



Exoformer: An improved transformer architecture for long-term time series forecasting based on multi-source data

Ngo Van Son^{a,b},¹ Vo Viet Minh Nhat^{b,c},*

^a University of Sciences, Hue University, 77 Nguyen Hue, Thuan Hoa, Hue, Vietnam

^b School of Hospitality and Tourism – Hue University, 22 Lam Hoang, Vy Da, Hue, Vietnam

^c Department of Academic and Students' Affairs – Hue University, 03 Le Loi, Thuan Hoa, Hue, Vietnam

ARTICLE INFO

Keywords:

Time series forecasting

Transformer

Exogenous variables

Placeholder

Reversible instance normalization

ABSTRACT

Transformer models have achieved significant success in time series forecasting, but relying solely on endogenous information is insufficient to achieve high accuracy. To enhance predictive performance, some systems have used information from multiple sources, providing additional insights to improve forecasting accuracy. Integrating additional information from external sources involves adding factors not present in the primary source, which affects forecasting results. These additional factors, called exogenous variables, can enhance the forecasting ability. We propose ExoFormer, a Transformer-based architecture that incorporates exogenous information for long-term time series forecasting. By leveraging relative cross-attention and a decoder-only design, ExoFormer efficiently models dependencies between endogenous and exogenous variables. Experiments on seven benchmark datasets demonstrate that ExoFormer consistently outperforms state-of-the-art models in both accuracy and computational efficiency.

1. Introduction

Time series forecasting is a topic of interest in many research fields, from weather forecasting, climate change, economic growth, stock market fluctuations, to tourism demand. Forecasting is making short-term or long-term future assumptions based on knowledge of past events. Time series forecasting is important in many applications, such as finance, healthcare, energy, weather, etc. (Bartholomew, 1971). Recent advances in deep learning techniques, especially Transformer-based models (Vaswani et al., 2017), have significantly improved forecasting accuracy by capturing short-term and long-term dependencies in sequential data (Vaswani et al., 2017). However, accurate forecasting becomes a significant challenge in increasingly diverse and complex time series data. Traditional models such as ARIMA (Box & Jenkins, 1968), Holt-Winters (Winters, 1960), or even some recent deep learning architectures such as LSTM (Hochreiter & Schmidhuber, 1997) and CNN (LeCun et al., 1998), despite achieving competitive forecasting performance, predominantly rely on endogenous information, while the incorporation of informative exogenous variables remains limited. Recent research efforts have attempted to address the limitations of traditional models by integrating external signals. For example, Ngo and Vo (2025) proposed a TFT-TPE integrated model that combines Temporal Fusion Transformer with Tree-structured Parzen Estimator

(TPE) to enhance forecasting accuracy under abnormal conditions such as the COVID-19 pandemic. The model also incorporates Google Trends as exogenous covariates, showing significant improvement in tourism demand forecasting (Son & Nhat, 2025).

In recent years, the Transformer architecture with attention mechanism has opened a new direction in time series data processing, thanks to the ability to learn long-term relationships. However, most Transformer models focus on endogenous information, and time series data are assumed to be stationary and regular. This assumption is challenging because forecasting models are difficult to handle in non-stationary time series data due to their instability and unpredictability. Integrating exogenous information into Transformer-based forecasting models has been proposed to improve the forecasting quality. Specifically, TimeXer (Wang et al., 2024) introduced a cross-attention mechanism to combine external information with endogenous information to improve the forecasting quality; however, it does not support handling the non-stationarity of time series data. PatchTST (Zeng et al., 2023a) applies a data segmentation method that divides input time series into consecutive patches, enabling effective extraction of local temporal features and enhancing forecasting performance. However, PatchTST primarily focuses on modeling endogenous temporal patterns and does not explicitly incorporate exogenous variables in its original

* Corresponding author.

E-mail addresses: ngovanon@hueuni.edu.vn (N.V. Son), vmmnhat@hueuni.edu.vn (V.V.M. Nhat).

¹ The author contributed equally to this work as corresponding author.

design, as observed in our analysis. This focus may limit its applicability in scenarios involving heterogeneous input sources. One drawback of these proposed models is that they have high computational complexity due to fully connected FFN layers (Rumelhart et al., 1986), and normalization is required when dealing with diverse data. This paper proposes an improved Transformer model with a Relative Multihead Attention (RMA) mechanism used in cross-attention to improve the ability to model position-based dependencies in time series data. In addition, the Reversible Instance Normalization (RevIN) function (Kim et al., 2022) is also integrated to normalize the input time series before processing and restore it after forecasting to ensure stability across different datasets. A data flattening technique is also applied to improve the ability to learn trends and patterns in time series. The proposed model thus enhances the ability to exploit external information by integrating exogenous features into the Transformer architecture, called ExoFormer (Exogenous Transformer). By integrating many additional techniques and functions, ExoFormer learns endogenous relationships in the data and takes advantage of exogenous information, which can provide more context and improve forecasting ability.

To provide a clearer experimental narrative, this study is guided by the following research questions. (RQ1) Does explicitly integrating exogenous variables via cross-attention improve long-horizon forecasting accuracy compared to models that primarily rely on endogenous information? (RQ2) Is the proposed architecture more robust as the forecasting horizon increases, especially under long-range dependencies? (RQ3) How does ExoFormer perform on non-stationary and highly volatile time series, where distribution shifts and noise are prominent? (RQ4) Can ExoFormer achieve these improvements while maintaining favorable computational complexity compared to existing Transformer-based approaches? The experimental evaluation in Section 4 is structured to directly address these questions.

The main contributions of the paper include:

- Proposing an improved Transformer architecture, called ExoFormer, with a decoder that integrates patching-embedding a homogeneous processing mechanism for input sequences, which is capable of effectively exploiting the relationship between the target variable and the exogenous variable without having to design each type of variable separately.
- Proposing a conditional decoder formulation based on learnable placeholder queries, which serve as a structural design choice to separate historical representation from future prediction.
- Integrating the Reversible Instance Normalization (RevIN) function for endogenous and exogenous variables to stabilize the input data and improve the model performance on datasets with non-stationary and time-varying distributions.
- Proposing the Positionwise Stochastic Dropout mechanism to adjust the probability of dropping each patch according to the time position, helping the model learn the time-varying uncertainty and limit overfitting in data regions far from the forecast time.
- Propose an improved Relative Multihead Attention mechanism with relative positional bias, which improves the ability to model positional relationships in time series data.
- Experiment on various data sets to demonstrate the superiority of ExoFormer over other similar forecasting models.

The rest of the paper is organized as follows. Section 2 reviews and evaluates recent related works, focusing on Transformer-based time series forecasting models and exogenous variable integration methods. Section 3 details the ExoFormer architecture, including an improved Relative Multihead Attention mechanism, RevIN functionality integration, and endogenous-exogenous information fusion via cross-attention. Section 4 presents the experimental process and compares the performance of ExoFormer with other similar models. Finally, Section 5 concludes and provides future research directions.

2. Related works

With the increasing demand for time series forecasting, traditional statistical models such as ARIMA and Holt-Winters primarily rely on historical observations of the target series itself (Bartholomew, 1971; Hyndman & Athanasopoulos, 2018). Although recurrent neural networks such as LSTM enhance nonlinear modeling capacity, they are typically trained on endogenous sequences and lack explicit mechanisms to incorporate heterogeneous exogenous factors, which may limit their effectiveness in complex real-world forecasting tasks (Lim et al., 2021). The advent of Transformer architecture has opened up a new direction, thanks to the attention mechanism that allows capturing long-term and nonlinear relationships in data. Many recent works have extended the Transformer architecture to integrate exogenous factors, providing context and additional information from external data, thereby improving forecasting accuracy. Forecasting with exogenous factors can be deployed in many fields, such as e-commerce, forecasting demand for medical supplies, electricity consumption, manufacturing of goods, and even tourism demand. There have been different proposals to integrate exogenous factors into forecasting. This section analyzes and evaluates recent studies on forecasting long-term time series data combined with exogenous variables.

To address the challenges of long-term time series forecasting, recent research has converged on three main directions: positional encoding to provide temporal context, exploiting intrinsic temporal structures, and integrating heterogeneous data sources (see Fig. 1).

Positional encoding plays a pivotal role in enabling Transformers, originally designed for natural language processing (NLP), to adapt to sequential data. In NLP, Vaswani et al. (2017) introduced this mechanism by employing sinusoidal functions to embed order information into the input representation, thereby allowing the model to capture positional relationships among tokens. In the context of time series forecasting, Li et al. (2019) pioneered the use of positional encoding in the LogTrans model, aiming to preserve the ability to model long-range dependencies while alleviating memory and computational bottlenecks, although limitations remain for very long sequences. Extending this approach, PatchTST (Zeng et al., 2023a) represents a major advancement in the positional encoding direction. Instead of relying on sinusoidal or learnable positional encodings, PatchTST segments the time series into fixed-length patches and treats each patch as an input token. This patch embedding approach reduces the sequence length and implicitly encodes the order and structure of the series, thereby substituting for traditional positional encodings. The core contribution of PatchTST thus lies in innovating the input representation mechanism, enabling the model to capture long-range dependencies in time series forecasting better.

