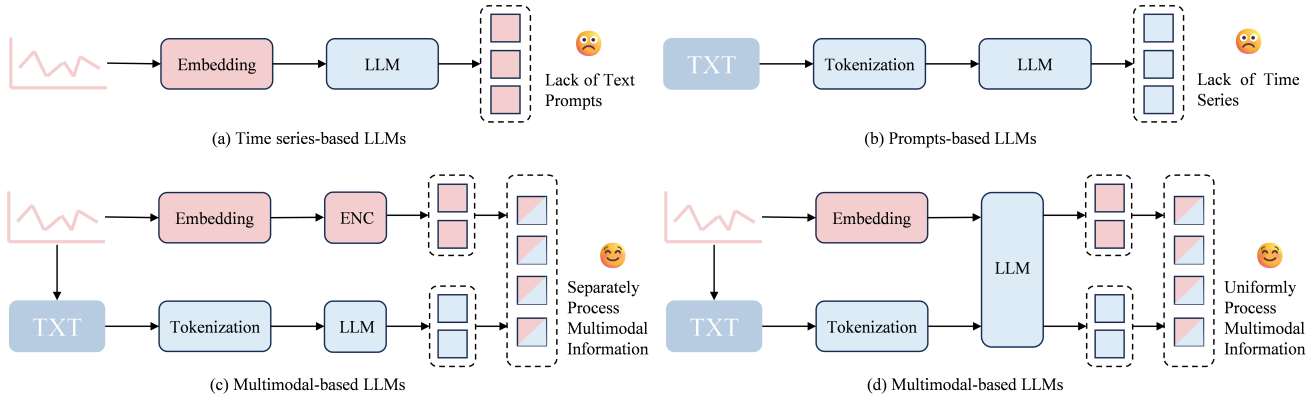


Fig. 1. a) Time series-based LLMs: Lack textual information, failing to leverage LLMs' potential for time series forecasting. (b) Prompt-based LLMs: Struggle to capture fine-grained temporal patterns from textual prompts. (c) Multimodal-based LLMs: Time series are processed via an encoder, and text via LLMs, separately. (d) Multimodal-based LLMs: Time series and text information are processed by LLMs in a unified manner.

hard prompts. We also reprogram the time series to endow it with a text-like representation. Furthermore, we construct a cross-modal alignment module to eliminate the modality gap between text and time, and we efficiently fine-tune the LLM with LoRA to enhance its ability to process continuous values.

In addition, given that carbon emissions have become a globally critical issue, we conduct a cross-scale study on carbon emission forecasting—spanning global, national, and urban levels—to provide technical support for the strategic goals of global carbon neutrality. Notably, beyond publicly available benchmark datasets, we collect the Munich carbon emission dataset to validate the effectiveness of Time-Prompt in practical scenarios [26], [27].

The main contributions of this study are summarized as follows:

- We propose Time-Prompt, a framework that activates the potential of LLMs in time series forecasting through strategies such as dual-path prompting, cross-modal alignment, and efficient fine-tuning.
- We design a dual-path prompting strategy, with learnable soft prompts guiding the LLM's behavior and hard prompts extracting enhanced time series representations.
- Unlike prior methods that simply concatenate multimodal tokens, we construct a semantic space embedding module and a cross-modal alignment module to enable cross-modal fusion of temporal data and textual information, eliminating the modality gap.
- Extensive experiments on 6 benchmark datasets and 3 carbon emission datasets demonstrate that Time-Prompt achieves superior performance.

## II. Related Work

Deep learning has demonstrated superior performance in time series forecasting, primarily categorized into MLP-based, CNN-based, RNN-based, and Transformer-based architectures. Transformer-based models have emerged as the core research direction in recent years by capturing long-term dependencies and inter-variable correlations through self-attention mechanisms, achieving better predictive performance compared to other architectures. iTransformer treats multivariate time series as independent variable sequences, capturing cross-variable dependencies via attention mechanisms [28]. PatchTST divides time series into fixed-length patches, enhancing local pattern recognition through patch-level tokenization [29]. Furthermore, addressing the challenge of non-stationary time series, Non-stationary Transformer proposes series stationarization and de-stationary attention to mitigate non-stationarity in raw time series [30]. Other variants include Autoformer [31], which integrates decomposition frameworks with autocorrelation mechanisms, and Fedformer [32], which employs frequency-domain enhanced decomposition strategies, both significantly improving long-term forecasting accuracy.

Beyond Transformers, DLinear [2] and TimeMixer [3] show that simple linear or multi-scale mixing designs remain competitive. CNN-based methods such as SCINet [33] and WaveNet [34] leverage receptive fields to capture temporal patterns, while Mamba [6] reestablishes state-space models as an efficient paradigm for long-term dependency modeling. However, these deep learning frameworks remain constrained by sample size, exhibiting suboptimal performance in few-shot or zero-shot scenarios.

Recently, leveraging the cross-domain transfer capabilities of LLMs, they have demonstrated exceptional performance in time series forecasting tasks, significantly outperforming traditional methods, especially in few-shot scenarios [35]. As shown in Figure 1, the most prevalent current approaches include: (a) Time series-based LLMs; (b) Prompt-based LLMs; (c) Multimodal-based LLMs; (d) Multimodal-based LLMs. Time series–based LLMs project time series into the semantic space of LLMs via an embedding layer, and subsequently map the embedding vectors to obtain prediction values [17]. In contrast, Prompt-based LLMs construct the time series as a text modality [36]. Lstprompt is a typical Prompt-based method that proposes a prompt construction technique, guiding the LLM to make predictions via a Chain-of-Thought approach [9].

