CALF: Aligning LLMs for Time Series Forecasting via Cross-modal Fine-Tuning

Peiyuan Liu[1,*], **Hang Guo**[1,*], **Tao Dai**[2,†], **Naiqi Li**[1,†], **Jigang Bao**[1],
Xudong Ren[1], **Yong Jiang**[1], **Shu-Tao Xia**[1,3]
[1]Tsinghua Shenzhen International Graduate School
[2]Shenzhen University
[3]Pengcheng Laboratory
{lpy23, guo-h23}@mails.tsinghua.edu.cn, {daitao.edu, linaiqi.thu}@gmail.com

Abstract

Deep learning (e.g., Transformer) has been widely and successfully used in multivariate time series forecasting (MTSF). Unlike existing methods that focus on training models from a single modal of time series input, large language models (LLMs) based MTSF methods with cross-modal text and time series input have recently shown great superiority, especially with limited temporal data. However, current LLM-based MTSF methods usually focus on adapting and fine-tuning LLMs, while neglecting the *distribution discrepancy* between textual and temporal input tokens, thus leading to sub-optimal performance. To address this issue, we propose a novel **C**ross-Mod**A**l **L**LM **F**ine-Tuning (CALF) framework for MTSF by reducing the distribution discrepancy between textual and temporal data, which mainly consists of the temporal target branch with temporal input and the textual source branch with aligned textual input. To reduce the distribution discrepancy, we develop the cross-modal match module to first align cross-modal input distributions. Additionally, to minimize the modality distribution gap in both feature and output spaces, feature regularization loss is developed to align the intermediate features between the two branches for better weight updates, while output consistency loss is introduced to allow the output representations of both branches to correspond effectively. Thanks to the modality alignment, CALF establishes state-of-the-art performance for both long-term and short-term forecasting tasks with low computational complexity, and exhibits favorable few-shot and zero-shot abilities similar to that in LLMs.

Code — https://github.com/Hank0626/CALF

Introduction

Multivariate time series forecasting (MTSF) plays a crucial role in the domain of time series analysis and has further boasted a wide range of real-world applications including weather forecasting (Angryk et al. 2020), energy prediction (Demirel et al. 2012), financial modeling (Patton 2013). To achieve more accurate forecasting performance, numerous deep learning-based MTSF methods trained on a single modal of time series input have been developed in recent years (Wu et al. 2023; Zhang and Yan 2023; Nie et al. 2023; Zeng et al. 2023; Zhou et al. 2022; Wen et al. 2022; Dai et al. 2024; Qiu et al. 2024b) and have gained great success.

However, previous single-modal MTSF methods may suffer from overfitting problems, due to the limited training data, thus limiting their real applications. To relieve such issues, some pioneering works attempt to introduce the powerful Large Language Models (LLMs) models in time series forecasting by employing the strong context modeling ability of LLMs. For example, Zhou et al. (Zhou et al. 2023) proposed a unified time series analysis framework by adapting and fine-tuning LLMs. Building upon this, other works have introduced additional enhancements to further expand the capabilities of LLMs in time series forecasting, including refining fine-tuning methods (Chang, Peng, and Chen 2023), sequence decomposition (Cao et al. 2024), and the incorporation of textual prompts (Jin et al. 2024). Benefiting from the large-scale pre-training, LLM-based methods not only exhibit strong context modeling capabilities but also help mitigate the problem of overfitting.

Despite the great success of LLM-based MTSF methods, existing methods suffer unfavorable results on common MTSF benchmarks, as well as pool performance on few-shot or zero-shot tasks. As shown in Fig. 1, we visualize the distribution of the textual and temporal tokens of existing LLM-based MTSF methods, and it can be seen that the temporal tokens in existing LLM-based methods cannot align well with the original textual tokens from LLMs (Zhou et al. 2023; Sun et al. 2024; Jin et al. 2024; Chang, Peng, and Chen 2023). As a result, this distribution discrepancy contributes in part to the sub-optimal performance of the current approaches. We argue that the main reason for the above misalignment is due to the fact that existing methods only investigate the input side to match the feature dimensions between time series and LLM, and this simple solution hinders the manipulation of LLM's powerful generalization capabilities. The above observations motivate us to develop a multi-level cross-modal aligning framework to alleviate the distribution discrepancy between temporal modal input and textural modal weights.

Inspired by the above observations, we propose a **C**ross-Mod**A**l **L**LM **F**ine-Tuning (CALF) framework, which employs cross-modal fine-tuning to allow more comprehensive alignment between temporal target modalities and tex-

*These authors contributed equally.
†Corresponding author: Tao Dai and Naiqi Li.

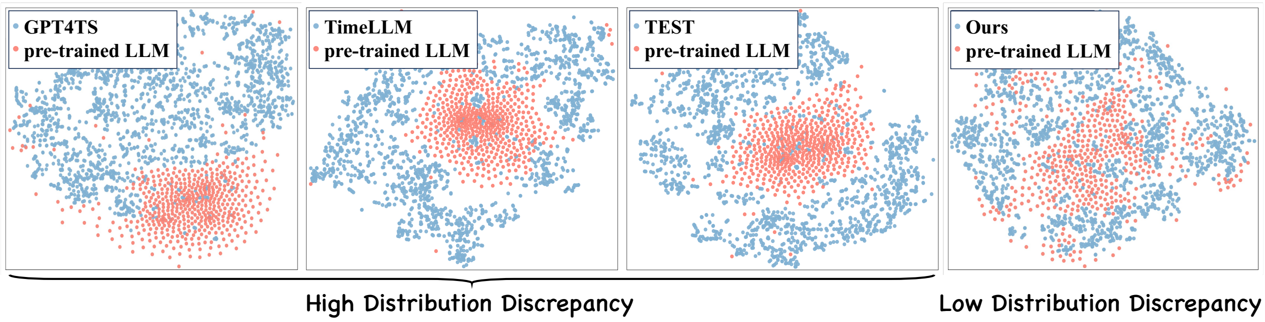

Figure 1: The t-SNE visualization of pre-trained word token embeddings of LLM with the hidden features from the penultimate layer from GPT4TS (Zhou et al. 2023), TimeLLM (Jin et al. 2024), TEST (Sun et al. 2024), and ours of ETTh2 dataset. Current LLM-based methods either use linear layers to project time series to the LLM's feature dimension (Zhou et al. 2023) or employ cross-attention and contrastive learning techniques (Jin et al. 2024; Sun et al. 2024), which address only the input side and overlook alignment in the deeper layers. Our CALF achieves better alignment through multi-level cross-modal fine-tuning.

