ExoTimer: Leveraging Large Language Models for Time Series Forecasting with Exogenous Variables

Anonymous submission

Abstract

Real-world systems often exhibit complex behaviors and are influenced by various external factors, making the integration of exogenous variables essential for accurate and robust time series forecasting. However, modeling time series with exogenous variables remains challenging due to dynamic cross-variable dependencies and the semantic gap between numerical time series data and external contextual knowledge. Large language models (LLMs) have demonstrated powerful language understanding and knowledge representation capabilities in real-world systems, offering a promising solution to bridge this gap. Motivated by this, we propose ExoTimer, a framework that deeply integrates LLMs for time series modeling with exogenous variables. We begin by introducing an Exo-Aware Endogenous Encoder to dynamically incorporate important exogenous variable information and generate patch-level representations for endogenous variables. To leverage the rich knowledge in LLMs, a Multi-Attribute Prompt Embedding module is elaborately designed to convert heterogeneous temporal features, contextual information and task specifications into LLM-interpretable textual prompts. Additionally, we propose Bi-Hash Alignment, a lightweight cross-modal alignment mechanism that bridges textual and temporal modalities in a shared hash space. Finally, a Dual-Branch Predictor with a learnable coefficient is employed to obtain the final time series prediction by integrating temporal-text and text-temporal representations. Extensive experiments on twelve real-world datasets demonstrate that ExoTimer achieves state-of-the-art performance and exhibits generalizability and scalability in both few-shot and zero-shot scenarios.

1 Introduction

Time series forecasting plays a crucial role in real-world applications in various domains, including climate modeling (Schneider and Dickinson 1974), energy management (Liu et al. 2023) and traffic analysis (Liu et al. 2022; Miao et al. 2024). Although deep learning models have shown remarkable success in time series forecasting, most existing approaches (Liu et al. 2024b; Nie et al. 2023; Wu et al. 2023) limit their scope to the target series (endogenous variables). Due to the complex and non-stationary nature of real-world systems, time series are often affected by external factors, such as traffic flow (Lv et al. 2014), economic trends (Niu et al. 2020) and social events (Huang et al. 2025). Thus, incorporating exogenous variables becomes indispensable for reliable and robust time series forecasting.

The core challenge in forecasting with exogenous variables lies in modeling the correlations and causal relationships between external factors and target (endogenous) variables. Recent works such as CATS (Lu et al. 2024) and TimeXer (Wang et al. 2024) employ attention mechanisms to capture inherent dependencies between observed exogenous and endogenous series. Considering complex intricate influences from external environments, ExoLLM (Huang et al. 2025) argues that relying solely on the time series modality is insufficient to capture external influences and may lead to spurious correlations. Instead, it introduces LLMs to leverage language knowledge for better comprehension of external factors. However, pre-trained LLMs are built based on static training corpora, failing to adapt to the ever-changing real-world knowledge and dynamic evolution of cross-variable correlations (Huang et al. 2023). To deal with the complex and dynamic influences of exogenous series, it is essential to leverage LLM-derived linguistic knowledge for contextual interpretation of external factors, while integrating time-series strategies that dynamically prioritize influential exogenous variables for downstream tasks and mitigate noise from irrelevant external factors.

However, integrating LLMs with temporal modeling poses two major challenges. The first challenge lies in cross-modal knowledge transfer. LLMs are trained based on textual corpora and demonstrate significant semantic disparities when processing numeric time series data (Sun et al. 2024; Pan et al. 2024; Liu et al. 2025c). To effectively activate LLMs' capability for temporal forecasting, elaborately designed prompts are required to bridge the modality gap and facilitate the interpretation of temporal features. The second challenge is the cross-modal alignment caused by the inherent distribution discrepancy between textual embeddings and temporal representations. Existing approaches (Liu et al. 2025a; Huang et al. 2025; Liu et al. 2024a) typically rely on attention mechanisms for cross-modal alignment, which calculate attention scores between different modalities to retrieve and fuse relevant information. However, these methods face two key limitations: 1) divergent feature space distributions impair retrieval accuracy; 2) quadratic computational complexity of attention mechanisms limits performance on long sequences.

In this paper, we proposes a framework called ExoTimer to leverage LLMs for **Time** Series Fo**r**ecasting with **Exo**genous variables (**ExoTimer**) to address the above issues. Technically, ExoTimer first introduces an Exo-Aware Endogenous Encoder to discover interrelationships between exogenous and endogenous variables by modeling their frequency-domain correlations, which adaptively identifies key exogenous variables and mitigates the adverse effects of noisy external factors. The encoder then yields comprehensive global embeddings and patch-level representations for endogenous variables, providing robust and informative sequence embeddings for downstream tasks. To help LLMs understand the characteristics of time series and leverage prior knowledge, a Multi-Attribute Prompt Embedding module is elaborately designed to populates heterogeneous temporal features, contextual information and task specifications into a structured prompt template. Additionally, ExoTimer proposes Bi-Hash Alignment to facilitate mutual interaction between textual and temporal modalities, which aligns their representations in a shared hash space. Finally, a Dual-Branch Predictor with a learnable coefficient is employed to obtain the final time series prediction by integrating temporal-text and text-temporal representations. In summary, Our main contributions are as follows:

- By introducing an Exo-Aware Endogenous Encoder, ExoTimer adaptively prioritizes task-beneficial exogenous variables while suppressing irrelevant external factors, enabling effective modeling of complex and dynamic exogenous-endogenous variable interactions.
- A Multi-Attribute Prompt Embedding module is designed to leverage the knowledge of LLMs via context, task instruction and time series analysis prompts.
- Bi-Hash Alignment is proposed to bridge the semantic gap between textual and temporal modalities in a shared hash space. To the best of our knowledge, this is the first work to introduce Locality Sensitive Hashing (LSH) for cross-modal alignment between LLMs and time series.
- ExoTimer consistently achieves state-of-the-art performance in mainstream forecasting tasks, including few-shot and zero-shot scenarios, while maintaining excellent cross-modality alignment efficiency.

2 Method

In forecasting with exogenous variables, given an endogenous time series $\mathbf{X} = \{x_1, x_2, \ldots, x_L\} \in \mathbb{R}^{1\times L}$ and its correlated exogenous variables $\mathbf{Z} = \{\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_C\} \in \mathbb{R}^{C\times L_x}$, the task aims to learn a forecasting model $\mathcal{F}(\cdot)$ that predicts future T time steps of the endogenous series $\hat{\mathbf{X}} = \{x_{L+1}, x_{L+2}, \ldots, x_{L+T}\}$ based on both its historical observations and exogenous variables. Specifically, x_i denotes the value of the endogenous series at the i-th time step, $\mathbf{Z}_i \in R^{1\times L_x}$ represents the i-th exogenous variable where $i \in \{1, 2, \ldots, C\}$, C is the number of exogenous variables, L and L_x are the look-back window sizes for the endogenous and exogenous variables, respectively and T is the prediction horizon.