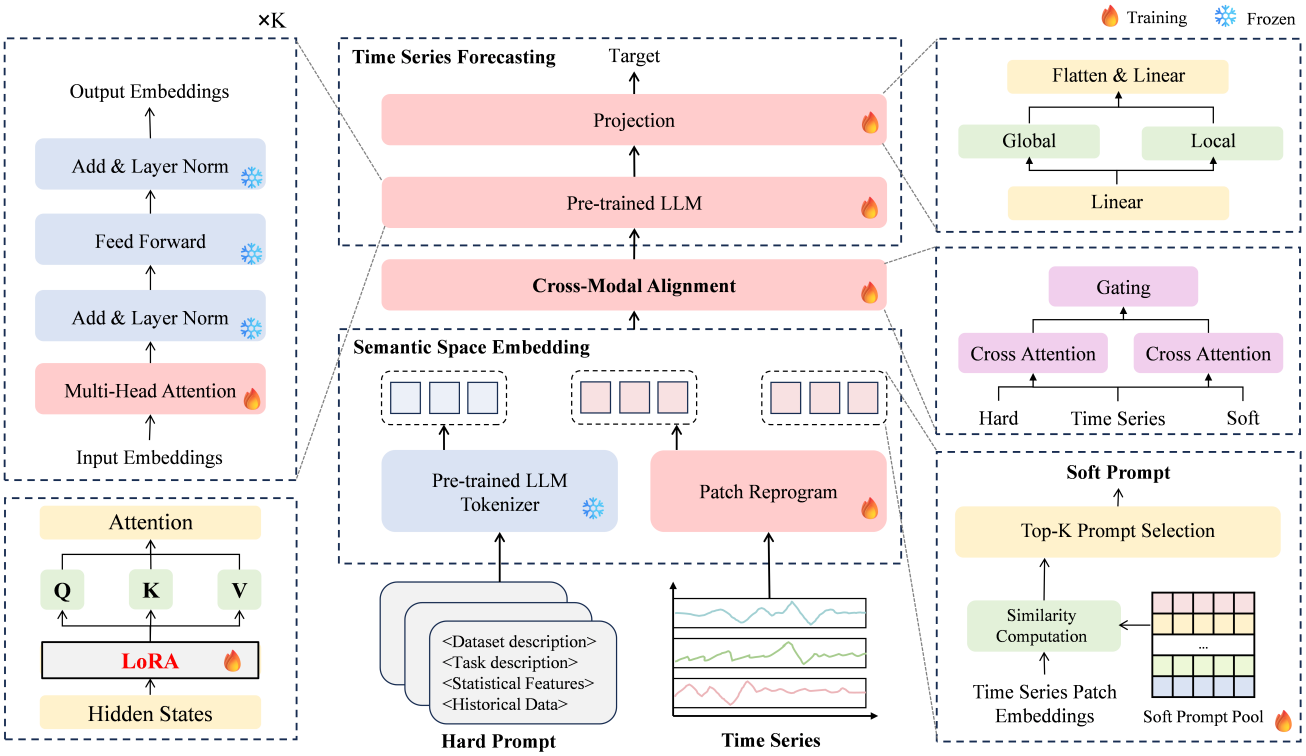


Fig. 2. Time-Prompt framework. First, hard prompts and soft prompts are constructed based on time series. Second, the time series and prompts are embedded into the LLM's semantic space, followed by cross-modal alignment. Finally, the fused information is fed into the pretrained LLM and projected through output layers to generate forecasting results.

However, single-modality time series forecasting often suffers from limited representational capacity, leading to suboptimal prediction performance. Therefore, many studies attempt to integrate Type (a) and Type (b) approaches, leveraging multimodal information to overcome the inherent limitations of single-modality methods. Early efforts predominantly fall into Type (d), which uniformly feeds time series and textual inputs into LLMs to obtain improved feature representations. GPT4MTS, a foundational work in this category, embeds time series and simple text prompts into the semantic space, from which the large model derives a joint representation [37]. Many subsequent works build upon this framework. Time-LLM enhances model comprehension of time series by reprogramming the input time series [21]. Meanwhile, S²IP-LLM and TEMPO employ learnable soft prompts to mark specific temporal patterns, thereby more effectively activating the large model [24], [25].

Subsequently, given the limitations of LLMs in directly representing time series, researchers have shifted towards a dual-path paradigm (Type (c)), where time series are handled by dedicated encoders and text prompts are input into the LLMs. TimeCMA encodes the time series with an encoder and the text with an LLM, followed by cross-modal alignment, and finally generates the prediction via a unified decoder [20]. In a related approach, another study further enhances the textual information using reinforcement learning [38].

The problem with Type (c) is that the LLM only plays an auxiliary role, contributing little to time series forecasting. The issue with Type (d), however, is the LLM's inability to handle continuous values like those in time series. Nevertheless, to bridge the knowledge gaps in these two categories, relevant research targeting both Type (c) [18], [38] and Type (d) [39], [40] continues to emerge. Our method falls into Type (d), aiming to unlock the LLM's latent reasoning ability for continuous-valued forecasting through appropriate guidance.

**Differences from Existing Methods.** Although Time-Prompt builds on established techniques, its novelty lies in the systematic integration of complementary components. In contrast to Time-LLM [21] and TimeCMA [20] that rely solely on hard prompts, or $S^2$IP-LLM [25] and TEMPO [24] that use only soft prompts, we unify both in a dual-path framework where soft prompts guide the LLM's behavioral patterns while hard prompts enrich statistical representations. Existing methods simply concatenate prompts with time-series tokens, which causes feature redundancy; we instead introduce a cross-modal alignment module with learnable gating to adaptively fuse information from multiple sources. Finally, unlike prior works that keep the LLM entirely frozen, we apply LoRA to the attention layers, enabling efficient adaptation to continuous-valued signals—one of the most impactful components in our ablation study.

