A novel attention-enhanced LLM approach for accurate power demand and generation forecasting

Zehuan Hu[a], Yuan Gao[b,*], Luning Sun[a], Masayuki Mae[a]

[a]*Department of Architecture, Graduate School of Engineering, The University of Tokyo, Tokyo, Japan*
[b]*International Institute for Carbon-Neutral Energy Research (WPI-I2CNER); Kyushu University, 744 Motooka, Nishi-ku, Fukuoka-shi, Fukuoka, 819-0395, Japan*

Abstract

Accurate forecasting of electricity demand and generation is crucial for efficient grid management and sustainable energy planning. While large language models (LLM) have shown promise in various fields, their application to time series forecasting presents challenges, including limited cross-channel information capture and the complexity of prompt design. In this study, we propose a novel framework that combines multiple attention mechanisms with LLM, enabling effective feature extraction from both target and non-target variables without the need for prompt engineering. We conducted extensive experiments using real-world electricity demand and generation data from multiple regions in Japan to evaluate the proposed model. The results demonstrate that our model outperforms state-of-the-art LLM-based and other time-series forecasting models in terms of electricity demand and generation forecast task, achieving better performance than the latest LLM-based models without using prompts or increasing model size. Compared with the Long short-term memory network (LSTM), the mean absolute error (MAE) is reduced by 20.8%. Compared with the previous time-series LLM, the proposed model reduces memory usage by 49.3% and shortens training time by 35.7%. Additionally, the proposed model exhibits superior generalization ability, maintaining high performance even in

*Corresponding author
Email address: yuangao1120@gmail.com (Yuan Gao)

zero-shot learning scenarios. Compared with the LSTM, MAE on the four test datasets is reduced by 16.6%.

Keywords: Large language model; Multi-Attention mechanism; Electricity demand and generation forecasting; Zero-shot learning

1. Introduction

1.1. Background

In recent years, global electricity consumption has surged due to rapid urbanization, industrialization, and the proliferation of digital technologies [1]. This increasing demand for power places immense pressure on energy providers to meet consumption needs while adhering to sustainability goals [2]. With the looming threat of climate change, governments and organizations worldwide are focusing on reducing carbon emissions and transitioning toward cleaner energy sources [3, 4]. Achieving these ambitious targets requires not only the development of renewable energy infrastructures but also the implementation of smart energy management systems that can efficiently balance consumption with production [5, 6].

Accurate forecasting of electricity demand and generation is crucial to ensuring a stable and efficient power grid [7]. Mismatches between power supply and demand can lead to significant operational challenges, including grid instability, power outages, and economic losses [8]. Traditional forecasting methods, while effective to some extent, struggle to capture the complexity of modern power systems, which are influenced by numerous factors such as fluctuating renewable energy outputs, weather conditions, and dynamic market demands [9]. As a result, there is an increasing need for advanced predictive models that can enhance the accuracy and reliability of electricity forecasting, enabling more proactive energy management and decision-making [10].

1.2. Electricity demand and generation forecasting

Currently, a variety of methods are employed for electricity demand and generation forecasting, ranging from traditional statistical approaches to more

advanced machine learning techniques [11]. Classical methods, such as autoregressive integrated moving average (ARIMA) and exponential smoothing, have been widely used due to their simplicity and effectiveness in modeling time series data methods often assume linear relationships and may struggle to capture the non-linear complexities of modern energy systems [12]. On the other hand, machine learning methods, such as support vector machines (SVM) and random forests, have been introduced to address these limitations by modeling non-linear relationships more effectively [13]. Despite their statistical methods, these approaches still require extensive feature engineering and may not always generalize well to unseen data [14].

In recent years, deep learning models, particularly artificial neural networks (ANN) and their variants, have gained significant attention in electricity forecasting due to their ability to learn complex patterns directly from raw data [15]. Recurrent neural networks (RNN), especially long short-term memory (LSTM) networks, have shown great promise in time series forecasting as they can capture temporal dependencies in the data [16, 17]. Additionally, hybrid models that combine LSTM or other techniques, such as attention mechanisms are becoming increasingly popular due to their enhanced prediction performance and adaptability to complex energy systems [18, 19]. A summary and review of relevant research on different machine learning methods of electricity demand forecasting is shown in Table 1.

Table 1: Literature review for electricity demand & generation forecasting.

Ref.	**Methods**	**Datasets**
[20]	ANN	Electricity demand data of Turkey
[21]	Autoregressive	Electricity demand data from the Nordic electricity market
[22]	LSTM; SVM	Electrical load data; weather factor data of school building

Ref.	Methods	Datasets
[23]	Classical statistical; autoregressive models	Electrical demand data of Ukraine
[24]	K-nearest neighbor (KNN), linear regression, random forest (RF); SVM	Demand and renewable power generation data from South African electricity public utility company
[25]	Transformer-based model	Electricity demand data of one city in China
[26]	Elman neural network	Power consumption statistics of Serbia
[27]	Hybrid deep learning	Four buildings data from Mendeley and Kaggle
[28]	ANN	Meteorological, operational, and economic data from Mexico
[29]	Deep-Autofomer	Microgrid system data of 24 families
[30]	CNN-LSTM	Load data from individual household
[31]	KNN; SVM; RF; ANN	Electricity demand data of ow-energy house

1.3. Large Language Models for time-series forecasting

Large language models (LLM), such as GPT (Generative Pretrained Transformer) and BERT (Bidirectional Encoder Representations from Transformers), have revolutionized numerous fields, including natural language processing (NLP), machine translation, and text generation [32]. These models, characterized by their vast number of parameters and ability to process massive datasets, have shown remarkable capabilities in understanding and generating human-like

text [33]. Beyond traditional NLP tasks, LLMs have been increasingly adopted in domains like healthcare, law, and finance, where complex, context-dependent decision-making is required [34]. The rapid advancements in transformer architectures, which LLMs are based on, have made it possible to capture intricate patterns in sequential data, offering potential applications in areas that extend beyond text [35].

Recent studies have started exploring the application of LLMs for time series forecasting, leveraging their ability to model sequential dependencies and long-range correlations effectively [36]. Traditional time series models, such as ARIMA or LSTM, are often limited in their ability to capture the full context of long sequences[37]. LLMs address these limitations by allowing parallel processing of sequential data and better handling of long-term dependencies [38] Research has demonstrated that LLM-based approaches, such as Time-series Transformers, can outperform classical methods in various forecasting tasks.

At present, approaches to employing LLMs in time series forecasting can be broadly classified into two categories. The first relies on prompt-based reprogramming of pretrained LLMs, whereby carefully crafted instructions are prepended to raw time series to steer the model's predictions [39], [40]. Inspired by their remarkable conversational capabilities, these methods leverage the LLM's inherent pattern-recognition power to interpret temporal dynamics without any weight updates [41]. The second paradigm involves fine-tuning the LLM itself on time series data [42][43]. Chang et al. [44] subsequently proposed a two-stage fine-tuning pipeline—first aligning the backbone network to time series representations, then training a lightweight prediction head. Sun et al. [45] introduced contrastive learning objectives to sharpen the model's temporal embeddings. A summary and review of relevant research on LLM for time-series forecasting is shown in Table 2.

Table 2: Literature review for time-series forecasting by LLM.

Ref.	Target	Approaches to employing LLMs
[46]	Energy load	Directly use with predefined template prompt
[47]	Wind speed	Directly use with spatial prompts and temporal prompts
[48]	CNY-USD exchange rate	Directly use with prompt adjusted by reinforcement learning with human feedback
[49]	Wind power	Fine-tuning
[50]	NASDAQ-100 stock price	Directly use with predefined template prompt $ fine-tuning
[51]	Energy load	Fine-tuning
[52]	Climate	Directly use with predefined template prompt
[53]	Wind power	Directly use with proposed soft and hard prompts
[54]	Methanol price	Fine-tuning
[55]	Stock return	Fine-tuning

1.4. Research contribution

Despite some initial applications of LLMs in the field of time series forecasting, significant challenges and limitations remain:

1) Pretrained LLMs have already undergone extensive training and validation, so naive fine-tuning often yields only marginal gains and can even degrade performance due to overfitting; moreover, the multiple modules within modern LLM architectures make identifying which components to fine-tune a nontrivial challenge.

2) While prompt design has been widely used in other LLMs applications to enhance model performance, designing effective prompts for time se-

ries forecasting is particularly challenging. Due to the diverse nature of datasets and the complexity involved in describing historical feature relationships and interdependencies, creating a simple and generalized prompt for time series prediction tasks remains elusive.

3) LLM applications in electricity demand and generation forecasting are still largely unexplored. Given the multivariate nature of electricity data, which typically involves numerous features, developing an LLM-based model that can efficiently handle and predict these dynamics is a significant challenge.

4) Japan spans a vast distance from north to south, and there are significant regional differences in power generation, electricity demand, and climate. These variations place high demands on the generalization ability of models. However, there is limited research focused on Japan's electricity data.

To address these challenges and fill the research gaps, this study proposes a novel LLM-based model for forecasting electricity demand and generation. The key contributions of this research are as follows:

1) To enhance the model's ability to understand the relationships between different features, we design a specialized framework that allows the LLM to extract meaningful interdependencies between the target variables and other features in a simple yet effective manner.

2) We introduce a preprocessing method based on cross-attention that eliminates the need for prompt engineering, enabling multiple time series of target features to be transformed and input into the LLM simultaneously. This approach simplifies the application of LLMs to multivariate time series forecasting.

3) To validate the effectiveness and generalizability of the proposed framework, we conduct extensive experiments using historical electricity demand, generation, and weather data from four different regions in Japan.

2. Methodology

2.1. Related works and Challenges

In the field of time series forecasting, there are currently two mainstream training strategies: Channel Independent (CI) (Fig. 1 (a)) and Channel Dependent (CD) (Fig. 1 (b)) approaches [56]. Each of these strategies has its strengths and limitations. The CD strategy theoretically offers higher modeling capacity because it allows the model to learn complex relationships between different variables. In contrast, the CI strategy is more robust, as it focuses on the self-correlation characteristics within each channel, making it less sensitive to noise and more generalizable [57].

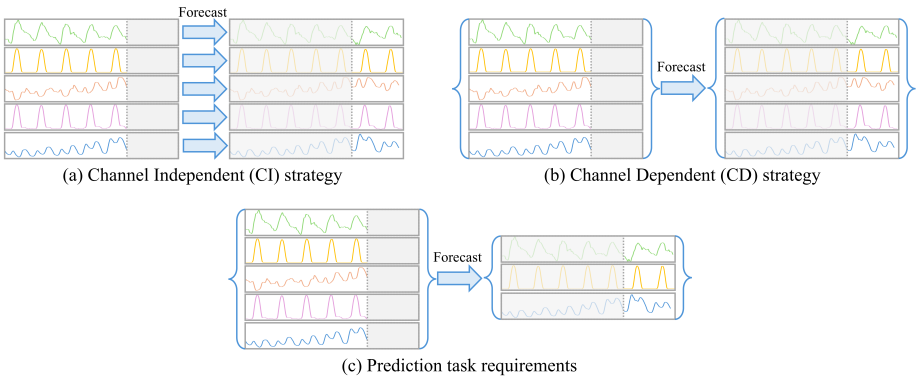


Figure 1: Training strategies and prediction requirement.

Meanwhile, in practical engineering applications, forecasting tasks often require using multiple input variables to predict a subset of target variables, such as predicting future power generation and electricity demand using historical power and weather data (Fig. 1 (c)). Unlike tasks that require predictions for all variables or single-variable forecasting, these tasks benefit from the model's ability to consider the relationships between variables without needing to predict every variable. However, considering the large scale of modern LLMs and their substantial computational costs, most existing applications of LLMs to time-series forecasting adopt a channel-independent training strategy [36, 43]. This approach computes a loss for every variable—even when only a subset

of variables requires prediction—resulting in wasted training resources. Conversely, restricting the input to only the target variables prevents the model from leveraging information in other channels, which can lead to reduced forecasting accuracy.

