S^{2}IP-LLM: Semantic Space Informed Prompt Learning with LLM for Time Series Forecasting

Zijie Pan [1] **Yushan Jiang** [1] **Sahil Garg** [2] **Anderson Schneider** [2] **Yuriy Nevmyvaka** [2] **Dongjin Song** [1]

Abstract

Recently, there has been a growing interest in leveraging pre-trained large language models (LLMs) for various time series applications. However, the semantic space of LLMs, established through the pre-training, is still underexplored and may help yield more distinctive and informative representations to facilitate time series forecasting. To this end, we propose Semantic Space Informed Prompt learning with LLM (S^2IP-LLM) to *align the pre-trained semantic space with time series embedding space* and perform time series forecasting based on learned prompts from the joint space. We first design a tokenization module tailored for cross-modality alignment, which explicitly concatenates patches of decomposed time series components to create embeddings that effectively encode the temporal dynamics. Next, we leverage the pre-trained word token embeddings to derive semantic anchors and align selected anchors with time series embeddings by maximizing the cosine similarity in the joint space. This way, S^2IP-LLM can retrieve relevant semantic anchors as prompts to provide strong indicators (context) for time series that exhibit different temporal dynamics. With thorough empirical studies on multiple benchmark datasets, we demonstrate that the proposed S^2IP-LLM can achieve superior forecasting performance over state-of-the-art baselines. Furthermore, our ablation studies and visualizations verify the necessity of prompt learning informed by semantic space.

[1]School of Computing, University of Connecticut, Storrs, USA [2]Department of Machine Learning Research, Morgan Stanley, New York, USA. Correspondence to: Yuriy Nevmyvaka <yuriy.nevmyvaka@ morganstanley.com>, Dongjin Song <dongjin.song@uconn.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

Over the past few years, pre-trained large language models (LLMs) such as GPT-4 (Achiam & et al., 2023) and LLaMA (Touvron et al., 2023b;c) not only achieved great success across a diverse range of natural language processing (NLP) tasks, *i.e.*, generate coherent and contextually relevant text, answer questions, and translate sentences between multiple languages, but also exhibited tremendous potential in tackling applications of more complex or structured domains, such as code generation, healthcare, finance, and autonomous systems, *etc* (Singhal et al., 2022; Cui et al., 2024; Li et al., 2023). As time series analysis is becoming increasingly important for strategic planning and operational efficiency in various real-world applications, *e.g.*, energy load management, traffic forecasting, weather forecasting, health risk analysis, *etc* (Friedman, 1962; Courty & Li, 1999; Böse et al., 2017; Gao et al., 2020; Li et al., 2022; Liu et al., 2023a; Dimri et al., 2020), a natural question to ask is *whether we should train a general purpose foundation model from scratch, or fine-tune pre-trained LLMs to perform time series forecasting?*

Recently, significant efforts have been made to build foundation models for general-purpose time series analysis (Wu et al., 2023; Garza & Mergenthaler-Canseco, 2023; Rasul et al., 2023). TimesNet (Wu et al., 2023) uses TimesBlock as a task-general backbone to capture multi-periodicity and extract complex intraperiod- and interperiod-variations via transformed 2D tensors. TimeGPT-1 describes a general pre-trained model for time series forecasting (Garza & Mergenthaler-Canseco, 2023). These approaches, however, are hindered by two main challenges. First, time series data can be acquired in various formats, such as univariate or multivariate, often in large volumes, and from different domains, like healthcare, finance, traffic, environmental sciences, *etc*. This escalates the complexity of model training and poses challenges in handling different scenarios. Second, time series data, in practice, often exhibit non-stationary characteristics, resulting in the underlying statistical properties, such as means, variances, and auto-correlations shifting during collection. This could also result in concept drift, where the statistical properties of target variables change over time. These realities present significant challenges for

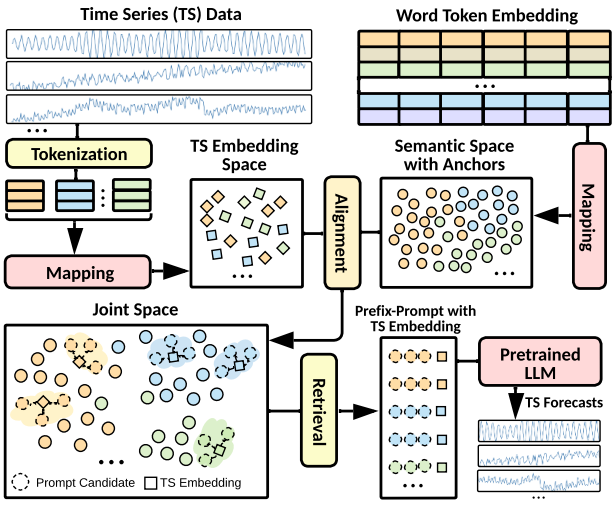

Figure 1. The demonstration of semantic space informed prompting in S^2IP-LLM. The input time series is decomposed and mapped to obtain time series (TS) embedding. Next, the TS embedding is aligned with semantic anchors derived from the pre-trained word token embedding. Finally, top-K similar semantic anchors are retrieved and used as prefix-prompts with TS embedding.

large models to be adapted and retrained effectively.

On the other hand, LLMs trained on extensive and diverse text corpora can serve as a foundational knowledge base that can be applied to a variety of downstream tasks with minimal task-specific prompt learning or fine-tuning. Inspired by this, there has been a growing interest in leveraging existing LLMs to facilitate time series analysis. For instance, Tian Zhou & Jin (2023) utilizes a frozen pre-trained language model to attain state-of-the-art or equivalent performance. Jin et al. (2024) develop time-LLM to reprogram the input time series via text prototype representations by incorporating the embeddings of the dataset's text descriptions as context information. In real-world applications, however, dataset description information may not always be available or informative. In addition, the patching operation (*i.e.*, tokenization), which splits a long time series sequence into overlapping segments over instance normalized time series input, may have limited expressibility as it could fail to capture the subtle variations of different components in time series.

In this paper, we argue that the semantic space in the form of word token embeddings (based on pre-trained LLMs) can already offer a more distinctive and informative representation space (Ethayarajh, 2019) to help align time series embeddings. Based on this, we develop Semantic Space Informed Prompt with LLM (S^2IP-LLM) for time series forecasting. Specifically, as shown in Figure 1, we first design a tokenization module tailored to semantic space alignment, which explicitly concatenates patches of decomposed time series components (*i.e.*, trend, seasonality, and residual) to create an embedding that effectively encodes the temporal dynamics more expressively. Next, we map the pre-trained word embeddings to obtain semantic anchors and align selected anchors with time series embeddings by maximizing the cosine similarity in the joint space. In this way, S^2IP-LLM can retrieve relevant semantic anchors as prefix-prompts to provide strong indicators (context) for time series embeddings that exhibit different temporal dynamics. Our experiments over several standard benchmark datasets demonstrate that S^2IP-LLM can achieve superior forecasting performance over state-of-the-art baselines. Moreover, our ablation studies and visualizations also verify the necessity of prompt learning in the joint space.

To summarize, our contributions include:

- We design a specialized tokenization module that concatenates patches of decomposed time series components to provide more expressive local contexts and facilitate semantic space informed prompting.
- We leverage semantic anchors derived from pre-trained word token embeddings (semantic space) to align time series embeddings and learn a distinctive and informative joint space. Moreover, aligned semantic anchors are used as prompt indicators (contexts) to enhance the representation of time series.
- Our experiments and analysis on multiple benchmark datasets demonstrate the superiority of S^2IP-LLM over state of the art and the necessity of prompt learning informed by semantic space.

2. Related Work

2.1. Time Series Forecasting

In recent years, a variety of statistical and machine learning methods have been developed for time series analysis, *e.g.*, ARIMA (Anderson & Kendall, 1976), Prophet (Taylor & Letham, 2018), *etc*. More recently, different types of deep neural networks have been applied for time series analysis. For instance, recurrent neural network (RNN) based models have been developed to capture auto-regressive temporal dynamics (Qin et al., 2017; Li et al., 2017; Lai et al., 2018; Gu et al., 2021). Graph neural networks (GNN) based methods are leveraged to capture variable dependencies among different time series (Cao et al., 2020; Wu et al., 2020; Shang et al., 2021; Pan et al., 2024). Transformer based models leverage the self-attention mechanisms tailored for time series to better capture the temporal dynamics, variable dependencies, or both (Woo et al., 2022; Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022; Liu et al., 2023b). More recently, MLP-based models (Challu et al., 2023; Zeng et al., 2023) and

convolution-based models (Wu et al., 2023) have achieved state-of-the-art performance on par with Transformers, but with much simpler designs. Nevertheless, while these deep forecasters perform well on specific datasets, they lack the flexibility and generalizability to adapt to real-world time series data from different domains.

2.2. Pre-trained Large Model for Time Series Analysis

Recent advancements in natural language processing (NLP) and computer vision (CV) demonstrate that pre-trained models can effectively adapt to a range of downstream tasks through fine-tuning (Bao et al., 2021; He et al., 2022; Brown et al., 2020; Devlin et al., 2018). Inspired by this, several different pre-trained models have been developed for time series based on either supervised (Fawaz et al., 2018) or self-supervised learning (Zhang et al., 2022b; Deldari et al., 2022). During the training stage, models can learn robust representations from a variety of input time series data. Then, these models can be fine-tuned for downstream tasks of similar domains to further enhance their performance (Tang et al., 2022). With the emergence and success of Large Language Models (LLMs), including T5 (Raffel et al., 2020), GPT-based models (Radford et al., 2018; 2019; Brown et al., 2020; Ouyang et al., 2022), and LLaMA (Touvron et al., 2023a), which have showcase their robust pattern recognition and reasoning abilities over complex sequences of tokens, there is a trend to explore how to effectively transfer knowledge from these powerful pre-trained LLM models to time series domain (Jiang et al., 2024). One line of research focuses on leveraging the pre-trained LLMs as zero-shot learners. For instance, Xue & Salim (2022) and Nate Gruver & Wilson (2023) directly convert time series data to corresponding text sequence inputs and achieve encouraging results for time series forecasting. Another line of research (Tian Zhou & Jin, 2023; Chang et al., 2023) involves tokenizing the input time series data into overlapping patches and strategically leveraging or fine-tuning LLMs for time series analysis. Following this paradigm, TEST (Sun et al., 2023) and Time-LLM (Jin et al., 2024) reprogram time series data with text prototype embedding and incorporate textual prompts for time series analysis. TEMPO (Cao et al., 2023) incorporates the decomposition of time series and retrieval-based prompt design for non-stationary time series data. Different from those methods, we explicitly leverage semantic anchors derived from pre-trained word token embeddings (semantic space) to align time series embeddings and develop a simple yet effective prompt mechanism to inform LLM for forecasting tasks.

3. Methodology

Overview: S^2IP-LLM consists of three key components as shown in Figure 2. Given the input time series, we first tokenize it and obtain the time series (TS) embedding based on time series decomposition and patching. Next, we will align the TS embedding with semantic anchors derived from the pre-trained word token embedding. Finally, top-K similar semantic anchors will be retrieved to serve as prefix-prompts for the TS embedding and the concatenated vector will be leveraged as the query for pre-trained LLMs.

In this paper, GPT-2 is used as the backbone. During the training stage, we not only learn the mapping functions of input and output but also fine-tune the positional embedding and layer norm block of GPT-2.

3.1. Problem Statement

We first formalize the time series forecasting problem. Let $X \in \mathbb{R}^{N \times T}$ denote the time series data containing N variables and T time steps, where $X_{:,t} \in \mathbb{R}^{N \times 1}$ denotes t-th time step across all variables and $X_{i,:} \in \mathbb{R}^{1 \times T}$ denotes i-th variable. Given a historical τ-step window of time series, we aim to learn a forecasting module $\mathcal{F}(\cdot)$ that will predict the next τ' time steps based on the input window. Mathematically, at a starting time step t, the corresponding forecast is given by $\hat{Y} = \hat{X}_{:,t:t+\tau'-1} = \mathcal{F}(X_{:,t-\tau:t-1})$.