## III. Methodology

### A. Framework Overview

The overall framework of Time-Prompt is illustrated in Figure 2. First, we construct a dual-path prompting framework for the time series forecasting task. This framework incorporates both hard prompts (including temporal statistical features and historical data) and learnable soft prompts. Our goal is to guide the LLM's internal behavioral patterns via soft prompts and enhance time series feature representation through hard prompts. Second, we embed both the time series and text into the LLM's semantic space via a semantic space embedding module to facilitate better comprehension by the LLM. Subsequently, to address the modality gap between text and time series, we employ cross-modal alignment. Finally, the global and local features of the time series are extracted and projected to obtain the predicted values. Here, the LLM is fine-tuned using LoRA to adapt it from discrete-value reasoning to continuous-value prediction.

### B. Dual-Path Prompting

**Soft Prompts.** Soft prompts include prompt pool construction, similarity computation, and Top-k prompt selection. The core idea is to dynamically select soft prompt vectors most relevant to the time series. First, initialize a prompt pool. Then, compute the similarity between the time series data and prompt keys in the pool, selecting the Top-k most relevant prompts. Notably, the prompt pool is adaptively updated during iterations.

**Prompt Pool Construction.** Given prompt keys $K \in \mathbb{R}^{P\times D}$ and prompt pool $V \in \mathbb{R}^{P\times L\times D}$, where $P$ is the pool size, $L$ is the soft prompt length, and $D$ is the embedding dimension. The prompt pool can be initialized through multiple strategies: zero initialization, uniform distribution initialization, or initialized from the vocabulary of pretrained LLMs. Notably, the prompt pool is updated during backpropagation.

**Similarity Computation.** Given the patch-embedded time series $\hat{X} \in \mathbb{R}^{B\times N\times M\times D}$, where $B$ is the batch size, $N$ is the number of variables, $M$ is the number of patches, and $D$ is the embedding dimension. To obtain a holistic representation for each variable, we aggregate $\hat{X}$ along the patch dimension via mean pooling: $\bar{X} = \text{MeanPool}(\hat{X}) \in \mathbb{R}^{B\times N\times D}$. The similarity between the aggregated representation $\bar{X}$ and the prompt keys $K$ is then computed via dot product:

$$S = \bar{X}K^T \in \mathbb{R}^{B\times N\times P} \tag{1}$$

**Top-k Prompt Selection.** Based on the similarity matrix $S$, we select the indices of the Top-k prompt keys with the highest similarity scores along the pool dimension for each variable:

$$I = \underset{j\in\{1,\dots,P\}}{\text{Top-k}}\ (S_{:,:,j}) \in \mathbb{Z}^{B\times N\times k} \tag{2}$$

The selected indices $I$ are then used to retrieve the corresponding prompt vectors from the prompt pool $V$, yielding the soft prompts:

You are an expert in time series forecasting algorithms. Please obtain accurate forecasting results based on the following information.

**### Instruction:**

**[Dataset description]:** The chinaCarbon dataset was collected daily from January 1, 2019 , to December 31, 2024 , containing 2,192 data points . It includes carbon emissions data for China across categories such as Domestic Aviation , Ground Transport , Industry , International Aviation , Power , Residential , and total carbon emissions.

**[Task description]:** Forecast the next < $H$ > steps from < $T$ > historical observations.

**### Statistical Characteristic:**

**[Statistical features]:** Min value is < min_val >. max value is < max_val >. Mean value is < mean_val >. Standard deviation is < std_val >. The overall trend is < UP or DOWN >. The top 5 lags are < lag_val >.

**### Historical Data:**

| | | | | | | |
|---|---|---|---|---|---|---|
| 4.22 | 2.411 | 1.706 | 0.782 | 2.619 | 1.492 | 19.486 |
| 4.756 | 2.344 | 1.635 | 0.711 | 3.076 | 1.492 | 19.134 |
| 5.626 | 2.88 | 2.523 | 1.208 | 3.076 | 1.492 | 20.682 |

Fig. 3. Hard prompt example on chinaCarbon dataset.

$$\text{softPrompt} = V[I] \in \mathbb{R}^{B\times N\times k\times L\times D} \tag{3}$$

**Hard Prompts.** Hard prompts provide a direct and effective way to activate the LLM's time series forecasting capability, as no modality gap exists between the LLM and the prompts. Hard prompts can also be viewed as enhanced representations of the time series. As illustrated in Figure 3, our paradigm comprises three key components: (1) Instruction, (2) Statistical Characteristics, and (3) Historical Data. The Instruction provides the LLM with the task context, the statistical characteristics provide informative priors that help the LLM recognize temporal patterns, and the Historical Data is the direct textualization of the time series data.

Notably, all statistical features in the hard prompts, including min, max, mean, standard deviation, median, trend, and top-k lags, are computed exclusively from the input look-back window of each individual sample. No information from the forecasting horizon or test set is used, thereby ensuring strict prevention of data leakage. Furthermore, the data normalization scaler is fitted solely on the training split.

### C. Cross-Modal Data Fusion

**Semantic Space Embedding.** This component primarily involves embedding the time series and prompts into the LLM's semantic space to enhance the LLM's understanding of the variables. For soft prompts, they are generated directly in the embedding space and thus require no embedding.

