

# Compositional Spectral Prompts for LLM-based Online Time Series Forecasting

Seungyeon Choi  
KAIST

Daejeon, Republic of Korea  
csyeon08@kaist.ac.kr

Jae-Gil Lee  
KAIST

Daejeon, Republic of Korea  
jaegil@kaist.ac.kr

Hyunchul Kim  
KAIST

Daejeon, Republic of Korea  
khchul@kaist.ac.kr

Chanyoung Park  
KAIST

Daejeon, Republic of Korea  
cy.park@kaist.ac.kr

## Abstract

To address the sequential and evolving nature of time series, the Online Time Series Forecasting (OTSF) task has been extensively studied in multiple domains. Existing research focuses on adapting to non-stationary environments by employing memory buffer-based retrieval strategies. However, we observe that such frameworks struggle with long-term adaptation and fail to generalize to unseen patterns. To this end, we introduce CoSPOT, an LLM-based online time series forecasting framework that leverages a pre-trained LLM as the backbone online forecaster, motivated by its strong few-shot capabilities. For efficient online adaptation, CoSPOT keeps the LLM frozen and employs compositional spectral prompts grounded in frequency-domain bases to guide the model with the overall distribution of the input, thereby substantially reducing the number of parameters updated during the online phase. Specifically, CoSPOT decomposes time series into frequency bases and composes the corresponding spectral basis prompts according to their amplitudes, allowing unseen patterns to be represented as new combinations of learned basis prompts. Our extensive experiments on real-world datasets demonstrate the superiority and practicality of CoSPOT across challenging online scenarios, including extended online phases and cross-dataset settings with substantial distribution shifts.

## Keywords

Time Series Forecasting, Online Learning, Prompt Learning, Large Language Models

## ACM Reference Format:

Seungyeon Choi, Hyunchul Kim, Jae-Gil Lee, and Chanyoung Park. 2026. Compositional Spectral Prompts for LLM-based Online Time Series Forecasting. In *Proceedings of Machine Learning in Time Series (MILETS) Workshop at ACM SIGKDD Conference on Knowledge Discovery and Data Mining (MILETS@KDD '26)*, August 09–13, 2026, Jeju Island, Republic of Korea. ACM, New York, NY, USA, 9 pages.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

MILETS '26, Jeju Island, Republic of Korea

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

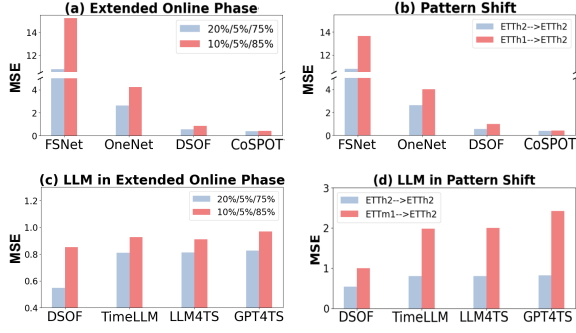
## 1 Introduction

Early research in deep learning-based time series forecasting [12, 19, 20, 24, 25, 27] primarily focused on batch learning methods utilizing static training and evaluation datasets. However, given the sequential and evolving nature of time series data, shifts in underlying patterns over time are inevitable. Consequently, traditional batch learning approaches often fail to adapt to such changes, while frequent model retraining to accommodate new patterns is both labor-intensive and impractical for real-world applications. To address these challenges, online learning paradigms, which enable models to incrementally update as new data arrive in dynamic environments, have been increasingly explored.

Existing studies on online time series forecasting (OTSF) [10, 13, 18] have focused on effective adaptation to evolving data streams. The pioneering work, FSNet [13], addressed rapid adaptation and pattern reuse by employing lightweight per-layer adapters and an associative memory for pattern retrieval. Building on this, OneNet [18] introduced reinforcement learning to dynamically ensemble models based on their real-time performance. Furthermore, DSOF [10] identified the information leakage issue in prior setups and proposed a dual-stream (i.e., teacher-student) residual framework to handle delayed adaptation. These methods share a strategy of employing memory buffer-based retrieval to adapt to non-stationary environments.

Despite the recent advancements, the buffer-based design of existing methods faces two fundamental challenges in practical OTSF scenarios. **(1) Inadaptability to extended online phases.** As new data is continuously streamed in online scenarios, models require sufficient capacity for long-term adaptation to effectively learn from evolving patterns. As shown in Figure 1 (a), increasing the proportion of the online phase (i.e., test data) leads to significant performance deterioration in previous approaches<sup>1</sup>. This degradation stems from the inherent capacity constraints of memory buffers: as the online period lengthens, the increasing diversity of recurring patterns makes it difficult to explicitly retain them all in the associative memory, leading to catastrophic forgetting. **(2) Inadaptability to unseen patterns.** In online forecasting scenarios, previously unseen patterns (i.e., distribution shifts in pattern) may

<sup>1</sup>To create a more challenging and extended online phase within the given dataset, we deviate from the typical 20%/5%/75% split used in prior studies and instead adopt a 10%/5%/85% train/validation/test split.



**Figure 1: (a) Performance of prior methods and our proposed method (i.e., CoSPOT) under an extended online phase. (b) Performance of prior methods and CoSPOT in a cross-dataset scenario with distribution shifts in pattern. (c & d) Performance of DSOF and LLM-based time series forecasting methods under an extended online phase and cross-dataset scenario. Note that the ETTh2 dataset is used for (a) and (c).**

emerge during test time (i.e., online phase). Thus, models must possess adaptability to such shifts for effective adjustments. Figure 1(b) shows experimental results obtained under a setting where the time series patterns in the training and online phases are intentionally made different. Specifically, we used two datasets from the same domain (i.e., ETTh1 and ETTh2) and conducted experiments under two scenarios: One where model training and online updates are both performed on ETTh2 (in blue), and the other where models are initially trained on ETTh1 dataset, then updated online as ETTh2 data streams in (in red). We observe that existing OTSF methods degrade substantially on patterns unseen during training, as they rely on associative memory to retrieve patterns similar to the new input. In other words, when an unprecedented pattern appears, retrieval from previously learned patterns becomes unreliable, leading to failure in adaptation under distribution shifts.

To address the challenges of extended online phases and adaptability to unseen patterns, we propose CoSPOT, a Large Language Model (LLM)-based online time series forecasting framework built upon compositional spectral prompting. Our design is motivated by the observation that integrating a pre-trained LLM is pivotal for maintaining adaptability over extended online phases. Recent studies [3, 8, 28] have shown that LLMs can significantly enhance forecasting performance when aligned with time series tasks, particularly in *few-shot* setting. Such transferability of LLMs is especially advantageous in OTSF, as it enables effective adaptation under limited data. In Figure 1 (c), we compare the state-of-the-art OTSF model that relies solely on a time series model (i.e., DSOF) against those that align the time series backbone with LLMs (i.e., TimeLLM [8], LLM4TS [3], and GPT4TS [28]) under an extended online phase. Surprisingly, we found that although these LLM-based models were not originally designed for online learning<sup>2</sup>, they demonstrate performance comparable to DSOF as the online phase lengthens. However, despite their effectiveness in extended online phases, LLM-based models struggle to generalize effectively

to pattern shifts (Figure 1 (d)), limiting their applicability in OTSF. This deficiency underscores the need for continuous adaptation of the model’s representational capacity. However, updating the LLM itself is impractical for online scenarios, as the large number of parameters hinders rapid adaptation. To address this, we propose a novel prompting mechanism that alleviates this burden by keeping the LLM frozen, while leveraging a trainable prompt to guide the model’s response to evolving patterns. To further enhance the model’s generalizability, CoSPOT grounds its prompts in frequency-domain representations, which we refer to as spectral prompts. This distinguishes our approach from prior OTSF methods that rely on time-domain representations [22, 27], as frequency features are superior at capturing underlying periodic structures. Specifically, we decompose the input time series into frequency components using the Discrete Fourier Transform (DFT). Each universal frequency basis, which serves as a fundamental building block of time series patterns, is associated with a learnable spectral basis prompt. These spectral basis prompts are then compositionally aggregated according to the amplitudes of the corresponding frequency components, forming an input-specific compositional spectral prompt fed to the LLM. This design effectively transforms the adaptation problem: instead of memorizing infinitely many specific patterns, the model represents any unseen pattern as a novel composition of spectral basis prompts that are pre-learned and frozen during the online phase. Consequently, our framework achieves two key advantages: (i) *Generalization*: Newly emerged patterns are represented by recombining spectral basis prompts associated with frequency bases, enabling robust adaptation to distribution shifts. (ii) *Efficiency*: Since the spectral prompt bank and the LLM are frozen during the online phase, CoSPOT achieves inductive adaptation without the computational overhead of large-scale parameter updates.