Beyond studies focusing on positional encoding for time series, a growing body of research has explored architectural innovations designed to process and capture temporal dependencies. Zhou et al. (2021) proposed the Informer model with a ProbSparse attention mechanism for long time series, which can reduce computational cost. Experimental results show that Informer achieves good forecasting performance on large data sets. However, it mainly exploits endogenous information and fails to use exogenous information to improve forecasting quality. Wu et al. (2021) developed Autoformer with a decomposition method to separate the trend and seasonal components, thereby improving the long-term forecasting performance. Similarly, FEDformer was proposed by Zhou et al. (2022), and Autoformer applies a frequency boosting technique to better capture temporal patterns. Autoformer and FEDformer both achieve good results, but neither fully incorporates exogenous factors and requires a rather complex hyperparameter tuning process. In addition to these models, Crossformer (Zhang & Yan, 2023) introduces a novel two-stage attention mechanism to jointly capture both temporal dependencies and cross-variable interactions. Specifically, the model applies time-axis attention to learn long-term temporal relations and dimension-axis attention to

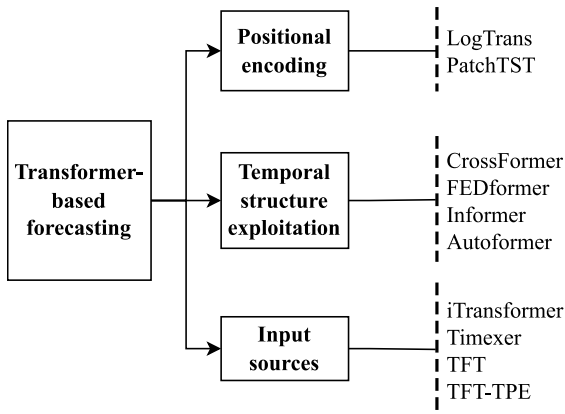


Fig. 1. Taxonomy of Transformer-based time series forecasting models, categorized by their primary design focus: positional encoding, temporal structure exploitation, and integration of heterogeneous input sources.

model correlations across multiple variables. This architectural design enables Crossformer to effectively exploit the multivariate nature of time series while maintaining the ability to capture long-range dependencies. Despite its improved performance on multivariate forecasting tasks, Crossformer still inherits the relatively high computational cost of attention-based models and does not explicitly address the incorporation of exogenous information.

In Transformer-based time series forecasting, the integration of heterogeneous data sources has been shown to play a crucial role in enhancing predictive performance. Liu et al. (2024) introduced iTransformer, which inverts the traditional time-variable design of Transformers by applying self-attention across the variable dimension instead of the temporal dimension. This architectural inversion allows the model to explicitly capture inter-variable dependencies, making it highly effective for multivariate time series forecasting. By emphasizing cross-variable relations, iTransformer reduces the reliance on positional encodings along the temporal axis while improving the ability to leverage heterogeneous input sources. Empirical results show that iTransformer achieves competitive accuracy on various benchmarks, particularly in scenarios with rich multivariate structures, although it still inherits the computational overhead common to attention-based models. Wang et al. (2024) proposed TimeXer, a variant of Transformer specifically designed to incorporate exogenous information through a cross-variable self-attention mechanism. Although TimeXer has achieved impressive results on many forecasting problems, using linear feed-forward layers in the architecture results in a model with many parameters and, thus, high computational cost. Lim et al. (2021) introduced Temporal Fusion Transformer (TFT), a multi-horizon forecasting model that can integrate known static covariates and temporal variables through gating and attention mechanisms. TFT not only achieves high forecasting efficiency but also provides a clear explanation of how the variables influence each other. However, the TFT architecture is quite complex, requiring a large amount of computation and heavily relying on variable selection through data preprocessing to classify relevant variables. A recent enhancement of TFT is the TFT-TPE model (Son & Nhat, 2025), which integrates TPE for automated hyperparameter tuning and introduces Google Trends as exogenous variables. This approach proves effective in capturing abnormal patterns in time series, especially during the COVID-19 period, demonstrating that search trends can be valuable contextual data in multivariate forecasting scenarios.

Unlike TFT and its variants, which explicitly leverage known future covariates through gated variable selection, ExoFormer is intentionally designed to use only historical observations of both the target and exogenous variables. This simplified design trades off the ability to exploit known future exogenous information at the forecasting

stage for a more streamlined architecture with reduced modeling complexity. This comparison highlights an important trade-off space in Transformer-based forecasting, where architectural simplicity and efficiency may come at the expense of richer conditioning on auxiliary information. Transformer-based forecasting models have achieved substantial progress by capturing long-term dependencies and nonlinear relationships in time series data. Variants such as Informer, Autoformer, FEDformer, PatchTST, TimeXer, and TFT have introduced important innovations for handling long sequences, improving computational efficiency, and integrating exogenous information. Their main advantages include:

- Self-attention mechanisms, which enable the capture of long-range dependencies that are challenging for traditional recurrent architectures.
- Decomposition and frequency-based techniques, such as trend-seasonality separation in Autoformer or frequency-enhanced attention in FEDformer, which allow models to extract clearer temporal patterns and improve interpretability.
- Integration of exogenous variables, as demonstrated in TimeXer and TFT, where incorporating auxiliary signals or cross-variable attention improves accuracy, particularly in multivariate forecasting.

Despite these advances, existing models still face several limitations and open challenges:

- Inefficient integration of endogenous and exogenous information, often treated separately, which limits the ability to capture the holistic context of the data.
- Complex architectures and heavy reliance on encoder-decoder structures, which increase training costs, especially for long time series.
- Difficulty in maintaining robustness across datasets with varying sequence lengths, sampling frequencies, and domain-specific characteristics.

These challenges highlight opportunities for developing new approaches that:

- Optimize the joint modeling of endogenous and exogenous information to exploit multi-source data better.
- Simplify architectural design to reduce computational overhead while preserving forecasting accuracy, particularly in long-term settings.
- Conduct comprehensive evaluations across diverse datasets to assess robustness and generalization.

The above analysis shows that although Transformer and its variants have achieved impressive results for time series forecasting, most have not yet made good use of exogenous factors and still have high computational complexity. This paper proposes the ExoFormer model, an improved Transformer architecture in which the decoder block is designed uniformly for both the primary and exogenous variables through the patching-embedding mechanism. In addition, the RevIN (Kim et al., 2022) function and data flattening technique are integrated to reduce the number of parameters and improve forecasting performance. The following section will describe the proposed ExoFormer model in detail.

3. ExoFormer model

3.1. Problem description

The objective of time series forecasting is to determine the value of a series at one or more future points in time based on past observations. At each time step t , the observed value of the target series is referred to as an endogenous variable. However, to increase the accuracy of the

forecast, other external information can also be used. External information is called exogenous variables, which can help the forecasting model exploit additional information and thus improve the forecasting accuracy. With the assumed input as

$$X = \{ \mathbf{x}_t = (x_t^1, x_t^2, \dots, x_t^d)^T \in \mathbb{R}^d \mid t = 1, 2, \dots, l \} \quad (1)$$

where d is the number of input variables, including endogenous variables and exogenous variables, and l is the length of the look-back window; the Transformer-based forecasting system will process each input variable independently, then collect them into patches of length w to transfer to the embedding space, creating input tokens. Assuming the system focuses on forecasting only a single endogenous variable,

$$X_t = \{ x_t \mid t = 1, 2, \dots, h \} \quad (2)$$

where h is the length of the forecast horizon, the proposed forecasting system architecture includes the following main components: (i) reversible normalization, (ii) patch-based sequence segmentation and feature space mapping, (iii) conditional decoding mechanism using learned queries (placeholders), and (iv) back-projection to a univariate forecast sequence.