TABLE I
LONG-TERM FORECASTING RESULTS. **BOLD**: THE BEST, UNDERLINE: THE SECOND BEST.

| Methods | Time-Prompt | | Time-LLM | | TimeCMA | | Mamba | | TimesNet | | DLinear | | Autoformer | | Crossformer | | iTransformer | | PatchTST | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | **0.430** | **0.439** | 0.449 | 0.445 | 0.449 | 0.447 | 0.520 | 0.492 | 0.460 | 0.455 | 0.460 | 0.457 | 0.477 | 0.475 | 0.549 | 0.530 | 0.464 | 0.455 | 0.446 | 0.447 |
| ETTh2 | **0.368** | **0.397** | 0.380 | 0.403 | 0.399 | 0.416 | 0.435 | 0.447 | 0.408 | 0.421 | 0.564 | 0.519 | 0.446 | 0.457 | 1.920 | 1.073 | 0.383 | 0.406 | 0.382 | 0.411 |
| ETTm1 | **0.380** | **0.398** | 0.421 | 0.413 | 0.400 | 0.410 | 0.466 | 0.450 | 0.408 | 0.416 | 0.404 | 0.408 | 0.53 | 0.496 | 0.697 | 0.605 | 0.407 | 0.412 | 0.390 | 0.405 |
| ETTm2 | **0.281** | **0.326** | 0.291 | 0.335 | 0.294 | 0.336 | 0.331 | 0.365 | 0.299 | 0.333 | 0.354 | 0.401 | 0.326 | 0.364 | 1.725 | 0.813 | 0.291 | 0.334 | 0.293 | 0.334 |
| Exchange | **0.331** | **0.390** | 0.361 | 0.403 | 0.448 | 0.451 | 0.761 | 0.584 | 0.425 | 0.447 | 0.340 | 0.414 | 0.843 | 0.596 | 0.757 | 0.645 | 0.364 | 0.405 | 0.391 | 0.417 |
| Weather | **0.252** | **0.277** | 0.266 | 0.285 | 0.257 | 0.284 | 0.288 | 0.314 | 0.258 | 0.285 | 0.265 | 0.317 | 0.337 | 0.379 | 0.259 | 0.320 | 0.261 | 0.281 | 0.255 | 0.278 |
| munichCarbon | 0.411 | 0.464 | 0.411 | 0.465 | 0.433 | 0.486 | 0.485 | 0.518 | 0.415 | 0.469 | 0.394 | 0.456 | 0.611 | 0.599 | **0.358** | **0.437** | 0.412 | 0.465 | 0.430 | 0.477 |
| chinaCarbon | **0.571** | **0.532** | 0.627 | 0.557 | 0.659 | 0.586 | 0.660 | 0.592 | 0.665 | 0.588 | 0.689 | 0.642 | 0.642 | 0.608 | 1.177 | 0.834 | 0.667 | 0.579 | 0.604 | 0.546 |
| worldCarbon | 0.617 | 0.544 | 0.649 | 0.565 | **0.546** | **0.518** | 0.687 | 0.614 | 0.751 | 0.639 | 0.754 | 0.698 | 0.653 | 0.603 | 0.619 | 0.539 | 0.735 | 0.616 | 0.615 | 0.547 |

TABLE II
FEW-SHOT LEARNING ON 10% TRAINING DATA. **BOLD**: THE BEST, UNDERLINE: THE SECOND BEST.

| Methods | Time-Prompt | | Time-LLM | | TimeCMA | | Mamba | | TimesNet | | DLinear | | Autoformer | | Crossformer | | iTransformer | | PatchTST | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | **0.452** | **0.445** | 0.477 | 0.460 | 0.549 | 0.504 | 0.716 | 0.572 | 0.847 | 0.632 | 0.632 | 0.548 | 0.510 | 0.499 | 0.583 | 0.531 | 0.701 | 0.570 | 0.477 | 0.455 |
| ETTh2 | **0.381** | **0.404** | 0.396 | 0.414 | 0.416 | 0.429 | 0.427 | 0.448 | 0.465 | 0.459 | 0.435 | 0.458 | 0.427 | 0.446 | 0.800 | 0.642 | 0.445 | 0.447 | 0.391 | 0.412 |
| ETTm1 | **0.394** | **0.402** | 0.424 | 0.411 | 0.425 | 0.427 | 0.511 | 0.467 | 0.474 | 0.454 | 0.496 | 0.467 | 0.623 | 0.532 | 0.427 | 0.439 | 0.491 | 0.456 | 0.414 | 0.406 |
| ETTm2 | **0.284** | **0.327** | 0.297 | 0.339 | 0.302 | 0.342 | 0.318 | 0.359 | 0.311 | 0.352 | 0.382 | 0.431 | 0.311 | 0.355 | 0.592 | 0.527 | 0.304 | 0.345 | 0.290 | 0.333 |
| Exchange | **0.337** | **0.395** | 0.368 | 0.407 | 0.423 | 0.448 | 0.677 | 0.554 | 0.439 | 0.459 | 0.368 | 0.431 | 0.471 | 0.483 | 2.043 | 1.136 | 0.411 | 0.442 | 0.373 | 0.411 |
| Weather | 0.265 | 0.286 | 0.269 | 0.288 | **0.255** | **0.282** | 0.293 | 0.322 | 0.276 | 0.301 | 0.275 | 0.339 | 0.348 | 0.380 | 0.343 | 0.397 | 0.281 | 0.295 | 0.273 | 0.289 |
| munichCarbon | 0.406 | **0.459** | 0.426 | 0.477 | 0.434 | 0.486 | 0.513 | 0.538 | 0.445 | 0.491 | 0.413 | 0.475 | 0.605 | 0.609 | **0.402** | 0.474 | 0.443 | 0.491 | 0.423 | 0.474 |
| chinaCarbon | **0.653** | **0.575** | 0.714 | 0.606 | 0.690 | 0.596 | 0.785 | 0.651 | 0.821 | 0.655 | 0.742 | 0.632 | 0.766 | 0.652 | 1.043 | 0.833 | 0.792 | 0.645 | 0.665 | 0.580 |
| worldCarbon | **0.705** | **0.601** | 0.802 | 0.669 | 0.720 | 0.607 | 0.950 | 0.733 | 0.959 | 0.740 | 0.845 | 0.699 | 0.781 | 0.659 | 0.966 | 0.782 | 0.937 | 0.727 | 0.725 | 0.609 |