Furthermore, prompt engineering plays a crucial role in optimizing the performance of LLMs, expanding their application range, and improving interaction efficiency. For example, Cao et al. [41] introduced a Semi-Soft Prompt strategy, where prompts are divided into explicit text-based prompts (hard prompts) and vector-based prompts (soft prompts). The authors propose a semi-soft prompting strategy that generates distinct prompts corresponding to key time series components: trend, seasonality, and residuals. Jin et al. [36] attached prompts as prefixes to the input time series, providing background information, task instructions, and data statistics. However, in practical applications, different forecasting tasks and time series often have unique characteristics, making it challenging to design effective prompts tailored for each scenario. This greatly increases the complexity of utilizing LLMs in time series forecasting.

2.2. Multi-attention large language model (MultiAttLLM)

To address these challenges, we propose a novel model that combines the advantages of CI and CD strategies while eliminating the need for prompt design, which called Multi-attention large language model (MultiAttLLM), as shown in Fig. 2. In this model, the LLM focuses on the self-correlations of target variables during the initial modeling phase, while cross-channel relationships are considered in a subsequent stage after the LLM output. This approach not only reduces the computational resources required for training but also enhances the model's ability to capture complex interactions between variables, resulting in improved forecasting performance. The architecture is composed of six main components:

① Word Projection: The first component of the model is the word projection layer, which addresses the issue of vocabulary redundancy in large

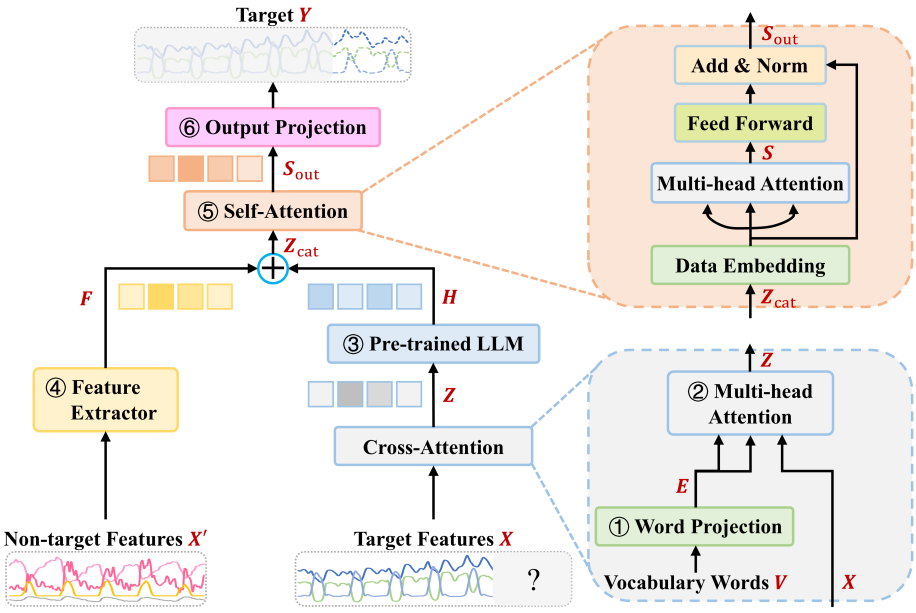


Figure 2: Topological general structure of the proposed LLM framework.

language models when applied to time series forecasting. Since the original vocabulary of LLM contains a significant amount of words unrelated to the context of time series data, the word projection layer maps the original vocabulary to a smaller, domain-specific word vector set. This allows the model to focus on the words and representations most relevant to describing time series, improving its efficiency and accuracy in forecasting tasks.

In this layer, we map the original LLM vocabulary embeddings $\boldsymbol{V} \in \mathbb{R}^{|\boldsymbol{V}| \times d_{\text{orig}}}$ into a smaller, domain-specific space of dimension d_{wproj}, and the output $\boldsymbol{E}$ of word projection layer can be calculated as follow:

$$\boldsymbol{E} = \boldsymbol{V}\boldsymbol{W}_{\text{wproj}} + \boldsymbol{b}_{\text{wproj}}, \quad \boldsymbol{W}_{\text{wproj}} \in \mathbb{R}^{d_{\text{orig}} \times d_{\text{wproj}}}, \quad \boldsymbol{b}_{\text{wproj}} \in \mathbb{R}^{d_{\text{wproj}}} \tag{1}$$

where d_{orig} is the dimension of original LLM vocabulary; d_{wproj} is the dimension of vocabulary after word projection, which is 2000 in this study.

② Cross-Attention: To enable the large language model to capture the rela-

tionships between different features and convert the time series data into a format that can be understood by the LLM, we employed a cross-attention module (Fig. 3). This module integrates the features by using a query-key-value (QKV) structure, where each feature can attend to all other features. This allows the model to capture complex interdependencies between the target features and other input variables, effectively bridging the gap between time series data and natural language processing.

In this study, the target feature is used as Q, and the word vectors are used as K and V. Given an input feature matrix $X \in \mathbb{R}^{T \times f}$ (where T is sequence length, f is number of features), we compute:

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_q, \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_k, \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_v \tag{2}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ represent the query, key, and value matrices respectively; $\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v \in \mathbb{R}^{f \times d}$, represent the learnable weights. The attention map and output are

$$A(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\boldsymbol{Q}\boldsymbol{K}^T/\sqrt{d_k})\boldsymbol{V} \tag{3}$$

$$\boldsymbol{Z} = A(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})\boldsymbol{W}_o, \quad \boldsymbol{W}_o \in \mathbb{R}^{d \times d}, \quad \boldsymbol{Z} \in \mathbb{R}^{T \times d} \tag{4}$$

where the attention mechanism is applied to each time step in the input sequence; d_k is the dimension of queries, keys and values, which used to reduce the impact of the input data dimension on the results.

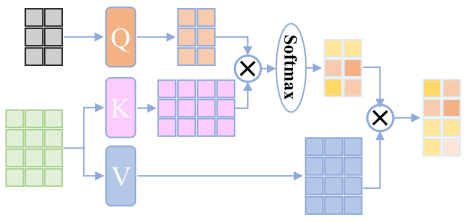


Figure 3: Topological general structure of attention mechanism.

③ Frozen pre-trained LLM: After cross-attention block is a pre-trained large language model. By keeping the LLM intact, we maintain the rich lan-

guage understanding capabilities of the pre-trained model while focusing on effective reprogramming of the input to align with its strengths.

In this layer, we feed the cross-attention outputs $\boldsymbol{Z}$ (after adding positional encodings) into a pre-trained LLM of L layers:

$$\boldsymbol{H} = \mathrm{LLM}_{\mathrm{frozen}}(\boldsymbol{Z} + \mathrm{PE}), \quad \boldsymbol{H} \in \mathbb{R}^{T \times d} \tag{5}$$

where no weights inside the LLM are updated; we merely reprogram its input to our time-series format.

④ Feature Extractor: The feature extractor module is responsible for capturing the relationships between non-target features, providing auxiliary information that helps improve the overall forecasting performance. This module can use various techniques, such as RNN or self-attention layers, to extract these relationships. In this study, we opt for a simple linear layer to validate the effectiveness of the proposed algorithm without introducing unnecessary complexity. This feature extraction helps the model understand the dynamics of auxiliary variables, which can influence the target features.

To model non-target feature interactions, we apply a simple linear extractor:

$$\boldsymbol{F} = \boldsymbol{X}'\boldsymbol{W}_f + \boldsymbol{b}_f, \quad \boldsymbol{W}_f \in \mathbb{R}^{f \times d}, \quad \boldsymbol{b}_f \in \mathbb{R}^d, \quad \boldsymbol{F} \in \mathbb{R}^{T \times d} \tag{6}$$

This provides auxiliary embeddings capturing global feature dynamics.

⑤ Self-Attention: The self-attention module is designed to enhance the understanding of the relationships between the target feature and the other input features. By applying self-attention, the model can learn the importance of each feature in relation to the target, thus improving its predictive accuracy. The self-attention mechanism assigns weights to each feature, allowing the model to focus more on the features that have a significant impact on the target. The calculation for the self-attention layer is the same as the cross-attention layer. The difference here is that we merge the

outputs of the Feature Extractor layer and pre-trained LLM layer along the feature dimension, and then use them as the query, key, and value inputs to compute self-attention. In addition, this layer also includes a feedforward component. In the feedforward layer, a feedforward neural network processes the output embeddings $\boldsymbol{Z}$ from the self-attention mechanism to generate the final output of the transformer model.

In this layer, we concatenate $\boldsymbol{H}$ and $\boldsymbol{F}$ along the feature dimension to form $\boldsymbol{Z}_{\text{cat}} \in \mathbb{R}^{T \times 2d}$, then project back to d:

$$\boldsymbol{Z}' = \boldsymbol{Z}_{\text{cat}} \boldsymbol{W}_{\text{cat}}, \quad \boldsymbol{W}_{\text{cat}} \in \mathbb{R}^{2d \times d} \tag{7}$$

and compute standard self-attention:

$$\boldsymbol{Q}' = \boldsymbol{Z}' \boldsymbol{W}_q, \quad \boldsymbol{K}' = \boldsymbol{Z}' \boldsymbol{W}_k, \quad \boldsymbol{V}' = \boldsymbol{Z}' \boldsymbol{W}_v, \tag{8}$$

$$\boldsymbol{A}' = \text{softmax}\left(\frac{\boldsymbol{Q}' \boldsymbol{K}'^{\top}}{\sqrt{d}} \right), \quad \boldsymbol{S} = \boldsymbol{A}' \boldsymbol{V}' \tag{9}$$

followed by a two-layer feedforward network:

$$FFN(\boldsymbol{S}) = \max(0, \boldsymbol{S}\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2, \quad \boldsymbol{W}_1 \in \mathbb{R}^{d \times d_{ff}}, \quad \boldsymbol{W}_2 \in \mathbb{R}^{d_{ff} \times d} \tag{10}$$

where $\boldsymbol{W}_1$, $\boldsymbol{b}_1$, $\boldsymbol{W}_2$, and $\boldsymbol{b}_2$ represent learnable weights and biases.

⑥ Output Projection: The output projection layer consists of two linear layers and serves to convert the final output of the self-attention module into the desired feature dimension and sequence length.

In last layer, we map the self-attention output $\boldsymbol{S}_{\text{out}}$ back to the original target dimension f_{out}:

$$\boldsymbol{Y} = \boldsymbol{S}_{\text{out}} \boldsymbol{W}_{\text{out}} + \boldsymbol{b}_{\text{out}}, \quad \boldsymbol{W}_{\text{out}} \in \mathbb{R}^{d \times f_{\text{out}}}, \quad \boldsymbol{b}_{\text{out}} \in \mathbb{R}^{f_{\text{out}}}, \quad \boldsymbol{Y} \in \mathbb{R}^{T \times f_{\text{out}}} \tag{11}$$

2.3. Benchmark

To validate the effectiveness and generalization ability of our proposed method, we selected five baseline models, including one traditional models, LSTM, a

simple linear model, DLinear, two latest novel transformer-based model, iTransformer and TimesNet, and a novel LLM-based model, TimeLLM.