3.2. Time Series Tokenization

In real-world applications, non-stationary data is prevalent. To tackle this problem, we first apply the reversible instance normalization (Kim et al., 2021) on time series input such that the data has zero mean and unit standard deviation to mitigate the distribution shift in time series. Specifically, given the i-th time series input at time step t, *i.e.*, $X_{i,t}$, the transformed value $X'_{i,t}$ can be given by:

$$X'_{i,t} = \gamma_T \left(X_{i,t} - \frac{\mathbb{E}_t \left[X_{i,t} \right]}{\sqrt{\mathrm{Var} \left[X_{i,t} \right] + \epsilon_T}} \right) + \beta_T \quad (1)$$

where $\mathbb{E}_t \left[X_{i,t} \right]$ and $\mathrm{Var} \left[X_{i,t} \right]$ are the instance-specific mean and variance, respectively. γ_T and β_T are trainable parameters. Next, we adopt an additive seasonal-trend decomposition method to decompose normalized time series into long-term trend, seasonal, and residual components. The additive seasonal-trend decomposition is given by $X'_{i,t} = X^{\text{tre}\prime}_{i,t} + X^{\text{sea}\prime}_{i,t} + X^{\text{res}\prime}_{i,t}$, where *tre*, *sea*, *res* denotes the long-term trend, seasonal, and residual component, respectively. There are several options for additive seasonal-trend decomposition. One option is the classical additive seasonal-trend decomposition that first obtains long-term trend components using moving averages. Then, the seasonal component is estimated by averaging the detrended time series with pre-defined season parameters. Finally, the residual component is obtained by subtracting the estimated trend and seasonal components from the normalized time series. Another option is the Seasonal-Trend decomposition

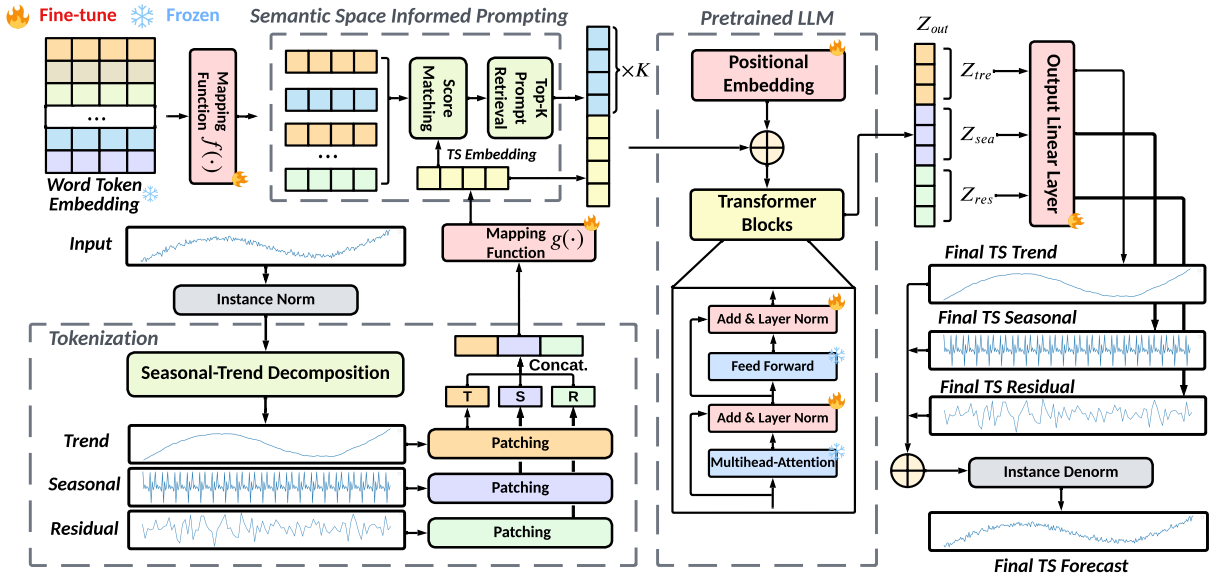

Figure 2. The model architecture of S^2IP-LLM. The input time series is normalized, decomposed, patched individually, and concatenated to represent the context of time series (TS). Semantic space informed prompting performs alignment between the contextual TS embeddings and the semantic anchors extracted from pre-trained word embeddings, and retrieves the most similar K ones as prefix-prompts. The decomposed TS representations from pre-trained LLM are linearly projected and combined as the TS forecast.

using Loess (STL) (Cleveland et al., 1990). The choice of decomposition method will based on validation results.

Next, we follow (Nie et al., 2023) to encode temporal information and local contexts of input time series by aggregating consecutive time steps into overlapped patched tokens. Take the trend component as an example, the normalized component series, $X^{\text{tre}\,\prime}_{i,t-\tau:t-1} \in \mathbb{R}^{1\times\tau}$ is converted to patched token representation $P^{\text{tre}}_{i,t-\tau:t-1} \in \mathbb{R}^{N_P\times L_P}$, in which L_P is the patch length, $N_P = \left\lfloor \frac{(\tau-L_P)}{S} \right\rfloor + 2$ is the number of patches and S is the horizontal sliding stride. We apply patching to each variable component over the temporal dimension and then concatenate the tokens of these components into a single meta-token, $P_{i,t-\tau:t-1} = \left[P^{\text{tre}}_{i,t-\tau:t-1}, P^{\text{sea}}_{i,t-\tau:t-1}, P^{\text{res}}_{i,t-\tau:t-1}\right] \in \mathbb{R}^{N_P\times 3L_P}$. We then feed the meta-token into a projection layer $g(\cdot)$ to get the time series embedding $\mathcal{P}_{i,t-\tau:t-1} = g\left(P_{i,t-\tau:t-1}\right) \in \mathbb{R}^{N_P\times D}$, where D is the embedding size for the pre-trained LLMs.

3.3. Semantic Space Informed Prompting

Prompting has emerged as an effective technique in various applications, enabling LLMs to utilize task-specific information to achieve enhanced reasoning capabilities (Yin et al., 2023). Existing works primarily focus on employing template-based and fixed prompts for pre-trained LLMs in time series analysis (Xue & Salim, 2022; Jin et al., 2024). While these methods are intuitive, straightforward, and yield satisfactory results, their rigid prompt contexts are in line with linguistic semantics. However, time series representation inherently lacks human semantics and is more closely tied to sequence patterns in the form of temporal dynamics. Conversely, Lester et al. (2021) demonstrate the effectiveness of soft prompts in enabling LLMs to comprehend inputs more effectively. In the realm of time series analysis with LLMs, recent works (Sun et al., 2023; Cao et al., 2023) start to consider soft prompts as task-specific, randomly initialized, trainable vectors that learn from the supervised loss between LLM's output and the ground truth. However, the semantic space of LLMs, established through the pre-training, is still underexplored and may help yield more distinctive and informative representations for time series data. Based on this intuition, we introduce a prompting mechanism informed by the pre-trained semantic space. Specifically, the pre-trained semantic word token embeddings, represented as $\mathbf{E} \in \mathbb{R}^{V\times D}$ where V is the vocabulary size, are inevitably large and dense. For example, the vocabulary size of GPT-2 (Radford et al., 2019) reaches 50,257 and may raise computational deficiency. Instead of directly using the semantic word token embedding, we derive a small set of semantic anchors $\mathbf{E}'$ in the hidden space using a generic mapping function $f(\cdot)$ on $\mathbf{E}$, which is denoted as $\mathbf{E}' = f\left(\mathbf{E}\right) \in \mathbb{R}^{V'\times D}$, where V' is the reduced number of

semantic anchors and $V' \ll V$. To properly retrieve relevant semantic anchors to enhance the time series embedding $\mathcal{P}_{i,t-\tau:t-1}$, we align the semantic anchors and time series embedding based on a score-matching function $\gamma(\cdot)$. In this paper, we implement the score-matching function based on cosine similarity

$$\gamma\left(\mathcal{P}_{i,t-\tau:t-1}, \boldsymbol{e}'_m\right) = \frac{\mathcal{P}_{i,t-\tau:t-1} \cdot \boldsymbol{e}'_m}{\|\mathcal{P}_{i,t-\tau:t-1}\| \, \|\boldsymbol{e}'_m\|}, \tag{2}$$

where $\boldsymbol{e}'_m \in \mathbf{E}'$. We select top-$K$ relevant semantic anchors based on the similarity scores and utilize them as prefix-prompt to enhance the input time series embedding, *i.e.*,

$$\begin{aligned} \boldsymbol{Z}_{i,t-\tau:t-1} &= [\boldsymbol{e}'_1; \cdots; \boldsymbol{e}'_K; \mathcal{P}_{i,t-\tau:t-1}] \\ &= \left[\boldsymbol{e}'_{\text{top-}K}; \mathcal{P}_{i,t-\tau:t-1}\right] \end{aligned} \tag{3}$$

which will serve as the input for the pre-trained LLMs.

3.4. Optimization Objective

We can obtain the output embedding $\boldsymbol{Z}_{\text{out}}$ after the forward path of the prompt enhanced time series embedding through LLMs. We will flatten it and use a linear mapping to project the representation to the forecasting horizon $\boldsymbol{Y}_{\text{out}}$. The overall forecasting should also be the additive combination of the individual component predictions due to the decomposition step. We further split and express $\boldsymbol{Y}_{\text{out}}$ into a concatenation form $\boldsymbol{Y}_{\text{out}} = [Y^{\text{tre}}_{\text{out}}, Y^{\text{sea}}_{\text{out}}, Y^{\text{res}}_{\text{out}}]$ and obtain the forecasting results as $\hat{Y} = Y^{\text{tre}}_{\text{out}} + Y^{\text{sea}}_{\text{out}} + Y^{\text{res}}_{\text{out}}$. At every training iteration, the overall training objective is:

$$\min \mathcal{L}\left(\hat{Y}, X_{:,t:t+\tau'-1}\right) - \lambda \sum \gamma\left(\mathcal{P}_{i,t-\tau:t-1}, \boldsymbol{e}'_{\text{top-}K}\right), \tag{4}$$

where the first term is the forecasting loss in the form of mean squared error (MSE), and the second term is a score-matching function to align selected semantic anchors with the time series embedding obtained via decomposition and patching. In this way, we could obtain a more informative space to facilitate the underlying forecasting task. $\lambda \geq 0$ is a hyper-parameter to trade-off the alignment.

3.5. Backbone and Fine-tuning Strategy

In this paper, we employ GPT-2 (Radford et al., 2019) as our pre-trained large language model (LLM) backbone. We choose to keep a significant portion of the parameters frozen, especially those parameters related to the multi-headed attention and the feed-forward networks within the Transformer blocks. This strategy can not only reduce the computational burden but also align with existing literature (Lu et al., 2022; Houlsby et al., 2019; Tian Zhou & Jin, 2023). They suggest that maintaining most of the parameters in their non-trainable state can achieve better outcomes compared to completely retraining LLMs. For GPT-2, we only fine-tune the positional embedding layer and the layer-normalization layers.

4. Experiments

In our experiments, we compare the proposed S^2IP-LLM against a variety of baselines on 11 public datasets. We validate the effectiveness of S^2IP-LLM over different time series tasks, including long-term forecasting (Section 4.1), short-term forecasting (Section 4.2), and few-shot forecasting (Section 4.3). We also provide the ablation studies and parameter sensitivity analysis in Section 4.4. Finally, we visualize the prompt enhanced time series embeddings to qualitatively assess the effectiveness of S^2IP-LLM. We follow the experimental configurations (Wu et al., 2023) for all baselines using the unified pipeline.[1]

Baselines. The baselines include a set of Transformer-based methods, *i.e.*, iTransformer (Liu et al., 2023b), PatchTST (Nie et al., 2023), FEDformer (Zhou et al., 2022), Autoformer (Wu et al., 2021), Non-Stationary Transformer (Liu et al., 2022), ETSformer (Woo et al., 2022) and Informer (Zhou et al., 2021). We also select a set of non-transformer based techniques, *i.e.*, DLinear (Zeng et al., 2023), TimesNet (Wu et al., 2023), and LightTS (Zhang et al., 2022a) for comparison. Finally, two approaches based on LLMs, *i.e.*, OFA (Tian Zhou & Jin, 2023) and Time-LLM (Jin et al., 2024)[2].