**Time Series.** First, segment the time series into patches, then project them into the LLM's embedding space to obtain $\hat{X} \in \mathbb{R}^{B\times N\times M\times D}$, where $M$ is the number of patches and $D$ is the embedding dimension. Given the pre-trained LLM's vocabulary $E \in \mathbb{R}^{V\times D}$, where $V$ is the vocabulary size, we realign time series data with the vocabulary through reprogramming, achieving textual representation of temporal data to enhance LLM comprehension. Specifically, learnable projection matrices generate query matrices $Q_r = \hat{X}W^Q$, key matrices $K_r = EW^K$, and value matrices $V_r = EW^V$ for each attention head. Then, multi-head cross-attention produces the reprogramming temporal representations.

$$\tilde{X} = \mathrm{MHCA}(Q_r, K_r, V_r) \in \mathbb{R}^{B\times N\times M\times D} \tag{4}$$

**Hard Prompt.** Given the hard prompt $T$, map it to discrete tokens using the LLM's pretrained tokenizer. Then, convert these discrete tokens into continuous vector representations via the LLM's input embedding layer. During this process, the LLM's tokenizer remains frozen with no parameter updates during training.

$$\mathrm{hardPrompt} = \mathrm{Embed}(\mathrm{Tokenize}(T)) \in \mathbb{R}^{B\times N\times L\times D} \tag{5}$$

**Cross-Modal Alignment.** This module is designed to achieve cross-modal fusion among the time series data, soft prompts, and hard prompts to eliminate the modality gap. Many existing works simply concatenate these variables, which can lead to severe feature redundancy when the number of prompt tokens is large [21]. Our specific approach is as follows: (a) We use a multi-head cross-attention mechanism to align the soft prompt with the time series, obtaining a fused representation $Z_1$. Similarly, the hard prompt is aligned with the time series to obtain $Z_2$. (b) We then fuse the original time series and the fused representations $Z_1$ and $Z_2$ via a gating mechanism, yielding the final representation $Z$.

$$Z = g_0 \odot \tilde{X} + g_1 \odot Z_1 + g_2 \odot Z_2 \tag{6}$$

where $g_0, g_1, g_2 \in \mathbb{R}$ are learnable scalar gating weights satisfying $g_0 + g_1 + g_2 = 1$ via softmax normalization.

### *D. Time Series Forecasting*

Many existing works rely solely on frozen LLMs. Given the high dimensionality of the LLM's semantic space, they often resort to direct truncation to reduce the computational overhead of the high-dimensional semantic space. This approach, however, is detrimental to activating the LLM's potential for time series forecasting. In this module, we first employ LoRA to efficiently fine-tune the LLM, enhancing its ability to comprehend continuous values. Second, to address the excessively high semantic dimensionality, we project the LLM's output

TABLE III
ABLATION EXPERIMENT. SP: SOFT PROMPT, HP: HARD PROMPT, CMA: CROSS-MODAL ALIGNMENT. **BOLD**: THE BEST.

| Dataset | Metric | Time-Prompt | w/o SP | w/o HP | w/o CMA | w/o LoRA |
|---|---|---|---|---|---|---|
| ETTh1 | MSE | **0.430** | 0.434 | 0.437 | 0.434 | 0.435 |
| | MAE | **0.439** | 0.441 | 0.442 | 0.442 | 0.440 |
| ETTm1 | MSE | **0.380** | 0.383 | 0.384 | 0.386 | 0.386 |
| | MAE | **0.398** | 0.399 | 0.400 | 0.402 | 0.401 |
| munichCarbon | MSE | **0.411** | 0.413 | 0.414 | 0.414 | 0.416 |
| | MAE | **0.464** | 0.466 | 0.467 | 0.467 | 0.468 |
| chinaCarbon | MSE | **0.571** | 0.581 | 0.584 | 0.587 | 0.583 |
| | MAE | **0.532** | 0.540 | 0.540 | 0.544 | 0.538 |
| worldCarbon | MSE | **0.617** | 0.636 | 0.633 | 0.633 | 0.630 |
| | MAE | **0.544** | 0.558 | 0.556 | 0.556 | 0.552 |

representations into a lower-dimensional space to obtain local features, aggregate them via mean pooling to capture global patterns, and then fuse local and global features through a linear projection. The fused representation is passed to an output head that directly produces the final prediction $Y$ in a single forward pass, without any autoregressive decoding.

Regarding the training strategy, the pre-trained GPT-2 backbone is kept frozen, with only the LoRA adapters (applied to the attention layers, rank $r = 8$) receiving gradient updates. The tokenizer and input embedding layer of the LLM also remain frozen throughout training. All other modules are fully trainable, including the patch embedding, reprogramming layer, soft prompt pool, cross-modal alignment, gating fusion weights, feature projection, and output head. This selective fine-tuning strategy preserves the LLM's pre-trained linguistic knowledge while efficiently adapting it to time series through lightweight parameter updates.

## IV. EXPERIMENTS

**Dataset.** We utilize six widely-used public datasets covering multiple domains, including electricity, finance, and weather, as well as three carbon emission datasets with different spatial scales. We conduct extensive experiments across these nine datasets to demonstrate the superiority of Time-Prompt, including tasks such as long-term forecasting and few-shot forecasting. Notably, we introduce a real-world dataset (munichCarbon)—distinct from publicly available benchmarks—to validate the model's effectiveness in practical scenarios.

**Baseline.** We select 9 representative SOTAs from 5 categories: LLM-based: Time-LLM [21], TimeCMA [20]; Transformer-based: Autoformer [31], Crossformer [41], iTransformer [28], PatchTST [29]; CNN-based: TimesNet [5]; RNN-based: Mamba [6]; Linear-based: DLinear [2].