tual source modalities. Specifically, CALF consists of two branches: the temporal target branch and the textual source branch. The temporal target branch processes time series information, while the textual source branch extracts and adapts information from pre-trained LLMs using aligned textual modal tokens. To bridge the modality gap between these branches, we introduce three meticulously designed cross-modal fine-tuning techniques (see Fig. 2): (1) **Cross-Modal Match Module** integrates time series and textual inputs through principal word embedding extraction and a cross-attention mechanism, ensuring efficient alignment of the marginal input distribution between time series and text; (2) **Feature Regularization Loss** aligns the outputs of each intermediate layer, ensuring that gradients at every layer are more effectively guided for better weight updates; (3) **Output Consistency Loss** ensures that the output representations of textual and temporal series modalities correspond effectively, resolving discrepancies in the representation space and maintaining consistent semantic context for time series data. Through a more comprehensive alignment, our CALF consistently achieves state-of-the-art performance in both long-term and short-term forecasting across multiple datasets, demonstrating excellent few/zero-shot generalization capabilities, while maintaining significantly low complexity.

The contributions of this paper are threefold:

1. We identify the significant distribution discrepancies between textual and temporal modalities in existing LLM-based forecasting models and highlight the importance of addressing this misalignment for improved performance.
2. We propose CALF, a novel framework that employs cross-modal fine-tuning techniques to comprehensively align temporal and textual data. The framework includes three specific methods: the Cross-Modal Match Module for aligning input distributions, Feature Regularization Loss for better gradient guidance and weight updates, and Output Consistency Loss for resolving output representation space discrepancies and maintaining consistent semantic context.
3. Extensive experiments on eight real-world datasets demonstrate that CALF achieves state-of-the-art performance on both long-term and short-term time series forecasting tasks, with favorable generalization ability and low computational complexity.

Related Work

Time Series Forecasting

In recent years, deep learning has significantly revolutionized the field of time series forecasting, with a plethora of methods emerging to enhance predictive accuracy (Zeng et al. 2023; Das et al. 2023; Wu et al. 2023; Wang et al. 2022; Liu et al. 2023a). Among these, Transformer-based models have emerged as the frontrunners, offering unparalleled performance due to their exceptional ability to model complex dependencies in data (Nie et al. 2023; Zhang and Yan 2023; Woo et al. 2022; Wu et al. 2021; Zhou et al. 2022; Dai et al. 2024). However, they often have limitations due to the scarcity of training data, overfitting in specific domains, and the necessity for intricate architectural designs.

To tackle these challenges, the integration of LLMs into time series forecasting has emerged as a novel and promising direction. This approach leverages the extensive pre-training of LLMs to enhance the context-modeling capacity in time series analysis. A groundbreaking framework proposed by Zhou et al. (Zhou et al. 2023) first demonstrated the potential of adapting LLMs for time series analysis. Following this paradigm, subsequent research has introduced further refinements and innovations. For example, Chang et al. (Chang, Peng, and Chen 2023) introduced a novel two-stage fine-tuning method and integrated time-series patching with additional temporal encoding into pre-trained LLMs. Cao et al. (Cao et al. 2024) incorporated decomposition of time series and selection-based prompts for adapting to non-stationary data. However, these works often directly input time series data into LLMs, overlooking the misalignment between time series and textual modalities. Some works have attempted to address this issue. Sun et al. (Sun et al. 2024) aligned time series data with LLM embeddings using contrastive learning and employed soft prompts for effective time series task handling. Jin et al. (Jin et al. 2024) reprogrammed time series input with text prototypes and enriches it using context as a prefix for LLM alignment. Despite these efforts, the alignment strategies have not been sufficiently effective.

Cross-Modal Fine-tuning

The objective of cross-modal fine-tuning is to apply models pre-trained on data-rich modalities to data-scarce modalities, addressing issues of data insufficiency and poor generalization (Shen et al. 2023). Many existing works focus on transferring LLMs to other modalities, such as vision (Kiela et al. 2019; Verma et al. 2024), audio (Jin et al. 2023; Hassid et al. 2024), and biology (Vinod, Chen, and Das 2023; Xiao et al. 2021). These efforts provide initial evidence of the cross-modal transfer capacity of pre-trained models. In the domain of time series, current research primarily leverages the powerful contextual modeling capabilities of LLMs to fine-tune them for improved forecasting performance (Zhou et al. 2023; Jin et al. 2024; Cao et al. 2024; Chang, Peng, and Chen 2023; Sun et al. 2024), often neglecting the gap between the input and output distributions of language and time series modalities. In this work, we apply cross-modal fine-tuning techniques to address the challenge of transferring pre-trained language model knowledge to the time series modality.

Methodology

As shown in Fig. 2, our proposed CALF consists of two branches: the textual source branch and the temporal target branch. In concrete, the textual source branch takes the aligned text tokens X_{text} as input and employs L stacked pre-trained LLM layers to obtain the hidden text feature F^l_{text}, where $l = \{1, \cdots, L\}$. A task-specific head is used to generate the output Y_{text}. Meanwhile, the temporal target branch works with the projected time series tokens X_{time}, and uses the same number of layers L with identical pre-trained weights as the textual source branch to obtain the hidden time feature F^l_{time}. The output of this branch is denoted as Y_{time}. To bridge the modality gap between these two branches, we utilize three cross-modal fine-tuning techniques to fine-tune the temporal target branch: the **Cross-Modal Match Module**, the **Feature Regularization Loss**, and the **Output Consistency Loss**. Detailed descriptions of these techniques will be provided in the following section.