The architecture of the **ExoTimer** is illustrated in Fig. 1, which comprises four key components: a) an **Exo-Aware Endogenous Encoder** to adaptively discover important exogenous variable information and capture inter-patch dependencies for endogenous variables; b) a **Multi-Attribute Prompt Embedding** module to extract multiple attribute features from endogenous time series and generate proper prompts for pre-trained LLMs; (c) **Bi-Hash Alignment**, an LSH-based module to enable bi-directional modality alignment between textual and temporal representations; (d) a **Dual-Branch Predictor** to employ a learnable coefficient to fuse predictions from both temporal and textual modalities.

2.1 Exo-Aware Endogenous Encoder

AdaExoFusion To robustly handle irregular and heterogeneous exogenous series, the AdaExoFusion module first projects the exogenous and endogenous variables into an embedding space as follows:

$$\mathbf{E}_{\text{exo}} = \text{Embed}(\mathbf{Z}) \in \mathbb{R}^{C\times d} \quad \mathbf{E}_{\text{endo}} = \text{Embed}(\mathbf{X}) \in \mathbb{R}^{1\times d} \tag{1}$$

where d is the embedding dimension. The embeddings are then transformed to the frequency domain by real Fast Fourier Transform (rFFT).

$$\mathbf{E}_f^{exo} = |\text{rFFT}(\mathbf{E}_{\text{exo}})| \in \mathbb{R}^{C\times \frac{d}{2}}, \mathbf{E}_f^{endo} = |\text{rFFT}(\mathbf{E}_{\text{endo}})| \in \mathbb{R}^{1\times \frac{d}{2}} \tag{2}$$

To adaptively capture dependencies between exogenous and endogenous variables in the frequency domain, we compute pairwise distances using a learnable Mahalanobis metric (Mahalanobis 1936):

$$\mathbf{D}_i = (\mathbf{E}_f^{endo} - \mathbf{E}_{i,f}^{exo})\mathbf{Q}(\mathbf{E}_f^{endo} - \mathbf{E}_{i,f}^{exo})^\top, i = \{1, \cdots, C\} \tag{3}$$

where $\mathbf{Q} = \mathbf{A}^\top\mathbf{A}$ is a positive semi-definite matrix with $\mathbf{A} \in \mathbb{R}^{\frac{d}{2}\times\frac{d}{2}}$ as learnable parameters. The distances $\mathbf{D}_i$ are converted into similarity scores and sparsified to emphasize strong external dependencies and reduce noise:

$$\mathbf{C}_i = \frac{1}{\mathbf{D}_i + \epsilon} \quad \mathbf{P}_i = \frac{\mathbf{C}_i \cdot \tau}{\max(\mathbf{C}_i)} \tag{4}$$

where $\epsilon > 0$ is a small constant ensuring numerical stability, and $\mathbf{C}_i$ denotes the normalized similarity between the i-th exogenous variable and the endogenous variable, τ is a temperature parameter that controls the sparsity of the adaptive mask, and $\mathbf{P}_i$ represents the probability of retaining correlations in the mask. A larger $\mathbf{P}_i$ indicates a stronger correlation with the endogenous variable.

The binary adaptive mask matrix $\mathbf{M} \in \mathbb{R}^{1\times C}$ is then sampled via a Gumbel-Softmax (Jang et al. 2017) distribution:

$$\mathbf{M}_i \sim \text{Gumbel-Softmax}(\mathbf{P}_i) \in \{0, 1\} \tag{5}$$

To fuse important exogenous information into the endogenous representation, we apply masked cross-attention:

$$\mathbf{Attn} = \text{Softmax}\left(\frac{\mathbf{E}_{\text{endo}}\mathbf{W}_Q(\mathbf{E}_{\text{exo}}\mathbf{W}_K)^\top}{\sqrt{d}} + \mathbf{mask}\right) \tag{6}$$

$$\mathbf{E}_{\text{endo}}^{\text{global}} = \text{LayerNorm}\left(\mathbf{E}_{\text{endo}} + \mathbf{Attn}\mathbf{E}_{\text{exo}}\mathbf{W}_V\right)$$

where $\mathbf{mask}_i = 1$ if $\mathbf{M}_i = 1$, otherwise $\mathbf{mask}_i = -\infty$ if $\mathbf{M}_i = 0$. $\mathbf{W}_Q, \mathbf{W}_K$ and $\mathbf{W}_V$ are learnable weight matrices for query, key, and value projections, respectively, and LayerNorm$(\cdot)$ denotes layer normalization (Ba et al. 2016).

Endogenous Patch Encoder The endogenous time series is segmented into overlapped or non-overlapped patches (Nie et al. 2023), which are embedded through linear projection and combined with position encodings. Then, self-attention (Vaswani et al. 2017) is applied to capture inter-

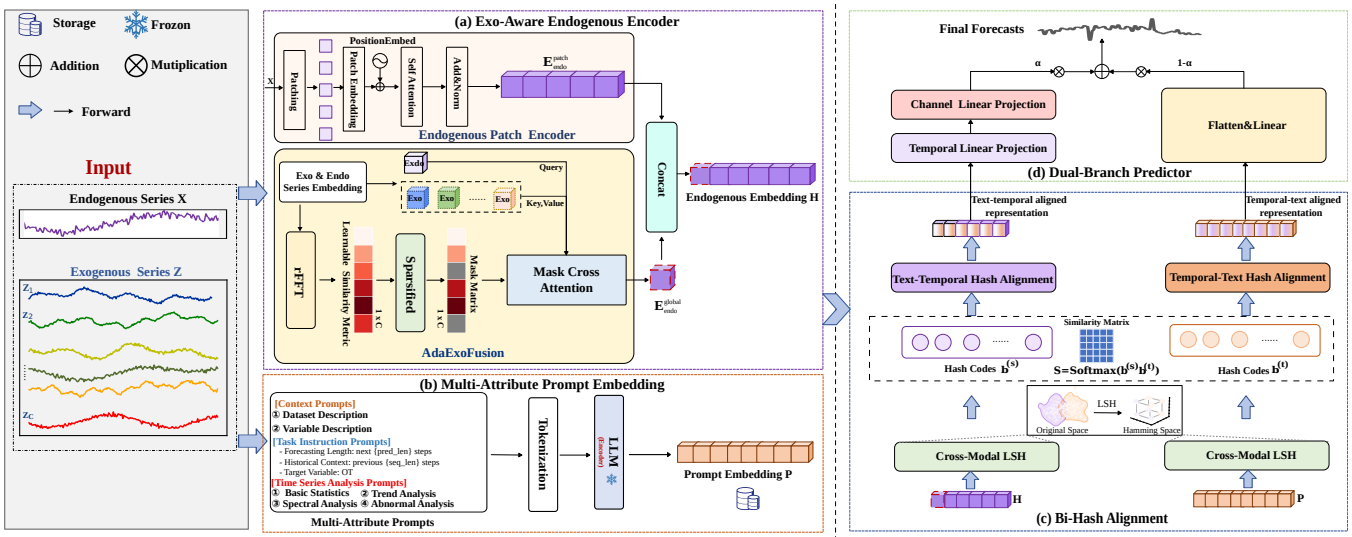

Figure 1: The model framework of ExoTimer.