**Implementation.** We use GPT-2 as the backbone LLM [42]. LoRA is applied to the attention projection layers with rank $r = 8$, scaling factor $\alpha = 16$, and dropout rate 0.1. The optimizer is Adam with an initial learning rate of $1 \times 10^{-4}$, scheduled by OneCycleLR. The batch size is 32, training runs for 10 epochs with early stopping patience of 3. The random seeds are fixed at 2021, 2023, and 2025 for reproducibility. All experiments are independently repeated under these three seeds; Tables I, II, and III report the average performance across seeds and across all four forecasting horizons. Tables IV and V report results for a single representative horizon ($H = 96$) to provide fine-grained analysis.

### A. Long-term Forecasting

**Setups.** We conduct experiments on 9 datasets, including 6 classical benchmarks (ETTh1, ETTh2, ETTm1, ETTm2, exchange-rate, weather) and 3 carbon emission datasets (munichCarbon, chinaCarbon, worldCarbon). The input sequence length $T$ is fixed at 96, and we adopt four different forecasting horizons: $H \in \{60, 90, 120, 150\}$ for chinaCarbon and worldCarbon, and $H \in \{96, 192, 336, 720\}$ for the others.

**Results.** The results summarized in Table I, averaged over different forecasting horizons, demonstrate that Time-Prompt outperforms all baselines in most cases. Time-Prompt achieves the best performance in 14 out of 18 benchmark settings. Overall, LLM-based methods demonstrate superior performance compared to conventional deep learning approaches. Specifically, Time-Prompt reduces MSE and MAE by 5.5% and 2.7% over Time-LLM, by 6.3% and 4.2% over TimeCMA, and by 4.3% and 2.4% compared to PatchTST—the strongest deep learning baseline.

### B. Few-shot Forecasting

**Setups.** LLMs demonstrate remarkable few-shot learning capabilities in CV and NLP. We further investigate whether LLMs exhibit similar capabilities in forecasting tasks. Most settings follow the long-term forecasting configuration. This section evaluates model performance under the scenario with only 10% of training samples.

**Results.** The results summarized in Table II, averaged over different forecasting horizons, demonstrate that LLM-based methods significantly outperform deep learning models, confirming that LLMs retain few-shot learning capabilities in forecasting tasks. The superiority of Time-Prompt is even more pronounced in few-shot scenarios compared to long-term forecasting. Under the 10% training data setting, Time-Prompt reduces MSE and MAE by 7.1% and 4.3% over Time-LLM, and by 8.0% and 5.5% over TimeCMA.

### C. Ablation Experiment

As shown in Table III, we conduct an ablation study on multiple datasets (ETTh1, ETTm1, munichCarbon, chinaCarbon, worldCarbon) to analyze the effects of soft prompts (SP), hard prompts (HP), cross-modal alignment (CMA), and LoRA. All experiments follow the long-term forecasting setting, and results are averaged across four prediction horizons. The results indicate that all these modules activate the LLM's forecasting capability, with LoRA and cross-modal alignment contributing more significantly than prompt engineering alone.

### D. Modal Analysis

We experiment with the settings for soft prompt length, hard prompt length, prompt pool size $P$, and Top-$k$ on multiple datasets (ETTh1, munichCarbon, chinaCarbon). The hard

TABLE IV
SENSITIVITY ANALYSIS OF PROMPT HYPERPARAMETERS ON THREE REPRESENTATIVE DATASETS ($H = 96$).

| **soft prompt** | **3** | | **5** | | **10** | | **15** | |
|---|---|---|---|---|---|---|---|---|
| | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | 0.376 | 0.402 | 0.376 | 0.402 | 0.374 | 0.401 | 0.377 | 0.402 |
| munichCarbon | 0.238 | 0.352 | 0.239 | 0.352 | 0.237 | 0.351 | 0.236 | 0.351 |
| chinaCarbon | 0.473 | 0.472 | 0.462 | 0.469 | 0.458 | 0.469 | 0.467 | 0.473 |
| **hard prompt** | **3** | | **5** | | **10** | | **15** | |
| | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | 0.378 | 0.405 | 0.376 | 0.402 | 0.376 | 0.402 | 0.375 | 0.400 |
| munichCarbon | 0.236 | 0.350 | 0.239 | 0.352 | 0.242 | 0.354 | 0.236 | 0.350 |
| chinaCarbon | 0.464 | 0.474 | 0.462 | 0.469 | 0.466 | 0.469 | 0.460 | 0.467 |
| **pool size** | **10** | | **50** | | **100** | | **200** | |
| | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | 0.383 | 0.404 | 0.383 | 0.404 | 0.383 | 0.403 | 0.383 | 0.404 |
| munichCarbon | 0.239 | 0.351 | 0.240 | 0.352 | 0.238 | 0.351 | 0.239 | 0.351 |
| chinaCarbon | 0.464 | 0.470 | 0.464 | 0.470 | 0.486 | 0.482 | 0.488 | 0.484 |
| **Top-k** | **1** | | **3** | | **5** | | **10** | |
| | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | 0.382 | 0.403 | 0.390 | 0.407 | 0.383 | 0.403 | 0.383 | 0.404 |
| munichCarbon | 0.243 | 0.355 | 0.244 | 0.356 | 0.238 | 0.351 | 0.238 | 0.352 |
| chinaCarbon | 0.485 | 0.483 | 0.473 | 0.475 | 0.486 | 0.482 | 0.495 | 0.488 |

prompt length refers to the number of tokens retained after the hard prompt is encoded by the LLM. Due to the relatively large number of tokens, we truncate it to reduce the computational overhead, as the key information is predominantly located in the latter part [20].