3.2. ExoFormer architecture

Despite substantial advances in Transformer-based forecasting models, three notable limitations remain: (i) endogenous and exogenous variables are often handled independently, which weakens the model's ability to capture their joint temporal relationships; (ii) many existing architectures depend on heavy encoder-decoder stacks with high computational overhead, making them less suitable for long look-back windows; and (iii) current models show inconsistent robustness across datasets that differ in noise level, sampling frequency, and temporal structure. These limitations motivate the architectural choices behind ExoFormer. To address (i), ExoFormer introduces a unified patching-embedding mechanism together with relative cross-attention to jointly learn from endogenous and exogenous variables. To mitigate (ii), the model removes the encoder entirely and adopts a streamlined decoder-only structure, which substantially reduces computational burden compared to traditional designs. To overcome (iii), ExoFormer incorporates RevIN and positionwise stochastic dropout to enhance adaptability under distribution shifts and heterogeneous data conditions. This targeted "gap → design choice" mapping clarifies the rationale behind the architecture and highlights ExoFormer's compactness, effective use of exogenous signals, and improved computational efficiency.

With these motivations established, we now present the overall architecture of ExoFormer. ExoFormer draws inspiration from TimeXer (Wang et al., 2024) and PETformer (Lin et al., 2024), while introducing modifications that simplify the structure and enhance forecasting flexibility. The integrated components of ExoFormer are depicted in Figure 2.

3.2.1. Reversible instance normalization

In time series forecasting, a significant challenge is non-stationarity (Liu et al., 2023), meaning that the statistical distribution of the series, such as mean and variance, can change significantly between different periods or series. Non-stationarity makes it difficult for deep learning models to learn consistent rules and generalize well. RevIN (Kim et al., 2022) can solve this problem by normalizing each instance and bringing the series to the same uniform distribution before feeding them into the forecasting model. RevIN helps the forecasting model focus on learning dynamic patterns instead of being influenced by factors such as amplitude or long-term trends. Not only that, but RevIN also provides the ability to restore the output to the original scale, which is especially important in forecasting problems that require meaningful outputs, such as temperature, traffic, price, etc.

With the ability to handle distributional transformations efficiently while preserving quantitative information, RevIN becomes an important component in increasing the accuracy and stability of time series forecasting models, especially when data comes from many different domains or has strong time-varying characteristics. In the ExoFormer architecture, RevIN is applied to endogenous inputs and exogenous features, helping the model achieve higher performance when handling data sets with many variables, diverse distributions, and high volatility, standard features in modern multi-source time series forecasting applications.

3.2.2. Patching and embedding

With n_{var} input sequences, each corresponding to a variable, denoted as $X^i \in \mathbb{R}^l$, $i = 1, 2, \dots, n_{\text{var}}$, each X^i represents the time series corresponding to the i th input variable and the observation window of length l . After normalization, X^i is divided into sub-patches of length w and forms a matrix

$$X_{\text{patch}}^i \in \mathbb{R}^{n \times w} \quad (3)$$

with

$$n = \left\lfloor \frac{l-w}{s} + 1 \right\rfloor \quad (4)$$

where s is the stride. Each sub-patch is then mapped to a vector of size d_{model} by a shared linear layer resulting in:

$$X_{\text{token}}^i \in \mathbb{R}^{n \times d_{\text{model}}} \quad (5)$$

3.2.3. Relative multihead attention

We propose the Relative Multihead Attention mechanism, a multi-head improvement from the classic Multihead Attention that integrates two key components: (1) relative position bias to enhance the ability to learn positional dependencies in time series and (2) learnable placeholders that act as queries in cross-attention.

From a modeling perspective, forecasting with exogenous variables is inherently a conditional generation problem, where future target values should be predicted by selectively attending to relevant historical patterns under the influence of auxiliary signals. Cross-attention provides a principled mechanism for such conditional fusion: future representations (queries) retrieve informative content from the historical memory (keys/values) composed of both endogenous and exogenous variables. Compared to early or feature-level fusion, cross-attention enables time-dependent and context-aware integration, allowing the model to dynamically adjust the contribution of exogenous information at each forecast step. This design aligns naturally with long-horizon forecasting, where the relevance of external factors varies across future time points.

ExoFormer inherits the idea of placeholders from PETformer (Lin et al., 2024) but adapts the implementation to accommodate the cross-attention architecture in the decoder. In PETformer, placeholders are directly connected to the end of the input sequence and processed with historical data through the self-attention layer in the encoder. The use of learnable placeholders is motivated by the need to decouple historical representation from future prediction in long-horizon forecasting. Since future values are not observed, forcing the decoder to rely on shifted targets or autoregressive inputs may introduce error accumulation and unstable training. Placeholders instead act as abstract query anchors, each corresponding to a future temporal segment, enabling the model to formulate predictions purely through retrieval from historical context. Furthermore, relative positional bias complements this design by encoding temporal proximity directly into attention scores. In long time series, relative distances often convey more meaningful inductive bias than absolute positions, allowing the decoder to generalize across varying horizons while preserving distance-aware dependencies. This design creates a sequence of $n + m$ tokens, where n is the number of past steps, and m is the number of future steps, allowing the past

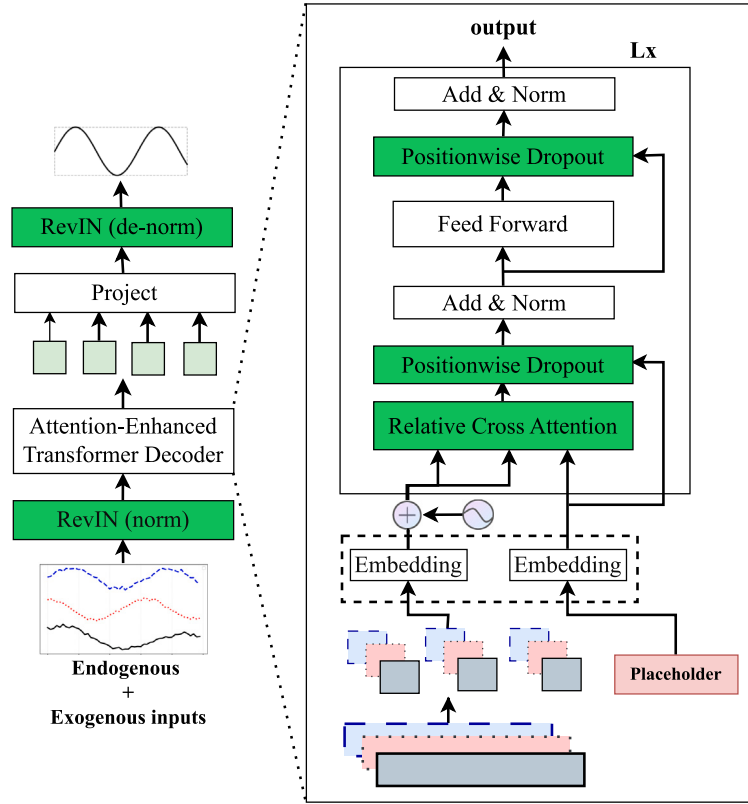


Fig. 2. ExoFormer architecture with decoder-only design and relative cross-attention over jointly embedded endogenous and exogenous variables.

and future representations to coexist in the same space. With the ExoFormer architecture, placeholders are implemented as a sequence of learnable queries in the decoder instead of processing the same input. The placeholders are designed to correspond to each forecast time step and do not go through the encoder, which separates the process of representing historical information from generating queries for the forecast. To represent the output sequence to be predicted in the future with the length h , placeholders are initialized as a unique learnable vector $p \in \mathbb{R}^{d_{\text{model}}}$ and are repeated m times to generate a sequence of learnable queries:

$$m = \left\lfloor \frac{h-w}{s} + 1 \right\rfloor \quad (6)$$

Each vector in this sequence represents a patch to be predicted. Therefore, the entire placeholder sequence is represented as a matrix:

$$P = \text{Repeat}(p, m), \quad P \in \mathbb{R}^{m \times d_{\text{model}}} \quad (7)$$

The matrix P will act as the set of input queries $Q = P$ of the decoder in ExoFormer, interacting with the tokens of the head X through the cross-attention mechanism to generate future time representations. The cross-attention mechanism is central to the decoder architecture of ExoFormer, allowing the model to flexibly integrate information between the past sequence and the future representation. Here, the placeholders act as queries Q , while the representations of the past sequence after encoding take on the roles of keys K and values V . Unlike TimeXer (Wang et al., 2024), where K and V are separated for exogenous variables and embedding according to the iTransformer structure, ExoFormer processes the entire input sequence, including both the target and exogenous variables, by dividing it into fixed patches. The results are then mapped to the representation space through the embedding layer. Patching helps the model extract segment-level information while embedding maps these segments to a higher-dimensional representation space. We add a relative position bias component to better model the temporal relationship between data

segments. Instead of simply computing the attention score as a scalar product QK^T , we add a learned bias matrix

$$\text{Bias} \in \mathbb{R}^{H \times (2 \cdot \text{max_pos} - 1)} \quad (8)$$

where each value corresponds to the relative distance between a query and a key. This matrix is mapped to the difference index between the query and key positions, and then added to the attention score before softmax normalization. This mechanism helps the model to be more aware of the temporal context, which is especially useful in time series forecasting tasks with clear sequential structures. The final attention process is calculated by the formula:

$$\text{Attention}(Q, K, V) = \text{Softmax} \left(\frac{QK^T}{\sqrt{d_k}} + \text{Bias} \right) V \quad (9)$$

where $Q \in \mathbb{R}^{m \times d_{\text{model}}}$ represents a sequence of placeholders consisting of m learnable queries, each of which corresponds to a patch in the future time series to be predicted. The value of m is calculated by formula (6). Similarly, $K, V \in \mathbb{R}^{n \times d_{\text{model}}}$ represents the encoded past series with the number of input time segments n .