2.3.1. Long-short term memory network (LSTM)

LSTM is a type of RNN designed to overcome the vanishing gradient problem that often occurs in traditional RNN when learning long-term dependencies in sequential data [58]. LSTM incorporates memory cells and gating mechanisms—input, forget, and output gates—that regulate the flow of information within the network. These gates allow LSTM models to selectively retain or forget information over time, making them highly effective for time series forecasting tasks that require capturing both short-term and long-term patterns in the data. The detail content and calculation equations of LSTM are provided in Appendix A.1 because of space constraints.

2.3.2. DLinear

Decomposition Linear (DLinear) is a simple linear model designed specifically for time series forecasting [59]. It improves forecasting accuracy by decomposing time series. Unlike the currently popular complex Transformer-based models, DLinear achieves excellent performance by processing the trend and cyclical components in time series through simple linear layers.

2.3.3. Informer

Informer is an efficient Transformer variant specifically designed for long-sequence time-series forecasting [18]. It introduces a ProbSparse self-attention mechanism that selects only the most informative query–key pairs, reducing the quadratic complexity of standard attention. To further accelerate processing, Informer employs a self-attention distilling operation that progressively shortens the sequence at intermediate layers, allowing it to scale to very long input horizons while maintaining high accuracy.

2.3.4. Autoformer

Autoformer advances Transformer architectures by embedding series decomposition directly into each model block [19]. It splits the input into trend and seasonal components using a learnable decomposition layer, then applies an auto-correlation mechanism to capture period-aware dependencies. This design both denoises the input and enables the model to learn long-term temporal patterns more effectively, yielding superior performance on long-horizon forecasting tasks.

2.3.5. iTransformer

iTransformer is a novel adaptation of the traditional Transformer architecture specifically designed for time series forecasting [60]. Unlike conventional Transformer-based models that process temporal tokens (where each token corresponds to a time step with multiple feature variables), the iTransformer adopts an inverted approach. It treats each time series as a separate variate token and uses self-attention mechanisms to capture the interdependencies between these variate tokens. This design helps the model more effectively learn multivariate correlations, making it particularly well-suited for time series data with complex feature relationships.

2.3.6. TimesNet

TimesNet is a novel model designed for general time series analysis by transforming one-dimensional (1D) time series data into two-dimensional (2D) tensors [17]. Traditional time series models often struggle with capturing intricate temporal variations because of the limitations of processing 1D data directly. TimesNet tackles this by leveraging the concept of multi-periodicity in time series, where complex variations occur both within and between periods. To better represent these temporal variations, TimesNet reshapes 1D time series into 2D tensors, where intraperiod-variations are represented by columns and interperiod-variations by rows.

2.3.7. TimeLLM

TimeLLM is a novel framework designed to adapt LLM for time series forecasting by reprogramming the input time series data [36]. Instead of fine-tuning the pre-trained LLM or altering their internal architectures, TimeLLM leverages a reprogramming technique that transforms the time series data into a format compatible with LLM. This is achieved by converting time series into text-like prototypes that the LLM can understand. To further enhance the model's reasoning capabilities, TimeLLM incorporates a "Prompt-as-Prefix" (PaP) approach, which enriches the input data with additional context, such as task-specific instructions and domain knowledge, allowing the LLM to better interpret and predict time series trends.

2.4. Model setup

For all baseline models except the LSTM, hyperparameters were set to the optimal values reported in their original publications. For the LSTM and our proposed model, to assess model robustness and determine optimal hyperparameter settings, we performed sensitivity analyses on both the LSTM baseline and our MultiAttLLM framework.

- LSTM: We evaluated combinations of layer depth $L \in \{1, 2, 3, 4\}$ and hidden-state dimension $h \in \{64, 128, 256, 512\}$. For each configuration, we recorded MSE on the validation set. The resulting performance grid is plotted in Fig. 4.

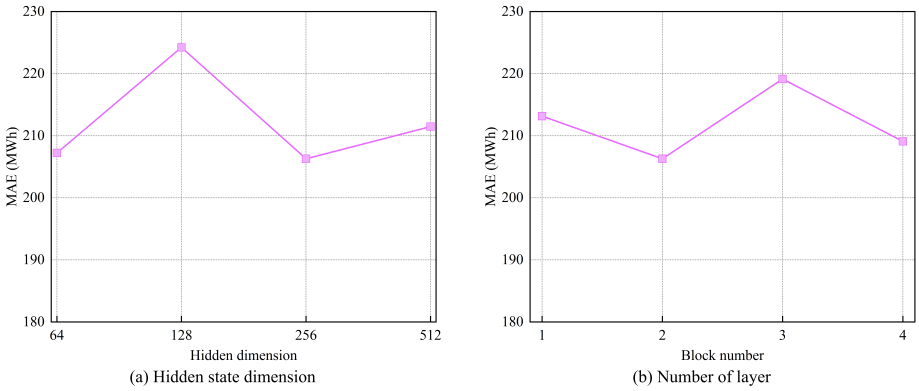


Figure 4: Hyperparameter sensitivity with respect to the hidden dimension size and number of LSTM layers (lookback window length: 72; forecast horizons: 168).

- MultiAttLLM: We varied the transformer model dimension $d_{model} \in \{8, 16, 32, 64, 128\}$ and the number of LLM layers $N \in \{2, 4, 6, 12, 16\}$. For each configuration, we recorded MSE on the validation set. The outcomes are illustrated in Fig. 5. These experiments show that the proposed MultiAttLLM model exhibits strong robustness to hyperparameter selection, achieving optimal or near-optimal accuracy across nearly all tested configurations.

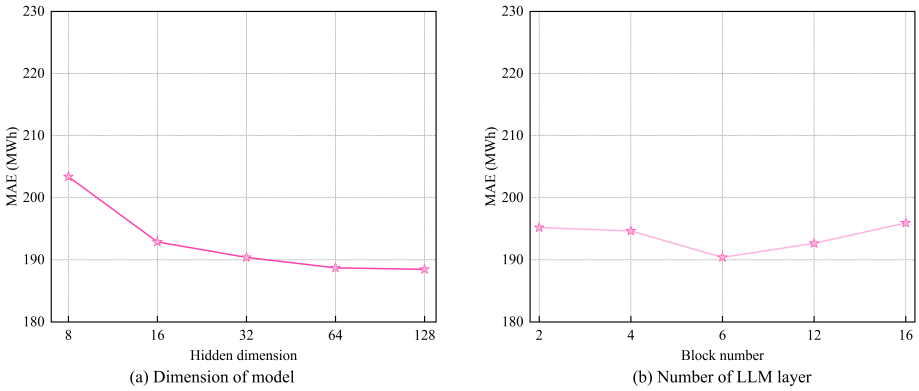


Figure 5: Hyperparameter sensitivity with respect to the dimension size of model and number of LLM layers (lookback window length: 72; forecast horizons: 168).

Drawing from these settings and further experimentation, we identified the parameter sets that delivered the best performance in terms of predictive accu-

racy and computational efficiency. Table 3 summarizes the chosen parameters for each model, including the number of layers, hidden dimensions, attention heads, and other key settings.

Each parameter set was chosen to balance model complexity with computational feasibility, ensuring that models could be effectively trained within the limits of available resources while achieving optimal forecasting performance. The parameters were adjusted based on the characteristics of the time series data, such as the number of features and the frequency of observations, to ensure that the models were well-suited for the forecasting tasks.

To ensure fair comparisons across all models, we used consistent training settings, employing the Adam optimizer with an initial learning rate of 0.001. Each model was trained for a total of 10 epochs, with the learning rate reduced to 95 percent of its value from the previous epoch after each epoch to facilitate convergence. The mean squared error (MSE) loss function was employed as the optimization objective for all models. The MSE is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \tag{12}$$

where y_i and $\hat{y_i}$ represent the measured value and the predicted value; n is the number of batch size which is 24 in this study. This loss function penalizes large prediction errors more heavily, making it particularly effective for minimizing overall prediction deviations. All experiments were conducted on a consistent computational environment to ensure reproducibility. All the code of models and datasets are open sourced in MultiAttLLM GitHub Repository.

All models were encoded using PyTorch and trained on a computer with a 14th Intel(R) Core(TM) i9-14900F CPU 3.20 GHz and 64 GB of working memory (RAM). The models were solved and calculated using a GPU (NVIDIA GeForce RTX 4090 24 GB).

Table 3: Hyperparameters configurations for benchmark models and our model.

Model	**Hyperparameter**	**Value**
LSTM	Hidden size	256
	Number of layers	2
Informer	Dimensions of model	512
	Dimensions of feed-forward	2048
	Number of encoder layers	4
	Number of decoder layers	2
Autoformer	Dimensions of model	512
	Dimensions of feed-forward	2048
	Number of encoder layers	2
	Number of decoder layers	1
iTransformer	Dimensions of model	512
	Dimensions of feed-forward	512
	Number of encoder layers	3
	Number of decoder layers	1
TimesNet	Dimensions of model	32
	Dimensions of feed-forward	64
	Number of encoder layers	2
	Number of decoder layers	1
TimeLLM	Dimensions of patch embedding	32
	Text Prototype	1000
	Number of LLM layers	6
MultiAttLLM	Dimensions of model	64
	Dimensions of feed-forward	128
	Number of LLM layers	6
	Number of decoder layers	4

3. Case study

3.1. Introduction of the dataset

The datasets used for the case study include hourly electricity data (demand and generation), as well as weather data. The electricity data spans from 2016 to 2023, covering four regions in Japan: Kyushu, Tokyo, Tohoku, and Hokkaido. For each region, the dataset includes hourly total electricity demand and generation data from various sources. Electricity data comes from public data of power companies in various regions. Fig. 6 illustrates the electricity demand and generation by different energy sources for the four selected regions in Japan in 2023. As shown in Fig. 6, the Tokyo and Hokkaido regions exhibit similar patterns, with total generation and demand both around 30 MWh. In these regions, renewable energy generation primarily comes from solar and hydroelectric sources. In contrast, the Tohoku and Kyushu regions have much higher electricity demand and generation, both reaching approximately 100 MWh. Notably, in these regions, nuclear energy also constitutes a significant portion of the renewable energy mix. This variation in energy profiles across regions reflects the diversity in energy infrastructure and resource availability, making them ideal for testing the generalization capability of the proposed model.

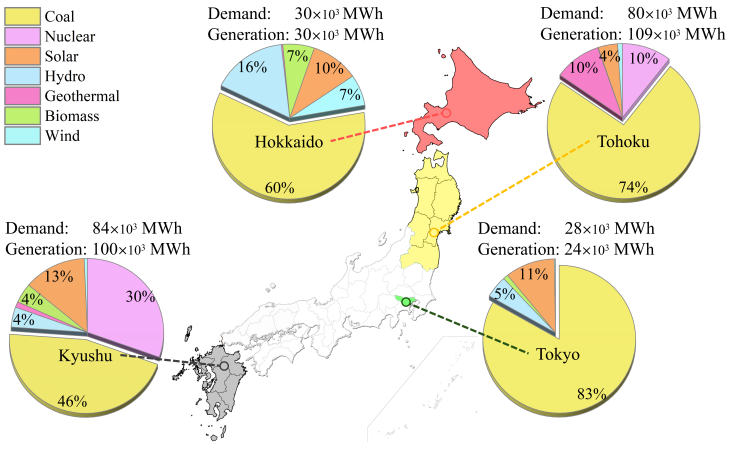
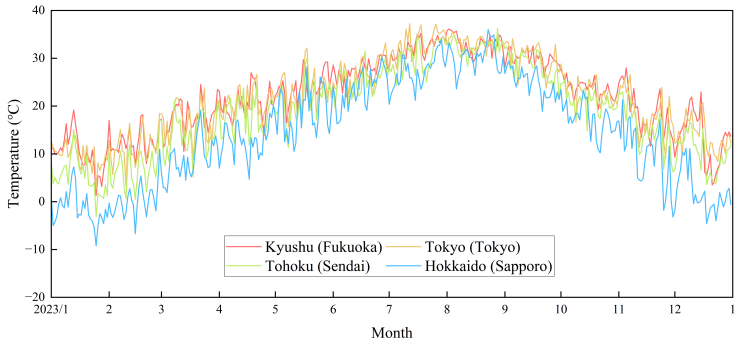


Figure 6: An Overview of electricity demand and generation data for selected regions of Japan in 2023.