Cross-Modal Match Module

As demonstrated in previous work (Mikolov et al. 2013), the matrices of word embedding layers in pre-trained LLMs constitute a well-structured context representation space, *e.g.*, semantic distances between different words can be quantified through vector similarity. This word embedding layer represents the input distribution of the language modality in pre-trained LLMs. Despite this promising property, previous LLM-based time series methods often overlook this distribution, instead projecting the time series data to match the input dimensions of the language model (Zhou et al. 2023; Cao et al. 2024; Chang, Peng, and Chen 2023).

In this work, we attempt to align the input distribution of time series with the word embedding of LLMs. Therefore, we propose a cross-modal match module to deal with this problem. Specifically, given a multivariate time series $I \in \mathbb{R}^{T \times C}$ as input, where T is the input sequence length and C is the number of variants, we first use the embedding layer similar to (Liu et al. 2024), followed by Multi-head Self Attention (MHSA) to get the projected time tokens X_{time}:

$$X_{time} = \mathrm{MHSA}(\mathrm{Embedding}(I)) \in \mathbb{R}^{C \times M}, \quad (1)$$

where M is the feature dimension of pre-trained LLMs. The embedding layer $\mathrm{Embedding}(\cdot)$ performs a channel-wise dimensional mapping from T to M.

After that, we consider using cross-attention to align X_{time} from the temporal modality and the word embedding dictionaries $\mathcal{D} \in \mathbb{R}^{|\mathcal{A}| \times M}$, where $|\mathcal{A}|$ is the size of the alphabet, to the textual modality. However, due to $|\mathcal{A}|$ is usually huge, *e.g.*, 50257 in GPT2 (Radford et al. 2019), directly using cross-attention incurs significant cost. Observing that semantic-similar words form "synonym clusters", we propose a principal word embedding extraction strategy, which uses the cluster center to represent surrounding words, to reduce the number of word entries. Specifically, we use Principal Component Analysis (PCA) to perform dimension reduction on $\mathcal{D}$ to obtain the principal word embeddings $\hat{\mathcal{D}} \in \mathbb{R}^{d \times M}$,

$$\hat{\mathcal{D}} = \mathrm{PCA}(\mathcal{D}), \quad (2)$$

where d is a pre-defined low dimension and satisfies $d \ll |\mathcal{A}|$.

It is worth noting that this process needs to be done only once before model training and does not incur much training overhead. We then use Multi-head Cross-Attention with $\hat{\mathcal{D}}$ as key and value, and X_{time} as query to align the principal word embeddings and temporal tokens to obtain the aligned text tokens $X_{text} \in \mathbb{R}^{C \times M}$,

$$\begin{aligned} X_{text} &= \mathrm{Softmax}(\frac{QK^T}{\sqrt{C}})V, \\ Q &= X_{time}W_q, K = \hat{\mathcal{D}}W_k, V = \hat{\mathcal{D}}W_v, \end{aligned} \quad (3)$$

where W_q, W_k and $W_v \in \mathbb{R}^{M \times M}$ are the projection matrices for the query (Q), key (K), and value (V), respectively.

Feature Regularization Loss

The pre-trained weights in LLMs are based on their original textual modality data. To more effectively adapt these pre-trained weights to time series data, we align the outputs of each intermediate layer in the temporal target branch with those of the textual source branch. This alignment process, facilitated by feature regularization loss, matches the intermediate features between two branches, allowing gradients at each intermediate layer to be more effectively guided for better weight updates. Formally, given F^l_{text} and F^l_{time} from the outputs of the l-th Transformer block in the textual