In this study, we make the following contributions:

- We identify that existing memory buffer-based OTSF methods struggle with long-term adaptation and fail to generalize to unseen patterns.
- We propose CoSPOT, the first approach to integrate LLMs into OTSF. CoSPOT employs compositional spectral prompting, which constructs input-specific prompts by composing spectral basis prompts grounded in frequency-domain bases.
- Through extensive experiments under various online learning scenarios, we demonstrate that CoSPOT consistently outperforms state-of-the-art OTSF methods with extremely few online parameter updates.

## 2 Related Works

**Online Time Series Forecasting.** Given the evolving nature of time series data, online forecasting has gained prominence for practical applications [2, 6, 9]. Recently, online deep learning models have been proposed to further capture complex patterns within time series data. FSNet [13] introduces calibration module to dynamically balance fast adaptation to recent changes with the retention of prior knowledge. OneNet [18] incorporates reinforcement learning to model cross-variable and cross-time concept drifts. Addressing the information leakage issue in previous research, DSOF [10] redefines the OTSF setting and proposes a dual-stream mechanism to update model parameters. Nevertheless, prior studies do not explicitly

<sup>2</sup>We adhere to their original training protocols and fine-tune only the output projection in the online phase.

model the patterns in input signals, hindering their ability to adapt to unobserved distributions. Additionally, their explicit storage of pattern information restricts the model’s adaptability to extended online phases.

**Time Series Forecasting with LLMs.** Recent advancements in LLMs have prompted researchers to investigate their transferability to forecasting tasks in data-sparse time series domains. LLM4TS [3] introduces two-stage fine-tuning approach to leverage LLMs for time series forecasting. GPT4TS [28] retrains the positional embeddings and normalization layers of LLMs to preserve pre-trained knowledge while enhancing performance on downstream tasks. Additionally, TimeLLM [8] employs reprogramming method to align time series data with word embeddings. Inspired by the proven adaptability of these models, we leverage LLMs to enable rapid adjustments in online scenarios.

**Prompt-based Continual Learning.** Rehearsal-free continual learning methods leverage the strong general representations of pre-trained models like ViT [5]. By fine-tuning only small, learnable *prompts* for each task, these methods achieve significant memory and computational efficiency, as the core model parameters remain unchanged. VPT [7] optimizes a single prompt, L2P [17] uses a shared pool of prompts. S-Prompts [16], train a unique prompt for each individual task to address catastrophic forgetting. However, the application of prompt learning to address distribution shifts in the time series domain remains unexplored. Given the continuous nature of time series data, the online learning scenario is more suitable than continual learning, which assumes distinct tasks. CoSPOT is the first to achieve an efficient and scalable prompting crucial for online learning scenarios by encoding knowledge from the underlying frequency bases.

**Frequency Analysis in Time Series Forecasting.** Due to the complex temporal variations in time series data, frequency analysis techniques such as the Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT) are widely used to capture recurring patterns. DFT analyzes global frequency components, while DWT provides localized frequency information at different scales. FEDformer [27] is a representative frequency-domain forecasting model that leverages DFT and DWT through Fourier Enhanced Structure and Wavelet Enhanced Structure, respectively. However, such frequency-domain approaches mainly exploit frequency coefficients without learning prompt-level knowledge associated with each frequency basis, limiting robustness to unseen patterns under distribution shifts. In contrast, CoSPOT grounds learnable spectral basis prompts in frequency-domain bases and composes them according to the input spectrum, allowing unobserved patterns to be represented as new combinations of learned prompts and improving adaptability in online scenarios.

### 3 Preliminaries

#### 3.1 Time Series Forecasting

Let  $\mathbf{X} = (x_1, \dots, x_{N_{data}}) \in \mathbb{R}^{N_{data} \times n}$  be the entire time series with  $N_{data}$  observations, where each observation  $x_i \in \mathbb{R}^n$  contains  $n$  dimensions. The dataset  $\mathbf{X}$  is then partitioned into  $N_{train}$ ,  $N_{val}$ , and,  $N_{online}$  time stamps according to predefined ratios, maintaining the chronological order of the data. Given the look-back window of

length  $L$ , denoted as  $\mathbf{X}_{i-L+1:i} = (x_{i-L+1}, x_{i-L+2}, \dots, x_i)$ , the objective of time series forecasting is to predict the following  $H$  steps (i.e.,  $\mathbf{X}_{i+1:i+H}$ ), where the model’s prediction at  $t = i$  for the next  $H$  steps is denoted by  $\hat{\mathbf{X}}_{i+1:i+H} = (\hat{x}_{i+1}, \hat{x}_{i+2}, \dots, \hat{x}_{i+H}) = f(\mathbf{X}_{i-L+1:i})$ . The objective is to minimize the mean squared error (MSE) between the ground truth and the predicted outputs, i.e.,  $\frac{1}{H} \sum_{h=1}^H \|\hat{x}_{i+h} - x_{i+h}\|_2^2$ .

#### 3.2 Online Time Series Forecasting

The OTSF scenario consists of two phases: training phase and online phase. In the training phase, the entire  $N_{train}$  time series are utilized to create  $(L + H)$  sized time sequences. The objective of the training phase is to let the model to learn the base knowledge through static batch training strategy. The online phase follows the training phase, where  $N_{online}$  time steps are streamed sequentially with a moving window of size 1. This mirrors real-world scenarios, and the model is updated in real-time.

**Objective and Evaluation Criterion.** Our ultimate goal is to accurately predict the ground truth by minimizing the cumulative MSE between the ground truth and predicted values over the entire prediction horizon of  $H$  steps, using  $MSE_{online}$  to evaluate performance as follows:

$$MSE_{online} = \frac{1}{N_{online} - L - H + 1} \sum_{i=N_{train}+N_{val}+L}^{N_{data}-H} \|f(\mathbf{X}_{i-L+1:i}) - \mathbf{X}_{i+1:i+H}\|_2^2. \quad (1)$$

### 4 Proposed Method: CoSPOT

In this section, we introduce our proposed method CoSPOT. The key components of this framework are as follows: (1) leveraging a pre-trained LLM together with a time series backbone and textual recent information to enhance adaptability in data-scarce online scenarios (Sec. 4.1), and (2) compositional spectral prompting, which composes frequency-grounded spectral basis prompts to robustly and efficiently adapt to pattern shifts (Sec. 4.2). Overall framework of CoSPOT is shown in Figure 2.

#### 4.1 Enhancing model adaptability using a pre-trained LLM

To enable effective and rapid adaptation in data-scarce online scenarios, CoSPOT leverages a pre-trained LLM together with a time-series backbone. The LLM remains frozen throughout the training and online phases, serving as a stable knowledge source, while the time-series backbone encodes the input sequence. This design exploits the LLM’s transferability to support adaptation when only limited online observations are available. However, online forecasting also requires contextual information about recent dynamics, which may not be sufficiently captured from numerical inputs alone. To this end, we introduce a text description as an additional modality, allowing recent contextual cues to be provided to the LLM in natural language. Given the input time series  $\mathbf{X}$ , the text description (i.e.,  $text_{\mathbf{X}}$ ) contains task details, dataset information, recent values, and recent frequency information. This text description provides short-term contextual cues, complementing the compositional spectral prompt in Section 4.2, which captures the overall distribution of the input. The recent values summarize the latest time-domain variation, while the recent frequency information captures local

frequency changes near the current time point. Although Short-Time Fourier Transform (STFT) can provide localized frequency information, its fixed window size limits its ability to capture non-stationary recent dynamics due to the trade-off between time and frequency resolution. Hence, we use the Discrete Wavelet Transform (DWT), whose adaptive time-frequency resolution is better suited for capturing recent variations in non-stationary signals. Given the input time series  $\mathbf{X}$ , the equation of DWT using a scaling function  $\phi$  and a wavelet function  $\psi$  is as follows:

$$A_j[k] = \sum_n \mathbf{X}[n] \phi_{j,k}[n], \quad D_j[k] = \sum_n \mathbf{X}[n] \psi_{j,k}[n], \quad (2)$$

where  $\mathbf{X}[n]$  refers to the  $n$ -th index in the time series  $\mathbf{X}$ ,  $A_j[k]$  and  $D_j[k]$  refer to the approximation and detail coefficient at level  $j$ , respectively, and  $\phi_{j,k}[n]$  and  $\psi_{j,k}[n]$  are the scaling and wavelet functions at level  $j$ , respectively. After the decomposition, the time series is passed through a filter bank that separates the low-pass and high-pass components, and downsampling is performed as follows:

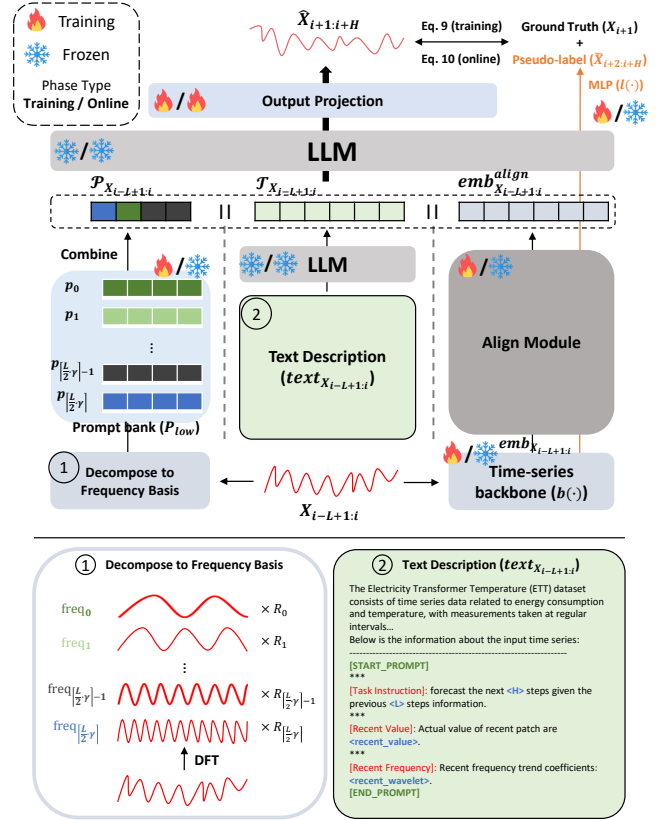
$$A_{j+1}[k] = \sum_n h[n-2k] A_j[n], \quad D_{j+1}[k] = \sum_n g[n-2k] A_j[n], \quad (3)$$

where  $h[n]$  and  $g[n]$  refer the low-/high-pass filters, respectively. Through this process, we utilize  $A_j[-1]$  to provide the model with the most recent frequency information where  $j$  is a hyperparameter.

An example of the text description is shown in Figure 2. The text description is first processed through the pre-trained LLM's tokenizer. The resulting token IDs are then passed through the frozen LLM's input embedding layer to retrieve their corresponding token embeddings. This sequence of text token embeddings is the resulting embedding (denoted as  $\mathcal{T}_{\mathbf{X}}$ ) that is used as input for the final prediction. Providing text descriptions enriches data in data-scarce online scenarios, offering recent information that aids effective adaptation, all without requiring additional training.

## 4.2 Robust Adaptation via Compositional Spectral Prompting

While the frozen LLM integrated with text descriptions provides a strong foundation for adaptability, it still faces challenges when encountering completely unseen patterns that deviate significantly from the patterns learned during the training phase. However, updating the LLM itself is impractical for online scenarios, as the large number of parameters hinders rapid adaptation. Therefore, we need an efficient mechanism to provide the frozen model with inductive bias about these evolving structures. To address this, we introduce *compositional spectral prompting*, which provides the frozen model with reliable guidance on the overall data distribution. Here, *spectral* indicates that the prompts are grounded in the frequency-domain structure of time series. We leverage the frequency domain because, compared with the time domain, it more effectively isolates underlying pattern components that are often entangled in raw temporal signals, making it well suited for representing evolving time-series patterns. Instead of directly storing complex patterns, which renders prior memory-based methods [10, 13, 18] unreliable when facing unseen patterns, CoSPOT decomposes time series into frequency bases and represents each input as a composition of learnable spectral basis prompts. This design allows newly emerging patterns to be expressed through new combinations of pre-learned prompts, without updating the LLM during the online phase. Specifically, we



**Figure 2: Overall model framework.** Given the input time series  $\mathbf{X}_{i-L+1:i}$ , the aligned embedding (i.e.,  $emb_{\mathbf{X}_{i-L+1:i}}$ ), embedded text description (i.e.,  $\mathcal{T}_{\mathbf{X}_{i-L+1:i}}$ ), and compositional spectral prompt (i.e.,  $\mathcal{P}_{\mathbf{X}_{i-L+1:i}}$ ) are provided as input to the LLM. The representation computed by the LLM is passed through the output projection layer to produce the final prediction, i.e.,  $\hat{\mathbf{X}}_{i+1:i+H}$ .

use the DFT to decompose the input time series  $\mathbf{X}$  of length  $L$ . The DFT converts the sequence from the time domain to the frequency domain, while its inverse (IDFT) converts it back. Their expressions are as follows:

$$\mathcal{F}(k) = DFT(\mathbf{X}) = \sum_{n=0}^{L-1} \mathbf{X}[n] \exp\left(-i \frac{2\pi kn}{L}\right), \quad k = 0, 1, \dots, L-1, \quad (4)$$

$$\mathbf{X}[n] = IDFT(\mathcal{F}) = \frac{1}{L} \sum_{k=0}^{L-1} \mathcal{F}(k) \exp\left(i \frac{2\pi kn}{L}\right), \quad n = 0, 1, \dots, L-1, \quad (5)$$

where  $\mathcal{F}$  refers to the frequency spectrum of the input and  $i$  represents the imaginary unit. From the perspective of frequency basis, assuming that  $L$  is even, both the DFT and IDFT can be represented using  $\frac{L}{2} + 1$  orthogonal cosine-based frequency basis because the DFT of a real-valued signal exhibits Hermitian symmetry. Thus, the IDFT can be rewritten as follows:

$$\begin{aligned} X[n] &= \frac{1}{L} \sum_{k=0}^{\frac{L}{2}} \left( R_k \cdot \cos\left(\frac{2\pi kn}{L} - \phi\right) \right) \\ &= \frac{1}{L} \sum_{k=0}^{\frac{L}{2}} \left( R_k \cdot \text{freq}_k \right), \quad n = 0, 1, \dots, L-1. \end{aligned} \quad (6)$$

where  $\text{freq}_k$  and  $R_k \in \mathbb{R}$  denote the basis of the  $k$ -th frequency and its amplitude, respectively.

Building on this formulation, which reconstructs the time series as a weighted combination of frequency bases, we introduce a spectral prompt bank that associates each frequency basis with a learnable spectral basis prompt. Specifically, each spectral basis prompt encodes the characteristic temporal pattern corresponding to its frequency basis. Let  $\mathbf{P} = [\mathbf{p}_0, \dots, \mathbf{p}_{\frac{L}{2}}] \in \mathbb{R}^{(\frac{L}{2}+1) \times d}$  denote the spectral prompt bank, where  $\mathbf{p}_k$  represents the learnable spectral basis prompt corresponding to the  $k$ -th frequency basis. However, learning knowledge for all frequency bases is not effective in capturing the overall pattern of the given time series. That is, high frequencies represent rapidly oscillating periodicities compared to low frequencies, and therefore, they do not capture the overall pattern information. Hence, in time series analysis, high frequencies are often treated as noise, which is why low-pass filtering techniques [21, 26] are widely used. Accordingly, to effectively capture the overall pattern while removing noise, we introduce a hyperparameter  $\gamma \in [0, 1]$  to eliminate the high-frequency bases, i.e.,  $\mathbf{P}_{low} = [\mathbf{p}_0, \dots, \mathbf{p}_{\lfloor \frac{L}{2} \cdot \gamma \rfloor}] \in \mathbb{R}^{(\lfloor \frac{L}{2} \cdot \gamma \rfloor + 1) \times d}$ . We then construct an input-specific compositional spectral prompt by weighting each spectral basis prompt according to the amplitude of its corresponding frequency component, which provides the LLM with guidance on the overall distribution (i.e., overall pattern) of the input:

$$\mathcal{P}_X = \text{Concat}\left(\mathbf{R}_0 \cdot \mathbf{p}_0, \mathbf{R}_1 \cdot \mathbf{p}_1, \dots, \mathbf{R}_{\lfloor \frac{L}{2} \cdot \gamma \rfloor} \cdot \mathbf{p}_{\lfloor \frac{L}{2} \cdot \gamma \rfloor}\right) \in \mathbb{R}^{(\lfloor \frac{L}{2} \cdot \gamma \rfloor + 1) \times d}, \quad (7)$$

where  $\mathcal{P}_X$  refers the compositional spectral prompt of the input time series  $X$  to be provided to the model. Through compositional spectral prompting, the model explicitly captures pattern-level information under continuous distribution shifts. Even when previously unseen patterns emerge in the online phase, the model can effectively represent and adapt to them by compositionally recombining the knowledge encoded in spectral basis prompts during the training phase, without requiring any additional training. Moreover, since newly emerging patterns can be expressed as compositions of a finite set of spectral basis prompts, the model maintains its memory efficiency without degradation.

### 4.3 Overall Framework

**Align Module.** In OTSF scenarios, effectively aligning continuous time series data with discrete token-based LLMs is crucial yet challenging. Since pre-trained LLMs lack inherent knowledge of time series patterns, prior studies [3, 8, 28] focus on aligning two modalities (i.e., time series and language) to leverage the knowledge within LLMs, enabling accurate, data-efficient, and task-agnostic forecasting. As the goal of this study is to enhance the LLM's ability in the online scenario, not to focus on the align module itself, we utilize a pre-developed align module [3, 8]. The align module aligns

the representation of the time series computed by the time series backbone (i.e.,  $\text{emb}_X = b(X)$ , where  $b(\cdot)$  is the time series backbone) with the natural language modality, and outputs the resulting aligned embeddings (i.e.,  $\text{emb}_X^{\text{align}}$ ).

**Utilizing a pre-trained LLM.** The input to the LLM is formed by concatenating the compositional spectral prompt (i.e.,  $\mathcal{P}$ ), embedded text description (i.e.,  $\mathcal{T}$ ), and aligned time series embedding (i.e.,  $\text{emb}_X^{\text{align}}$ ). Within the text description, we utilize special tokens (e.g., [START PROMPT] and [END PROMPT]) as delimiters, enabling the LLM to clearly discern the boundaries between these heterogeneous modalities. The concatenated embedding sequence is input to the LLM's transformer layers, all of which remain frozen. The last hidden state representation of the LLM serves as the time series representation, which is flattened and linearly projected to generate the final forecast.

**Training Phase.** The training phase serves to learn the overall base knowledge before entering the online phase. Therefore, during the training phase, except for the pre-trained LLM, we train the time series backbone network, align module, spectral prompt bank, and output projection layer by minimizing the following objective:

$$\mathcal{L}_{\text{training}} = \frac{1}{N_{\text{train}} - L + 1} \sum_{i=L}^{N_{\text{train}}} \|f(X_{i-L+1:i}) - X_{i+1:i+H}\|_2^2. \quad (8)$$

where  $f(\cdot)$  is the overall framework of CoSPOT.

**Online Phase.** In the online phase, only the output projection layer is tuned to match the streaming data distribution, while all other parameters, including the pre-trained LLM, time series backbone network, align module, and spectral prompt bank, remain frozen.<sup>3</sup> According to [10], when the prediction horizon  $H$  is greater than 1, calculating the loss using the ground truth for all  $H$  time steps ( $X_{i+1:i+H}$ ) at each moving window step for model updates leads to information leakage. To avoid this, the moving window should be extended to  $H$  steps rather than updating at each step, which however introduces an update delay and hinders effective adaptation. Hence, we employ a pseudo-labeling technique while keeping updates at each step. When calculating the loss for the model's output at  $t = i$  to update the model, only the ground truth for the immediate next time point, i.e.,  $X_{i+1}$  is used, and pseudo-labels are generated and utilized for the remaining time steps (i.e.,  $\tilde{X}_{i+2:i+H}$ ). We propagate the representation of the time series backbone network (i.e.,  $b(\cdot)$ ) through a linear layer (i.e.,  $l(\cdot)$ ) to project it into the output space and obtain the pseudo-label. This linear layer is utilized while being frozen during the online phase, and is trained along with Equation 8 during the training phase using MSE loss. Therefore, Equation 8 is modified as follows:

$$\begin{aligned} \mathcal{L}_{\text{training}}^+ &= \frac{1}{N_{\text{train}} - L + 1} \sum_{i=L}^{N_{\text{train}}} \left( \|f(X_{i-L+1:i}) - X_{i+1:i+H}\|_2^2 \right. \\ &\quad \left. + \|l(b(X_{i-L+1:i})) - X_{i+1:i+H}\|_2^2 \right). \end{aligned} \quad (9)$$

where  $l(\cdot)$  denotes the pseudo-label projection layer which projects the time series representation into the output space. During the online phase, we use the frozen  $b(\cdot)$  and  $l(\cdot)$ , which are trained in the training phase, to generate pseudo-labels:  $\tilde{X}_{i+2:i+H} = l(b(X_{i-L+1:i})) [1 :$

<sup>3</sup>We emphasize that CoSPOT is an efficient framework despite utilizing an LLM, as the number of parameters updated during the online phase is extremely small. Please see Section 5.3.3 for a detailed analysis.

]. These pseudo-labels are then employed to tune the model. To mitigate the impact of prediction errors and pseudo-labels as the forecast horizon extends from the current observation, we apply a geometric decay factor  $\delta \in [0, 1]$  to the online loss as follows:

$$\mathcal{L}_{\text{online}} = \frac{1}{N_{\text{online}} - L - H + 1} \sum_{i=N_{\text{train}}+N_{\text{val}}+L}^{N_{\text{data}}-H} \left( \frac{1}{H} \sum_{h=1}^H \delta^{h-1} \|\hat{\mathbf{X}}_{i+h} - \bar{\mathbf{X}}_{i+h}\|_2^2 \right). \quad (10)$$

where  $\bar{\mathbf{X}}_{i+1:i+H} = \text{Concat}(\mathbf{X}_{i+1}, \hat{\mathbf{X}}_{i+2:i+H})$  is a concatenated sequence of the ground truth and pseudo-labels. Finally, the model's online performance is evaluated using Equation 1.

## 5 Experiments

**Datasets.** Following prior studies [10, 13, 18], we evaluate our method on five widely used time series forecasting benchmarks from various domains, splitting each into training, validation, and testing sets with a 20%, 5%, and 75% ratio, respectively. **ETT**<sup>4</sup> contains electricity load and oil temperature data collected at 15-minute (ETTM1, ETTM2) and hourly (ETTh1, ETTh2) intervals, each with 6 covariates. **Weather**<sup>5</sup> consists of 21 meteorological variables recorded at hourly intervals, covering diverse climate features such as temperature, humidity, and wind speed. **ECL**<sup>6</sup> comprises hourly electricity consumption records from 321 clients over two years. **Traffic**<sup>7</sup> contains occupancy rate measurements from 862 free-way sensors in the San Francisco Bay Area, recorded at 5-minute intervals. **Exchange Rate**<sup>8</sup> contains daily exchange-rate records across multiple countries. Compared with the other benchmarks, Exchange Rate exhibits less pronounced periodicity and is more influenced by random economic and market events, providing a challenging testbed for evaluating robustness beyond strongly seasonal patterns.