Compared with TimeXer (Wang et al., 2024), ExoFormer makes a critical adjustment. Specifically, in TimeXer, the query embeds the target variable after self-attention. At the same time, the key/value is the embedding of the exogenous variable sequence, which is processed separately according to the iTransformer structure (Liu et al., 2024). This creates a separate information flow: the model learns to predict the target variable by relying on exogenous features. Meanwhile, ExoFormer does not separate the target and exogenous data at the encoding stage but instead takes the entire historical sequence (including both the target and exogenous) as the key/value. At the same time, the query is a learnable representation that is entirely independent of the real data. This allows ExoFormer to learn the query more flexibly while maintaining a simple model and allowing each prediction step to select the most relevant information from the historical sequence while ensuring that the model does not directly access the actual output data during training.

3.2.4. Positionwise stochastic dropout

Traditional dropout, which randomly deactivates neurons during training to prevent overfitting (Srivastava et al., 2014), is commonly applied in neural networks. Building upon this mechanism, ExoFormer extends traditional dropout into a Positionwise Stochastic Dropout, which is implemented after the cross-attention layer and the feedforward network. Unlike traditional dropout, which uses a fixed dropout probability across the entire model, Positionwise Stochastic Dropout assigns each patch in the series a unique dropout probability based on the position of each patch. The main idea behind this mechanism is that the current positions near the forecasting series often contain more important and reliable information than the more distant positions, which are susceptible to noise and high uncertainty. Therefore, Positionwise Stochastic Dropout prioritizes retaining the near segments while increasing the probability of dropping the distant segments during training. This approach helps the model learn to forecast flexibly without being overly dependent on uncertain information from far away. Specifically, at each patch, a binary value (0 or 1) is sampled from a Bernoulli distribution with a retention probability p determined by the position of the patch. If the sampled value is 0, the corresponding feature vector of that patch is excluded from the computation, effectively “dropping out” the patch. As a result, the model can learn a more efficient and adaptive strategy for using location information. At each output patch (a total of m patches), the retention probability is linearly interpolated from p_{start} to p_{end} , according to the formula:

$$p_t = p_{\text{start}} + \frac{t}{m-1} (p_{\text{end}} - p_{\text{start}}), \quad t = 0, 1, \dots, m-1 \quad (10)$$

where p_{start} is the retention probability of the first patch in the output sequence, and p_{end} is the retention probability of the last patch in the output sequence. For each location t , we generate a binary variable z_t from the Bernoulli distribution with a retention probability of $1 - p_t$:

$$z_t \sim \text{Bernoulli}(1 - p_t) \quad (11)$$

The entire feature vector at patch t , denoted by x_t , will be discarded or retained when multiplied by the variable z_t as follows:

$$\tilde{x}_t = z_t \cdot x_t \quad (12)$$

If $z_t = 0$, the entire representation at that location is discarded; if $z_t = 1$, the representation is retained. This process is performed independently at each patch and only during the training phase, helping the model learn more diverse representations, limiting overfitting, and at the same time, correctly reflecting the decreasing importance of distant locations in the forecast sequence.

3.2.5. Computational complexity

Unlike the standard Transformer, which adopts a stacked encoder-decoder architecture with quadratic self-attention, ExoFormer retains only the decoder equipped with relative cross-attention. This design entails two major implications for computational complexity. The computational complexity of ExoFormer is fundamentally different from conventional Transformer architectures due to its patching mechanism and the joint processing of endogenous and exogenous variables. Instead of applying self-attention over the entire historical sequence of length l , ExoFormer generates queries of length h for the forecasting horizon, while the keys and values are constructed from the patched historical sequences that incorporate both endogenous and exogenous variables.

As described in Eq. (4), for an input sequence of length l with patch length w and stride s , the number of patches n is approximately proportional to $\frac{l}{s}$. Taking into account the number of variables d , stride s , and forecasting horizon h , the computational complexity at the decoder layer can be expressed as $O\left(\frac{d}{s^2} hl\right)$ (see Table 1).

This expression shows that the computational cost grows linearly with both the historical length l and the forecasting horizon h , with an adjustment factor of $\frac{d}{s^2}$. The number of variables d increases the

cost proportionally, which is especially important in multivariate forecasting tasks involving hundreds of exogenous features. Conversely, increasing the stride size s reduces the number of patches quadratically, thereby significantly lowering the cost and allowing a trade-off between computational efficiency and representational capacity.

If the number of variables d and the stride size s are bounded and do not scale with the sequence length, they can be treated as constants. In this case, the complexity reduces to: $O(hl)$. This indicates that the cost grows linearly with both the historical length and the forecasting horizon, representing a clear improvement over standard Transformers, which exhibit a complexity of $O(dl^2)$. In an even more constrained scenario, when the number of variables d , the stride size s , and the forecasting horizon h are all treated as constants independent of the input length l , the complexity further reduces to $O(l)$. This means that the computational cost grows only linearly with the input length. As a result, ExoFormer demonstrates strong scalability for long-term forecasting tasks while being more efficient than conventional quadratic-complexity Transformers.

Moreover, the relative positional encoding mechanism introduces only a constant overhead, leaving the asymptotic complexity unchanged. Additional components, such as RevIN and position-wise dropout, remain linear in cost and negligible compared to attention operations. In summary, ExoFormer achieves higher computational efficiency by eliminating the encoder and replacing quadratic self-attention with cross-attention, while simultaneously preserving the ability to effectively incorporate exogenous variables.

4. Simulation and analysis

4.1. Data collection

The ExoFormer model is evaluated on a long-term univariate time series forecasting problem using exogenous variables. The goal is to verify the forecasting performance, generalization ability, and adaptability according to the input length. In the experimental part, the lengths of the lookback windows used are $L \in \{96, 192, 336, 512, 720\}$, corresponding to the forecast horizons $T \in \{96, 192, 336, 720\}$. In practical applications, the forecast horizon (forecasting horizon, or PreLen) is often pre-determined according to business requirements (e.g., 1 day, 1 week, 1 month). In contrast, the input length (Lookback) is an adjustable hyperparameter that optimizes forecasting accuracy. Therefore, the input length is not fixed in the experimental design. However, the model is run with different Lookback values (96, 192, 336, 512), and the best result with the lowest MSE is selected for each dataset and forecast horizons. This approach better reflects real-world applications where the model needs to be tuned to the input to suit each data feature and specific forecasting goal. With the default input $L = 96$, ExoFormer is compared with TimeXer (Wang et al., 2024), PatchTST (Zeng et al., 2023a), iTransformer (Liu et al., 2024), Crossformer (Zhang & Yan, 2023), TiDE (Das et al., 2023), DLinear (Zeng et al., 2023b), and TimesNet (Wu et al., 2023).

The experiments are conducted with seven popular datasets from economics, energy, weather, and power systems. Each dataset selects a specific target variable for forecasting, while the remaining variables support exogenous variables. Table 2 summarizes the characteristics of the datasets.

The ETT datasets (ETTth1, ETTth2, ETTm1, and ETTm2) are high-frequency industrial time series with strong daily periodicity and increasing noise levels from hourly (ETTth) to 15-minute resolution (ETTm). These datasets are commonly used to stress-test a model's ability to capture fine-grained temporal dependencies, handle high sampling frequency, and maintain stability under noisy and non-stationary conditions, while leveraging multiple correlated exogenous sensor variables.

The Electricity dataset (ECL) exhibits clear daily and weekly seasonality with moderate volatility and a large number of correlated

Table 1

Computational complexity per layer of representative Transformer-based forecasting models. Complexity is expressed in terms of input sequence length l , forecast horizon h , number of variables d , and patch size s .