As shown in Table IV, the soft prompt length should neither be too short nor too long. Since soft prompts are designed to guide the LLM's behavior, an excessively long sequence of tokens is unnecessary. We set the soft prompt length to 5 and initialize the keys and values using a uniform distribution. For hard prompts, retaining more tokens is generally better, but we keep only the most informative segment considering the trade-off between efficiency and performance. The pool size $P$ shows stable performance across $\{10, 50, 100, 200\}$ on ETTh1 and munichCarbon. For Top-$k$, $k = 5$ provides a good balance, while very small values ($k = 1$) may limit prompt diversity. We default to $P = 100$ and $k = 5$.

### *E. Statistical Stability Analysis*

To verify that Time-Prompt's improvements are statistically robust rather than artifacts of a particular random initialization, we report the mean±std across three random seeds (2021, 2023, 2025) on three representative datasets: ETTh1 and ETTm1 as classical benchmarks, and chinaCarbon as a domain-specific task. The results are summarized in Table V.

Time-Prompt attains the lowest mean error in 5 out of 6 metric/dataset combinations. The only exception occurs on ETTh1 MAE, where TimeCMA marginally leads by $0.0002$—a gap well within one standard deviation of either method and therefore statistically indistinguishable. In all other cases, Time-Prompt produces clear improvements: on ETTh1 MSE, the $0.0061$ gap over the second-best TimeCMA exceeds the sum of their standard deviations ($0.0044$); on chinaCarbon, Time-Prompt reduces MSE by $1.6\%$ relative to Time-LLM and $11.0\%$ relative to TimeCMA, both well outside the observed run-to-run variance. Moreover, Time-Prompt exhibits among the smallest standard deviations across all methods (typically in the $0.001$–$0.01$ range), comparable to or better than the strongest baselines, indicating that the proposed dual-path prompting and cross-modal alignment lead to stable optimization. Overall, these results confirm that the improvements reported in Tables I and II are robust across random seeds rather than incidental outcomes of specific initializations.

TABLE V
STATISTICAL STABILITY ANALYSIS ($H = 96$). RESULTS ARE REPORTED AS MEAN$_{\pm\text{STD}}$. **BOLD**: THE BEST, UNDERLINE: THE SECOND BEST.

| Dataset | Metric | Time-Prompt | Time-LLM | TimeCMA | PatchTST |
|---|---|---|---|---|---|
| ETTh1 | MSE | $\mathbf{0.3764}_{\pm.0008}$ | $0.3841_{\pm.0135}$ | $\underline{0.3825}_{\pm.0036}$ | $0.3870_{\pm.0004}$ |
| | MAE | $\underline{0.4004}_{\pm.0018}$ | $0.4032_{\pm.0052}$ | $\mathbf{0.4002}_{\pm.0020}$ | $0.4024_{\pm.0004}$ |
| ETTm1 | MSE | $\mathbf{0.3173}_{\pm.0086}$ | $0.3285_{\pm.0073}$ | $\underline{0.3265}_{\pm.0047}$ | $0.3281_{\pm.0099}$ |
| | MAE | $\mathbf{0.3641}_{\pm.0033}$ | $0.3676_{\pm.0057}$ | $0.3676_{\pm.0037}$ | $\underline{0.3689}_{\pm.0072}$ |
| chinaCarbon | MSE | $\mathbf{0.4736}_{\pm.0104}$ | $\underline{0.4815}_{\pm.0107}$ | $0.5322_{\pm.0113}$ | $0.5400_{\pm.0010}$ |
| | MAE | $\mathbf{0.4751}_{\pm.0050}$ | $\underline{0.4803}_{\pm.0081}$ | $0.5083_{\pm.0061}$ | $0.5101_{\pm.0014}$ |

## V. CONCLUSION

This paper proposes Time-Prompt to unlock the potential of LLMs in time series forecasting. This method employs a dual-path prompting mechanism. It utilizes soft prompts to guide the reasoning patterns of the LLM and enhances representations with hard prompts and time series reprogramming. Subsequently, it bridges the modality gap between time series and text through semantic space embedding and cross-modal alignment. Finally, the LLM's capability for processing continuous values is enhanced through LoRA fine-tuning. Extensive experiments on multiple real-world datasets validate the effectiveness of Time-Prompt in time series forecasting.

While Time-Prompt demonstrates that LLMs can be effectively activated for time series forecasting, a gap remains between LLM-based and foundation models. We attribute this primarily to the text-centric training paradigm of LLMs, which requires targeted adaptation strategies for forecasting tasks. Moreover, according to the experiments in this paper, we find that fine-tuning is more effective than prompt engineering in activating predictive capabilities, especially in few-shot scenarios. Closing this gap, whether through more expressive fine-tuning or dedicated time series foundation models, is a promising direction for future work.


## ACKNOWLEDGMENT

During the preparation of this work, the authors used ChatGPT in order to polish the language and improve readability. After using this service, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.


## REFERENCES


[1] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, "Deep time series models: A comprehensive survey and benchmark," *arXiv preprint arXiv:2407.13278*, 2024.

[2] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, 2023, pp. 11 121–11 128.

[3] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. Zhou, "Timemixer: Decomposable multiscale mixing for time series forecasting," in *International Conference on Learning Representations*, 2024.

[4] H. Wang, J. Peng, F. Huang, J. Wang, J. Chen, and Y. Xiao, "Micn: Multi-scale local and global context modeling for long-term series forecasting," in *The eleventh international conference on learning representations*, 2023.

[5] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *International Conference on Learning Representations*, 2023.

[6] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, vol. 30, 2017.

[8] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, "Large language models are zero-shot time series forecasters," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 19 622–19 635.

[9] H. Liu, Z. Zhao, J. Wang, H. Kamarthi, and B. A. Prakash, "Lstprompt: Large language models as zero-shot time series forecasters by long-short-term prompting," *arXiv preprint arXiv:2402.16132*, 2024.

[10] P. Chen, Y. Zhang, Y. Cheng, Y. Shu, Y. Wang, Q. Wen, B. Yang, and C. Guo, "Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting," *arXiv preprint arXiv:2402.05956*, 2024.