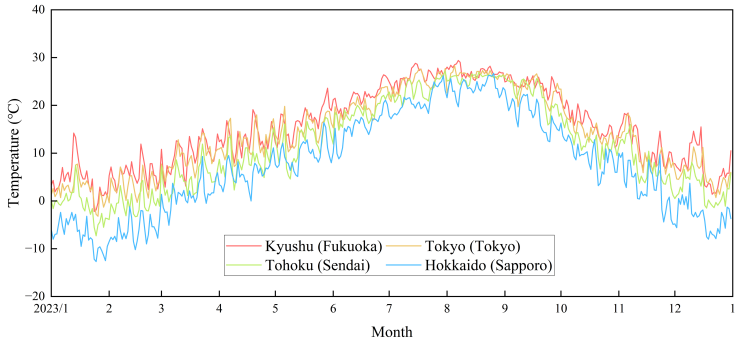
In addition to electricity demand and generation data, the weather conditions in the four selected regions—Fukuoka (Kyushu region), Tokyo (Tokyo region), Sendai (Tohoku region), and Sapporo (Hokkaido region)—play a crucial role in influencing both energy consumption and generation from renewable sources. The weather data, sourced from the Japan Meteorological Agency, also covers the period from 2016 to 2023. The weather data from 2023, as shown in Fig. 7, captures key meteorological variables such as maximum and minimum daily temperatures and solar radiation for each region, providing important context for understanding the seasonal variations in energy patterns.

Hokkaido, located in the northernmost part of Japan, experiences long, cold winters with significant snowfall and relatively cool summers. Solar radiation is lower compared to other regions. Tohoku, in northeastern Japan, also has cold winters, though less severe than Hokkaido. The region experiences distinct seasons, with cooler temperatures in winter and moderate summers. Tokyo, located in the central part of Japan, has a temperate climate with hot, humid summers and mild winters. The relatively higher solar radiation in Tokyo, particularly during summer months, contributes significantly to solar power generation. Kyushu, in the southernmost part of Japan, enjoys a warm climate year-round with hot, humid summers and mild winters. The region receives the highest solar radiation among the four regions, making solar power a key contributor to its renewable energy mix. Kyushu also benefits from a higher proportion of nuclear energy generation, complementing its renewable energy sources.

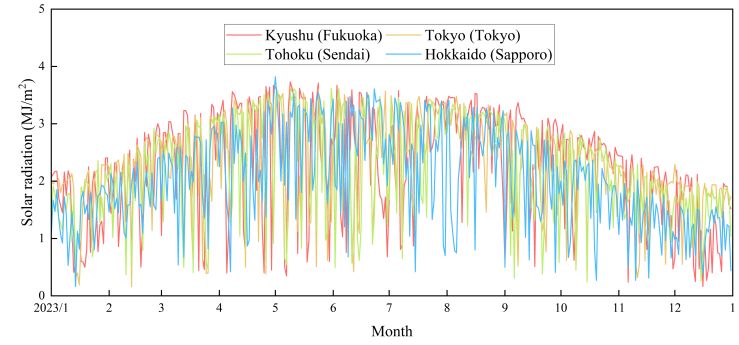
The details of all features in our datasets are summarized in Table 4. The selection of these four regions was made to ensure a comprehensive representation of Japan's diverse climatic zones, ranging from the warmer southern regions to the cooler northern areas. Additionally, these regions feature varying compositions of electricity generation sources, which allows for a thorough evaluation of the proposed model's generalization capability across different climatic and energy production conditions.



(a) Daily maximum temperature



(b) Daily minimum temperature



(c) Daily maximum solar radiation

Figure 7: An Overview of daily weather data for selected regions of Japan in 2023.

Table 4: Description of the datasets(*:The value of renewable energy generation is equal to the sum of all other generation except fossil energy generation; data interval: one-hour).

Data	**Features**
Electricity demand	Total demand
Electricity generation	Fossil
	Nuclear
	Hydro
	Geothermal
	Biomass
	Solar
	Wind
	Renewable energy*
Weather	Temperature
	Relative humidity
	Precipitation
	Dew point
	Vapor pressure
	Wind speed
	Sunshine duration
	Snowfall
	Global horizontal irradiance

3.2. Data standardization

Data standardization adjusts different data ranges to a common scale, which helps to minimize regression errors while preserving correlations within the dataset [61]. This study utilizes Z-score standardization, which normalizes the data to have a mean of zero and a standard deviation of one [62]. The formula for Z-score standardization is:

$$x' = \frac{x - \mu}{\delta} \tag{13}$$

where x' and x represent the standardized data and original data, respectively; and μ and δ represent the mean and the standard deviation of the original data, respectively. Standardization of all feature data is essential before model integration. The formula for inverse standardization can be expressed as

$$x = x' \times \delta + \mu \tag{14}$$

3.3. Data splitting methodology

To validate model effectiveness, we collected eight years of data from 2016 to 2023. First, we assessed how training-set length impacts forecasting accuracy by training representative models on spans ranging from one to six years. The results, shown in Fig. 8, indicate that performance stabilizes once the training period exceeds four years. Balancing predictive gains against computational cost, we therefore selected 2018–2021 as our training dataset. To evaluate generalization to the most recent data, 2022 was used as the validation dataset and 2023 as the test dataset, with the checkpoint achieving the lowest validation loss retained during training.

We employed a fixed-window, interval-output forecasting protocol: at each prediction point, the model ingests the entire historical lookback window and simultaneously outputs all future horizon values in a single forward pass. This interval-output approach prevents any overlap between inputs and targets—unlike rolling-window schemes—thereby avoiding data leakage.

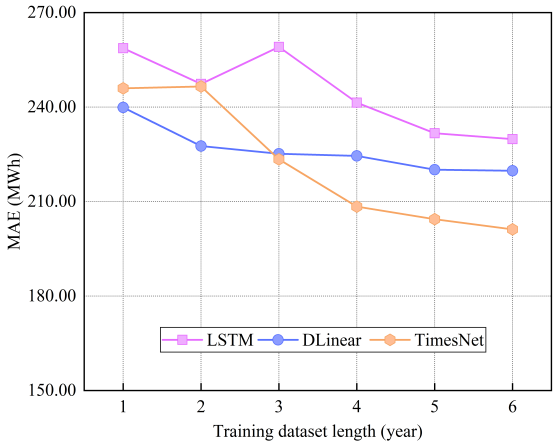


Figure 8: Impact of training dataset length (1–6 years) on MAE for LSTM, DLinear, and TimesNet models (2022 as the validation dataset and 2023 as the test dataset).

3.4. Evaluation metrics

We incorporated three widely used criteria to assess the predictive performance of the model from multiple perspectives: mean absolute error (MAE), mean absolute percentage error (MAPE), root mean squared error (RMSE) and correlation coefficient (R^2). These evaluation metrics enable us to gauge the predictive ability of the model from various angles. The formulas to calculate MAE, MAPE, RMSE and R^2 are

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{15}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{16}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{17}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{18}$$

where y_i, $\hat{y}_i$, and $\bar{y}$ represent the measured value, predicted value, and mean of measured value, respectively; n is the length of sequence. In addition, in order to evaluate the versatility of the model on different data sets, we also used a

dimensionless indicator relative absolute error (RAE), which is calculated as follows:

$$\mathrm{RAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\sum_{i=1}^{n} |y_i - \bar{y}|} \tag{19}$$

As all input data is standardized beforehand, the output of the model represents the predicted value after standardization. We de-standardize the predicted value output by the model and compute the evaluation metrics based on the measured values to more intuitively compare the performance of the model.

3.5. Experimental Setup

Fossil fuel energy and renewable energy play critical roles in shaping energy policies and grid management strategies [63]. Accurate forecasting of electricity demand, as well as generation from fossil fuels and renewable sources, is essential for optimizing grid operations, planning energy transitions, and ensuring energy security [64][65]. Given the growing emphasis on carbon emission reduction and renewable energy integration, accurate forecasting is essential. Our model aims to provide reliable predictions for these key metrics. Therefore, the selected targets for prediction include overall electricity demand, fossil fuel-based power generation, and renewable energy generation, as shown in Fig. 9. By focusing on these aspects, the proposed model can support more informed decision-making for energy policy and grid management.

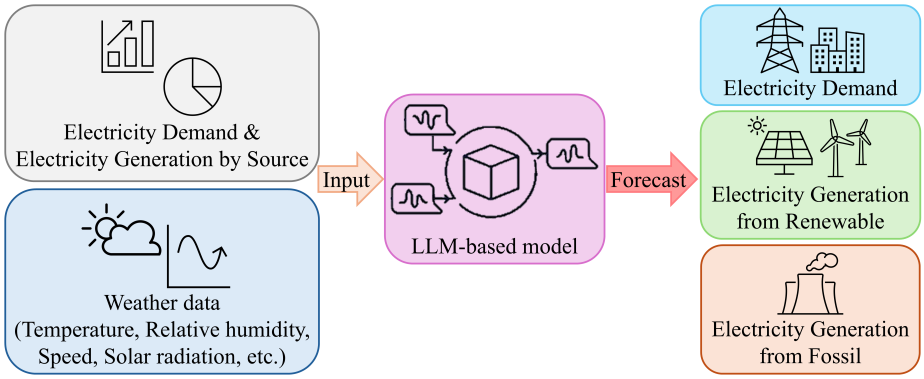


Figure 9: Electricity demand and generation forecasting task.

To further validate the generalization capability of the proposed model, we employed a zero-shot learning approach. The model, trained solely on the Tokyo dataset, was tested on the remaining three regions: Hokkaido, Tohoku, and Kyushu. This approach allowed us to evaluate the model's ability to adapt and provide accurate forecasts without additional training on the new regions' datasets. By comparing the zero-shot performance across these regions, we could assess the robustness and generalization capacity of the proposed model under different climatic and energy generation conditions. In all the above tasks, we fixed the input of historical data for 3 days (72 hours) and predicted the data for the next week (168 hours).

To select an appropriate LLM backbone, we evaluated four candidate models—BERT (420 MB) [66], GPT2 (522 MB) [67], LLAMA-3.2-1b (2.30 GB) [68], and LLAMA-3.2-3b (5.97 GB) [68]. Their memory footprints and per-iteration training times are plotted in Fig. 10, and the average MAE across three forecasting targets was 213, 202, 201, and 198, respectively. While LLAMA-3.2-3b achieves the lowest error, it incurs a very large memory footprint and training latency. GPT2, by contrast, delivers near-state-of-the-art accuracy with the shortest training time, making it the most practical choice on our single-GPU setup. Accordingly, we adopt GPT2 as the LLM component in all subsequent experiments.

4. Results and discussions

4.1. Ablation study

The results of the ablation study, shown in Table 5, demonstrate that each module in the proposed model positively contributes to its predictive performance. The pre-trained LLM has the most significant impact on the model's accuracy, as removing this module leads to a 17.3% drop in MAE. The self-attention module also plays a critical role in extracting relationships between non-target and target features, with its removal resulting in a 16.3% decrease in MAE.