Model	Encoder layer	Decoder layer
Transformer (Vaswani et al., 2017)	$\mathcal{O}(d l^2)$	$\mathcal{O}(h(h+l))$
iTransformer (Liu et al., 2024)	$\mathcal{O}(d l^2)$	N/A
Crossformer (Zhang & Yan, 2023)	$\mathcal{O}\left(\frac{d}{s} l^2\right)$	$\mathcal{O}\left(\frac{h(h+l)}{s^2}\right)$
PatchTST (Zeng et al., 2023a)	$\mathcal{O}\left(\frac{d}{s} l^2\right)$	N/A
TimeXer (Wang et al., 2024)	$\mathcal{O}\left(\frac{d}{s} l^2\right)$	N/A
ExoFormer	N/A	$\mathcal{O}\left(\frac{d}{s} h l\right)$

Table 2

Description of the datasets.

No.	Dataset	Target variable	Exog. vars	Time steps	Frequency
1	ETTh1	OT (Oil Temperature)	6	17,420	1 h
2	ETTh2			17,420	1 h
3	ETTm1			69,680	15 min
4	ETTm2			69,680	15 min
5	ECL	Electricity consumption of a customer	320	26,304	1 h
6	Exchange	Selected rate	7	7588	1 d
7	Weather	T1 (air temperature)	20	52,696	10 min

exogenous variables (multiple customers). It is particularly suitable for evaluating how effectively a model can exploit exogenous information and long-term seasonal patterns in large-scale multivariate forecasting scenarios.

The Weather dataset consists of dense multivariate meteorological observations with relatively low volatility and smooth temporal dynamics. This dataset serves as a stability benchmark, assessing whether the proposed model can perform competitively on well-behaved series without overfitting or introducing unnecessary modeling complexity.

The Exchange Rate dataset represents a highly volatile and non-stationary economic time series with weak seasonality and strong irregular fluctuations. It is used to stress-test the robustness of the model under distribution shifts and noisy dynamics, where long-term forecasting is particularly challenging and exogenous signals play a limited but critical contextual role. We acknowledge that these benchmark datasets do not exhaustively represent all real-world forecasting scenarios. In particular, they mainly consist of regularly sampled numerical time series and do not cover irregular sampling, sparse event-driven data, or non-numeric modalities such as text or images. Furthermore, while widely adopted, these datasets may not fully capture domain-specific operational constraints encountered in real deployment. Nevertheless, they provide a diverse and controlled testbed to evaluate the core modeling capabilities targeted in this work, including long-horizon dependency modeling, exogenous variable integration, and robustness to non-stationarity. As described in Table 2, the ETT (Electricity Transformer Temperature) (Zhou et al., 2021) dataset contains transformers' load and oil temperature data, recorded every 15 min from 2016 to 2018. Electricity (ECL) (Weron, 2014) is the hourly electricity consumption data of 321 customers from 2012 to 2014, while Exchange Rate (Lai et al., 2018) records the daily exchange rates of 8 countries from 1990 to 2016. Weather (Wu et al., 2021) provides 21 meteorological indices (temperature, humidity, etc.) recorded every 10 min throughout 2020. All datasets are divided in time order. The ETT dataset is divided into a 6:2:2 ratio for training, validation, and testing. The remaining datasets' split ratio is 6:2:2 for training, validation, and testing, respectively.

4.2. Experimental setup

To ensure reproducibility and clarity, we provide a full description of the experimental design and hyperparameter configurations (see

Table 3

Hyperparameter configurations used in ExoFormer experiments.

Hyperparameter	Description	Value
d_{model}	Model dimension	256
d_{ff}	Feed-forward dimension	512
d_{layers}	Number of decoder layers	3
n_{heads}	Number of attention heads	32
patch_len	Patch length	24
stride	Stride length	24
PosDro _{start}	Positionwise Stochastic Dropout Start	0.1
PosDro _{end}	Positionwise Stochastic Dropout End	0.2
loss	Loss function	MSE
learning_rate	Adam optimizer	1×10^{-4}
batch_size	Batch size	256 (ETTh1, ETTh2), 128 (ETTm1, ETTm2), 32 (Traffic), 64 (Weather), 8 (ECL)
train_epochs	Training epochs	100

Table 3). The proposed ExoFormer is implemented in PyTorch and optimized with the Adam optimizer using a learning rate of $1e-4$. The Mean Squared Error (MSE) is adopted as the loss function. The model dimension is set to $d_{\text{model}} = 256$, while the feed-forward dimension is $d_{\text{ff}} = 512$. The decoder consists of three layers, each with 32 attention heads. Both the patch length and stride are fixed at 24 to segment the input sequence. Positionwise stochastic dropout is applied with a rate that increases linearly from 0.1 (at the first layer) to 0.2 (at the final layer). All models are trained for 100 epochs. Dataset-specific batch sizes are adopted to account for different data scales: 256 for ETTh1 and ETTh2, 128 for ETTm1 and ETTm2, 32 for Traffic, 64 for Weather, and 8 for ECL. This setting ensures stable training and efficient GPU utilization across diverse datasets.

Following standard long-term forecasting benchmarks, we evaluate ExoFormer on seven datasets (ETTh1, ETTh2, ETTm1, ETTm2, ECL, Exchange, Weather) using the official train/validation/test splits (6:2:2). Forecasting performance is assessed using MSE and MAE, consistent with prior works.

4.3. Main results

The experimental results in Table 4 show the forecasting performance of the models on different datasets, with forecast horizons $T \in \{96, 192, 336, 720\}$. The Mean Squared Error (MSE) and Mean Absolute Error (MAE) indices are used as evaluation metrics, where lower values indicate better performance. In addition, the average value (AVG) and the number of best results (1st Count) of each model are also summarized to reflect the forecasting performance comprehensively. These datasets are selected to represent different temporal characteristics. Specifically, ECL has apparent daily and weekly seasonality; Weather is a stable dataset with low noise and low volatility; The ETT group (including ETTh1, ETTh2, ETTm1, and ETTm2) belongs to the high-frequency time series type, has intense cycles with increasing noise

Table 4

Long-term forecasting results using exogenous variables on seven datasets. The AVG line is the average results at different forecast horizons, including 96, 192, 336, and 720 in each dataset. The 1st Count line shows the number of times each model achieved the best result under each configuration. The best results are shown in bold.