[11] J. Ni, Z. Liu, S. Wang, M. Jin, and W. Jin, "Timedistill: Efficient long-term time series forecasting with mlp via cross-architecture distillation," *arXiv preprint arXiv:2502.15016*, 2025.

[12] C. Shyalika, H. K. Bagga, A. Bhatt, R. Prasad, A. A. Ghazo, and A. Sheth, "Time series foundational models: Their role in anomaly detection and prediction," *arXiv preprint arXiv:2412.19286*, 2024.

[13] M. Jin, Q. Wen, Y. Liang, C. Zhang, S. Xue, X. Wang, J. Zhang, Y. Wang, H. Chen, X. Li *et al.*, "Large models for time series and spatio-temporal data: A survey and outlook," *arXiv preprint arXiv:2310.10196*, 2023.

[14] X. Zhang, R. R. Chowdhury, R. K. Gupta, and J. Shang, "Large language models for time series: A survey," *arXiv preprint arXiv:2402.01801*, 2024.

[15] M. Goswami, K. Szafer, A. Choudhry, Y. Cai, S. Li, and A. Dubrawski, "Moment: A family of open time-series foundation models," in *International Conference on Machine Learning*, 2024.

[16] A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor *et al.*, "Chronos: Learning the language of time series," *arXiv preprint arXiv:2403.07815*, 2024.

[17] T. Zhou, P. Niu, L. Sun, R. Jin *et al.*, "One fits all: Power general time series analysis by pretrained lm," in *Advances in neural information processing systems*, vol. 36, 2023, pp. 43 322–43 355.

[18] S. Zhou, H. Schöner, H. Lyu, E. Fouché, and S. Wang, "Balm-tsf: Balanced multimodal alignment for llm-based time series forecasting," in *International Conference on Information and Knowledge Management*, 2025.

[19] T. Zhao, X. Chen, and M. Sun, "Enhancing time series forecasting via multi-level text alignment with llms," in *Database Systems for Advanced Applications*, 2025.

[20] C. Liu, Q. Xu, H. Miao, S. Yang, L. Zhang, C. Long, Z. Li, and R. Zhao, "Timecma: Towards llm-empowered time series forecasting via cross-modality alignment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[21] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan *et al.*, "Time-llm: Time series forecasting by reprogramming large language models," in *The International Conference on Learning Representations*, 2024.

[22] M. Tan, M. Merrill, V. Gupta, T. Althoff, and T. Hartvigsen, "Are language models actually useful for time series forecasting?" in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 60 162–60 191.

[23] Z. Li, X. Qiu, P. Chen, Y. Wang, H. Cheng, Y. Shu, J. Hu, C. Guo, A. Zhou, C. S. Jensen *et al.*, "Tsfm-bench: A comprehensive and unified benchmark of foundation models for time series forecasting," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 5595–5606.

[24] D. Cao, F. Jia, S. O. Arik, T. Pfister, Y. Zheng, W. Ye, and Y. Liu, "Tempo: Prompt-based generative pre-trained transformer for time series forecasting," in *The International Conference on Learning Representations*, 2024.

[25] Z. Pan, Y. Jiang, S. Garg, A. Schneider, Y. Nevmyvaka, and D. Song, "$s^2$ip-llm: Semantic space informed prompt learning with llm for time series forecasting," in *Forty-first International Conference on Machine Learning*, 2024.

[26] L. Lan, J. Chen, Y. Wu, Y. Bai, X. Bi, and Y. Li, "Self-calibrated multiharmonic co 2 sensor using vcsel for urban in situ measurement," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 4, pp. 1140–1147, 2018.

[27] L. Lan, J. Chen, X. Zhao, and H. Ghasemifard, "Vcsel-based atmospheric trace gas sensor using first harmonic detection," *IEEE Sensors Journal*, vol. 19, no. 13, pp. 4923–4931, 2019.

[28] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," in *International Conference on Learning Representations*, 2024.

[29] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *International Conference on Learning Representations*, 2023.

[30] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," in *Advances in neural information processing systems*, vol. 35, 2022, pp. 9881–9893.

[31] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Advances in neural information processing systems*, vol. 34, 2021, pp. 22 419–22 430.

[32] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International conference on machine learning*. PMLR, 2022, pp. 27 268–27 286.

[33] M. Liu, A. Zeng, M. Chen, Z. Xu, Q. Lai, L. Ma, and Q. Xu, "Scinet: Time series modeling and forecasting with sample convolution and interaction," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 5816–5828.

[34] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu *et al.*, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, vol. 12, 2016.

[35] Y. Liu, G. Qin, X. Huang, J. Wang, and M. Long, "Autotimes: Autoregressive time series forecasters via large language models," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 122 154–122 184.

[36] H. Xue and F. D. Salim, "Promptcast: A new prompt-based learning paradigm for time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6851–6864, 2023.

[37] F. Jia, K. Wang, Y. Zheng, D. Cao, and Y. Liu, "Gpt4mts: Prompt-based large language model for multimodal time-series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 23 343–23 351.

[38] C. Su, Y. Tian, Y. Song, and Y. Zhang, "Text reinforcement for multimodal time series forecasting," in *International Conference on Information and Knowledge Management*, 2025.

[39] P. Chen, Y. Wang, Y. Shu, Y. Cheng, K. Zhao, Z. Rao, L. Pan, B. Yang, and C. Guo, "Cc-time: Cross-model and cross-modality time series forecasting," in *International Conference on Information and Knowledge Management*, 2025.

[40] W. Zhang, J. Ye, Z. Li, J. Li, and F. Tsung, "Dualtime: A dual-adapter multimodal language model for time series representation," in *International Conference on Information and Knowledge Management*, 2025.

[41] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The eleventh international conference on learning representations*, 2023.

[42] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.