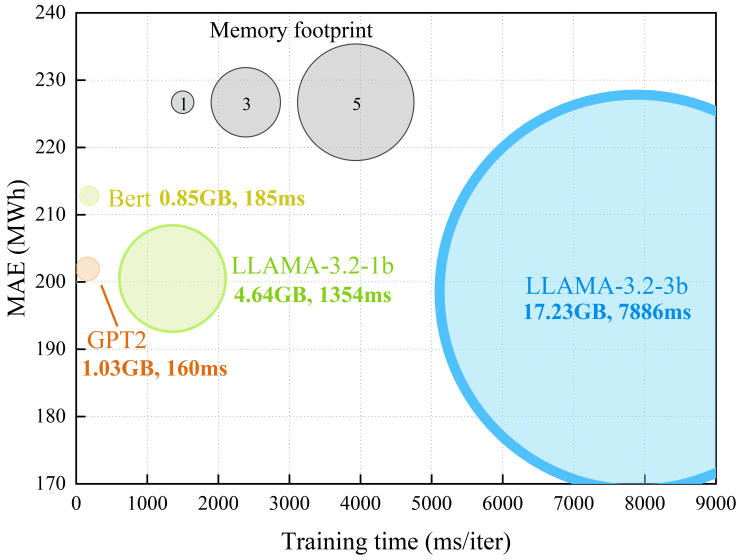


Figure 10: LLM efficiency comparison for proposed model (lookback window length: 72; forecast horizons: 168).

Table 5: Performance of the MultiAttLLM model after removing specific components (- indicates the full model without the corresponding part; * indicates that only the target variable is used as input; ①: Word Projection; ②: Cross-Attention; ③: Pre-trained LLM; ④: Feature Extractor; ⑤: Self-Attention; ⑥: Output Projection).

Descriptions	①	②	③	④	⑤	⑥	MAE
Full model	✓	✓	✓	✓	✓	✓	202 (0.0%)
- Cross-Attention	✗	✗	✓	✓	✓	✓	214 (5.9% ↓)
- Pre-trained LLM	✗	✗	✗	✓	✓	✓	237 (17.3% ↓)
- Feature Extractor	✗	✗	✗	✗	✓	✓	241 (19.3% ↓)
- Feature Extractor	✓	✓	✓	✗	✓	✓	223 (10.4% ↓)
- Self-Attention	✓	✓	✓	✓	✗	✓	235 (16.3% ↓)
Full model (CI)	✓	✓	✓	✓	✓	✓	241 (19.3% ↓)
- Feature Extractor* (CI)	✓	✓	✗	✓	✓	✓	232 (14.9% ↓)

Other modules, such as the feature extractor and the word projection, have relatively smaller effects on the model's performance. Removing the feature extractor decreases the MAE by 10.4%, while removing the word projection module decreases the MAE by 5.9%. These findings indicate that, while all components contribute positively to the model's overall performance, the pre-trained LLM and the self-attention mechanisms are particularly crucial for achieving the model's high accuracy.

Additionally, we also evaluated the model under the CI training strategy. As shown in Table 5 last two row, adopting CI causes a substantial degradation in accuracy: MAE increases by 19.3% compared to our standard setup. When CI is applied with only the target variable as input (i.e., only use pre-trained LLM without feature extractor), MAE still increases by 14.9%. This drop arises because CI computes the loss for a single variable at each iteration. With all variables supplied, the model optimizes to minimize the aggregate loss across every channel, rather than focusing on the intended target, which harms its ability to predict that target accurately. Meantime, when only the target variable is provided, the model cannot leverage cross-channel information, resulting in a further reduction in predictive performance.

4.2. Performance under different prompt engineering

To evaluate the influence of prompt engineering on model performance, we compared the LLM-based TimeLLM model with the proposed MultiAttLLM model under different prompt conditions. The results, shown in Table 6, include scenarios without a prompt and with three progressively more complex prompts (Prompt1, Prompt2, and Prompt3). A detailed introduction to prompt is shown in Appendix A.2.

The results indicate that designing more complex and accurate prompts can improve the prediction accuracy of the TimeLLM model. Specifically, for TimeLLM, using Prompt3 improves the MAE by up to 8.1% compared to the no-prompt condition. However, the proposed MultiAttLLM model shows minimal sensitivity to prompt complexity. Regardless of the prompt condition, the

Table 6: Performance of LLM-based models under different prompts (Prompt1: description of the prediction task only; Prompt2: description of the prediction task and dataset; Prompt3: description of the prediction task, dataset, and input sequence statistics; lookback window length: 72; forecast horizons: 168; the unit of MAE and RMSE is MWh, MAPE is %).

Model	TimeLLM				MultiAttLLM			
Metrics	MAE	MAPE	RMSE	R^2	MAE	MAPE	RMSE	R^2
Without prompt	236	21.02	325	0.70	202	15.40	283	0.77
Prompt1	235	21.79	325	0.70	209	16.46	292	0.75
Prompt2	230	19.20	320	0.71	208	16.40	291	0.75
Prompt3	217	16.24	307	0.73	205	16.20	288	0.76

MultiAttLLM consistently outperforms TimeLLM, achieving the highest accuracy (MAE: 202, RMSE: 283, R^2: 0.77) even without any prompt.

These findings suggest that the proposed MultiAttLLM model can effectively extract and process feature relationships through its multi-module architecture without relying on prompt design. This highlights its robustness and efficiency in utilizing LLMs, making it highly suitable for time series forecasting tasks without the added complexity of prompt engineering.

4.3. Performance comparison of benchmark models and the proposed MultiAttLLM model

The electricity demand and generation forecasting results for both the benchmark models and the proposed model are presented in Table 7. The results demonstrate that the traditional LSTM model performs the worst across all prediction targets and metrics. The proposed MultiAttLLM model achieves the best results for all targets and metrics, outperforming the other models in terms of MAE, RMSE, and R^2 scores. The models that improve upon traditional approaches, Informer, iTransformer, Autoformer and TimesNet, show similar performance. For some specific variables and metrics, these models can match the best results achieved by the proposed model.

Table 7: Electricity demand and generation forecasting results (lookback window length: 72; forecast horizons: 168; the unit of MAE and RMSE is MWh, MAPE is %; boldface indicates the best performance; underlined values denote the second-best performance).

Target variable	Metrics	Method							
		LSTM	DLinear	Informer	Autoformer	iTransformer	TimesNet	TimeLLM	**MultiAttLLM**
Electricity demand	MAE	288	259	235	265	232	251	250	**228**
	MAPE	8.88	7.77	7.15	8.23	7.01	7.58	7.42	**6.89**
	RMSE	384	355	317	356	313	337	346	**311**
	R^2	0.73	0.77	0.80	0.75	**0.82**	0.79	0.78	**0.82**
Generation (Renewable energy)	MAE	165	153	135	154	150	148	139	**127**
	MAPE	48.35	39.25	**26.39**	43.43	36.46	35.63	29.34	27.47
	RMSE	237	243	222	222	237	235	232	**213**
	R^2	0.75	0.73	0.73	0.73	0.75	0.75	0.76	**0.80**
Generation (Fossil energy)	MAE	278	257	271	258	252	**251**	262	**251**
	MAPE	13.19	11.97	13.05	11.86	11.85	**11.49**	11.96	11.83
	RMSE	360	336	343	335	326	332	342	**325**
	R^2	0.60	0.66	0.62	0.64	**0.68**	0.66	0.64	**0.68**
Mean	MAE	244	223	214	226	211	216	217	**202**
	MAPE	23.47	19.66	15.53	21.17	18.44	18.23	16.24	**15.40**
	RMSE	327	311	294	304	292	302	307	**283**
	R^2	0.69	0.72	0.72	0.71	0.75	0.75	0.73	**0.77**

Furthermore, we also compared each model's per-iteration training time and memory footprint, as shown in Fig. 11. DLinear exhibits the smallest memory footprint (≈0.02GB) and the fastest speed (≈2ms/iter), though with only moderate forecasting accuracy. The transformer-based baselines (Informer, Autoformer, iTransformer, TimesNet) cluster around similar MAE values and training times, each outperforming the vanilla LSTM while maintaining comparable efficiency. Under nearly identical parameter counts (TimeLLM: 61.7M; MultiAttLLM: 62.4M), our MultiAttLLM reduces memory usage by about 49.3% (from 2.03GB down to 1.03GB) and cuts per-iteration training time by roughly 35.7% (from 249ms to 160ms). Meanwhile, the proposed model outperforms TimeLLM, with average improvements of 6.9%, 7.8%, and 5.5% in MAE, RMSE, and R2, respectively.

4.4. Model performance for different target variables

The relative errors for predicting electricity demand, renewable energy generation, and fossil fuel energy generation across all models are shown in Fig. 12.

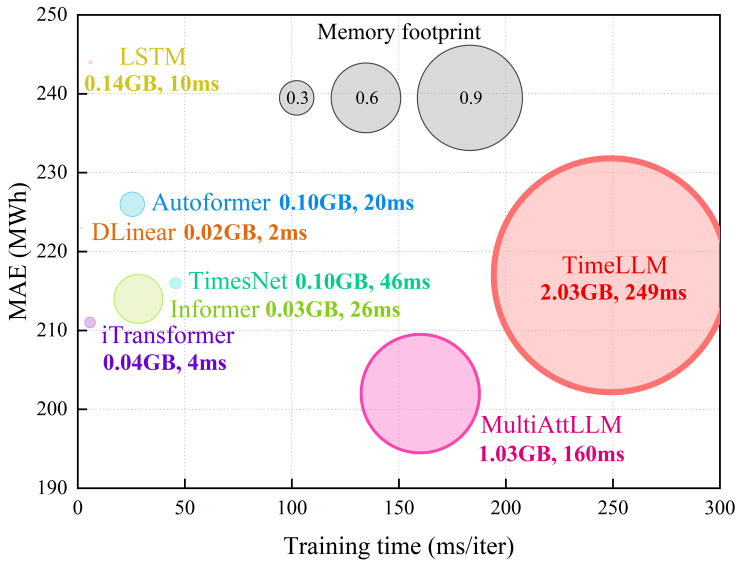


Figure 11: Model efficiency comparison (lookback window length: 72; forecast horizons: 168).

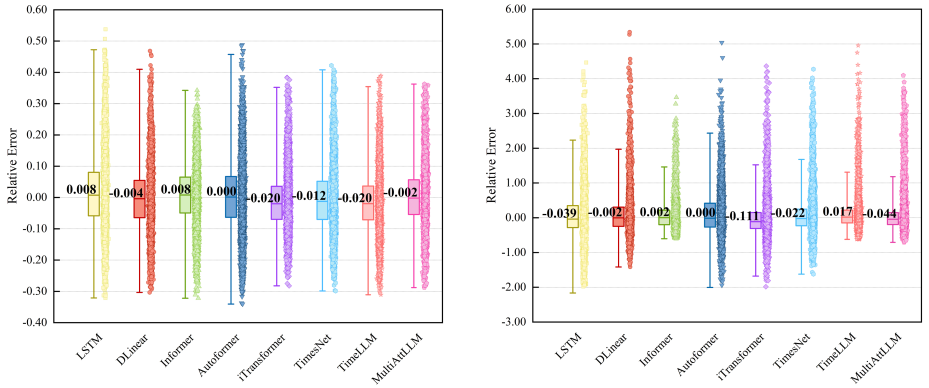
The calculation method for relative error (RE) is as follows:

$$\mathrm{RE} = \frac{\hat{y}_i - y_i}{y_i} \tag{20}$$

where y_i and $\hat{y}_i$ represent the measured value and predicted value.