Dataset	T	ExoFormer		TimeXer		iTrans.		DLinear		PatchTST		Cross.		TiDE		TimesNet	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	96	0.232	0.336	0.238	0.350	0.299	0.403	0.433	0.480	0.339	0.412	0.265	0.364	0.405	0.459	0.342	0.437
	192	0.276	0.366	0.271	0.372	0.321	0.413	0.407	0.461	0.361	0.425	0.313	0.390	0.383	0.442	0.384	0.461
	336	0.326	0.403	0.337	0.405	0.379	0.446	0.440	0.481	0.393	0.440	0.380	0.431	0.418	0.464	0.439	0.493
	720	0.415	0.471	0.365	0.439	0.461	0.504	0.495	0.523	0.482	0.507	0.418	0.463	0.471	0.507	0.473	0.514
	AVG	0.312	0.394	0.303	0.392	0.365	0.442	0.444	0.486	0.394	0.446	0.344	0.412	0.419	0.468	0.410	0.476
Weather	96	0.001	0.026	0.001	0.026	0.001	0.026	0.001	0.025	0.001	0.027	0.004	0.048	0.001	0.025	0.002	0.029
	192	0.001	0.027	0.001	0.028	0.002	0.029	0.001	0.028	0.002	0.030	0.005	0.053	0.001	0.028	0.002	0.031
	336	0.001	0.028	0.001	0.029	0.002	0.031	0.002	0.029	0.002	0.032	0.004	0.051	0.002	0.029	0.002	0.031
	720	0.002	0.033	0.002	0.033	0.002	0.036	0.002	0.033	0.002	0.036	0.007	0.067	0.002	0.033	0.381	0.368
	AVG	0.001	0.029	0.001	0.029	0.002	0.031	0.002	0.029	0.002	0.031	0.005	0.055	0.002	0.029	0.097	0.115
ETTh1	96	0.051	0.174	0.057	0.181	0.057	0.183	0.059	0.185	0.055	0.178	0.133	0.297	0.059	0.184	0.059	0.188
	192	0.062	0.195	0.069	0.201	0.074	0.209	0.078	0.214	0.072	0.206	0.232	0.409	0.078	0.214	0.080	0.217
	336	0.070	0.209	0.076	0.217	0.084	0.223	0.093	0.240	0.087	0.231	0.244	0.423	0.093	0.240	0.083	0.224
	720	0.075	0.218	0.084	0.229	0.084	0.229	0.106	0.256	0.098	0.247	0.530	0.660	0.104	0.255	0.083	0.231
	AVG	0.065	0.199	0.072	0.207	0.075	0.211	0.084	0.224	0.078	0.215	0.285	0.447	0.083	0.223	0.076	0.215
ETTh2	96	0.126	0.277	0.132	0.280	0.137	0.287	0.136	0.286	0.136	0.285	0.261	0.413	0.136	0.285	0.159	0.310
	192	0.158	0.315	0.180	0.333	0.187	0.341	0.187	0.339	0.185	0.337	1.240	1.028	0.187	0.339	0.196	0.351
	336	0.180	0.344	0.179	0.341	0.221	0.376	0.231	0.384	0.217	0.373	0.974	0.877	0.231	0.384	0.232	0.385
	720	0.208	0.370	0.196	0.359	0.253	0.403	0.267	0.417	0.229	0.384	1.633	1.177	0.267	0.417	0.254	0.403
	AVG	0.168	0.327	0.172	0.328	0.199	0.352	0.205	0.356	0.192	0.345	1.027	0.873	0.205	0.356	0.210	0.362
ETTh1	96	0.026	0.122	0.026	0.123	0.029	0.128	0.030	0.129	0.029	0.126	0.171	0.355	0.030	0.129	0.029	0.128
	192	0.039	0.150	0.039	0.152	0.045	0.163	0.044	0.160	0.045	0.160	0.293	0.474	0.044	0.160	0.044	0.160
	336	0.053	0.175	0.052	0.174	0.060	0.190	0.057	0.184	0.058	0.184	0.330	0.503	0.057	0.184	0.061	0.190
	720	0.068	0.201	0.071	0.206	0.079	0.218	0.080	0.217	0.082	0.221	0.852	0.861	0.080	0.217	0.083	0.223
	AVG	0.047	0.162	0.047	0.164	0.053	0.175	0.053	0.173	0.053	0.173	0.411	0.548	0.053	0.173	0.054	0.175
ETTh2	96	0.062	0.182	0.067	0.188	0.071	0.194	0.074	0.199	0.068	0.188	0.149	0.309	0.073	0.199	0.073	0.200
	192	0.095	0.232	0.097	0.236	0.108	0.247	0.104	0.241	0.100	0.236	0.686	0.740	0.104	0.241	0.106	0.247
	336	0.122	0.267	0.128	0.273	0.140	0.288	0.131	0.276	0.128	0.271	0.546	0.602	0.131	0.276	0.150	0.296
	720	0.169	0.322	0.177	0.332	0.188	0.340	0.180	0.329	0.185	0.335	2.524	1.424	0.180	0.329	0.186	0.338
	AVG	0.112	0.251	0.117	0.257	0.127	0.267	0.122	0.261	0.120	0.258	0.976	0.769	0.122	0.261	0.129	0.271
Exchange	96	0.093	0.227	0.104	0.242	0.103	0.239	0.090	0.228	0.112	0.245	0.345	0.481	0.106	0.243	0.103	0.237
	192	0.215	0.353	0.235	0.370	0.204	0.346	0.176	0.333	0.227	0.355	0.640	0.656	0.200	0.338	0.245	0.374
	336	0.420	0.498	0.427	0.504	0.407	0.483	0.359	0.472	0.475	0.507	1.246	0.934	0.404	0.478	0.514	0.547
	720	0.739	0.679	1.007	0.764	1.071	0.788	0.880	0.717	1.388	0.891	1.604	1.066	1.082	0.795	1.239	0.860
	AVG	0.367	0.439	0.443	0.470	0.446	0.464	0.376	0.438	0.551	0.500	0.959	0.784	0.448	0.464	0.525	0.505
1st Count		26	27	14	7	2	0	4	5	2	0	0	0	5	3	0	0

levels from ETTh to ETTm. With Exchange, this is a highly volatile, nonlinear economic series with no clear cycle. These characteristics are described and illustrated in detail in Wu et al. (2021) through chain structure analysis, hysteresis diagrams, and experimental results.

To answer RQ1, we first evaluate the effectiveness of exogenous variable integration by comparing ExoFormer with models that do not explicitly model cross-variable dependencies. ExoFormer achieves outstanding performance for the ECL dataset with the best results in both MSE and MAE at three out of four forecast horizons. On average, overall timescales, ExoFormer achieves MSE = 0.312 and MAE = 0.394, which are significantly lower than models such as DLinear (Zeng et al., 2023b) (a linear approach that decomposes input series into trend and seasonal components) with MSE = 0.365 or TimesNet (Wu et al., 2023) with MSE = 0.410. This result shows that ExoFormer can learn exogenous variables effectively in data series with apparent seasonality. For the Weather dataset, the difference between the models is insignificant due to low noise and low volatility characteristics. ExoFormer achieves similar (and best) results to TimeXer and iTransformer (MSE = 0.001 and MAE = 0.029). This result shows that ExoFormer is relatively stable with “easy to learn” datasets.

For the ETT datasets, including ETTh1, ETTh2, ETTm1, and ETTm2, which are high-frequency time series with a clear daily cycle and increasing noise levels, ExoFormer continues to achieve high accuracy. On ETTh1, ExoFormer achieves the lowest average MSE (0.065), surpassing TimeXer (0.072) and TiDE (0.083). For ETTh2 and ETTm1, where the series shows more noise and volatility, ExoFormer continues to lead or equal TimeXer. RQ3 focuses on non-stationary and highly

volatile series. The Exchange and ETTm2 datasets provide representative benchmarks for this scenario. For the ETTm2 dataset with the highest noise level, ExoFormer still achieves the lowest average MSE (0.112) among the compared models. A notable point for the Exchange dataset is that, despite being characterized by high volatility and a strong nonlinear structure, ExoFormer still maintains superior forecasting performance. Specifically, ExoFormer achieves an average MSE of 0.367, which is lower than TimeXer (0.443) and significantly lower than Crossformer (0.959) and TimesNet (0.525). To address RQ2, we analyze model performance as the forecasting horizon increases from 96 to 720 time steps. Although some models, such as DLinear or PatchTST, perform exceptionally well at long forecast intervals (720), ExoFormer maintains stable accuracy across all forecast horizons.

Aggregated over all 56 evaluation configurations (7 datasets $\times$ 4 forecasting points $\times$ 2 metrics), ExoFormer achieved a total of 26 top scores, outperforming TimeXer (14 times), TiDE (5 times), DLinear (4 times), and the remaining models. This result demonstrates ExoFormer’s generalization ability and consistent accuracy in long-term time series forecasting problems using multi-source data and exogenous variables. At the same time, the results also reinforce the role of the improved attention architecture in effectively exploiting the complex characteristics of real-time series. Fig. 3 visually compares ExoFormer and TimeXer’s forecast results on six representative datasets: ECL, ETTh1, ETTh2, Exchange, ETTm2, and Weather, with a forecast period of $T = 96$. The black line represents the ground truth series, while the blue and red lines represent the forecast results of ExoFormer and TimeXer, respectively.