4.1. Long-term Forecasting

Setup. For long-term forecasting, we evaluate the effectiveness of S^2IP-LLM on Weather, Electricity, Traffic, and four ETT datasets (*i.e.*, ETTh1, ETTh2, ETTm1, and ETTm2), which have been widely adopted as benchmarking datasets for long-term forecasting tasks. Details of these datasets are shown in Appendix A.3, Table 5. The input time series length is 512, and we evaluate the performance on four different horizons $\{96, 192, 336, 720\}$. The evaluation metrics include the mean square error (MSE) and the mean absolute error (MAE).

Results. We compare the forecasting results of S^2IP-LLM to 6 selected baselines in Table 1. Due to the space limitation, the comparisons with the other 6 baselines are provided in Appendix B and Table 6. We can observe that LLMs based forecasting methods, *i.e.*, Time-LLM and OFA, generally achieve better performance than other baseline methods. This should be attributed to the prevalent expressibility of LLMs and their associated prompt-tuning and fine-tuning

[1] https://github.com/thuml/Time-Series-Library

[2] We reproduced results through public available code: https://github.com/KimMeen/Time-LLM. For Time-LLM variants, 'L' denotes the LLaMA backbone, and 'G' refers to the GPT-2 backbone.

Methods		**S^{2}IP-LLM**		**Time-LLM(L)**		**Time-LLM(G)**		**OFA**		**iTransformer**		**Dlinear**		**PatchTST**	
Datasets\Horizon		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.145**	**0.195**	0.148	0.197	0.158	0.210	0.162	0.212	0.253	0.304	0.176	0.237	0.149	0.198
	192	**0.190**	**0.235**	0.194	0.246	0.197	0.245	0.204	0.248	0.280	0.319	0.220	0.282	0.194	0.241
	336	**0.243**	**0.280**	0.248	0.285	0.248	0.285	0.254	0.286	0.321	0.344	0.265	0.319	0.245	0.282
	720	**0.312**	**0.326**	0.317	0.332	0.319	0.334	0.326	0.337	0.364	0.374	0.333	0.362	0.314	0.334
	Avg.	**0.222**	**0.259**	0.226	0.265	0.230	0.268	0.237	0.270	0.304	0.335	0.248	0.300	0.225	0.264
Electricity	96	0.135	0.230	0.140	0.246	0.137	0.237	0.139	0.238	0.147	0.248	0.140	0.237	**0.129**	**0.222**
	192	**0.149**	0.247	0.155	0.253	0.150	0.249	0.153	0.251	0.165	0.267	0.153	0.249	0.157	**0.240**
	336	0.167	0.266	0.175	0.279	0.168	0.266	0.169	0.266	0.178	0.279	0.169	0.267	**0.163**	**0.259**
	720	0.200	**0.287**	0.204	0.305	0.203	0.293	0.206	0.297	0.322	0.398	0.203	0.301	**0.197**	0.290
	Avg.	**0.161**	0.257	0.168	0.270	0.164	0.261	0.167	0.263	0.203	0.298	0.166	0.263	**0.161**	**0.252**
Traffic	96	0.379	0.274	0.383	0.280	0.380	0.277	0.388	0.282	0.367	0.288	0.410	0.282	**0.360**	**0.249**
	192	0.397	0.282	0.399	0.294	0.399	0.288	0.407	0.290	**0.378**	0.293	0.423	0.287	0.379	**0.256**
	336	0.407	0.289	0.411	0.306	0.408	0.290	0.412	0.294	**0.389**	0.294	0.436	0.296	0.392	**0.264**
	720	0.440	0.301	0.448	0.319	0.445	0.308	0.450	0.312	**0.401**	0.304	0.466	0.315	0.432	**0.286**
	Avg.	0.405	0.286	0.440	0.301	0.408	0.290	0.414	0.294	**0.389**	0.295	0.433	0.295	0.390	**0.263**
ETTh1	96	**0.366**	**0.396**	0.380	0.406	0.383	0.410	0.379	0.402	0.395	0.420	0.367	**0.396**	0.379	0.407
	192	**0.401**	0.420	0.426	0.438	0.419	0.435	0.415	0.424	0.427	0.441	**0.401**	**0.419**	0.428	0.442
	336	**0.412**	**0.431**	0.437	0.451	0.426	0.440	0.435	0.440	0.445	0.457	0.434	0.449	0.465	0.465
	720	0.440	0.458	0.515	0.509	**0.428**	**0.456**	0.441	0.459	0.537	0.530	0.472	0.493	0.504	0.500
	Avg.	**0.406**	**0.427**	0.439	0.451	0.414	0.435	0.418	0.431	0.451	0.462	0.418	0.439	0.444	0.453
ETTh2	96	**0.278**	**0.340**	0.306	0.362	0.297	0.357	0.289	0.347	0.304	0.360	0.301	0.367	0.296	0.353
	192	**0.346**	**0.385**	**0.346**	**0.385**	0.349	0.390	0.358	0.392	0.377	0.403	0.394	0.427	0.382	0.404
	336	**0.367**	**0.406**	0.393	0.422	0.373	0.408	0.383	0.414	0.405	0.429	0.506	0.495	0.402	0.425
	720	0.400	0.436	**0.397**	**0.433**	0.400	0.436	0.438	0.456	0.443	0.464	0.805	0.635	0.444	0.465
	Avg.	**0.347**	**0.391**	0.360	0.400	0.355	0.398	0.367	0.402	0.382	0.414	0.502	0.481	0.381	0.411
ETTm1	96	**0.288**	**0.346**	0.311	0.365	0.291	**0.346**	0.296	0.353	0.312	0.366	0.304	0.348	0.303	0.351
	192	**0.323**	**0.365**	0.364	0.395	0.336	0.373	0.335	0.373	0.347	0.385	0.336	0.367	0.341	0.376
	336	**0.359**	0.390	0.369	0.398	0.362	0.390	0.369	0.394	0.379	0.404	0.368	**0.387**	0.377	0.401
	720	**0.403**	**0.418**	0.416	0.425	0.410	0.421	0.418	0.424	0.441	0.442	0.421	**0.418**	0.431	0.436
	Avg.	**0.343**	**0.379**	0.365	0.395	0.349	0.382	0.355	0.386	0.370	0.399	0.357	0.389	0.363	0.391
ETTm2	96	**0.165**	**0.257**	0.170	0.262	0.184	0.275	0.170	0.264	0.179	0.271	0.168	0.263	0.173	0.262
	192	**0.222**	**0.299**	0.229	0.303	0.238	0.310	0.231	0.306	0.242	0.313	0.229	0.310	0.231	0.300
	336	**0.277**	**0.330**	0.281	0.335	0.286	0.340	0.280	0.339	0.288	0.344	0.289	0.352	0.292	0.345
	720	**0.363**	**0.390**	0.379	0.403	0.379	0.403	0.373	0.402	0.378	0.397	0.416	0.437	0.371	0.394
	Avg.	**0.257**	**0.319**	0.264	0.325	0.271	0.332	0.265	0.328	0.272	0.331	0.275	0.340	0.267	0.325

Table 1: Long-term forecasting results for {96, 192, 336, 720} horizons. Lower values indicate better performance. For Time-LLM variants, 'L' denotes the LLaMA backbone (Touvron et al., 2023a), and 'G' refers to the GPT-2 backbone (Radford et al., 2019). More results are in Appendix B, Table 6

strategies, respectively. Moreover, most of the time, S^2IP-LLM outperforms Time-LLM and OFA over 7 different datasets. This is because (1) the unique way S^2IP-LLM tokenized the input time series data can yield better time series representations, and (2) the semantic space informed prompting can help further enhance the time series representation which will be further demonstrated in Section 4.5.

4.2. Short-term Forecasting

Setup. We also evaluate the effectiveness of S^2IP-LLM with the short-term forecasting setting based on the M4 datasets (Makridakis et al., 2018). It contains a collection of marketing data that are sampled at different frequencies. Details of the datasets can be found in Appendix A.3. The prediction horizons are significantly shorter than the long-term forecasting setting and are set between 6 and 48. The input lengths are twice the prediction horizons, similar to the experiment setting in (Jin et al., 2024; Tian Zhou & Jin, 2023). The evaluation metrics for short-term forecasting are symmetric mean absolute percentage error (SMAPE), mean absolute scaled error (MASE), and overall weighted average (OWA). The details of these evaluation metrics are provided in Appendix A.4.

Results. Table 2 summarizes the short-term forecasting results and the full experiment results are shown in Appendix Appendix C, Table 7. We observe that S^2IP-LLM outperforms all other baselines by a large margin and is slightly better than PatchTST. This could attribute to the tokenization design as well as the semantic space informed prompting within S^2IP-LLM.

Methods		S^2IP-LLM	Time-LLM(G)	OFA	iTransformer	Dlinear	PatchTST	TimesNet	FEDformer	Autoformer
Avg.	**SMAPE**	**12.021**	12.494	12.690	12.142	13.639	12.059	12.880	13.160	12.909
	MASE	**1.612**	1.731	1.808	1.631	2.095	1.623	1.836	1.775	1.771
	OWA	**0.857**	0.913	0.94	0.874	1.051	0.869	0.955	0.949	0.939

Table 2: Short-term time series forecasting results on M4 datasets. The forecasting horizons are in [6, 48] and the three rows provided are weighted averaged from all datasets under different sampling intervals. A lower value indicates better performance. Detailed short-term forecasting results are in Appendix C, Table 7

Methods	S^2IP-LLM		Time-LLM(G)		OFA		iTransformer		Dlinear		PatchTST		TimesNet	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	**0.233**	**0.272**	0.237	0.275	0.238	0.275	0.308	0.338	0.241	0.283	0.242	0.279	0.279	0.301
Electricity	**0.175**	0.271	0.177	0.273	0.176	**0.269**	0.196	0.293	0.180	0.280	0.180	0.273	0.323	0.392
Traffic	**0.427**	0.307	0.429	0.307	0.440	0.310	0.495	0.361	0.447	0.313	0.430	**0.305**	0.951	0.535
ETTh1	0.593	0.529	0.785	0.553	**0.590**	**0.525**	0.910	0.860	0.691	0.600	0.633	0.542	0.869	0.628
ETTh2	0.419	0.439	0.424	0.441	**0.397**	**0.421**	0.489	0.483	0.605	0.538	0.415	0.431	0.479	0.465
ETTm1	0.455	0.435	0.487	0.461	0.464	0.441	0.728	0.565	**0.411**	**0.429**	0.501	0.466	0.677	0.537
ETTm2	**0.284**	**0.332**	0.305	0.344	0.293	0.335	0.336	0.373	0.316	0.368	0.296	0.343	0.320	0.353

Table 3: Long-term forecasting results for {96, 192, 336, 720} horizons. A lower value indicates a better performance. Few-shot learning on 10% training data setting. All results are averaged from four forecasting horizons{96, 192, 336, 720}. Detailed results are in Appendix C,Table 8.

4.3. Few-shot Forecasting

Setup. We follow the experimental settings in Tian Zhou & Jin (2023) to evaluate the performance in the few-shot forecasting setting, which allows us to examine whether the model can generate accurate forecasting with limited training data. We use the first 5% and 10% of the training data in these experiments.