patch dependencies:

$$\begin{aligned}\mathcal{P} &= \text{Patching}\,(\mathbf{X}) \in \mathbb{R}^{N\times P}\\ \mathbf{E}_{\text{patch}} &= \text{PatchEmbed}(\mathcal{P}) \in \mathbb{R}^{N\times d}\\ \mathbf{E}_{\text{pos}} &= \text{PositionEmbed}(\mathcal{P}) \in \mathbb{R}^{N\times d}\\ \mathbf{E}^{\text{patch}}_{\text{endo}} &= \mathbf{E}_{\text{pos}} + \mathbf{E}_{\text{patch}} \in \mathbb{R}^{N\times d}\\ \mathbf{E}^{\text{patch}}_{\text{endo}} &= \text{LayerNorm}(\mathbf{E}^{\text{patch}}_{\text{endo}} + \text{MSA}(\mathbf{E}^{\text{patch}}_{\text{endo}}))\end{aligned} \tag{7}$$

where P is the length of each patch, $N = \lfloor \frac{(L-P)}{S} \rfloor + 2$ represents the number of patches where S denotes the sliding stride, PatchEmbed$(\cdot)$ projects each patch into a d-dimensional vector, PositionEmbed$(\cdot)$ is applied to generate embeddings to encode the temporal order of patches, $\mathbf{E}^{\text{patch}}_{\text{endo}}$ denotes the temporal embedding for the endogenous variable, MSA$(\cdot)$ denotes the multi-head self-attention layer.

The final endogenous embedding $\mathbf{H}$ is generated by concatenating the exogenous variable fused global embedding $\mathbf{E}^{\text{global}}_{\text{endo}}$ and patch embedding $\mathbf{E}^{\text{patch}}_{\text{endo}}$:

$$\mathbf{H} = [\mathbf{E}^{\text{global}}_{\text{endo}}, \mathbf{E}^{\text{patch}}_{\text{endo}}] \in \mathbb{R}^{(N+1)\times d} \tag{8}$$

where $[\cdot]$ denotes the concatenation operation along the sequence dimension.

2.2 Multi-Attribute Prompt Embedding

Since LLMs are pre-trained on textual corpora, they encounter the inherent challenge in comprehending the underlying dynamics of temporal patterns in time series (Jin et al. 2024). To address this challenge, we design a structured prompt template that bridges raw temporal features and the language understanding capabilities of LLMs, as illustrated in Fig. 1. The prompts are composed of three key parts:

1. **Context Prompts** ($PT_{context}$) provide an overview of the dataset along with definitions of exogenous and endogenous variables.
2. **Task Instruction Prompts** (PT_{task}) specify the analytical objectives.
3. **Time Series Analysis Prompts** (PT_{time}) encompass multi-category features extracted from time series.

Concretely, Time Series Analysis Prompts incorporate four types of features: (1) **Basic Statistical Features**, e.g., mean, median, standard deviation and range, which summarize central tendency and dispersion, enabling LLMs to effectively capture fundamental data patterns. (2) **Trend Features**, quantifying long-term evolutions using trend direction (e.g., upward or downward slope), trend strength (e.g., the magnitude of the slope) and trend stability (e.g., variability of residuals around the trend line). (3) **Spectral Features**, employing Fourier transforms to reveal periodic patterns. (4) **Anomaly Features**, identifying deviations from expected patterns to enhance analysis reliability. The relevant features for each category and their corresponding prompt templates are summarized in Table 7 (see Appendix D). All features are extracted using a dedicated time series feature extractor to capture diverse dynamic attributes. The extracted features are inserted into pre-defined textual templates, enabling LLMs to effectively leverage their prior knowledge for time-series pattern understanding.

To further improve efficiency and reduce computational overhead, we pre-generate and cache the final prompt embeddings by encoding the prompts using the LLM before training:

$$\mathbf{P} = \text{LLM}(PT_{context}, PT_{task}, PT_{time}) \in \mathbb{R}^{d_{llm}} \tag{9}$$

where LLM$(\cdot)$ denotes the encoder of the pre-trained LLM, $\mathbf{P}$ is the resulting prompt embedding, and d_{llm} denotes the embedding dimension of the LLM.

2.3 Bi-Hash Alignment

Recent approaches to align time series with LLMs falls into three main categories: cross-modal retrieval, contrastive learning and knowledge distillation (Liu et al. 2025b). Among these, cross-modal retrieval methods are most widely adopted in current research (Huang et al. 2024; Liu et al. 2024a; Jin et al. 2024). These methods typically imple-

ment retrieval between temporal representations and textual embeddings by using cross-attention mechanisms. However, these attention-based approaches suffer from quadratic computational complexity, limiting their scalability to long sequences. More importantly, the inherent heterogeneity between time series and textual modalities results in divergent feature space distributions, posing a substantial challenge to attention mechanisms to achieve robust feature alignment.

To address these limitations, we leverage the theoretical properties of Locality Sensitive Hashing (LSH) (Indyk and Motwani 1998) to enable robust and order-preserving cross-modal retrieval. By mapping high-dimensional features from heterogeneous modalities into a shared, compact and discrete hash space, LSH offers several key theoretical guarantees for effective retrieval and alignment.

Order Preservation Hashing A key theoretical advantage of employing LSH for cross-modal alignment is order-preserving property, which guarantees that semantic similarity in the original feature spaces is faithfully preserved in the resulting binary Hamming space. This property is crucial for aligning heterogeneous modalities, such as temporal features and textual knowledge, where the original feature spaces may exhibit divergent statistical distributions and scales.