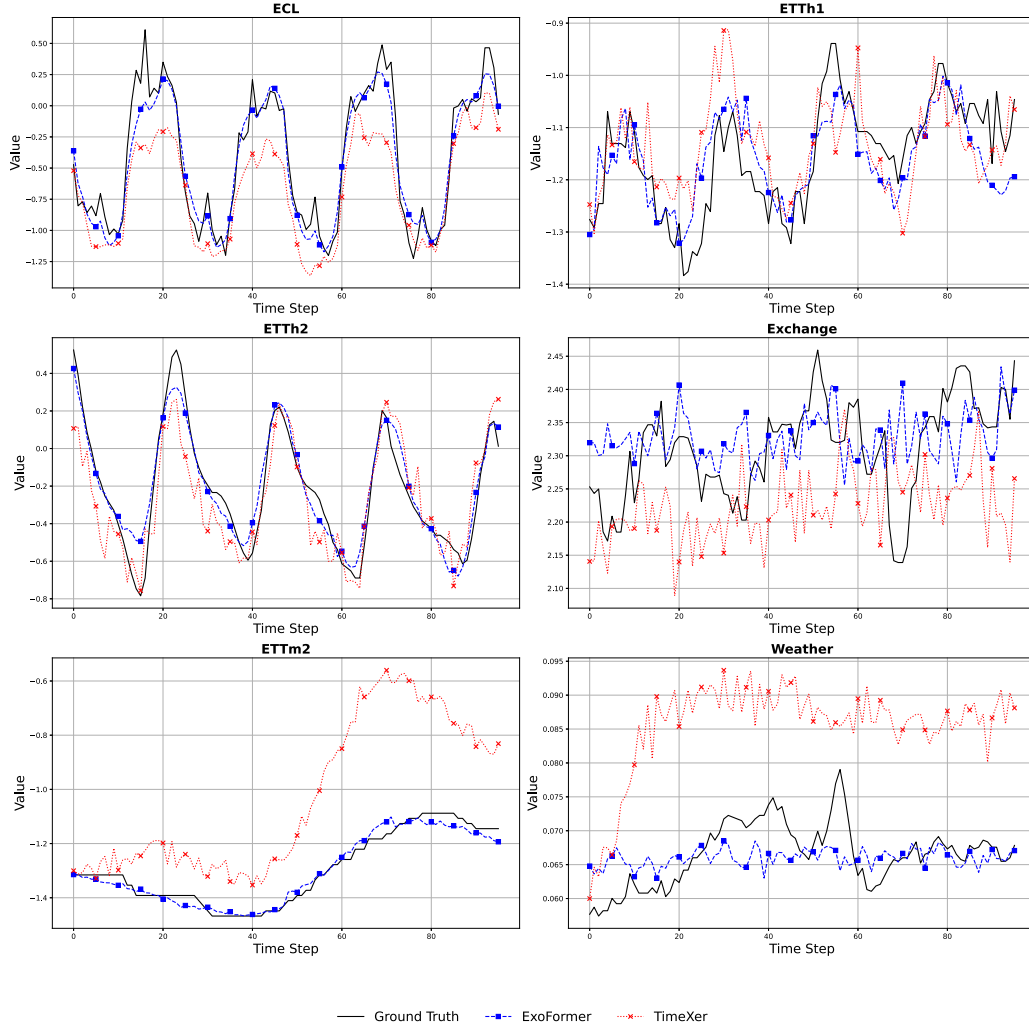


Fig. 3. Illustration of the prediction results with a length of 96 steps on a segment of the test set from seven data sets. The forecast results of ExoFormer and TimeXer are compared with the actual data.

ExoFormer and TimeXer are pretty good at forecasting time series trends across datasets. However, ExoFormer is more accurate in cases where the series are cyclical or noisy, such as ECL, Weather, and ETTm2. Meanwhile, TimeXer sometimes exhibits out-of-phase or less stable forecasts at longer periods. Overall, the visual results are consistent with the quantitative evaluation, suggesting that ExoFormer tends to be more stable in capturing the shape and trend of the forecasted data series. To answer RQ4, we jointly consider forecasting accuracy and computational efficiency. Across all evaluated datasets and forecasting horizons, ExoFormer achieves performance that is comparable to or consistently better than both Transformer-based and non-Transformer forecasting baselines, including linear, convolutional, and MLP-based models. At the same time, ExoFormer requires fewer parameters and lower computational cost, owing to its non-autoregressive decoding and lightweight cross-attention design. These results demonstrate that ExoFormer effectively balances predictive performance and computational efficiency, thereby validating the motivation behind its architectural design.

An important consideration when interpreting the results is that prediction horizons are defined in terms of step counts (96, 192, 336, 720), while the effective temporal span varies substantially across datasets due to different sampling frequencies. For example, a horizon of 96 corresponds to several months in the Exchange dataset but only hours in the Weather dataset. As a result, the notion of “long-term”

forecasting should be interpreted relative to the data frequency rather than as an absolute temporal duration.

Moreover, the advantages of ExoFormer are not uniform across all datasets and appear to be influenced by dataset characteristics such as temporal granularity, volatility, and the richness of exogenous variables. ExoFormer tends to achieve larger gains on datasets with strong multivariate interactions and longer effective forecasting spans (e.g., Electricity and Exchange), while its improvements are less pronounced on datasets with smoother dynamics or shorter effective horizons.

We also observe several cases where ExoFormer does not outperform all baselines, particularly at shorter horizons or on datasets with relatively stable temporal patterns, such as Weather and parts of the ETT benchmarks. These results indicate that ExoFormer is not universally superior and that simpler or more specialized models may remain competitive under certain conditions.

Overall, these findings suggest that the effectiveness of ExoFormer in long-term forecasting is closely tied to the interaction between forecasting horizon, data frequency, and multivariate structure, highlighting the importance of contextual interpretation when comparing long-horizon forecasting models.

4.4. Ablation studies

To isolate the contribution of the main architectural components in ExoFormer, we conduct a series of ablation studies in which individual

Table 5

Long-term forecasting ablation results of ExoFormer. Bold numbers in the ablated model columns indicate worse performance (higher error) compared to the original ExoFormer under the same dataset and forecast horizon.

Dataset	T	Models							
		ExoFormer		NoPosDrop		NoPosBias		NoRevIN	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	96	0.232	0.336	0.228	0.334	0.234	0.342	0.227	0.335
	192	0.276	0.366	0.268	0.356	0.270	0.356	0.287	0.369
	336	0.326	0.403	0.393	0.431	0.310	0.393	0.315	0.408
	720	0.415	0.471	0.506	0.525	0.391	0.459	0.430	0.474
	AVG	0.312	0.394	0.349	0.412	0.301	0.388	0.315	0.397
Weather	96	0.001	0.026	0.001	0.029	0.001	0.029	0.002	0.035
	192	0.001	0.027	0.001	0.030	0.001	0.030	0.003	0.039
	336	0.001	0.028	0.002	0.029	0.002	0.030	0.003	0.039
	720	0.002	0.033	0.002	0.033	0.002	0.033	0.003	0.042
	AVG	0.001	0.029	0.002	0.030	0.002	0.031	0.003	0.039
ETTh1	96	0.051	0.174	0.051	0.172	0.056	0.183	0.073	0.207
	192	0.062	0.195	0.061	0.190	0.061	0.191	0.250	0.429
	336	0.070	0.209	0.070	0.207	0.070	0.210	0.070	0.205
	720	0.075	0.218	0.078	0.216	0.090	0.233	0.350	0.528
	AVG	0.065	0.199	0.065	0.196	0.069	0.204	0.186	0.342
ETTh2	96	0.126	0.277	0.123	0.271	0.130	0.280	0.145	0.301
	192	0.158	0.315	0.162	0.322	0.164	0.323	0.204	0.358
	336	0.180	0.344	0.195	0.355	0.185	0.345	0.244	0.400
	720	0.208	0.370	0.251	0.388	0.195	0.355	0.367	0.488
	AVG	0.168	0.327	0.183	0.334	0.169	0.326	0.240	0.387
ETTm1	96	0.026	0.122	0.026	0.122	0.025	0.121	0.076	0.222
	192	0.039	0.150	0.040	0.152	0.040	0.153	0.157	0.331
	336	0.053	0.175	0.052	0.175	0.053	0.175	0.187	0.368
	720	0.068	0.201	0.073	0.206	0.073	0.205	0.198	0.376
	AVG	0.047	0.162	0.048	0.164	0.048	0.164	0.155	0.324
ETTm2	96	0.062	0.182	0.061	0.178	0.064	0.186	0.067	0.196
	192	0.095	0.232	0.098	0.240	0.093	0.230	0.095	0.240
	336	0.122	0.267	0.129	0.278	0.126	0.272	0.144	0.304
	720	0.169	0.322	0.175	0.328	0.175	0.327	0.166	0.329
	AVG	0.112	0.251	0.116	0.256	0.115	0.254	0.118	0.267
Exchange	96	0.093	0.227	0.104	0.232	0.111	0.245	0.782	0.762
	192	0.215	0.353	0.216	0.352	0.221	0.364	0.737	0.728
	336	0.420	0.498	0.410	0.478	0.389	0.486	1.954	1.225
	720	0.739	0.679	1.537	1.024	0.751	0.662	1.115	0.898
	AVG	0.367	0.439	0.567	0.522	0.368	0.439	1.147	0.903

modules are removed while keeping all other settings unchanged, including the training protocol, hyperparameters, and evaluation horizons. Specifically, we evaluate three ablated variants: NoRevIN, which removes the Reversible Instance Normalization module; NoPosBias, which disables the relative positional bias in the attention mechanism; and NoPosDrop, which replaces the proposed position-wise stochastic dropout with standard dropout. These ablations are performed across representative datasets and multiple forecasting horizons to ensure a fair and comprehensive comparison. The detailed ablation results are reported in Table 5.

Across most datasets and horizons, removing any of the core components results in a clear degradation of forecasting accuracy, as reflected by increased MSE and MAE values compared to the original ExoFormer. This consistent performance drop confirms that the improvements achieved by ExoFormer cannot be attributed solely to training choices, but instead arise from the combined effect of its architectural design. In particular, the impact of each component becomes more pronounced as the forecasting horizon increases, highlighting their importance for long-term prediction.

The effect of Reversible Instance Normalization (RevIN) is especially evident on non-stationary and highly volatile datasets. When RevIN is removed, NoRevIN exhibits substantial performance degradation, most notably on the Exchange and ETTm-series datasets. For example, on the Exchange dataset, the average MSE increases dramatically compared to the original model. This observation indicates that RevIN plays a crucial role in mitigating distribution shifts and stabilizing the learning process, enabling the model to focus on temporal dynamics rather than being affected by changing data scales and trends.