**Baselines.** We utilize various deep learning-based time series forecasting models as baselines, categorized into four groups. *Static forecasting models:* **DLinear** [23] employs patch-wise tokenization with channel-independent linear layers. **PatchTST** [12] applies patch-based Transformer encoding to preserve local temporal patterns. **iTransformer** [11] attends along the feature dimension to capture cross-variable interactions. **TimeMixer** [15] introduces a multiscale-mixing architecture that decomposes and mixes seasonal and trend components across different sampling scales. *Time series foundation models:* **Chronos-2** [1] is a time series foundation model that supports zero-shot univariate, multivariate, and covariate-informed forecasting through a unified architecture. **TimesFM** [4] is a decoder-only foundation model pre-trained on large-scale time series corpora for zero-shot forecasting. *LLM-based models:* **LLM4TS** [3] aligns time series representations with pre-trained LLMs through a two-stage fine-tuning strategy for forecasting. **GPT4TS** [28] repurposes frozen GPT-style Transformers by tuning only lightweight components for time series prediction. **Time-LLM**

[8] reprograms time series into text prototypes with a Prompt-as-Prefix strategy. *Online time series forecasting (OTSF) models:* **FSNet** [13] captures short- and long-term patterns via dual fast and slow learners. **OneNet** [18] adaptively weights an online ensemble based on recent performance. **DSOF** [10] introduces a dual-stream framework that updates parameters through distinct short- and long-term temporal contexts.

**Implementation Details.** Following previous studies [10, 13, 18], we use prediction lengths  $H \in \{1, 24, 48\}$  with a lookback length  $L = 96$ . Unless otherwise specified, PatchTST [12], Llama-7B [14], and the Time-LLM align module [8] are used as the time-series backbone  $b(\cdot)$ , default LLM, and align module, respectively. We evaluate performance using mean square error (MSE) and mean absolute error (MAE). The model is implemented in Python 3.11 and PyTorch 2.2.2, trained with AdamW for 10 epochs, and updated once per instance during the online phase. We set  $\gamma = 0.3$ ,  $j = 2$ , and  $\delta = 0.8$  in Eqs. 7, 3, and 10, respectively. Experiments are conducted on a 48GB NVIDIA RTX A6000.

### 5.1 Overall Performance

The experimental results on eight datasets are summarized in Table 1. The reported results represent averages over three runs. We make the following key observations: (1) In settings without information leakage, FSNet and OneNet, which are designed for OTSF, underperform static models such as DLinear, iTransformer, and TimeMixer. This implies that FSNet and OneNet mainly exploit the leaked data on which they are trained—leading to rapid convergence—rather than truly acquiring the underlying structure of newly arriving patterns or learning how to adapt to them. (2) Time series foundation models, i.e., Chronos-2 and TimesFM, underperform CoSPOT. Although these models exhibit strong zero-shot forecasting ability from large-scale pre-training, they lack an explicit mechanism to adapt to evolving target distributions in online scenarios. In contrast, CoSPOT combines compositional spectral prompts with text descriptions, providing both distribution-aware guidance and recent contextual information for more effective adaptation. (3) LLM-based models, i.e., LLM4TS, GPT4TS, and Time-LLM, substantially underperform CoSPOT. This gap arises because their alignment modules cannot handle continuous distribution shifts, leading to a breakdown in modality alignment. Conversely, CoSPOT successfully adapts to these shifts using compositional spectral prompts and text descriptions, thereby preserving the crucial alignment between the LLM and the time series backbone. (4) While the state-of-the-art OTSF method, DSOF, demonstrates strong performance compared with other baselines, it significantly underperforms CoSPOT. This demonstrates that while the dual-stream framework of DSOF prevents update delays, allowing the model to adapt quickly without information leakage, its emphasis on rapid convergence to incoming data limits its ability to capture underlying patterns, leaving the model poorly equipped to adapt when data are scarce. (5) CoSPOT demonstrates robust performance across multiple datasets by combining the adaptability of a pre-trained LLM with compositional spectral prompts, which provide explicit guidance on the underlying time-series patterns under continuous distribution shifts.

<sup>4</sup><https://github.com/zhouhaoyi/ETDataset>

<sup>5</sup><https://www.ncei.noaa.gov/data/local-climatological-data/>

<sup>6</sup><https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014>

<sup>7</sup><https://pems.dot.ca.gov/>

<sup>8</sup><https://github.com/laiguokun/multivariate-time-series-data>



**Table 1: Comparison of MSE and MAE results in OTSF for predicting 1, 24, and 48 prediction horizon (i.e.,  $H$ ) with a lookback length  $L = 96$  (Best: bold red, the second-best: underlined in blue).**