Let $\mathbf{h}, \mathbf{p}$ denote two vectors. For a random Gaussian projection vector $\mathbf{a} \sim \mathcal{N}(0, \mathbf{I}_k)$, LSH produces $h_{\mathbf{a}}(\mathbf{x}) = \text{sign}(\mathbf{a}^\top \mathbf{x})$. The collision probability between two vectors depends solely on their angle:

$$\mathbb{P}\left[\text{sign}(\mathbf{a}^\top \mathbf{p}) \neq \text{sign}(\mathbf{a}^\top \mathbf{h})\right] = \frac{1}{\pi}\cos^{-1}\left(\frac{\mathbf{p}^\top \mathbf{h}}{\|\mathbf{p}\|\,\|\mathbf{h}\|}\right) \quad (10)$$

Therefore, for any triplet $\mathbf{p}, \mathbf{h}_1, \mathbf{h}_2$ with $\text{similarity}(\mathbf{p}, \mathbf{h}_1) > \text{similarity}(\mathbf{p}, \mathbf{h}_2)$, it follows that:

$$\mathbb{E}[d_H(\mathbf{b}^{(\mathbf{p})}, \mathbf{b}^{(\mathbf{h}_1)})] < \mathbb{E}[d_H(\mathbf{b}^{(\mathbf{p})}, \mathbf{b}^{(\mathbf{h}_2)})] \quad (11)$$

where $\mathbf{b}^{(\cdot)} \in \{-1, +1\}^k$ denotes the binary codes of a vector, obtained by concatenating the results of $h_{\mathbf{a}}$ over k independent random projections, and d_H denotes the normalized Hamming distance. As the LSH binarization dimension k increases, the empirical Hamming distance sharply converges around its expected value, as guaranteed by the Chernoff-Hoeffding bound:

$$\mathbb{P}\left[\left|d_H(\mathbf{b}^{(\mathbf{p})}, \mathbf{b}^{(\mathbf{h})}) - \frac{1}{\pi}\cos^{-1}\left(\frac{\mathbf{p}^\top \mathbf{h}}{\|\mathbf{p}\|\,\|\mathbf{h}\|}\right)\right| \geq \epsilon\right] \leq 2\exp(-2k\epsilon^2) \quad (12)$$

Consequently, LSH provides an exponential convergence guarantee for order preservation in cross-modal retrieval.

Generalization Error and Robustness Analysis Let $\mathcal{X} \subset \mathbb{R}^D$ denotes the original continuous feature space and $\mathcal{B} = \{-1, +1\}^k$ the corresponding hash space. We define the family of hash functions as $\mathcal{H} = \{h : \mathbf{x} \mapsto \mathbf{b}\}$, mapping from $\mathcal{X}$ to $\mathcal{B}$. The empirical Rademacher complexity (Koltchinskii and Panchenko 2002) of $\mathcal{H}$ is defined as:

$$\hat{\mathfrak{R}}n(\mathcal{H}) = \mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}\langle \sigma_i, h(\mathbf{x}_i)\rangle\right] \quad (13)$$

where $\sigma_i \in \{-1, +1\}^k$ are independent Rademacher random vectors and $\langle\cdot,\cdot\rangle$ denotes the standard inner product. Since the Vapnik-Chervonenkis (VC) (Vapnik and Chervonenkis 1971) dimension of the discrete hash space $\mathcal{B}$ is upper-bounded by k, which is significantly lower than that of the full continuous space $\mathbb{R}^D$, the generalization error is bounded by:

$$R(h) \leq \hat{R}_S(h) + 2\hat{\mathfrak{R}}_n(\mathcal{H}) + 3\sqrt{\frac{\log(2/\delta)}{2n}}, \quad (14)$$

where $\hat{R}_S(h)$ denotes the empirical risk. Since $\hat{\mathfrak{R}}_n(\mathcal{H}) \leq \sqrt{\frac{k\log 2}{n}}$ (see proof in Appendix B.1), this bound grows sublinearly with the hash dimension k, thereby effectively mitigating the risk of overfitting. Moreover, the hash space naturally serves as an information bottleneck transmitting only alignment-relevant semantic information. This makes the approach robust to both noise and intra- and inter-modal distribution shift.

Based on above theoretical insights, we introduce Bi-Hash Alignment, an LSH-based module that bridges heterogeneous modalities by aligning their representations in a shared hash space. This module consists of temporal-text hash alignment and text-temporal hash alignment components, both implemented by Cross-Modal LSH, which facilitates multi-modal feature fusion through differentiable LSH with similarity-guided alignment.

Specifically, given the input representations $\mathbf{P} \in \mathbb{R}^{d_{llm}}$ and $\mathbf{H} \in \mathbb{R}^d$ from textual and time series modality, respectively. To obtain k-dimensional hash codes, Cross-Modal LSH first applies learnable linear projections and then employs Gumbel-Softmax relaxation with temperature γ for differentiable optimization during training. At inference, sign activation is used to produce deterministic binary codes:

$$\begin{aligned} \text{Training: } \mathbf{b}^{(t)} &= 2\text{GS}_\gamma([\mathbf{I}^{(t)}, -\mathbf{I}^{(t)}])_{:,1} - 1, \\ \mathbf{b}^{(s)} &= 2\text{GS}_\gamma([\mathbf{I}^{(s)}, -\mathbf{I}^{(s)}])_{:,1} - 1 \\ \text{Inference: } \mathbf{b}^{(t)} &= \text{sign}(\mathbf{I}^{(t)}), \mathbf{b}^{(s)} = \text{sign}(\mathbf{I}^{(s)}) \end{aligned} \quad (15)$$

where $\mathbf{I}^{(t)} = \mathbf{P}\mathbf{W}_t + \mathbf{c}_t$ and $\mathbf{I}^{(s)} = \mathbf{H}\mathbf{W}_s + \mathbf{c}_s$. $\mathbf{W}_t \in \mathbb{R}^{d_{llm}\times k}$, $\mathbf{W}s \in \mathbb{R}^{d\times k}$, $\mathbf{c}_t, \mathbf{c}_s \in \mathbb{R}^k$ are learnable projection parameters. $\text{GS}_\gamma([\mathbf{I}^*, -\mathbf{I}^*])_{:,1}$ denotes the probability of selecting $+1$ for each bit by Gumbel-Softmax relaxation. To align the modalities, we compute the normalized cosine similarity $\mathbf{S}$ between the hash codes:

$$\mathbf{S} = \text{Softmax}\left({\mathbf{b}^{(t)}}^\top \mathbf{b}^{(s)}\right) \quad (16)$$

Hybrid representations are then generated as:

$$\mathbf{F}_{st} = \text{MLP}\left([\mathbf{P}, \mathbf{S}\mathbf{H}]\right) \in \mathbb{R}^{d_{llm}} \quad (17)$$

where $[\cdot]$ denotes concatenation and $\text{MLP}(\cdot)$ consists of two linear layers with GELU activation.