Removing the relative positional bias (NoPosBias) also leads to consistent performance deterioration, particularly at longer forecasting horizons. This effect is clearly observed on datasets with strong temporal dependencies, such as ECL and ETTm2, where errors increase markedly at horizons of 336 and 720 steps. These results demonstrate that relative positional bias is essential for capturing long-range temporal relationships, as it allows the attention mechanism to model relative time distances more effectively than absolute positional encoding.

The ablation of position-wise stochastic dropout (NoPosDrop) results in a moderate but systematic reduction in accuracy across most datasets. Although the degradation is less severe than that observed when removing RevIN, the increase in error becomes more evident as the prediction horizon grows. This behavior supports the design motivation of position-wise stochastic dropout, which regularizes the model in a horizon-aware manner and reflects the increasing uncertainty associated with farther future time steps, thereby improving robustness and generalization in long-term forecasting.

Regarding the placeholder mechanism, we note that it is not an optional or independent module that can be switched off without fundamentally altering the model. Placeholders serve as learnable query representations corresponding to future prediction steps in the non-autoregressive decoder. Removing them would eliminate the decoder's ability to generate multi-step forecasts altogether, rendering the architecture non-functional rather than merely reducing its accuracy. Consequently, a conventional "turn-off" ablation is not applicable for placeholders, and their contribution cannot be isolated in the same manner as other components without redesigning the overall model structure.

Overall, the ablation results provide strong empirical evidence that each core component of ExoFormer contributes meaningfully to its forecasting performance. The combined use of RevIN, relative positional bias, and position-wise stochastic dropout leads to consistent and robust improvements across diverse datasets and forecasting horizons, validating the architectural choices underlying ExoFormer for long-term time series forecasting with exogenous information.

5. Conclusion

This paper introduces ExoFormer, an enhanced Transformer-based architecture designed to effectively exploit exogenous information for long-term time series forecasting. Unlike conventional approaches that primarily emphasize endogenous variables, ExoFormer incorporates relative cross-attention with placeholders, an improved Relative Multi-head Attention mechanism, RevIN for adaptive stationarity restoration, and position-wise dropout to improve generalization and robustness.

From a theoretical perspective, ExoFormer addresses several gaps in existing literature. While many Transformer variants, such as Informer, Autoformer, FEDformer, PatchTST, and TimeXer, either ignore non-stationarity or rely solely on global normalization, ExoFormer explicitly integrates RevIN to adaptively restore stationarity. Furthermore, it overcomes the limitations of PatchTST in handling exogenous variables by employing placeholders and relative cross-attention to better align endogenous and exogenous information. The position-wise dropout also provides a novel way to regularize the forecasting horizon, reflecting increasing uncertainty over time. Empirically, ExoFormer demonstrates consistent advantages across seven widely used benchmark datasets. Compared to models like TFT, TimeXer, PatchTST, and DLinear, ExoFormer achieves higher forecasting accuracy (MSE, MAE) and improved stability without incurring high computational costs. Its robust generalization across volatile, cyclical, and high-frequency series confirms its potential for diverse real-world applications.

Nevertheless, several limitations remain. The interpretability of attention mechanisms and the role of future exogenous variables (whose values are assumed to be known at forecasting time) are not yet fully understood. In addition, the current evaluation does not explicitly analyze the impact of exogenous noise and missingness, the scalability of the model with respect to an increasing number of variables, or the sensitivity of forecasting performance to patching configurations. A more systematic evaluation of training cost and scalability across multi-domain datasets is still needed. Future research directions include: (i) visualization of attention to enhance interpretability, (ii) detailed analysis of computational efficiency and scalability, and (iii) extending ExoFormer to multi-task learning and online forecasting scenarios. Despite these limitations, ExoFormer represents a technical advancement for Transformer-based forecasting models, demonstrating effectiveness in integrating exogenous information and opening up prospects for more accurate and practical long-term forecasting solutions.

CRedit authorship contribution statement

Ngo Van Son: Methodology, Investigation, Software, Writing – original draft. **Vo Viet Minh Nhat:** Conceptualization, Supervision, Validation, Reviewing and editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors thank the School of Hospitality and Tourism – Hue University, Vietnam, for funding this project (Code DHH2025-10-45)

Data availability

Data will be made available on request.

References

- Bartholomew, D. J. (1971). *Time series analysis forecasting and control*. Duxbury Press Boston.
- Box, G. E. P., & Jenkins, G. M. (1968). Some Recent Advances in Forecasting and Control. *Applied Statistics*, 17(2), 91. <http://dx.doi.org/10.2307/2985674>.
- Das, A., Kong, W., Leach, A., Mathur, S. K., Sen, R., & Yu, R. (2023). Long-term forecasting with tiDE: Time-series dense encoder. *Transactions on Machine Learning Research*, URL <https://openreview.net/forum?id=pCbC3aQB5W>.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Hyndman, R., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice* (2nd ed.). Australia: OTexts.
- Kim, T., Kim, J., Tae, Y., Park, C., Choi, J. H., & Choo, J. (2022). Reversible Instance Normalization for Accurate Time-Series Forecasting Against Distribution Shift. In *ICLR 2022 - 10th International conference on learning representations* (pp. 1–25).
- Lai, G., Chang, W.-C., Yang, Y., & Liu, H. (2018). Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval* (pp. 95–104). New York, NY, USA: Association for Computing Machinery, <http://dx.doi.org/10.1145/3209978.3210006>.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2323. <http://dx.doi.org/10.1109/5.726791>.
- Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., & Yan, X. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, & R. Garnett (Eds.), vol. 32, *Advances in neural information processing systems*. Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2019/file/6775a0635c302542da2c32aa19d86be0-Paper.pdf.
- Lim, B., Arik, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4), 1748–1764. <http://dx.doi.org/10.1016/j.ijforecast.2021.03.012>.
- Lin, S., Lin, W., Wu, W., Wang, S., & Wang, Y. (2024). PETformer: Long-Term Time Series Forecasting via Placeholder-Enhanced Transformer. *IEEE Transactions on Emerging Topics in Computational Intelligence*, <http://dx.doi.org/10.1109/TETCI.2024.3502437>.
- Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., & Long, M. (2024). Itransformer: Inverted transformers are effective for time series forecasting. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, & Y. Sun (Eds.), vol. 2024, *International conference on representation learning* (pp. 11116–11140). URL https://proceedings.iclr.cc/paper_files/paper/2024/file/2ea18fde667e0ef2ad82b2b4d65147ad-Paper-Conference.pdf.
- Liu, Y., Li, C., Wang, J., & Long, M. (2023). Koopa: Learning non-stationary time series dynamics with koopman predictors. In *Thirty-seventh conference on neural information processing systems*. URL <https://openreview.net/forum?id=A4zzxu82a7>.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence*, 399–421. <http://dx.doi.org/10.1016/B978-1-4832-1446-7.50035-2>.
- Son, N. V., & Nhat, V. V. M. (2025). Improving forecasting performance for abnormal time series data with the TFT-tpe integrated model and google trends. *Bulgarian Academy of Sciences Cybernetics and Information Technologies*, 25(2), <http://dx.doi.org/10.2478/cait-2025-0017>.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56), 1929–1958, URL <http://jmlr.org/papers/v15/srivastava14a.html>.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), vol. 30, *Advances in neural information processing systems*. Curran Associates, Inc..
- Wang, Y., Wu, H., Dong, J., Qin, G., Zhang, H., Liu, Y., Qiu, Y., Wang, J., & Long, M. (2024). TimeXer: Empowering Transformers for Time Series Forecasting with Exogenous Variables. *Advances in Neural Information Processing Systems*, 37, 469–498.
- Weron, R. (2014). Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, 30(4), 1030–1081. <http://dx.doi.org/10.1016/J.IJFORECAST.2014.08.008>.
- Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3), 324–342.

- Wu, H., Hu, T., Liu, Y., et al. (2023). Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International conference on learning representations*.
- Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems*. URL <https://openreview.net/forum?id=J4gRj6d5Qm>.
- Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023a). A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI 2023*, 37, 11121–11128. <http://dx.doi.org/10.1609/aaai.v37i9.26317>.
- Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023b). Are Transformers Effective for Time Series Forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9), 11121–11128. <http://dx.doi.org/10.1609/aaai.v37i9.26317>.
- Zhang, Y., & Yan, J. (2023). Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *The eleventh international conference on learning representations*. URL <https://openreview.net/forum?id=vSVLM2j9eie>.
- Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. *Proceedings of Machine Learning Research*, 162, 27268–27286.
- Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence EAAI 2021, virtual event, February 2-9, 2021*, (pp. 11106–11115). AAAI Press, <http://dx.doi.org/10.1609/AAAI.V35I12.17325>.