Results. To ensure a fair comparison in the long-term forecasting setting, we summarize the few-shot learning experiment results under 10% and 5% training data in Table 3 and Table 4, respectively. We also report the full experiment results in Table 8 and Table 9 of Appendix D, respectively. When trained with only 10% of the data, S^2IP-LLM typically ranks as either the best or the second-best compared to other baseline models across different datasets. Meanwhile, we also observe that LLMs based methods, S^2IP-LLM, Time-LLM, and OFA significantly outperform other baseline methods. This is because other baseline methods are trained from scratch and they only have limited training data in this case. On the other hand, LLMs based methods can adapt/align the pre-trained knowledge with the time series embedding to enhance its representation. Even with only 5% of training data, S^2IP-LLM still exhibits, if not superior, comparable performance to time-LLM and OFA.

4.4. Ablation Studies and Parameter Sensitivity

We conduct ablation studies on the ETTh2 and ETTm2 datasets to evaluate the parameter sensitivity for S^2IP-LLM. Figure 3 (1) and (4) presents the experiment results when the length of the prompt varies on ETTh2 and ETTm2, respectively. Within a limited range, *i.e.* 2 to 8, an increase in the prompt length tends to improve the forecasting performance. However, excessive prompt length, such as lengths of 16 or 32, results in a significant decline in the forecasting accuracy. A similar pattern can be observed in the hyperparameter analysis of the λ, which controls the strength of alignment. As shown in Figure 3 (2) and (5), when λ varies from 0 to 0.05, slightly larger λ is beneficial for representation learning within the joint space, showing better forecasting results. On the other hand, larger λ tends to lead to indistinguishable time series representation and the forecasting performance will thus decrease. Finally, Figure 3 (3) and (6) show the effects of choosing different numbers of semantic anchors. Generally, an increased number of semantic anchors improves the forecasting results. We conjecture that the small number hinders the learning of highly representative semantic anchors in the joint space and thus will generate less informed prompts for time series embedding. We visualize the prompted time series embeddings with different numbers of semantic anchors in Appendix E, Figure 6. We notice that a smaller quantity of semantic anchors leads to a less dispersed distribution in the joint space, indicating that the generated prompts could be less informative for time series embedding. We also perform ablation studies by incrementally adding the "alignment & prompting" and "decomposition" modules. In Appendix E Table 10, we observe the forecasting performance increases when we sequentially activate the prompting & alignment component and the decomposition component, which implies the importance of these modules in S^2IP-LLM.

Methods	S^2IP-LLM		Time-LLM(G)		OFA		iTransformer		Dlinear		PatchTST		TimesNet	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	**0.260**	**0.297**	0.264	0.301	0.263	0.301	0.309	0.339	0.263	0.308	0.269	0.303	0.298	0.318
Electricity	0.179	0.275	0.181	0.279	0.178	**0.273**	0.201	0.296	**0.176**	0.275	0.181	0.277	0.402	0.453
Traffic	0.420	0.299	0.423	0.302	0.434	0.305	0.450	0.324	0.450	0.317	**0.418**	**0.296**	0.867	0.493
ETTh1	**0.650**	**0.550**	0.891	0.627	0.681	0.560	1.070	0.710	0.750	0.611	0.694	0.569	0.925	0.647
ETTh2	**0.380**	**0.413**	0.581	0.519	0.400	0.433	0.488	0.475	0.694	0.577	0.827	0.615	0.439	0.448
ETTm1	0.455	0.446	0.524	0.479	0.472	0.450	0.784	0.596	**0.400**	**0.417**	0.526	0.476	0.717	0.561
ETTm2	**0.296**	**0.342**	0.325	0.361	0.308	0.346	0.356	0.388	0.399	0.426	0.314	0.352	0.344	0.372

Table 4: Long-term forecasting results for {96, 192, 336, 720} horizons. A lower value indicates a better performance. Few-shot learning on 5% training data setting. All results are averaged from four forecasting horizons{96, 192, 336, 720}. Detailed results are in Appendix C, Table 9.

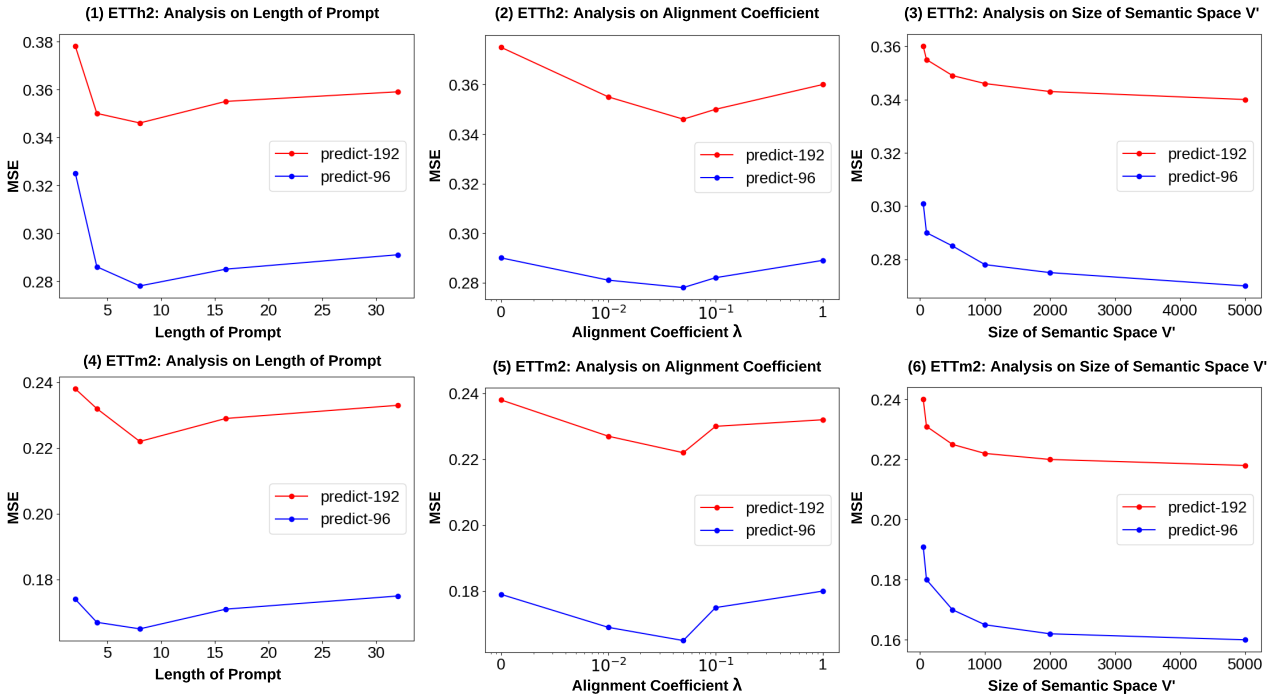

Figure 3. Parameter sensitivity analysis in predicting 96 and 192 steps: (1) and (4) show the effect of prompt length on ETTh2 and ETTm2 datasets; (2) and (5) show the effect of alignment coefficient λ on ETTh2 and ETTm2 datasets; (3) and (6) show the effect of semantic space size V' on ETTh2 and ETTm2 datasets.

4.5. Qualitative Analysis

In this section, we perform a qualitative analysis of how semantic space informed prompting can facilitate time series representation. Figure 4 shows the visualization of learned semantic anchor embeddings, time series embeddings, and the prompted time series embeddings. The semantic anchor embeddings from the pre-trained language model show distinct clusters, suggesting a robust and differentiated embedding space. In contrast, the raw time series embeddings reveal a more spread-out and less clustered pattern, suggesting that before the alignment, the time series representation is comparatively less informative. After the alignment, the prompted time series embeddings show a clear clustered pattern, suggesting that by aligning with the semantic anchors, time series representation becomes more distinguishable in the joint space.

We also provide the visualizations of prompted time series embeddings under different hyperparameters (when λ varies). Within a smaller range, the increase of λ appears to enhance the separation of time series embeddings, indicating a more distinct and informative representation. However, as λ becomes excessively large, we observe a significant

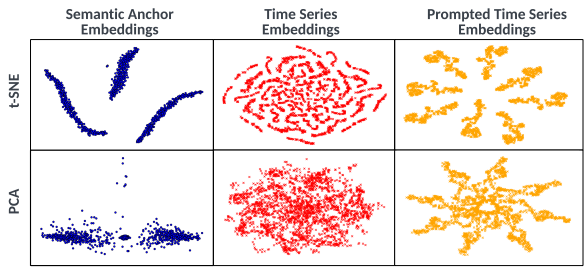

Figure 4. The t-SNE and PCA plots of embeddings space: **blue**: semantic anchor embeddings; **red**: time series embeddings; **orange**: prefix-prompted time series embeddings

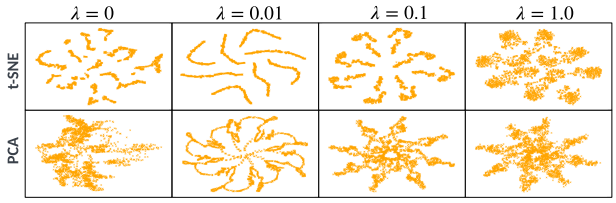

Figure 5. The t-SNE and PCA plots prefix-prompted time series embeddings with different λ

decline in the clustering quality of the prompted time series embeddings, which suggests that beyond a threshold, a higher λ value leads to less informative embeddings.

5. Conclusion

In this paper, we present S^2IP-LLM, a novel framework for time series forecasting utilizing pre-trained language models. S^2IP-LLM introduces a time series tokenization module that provides expressive local contexts by the concatenation of decomposed time series patches. It creates informative joint space by aligning time series contexts with semantics anchors derived from the pre-trained word token embeddings. The selected aligned semantic anchors are retrieved as prompt indicators to enhance the time series representation and facilitate underlying forecasting tasks. Our thorough empirical studies justified the effectiveness of S^2IP-LLM.

Impact Statement

This work introduces significant advancements in time series forecasting, leveraging the power of pre-trained language models and semantic information. The broader impact of this work can be multifaceted. It may enhance decision-making in critical domains such as finance, healthcare, and environmental monitoring by providing more accurate and reliable forecasts and could lead to better resource allocation, improved patient care, and more effective responses to climate change. No ethical concerns must be considered. The social impacts are significant, as it has the potential to revolutionize our approach to complex time series data and the integration of emerging AI tools, including foundational models. It could change how we analyze and leverage time series data in various fields.

References

Achiam, O. J. and et al., S. A. Gpt-4 technical report. 2023. URL https://api.semanticscholar.org/CorpusID:257532815.

Anderson, O. D. and Kendall, M. G. Time-series. 2nd edn. *The Statistician*, 25:308, 1976. URL https://api.semanticscholar.org/CorpusID:134001785.

Bao, H., Dong, L., Piao, S., and Wei, F. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Böse, J.-H., Flunkert, V., Gasthaus, J., Januschowski, T., Lange, D., Salinas, D., Schelter, S., Seeger, M., and Wang, Y. Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment*, 10(12):1694–1705, 2017.

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

Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., Tong, Y., Xu, B., Bai, J., Tong, J., et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33:17766–17778, 2020.

Cao, D., Jia, F., Arik, S. O., Pfister, T., Zheng, Y., Ye, W., and Liu, Y. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*, 2023.

Challu, C., Olivares, K. G., Oreshkin, B. N., Ramirez, F. G., Canseco, M. M., and Dubrawski, A. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997, 2023.

Chang, C., Peng, W.-C., and Chen, T.-F. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *arXiv preprint arXiv:2308.08469*, 2023.

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

Courty, P. and Li, H. Timing of seasonal sales. *The Journal of Business*, 72(4):545–572, 1999.

Cui, C., Ma, Y., Cao, X., Ye, W., Zhou, Y., Liang, K., Chen, J., Lu, J., Yang, Z., Liao, K.-D., et al. A survey on multimodal large language models for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 958–979, 2024.

Deldari, S., Xue, H., Saeed, A., He, J., Smith, D. V., and Salim, F. D. Beyond just vision: A review on self-supervised representation learning on multimodal and temporal data. *arXiv preprint arXiv:2206.02353*, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dimri, T., Ahmad, S., and Sharif, M. Time series analysis of climate variables using seasonal arima approach. *Journal of Earth System Science*, 129:1–16, 2020.

Ethayarajh, K. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*, 2019.