Applying the above procedure yields the temporal-text hash alignment result $\mathbf{F}_{st}$. The text-temporal hash alignment result $\mathbf{F}_{ts}$ can be obtained in a symmetrical manner.

2.4 Dual-Branch Predictor

We generate predictions through a dual-branch inference method. For the temporal-text aligned representation, we flatten the features and apply a linear projection layer to obtain the prediction $\mathbf{Y}_{st}$. For the text-temporal aligned representation, we first project features to the prediction horizon, followed by a projection to the target channel dimension, resulting in $\mathbf{Y}_{ts}$. The final prediction $\mathbf{Y}$ is computed as a weighted combination of both branches using a learnable

coefficient α:

$$\mathbf{Y} = \alpha \mathbf{Y}_{st} + (1-\alpha)\mathbf{Y}_{ts}, \alpha \in [0,1] \tag{18}$$

3 Experiment

Datasets We conduct experiments on twelve real-world datasets. For long-term forecasting, we employ seven established benchmarks: ETT datasets (including four subsets: ETTh1, ETTh2, ETTm1, ETTm2) (Zhou et al. 2021), Traffic (Wu et al. 2023), ECL (Li et al. 2019), and Weather (Zhou et al. 2021). For short-term forecasting, we evaluate on five electricity price forecasting (EPF) datasets: NP, PJM, BE, FR, and DE (Lago et al. 2021). More details, such as the number of endogenous and exogenous variables, are provided in Appendix C.1.

Baselines We compare ExoTimer with 11 state-of-the-art (SOTA) time series models, and cite their performance from ExoLLM and TimeXer if applicable. Our baselines include LLM-based models: LLM4TS (Chang et al. 2023), GPT4TS (Zhou et al. 2023), TimeLLM (Jin et al. 2024), TimeCMA (Liu et al. 2025a), ExoLLM (Huang et al. 2025), Transformer-based models: TimeXer (Wang et al. 2024), PatchTST (Nie et al. 2023), iTransformer (Liu et al. 2024b), Crossformer (Zhang and Yan 2023), CNN-based models: TimesNet (Wu et al. 2023), and Linear-based model: TiDE (Das et al. 2023). Among these, TimeXer and TiDE are recently proposed deep learning methods designed for exogenous variables, while ExoLLM is a recent SOTA LLM-based forecasting model tailored for exogenous variables.

Setups For short-term forecasting, we set the look-back window size to 168, the prediction horizon to 24 following the configuration in NBEATSx (Olivares et al. 2023), and use a patch size of 24. For long-term forecasting, we consistently use a patch size of 16 with look-back window size of 96 and prediction horizons $\{96, 192, 336, 720\}$. We select GPT-2 as the LLM to generate the prompt embeddings. Details of the implementation are provided in Appendix C.4.

3.1 Main Results

Long-term forecasting The results in Table 1 demonstrate that the average performance of ExoTimer outperforms all baseline models in most cases. Notably, compared with ExoLLM, ExoTimer achieves superior performance with 13.3% and 8.9% improvements in MSE and MAE, respectively. Compared to TimeCMA, the latest cross-modal alignment model, ExoTimer demonstrates relative reductions of 40.7% and 24.8% in MSE and MAE, respectively. Compared with TimeXer, the current SOTA deep learning model with exogenous variables, ExoTimer improves performance in MSE and MAE by 24.1% and 13.1%, respectively.

Short-term forecasting As shown in Table 2, ExoTimer outperforms all baselines in all cases. Compared with ExoLLM, ExoTimer achieves reductions of 5.2% in MAE and 6.0% in MSE, respectively. Furthermore, ExoTimer surpasses both TimeXer and TiDE, which are specifically designed for forecasting with exogenous variables.

Few-shot forecasting In few-shot learning, only 10% of the training data are used. As shown in Table 3, ExoTimer outperforms all baseline methods. We attribute this to the effective knowledge activation via multi-attribute prompts. Specifically, ExoTimer achieves an average reduction of 11.8% in MSE and 7.9% in MAE compared with ExoLLM.

Zero-shot forecasting The results presented in Table 4 demonstrate that ExoTimer consistently outperforms the most competitive baseline model by a significant margin, achieving an improvement of 8.9% in MSE and 5.6% in MAE compared with ExoLLM. The results further indicate that ExoTimer can effectively leverage inherent knowledge in LLMs for time series forecasting while exhibiting strong generalization capabilities.

3.2 Ablation Study

Model Design We conduct ablation studies of model design by removing each module from ExoTimer on five datasets. **w/o Exo** does not integrate any exogenous information into endogenous temporal embedding. **w/o ExoSelect** employs all exogenous variables without selecting the important ones. **w/o Text2TsAlign** removes the text-temporal hash alignment and the corresponding temporal branch predictor. **w/o Ts2TextAlign** removes the temporal-text hash alignment and the corresponding textual branch predictor. **w/o Bi-HA** removes all alignments between temporal and textual modalities. **w/o AdaWeight** removes the adaptive weighting in the Dual-Branch Predictor. As shown in Table 5, ExoTimer exhibits the best performance compared to architectural variants across the five datasets. **w/o Bi-HA** has the most significant impact on prediction performance, highlighting the superiority of our LSH-based hash space retrieval for cross-modal alignment.

Prompt Design We evaluate four different prompt settings on ETTh1 and Weather datasets: (1) **Prompt-1** removes the context instruction prompt. (2) **Prompt-2** removes the task instruction prompt. (3) **Prompt-3** removes the time series analysis prompt. (4) **Prompt-4** transforms the numerical data of time series into texts as adopted in TimeCMA. The results in Fig. 2 demonstrate that the multi-attribute prompt design in ExoTimer can effectively activate the prior knowledge in LLMs compared to ablated prompt settings. Furthermore, our results indicate that directly using raw numerical data as prompts provides limited information and fails to utilize the textual understanding capability of LLMs. These findings underscore the importance of well-designed prompt engineering in unlocking the reasoning potential of LLMs for time series forecasting.