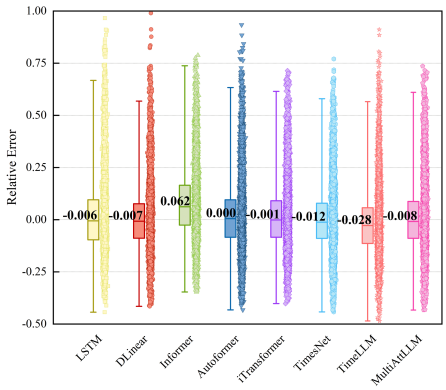
The results indicate that all models perform best when predicting electricity demand, followed by fossil fuel energy generation. The prediction accuracy for renewable energy generation is the lowest. This is primarily because electricity demand and fossil fuel generation in a given region tend to be more stable and are less influenced by seasonal or weather changes. In contrast, renewable energy generation, which includes solar, hydro, and wind power, is highly dependent on weather and seasonal variations, making it more challenging to predict accurately.

As illustrated in Fig. 13, a portion of the electricity demand forecasting results shows that the daily variation patterns are fairly consistent across all days, with peaks occurring during the day and troughs at night. When the peak demand and fluctuations between consecutive days are similar, all models show strong predictive performance. However, when there are significant deviations in



(a) Electricity demand

(b) Generation from renewable energy



(c) Generation from fossil energy

Figure 12: Relative error for electricity demand, renewable energy generation, and fossil energy generation predictions across different models(lookback window length: 72; forecast horizons: 168).

peak demand or fluctuations compared to the preceding and following days, the performance of all models decreases to some extent. Nonetheless, the proposed model continues to exhibit the best overall performance, particularly in scenarios with greater variability in demand.

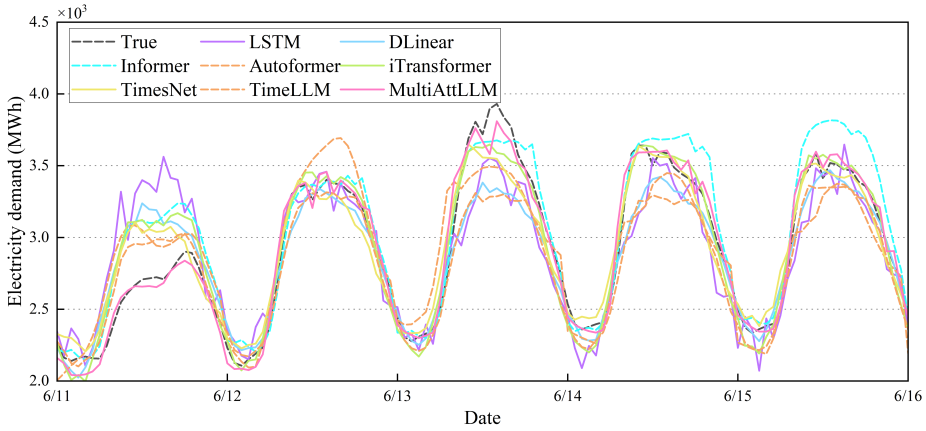


Figure 13: Comparison of electricity demand forecasting results across different models.

4.5. Performance across different forecasting horizons

To assess how horizon length impacts predictive accuracy, we evaluated every model on horizons ranging from 1 step (1 h) to 720 steps (30 days) for electricity demand forecasting, with the average results of three target variables plotted in Fig. 14. When forecasting just one step ahead, all models achieve very high accuracy due to the strong correlation between immediate history and the next value. As the horizon extends from 1 to 24 steps, performance for every method declines sharply. Beyond 24 steps, however, the rate of accuracy degradation slows markedly—longer-term targets bear weaker links to input history, so error growth tapers off. Among the baselines, Informer and TimesNet show the slowest drop in precision over long horizons, but our MultiAttLLM model consistently maintains the highest accuracy across both short-term and long-term forecasts.

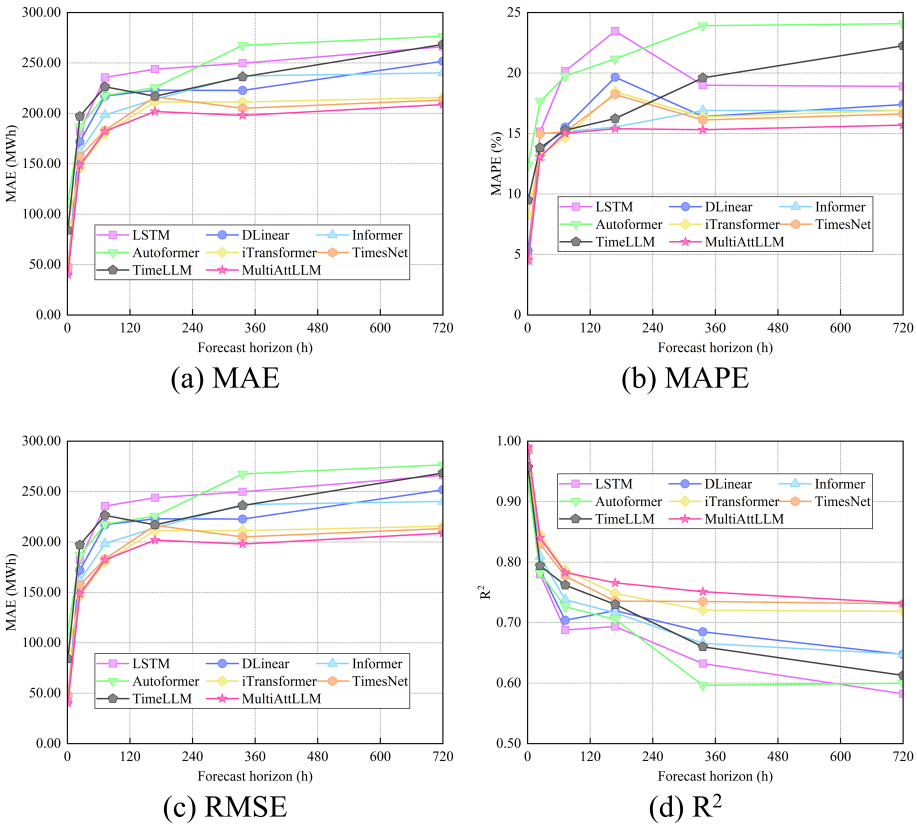


Figure 14: Performance across different forecasting horizons (from 1 steps to 720 steps)

4.6. Generalization performance through zero-shot learning

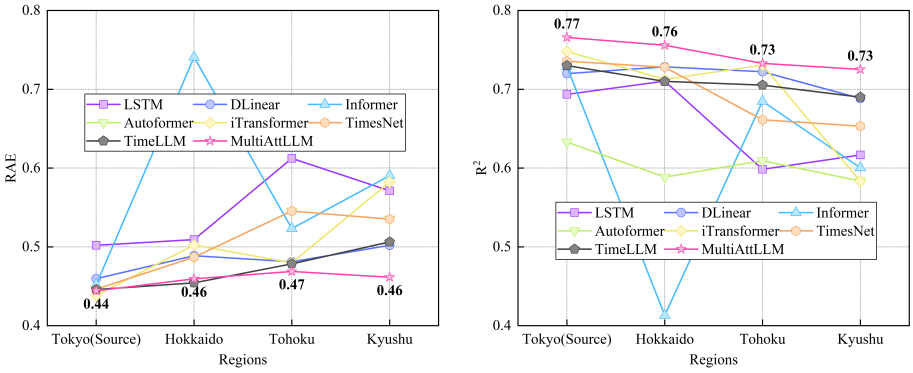
To evaluate the generalization capability of the proposed model, we conducted zero-shot learning experiments. For each experiment, data from one of the four regions—Tokyo, Hokkaido, Tohoku, or Kyushu—was used as the source domain to train the model. The trained model was then tested on all four target domains without any fine-tuning or adjustments. The average results for each target domain are presented in Table 8. These results demonstrate that the proposed MultiAttLLM model consistently achieves the best generalization performance across all target domains. The TimeLLM model achieves the second-best accuracy, highlighting the superior generalization ability of LLM-based models. The DLinear and TimesNet model, which leverages time series decomposition, demonstrates a moderate level of generalization. In contrast, the self-attention based models, which perform reasonably well in non-transfer learning settings, show significant drops in performance when tested on unseen datasets. Detailed results for using each region as the source domain and the corresponding predictions on different target domains are provided in Appendix A.3.

Table 8: Zero-shot learning performance across target domains using different source regions (Results averaged across target domains for each source region; lookback window length: 72; forecast horizons: 168; the unit of MAE and RMSE is MWh).

Target domain	Tokyo				Hokkaido				Tohoku				Kyushu			
Metrics	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2
LSTM	253	351	0.52	0.64	250	330	0.48	0.74	1097	1426	0.57	0.64	880	1210	0.52	0.66
DLinear	227	316	0.47	0.71	241	318	0.46	0.75	928	1245	0.48	0.73	824	1159	0.49	0.69
Informer	250	341	0.53	0.63	307	395	0.62	0.57	928	1240	0.54	0.64	906	1218	0.56	0.62
Autoformer	276	367	0.59	0.58	277	359	0.56	0.65	1018	1339	0.60	0.58	928	1238	0.58	0.60
iTransformer	288	380	0.60	0.56	352	474	0.60	0.57	1164	1528	0.59	0.60	1133	1501	0.67	0.48
TimesNet	237	328	0.49	0.69	244	323	0.46	0.75	1201	1576	0.62	0.54	841	1170	0.50	0.68
TimeLLM	229	318	0.47	0.71	238	314	0.45	0.76	943	1261	0.49	0.72	849	1185	0.51	0.68
MultiAttLLM	**211**	**304**	**0.43**	**0.73**	**234**	**310**	**0.44**	**0.77**	**891**	**1220**	**0.46**	**0.74**	**796**	**1131**	**0.47**	**0.71**

When using the Tokyo dataset as the source domain, the zero-shot learning results for Hokkaido, Tohoku, and Kyushu are shown in Fig. 15. The dimensionless metrics reveal that transformer-based models and the LSTM model exhibit the poorest generalization capability. While these models maintain reasonable

accuracy on the Tohoku dataset, which shares geographic and climatic similarities with Tokyo, their performance significantly deteriorates on the Hokkaido and Kyushu datasets, which feature distinct climatic and energy generation characteristics. In contrast, the proposed MultiAttLLM model demonstrates the best generalization performance across all test regions, achieving the lowest relative errors and maintaining robust predictions even under varying conditions.



(a) RAE (b) R^2

Figure 15: Zero-shot learning results of two dimensionless metrics (Source domain: Tokyo; Target domains: Hokkaido, Tohoku, Kyushu; lookback window length: 72; forecast horizons: 168).

5. Conclusion

Large language models (LLMs) have garnered significant attention in time series forecasting due to their strong performance across diverse tasks. However, their application to multivariate forecasting remains challenged by channel-independent training strategies that impede learning of cross-channel interactions and by the complexity of prompt engineering. To overcome these obstacles, we introduced a novel multi-attention LLM framework that integrates cross-attention, self-attention, and a frozen pre-trained LLM to extract rich interdependencies among target and auxiliary variables without any manual prompt design.

Extensive ablation studies demonstrate that every component of our architecture contributes positively to forecasting accuracy, with the pre-trained LLM backbone and the attention modules yielding the largest gains. In benchmark comparisons, our model outperforms state-of-the-art transformer-based and LLM-based approaches, reducing mean absolute error by up to 20.8% relative to standard LSTM models. Compared with the current best time-series LLM, the proposed model reduces memory usage by 49.3% and shortens training time by 35.7%. Moreover, it maintains superior trend-following capabilities under volatile conditions, achieving the highest accuracy across electricity demand, renewable generation, and fossil generation forecasts. Crucially, in zero-shot evaluations on four geographically diverse Japanese regions, our framework improves MAE by an average of 16.6% over LSTM, demonstrating exceptional generalization ability. These findings underscore the effectiveness of multi-attention reprogramming in unlocking the full potential of LLMs for accurate, robust, and generalizable multivariate time series forecasting.