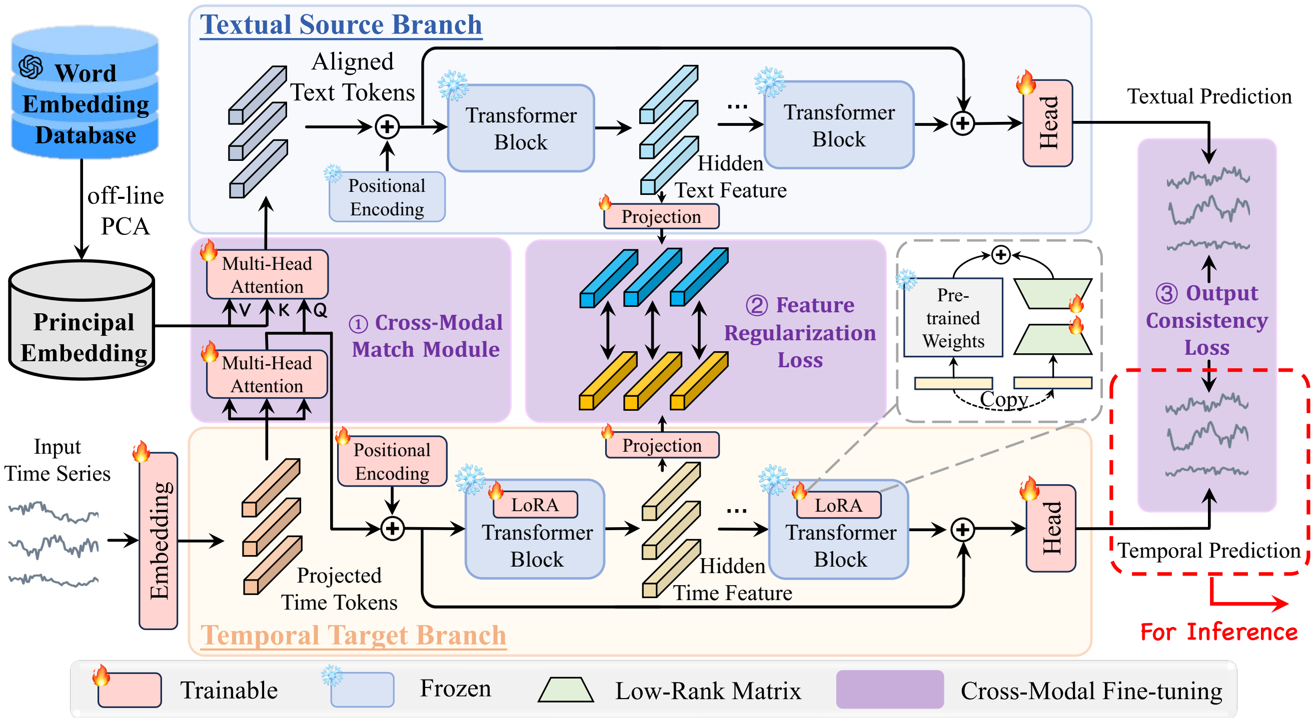

Figure 2: An overview of the proposed cross-modal fine-tuning framework. Above is the Textual Source Branch, and below is the Temporal Target Branch. To bridge the modality gap, the framework employs three cross-modal fine-tuning techniques: ①Cross-Modal Match Module, ②Feature Regularization Loss, and ③Output Consistency Loss.

source branch and temporal target branches, respectively, the feature regularization loss is defined as:

$$\mathcal{L}_{feature} = \sum_{l=1}^{L} \gamma^{(L-l)} \mathrm{sim}(\phi_l^{text}(F_{text}^l), \phi_l^{time}(F_{time}^l)), \quad (4)$$

where γ is a hyper-parameter that controls the loss scale from different layers, and $\mathrm{sim}(\cdot,\cdot)$ is a chosen similarity function, such as L_1 loss. Following (Chen et al. 2020), we introduce two trainable projection layers $\phi_l^{text}(\cdot)$ and $\phi_l^{time}(\cdot)$ to transform the features from textual and temporal modalities to the shared representation space.

Output Consistency Loss

Building on the feature regularization loss, we further ensure consistent semantic context between the textual and temporal modalities. Output consistency loss achieves this by ensuring that the output distributions correspond effectively, resolving discrepancies in the representation space. This alignment maintains a coherent and unified semantic representation for both the time series and textual data, facilitating more accurate and reliable model predictions. Specifically, given the outputs Y_{text} and Y_{time} from the textual source branch and temporal target branch respectively, the output consistency loss is defined as:

$$\mathcal{L}_{output} = \mathrm{sim}(Y_{text}, Y_{time}). \quad (5)$$

Parameter Efficient Training

To avoid catastrophic forgetting and improve training efficiency, we employ the parameter-efficient training technique to fine-tune the pre-trained LLMs. Specifically, for the temporal target branch, we introduce Low-rank Adaptation (LoRA) (Hu et al. 2021) and fine-tune the positional encoding weights. The total loss during training is the weighted summation of the supervised loss $\mathcal{L}_{sup}$, the feature regularization loss $\mathcal{L}_{feature}$, and the output consistency loss $\mathcal{L}_{output}$:

$$\mathcal{L}_{total} = \mathcal{L}_{sup} + \lambda_1 \mathcal{L}_{feature} + \lambda_2 \mathcal{L}_{output}, \quad (6)$$

where λ_1 and λ_2 are hyper-parameters. In the inference stage, only the output of the temporal target branch will serve as the model output.

Experiments

To demonstrate the effectiveness of the CALF, we conduct extensive experiments on various time series forecasting tasks, including long/short-term forecasting and few/zero-shot learning. Besides, we validate the model with low complexity, highlighting its efficiency in practical applications.

Baselines. We carefully select representative baselines from the recent time series forecasting landscape, including the following categories: (1) LLMs-based models: TimeLLM (Jin et al. 2024) and GPT4TS (Zhou et al. 2023); (2) Transformer-based models: PatchTST (Nie et al.

Models	CALF		TimeLLM†		GPT4TS†		PatchTST		iTransformer		Crossformer		FEDformer		TimesNet		MICN		DLinear		TiDE	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	0.395	**0.390**	0.410	0.409	0.389	0.397	**0.381**	0.395	0.407	0.411	0.502	0.502	0.448	0.452	0.400	0.406	0.392	0.413	0.403	0.407	0.412	0.406
ETTm2	**0.281**	**0.321**	0.296	0.340	0.285	0.331	0.285	0.327	0.291	0.335	1.216	0.707	0.305	0.349	0.291	0.333	0.328	0.382	0.350	0.401	0.289	0.326
ETTh1	**0.432**	**0.428**	0.460	0.449	0.447	0.436	0.450	0.441	0.455	0.448	0.620	0.572	0.440	0.460	0.458	0.450	0.558	0.535	0.456	0.452	0.445	0.432
ETTh2	**0.349**	**0.382**	0.389	0.408	0.381	0.408	0.366	0.394	0.381	0.405	0.942	0.684	0.437	0.449	0.414	0.427	0.587	0.525	0.559	0.515	0.611	0.550
Weather	0.250	**0.274**	0.274	0.290	0.264	0.284	0.258	0.280	0.257	0.279	0.259	0.315	0.309	0.360	0.259	0.287	**0.242**	0.299	0.265	0.317	0.271	0.320
ECL	**0.175**	**0.265**	0.223	0.309	0.205	0.290	0.216	0.304	0.178	0.270	0.244	0.334	0.214	0.327	0.192	0.295	0.186	0.294	0.212	0.300	0.251	0.344
Traffic	0.439	**0.281**	0.541	0.358	0.488	0.317	0.555	0.361	**0.428**	0.282	0.550	0.304	0.610	0.376	0.620	0.336	0.541	0.315	0.625	0.383	0.760	0.473

† We utilize their official codebase with the same experimental setup as ours, including input length and a GPT2 model with 6 layers, to ensure the fairness of the results. Other results are obtained from (Liu et al. 2024).