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. Transfer learning for time series classification. In *2018 IEEE international conference on big data (Big Data)*, pp. 1367–1376. IEEE, 2018.

Friedman, M. The interpolation of time series by related series. *Journal of the American Statistical Association*, 57(300):729–757, 1962.

Gao, J., Song, X., Wen, Q., Wang, P., Sun, L., and Xu, H. Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks. *arXiv preprint arXiv:2002.09545*, 2020.

Garza, A. and Mergenthaler-Canseco, M. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Jiang, Y., Pan, Z., Zhang, X., Garg, S., Schneider, A., Nevmyvaka, Y., and Song, D. Empowering time series analysis with large language models: A survey. *arXiv preprint arXiv:2402.03182*, 2024.

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. Time-llm: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024.

Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., and Choo, J. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2021.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Li, N., Arnold, D. M., Down, D. G., Barty, R., Blake, J., Chiang, F., Courtney, T., Waito, M., Trifunov, R., and Heddle, N. M. From demand forecasting to inventory ordering decisions for red blood cells through integrating machine learning, statistical modeling, and inventory optimization. *Transfusion*, 62(1):87–99, 2022.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

Li, Y., Wang, S., Ding, H., and Chen, H. Large language models in finance: A survey. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pp. 374–382, 2023.

Liu, H., Ma, Z., Yang, L., Zhou, T., Xia, R., Wang, Y., Wen, Q., and Sun, L. Sadi: A self-adaptive decomposed interpretable framework for electric load forecasting under extreme events. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023a.

Liu, Y., Wu, H., Wang, J., and Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35:9881–9893, 2022.

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., and Long, M. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023b.

Lu, K., Grover, A., Abbeel, P., and Mordatch, I. Frozen pretrained transformers as universal computation engines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7628–7636, 2022.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4): 802–808, 2018.

Nate Gruver, Marc Finzi, S. Q. and Wilson, A. G. Large Language Models Are Zero Shot Time Series Forecasters. In *Advances in Neural Information Processing Systems*, 2023.

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

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Pan, Z., Jiang, Y., Song, D., Garg, S., Rasul, K., Schneider, A., and Nevmyvaka, Y. Structural knowledge informed continual multivariate time series forecasting. *arXiv preprint arXiv:2402.12722*, 2024.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Qin, Y., Song, D., Cheng, H., Cheng, W., Jiang, G., and Cottrell, G. W. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2627–2633, 2017.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Rasul, K., Ashok, A., Williams, A. R., Khorasani, A., Adamopoulos, G., Bhagwatkar, R., Biloš, M., Ghonia, H., Hassen, N. V., Schneider, A., et al. Lag-llama: Towards foundation models for time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.

Shang, C., Chen, J., and Bi, J. Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*, 2021.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., et al. Large language models encode clinical knowledge. *arXiv preprint arXiv:2212.13138*, 2022.

Sun, C., Li, Y., Li, H., and Hong, S. Test: Text prototype aligned embedding to activate llm's ability for time series. *arXiv preprint arXiv:2308.08241*, 2023.

Tang, Y., Qu, A., Chow, A. H., Lam, W. H., Wong, S., and Ma, W. Domain adversarial spatial-temporal network: a transferable framework for short-term traffic forecasting across cities. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 1905–1915, 2022.

Taylor, S. J. and Letham, B. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

Tian Zhou, Peisong Niu, X. W. L. S. and Jin, R. One Fits All: Power general time series analysis by pretrained lm. In *NeurIPS*, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023b.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023c.

Woo, G., Liu, C., Sahoo, D., Kumar, A., and Hoi, S. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

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

Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., and Zhang, C. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Xue, H. and Salim, F. D. Promptcast: A new prompt-based learning paradigm for time series forecasting. 2022.

Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., and Chen, E. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023.

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

Zhang, T., Zhang, Y., Cao, W., Bian, J., Yi, X., Zheng, S., and Li, J. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186*, 2022a.

Zhang, X., Zhao, Z., Tsiligkaridis, T., and Zitnik, M. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems*, 35:3988–4003, 2022b.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

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

A. Experimental Details

A.1. Implementation

We mainly follow the experimental configurations in (Wu et al., 2023) across all baselines within a unified evaluation pipeline, available at https://github.com/thuml/Time-Series-Library, for a fair comparison. We use GPT2-small (Radford et al., 2019) with the first 6 hidden layers enabled as the default backbone model. All our experiments are repeated three times and we report the averaged results. We implemented the model on PyTorch (Paszke et al., 2019) with all experiments conducted on NVIDIA RTX A6000 GPUs and NVIDIA A100 GPUs. We configure the patch length P as 16 with a stride S of 8. Our experimental results were obtained by performing a grid search over combinations within the following search spaces: long-term trend length (24, 48, 96, 144), seasonal trend length (2, 4, 8, 12, 24), prompt length (2, 4, 8, 16, 32), and semantic space size (1000, 2000, 5000). Our code is available at https://github.com/panzijie825/S2IP-LLM.

A.2. Baseline Introduction

We introduce the baseline models that we choose to compare in the following section:

- **Time-LLM** (Jin et al., 2024): Time-LLM reprograms time series tokens with NLP representation using multi-head attention and fine-tunes the pre-trained LLM with the prefix prompting to perform time series analysis. We reproduced and reported the experimental results for long-term forecasting using two LLM backbones: 'L' represents LLaMA (Touvron et al., 2023a), and 'G' represents GPT-2 (Radford et al., 2019).
- **OFA** (Tian Zhou & Jin, 2023): OFA represents time series data into patched tokens to fine-tune the pre-trained GPT2 (Radford et al., 2019) for various time series analysis tasks.
- **iTransformer** (Liu et al., 2023b): iTransformer applies the attention and feed-forward network on the inverted dimensions of the time series data to capture multivariate correlations.
- **Dlinear** (Zeng et al., 2023): Dlinear incorporates the decomposition with linear layer to model the time series data via modeling trend and seasonal components separately.
- **PatchTST** (Nie et al., 2023): PatchTST leverages a Transformer-based model for time series forecasting by segmenting data into patches and using a channel-independent design to efficiently reduce computational costs and boost forecasting performance.
- **TimesNet** (Wu et al., 2023): TimesNet converts 1D time series data into 2D representation and capture intra- and inter-period relations. It designs TimesBlock with an inception block to extract complex temporal patterns, leading to multiple time series tasks.
- **FEDformer** (Zhou et al., 2022): FEDformer incorporates seasonal-trend decomposition with Transformers for time series forecasting. It leverages information from the frequency domain, gaining efficiency and accuracy in time series analysis.
- **Autoformer** (Wu et al., 2021): Autoformer proposes the decomposition architecture with Auto-Correlation mechanisms to efficiently and accurately perform long-term forecasting.
- **Stationary** (Liu et al., 2022): Non-stationary Transformers proposes a framework with two interdependent modules, namely series stationarization and de-stationary attention to gain robust time series forecasting results.
- **ETSformer** (Woo et al., 2022): ETSformer integrates exponential smoothing principles by replacing traditional self-attention with exponential smoothing attention and frequency attention for time series forecasting
- **LightTS** (Zhang et al., 2022a): LightTS is a time series classification framework that includes adaptive ensemble distillation and Pareto optimization, resulting in accurate classification with limited resources.

We note that patching-based methods, *i.e.* OFA (Tian Zhou & Jin, 2023), PatchTST (Nie et al., 2023), and Time-LLM (Jin et al., 2024) treat multivariate time series as independently univariate time series, which essentially provide more training data for those models. For transformer-based models which rely on multivariate times input, this could be the reason that their performances are not as good as patching-based ones.

A.3. Details of Datasets

We experiment the long-term forecasting on the widely adopted Electricity Transformer Temperature (ETT) datasets (Zhou et al., 2021), Weather, Electricity, and Traffic from (Wu et al., 2023). We also experiment the short-term forecasting using the M4 benchmark dataset (Makridakis et al., 2018).
ETT datasets are comprised of roughly two years of data from two locations in China. The data are further divided into four

distinct datasets, each with different sampling rates: ETTh1 and ETTh2 are sampled hourly, and ETTm1 and ETTm2 are sampled every 15 minutes. Every ETT dataset includes six power load features and a target variable: the oil temperature. The Electricity dataset comprises records of electricity consumption from 321 customers and is measured with a 1-hour sampling rate. The Weather dataset contains one-year records from 21 meteorological stations located in Germany. The sampling rate for the Weather dataset is 10 minutes. The Traffic dataset includes the per-hour sampled occupancy rates of the freeway system, which were recorded from 862 sensors in California. The M4 benchmark dataset has 100 thousand time series, which were collected from various domains ranging from business to economic forecasting. The time series data are partitioned into six groups with varied sampling rates from yearly to hourly.

The full data statistics are summarized in Table 5

Tasks	Datasets	Dim.	Series Length	Dataset Size	Frequency	Information
Long-term Forecasting	ETTm1	7	{96,192,336,720}	(34465, 11521, 11521)	15 min	Temperature
	ETTm2	7	{96,192,336,720}	(34465, 11521, 11521)	15 min	Temperature
	ETTh1	7	{96,192,336,720}	(8545, 2881, 2881)	1 hour	Temperature
	ETTh2	7	{96,192,336,720}	(8545, 2881, 2881)	1 hour	Temperature
	Electricity	321	{96,192,336,720}	(18317, 2633, 5261)	1 hour	Electricity
	Traffic	862	{96,192,336,720}	(12185, 1757, 3509)	1 hour	Transportation
	Weather	21	{96,192,336,720}	(36792, 5271, 10540)	10 min	Weather
Short-term Forecasting	M4-Yearly	1	6	(23000, 0, 23000)	Yearly	Demographic
	M4-Quarterly	1	8	(24000, 0, 24000)	Quarterly	Finance
	M4-Monthly	1	18	(48000, 0, 48000)	Monthly	Industry
	M4-Weekly	1	13	(359, 0, 359)	Weekly	Macro
	M4-Daily	1	14	(4227, 0, 4227)	Daily	Micro
	M4-Hourly	1	48	(414, 0, 414)	Hourly	Other

Table 5: Dataset statistics are from (Wu et al., 2023). The dimension indicates the number of time series variables, and the dataset size is organized in (training, validation, and testing).

A.4. Evaluation Metrics

For evaluation metrics, we use the mean square error (MSE) and mean absolute error (MAE) for long-term forecasting. For short-term forecasting on the M4 benchmark, we use the symmetric mean absolute percentage error (SMAPE), mean absolute scaled error (MASE), and overall weighted average (OWA) (Oreshkin et al., 2019), which is a specific metric for the M4 competition. We present the calculations of these metrics as follows:

$$\text{MSE} = \frac{1}{H}\sum_{h=1}^{T}\left(\mathbf{Y}_h - \hat{\mathbf{Y}}_h\right)^2, \qquad \text{MAE} = \frac{1}{H}\sum_{h=1}^{H}\left|\mathbf{Y}_h - \hat{\mathbf{Y}}_h\right|,$$

$$\text{SMAPE} = \frac{200}{H}\sum_{h=1}^{H}\frac{\left|\mathbf{Y}_h - \hat{\mathbf{Y}}_h\right|}{|\mathbf{Y}_h| + \left|\hat{\mathbf{Y}}_h\right|}, \qquad \text{MAPE} = \frac{100}{H}\sum_{h=1}^{H}\frac{\left|\mathbf{Y}_h - \hat{\mathbf{Y}}_h\right|}{|\mathbf{Y}_h|},$$

$$\text{MASE} = \frac{1}{H}\sum_{h=1}^{H}\frac{\left|\mathbf{Y}_h - \hat{\mathbf{Y}}_h\right|}{\frac{1}{H-s}\sum_{j=s+1}^{H}|\mathbf{Y}_j - \mathbf{Y}_{j-s}|}, \qquad \text{OWA} = \frac{1}{2}\left[\frac{\text{SMAPE}}{\text{SMAPE}_{\text{Naïve2}}} + \frac{\text{MASE}}{\text{MASE}_{\text{Naïve2}}}\right]$$

where s is the time series data periodicity. H denotes the prediction intervals. Y_h and $\hat{Y_h}$ are the h-th ground truth and prediction where h $\in \{1, ..., H\}$. For the evaluation metrics in long-term forecasting, we clarify that the reported metrics are the normalized versions of MAE/MSE. Although we apply global standardization to the data, the information that the scaler used is from training data solely.