|          | $H$ | DLinear      |              | PatchTST |       | iTransformer |       | TimeMixer    |       | Chronos-2    |              | TimesFM      |              | LLM4TS |       | GPT4TS |       | Time-LLM |       | FSNet |       | OneNet |              | DSOF         |              | CoSPOT       |              |
|----------|-----|--------------|--------------|----------|-------|--------------|-------|--------------|-------|--------------|--------------|--------------|--------------|--------|-------|--------|-------|----------|-------|-------|-------|--------|--------------|--------------|--------------|--------------|--------------|
|          |     | MSE          | MAE          | MSE      | MAE   | MSE          | MAE   | MSE          | MAE   | MSE          | MAE          | MSE          | MAE          | MSE    | MAE   | MSE    | MAE   | MSE      | MAE   | MSE   | MAE   | MSE    | MAE          | MSE          | MAE          | MSE          | MAE          |
| ETTh1    | 1   | <u>0.502</u> | <u>0.609</u> | 0.779    | 0.828 | 0.993        | 0.976 | 0.557        | 0.706 | 0.579        | 0.681        | 0.621        | 0.708        | 1.436  | 1.058 | 1.539  | 1.141 | 1.382    | 1.076 | 13.26 | 3.441 | 4.023  | 1.956        | 0.802        | 0.857        | <b>0.425</b> | <b>0.601</b> |
|          | 24  | 2.333        | 1.427        | 3.797    | 1.849 | 3.028        | 1.690 | 2.719        | 1.528 | 2.484        | 1.396        | 2.418        | 1.455        | 3.155  | 1.623 | 3.028  | 1.640 | 2.947    | 1.417 | 19.33 | 4.097 | 9.001  | 2.801        | <u>2.311</u> | <u>1.372</u> | <b>1.217</b> | <b>0.993</b> |
|          | 48  | 2.802        | 1.553        | 5.135    | 2.066 | 3.998        | 1.929 | 3.274        | 1.699 | <u>2.748</u> | <u>1.458</u> | 2.994        | 1.630        | 6.256  | 2.312 | 6.738  | 2.397 | 5.997    | 2.049 | 25.82 | 4.881 | 12.21  | 3.334        | 4.233        | 1.973        | <b>1.482</b> | <b>1.073</b> |
| ETTh2    | 1   | 0.468        | 0.643        | 0.895    | 0.924 | 0.874        | 0.888 | <u>0.439</u> | 0.563 | 0.476        | 0.590        | 0.597        | 0.572        | 0.811  | 0.875 | 0.827  | 0.829 | 0.810    | 0.836 | 10.78 | 3.083 | 2.634  | 1.531        | 0.549        | <u>0.537</u> | <b>0.398</b> | <b>0.423</b> |
|          | 24  | 2.187        | 1.338        | 4.885    | 2.002 | 2.688        | 1.451 | 1.938        | 1.296 | <u>1.343</u> | <u>1.056</u> | 1.419        | 1.091        | 3.023  | 1.538 | 2.887  | 1.509 | 2.381    | 1.503 | 17.99 | 4.041 | 7.019  | 2.338        | 2.179        | 1.410        | <b>0.932</b> | <b>0.607</b> |
|          | 48  | 2.349        | 1.412        | 6.368    | 2.349 | 3.768        | 1.541 | 2.216        | 1.398 | <u>2.103</u> | <u>1.250</u> | 2.108        | 1.252        | 5.411  | 2.026 | 5.510  | 2.237 | 5.231    | 2.087 | 22.79 | 4.673 | 9.930  | 3.019        | 4.003        | 1.991        | <b>1.245</b> | <b>0.699</b> |
| ETTm1    | 1   | 0.132        | 0.312        | 0.135    | 0.352 | 0.183        | 0.397 | 0.137        | 0.341 | 0.145        | 0.224        | 0.173        | 0.260        | 0.371  | 0.489 | 0.387  | 0.522 | 0.309    | 0.506 | 0.190 | 0.231 | 0.154  | 0.271        | <u>0.096</u> | <u>0.142</u> | <b>0.083</b> | <b>0.128</b> |
|          | 24  | 0.618        | 0.771        | 1.102    | 1.029 | 1.101        | 1.009 | 0.602        | 0.726 | 0.717        | 0.600        | 0.912        | 0.731        | 0.702  | 0.707 | 0.711  | 0.741 | 0.671    | 0.795 | 1.520 | 1.202 | 1.103  | 1.003        | <u>0.412</u> | <u>0.452</u> | <b>0.372</b> | <b>0.403</b> |
|          | 48  | 0.829        | 0.894        | 2.083    | 1.243 | 1.298        | 1.039 | 0.811        | 0.882 | 1.052        | 0.781        | 1.152        | 0.890        | 0.935  | 0.846 | 0.933  | 0.886 | 0.897    | 0.836 | 2.283 | 1.390 | 1.492  | 1.21         | <u>0.559</u> | <u>0.540</u> | <b>0.451</b> | <b>0.515</b> |
| ETTm2    | 1   | 0.111        | 0.303        | 0.113    | 0.296 | 0.137        | 0.350 | 0.114        | 0.326 | 0.107        | 0.257        | 0.126        | 0.305        | 0.302  | 0.525 | 0.321  | 0.536 | 0.187    | 0.333 | 0.127 | 0.326 | 0.111  | 0.297        | <u>0.066</u> | <u>0.219</u> | <b>0.052</b> | <b>0.208</b> |
|          | 24  | 0.589        | 0.667        | 1.035    | 0.917 | 0.799        | 0.853 | 0.525        | 0.688 | 0.647        | 0.774        | 0.624        | 0.710        | 0.631  | 0.714 | 0.587  | 0.706 | 0.499    | 0.624 | 1.180 | 0.976 | 0.603  | 0.716        | <u>0.348</u> | <u>0.519</u> | <b>0.287</b> | <b>0.517</b> |
|          | 48  | 0.809        | 0.794        | 1.802    | 1.202 | 1.003        | 0.950 | 0.784        | 0.854 | 0.815        | 0.823        | 0.846        | 0.880        | 0.775  | 0.833 | 0.713  | 0.749 | 0.589    | 0.707 | 1.847 | 1.194 | 0.843  | 0.881        | <u>0.412</u> | <u>0.628</u> | <b>0.348</b> | <b>0.539</b> |
| WTH      | 1   | 0.359        | 0.499        | 0.521    | 0.691 | 0.491        | 0.607 | 0.338        | 0.543 | <u>0.205</u> | <u>0.352</u> | 0.360        | 0.400        | 0.489  | 0.659 | 0.493  | 0.602 | 0.488    | 0.649 | 0.731 | 0.754 | 0.501  | 0.536        | 0.298        | 0.398        | <b>0.052</b> | <b>0.114</b> |
|          | 24  | 1.219        | 1.008        | 1.504    | 1.208 | 1.482        | 1.017 | 1.208        | 1.002 | <u>0.436</u> | <u>0.560</u> | 0.537        | 0.533        | 1.311  | 1.041 | 1.287  | 1.034 | 0.999    | 0.895 | 1.756 | 1.218 | 1.377  | 1.023        | 0.677        | 0.649        | <b>0.110</b> | <b>0.165</b> |
|          | 48  | 1.741        | 1.247        | 2.001    | 1.257 | 1.699        | 1.146 | 1.733        | 1.216 | <u>0.710</u> | <u>0.643</u> | 0.731        | 0.655        | 1.610  | 1.169 | 1.651  | 1.185 | 1.483    | 1.078 | 2.620 | 1.609 | 2.703  | 1.591        | 0.919        | 0.813        | <b>0.215</b> | <b>0.297</b> |
| ECL      | 1   | 2.911        | 1.617        | 4.279    | 1.968 | 1.897        | 1.232 | 2.708        | 1.549 | <u>1.096</u> | <u>0.846</u> | 1.101        | 0.849        | 6.354  | 2.371 | 6.384  | 2.327 | 6.045    | 2.355 | 311   | 16.18 | 29.88  | 4.99         | 4.639        | 2.141        | <b>0.160</b> | <b>0.220</b> |
|          | 24  | 13.21        | 3.456        | 15.61    | 3.751 | 4.012        | 1.803 | 7.209        | 2.583 | 1.231        | <u>0.809</u> | <u>1.211</u> | 0.901        | 6.711  | 2.539 | 6.813  | 2.211 | 6.317    | 2.413 | 428   | 19.68 | 83.27  | 9.11         | 4.551        | 2.072        | <b>0.263</b> | <b>0.264</b> |
|          | 48  | 25.98        | 4.906        | 15.88    | 3.849 | 4.877        | 2.008 | 9.243        | 2.840 | 1.600        | 1.065        | <u>1.554</u> | <u>1.047</u> | 7.342  | 2.596 | 7.523  | 2.474 | 7.321    | 2.557 | 469   | 19.65 | 144.89 | 11.43        | 5.819        | 2.355        | <b>0.344</b> | <b>0.317</b> |
| Traffic  | 1   | 0.298        | 0.489        | 0.279    | 0.498 | <u>0.239</u> | 0.458 | 0.280        | 0.509 | 0.580        | 0.562        | 0.662        | 0.614        | 0.602  | 0.676 | 0.653  | 0.698 | 0.511    | 0.615 | 0.612 | 0.712 | 0.259  | 0.381        | 0.273        | <u>0.371</u> | <b>0.211</b> | <b>0.259</b> |
|          | 24  | 0.656        | 0.704        | 0.591    | 0.668 | 0.458        | 0.646 | 0.661        | 0.700 | 0.636        | 0.752        | 0.667        | 0.796        | 0.697  | 0.735 | 0.711  | 0.803 | 0.636    | 0.747 | 0.759 | 0.811 | 0.581  | 0.589        | <u>0.369</u> | <u>0.386</u> | <b>0.307</b> | <b>0.281</b> |
|          | 48  | 0.791        | 0.868        | 0.633    | 0.756 | 0.517        | 0.690 | 0.759        | 0.828 | 0.800        | 0.868        | 0.732        | 0.856        | 0.724  | 0.801 | 0.783  | 0.824 | 0.684    | 0.774 | 0.814 | 0.831 | 0.701  | 0.633        | <u>0.381</u> | <u>0.389</u> | <b>0.357</b> | <b>0.391</b> |
| Exchange | 1   | 0.579        | 0.561        | 0.887    | 0.842 | 0.836        | 0.814 | 0.432        | 0.557 | 0.502        | 0.639        | 0.531        | 0.649        | 0.847  | 0.820 | 0.859  | 0.827 | 0.778    | 0.782 | 0.460 | 0.578 | 0.275  | 0.424        | <u>0.243</u> | <u>0.403</u> | <b>0.153</b> | <b>0.256</b> |
|          | 24  | 1.427        | 0.995        | 1.273    | 1.028 | 1.047        | 0.923 | <u>0.980</u> | 0.890 | 1.131        | 0.963        | 1.074        | 0.936        | 1.283  | 1.043 | 1.332  | 1.054 | 1.295    | 1.038 | 1.250 | 0.918 | 1.129  | <u>0.863</u> | 1.039        | 0.939        | <b>0.803</b> | <b>0.618</b> |
|          | 48  | 2.473        | 1.273        | 2.711    | 1.547 | 2.279        | 1.410 | 1.748        | 1.122 | 1.693        | 1.201        | 1.576        | 1.155        | 2.371  | 1.440 | 2.162  | 1.370 | 1.884    | 1.273 | 2.379 | 1.242 | 1.507  | <u>1.028</u> | <u>1.374</u> | 1.072        | <b>1.100</b> | <b>0.843</b> |