3.3 Model Analysis

Efficiency Analysis. We evaluate the parameter numbers and training time of ExoTimer on the ECL dataset (320 exogenous variables) compared with ten baseline models with the identical batch size 8 for a fair comparison. As illustrated in Fig. 3, ExoTimer achieves the best trade-off between accuracy and efficiency, attaining the lowest MSE with smaller parameter size and faster training speed among all compared models. This demonstrates the superiority of

Model	**ExoTimer**		ExoLLM		TimeXer		iTransformer		PatchTST		Crossformer		TiDE		TimesNet		TimeCMA		TimeLLM		GPT4TS		LLM4TS	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	**0.317**	**0.397**	0.330	0.404	0.327	0.408	0.365	0.442	0.394	0.446	0.344	0.412	0.419	0.468	0.410	0.476	0.477	0.529	0.365	0.413	0.392	0.442	0.378	0.427
Weather	**0.001**	**0.023**	0.001	0.027	0.002	0.031	0.002	0.031	0.002	0.031	0.005	0.055	0.002	0.029	0.097	0.115	0.002	0.029	0.003	0.036	0.005	0.056	0.004	0.046
ETTh1	**0.048**	**0.176**	0.069	0.205	0.073	0.209	0.075	0.211	0.078	0.215	0.285	0.447	0.083	0.223	0.076	0.215	0.087	0.227	0.104	0.277	0.126	0.305	0.115	0.304
ETTh2	**0.158**	**0.306**	0.175	0.327	0.189	0.342	0.199	0.352	0.192	0.345	1.027	0.873	0.205	0.356	0.210	0.362	0.230	0.377	0.226	0.388	0.277	0.443	0.251	0.415
ETTm1	**0.026**	**0.124**	0.049	0.165	0.052	0.171	0.053	0.175	0.053	0.173	0.411	0.548	0.053	0.173	0.054	0.175	0.055	0.176	0.080	0.233	0.106	0.264	0.093	0.248
ETTm2	**0.107**	**0.243**	0.113	0.249	0.120	0.258	0.127	0.267	0.120	0.258	0.976	0.769	0.122	0.261	0.129	0.271	0.132	0.276	0.162	0.311	0.196	0.349	0.179	0.330
Traffic	0.150	0.225	**0.145**	**0.220**	0.156	0.234	0.161	0.246	0.173	0.253	0.182	0.268	0.240	0.326	0.171	0.264	0.324	0.410	0.186	0.271	0.166	0.247	0.177	0.260

Table 1: Long-term forecasting with exogenous variables. The look-back window size L is 96 for all baselines. Results are averaged from all prediction horizons $T \in \{96, 192, 336, 720\}$. A lower value indicates better performance. The best results are highlighted in **bold** and the second best are underlined. The complete results are listed in the Appendix E.1.

Model	**ExoTimer**		ExoLLM		TimeXer		iTransformer		PatchTST		Crossformer		TiDE		TimesNet		TimeCMA		TimeLLM		GPT4TS		LLM4TS	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NP	**0.197**	**0.219**	0.216	0.234	0.236	0.268	0.265	0.300	0.267	0.284	0.240	0.285	0.335	0.340	0.250	0.289	0.309	0.321	0.477	0.434	0.275	0.303	0.265	0.315
PJM	**0.063**	**0.162**	0.076	0.175	0.093	0.192	0.097	0.197	0.106	0.209	0.101	0.199	0.124	0.228	0.097	0.195	0.108	0.215	0.154	0.268	0.118	0.207	0.255	0.308
BE	**0.335**	**0.213**	0.358	0.225	0.379	0.243	0.394	0.270	0.400	0.262	0.420	0.290	0.523	0.336	0.419	0.288	0.463	0.313	0.488	0.310	0.502	0.288	0.426	0.258
FR	**0.358**	**0.192**	0.365	0.203	0.385	0.208	0.439	0.233	0.411	0.220	0.434	0.208	0.510	0.290	0.431	0.234	0.429	0.260	0.526	0.260	0.570	0.497	0.519	0.459
DE	**0.413**	**0.393**	0.422	0.401	0.440	0.415	0.479	0.443	0.461	0.432	0.574	0.430	0.568	0.496	0.502	0.446	0.520	0.463	0.468	0.440	0.569	0.490	0.517	0.460
Avg	**0.273**	**0.236**	0.288	0.251	0.307	0.265	0.335	0.289	0.330	0.282	0.354	0.284	0.412	0.338	0.340	0.290	0.366	0.314	0.423	0.342	0.325	0.326	0.399	0.408

Table 2: Full results of the short-term forecasting with exogenous variables task on EPF dataset. The look-back window size and predict horizon are 168 and 24 respectively for all baselines. Avg means the average results from all five subsets.

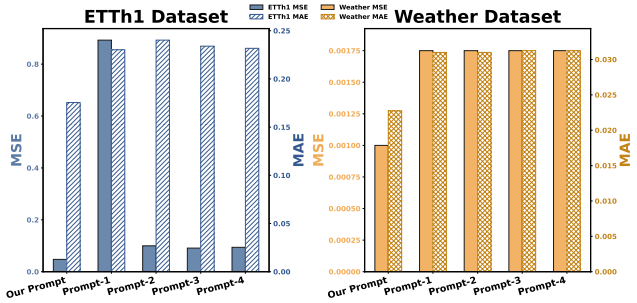

Figure 2: Model Performance with Different Prompt Settings.

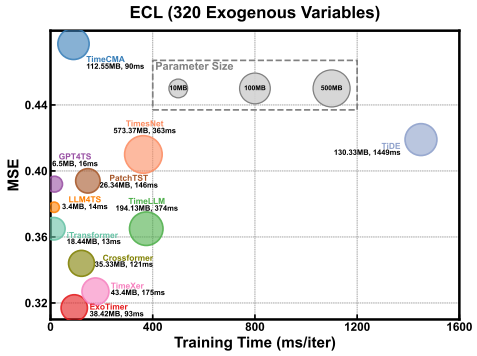

Figure 3: Model Efficiency Comparison on ECL Dataset.

ExoTimer for time series forecasting with high-dimensional exogenous variables.

Exogenous Variable Selection We visualize the masked attention scores of endogenous variables with different exogenous variables on the ETTh1 dataset. Fig. 4 demonstrates that ExoTimer is able to ignore irrelevant factors and focus on the most informative variables for prediction. Additionally, we observe that exogenous series with shapes and temporal dynamics similar to those of the endogenous series tend to receive greater attention.