However, this study has some limitations. First, the model was not fine-tuned for specific time series forecasting tasks, which could further improve its performance. Additionally, the computational complexity of LLMs remains a challenge, particularly for real-time applications. In future work, we aim to explore further improvements by fine-tuning the LLMs module for specific time series forecasting tasks and incorporating more complex data sources, such as real-time data streams. Additionally, extending the model to handle multi-horizon forecasting and integrating domain-specific knowledge into the attention mechanisms could further enhance the model's accuracy and applicability in various industries.

Acknowledgment

This work was supported by project No. 23KJ0766 funded by the Japan Society for The Promotion of Science.

Appendix A.1 Long short-term memory network

The core functionality of the long short-term memory (LSTM) unit lies in its three gating mechanisms, which are controlled by sigmoid functions. These gates regulate the proportion of past and current input information used in the unit's calculations, ultimately determining the output for the current time step through the output gate (denoted as $\boldsymbol{h}^{(t)}$ in Fig. 1). The cell state, $\boldsymbol{c}^{(t)}$, remains largely unaffected by the gates and stays relatively stable throughout the computation process. This "conveyor belt" system plays a critical role in maintaining the long-term memory capabilities of the LSTM network.

LSTM is calculated using:

$$\boldsymbol{f}^{(t)} = \sigma(\boldsymbol{W}_f[\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}] + \boldsymbol{b}_f) \tag{1}$$

$$\boldsymbol{i}^{(t)} = \sigma(\boldsymbol{W}_i[\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}] + \boldsymbol{b}^{(t)}) \tag{2}$$

$$\tilde{\boldsymbol{c}}^{(t)} = \tanh(\boldsymbol{W}_c[\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}] + \boldsymbol{b}_c) \tag{3}$$

$$\boldsymbol{c}^{(t)} = \boldsymbol{f}^{(t)} \odot \boldsymbol{c}^{(t-1)} + \boldsymbol{i}^{(t)} \odot \tilde{\boldsymbol{c}}^{(t)} \tag{4}$$

$$\boldsymbol{o}^{(t)} = \sigma(\boldsymbol{W}_o[\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}] + \boldsymbol{b}_o) \tag{5}$$

$$\boldsymbol{h}^{(t)} = \boldsymbol{o}^{(t)} \odot \tanh(\boldsymbol{c}^{(t)}) \tag{6}$$

where $\boldsymbol{f}^{(t)}$, $\boldsymbol{i}^{(t)}$, and $\boldsymbol{o}^{(t)}$ represents the calculation results of the forget gate, input gate, and output gate, respectively; $\tilde{\boldsymbol{c}}^{(t)}$ is the cell state update; $\boldsymbol{h}^{(t)}$ is the hidden state; $\boldsymbol{W}$, $\boldsymbol{U}$, and $\boldsymbol{b}$ refer to the weights and biases in the model; $\odot$ is the Hadamard product; σ is the sigmoid function.

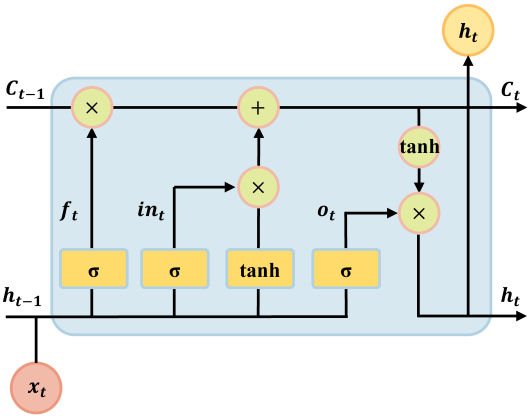


Figure 1: Unfold structure of the LSTM unit.

Appendix A.2 Prompt engineering

Based on the prompt engineering described in [36], we designed three types of prompts tailored for time series forecasting tasks. These prompts are detailed as follows ({} is task-specific configurations or calculated input statistics):

1) Prompt1 (prediction task description only): ¡—start_prompt—¿ Task description: forecast the next {forecast sequence length} steps given the previous {input sequence length} steps information ¡—end_prompt—¿

2) Prompt2 (prediction task and dataset description): ¡—start_prompt—¿ Task description: forecast the next {forecast sequence length} steps given the previous {input sequence length} steps information; Dataset description: Electricity dataset is recorded every 1 hour, which contains meteorological indicators and power source composition of Japan , such as air temperature, humidity, solar, coal nuclear, etc. ¡—end_prompt—¿

3) Prompt3 (prediction task, dataset description, and input sequence statistics): ¡—start_prompt—¿ Task description: forecast the next {forecast sequence length} steps given the previous {input sequence length} steps information; Dataset description: Electricity dataset is recorded every 1

hour, which contains meteorological indicators and power source composition of Japan, such as air temperature, humidity, solar, coal nuclear, etc.; Input statistics: min value {min values}, max value {max values}, median value {median values}, the trend of input is {upward or downward}, top {top$_k$}, lags are : {lags values} ¡—end_prompt—¿

Appendix A.3 Results of zero-shot learning

Table 9: Zero-shot learning performance across target domains using Kyushu region as source domain (The unit of MAE and RMSE is MWh).

Target domain	Tokyo				Hokkaido				Tohoku				Kyushu			
Metrics	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2
LSTM	268	379	0.55	0.59	264	349	0.50	0.71	1036	1345	0.54	0.68	804	1126	0.48	0.71
DLinear	231	329	0.48	0.68	248	327	0.47	0.74	917	1232	0.48	0.73	784	1109	0.47	0.72
Informer	238	321	0.51	0.68	302	393	0.61	0.58	882	1203	0.52	0.67	718	985	0.45	0.75
Autoformer	284	383	0.61	0.55	274	355	0.55	0.65	1053	1400	0.63	0.54	844	1152	0.53	0.66
iTransformer	282	375	0.59	0.58	314	418	0.53	0.67	1114	1470	0.57	0.63	1056	1417	0.62	0.54
TimesNet	235	328	0.49	0.68	258	350	0.49	0.70	1554	2054	0.79	0.26	761	1072	0.45	0.74
TimeLLM	229	318	0.47	0.71	239	315	0.46	0.76	934	1251	0.49	0.72	849	1184	0.50	0.68
MultiAttLLM	**211**	**304**	**0.43**	**0.73**	**237**	**314**	**0.45**	**0.76**	**891**	**1223**	**0.46**	**0.74**	**796**	**1130**	**0.47**	**0.71**

Table 10: Zero-shot learning performance across target domains using Hokkaido region as source domain (The unit of MAE and RMSE is MWh).

Target domain	Tokyo				Hokkaido				Tohoku				Kyushu			
Metrics	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2
LSTM	261	359	0.54	0.63	238	317	0.45	0.76	1177	1521	0.61	0.59	907	1237	0.54	0.65
DLinear	234	321	0.48	0.70	226	299	0.43	0.78	938	1256	0.49	0.72	808	1136	0.48	0.70
Informer	285	380	0.61	0.56	266	343	0.54	0.68	1182	1560	0.70	0.44	1054	1398	0.65	0.51
Autoformer	307	405	0.65	0.50	261	341	0.53	0.68	1073	1408	0.63	0.54	958	1273	0.60	0.58
iTransformer	295	387	0.62	0.54	371	503	0.63	0.52	1187	1555	0.59	0.59	1159	1533	0.68	0.45
TimesNet	255	349	0.52	0.66	231	305	0.44	0.77	1053	1384	0.55	0.66	862	1196	0.51	0.67
TimeLLM	231	321	0.47	0.70	236	312	0.45	0.76	950	1272	0.49	0.71	853	1192	0.51	0.67
MultiAttLLM	**210**	**303**	**0.43**	**0.73**	**226**	**302**	**0.43**	**0.78**	**888**	**1216**	**0.46**	**0.74**	**791**	**1130**	**0.47**	**0.71**

References

[1] Y. Oswald, A. Owen, J. K. Steinberger, Large inequality in international and intranational energy footprints between income groups and across consumption categories, Nature Energy 5 (3) (2020) 231–239.

Table 11: Zero-shot learning performance across target domains using Tokyo region as source domain (The unit of MAE and RMSE is MWh).

Target domain	Tokyo				Hokkaido				Tohoku				Kyushu			
Metrics	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2
LSTM	222	305	0.46	0.73	261	338	0.50	0.72	999	1318	0.52	0.69	902	1241	0.54	0.64
DLinear	208	293	0.43	0.75	264	347	0.50	0.71	919	1236	0.47	0.73	894	1258	0.53	0.63
Informer	208	288	0.45	0.73	370	470	0.74	0.41	898	1190	0.52	0.68	949	1252	0.59	0.60
Autoformer	256	341	0.55	0.63	307	391	0.62	0.59	1035	1335	0.61	0.59	975	1270	0.61	0.58
iTransformer	280	370	0.60	0.57	354	473	0.60	0.57	1168	1534	0.59	0.60	1160	1523	0.68	0.47
TimesNet	203	287	0.42	0.76	254	331	0.48	0.73	1142	1483	0.60	0.59	880	1216	0.52	0.66
TimeLLM	226	313	0.46	0.72	240	315	0.46	0.76	938	1250	0.49	0.72	843	1171	0.50	0.68
MultiAttLLM	**202**	**283**	**0.42**	**0.77**	**246**	**323**	**0.46**	**0.76**	**898**	**1227**	**0.47**	**0.73**	**806**	**1135**	**0.46**	**0.73**

Table 12: Zero-shot learning performance across target domains using Tohoku region as source domain (The unit of MAE and RMSE is MWh).

Target domain	Tokyo				Hokkaido				Tohoku				Kyushu			
Metrics	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2	MAE	RMSE	RAE	R^2
LSTM	289	375	0.60	0.59	287	369	0.55	0.67	961	1267	0.50	0.72	1255	1594	0.73	0.42
DLinear	222	304	0.46	0.73	236	308	0.45	0.77	892	1195	0.46	0.75	828	1177	0.49	0.68
Informer	267	374	0.56	0.57	290	374	0.59	0.62	748	1009	0.44	0.77	902	1237	0.57	0.60
Autoformer	256	340	0.55	0.64	265	349	0.54	0.67	909	1215	0.53	0.66	934	1257	0.58	0.60
iTransformer	292	384	0.61	0.55	360	482	0.61	0.56	1168	1535	0.59	0.60	1159	1529	0.68	0.46
TimesNet	543	696	1.08	-0.33	396	515	0.75	0.36	928	1262	0.48	0.72	1489	1981	0.89	0.06
TimeLLM	234	325	0.48	0.69	238	315	0.45	0.76	961	1284	0.50	0.71	873	1211	0.52	0.67
MultiAttLLM	**210**	**304**	**0.43**	**0.73**	**230**	**307**	**0.44**	**0.77**	**886**	**1215**	**0.46**	**0.74**	**791**	**1129**	**0.47**	**0.71**

[2] C. Dingbang, C. Cang, C. Qing, S. Lili, C. Caiyun, Does new energy consumption conducive to controlling fossil energy consumption and carbon emissions?-evidence from china, Resources policy 74 (2021) 102427.

[3] R. Khalili, A. Khaledi, M. Marzband, A. F. Nematollahi, B. Vahidi, P. Siano, Robust multi-objective optimization for the iranian electricity market considering green hydrogen and analyzing the performance of different demand response programs, Applied Energy 334 (2023) 120737.

[4] M. S. S. Danish, T. Senjyu, T. Funabashia, M. Ahmadi, A. M. Ibrahimi, R. Ohta, H. O. R. Howlader, H. Zaheb, N. R. Sabory, M. M. Sediqi, A sustainable microgrid: A sustainability and management-oriented approach, Energy Procedia 159 (2019) 160–167.