**Table 2: Ablation studies of each component of CoSPOT(MSE).**

| $\mathcal{P}$ | $\mathcal{T}$                      | ETTh2       |             |             | ETTm1       |             |             | WTH         |             |             | ECL         |             |             | Traffic     |             |             |
|---------------|------------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|               |                                    | 1           | 24          | 48          | 1           | 24          | 48          | 1           | 24          | 48          | 1           | 24          | 48          | 1           | 24          | 48          |
| (1)           | $\times \times \times$             | 0.93        | 3.12        | 4.79        | 0.21        | 0.56        | 0.67        | 0.24        | 0.60        | 0.77        | 3.47        | 3.48        | 3.69        | 0.52        | 0.77        | 0.80        |
| (2)           | $\checkmark \times \times$         | 0.50        | 1.72        | 2.03        | 0.10        | 0.39        | 0.50        | 0.10        | 0.23        | 0.30        | 0.99        | 1.32        | 1.39        | 0.28        | 0.39        | 0.40        |
| (3)           | $\times \checkmark \checkmark$     | 0.72        | 2.32        | 2.89        | 0.12        | 0.48        | 0.54        | 0.18        | 0.43        | 0.60        | 2.13        | 2.77        | 2.91        | 0.41        | 0.60        | 0.64        |
| (4)           | $\checkmark \checkmark \checkmark$ | <b>0.40</b> | <b>0.93</b> | <b>1.25</b> | <b>0.08</b> | <b>0.37</b> | <b>0.45</b> | <b>0.05</b> | <b>0.11</b> | <b>0.22</b> | <b>0.16</b> | <b>0.26</b> | <b>0.34</b> | <b>0.21</b> | <b>0.31</b> | <b>0.36</b> |

**Table 3: Comparison of MSE and MAE results across various datasets under scenarios with an extended online phase, where the train/valid/test split is set to 10%/5%/85%.**

|        |     | ETTh2       |             |             | ETTm1       |             |             | WTH         |             |             | ECL         |             |             | Traffic     |             |             |
|--------|-----|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|        |     | 1           | 24          | 48          | 1           | 24          | 48          | 1           | 24          | 48          | 1           | 24          | 48          | 1           | 24          | 48          |
| FSNet  | MSE | 15.23       | 24.59       | 28.94       | 0.40        | 2.45        | 5.11        | 1.22        | 2.34        | 4.51        | 299.3       | 435.1       | 459         | 1.21        | 1.75        | 2.00        |
|        | MAE | 3.80        | 4.76        | 5.18        | 0.52        | 1.37        | 2.06        | 1.00        | 1.33        | 2.04        | 15.30       | 17.86       | 20.42       | 1.00        | 1.23        | 1.35        |
| OneNet | MSE | 4.24        | 9.34        | 13.58       | 0.28        | 1.43        | 1.68        | 0.81        | 1.54        | 2.99        | 30.09       | 82.29       | 142.27      | 0.53        | 0.78        | 0.93        |
|        | MAE | 2.01        | 3.01        | 3.49        | 0.43        | 1.07        | 1.20        | 0.80        | 1.10        | 1.60        | 5.29        | 8.04        | 10.80       | 0.59        | 0.74        | 0.91        |
| DSOF   | MSE | 0.85        | 2.89        | 5.03        | 0.17        | 0.56        | 0.78        | 0.31        | 0.71        | 1.14        | 4.90        | 5.17        | 6.09        | 0.50        | 0.71        | 0.79        |
|        | MAE | 0.87        | 1.50        | 2.04        | 0.40        | 0.67        | 0.85        | 0.54        | 0.80        | 1.01        | 2.13        | 2.20        | 2.37        | 0.66        | 0.76        | 0.80        |
| CoSPOT | MSE | <b>0.41</b> | <b>0.98</b> | <b>1.31</b> | <b>0.09</b> | <b>0.41</b> | <b>0.49</b> | <b>0.05</b> | <b>0.10</b> | <b>0.22</b> | <b>0.18</b> | <b>0.29</b> | <b>0.36</b> | <b>0.49</b> | <b>0.36</b> | <b>0.38</b> |
|        | MAE | <b>0.43</b> | <b>0.61</b> | <b>0.73</b> | <b>0.19</b> | <b>0.60</b> | <b>0.64</b> | <b>0.21</b> | <b>0.29</b> | <b>0.45</b> | <b>0.40</b> | <b>0.51</b> | <b>0.58</b> | <b>0.43</b> | <b>0.48</b> | <b>0.58</b> |

## 5.2 Ablation Study

To assess the impact of the compositional spectral prompt (i.e.,  $\mathcal{P}$ ) and text description (i.e.,  $\mathcal{T}$ ) in CoSPOT, Table 2 presents ablation studies across four cases, including the vanilla CoSPOT (Row (4)), with key observations as follows: (1) Introducing the compositional spectral prompt is helpful (Row (1) vs. (2)). Given the sequential nature of time-series data, in which continuous distribution shifts are inevitable, providing the model with compositional spectral prompts that capture these dynamics is highly effective. Specifically, CoSPOT captures overall time-series patterns by learning spectral basis prompts grounded in decomposed frequency bases. This enables effective adaptation to unseen patterns by compositionally recombining the learned spectral basis prompts, resulting in nearly a 50% improvement in MSE. (2) Providing recent pattern information to the LLM in the form of text descriptions is effective

**Table 4: Comparison of MSE and MAE results in a cross-dataset scenario, where different datasets from the same ETT domain are used in the training and online phases to induce distribution shifts, with the prediction horizon (i.e.,  $H$ ) set to 1. Training refers to the dataset used in the training phase, while Online refers to the dataset used in the online phase.**

| Training | Online  | FSNet  |       | OneNet |       | DSOF  |       | CoSPOT       |              |
|----------|---------|--------|-------|--------|-------|-------|-------|--------------|--------------|
|          |         | MSE    | MAE   | MSE    | MAE   | MSE   | MAE   | MSE          | MAE          |
| ETTh1    | → ETTh2 | 13.664 | 3.124 | 4.025  | 1.862 | 0.993 | 0.806 | <b>0.427</b> | <b>0.453</b> |
|          | → ETTm1 | 1.815  | 1.132 | 1.732  | 1.016 | 0.723 | 0.702 | <b>0.178</b> | <b>0.319</b> |
| ETTh2    | → ETTh1 | 12.667 | 3.011 | 3.833  | 1.808 | 0.901 | 0.849 | <b>0.399</b> | <b>0.531</b> |
|          | → ETTm2 | 1.791  | 1.029 | 1.489  | 1.022 | 0.818 | 0.744 | <b>0.211</b> | <b>0.359</b> |
| ETTm1    | → ETTh2 | 15.371 | 3.571 | 3.989  | 1.610 | 1.003 | 0.901 | <b>0.521</b> | <b>0.621</b> |
|          | → ETTm2 | 1.335  | 1.003 | 0.989  | 0.861 | 0.542 | 0.536 | <b>0.128</b> | <b>0.257</b> |
| ETTm2    | → ETTh2 | 14.989 | 3.199 | 4.138  | 1.734 | 1.211 | 0.945 | <b>0.513</b> | <b>0.616</b> |
|          | → ETTm1 | 1.299  | 1.039 | 1.315  | 0.946 | 0.517 | 0.619 | <b>0.132</b> | <b>0.263</b> |

online phase. Prior methods (i.e., FSNet, OneNet, and DSOF) experience significant performance degradation compared to the results in Table 1 for the following two reasons: (1) They are incapable of storing all the recurring patterns in an associative memory as the online phase is extended, and (2) the continuous occurrence of unseen patterns prevents the model from maintaining adaptability. In contrast, CoSPOT, thanks to the rich knowledge and transferability of the pre-trained LLM, effectively maintains adaptability even in data-scarce online scenarios, demonstrating performance comparable to the results in Table 1. We argue that adaptability to an extended online phase is enhanced by two key designs: (i) compositional spectral prompting, which represents evolving patterns using a finite set of spectral basis prompts, and (ii) a text description of recent patterns provided to the LLM, which compensates for data scarcity and enables efficient adaptation without additional training.

**5.3.2 Robustness to distribution shifts.** Table 4 shows the cross-dataset experiments using the ETT datasets. The ETTh series datasets (i.e., ETTh1 and ETTh2) are collected at an hourly interval, with ETTh1 representing a shorter time window and ETTh2 covering a longer period. The ETTm series datasets (i.e., ETTm1 and ETTm2), collected at 15-minute intervals, differ in feature count and time span, with ETTm2 providing finer temporal resolution compared to ETTm1. To induce distribution shifts, we deliberately use different datasets for the training and online phases. The model learns the base knowledge from the training data and adapts to the streaming online data, with the prediction horizon set to 1. We observe that CoSPOT outperforms all baselines across the 8 scenarios. FSNet, OneNet, and DSOF rely on associative memory to store recurring patterns and adapt by retrieving similar ones. This approach fails when unseen patterns arise, as no meaningful associations can be found, leading to substantial performance degradation (compare with the results in Table 1). In contrast, CoSPOT represents patterns through compositional spectral prompts by learning a spectral basis prompt for each frequency basis, rather than directly storing the time series patterns. As a result, even unseen patterns can be reliably expressed as new compositions of spectral basis prompts, making the model robust to distribution shifts and allowing its performance to remain largely consistent with the results in Table 1.