Methods		**S^{2}IP-LLM**		**TimesNet**		**FEDformer**		**Autoformer**		**Stationary**		**ETSformer**		**LightTS**	
Datasets\Horizon		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.145**	**0.195**	0.172	0.220	0.217	0.296	0.266	0.336	0.173	0.223	0.197	0.281	0.182	0.242
	192	**0.190**	**0.235**	0.219	0.261	0.276	0.336	0.307	0.367	0.245	0.285	0.237	0.312	0.227	0.287
	336	**0.243**	**0.280**	0.280	0.306	0.339	0.380	0.359	0.395	0.321	0.338	0.298	0.353	0.282	0.334
	720	**0.312**	**0.326**	0.365	0.359	0.403	0.428	0.419	0.428	0.414	0.410	0.352	0.288	0.352	0.386
	Avg	**0.222**	**0.259**	0.259	0.287	0.309	0.360	0.338	0.382	0.288	0.314	0.271	0.334	0.261	0.312
Electricity	96	0.135	0.230	0.168	0.272	0.193	0.308	0.201	0.317	0.169	0.273	0.187	0.304	0.207	0.307
	192	**0.149**	0.247	0.184	0.289	0.201	0.315	0.222	0.334	0.182	0.286	0.199	0.315	0.213	0.316
	336	0.167	0.266	0.198	0.300	0.214	0.329	0.231	0.338	0.200	0.304	0.212	0.329	0.230	0.333
	720	0.200	**0.287**	0.220	0.320	0.246	0.355	0.254	0.361	0.222	0.321	0.233	0.345	0.265	0.360
	Avg	**0.161**	0.257	0.192	0.295	0.214	0.327	0.227	0.338	0.193	0.296	0.208	0.323	0.229	0.329
Traffic	96	0.379	0.274	0.593	0.321	0.587	0.366	0.613	0.388	0.612	0.338	0.607	0.392	0.615	0.391
	192	0.397	0.282	0.617	0.336	0.604	0.373	0.616	0.382	0.613	0.340	0.621	0.399	0.601	0.382
	336	0.407	0.289	0.629	0.336	0.621	0.383	0.622	0.337	0.618	0.328	0.622	0.396	0.613	0.386
	720	0.440	0.301	0.640	0.350	0.626	0.382	0.660	0.408	0.653	0.355	0.632	0.396	0.658	0.407
	Avg	0.405	0.286	0.620	0.336	0.610	0.376	0.628	0.379	0.624	0.340	0.621	0.396	0.622	0.392
ETTh1	96	**0.366**	**0.396**	0.468	0.475	0.376	0.419	0.530	0.517	0.513	0.491	0.644	0.589	0.440	0.450
	192	**0.401**	0.420	0.484	0.485	0.420	0.448	0.537	0.521	0.534	0.504	0.736	0.648	0.498	0.479
	336	**0.412**	**0.431**	0.536	0.516	0.459	0.465	0.596	0.583	0.588	0.535	0.827	0.707	0.550	0.510
	720	0.440	0.458	0.593	0.537	0.506	0.507	0.713	0.639	0.643	0.616	0.946	0.766	0.615	0.571
	Avg	**0.406**	**0.427**	0.520	0.505	0.440	0.460	0.594	0.565	0.570	0.537	0.788	0.677	0.526	0.502
ETTh2	96	**0.278**	**0.340**	0.376	0.415	0.358	0.397	0.454	0.490	0.476	0.458	0.340	0.391	0.408	0.445
	192	**0.346**	**0.385**	0.409	0.440	0.429	0.439	0.486	0.517	0.512	0.493	0.430	0.439	0.561	0.526
	336	**0.367**	**0.406**	0.425	0.455	0.496	0.487	0.493	0.533	0.552	0.551	0.485	0.479	0.673	0.580
	720	0.400	0.436	0.488	0.494	0.463	0.474	0.515	0.543	0.562	0.560	0.500	0.497	1.006	0.721
	Avg	**0.347**	**0.391**	0.425	0.451	0.437	0.449	0.487	0.520	0.526	0.516	0.439	0.452	0.662	0.568
ETTm1	96	**0.288**	**0.346**	0.329	0.377	0.379	0.419	0.568	0.516	0.386	0.398	0.375	0.398	0.383	0.409
	192	**0.323**	**0.365**	0.371	0.401	0.426	0.441	0.573	0.528	0.459	0.444	0.408	0.410	0.421	0.431
	336	**0.359**	0.390	0.417	0.428	0.445	0.459	0.587	0.534	0.495	0.464	0.435	0.428	0.454	0.456
	720	**0.403**	**0.418**	0.483	0.464	0.543	0.490	0.589	0.536	0.585	0.516	0.499	0.462	0.549	0.520
	Avg	**0.343**	**0.379**	0.400	0.417	0.448	0.452	0.579	0.529	0.481	0.456	0.429	0.425	0.452	0.454
ETTm2	96	**0.165**	**0.257**	0.201	0.286	0.203	0.287	0.287	0.359	0.192	0.274	0.189	0.280	0.239	0.335
	192	**0.222**	**0.299**	0.260	0.329	0.269	0.328	0.325	0.388	0.280	0.339	0.253	0.319	0.346	0.412
	336	**0.277**	**0.330**	0.331	0.376	0.325	0.366	0.498	0.491	0.334	0.361	0.314	0.357	0.506	0.506
	720	**0.363**	**0.390**	0.428	0.430	0.421	0.415	0.548	0.517	0.417	0.413	0.414	0.413	0.702	0.606
	Avg	**0.257**	**0.319**	0.305	0.355	0.305	0.349	0.414	0.439	0.306	0.347	0.293	0.342	0.448	0.465

Table 6: Transformer-based Models Long-term forecasting results for {96, 192, 336, 720} horizons. A lower value indicates a better performance.

B. Long-term Forecasting Results

Table 6 shows the detailed results of all prediction lengths of five Transformer-based forecasting models. S^2IP-LLM shows a strong and relatively stable performance across different datasets compared to other transformer-based models.

We note that patching-based methods, *i.e.* OFA (Tian Zhou & Jin, 2023), PatchTST (Nie et al., 2023) treat multivariate time series independently as univariate time series, which essentially provide more training data for univariate time series input based models. This potentially contributes to their advantages in terms of performance. Thus, it may create an unfair comparison to methods with truly multivariate inputs, *i.e.* the other transformer-based models.

C. Full Short-term Forecasting Results

Table 7 shows the full short-term forecasting experiment results on M4 datasets. S^2IP-LLM consistently outperforms the majority of baseline models in most cases. It surpasses the performance of OFA significantly and achieves slightly better forecasting performance than PatchTST, which can be attributed to proposed semantic space informed prompting.

D. Full Few-shot Forecasting Results

Table 8 and Table 9 show the full few-short forecasting experiment results with 10% and 5% of the training data respectively.

E. Ablation Studies and Parameter Sensitivity

We provide the t-SNE and PCA visualization of semantic anchor and prefix-prompted time series embeddings with different V' in Figure 6. We observe semantic anchor embeddings display a continued spanning pattern among the joint space, whereas the prompted time series representation shows only a slight visual difference. It is reasonable since it is primarily

Methods		**S^{2}IP-LLM**	Time-LLM(G)	OFA	iTrans.	Dlinear	PatchTST	N-HiTS	N-BEATS	TimesNet	FEDformer	Autoformer	Stationary	ETSformer
Year.	SMAPE	**13.413**	13.75	15.11	13.652	16.965	13.477	13.422	13.487	15.378	14.021	13.974	14.727	18.009
	MASE	3.024	3.055	3.565	3.095	4.283	**3.019**	3.056	3.036	3.554	3.036	3.134	3.078	4.487
	OWA	**0.792**	0.805	0.911	0.807	1.058	**0.792**	0.795	0.795	0.918	0.811	0.822	0.807	1.115
Quart.	SMAPE	10.352	10.671	10.597	10.353	12.145	10.38	**10.185**	10.564	10.465	11.100	11.338	10.958	13.376
	MASE	1.228	1.276	1.253	1.209	1.520	1.233	**1.18**	1.252	1.227	1.35	1.365	1.325	1.906
	OWA	0.922	0.95	0.938	0.911	1.106	0.921	**0.893**	0.936	0.923	0.996	1.012	0.981	1.302
Month.	SMAPE	12.995	13.416	13.258	13.079	13.514	**12.959**	13.059	13.089	13.513	14.403	13.958	13.917	14.588
	MASE	**0.97**	1.045	1.003	0.974	1.037	**0.970**	1.013	0.996	1.039	1.147	1.103	1.097	1.368
	OWA	0.91	0.957	0.931	0.911	0.956	**0.905**	0.929	0.922	0.957	1.038	1.002	0.998	1.149
Others.	SMAPE	4.805	4.973	6.124	4.78	6.709	4.952	**4.711**	6.599	6.913	7.148	5.485	6.302	7.267
	MASE	3.247	3.412	4.116	3.231	4.953	3.347	**3.054**	4.430	4.507	4.064	3.865	4.064	5.240
	OWA	1.017	1.053	1.259	1.012	1.487	1.049	**0.977**	1.393	1.438	1.304	1.187	1.304	1.591
Avg.	SMAPE	**12.021**	12.494	12.690	12.142	13.639	12.059	12.035	12.250	12.880	13.160	12.909	12.780	14.718
	MASE	**1.612**	1.731	1.808	1.631	2.095	1.623	1.625	1.698	1.836	1.775	1.771	1.756	2.408
	OWA	**0.857**	0.913	0.94	0.874	1.051	0.869	0.869	0.896	0.955	0.949	0.939	0.930	1.172

Table 7: Detailed short-term time series forecasting results on M4 datasets. The forecasting horizons are in [6, 48] and the last three rows are weighted averaged from all datasets under different sampling intervals. A lower value indicates better performance.