Hash Space Visualization We perform representation analysis for text-temporal hash alignment using t-SNE. Fig. 5 (a) shows that textual embeddings exhibit broad dispersion with rich semantic diversity, while the temporal representations are highly clustered, indicating the heterogeneity gap between two modalities. After hashing, as shown in Fig. 5 (b), the textual embeddings become less dispersed, yet still present a relatively scattered distribution. The hash results suggest that ExoTimer preserves the most relevant semantics and informative representations. Fig. 5 (c) shows the pairwise cosine similarities between hash vectors of both modalities. In Fig. 5 (d), blue and green points denote hashed and fused temporal representations, respectively, with grey

Model	**ExoTimer**		ExoLLM		TimeXer		iTransformer		PatchTST		Crossformer		TiDE		TimesNet		TimeCMA		TimeLLM		GPT4TS		LLM4TS	
Dataset	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	**0.076**	**0.215**	0.084	0.230	0.094	0.248	0.091	0.251	0.153	0.344	0.346	0.506	0.126	0.312	0.104	0.251	0.101	0.246	0.101	0.251	0.095	0.242	0.140	0.342
ETTh2	**0.210**	**0.361**	0.253	0.403	0.279	0.435	0.290	0.439	0.401	0.546	1.501	1.080	0.327	0.478	0.363	0.487	0.325	0.454	0.298	0.439	0.278	0.425	0.364	0.512
ETTm1	**0.052**	**0.171**	0.057	0.181	0.062	0.194	0.062	0.190	0.124	0.290	0.475	0.601	0.094	0.256	0.059	0.185	0.057	0.182	0.062	0.190	0.062	0.190	0.109	0.273
ETTm2	**0.127**	**0.265**	0.144	0.291	0.156	0.310	0.163	0.306	0.253	0.410	1.187	0.882	0.209	0.365	0.163	0.309	0.138	0.285	0.158	0.306	0.155	0.301	0.231	0.388

Table 3: Few shot learning on 10% training data. The look-back window size L is 96 for all baselines. Results are averaged from all prediction horizons $T \in \{96, 192, 336, 720\}$. The complete results are listed in the Appendix E.2.

Model	**ExoTimer**		ExoLLM		TimeXer		iTransformer		PatchTST		Crossformer		TiDE		TimesNet		TimeCMA		TimeLLM		GPT4TS		LLM4TS	
Metric	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→ETTh2	**0.190**	**0.345**	0.204	0.359	0.228	0.390	0.221	0.395	0.380	0.544	0.875	0.796	0.308	0.490	0.252	0.400	0.249	0.399	0.248	0.394	0.232	0.381	0.344	0.538
ETTh2→ETTh1	**0.072**	**0.206**	0.074	0.212	0.082	0.228	0.085	0.230	0.118	0.287	0.429	0.562	0.096	0.251	0.094	0.238	0.105	0.250	0.087	0.230	0.082	0.223	0.107	0.269
ETTm1→ETTm2	**0.126**	**0.264**	0.162	0.309	0.177	0.332	0.178	0.324	0.353	0.495	1.348	1.025	0.267	0.437	0.278	0.409	0.213	0.358	0.176	0.324	0.178	0.324	0.310	0.466
ETTm2→ETTm1	**0.052**	**0.174**	0.054	0.176	0.058	0.187	0.061	0.185	0.094	0.248	0.455	0.538	0.078	0.220	0.060	0.185	0.059	0.183	0.059	0.185	0.058	0.182	0.086	0.234

Table 4: Zero-shot learning. The look-back window size L is 96 for all baselines. Results are averaged from all prediction horizons $T \in \{96, 192, 336, 720\}$. The complete results are listed in the Appendix E.3.

Dataset	**ECL**		**Weather**		**ETTh1**		**ETTm1**		**PJM**	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ExoTimer	**0.317**	**0.397**	**0.001**	**0.023**	**0.048**	**0.176**	**0.026**	**0.124**	**0.063**	**0.162**
w/o Exo	0.392	0.446	0.002	0.031	0.084	0.223	0.057	0.177	0.101	0.203
w/o ExoSelect	0.400	0.456	0.002	0.031	0.090	0.231	0.054	0.174	0.127	0.233
w/o Text2TsAlign	0.394	0.451	0.002	0.031	0.104	0.234	0.053	0.173	0.110	0.211
w/o Ts2TextAlign	0.379	0.440	0.002	0.031	0.088	0.229	0.054	0.174	0.107	0.208
w/o Bi-HA	0.994	0.802	0.002	0.031	0.083	0.226	0.058	0.185	0.270	0.371
w/o AdaWeight	0.382	0.442	0.002	0.031	0.084	0.224	0.054	0.174	0.114	0.210

Table 5: Ablation study of model design. Results are averaged from all prediction horizons $T \in \{96, 192, 336, 720\}$ for ECL, Weather, ETTh1, ETTm1, and $\{24\}$ for PJM. The full results are listed in the Appendix F.

arrows indicating the fusion process. The fused results validate that our cross-modal alignment method effectively enriches temporal features and produces semantically consistent representations.

4 Conclusion

This paper proposes ExoTimer, a promising framework that leverages large language models (LLMs) for time series forecasting with exogenous variables. An Exo-Aware Endogenous Encoder is employed to identify important exogenous variables, and generate comprehensive global embeddings and patch-level representations for endogenous variables. A Multi-Attribute Prompt Embedding module is designed to activate prior knowledge in LLMs. Moreover, a cross-modality alignment method based on Locality Sensitive Hashing (LSH) is proposed to bridge the heterogeneity gap between textual and temporal modalities. Extensive experiments demonstrate that ExoTimer achieves state-of-the-art performance on both long-term and short-term forecasting tasks, and exhibits strong generalizability and scalability in both few-shot and zero-shot scenarios. Our results also provide novel insights into the importance of well-designed prompts in unlocking the reasoning potential of LLMs, as well as the effectiveness and efficiency of the LSH-based modality alignment method.

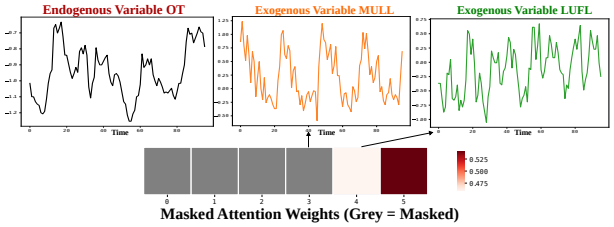

Figure 4: Visualization of Exogenous Variable Selection.

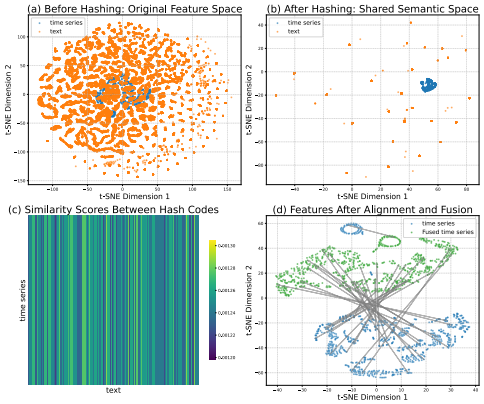

Figure 5: Visualization of Text-temporal Alignment.

References

Ba, J. L.; Kiros, J. R.; Hinton, G. E.; et al. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

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

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

Das, A.; Kong, W.; Leach, A.; Sen, R.; and Yu, R. 2023. Long-term Forecasting with TiDE: Time-series Dense Encoder. *Transactions on Machine Learning Research*.

Gruver, N.; Finzi, M.; Qiu, S.; and Wilson, A. G. 2023. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems*.