**5.3.3 Online Updating Cost.** To analyze the efficiency of CoSPOT, we compare the number of updated parameters during the online phase and runtime statistics with those of existing online time series forecasting models.

**Table 5: The number of parameters updated during the online phase for prediction horizon 1 on the ETTh2 dataset.**

| FSNet     | OneNet    | DSOF      | CoSPOT |
|-----------|-----------|-----------|--------|
| 2,037,115 | 1,018,045 | 1,236,349 | 897    |

*Parameters.* Table 5 reports the number of parameters updated during the online phase (i.e., those that are not frozen) for each OTSF model in experiments using the ETTh2 dataset. CoSPOT updates significantly fewer parameters compared to the baselines, indicating its efficiency. This demonstrates that CoSPOT can leverage model adaptability to achieve strong performance (see Table 1) with fewer

updated parameters. Updating many parameters during the online phase can make models vulnerable to distribution shifts and cause them to forget the base knowledge learned during training phase. In this regard, CoSPOT effectively and efficiently adapts to new data without forgetting previously learned knowledge by updating only a small subset of parameters.

**Table 6: Comparison of runtime statistics between CoSPOT and existing OTSF methods for the scenario with a prediction horizon of 1 on each dataset, utilizing the total Training Phase Duration (sec), the total Online Phase Duration (sec), and the Inference Latency (sec/itr), defined as the time required per update.**

|        | Metric                        | ETTh1 | ETTh2 | ETTh1  | ETTh2  | WTH    | ECL   | Traffic |
|--------|-------------------------------|-------|-------|--------|--------|--------|-------|---------|
| FSNet  | Training Phase Duration (sec) | 275   | 274   | 801    | 830    | 775    | 398   | 159     |
|        | Online Phase Duration (sec)   | 341   | 332   | 1,024  | 1,019  | 998    | 463   | 351     |
|        | Inference Latency (sec/itr)   | 0.031 | 0.030 | 0.022  | 0.022  | 0.025  | 0.023 | 0.026   |
| OneNet | Training Phase Duration (sec) | 551   | 540   | 1,621  | 1,633  | 1,503  | 848   | 303     |
|        | Online Phase Duration (sec)   | 701   | 690   | 2,237  | 2,241  | 2,079  | 994   | 741     |
|        | Inference Latency (sec/itr)   | 0.063 | 0.061 | 0.050  | 0.050  | 0.051  | 0.050 | 0.053   |
| DSOF   | Training Phase Duration (sec) | 622   | 613   | 1,904  | 1,936  | 1,789  | 994   | 379     |
|        | Online Phase Duration (sec)   | 733   | 705   | 2,588  | 2,559  | 2,371  | 1,201 | 855     |
|        | Inference Latency (sec/itr)   | 0.064 | 0.062 | 0.058  | 0.056  | 0.057  | 0.059 | 0.062   |
| CoSPOT | Training Phase Duration (sec) | 4,650 | 4,650 | 14,428 | 14,501 | 13,786 | 7,249 | 2,788   |
|        | Online Phase Duration (sec)   | 787   | 741   | 2,938  | 2,973  | 2,841  | 1,389 | 931     |
|        | Inference Latency (sec/itr)   | 0.068 | 0.065 | 0.064  | 0.066  | 0.068  | 0.065 | 0.067   |

*Runtime Comparison.* Table 6 presents the runtime statistics for each dataset in the scenario where the prediction horizon is 1. We utilize the total Training Phase Duration (sec), the total Online Phase Duration (sec), and the Inference Latency (sec/itr), defined as the time required per update. OTSF requires a model to be sufficiently pre-trained on initial data (i.e., training phase) and then continually adapted to the subsequent data stream (i.e., online phase). Consequently, resource optimization must prioritize the cost-sensitive online phase over the training phase. The duration of the online phase and the inference latency for CoSPOT require only slightly more time compared to existing OTSF models. This demonstrates that CoSPOT continuously adapts to new online distributions with significantly less parameter tuning (see Table 5) by simultaneously leveraging the LLM’s superior transferability and the distribution guidance provided by compositional spectral prompts. While CoSPOT incurs a higher initial cost for training duration, this cost is justifiable as the training phase is less resource-sensitive and prioritizes the sufficient acquisition of base knowledge. In summary, CoSPOT achieves significantly superior performance (please refer to Table 1) in the cost-sensitive online phase with only a comparable cost, despite its larger initial overhead during the initial training.

## 6 Conclusion

In this paper, we present the first LLM-based OTSF framework, called CoSPOT, which excels in both continuous distribution shifts and extended online scenarios. We devise a compositional spectral prompting strategy that captures the overall distribution (i.e., overall pattern) of each input by recombining learnable spectral basis prompts according to its frequency-domain structure. These distribution-aware prompts guide the pre-trained LLM to adapt its rich knowledge and transferability to continuously shifting online



distributions, enabling efficient rapid adaptation in data-scarce scenarios. Furthermore, text descriptions containing recent pattern information enrich the limited online data without requiring additional training. CoSPOT demonstrates promising performance across various real-world datasets and exhibits robust performance against distribution shifts and the extension of the online phase, highlighting its applicability to real-world online forecasting scenarios.

## References

- [1] Abdul Fatir Ansari, Oleksandr Shchur, Jari Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. 2025. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821* (2025).
- [2] Sergul Aydore, Tianhao Zhu, and Dean P Foster. 2019. Dynamic local regret for non-convex online forecasting. *Advances in neural information processing systems* 32 (2019).
- [3] Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. 2023. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *CoRR* (2023).
- [4] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2023. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688* (2023).
- [5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiuhua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).
- [6] San Gultekin and John Paisley. 2018. Online forecasting matrix factorization. *IEEE Transactions on Signal Processing* 67, 5 (2018), 1223–1236.
- [7] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022. Visual prompt tuning. In *European conference on computer vision*. Springer, 709–727.
- [8] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728* (2023).
- [9] Vitaly Kuznetsov and Mehryar Mohri. 2016. Time series prediction and online learning. In *Conference on Learning Theory*. PMLR, 1190–1213.
- [10] Ying-yee Ava Lau, Zhiwen Shao, and Dit-Yan Yeung. 2025. Fast and Slow Streams for Online Time Series Forecasting Without Information Leakage. In *The Thirteenth International Conference on Learning Representations*.
- [11] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625* (2023).
- [12] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730* (2022).
- [13] Quang Pham, Chenghao Liu, Doyen Sahoo, and Steven CH Hoi. 2022. Learning fast and slow for online time series forecasting. *arXiv preprint arXiv:2202.11672* (2022).
- [14] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
- [15] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. 2024. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616* (2024).
- [16] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. 2022. S-prompts learning with pre-trained transformers: An occam’s razor for domain incremental learning. *Advances in Neural Information Processing Systems* 35 (2022), 5682–5695.
- [17] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 139–149.
- [18] Qingsong Wen, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, Tieniu Tan, et al. 2023. Onenet: Enhancing time series forecasting models under concept drift by online ensembling. *Advances in Neural Information Processing Systems* 36 (2023), 69949–69980.
- [19] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186* (2022).
- [20] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems* 34 (2021), 22419–22430.
- [21] Zhijian Xu, Ailing Zeng, and Qiang Xu. 2023. FITS: Modeling time series with 10k parameters. *arXiv preprint arXiv:2307.03756* (2023).
- [22] Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. 2023. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems* 36 (2023), 76656–76679.
- [23] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 11121–11128.
- [24] Yunhao Zhang and Junchi Yan. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*.
- [25] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 11106–11115.
- [26] Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. 2022. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in neural information processing systems* 35 (2022), 12677–12690.
- [27] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*. PMLR, 27268–27286.
- [28] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems* 36 (2023), 43322–43355.