Table 1: Multivariate long-term forecasting results. The input sequence length T is set to 96 for all baselines. The best and second best results are in **bold** and underlined.

Models		CALF	TimeLLM	GPT4TS	PatchTST	ETSformer	FEDformer	Autoformer	TimesNet	TCN	N-HiTS	N-BEATS	DLinear
Yearly	SMAPE	**13.351**	13.419	13.531	13.477	18.009	13.728	13.974	13.387	14.920	13.418	13.436	16.965
	MASE	3.003	3.005	3.015	3.019	4.487	3.048	3.134	**2.996**	3.364	3.045	3.043	4.283
	OWA	**0.786**	0.789	0.793	0.792	1.115	0.803	0.822	**0.786**	0.880	0.793	0.794	1.058
Quarterly	SMAPE	**9.990**	10.110	10.177	10.380	13.376	10.792	11.338	10.100	11.122	10.202	10.124	12.145
	MASE	**1.164**	1.178	1.194	1.233	1.906	1.283	1.365	1.182	1.360	1.194	1.169	1.520
	OWA	**0.878**	0.889	0.898	0.921	1.302	0.958	1.012	0.890	1.001	0.899	0.886	1.106
Monthly	SMAPE	**12.643**	12.980	12.894	12.959	14.588	14.260	13.958	12.679	15.626	12.791	12.677	13.514
	MASE	**0.922**	0.963	0.956	0.970	1.368	1.102	1.103	0.933	1.274	0.969	0.937	1.037
	OWA	**0.872**	0.903	0.897	0.905	1.149	1.012	1.002	0.878	1.141	0.899	0.880	0.956
Others	SMAPE	**4.552**	4.795	4.940	4.952	7.267	4.954	5.485	4.891	7.186	5.061	4.925	6.709
	MASE	**3.092**	3.178	3.228	3.347	5.240	3.264	3.865	3.302	4.677	3.216	3.391	4.953
	OWA	**0.967**	1.006	1.029	1.049	1.591	1.036	1.187	1.035	1.494	1.040	1.053	1.487

Table 2: Short-term forecasting results on M4 dataset. The input length and prediction length are set to $[12, 96]$ and $[6, 48]$, respectively.

2023), iTransformer (Liu et al. 2024), Crossformer (Zhang and Yan 2023), ETSformer (Woo et al. 2022), FEDformer (Zhou et al. 2022) and Autoformer (Wu et al. 2021); (3) CNN-based models: TCN (Bai, Kolter, and Koltun 2018), MICN (Wang et al. 2022) and TimesNet (Wu et al. 2023); (4) MLP-based models: DLinear (Zeng et al. 2023) and TiDE (Das et al. 2023). Besides, N-HiTS (Challu et al. 2022) and N-BEATS (Oreshkin et al. 2019) are included for short-term forecasting.

Implementation Details. Following (Zhou et al. 2023), we use pre-trained GPT2 based model (Radford et al. 2019) with the first 6 Transformer layers as our backbone. Optimization is conducted using the Adam optimizer (Kingma and Ba 2014), with a learning rate of 0.0005. For the total loss function, we set the hyper-parameters $\gamma = 0.8$, $\lambda_1 = 1$ and $\lambda_2 = 0.01$. In terms of loss functions for long-term forecasting, we apply L1 loss across all three loss types for ETT datasets, while for the other three datasets, smooth L1 loss is utilized. For short-term forecasting, we compute supervised loss with SMAPE, modal consistency loss with MASE, and feature regularization loss with smooth L1 loss, respectively.

Long-term Forecasting

Setups. We conduct experiments on seven widely-used real-world datasets, including the Electricity Transformer Temperature (ETT) dataset with its four subsets (ETTh1, ETTh2, ETTm1, ETTm2), Weather, ECL, and Traffic (Wu et al. 2021). The input time series length T is fixed as 96 for a fair comparison, and we adopt four distinct prediction horizons $H \in \{96, 192, 336, 720\}$. Consistent with prior works (Li et al. 2024), the Mean Square Error (MSE) and Mean Absolute Error (MAE) are chosen as evaluation metrics. Following TBF's (Qiu et al. 2024a) setting, we do not use the drop last trick to ensure fair comparison.

Results. Comprehensive long-term forecasting results are presented in Tab. 1, where all the results are averaged from different prediction lengths. Notably, our approach reduces MSE/MAE by 7.05%/6.53% compared to the state-of-the-art Transformer-based model PatchTST. In comparison with the LLM-powered method TimeLLM, we observe a reduction of 5.98%/5.34% in MSE/MAE. Moreover, our improvements are substantial against other baseline methods, exceeding 10% in most cases.

Short-term Forecasting

Setups. We adopt the M4 datasets (Makridakis, Spiliotis, and Assimakopoulos 2018), which comprise univariate marketing data collected yearly, quarterly, and monthly. In this case, the prediction horizons are comparatively short, ranging in $[6, 48]$. Correspondingly, the input lengths are set to

Models	ETTm1		ETTm2		ETTh1		ETTh2	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TiDE	0.515	0.469	0.303	0.337	0.779	0.604	0.421	0.428
DLinear	0.567	0.499	0.329	0.382	0.647	0.552	0.441	0.458
MICN	0.970	0.674	1.073	0.716	1.405	0.814	2.533	1.158
TimesNet	0.673	0.534	0.321	0.354	0.865	0.625	0.476	0.463
FEDformer	0.696	0.572	0.356	0.392	0.750	0.607	0.553	0.525
Crossformer	1.340	0.848	1.985	1.048	1.744	0.914	3.139	1.378
PatchTST	0.557	0.483	**0.295**	0.334	0.683	0.546	0.550	0.487
GPT4TS	0.608	0.500	0.303	0.336	0.689	0.555	0.579	0.497
TimeLLM	0.636	0.512	0.348	0.343	0.765	0.584	0.589	0.498
CALF	**0.504**	**0.462**	0.302	**0.330**	**0.644**	**0.541**	**0.419**	**0.427**

Table 3: Few-shot learning results on 10% training data of ETT datasets.

be twice the size of the prediction horizons. The evaluation metrics are symmetric mean absolute percentage error (SMAPE), mean absolute scaled error (MSAE), and overall weighted average (OWA).