Hastie, T.; Tibshirani, R.; Friedman, J.; et al. 2009. *The Elements of Statistical Learning*. Springer.

Huang, Q.; Shen, L.; Zhang, R.; Ding, S.; Wang, B.; Zhou, Z.; and Wang, Y. 2023. CrossGNN: confronting noisy multivariate time series via cross interaction refinement. In *International Conference on Neural Information Processing Systems*.

Huang, Q.; Zhou, Z.; Yang, K.; Lin, G.; Yi, Z.; and Wang, Y. 2024. Leret: Language-empowered retentive network for time series forecasting. In *International Joint Conference on Artificial Intelligence*.

Huang, Q.; Zhou, Z.; Yang, K.; and Wang, Y. 2025. Exploiting Language Power for Time Series Forecasting with Exogenous Variables. In *International World Wide Web Conference*, 4043–4052.

Indyk, P.; and Motwani, R. 1998. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, 604–613. ACM.

Jang, E.; Gu, S.; Poole, B.; et al. 2017. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

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

Kingma, D. P.; and Ba, J. 2014. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*.

Koltchinskii, V.; and Panchenko, D. 2002. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1): 1–50.

Lago, J.; Marcjasz, G.; De Schutter, B.; and Weron, R. 2021. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293: 116983.

Li, S.; Jin, X.; Xuan, Y.; Zhou, X.; Chen, W.; Wang, Y.-X.; and Yan, X. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in Neural Information Processing Systems*.

Liu, C.; Xiao, Z.; Wang, D.; Cheng, M.; Chen, H.; and Cai, J. 2022. Foreseeing private car transfer between urban regions with multiple graph-based generative adversarial networks. *World Wide Web: Internet and Web Information Systems*, 25(6): 2515–2534.

Liu, C.; Xu, Q.; Miao, H.; Yang, S.; Zhang, L.; Long, C.; Li, Z.; and Zhao, R. 2025a. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *Association for the Advancement of Artificial Intelligence*, volume 39, 18780–18788.

Liu, C.; Zhou, S.; Xu, Q.; Miao, H.; Long, C.; Li, Z.; and Zhao, R. 2025b. Towards Cross-Modality Modeling for Time Series Analytics: A Survey in the LLM Era.

Liu, H.; Ma, Z.; Yang, L.; Zhou, T.; Xia, R.; Wang, Y.; Wen, Q.; and Sun, L. 2023. SADI: A Self-Adaptive Decomposed Interpretable Framework for Electric Load Forecasting Under Extreme Events. In *IEEE International Conference on Acoustics, Speech and Signal Processing*.

Liu, P.; Guo, H.; Dai, T.; Li, N.; Bao, J.; Ren, X.; Jiang, Y.; and Xia, S.-T. 2025c. Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. In *Association for the Advancement of Artificial Intelligence*, volume 39, 18915–18923.

Liu, Q.; Liu, X.; Liu, C.; Wen, Q.; and Liang, Y. 2024a. Time-ffm: Towards lm-empowered federated foundation model for time series forecasting. volume 37, 94512–94538.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024b. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations*.

Lu, J.; Han, X.; Sun, Y.; and Yang, S. 2024. Cats: Enhancing multivariate time series forecasting by constructing auxiliary time series as exogenous variables.

Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; and Wang, F.-Y. 2014. Traffic flow prediction with big data: A deep learning approach. *Ieee transactions on intelligent transportation systems*, 16(2): 865–873.

Mahalanobis, P. C. 1936. On the Generalized Distance in Statistics. *Proceedings of the National Institute of Sciences of India*, 2(1): 49–55.

Massart, P. 2000. About the Constants in Talagrand's Concentration Inequalities for Empirical Processes. *Annals of Probability*, 28(2): 863–884.

Miao, H.; Zhao, Y.; Guo, C.; Yang, B.; Kai, Z.; Huang, F.; Xie, J.; and Jensen, C. S. 2024. A unified replay-based continuous learning framework for spatio-temporal prediction on streaming data. In *IEEE International Conference on Data Engineering*.

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

Niu, T.; Wang, J.; Lu, H.; Yang, W.; and Du, P. 2020. Developing a deep learning framework with two-stage feature selection for multivariate financial time series forecasting. *Expert Systems with Applications*, 148: 113237.

Olivares, K. G.; Challu, C.; Marcjasz, G.; Weron, R.; and Dubrawski, A. 2023. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*.

Pan, Z.; Jiang, Y.; Garg, S.; Schneider, A.; Nevmyvaka, Y.; and Song, D. 2024. S^2IP-LLM: Semantic Space Informed Prompt Learning with LLM for Time Series Forecasting. In *International Conference on Machine Learning*.

Schneider, S. H.; and Dickinson, R. E. 1974. Climate modeling. *Reviews of Geophysics*, 12(3): 447–493.

Sun, C.; Li, H.; Li, Y.; and Hong, S. 2024. Test: Text prototype aligned embedding to activate llm's ability for time series.

Vagropoulos, S. I.; Chouliaras, G.; Kardakos, E. G.; Simoglou, C. K.; and Bakirtzis, A. G. 2016. Comparison of SARIMAX, SARIMA, modified SARIMA and ANN-based models for short-term PV generation forecasting. In *ENERGYCON*.

Vapnik, V. N.; and Chervonenkis, A. Y. 1971. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability Its Applications*, 16(2): 264–280.

Vaswani, A.; Shazeer, N. M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *International Conference on Neural Information Processing Systems*, 6000–6010.

Wang, Y.; Wu, H.; Dong, J.; Qin, G.; Zhang, H.; Liu, Y.; Qiu, Y.; Wang, J.; and Long, M. 2024. Timexer: Empowering transformers for time series forecasting with exogenous variables. volume 37, 469–498.

Williams, B. M. 2001. Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. *Transportation Research Record*.

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

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

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Association for the Advancement of Artificial Intelligence*, volume 35, 11106–11115.

Zhou, T.; Niu, P.; Sun, L.; Jin, R.; et al. 2023. One fits all: Power general time series analysis by pretrained lm. In *Advances in Neural Information Processing Systems*.

Reproducibility Checklist

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) yes

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) yes

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) yes

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) no

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) Type your response here

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) Type your response here

2.4. Proofs of all novel claims are included (yes/partial/no) Type your response here

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) Type your response here

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) Type your response here

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) Type your response here

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) Type your response here

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) yes

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) yes

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) NA

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) NA

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) yes

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) yes

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) NA

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) yes

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) yes

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) no

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) partial

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) yes

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) yes

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) yes

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) yes

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) yes

4.10. This paper states the number of algorithm runs used

to compute each reported result (yes/no) yes

4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) yes

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) no

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) yes