Methods		**S^{2}IP-LLM**		Time-LLM(G)		OFA		iTrans.		Dlinear		PatchTST		TimesNet		FEDformer		Autoformer		Stationary		ETSformer		LightTS	
Data.	Hori.	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.159**	**0.210**	0.160	0.213	0.163	0.215	0.253	0.307	0.171	0.224	0.165	0.215	0.184	0.230	0.188	0.253	0.221	0.297	0.192	0.234	0.199	0.272	0.217	0.269
	192	**0.200**	**0.251**	0.204	0.254	0.210	0.254	0.292	0.328	0.215	0.263	0.210	0.257	0.245	0.283	0.250	0.304	0.270	0.322	0.269	0.295	0.279	0.332	0.259	0.304
	336	0.257	0.293	**0.255**	**0.291**	0.256	0.292	0.322	0.346	0.258	0.299	0.259	0.297	0.305	0.321	0.312	0.346	0.320	0.351	0.370	0.357	0.356	0.386	0.303	0.334
	720	**0.317**	**0.335**	0.329	0.345	0.321	0.339	0.365	0.374	0.320	0.346	0.332	0.346	0.381	0.371	0.387	0.393	0.390	0.396	0.441	0.405	0.437	0.448	0.377	0.382
	Avg	**0.233**	**0.272**	0.237	0.275	0.238	0.275	0.308	0.338	0.241	0.283	0.242	0.279	0.279	0.301	0.284	0.324	0.300	0.342	0.318	0.323	0.318	0.360	0.289	0.322
Electricity	96	0.143	0.243	**0.137**	0.240	0.139	**0.237**	0.154	0.257	0.150	0.253	0.140	0.238	0.299	0.373	0.231	0.323	0.261	0.348	0.420	0.466	0.599	0.587	0.350	0.425
	192	0.159	0.258	0.159	0.258	**0.156**	**0.252**	0.171	0.272	0.164	0.264	0.160	0.255	0.305	0.379	0.261	0.356	0.338	0.406	0.411	0.459	0.620	0.598	0.376	0.448
	336	**0.170**	**0.269**	0.181	0.278	0.175	0.270	0.196	0.295	0.181	0.282	0.180	0.276	0.319	0.391	0.360	0.445	0.410	0.474	0.434	0.473	0.662	0.619	0.428	0.485
	720	0.230	**0.315**	0.232	0.317	0.233	0.317	0.263	0.348	**0.223**	0.321	0.241	0.323	0.369	0.426	0.530	0.585	0.715	0.685	0.510	0.521	0.757	0.664	0.611	0.597
	Avg	**0.175**	0.271	0.177	0.273	0.176	**0.269**	0.196	0.293	0.180	0.280	0.180	0.273	0.323	0.392	0.346	0.427	0.431	0.478	0.444	0.480	0.660	0.617	0.441	0.489
Traffic	96	**0.403**	0.293	0.406	0.295	0.414	0.297	0.448	0.329	0.419	0.298	**0.403**	**0.289**	0.719	0.416	0.639	0.400	0.672	0.405	1.412	0.802	1.643	0.855	1.157	0.636
	192	**0.412**	**0.295**	0.416	0.300	0.426	0.301	0.487	0.360	0.434	0.305	0.415	0.296	0.748	0.428	0.637	0.416	0.727	0.424	1.419	0.806	1.641	0.854	1.207	0.661
	336	0.427	0.316	0.430	0.309	0.434	**0.303**	0.514	0.372	0.449	0.313	**0.426**	0.304	0.853	0.471	0.655	0.427	0.749	0.454	1.443	0.815	1.711	0.878	1.334	0.713
	720	0.469	0.325	**0.467**	**0.324**	0.487	0.337	0.532	0.383	0.484	0.336	0.474	0.331	1.485	0.825	0.722	0.456	0.847	0.499	1.539	0.837	2.660	1.157	1.292	0.726
	Avg	**0.427**	0.307	0.429	0.307	0.440	0.310	0.495	0.361	0.447	0.313	0.430	**0.305**	0.951	0.535	0.663	0.425	0.749	0.446	1.453	0.815	1.914	0.936	1.248	0.684
ETTh1	96	0.481	0.474	0.720	0.533	**0.458**	**0.456**	0.790	0.586	0.492	0.495	0.516	0.485	0.861	0.628	0.512	0.499	0.613	0.552	0.918	0.639	1.112	0.806	1.298	0.838
	192	**0.518**	**0.491**	0.747	0.545	0.570	0.516	0.837	0.609	0.565	0.538	0.598	0.524	0.797	0.593	0.624	0.555	0.722	0.598	0.915	0.629	1.155	0.823	1.322	0.854
	336	0.664	0.570	0.793	0.551	**0.608**	**0.535**	0.780	0.575	0.721	0.622	0.657	0.550	0.941	0.648	0.691	0.574	0.750	0.619	0.939	0.644	1.179	0.832	1.347	0.870
	720	**0.711**	**0.584**	0.880	**0.584**	0.725	0.591	1.234	0.811	0.986	0.743	0.762	0.610	0.877	0.641	0.728	0.614	0.721	0.616	0.887	0.645	1.273	0.874	1.534	0.947
	Avg	0.593	0.529	0.785	0.553	**0.590**	**0.525**	0.910	0.860	0.691	0.600	0.633	0.542	0.869	0.628	0.639	0.561	0.702	0.596	0.915	0.639	1.180	0.834	1.375	0.877
ETTh2	96	0.354	0.400	0.334	0.381	**0.331**	**0.374**	0.404	0.435	0.357	0.411	0.353	0.389	0.378	0.409	0.382	0.416	0.413	0.451	0.389	0.411	0.678	0.619	2.022	1.006
	192	**0.401**	0.423	0.430	0.438	0.402	**0.411**	0.470	0.474	0.569	0.519	0.403	0.414	0.490	0.467	0.478	0.474	0.474	0.477	0.473	0.455	0.785	0.666	2.329	1.104
	336	0.442	0.450	0.449	0.458	**0.406**	**0.433**	0.489	0.485	0.671	0.572	0.426	0.441	0.537	0.494	0.504	0.501	0.547	0.543	0.477	0.472	0.839	0.694	2.453	1.122
	720	0.480	0.486	0.485	0.490	**0.449**	**0.464**	0.593	0.538	0.824	0.648	0.477	0.480	0.510	0.491	0.499	0.509	0.516	0.523	0.507	0.480	1.273	0.874	3.816	1.407
	Avg	0.419	0.439	0.424	0.441	**0.397**	**0.421**	0.489	0.483	0.605	0.538	0.415	0.431	0.479	0.465	0.466	0.475	0.488	0.499	0.462	0.455	0.894	0.713	2.655	1.160
ETTm1	96	0.388	0.401	0.412	0.422	0.390	0.404	0.709	0.556	**0.352**	**0.392**	0.410	0.419	0.583	0.501	0.578	0.518	0.774	0.614	0.761	0.568	0.911	0.688	0.921	0.682
	192	0.422	0.421	0.447	0.438	0.429	0.423	0.717	0.548	**0.382**	**0.412**	0.437	0.434	0.630	0.528	0.617	0.546	0.754	0.592	0.781	0.574	0.955	0.703	0.957	0.701
	336	0.456	**0.430**	0.497	0.465	0.469	0.439	0.735	0.575	**0.419**	0.434	0.476	0.454	0.725	0.568	0.998	0.775	0.869	0.677	0.803	0.587	0.991	0.719	0.998	0.716
	720	0.554	0.490	0.594	0.521	0.569	0.498	0.752	0.584	**0.490**	**0.477**	0.681	0.556	0.769	0.549	0.693	0.579	0.810	0.630	0.844	0.581	1.062	0.747	1.007	0.719
	Avg	0.455	0.435	0.487	0.461	0.464	0.441	0.728	0.565	**0.411**	**0.429**	0.501	0.466	0.677	0.537	0.722	0.605	0.802	0.628	0.797	0.578	0.980	0.714	0.971	0.705
ETTm2	96	0.192	0.274	0.224	0.296	**0.188**	**0.269**	0.245	0.322	0.213	0.303	0.191	0.274	0.212	0.285	0.291	0.399	0.352	0.454	0.229	0.308	0.331	0.430	0.813	0.688
	192	**0.246**	0.313	0.260	0.317	0.251	**0.309**	0.274	0.338	0.278	0.345	0.252	0.317	0.270	0.323	0.307	0.379	0.694	0.691	0.291	0.343	0.400	0.464	1.008	0.768
	336	**0.301**	**0.340**	0.312	0.349	0.307	0.346	0.361	0.394	0.338	0.385	0.306	0.353	0.323	0.353	0.543	0.559	2.408	1.407	0.348	0.376	0.469	0.498	1.031	0.775
	720	**0.400**	**0.403**	0.424	0.416	0.426	0.417	0.467	0.442	0.436	0.440	0.433	0.427	0.474	0.449	0.712	0.614	1.913	1.166	0.461	0.438	0.589	0.557	1.096	0.791
	Avg	**0.284**	**0.332**	0.305	0.344	0.293	0.335	0.336	0.373	0.316	0.368	0.296	0.343	0.320	0.353	0.463	0.488	1.342	0.930	0.332	0.366	0.447	0.487	0.987	0.756

Table 8: Detailed few-shot learning results on 10% training data

controlled by the scaling factor λ.

F. Visualization

In this section, we provide the visualizations of the forecasting cases of S^2IP-LLM on ETTm2, Electricity, and Weather datasets under the input-512-predict-96 setting. As shown in Figure 7, S^2IP-LLM achieves exceptionally good forecasting results across various datasets.