Results. As shown in Tab. 2, our method demonstrates superior performance in short-term forecasting across various evaluation metrics. In comparison with TimesNet, currently the leading method in short-term forecasting, our model achieves a 1% overall improvement in performance.

Few/zero-shot Learning

LLMs have demonstrated remarkable performance in both few-shot and zero-shot tasks. The capabilities of few-shot and zero-shot learning are critically important for general time series forecasting models (Brown et al. 2020; Liu et al. 2023b; Kojima et al. 2022). To thoroughly assess the generalized ability of our method in time series forecasting, we conduct experiments under few-shot and zero-shot learning settings. In few-shot learning, only a small ratio of the training data is utilized. For zero-shot learning, the model trained on one dataset is directly employed for testing on another dataset without any additional training. All the results are averaged from 4 different prediction lengths $H \in \{96, 192, 336, 720\}$.

Few-shot Learning. We conduct few-shot experiments on four ETT datasets. Specifically, for each dataset, we utilize only the first 10% of the training data. This constrained data scenario presents a considerable challenge, testing the ability of the model to learn effectively with limited information. Tab. 3 demonstrates that our method outperforms other baselines, highlighting its robustness in the few-shot setting. Compared with TimeLLM and PatchTST, our method achieves an average reduction of 8% and 9%, respectively.

Zero-shot Learning. Going beyond few-shot scenarios, we further delve into zero-shot learning, where LLMs demonstrate their prowess as adept and intuitive reasoners. In this setting, models trained on one dataset ♦ are evaluated on an entirely different dataset ★, without any further training. As shown in Tab. 4, our method stands out for its exceptional performance, surpassing TimeLLM and PatchTST by 4% and 9% respectively. This indicates that our approach significantly enhances the model's capability for effective learning transfer across different domains.

Models	h1 → m1		h1 → m2		h2 → m1		h2 → m2	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TiDE	0.774	**0.574**	**0.314**	**0.355**	0.841	0.590	0.321	0.364
DLinear	0.760	0.577	0.399	0.439	0.778	0.594	0.496	0.496
MICN	1.439	0.780	2.428	1.236	0.764	0.601	0.527	0.519
TimesNet	0.794	0.575	0.339	0.370	1.286	0.705	0.361	0.390
FEDformer	0.765	0.588	0.357	0.403	**0.741**	0.588	0.365	0.405
Crossformer	0.999	0.736	1.120	0.789	1.195	0.711	2.043	1.124
PatchTST	0.894	0.610	0.318	0.362	0.871	0.596	0.420	0.433
GPT4TS	0.798	**0.574**	0.317	0.359	0.920	0.610	0.331	0.371
TimeLLM	0.847	0.565	0.315	0.357	0.868	0.595	0.322	0.363
CALF	**0.755**	**0.574**	0.316	**0.355**	0.836	**0.586**	**0.319**	**0.360**

Table 4: Zero-shot learning results on ETT datasets. "♦ → ★" indicates that models trained on the dataset ♦ are evaluated on a distinct dataset ★.

Dataset	GPT4TS		Time-LLM		CALF	
	Time (s)	MSE / MAE	Time (s)	MSE / MAE	Time (s)	MSE / MAE
ETTm1	626	0.329 / 0.364	1476	0.359 / 0.381	**135**	**0.323 / 0.349**
ECL	8274	0.185 / 0.272	33209	0.204 / 0.293	**251**	**0.145 / 0.238**
Traffic	15067	0.468 / 0.307	62412	0.536 / 0.359	**614**	**0.407 / 0.268**
Weather	596	0.182 / 0.223	1262	0.195 / 0.233	**123**	**0.164 / 0.204**

Table 5: Comparison of different LLM-based time series forecasting methods in terms of computation time and performance (MSE/MAE) across various datasets.

Efficiency Analysis

We conduct experiments on four datasets: ETTm1, ECL, Traffic, and Weather. The input and prediction lengths are both set to 96. As shown in Tab. 5, our proposed CALF shows significant improvements in both efficiency and accuracy compared with other LLM-based methods, largely due to treating each channel sequence as a token, employing an efficient fine-tuning strategy, and requiring only a single time branch during inference.

Ablation Study

Ablation on Different Loss Functions. The feature regularization loss $\mathcal{L}_{feature}$ aligns the intermediate features between the textual source branch and the temporal target branch, while the output consistency loss $\mathcal{L}_{output}$ ensures output coherence across modalities. The supervised loss $\mathcal{L}_{sup}$ directly guides learning with ground truth data. We analyze the specific effects of each proposed loss function as detailed in Tab. 6. Employing only the supervised loss resulted in MSE/MAE of 0.446/0.438 on ETTh1 and 0.263/0.286 on Weather, respectively. The addition of feature regularization loss $\mathcal{L}_{feature}$ or output consistency loss $\mathcal{L}_{output}$ led to incremental improvements, with the best performance observed when all three losses were combined, achieving the lowest MSE and MAE on both datasets.