[5] C.-T. Hsiao, C.-S. Liu, D.-S. Chang, C.-C. Chen, Dynamic modeling of the policy effect and development of electric power systems: A case in taiwan, Energy policy 122 (2018) 377–387.

[6] M. Ghiasi, T. Niknam, Z. Wang, M. Mehrandezh, M. Dehghani, N. Ghadimi, A comprehensive review of cyber-attacks and defense mechanisms for improving security in smart grid energy systems: Past, present and future, Electric Power Systems Research 215 (2023) 108975.

[7] T. Hong, P. Pinson, Y. Wang, R. Weron, D. Yang, H. Zareipour, Energy forecasting: A review and outlook, IEEE Open Access Journal of Power and Energy 7 (2020) 376–388.

[8] P. Akbary, M. Ghiasi, M. R. R. Pourkheranjani, H. Alipour, N. Ghadimi, Extracting appropriate nodal marginal prices for all types of committed reserve, Computational Economics 53 (2019) 1–26.

[9] M. Sharma, N. Mittal, A. Mishra, A. Gupta, Survey of electricity demand forecasting and demand side management techniques in different sectors to identify scope for improvement, Smart Grids and Sustainable Energy 8 (2) (2023) 9.

[10] M. Ghiasi, Z. Wang, M. Mehrandezh, S. Jalilian, N. Ghadimi, Evolution of smart grids towards the internet of energy: Concept and essential components for deep decarbonisation, IET Smart Grid 6 (1) (2023) 86–102.

[11] A. O. Aderibigbe, E. C. Ani, P. E. Ohenhen, N. C. Ohalete, D. O. Daraojimba, Enhancing energy efficiency with ai: a review of machine learning models in electricity demand forecasting, Engineering Science & Technology Journal 4 (6) (2023) 341–356.

[12] A. Román-Portabales, M. López-Nores, J. J. Pazos-Arias, Systematic review of electricity demand forecast using ann-based machine learning algorithms, Sensors 21 (13) (2021) 4544.

[13] N. Sultana, S. Z. Hossain, S. H. Almuhaini, D. Düştegör, Bayesian optimization algorithm-based statistical and machine learning approaches for forecasting short-term electricity demand, Energies 15 (9) (2022) 3425.

[14] C. E. Velasquez, M. Zocatelli, F. B. Estanislau, V. F. Castro, Analysis of time series models for brazilian electricity demand forecasting, Energy 247 (2022) 123483.

[15] W. Jiang, X. Wang, H. Huang, D. Zhang, N. Ghadimi, Optimal economic scheduling of microgrids considering renewable energy sources based on energy hub model using demand response and improved water wave optimization algorithm, Journal of Energy Storage 55 (2022) 105311.

[16] J. F. Torres, F. Martínez-Álvarez, A. Troncoso, A deep lstm network for the spanish electricity consumption forecasting, Neural Computing and Applications 34 (13) (2022) 10533–10545.

[17] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, M. Long, Timesnet: Temporal 2d-variation modeling for general time series analysis, arXiv preprint arXiv:2210.02186 (2022).

[18] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in:

Proceedings of the AAAI conference on artificial intelligence, Vol. 35, 2021, pp. 11106–11115.

[19] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, Advances in neural information processing systems 34 (2021) 22419–22430.

[20] M. E. Günay, Forecasting annual gross electricity demand by artificial neural networks using predicted values of socio-economic indicators and climatic conditions: Case of turkey, Energy Policy 90 (2016) 92–101.

[21] H. Iftikhar, J. E. Turpo-Chaparro, P. Canas Rodrigues, J. L. López-Gonzales, Day-ahead electricity demand forecasting using a novel decomposition combination method, Energies 16 (18) (2023) 6675.

[22] F. Pallonetto, C. Jin, E. Mangina, Forecast electricity demand in commercial building with machine learning models to enable demand response programs, Energy and AI 7 (2022) 100121.

[23] T. G. Grandón, J. Schwenzer, T. Steens, J. Breuing, Electricity demand forecasting with hybrid classical statistical and machine learning algorithms: Case study of ukraine, Applied Energy 355 (2024) 122249.

[24] E. Cebekhulu, A. J. Onumanyi, S. J. Isaac, Performance analysis of machine learning algorithms for energy demand–supply prediction in smart grids, Sustainability 14 (5) (2022) 2546.

[25] Z. Wang, Z. Chen, Y. Yang, C. Liu, X. Li, J. Wu, A hybrid autoformer framework for electricity demand forecasting, Energy Reports 9 (2023) 3800–3812.

[26] C. Wu, J. Li, W. Liu, Y. He, S. Nourmohammadi, Short-term electricity demand forecasting using a hybrid anfis–elm network optimised by an improved parasitism–predation algorithm, Applied Energy 345 (2023) 121316.

[27] C. Sekhar, R. Dahiya, Robust framework based on hybrid deep learning approach for short term load forecasting of building electricity demand, Energy 268 (2023) 126660.

[28] E. C. May, A. Bassam, L. J. Ricalde, M. E. Soberanis, O. Oubram, O. M. Tzuc, A. Y. Alanis, A. Livas-García, Global sensitivity analysis for a real-time electricity market forecast by a machine learning approach: A case study of mexico, International Journal of Electrical Power & Energy Systems 135 (2022) 107505.

[29] Y. Jiang, T. Gao, Y. Dai, R. Si, J. Hao, J. Zhang, D. W. Gao, Very short-term residential load forecasting based on deep-autoformer, Applied Energy 328 (2022) 120120.

[30] M. Alhussein, K. Aurangzeb, S. I. Haider, Hybrid cnn-lstm model for short-term individual household load forecasting, Ieee Access 8 (2020) 180544–180557.

[31] R.-x. Nie, Z.-p. Tian, R.-y. Long, W. Dong, Forecasting household electricity demand with hybrid machine learning-based methods: Effects of residents' psychological preferences and calendar variables, Expert Systems with Applications 206 (2022) 117854.

[32] T. B. Brown, Language models are few-shot learners, arXiv preprint arXiv:2005.14165 (2020).

[33] A. Vaswani, Attention is all you need, Advances in Neural Information Processing Systems (2017).

[34] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili, et al., A survey on large language models: Applications, challenges, limitations, and practical usage, Authorea Preprints (2023).

[35] J. Kaddour, J. Harris, M. Mozes, H. Bradley, R. Raileanu, R. McHardy, Challenges and applications of large language models, arXiv preprint arXiv:2307.10169 (2023).

[36] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, et al., Time-llm: Time series forecasting by reprogramming large language models, arXiv preprint arXiv:2310.01728 (2023).

[37] A. Nazir, A. K. Shaikh, A. S. Shah, A. Khalil, Forecasting energy consumption demand of customers in smart grid using temporal fusion transformer (tft), Results in Engineering 17 (2023) 100888.

[38] J. Su, C. Jiang, X. Jin, Y. Qiao, T. Xiao, H. Ma, R. Wei, Z. Jing, J. Xu, J. Lin, Large language models for forecasting and anomaly detection: A systematic literature review, arXiv preprint arXiv:2402.10350 (2024).

[39] N. Gruver, M. Finzi, S. Qiu, A. G. Wilson, Large language models are zero-shot time series forecasters, Advances in Neural Information Processing Systems 36 (2023) 19622–19635.

[40] H. Xue, F. D. Salim, Promptcast: A new prompt-based learning paradigm for time series forecasting, IEEE Transactions on Knowledge and Data Engineering 36 (11) (2023) 6851–6864.

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

[42] M. Tan, M. Merrill, V. Gupta, T. Althoff, T. Hartvigsen, Are language models actually useful for time series forecasting?, Advances in Neural Information Processing Systems 37 (2024) 60162–60191.

[43] T. Zhou, P. Niu, L. Sun, R. Jin, et al., One fits all: Power general time series analysis by pretrained lm, Advances in neural information processing systems 36 (2023) 43322–43355.

[44] C. Chang, W.-Y. Wang, W.-C. Peng, T.-F. Chen, Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters, arXiv preprint arXiv:2308.08469 (2023).

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

[46] H. Xue, F. D. Salim, Utilizing language models for energy load forecasting, in: Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, 2023, pp. 224–227.

[47] T. Wu, Q. Ling, Stellm: Spatio-temporal enhanced pre-trained large language model for wind speed forecasting, Applied Energy 375 (2024) 124034.

[48] D. Han, W. Guo, H. Chen, B. Wang, Z. Guo, Lest: Large language models and spatio-temporal data analysis for enhanced sino-us exchange rate forecasting, International Review of Economics & Finance 96 (2024) 103508.

[49] Z. Lai, T. Wu, X. Fei, Q. Ling, Bert4st:: Fine-tuning pre-trained large language model for wind power forecasting, Energy Conversion and Management 307 (2024) 118331.

[50] X. Yu, Z. Chen, Y. Ling, S. Dong, Z. Liu, Y. Lu, Temporal data meets llm–explainable financial time series forecasting. arxiv 2023, arXiv preprint arXiv:2306.11025.

[51] G. Liu, Y. Bai, K. Wen, X. Wang, Y. Liu, G. Liang, J. Zhao, Z. Y. Dong, Lfllm: A large language model for load forecasting, Authorea Preprints (2024).

[52] Y. Wang, H. A. Karimi, Exploring large language models for climate forecasting, arXiv preprint arXiv:2411.13724 (2024).

[53] Z. Duan, C. Bian, S. Yang, C. Li, Prompting large language model for multi-location multi-step zero-shot wind power forecasting, Expert Systems with Applications (2025) 127436.

[54] W. Wang, Y. Luo, M. Ma, J. Wang, C. Sui, A novel forecasting framework leveraging large language model and machine learning for methanol price, Energy 320 (2025) 135123.

[55] T. Guo, E. Hauptmann, Fine-tuning large language models for stock return prediction using newsflow, arXiv preprint arXiv:2407.18103 (2024).

[56] E. Spiliotis, Time series forecasting with statistical, machine learning, and deep learning methods: Past, present, and future, in: Forecasting with Artificial Intelligence: Theory and Applications, Springer, 2023, pp. 49–75.

[57] L. Han, H.-J. Ye, D.-C. Zhan, The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting, IEEE Transactions on Knowledge and Data Engineering (2024).

[58] S. Hochreiter, Long short-term memory, Neural Computation MIT-Press (1997).

[59] A. Zeng, M. Chen, L. Zhang, Q. Xu, Are transformers effective for time series forecasting?, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 37, 2023, pp. 11121–11128.

[60] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, M. Long, itransformer: Inverted transformers are effective for time series forecasting, arXiv preprint arXiv:2310.06625 (2023).

[61] M. Shanker, M. Y. Hu, M. S. Hung, Effect of data standardization on neural network training, Omega 24 (4) (1996) 385–397.

[62] N. Fei, Y. Gao, Z. Lu, T. Xiang, Z-score normalization, hubness, and few-shot learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 142–151.

[63] IEA, World Energy Outlook 2023, IEA, Paris, 2023, licence: CC BY 4.0 (report); CC BY NC SA 4.0 (Annex A).
URL https://www.iea.org/reports/world-energy-outlook-2023

[64] X. Fang, S. Misra, G. Xue, D. Yang, Smart grid—the new and improved power grid: A survey, IEEE communications surveys & tutorials 14 (4) (2011) 944–980.

[65] A. Olabi, M. A. Abdelkareem, Renewable energy and climate change, Renewable and Sustainable Energy Reviews 158 (2022) 112111.

[66] J. Devlin, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

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

[68] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al., The llama 3 herd of models, arXiv preprint arXiv:2407.21783 (2024).