Methods		**S^{2}IP-LLM**		**Time-LLM(G)**		**OFA**		**iTrans.**		**Dlinear**		**PatchTST**		**TimesNet**		**FEDformer**		**Autoformer**		**Stationary**		**ETSformer**		**LightTS**	
Data.	Hori.	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	0.175	0.228	0.176	0.230	0.175	0.230	0.264	0.307	0.184	0.242	**0.171**	**0.224**	0.207	0.253	0.229	0.309	0.227	0.299	0.215	0.252	0.218	0.295	0.230	0.285
	192	**0.225**	**0.271**	0.226	0.275	0.227	0.276	0.284	0.326	0.228	0.283	0.230	0.277	0.272	0.307	0.265	0.317	0.278	0.333	0.290	0.307	0.294	0.331	0.274	0.323
	336	0.282	**0.321**	0.292	0.325	0.286	0.322	0.323	0.349	**0.279**	0.322	0.294	0.326	0.313	0.328	0.353	0.392	0.351	0.393	0.353	0.348	0.359	0.398	0.318	0.355
	720	**0.361**	**0.371**	0.364	0.375	0.366	0.379	0.366	0.375	0.364	0.388	0.384	0.387	0.400	0.385	0.391	0.394	0.387	0.389	0.452	0.407	0.461	0.461	0.401	0.418
	Avg	**0.260**	**0.297**	0.264	0.301	0.263	0.301	0.309	0.339	0.263	0.308	0.269	0.303	0.298	0.318	0.309	0.353	0.310	0.353	0.327	0.328	0.333	0.371	0.305	0.345
Electricity	96	0.148	0.248	0.148	0.248	**0.143**	**0.241**	0.162	0.264	0.150	0.251	0.145	0.244	0.315	0.389	0.235	0.322	0.297	0.367	0.484	0.518	0.697	0.638	0.639	0.609
	192	**0.159**	**0.255**	0.160	0.257	**0.159**	**0.255**	0.180	0.278	0.163	0.263	0.163	0.260	0.318	0.396	0.247	0.341	0.308	0.375	0.501	0.531	0.718	0.648	0.772	0.678
	336	**0.175**	**0.271**	0.183	0.282	0.179	0.274	0.207	0.305	**0.175**	0.278	0.183	0.281	0.340	0.415	0.267	0.356	0.354	0.411	0.574	0.578	0.758	0.667	0.901	0.745
	720	0.235	0.326	0.236	0.329	0.233	0.323	0.258	0.339	**0.219**	**0.311**	0.233	0.323	0.635	0.613	0.318	0.394	0.426	0.466	0.952	0.786	1.028	0.788	1.200	0.871
	Avg	0.179	0.275	0.181	0.279	0.178	**0.273**	0.201	0.296	**0.176**	0.275	0.181	0.277	0.402	0.453	0.266	0.353	0.346	0.404	0.627	0.603	0.800	0.685	0.878	0.725
Traffic	96	0.410	0.288	0.414	0.293	0.419	0.298	0.431	0.312	0.427	0.304	**0.404**	**0.286**	0.854	0.492	0.670	0.421	0.795	0.481	1.468	0.821	1.643	0.855	1.157	0.636
	192	0.416	0.298	0.419	0.300	0.434	0.305	0.456	0.326	0.447	0.315	**0.412**	**0.294**	0.894	0.517	0.653	0.405	0.837	0.503	1.509	0.838	1.856	0.928	1.688	0.848
	336	**0.435**	0.313	0.438	0.315	0.449	0.313	0.465	0.334	0.478	0.333	0.439	**0.310**	0.853	0.471	0.707	0.445	0.867	0.523	1.602	0.860	2.080	0.999	1.826	0.903
	720	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Avg	0.420	0.299	0.423	0.302	0.434	0.305	0.450	0.324	0.450	0.317	**0.418**	**0.296**	0.867	0.493	0.676	0.423	0.833	0.502	1.526	0.839	1.859	0.927	1.557	0.795
ETTh1	96	**0.500**	**0.493**	0.732	0.556	0.543	0.506	0.808	0.610	0.547	0.503	0.557	0.519	0.892	0.625	0.593	0.529	0.681	0.570	0.952	0.650	1.169	0.832	1.483	0.910
	192	0.690	**0.539**	0.872	0.604	0.748	0.580	0.928	0.658	0.720	0.604	0.711	0.570	0.940	0.665	**0.652**	0.563	0.725	0.602	0.943	0.645	1.221	0.853	1.525	0.930
	336	0.761	0.620	1.071	0.721	0.754	0.595	1.475	0.861	0.984	0.727	0.816	0.619	0.945	0.653	**0.731**	**0.594**	0.761	0.624	0.935	0.644	1.179	0.832	1.347	0.870
	720	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Avg	**0.650**	**0.550**	0.891	0.627	0.681	0.560	1.070	0.710	0.750	0.611	0.694	0.569	0.925	0.647	0.658	0.562	0.722	0.598	0.943	0.646	1.189	0.839	1.451	0.903
ETTh2	96	**0.363**	**0.409**	0.399	0.420	0.376	0.421	0.397	0.427	0.442	0.456	0.401	0.421	0.409	0.420	0.390	0.424	0.428	0.468	0.408	0.423	0.678	0.619	2.022	1.006
	192	**0.375**	**0.411**	0.487	0.479	0.418	0.441	0.438	0.445	0.617	0.542	0.452	0.455	0.483	0.464	0.457	0.465	0.496	0.504	0.497	0.468	0.845	0.697	3.534	1.348
	336	**0.403**	**0.421**	0.858	0.660	0.408	0.439	0.631	0.553	1.424	0.849	0.464	0.469	0.499	0.479	0.477	0.483	0.486	0.496	0.507	0.481	0.905	0.727	4.063	1.451
	720	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Avg	**0.380**	**0.413**	0.581	0.519	0.400	0.433	0.488	0.475	0.694	0.577	0.827	0.615	0.439	0.448	0.463	0.454	0.441	0.457	0.470	0.489	0.809	0.681	3.206	1.268
ETTm1	96	0.357	0.390	0.422	0.424	0.386	0.405	0.589	0.510	**0.332**	**0.374**	0.399	0.414	0.606	0.518	0.628	0.544	0.726	0.578	0.823	0.587	1.031	0.747	1.048	0.733
	192	0.432	0.434	0.448	0.440	0.440	0.438	0.703	0.565	**0.358**	**0.390**	0.441	0.436	0.681	0.539	0.666	0.566	0.750	0.591	0.844	0.591	1.087	0.766	1.097	0.756
	336	0.440	0.442	0.519	0.482	0.485	0.459	0.898	0.641	**0.402**	**0.416**	0.499	0.467	0.786	0.597	0.807	0.628	0.851	0.659	0.870	0.603	1.138	0.787	1.147	0.775
	720	0.593	0.521	0.708	0.573	0.577	0.499	0.948	0.671	**0.511**	**0.489**	0.767	0.587	0.796	0.593	0.822	0.633	0.857	0.655	0.893	0.611	1.245	0.831	1.200	0.799
	Avg	0.455	0.446	0.524	0.479	0.472	0.450	0.784	0.596	**0.400**	**0.417**	0.526	0.476	0.717	0.561	0.730	0.592	0.796	0.620	0.857	0.598	1.125	0.782	1.123	0.765
ETTm2	96	**0.197**	**0.278**	0.225	0.300	0.199	0.280	0.265	0.339	0.236	0.326	0.206	0.288	0.220	0.299	0.229	0.320	0.232	0.322	0.238	0.316	0.404	0.485	1.108	0.772
	192	**0.254**	0.322	0.275	0.334	0.256	**0.316**	0.310	0.362	0.306	0.373	0.264	0.324	0.311	0.361	0.394	0.361	0.291	0.357	0.298	0.349	0.479	0.521	1.317	0.850
	336	**0.315**	**0.350**	0.339	0.371	0.318	0.353	0.373	0.399	0.380	0.423	0.334	0.367	0.338	0.366	0.378	0.427	0.478	0.517	0.353	0.380	0.552	0.555	1.415	0.879
	720	**0.421**	**0.421**	0.464	0.441	0.460	0.436	0.478	0.454	0.674	0.583	0.454	0.432	0.509	0.465	0.523	0.510	0.553	0.538	0.475	0.445	0.701	0.627	1.822	0.984
	Avg	**0.296**	**0.342**	0.325	0.361	0.308	0.346	0.356	0.388	0.399	0.426	0.314	0.352	0.344	0.372	0.381	0.404	0.388	0.433	0.341	0.372	0.534	0.547	1.415	0.871

Table 9: Detailed few-shot learning results on 5% training data.'-' means 5% data is not sufficient to constitute a training set.

Ablation Setting	Long-term Forecasting			
	ETTh2-96	ETTh2-192	ETTm2-96	ETTm2-192
w/o Prompt & Alignment and w/o Decomposition	0.289	0.358	0.170	0.231
w/ Prompt & Alignment and w/o Decomposition	0.287	0.353	0.166	0.228
w/ Prompt & Alignment and w/ Decomposition	0.278	0.346	0.165	0.222

Table 10: Ablation studies on ETTh2 and ETTm2 in predicting 96 and 192 steps (MSE reported).

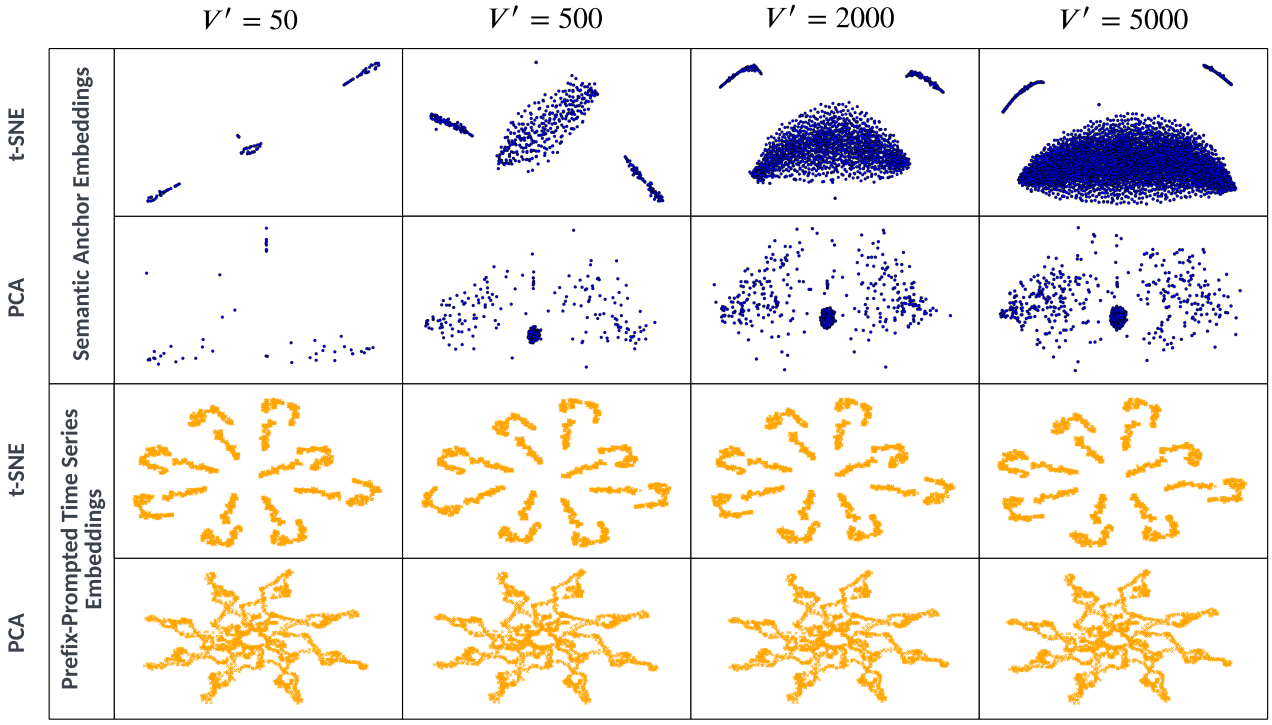

Figure 6. The t-SNE and PCA plots of semantic anchor and prefix-prompted time series embeddings with different V'

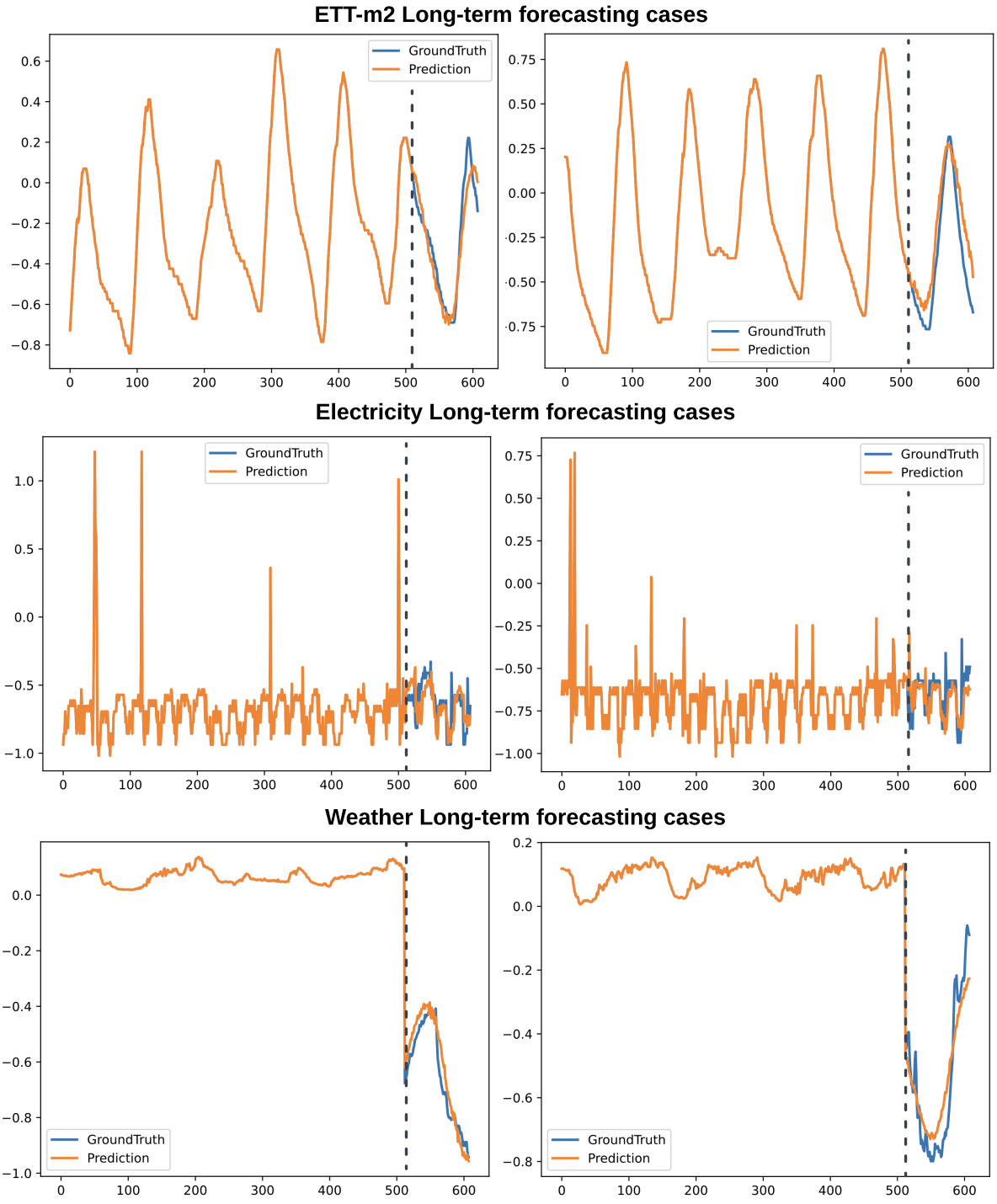

Figure 7. Long-term forecasting visualization cases for ETTm2, Electricity, and Weather. Blue lines are the ground truths and orange lines are the model predictions. The vertical line indicates where the prediction starts.