Ablation on the Number of Principal Components. We employ PCA to conduct dimensional reduction on the original word embeddings for efficient training. Despite the reduced cost, however, PCA may inevitably lead to information loss. In this section, we ablate the number of principal components d to present the effects. The experimental results are given in Fig. 4. It can be seen that the performance

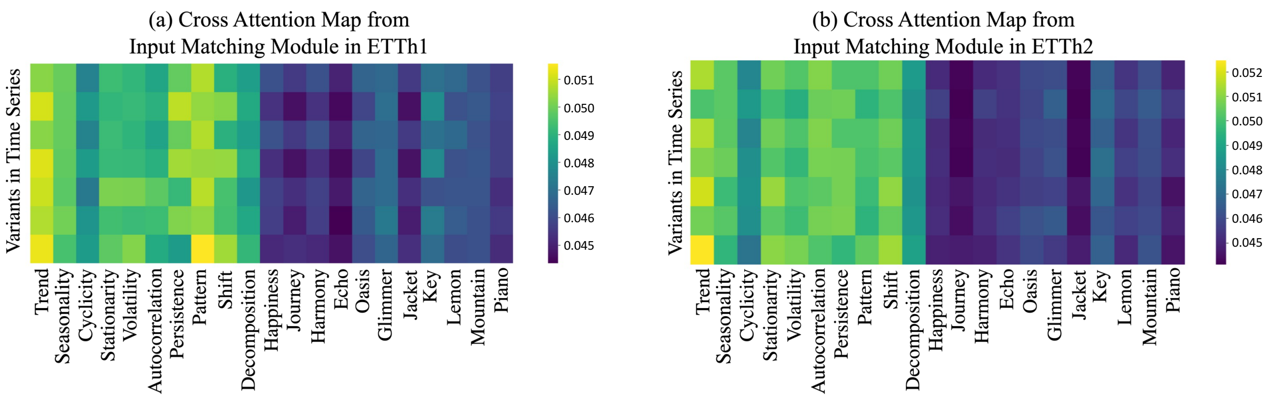

Figure 3: Cross-attention maps from the Cross-Modal Match Module for ETTh1 (left) and ETTh2 (right). Each row represents a time series instance, while columns correspond to selected words, including both time-related terms (*e.g.*, trend, seasonality) and general terms (*e.g.*, echo, key). Each cell indicates the relevance of the respective channel to the selected word.

$\mathcal{L}_{feature}$	$\mathcal{L}_{output}$	$\mathcal{L}_{sup}$	ETTh1 MSE	ETTh1 MAE	Weather MSE	Weather MAE
–	–	✓	0.446	0.438	0.263	0.286
✓	–	✓	0.434	0.431	0.254	0.276
–	✓	✓	0.438	**0.426**	0.258	0.283
✓	✓	✓	**0.432**	0.428	**0.250**	**0.274**

Table 6: Ablation on different loss functions on ETTh1 and Weather datasets.

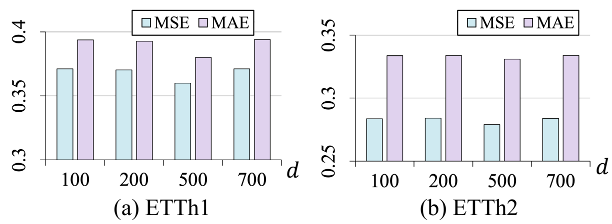

Figure 4: Ablation on different low dimension d of PCA on (a) ETTh1 and (b) ETTh2 datasets.

is not that sensitive to different numbers of principal components. In addition, a smaller d causes performance degradation due to the missing key information, while a larger d causes information redundancy which causes learning difficulty. In practice, we chose $d = 500$, which can attain an explainable variance ratio of 88% while achieving satisfactory performance.

Discussion

Difference from Other Work. Existing LLM-based methods, such as TimeLLM (Jin et al. 2024) and TEST (Sun et al. 2024), also consider the alignment between the time-modal input and the text-modal parameters by employing the cross-attention layer or contrastive learning at the input side. However, we point out that this simple scheme is not sufficiently powerful thus leading to partial alignment (Fig. 1) as well as weak generalization performance (Tab. 3, Tab. 4). By contrast, we identify the alignment problem and further propose a multi-level cross-modal fine-tuning framework to achieve finer-grained alignment. Moreover, although the previous approaches adopt one cross-attention layer as an intuitive solution, it can raise significant computational costs due to the huge alphabet size of LLM. In this work, we introduce the offline PCA to generate synonym clusters from the huge alphabet as an alternative, allowing high efficiency while achieving better performance.

Interpretability on Implicit Input Alignment. To narrow the temporal-textual modality gap, we perform cross-attention on word embedding weights to generate aligned text tokens instead of intuitive natural language. As shown in Fig. 3, we visualize the cross-attention maps from the Cross-Modal Match Module for the ETTh1 and ETTh2 datasets. Each row in the maps represents a time series instance, while columns correspond to selected words, including both time-related terms (*e.g.*, trend, seasonality) and general terms (*e.g.*, echo, key). Each cell indicates the relevance of the respective channel to the selected word. Our analysis reveals that the Cross-Modal Match Module effectively aligns time series tokens with word embeddings that describe temporal characteristics. The attention distributions show that time series data align well with relevant textual descriptions, indicating that our module successfully bridges the gap between temporal and textual modalities.

Conclusion

In this work, we propose CALF, a novel cross-modal fine-tuning framework that leverages the robust capabilities of Large Language Models (LLMs) for time series forecasting. CALF effectively bridges the distribution discrepancy between temporal data and the textual nature of LLMs through the Cross-Modal Match Module, Feature Regularization Loss, and Output Consistency Loss. Extensive experiments across several real-world datasets validate that CALF sets a new benchmark in both long- and short-term forecasting, demonstrating strong generalization and low computational complexity.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China, under Grant (62302309, 62171248), Shenzhen Science and Technology Program (JCYJ20220818101014030, JCYJ20220818101012025), and the PCNL KEY project (PCL2023AS6-1).

References

Angryk, R. A.; Martens, P. C.; Aydin, B.; Kempton, D.; Mahajan, S. S.; Basodi, S.; Ahmadzadeh, A.; Cai, X.; Filali Boubrahimi, S.; Hamdi, S. M.; et al. 2020. Multivariate time series dataset for space weather data analytics. *Scientific data*, 7(1): 227.

Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901.

Cao, D.; Jia, F.; Arik, S. O.; Pfister, T.; Zheng, Y.; Ye, W.; and Liu, Y. 2024. TEMPO: Prompt-based Generative Pretrained Transformer for Time Series Forecasting. *International Conference on Learning Representations*.

Challu, C.; Olivares, K. G.; Oreshkin, B. N.; Garza, F.; Mergenthaler, M.; and Dubrawski, A. 2022. N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting. *arXiv preprint arXiv:2201.12886*.

Chang, C.; Peng, W.-C.; and Chen, T.-F. 2023. LLM4TS: Two-stage fine-tuning for time-series forecasting with pretrained llms. *arXiv preprint arXiv:2308.08469*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 1597–1607. PMLR.

Dai, T.; Wu, B.; Liu, P.; Li, N.; Bao, J.; Jiang, Y.; and Xia, S.-T. 2024. Periodicity Decoupling Framework for Long-term Series Forecasting. In *International Conference on Learning Representations*.

Das, A.; Kong, W.; Leach, A.; Sen, R.; and Yu, R. 2023. Long-term Forecasting with TiDE: Time-series Dense Encoder. *arXiv preprint arXiv:2304.08424*.

Demirel, Ö. F.; Zaim, S.; Çalişkan, A.; and Özuyar, P. 2012. Forecasting natural gas consumption in Istanbul using neural networks and multivariate time series methods. *Turkish Journal of Electrical Engineering and Computer Sciences*, 20(5): 695–711.

Hassid, M.; Remez, T.; Nguyen, T. A.; Gat, I.; Conneau, A.; Kreuk, F.; Copet, J.; Defossez, A.; Synnaeve, G.; Dupoux, E.; et al. 2024. Textually pretrained speech language models. *Advances in Neural Information Processing Systems*, 36.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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

Jin, Y.; Hu, G.; Chen, H.; Miao, D.; Hu, L.; and Zhao, C. 2023. Cross-modal distillation for speaker recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 12977–12985.

Kiela, D.; Bhooshan, S.; Firooz, H.; Perez, E.; and Testuggine, D. 2019. Supervised multimodal bitransformers for classifying images and text. *arXiv preprint arXiv:1909.02950*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35: 22199–22213.

Li, Z.; Qiu, X.; Chen, P.; Wang, Y.; Cheng, H.; Shu, Y.; Hu, J.; Guo, C.; Zhou, A.; Wen, Q.; et al. 2024. FoundTS: Comprehensive and Unified Benchmarking of Foundation Models for Time Series Forecasting. *arXiv preprint arXiv:2410.11802*.

Liu, P.; Wu, B.; Li, N.; Dai, T.; Lei, F.; Bao, J.; Jiang, Y.; and Xia, S.-T. 2023a. WFTNet: Exploiting Global and Local Periodicity in Long-term Time Series Forecasting. *IEEE International Conference on Acoustics, Speech and Signal Processing*.

Liu, X.; McDuff, D.; Kovacs, G.; Galatzer-Levy, I.; Sunshine, J.; Zhan, J.; Poh, M.-Z.; Liao, S.; Di Achille, P.; and Patel, S. 2023b. Large Language Models are Few-Shot Health Learners. *arXiv preprint arXiv:2305.15525*.

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

Makridakis, S.; Spiliotis, E.; and Assimakopoulos, V. 2018. The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4): 802–808.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *International Conference on Learning Representations*.

Patton, A. 2013. Copula methods for forecasting multivariate time series. *Handbook of economic forecasting*, 2: 899–960.

Qiu, X.; Hu, J.; Zhou, L.; Wu, X.; Du, J.; Zhang, B.; Guo, C.; Zhou, A.; Jensen, C. S.; Sheng, Z.; et al. 2024a. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *arXiv preprint arXiv:2403.20150*.

Qiu, X.; Wu, X.; Lin, Y.; Guo, C.; Hu, J.; and Yang, B. 2024b. DUET: Dual Clustering Enhanced Multivariate Time Series Forecasting. *arXiv preprint arXiv:2412.10859*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners.

Shen, J.; Li, L.; Dery, L. M.; Staten, C.; Khodak, M.; Neubig, G.; and Talwalkar, A. 2023. Cross-modal fine-tuning: Align then refine. In *International Conference on Machine Learning*, 31030–31056. PMLR.

Sun, C.; Li, H.; Li, Y.; and Hong, S. 2024. TEST: Text Prototype Aligned Embedding to Activate LLM's Ability for Time Series. In *The International Conference on Learning Representations*.

Verma, G.; Choi, M.; Sharma, K.; Watson-Daniels, J.; Oh, S.; and Kumar, S. 2024. Mysterious Projections: Multimodal LLMs Gain Domain-Specific Visual Capabilities Without Richer Cross-Modal Projections. *arXiv preprint arXiv:2402.16832*.

Vinod, R.; Chen, P.-Y.; and Das, P. 2023. Reprogramming pretrained language models for protein sequence representation learning. *arXiv preprint arXiv:2301.02120*.

Wang, H.; Peng, J.; Huang, F.; Wang, J.; Chen, J.; and Xiao, Y. 2022. MICN: Multi-scale local and global context modeling for long-term series forecasting. In *International Conference on Learning Representations*.

Wen, Q.; Zhou, T.; Zhang, C.; Chen, W.; Ma, Z.; Yan, J.; and Sun, L. 2022. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*.

Woo, G.; Liu, C.; Sahoo, D.; Kumar, A.; and Hoi, S. C. H. 2022. ETSformer: Exponential Smoothing Transformers for Time-series Forecasting. *arXiv preprint arXiv:2202.01381*.

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

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

Xiao, Y.; Qiu, J.; Li, Z.; Hsieh, C.-Y.; and Tang, J. 2021. Modeling protein using large-scale pretrain language model. *arXiv preprint arXiv:2108.07435*.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 27268–27286. PMLR.

Zhou, T.; Niu, P.; Wang, X.; Sun, L.; and Jin, R. 2023. One Fits All: Power General Time Series Analysis by Pretrained LM. *Advances in Neural Information Processing Systems*, 36.