PENGUIN: Enhancing Transformer with Periodic-Nested Group Attention for Long-term Time Series Forecasting

Tian Sun[1,2] **Yuqi Chen**[1,2,3] **Weiwei Sun**[1,2]
[1] School of Computer Science, Fudan University
[2] Shanghai Key Laboratory of Data Science, Fudan University
[3] Xiaohongshu Inc.

Abstract

Despite advances in the Transformer architecture, their effectiveness for long-term time series forecasting (LTSF) remains controversial. In this paper, we investigate the potential of integrating explicit periodicity modeling into the self-attention mechanism to enhance the performance of Transformer-based architectures for LTSF. Specifically, we propose *PENGUIN*, a simple yet effective periodic-nested group attention mechanism. Our approach introduces a periodic-aware relative attention bias to directly capture periodic structures and a grouped multi-query attention mechanism to handle multiple coexisting periodicities (e.g., daily and weekly cycles) within time series data. Extensive experiments across diverse benchmarks demonstrate that *PENGUIN* consistently outperforms both MLP-based and Transformer-based models. Code is available at https://github.com/ysygMhdxw/AISTATS2026_PENGUIN.

1 INTRODUCTION

Long-term time series forecasting (LTSF) is a pivotal task in various domains, including finance (Chen et al., 2019; Zhang et al., 2017), traffic (Zhang et al., 2018; Zheng et al., 2020; Chen et al., 2023; Sun et al., 2025), and healthcare (Chen et al., 2024; Yi et al., 2024), where accurate prediction of future values is essential for decision-making (Zhou et al., 2022; Zhang and Yan, 2023; Wang et al., 2024; Zhang et al., 2024; Li et al., 2022; Zhang et al., 2025). Transformers, with their ability to capture long-range dependencies (Vaswani, 2017), have shown promising results in sequence-based tasks (Touvron et al., 2023; Bai et al., 2023; Brown et al., 2020; Raffel et al., 2020). However, their effectiveness in LTSF remains contentious (Zeng et al., 2023). Recent studies suggest that simple linear models can outperform Transformer-based approaches (Xu et al., 2024; Lin et al., 2024; Xu et al., 2023), raising questions about whether Transformers are effective for long-term time series forecasting.

In this paper, we rethink the design principle of self-attention and demonstrate that, when enhanced with **P**eriodic-**N**ested **Gro**u**p** Attent**io**n mechanism (short for *PENGUIN*), self-attention can be highly effective for LTSF. Specifically, *PENGUIN* is designed to explicitly model temporal dependencies and intrinsic periodic patterns. *PENGUIN* incorporates two core components: a periodic-nested relative attention bias that directly captures periodic structures, and a grouped multi-query attention mechanism Ainslie et al. (2023) that effectively handles multiple coexisting periodicities. The motivations behind this design can be summarized in the following two aspects.

Firstly, time series data often exhibit periodic patterns (Xu et al., 2024; Lin et al., 2024; Qin et al., 2024; Wu et al., 2022; Elfeky et al., 2005), which conventional attention mechanisms struggle to capture over long horizons (Xu et al., 2024; Kim et al., 2024; Zeng et al., 2023). Existing methods, such as Autoformer (Wu et al., 2021) and FEDformer (Zhou et al., 2022), attempt to model these patterns implicitly via frequency decomposition. However, recent work (Xu et al., 2024) suggests that explicitly modeling periodicity yields better results. Motivated by this, we pioneer integrating explicit periodicity modeling within

Tian Sun and Yuqi Chen contributed equally to the research. Work was conducted during Yuqi Chen's affiliation with Fudan University. Correspondence to Weiwei Sun. contact: wwsun@fudan.edu.cn

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

the attention mechanism to enhance the LTSF performance, while retaining the strong modeling capability of Transformers.

Secondly, capturing multiple periodicities is essential for accurate time series forecasting, as real-world data often exhibits complex, overlapping temporal cycles (Xu et al., 2024; Wang et al., 2024). Existing methods, such as TimesNet (Wu et al., 2022), address this issue by decomposing time series into multiple temporal sequences, each corresponding to a different period length. Such an approach can be computationally inefficient. In contrast, we propose a group attention mechanism that partitions the multi-head attention into several groups, each dedicated to modeling a specific period length. To further improve efficiency, each group employs multi-query attention (Shazeer, 2019). This design facilitates more effective and specialized learning of diverse periodic patterns. In summary, the contributions of the paper are as follows:

- We pioneer the exploration of explicitly modeling periodicity within the attention structure to enhance the performance of the LTSF task.
- Technically, we introduce periodic-nested group attention, named *PENGUIN*, an effective and efficient approach for learning multiple periodic patterns for time series data.
- Extensive experiments on benchmark datasets demonstrate that *PENGUIN* achieves state-of-the-art performance.

2 RELATED WORK

2.1 Transformers for Long-term Time Series Forecasting

Applying Transformer architectures to time series forecasting has emerged as a prominent research topic within the time series community in recent years Zhou et al. (2021); Liu et al. (2022,?). Concurrently, PatchTST (Nie et al., 2022) has established itself as the benchmark Transformer model by patching the time series input in a channel-independent (CI) manner (Kim et al., 2024), a.k.a. CI strategy (Shao et al., 2024). Following this strategy, PathFormer (Chen et al., 2024) introduces a multi-scale Transformer architecture with adaptive pathways, partitioning time series data into multiple temporal resolutions. CATS (Kim et al., 2024) replaces self-attention with a cross-attention-only Transformer approach, highlighting the effectiveness of cross-attention for LTSF. Another strategy, known as the channel-dependent (CD) strategy, focuses on modeling the relationships among variates (Liu et al., 2023). iTransformer (Liu et al., 2023) emerges as a straightforward yet effective implementation of the CD strategy, making no modification to the vanilla Transformer architecture (Vaswani, 2017). More recently, DGCformer (Liu et al., 2024) proposed relatively balanced channel strategies called channel-hard clustering (CHC), and DUET (Qiu et al., 2024) adopted a channel-soft clustering (CSC) strategy and developed a fully adaptive sparsity module to dynamically build groups for each channel.

In this paper, we intend to answer a fundamental question: Can periodic temporal patterns be explicitly modeled within the attention mechanism to enhance the performance for LTSF tasks? To this end, we observe that existing Transformers lack the ability to explicitly model multiple periodicities. Moreover, relative attention bias (RAB), successful in the field of natural language processing (Raffel et al., 2020; Press et al., 2021) and large language models (Sturua et al., 2024), remains under-exploited in time series forecasting. To overcome these limitations, we propose a simple yet effective periodic-nested group attention mechanism for long-term time series forecasting.

2.2 Linear Models for Long-term Time Series Forecasting

While Transformers have been extensively studied for LSTF, their effectiveness remains debatable. Recent works have demonstrated that simple linear models can outperform Transformer-based approaches (Xu et al., 2024; Ni et al., 2024). DLinear (Zeng et al., 2023) proposes a one-layer linear model and demonstrates the effectiveness of simple architectures for LTSF. Both FITS (Xu et al., 2023) and SparseTSF (Lin et al., 2024) adopt lightweight designs, with FITS employing complex-valued neural networks and SparseTSF utilizing sparse forecasting technique. Furthermore, CycleNet (Xu et al., 2024) explicitly models periodic patterns to capture cyclical dependencies in time series data. FreqMoE (Liu, 2025) introduces frequency decomposition to implicitly handle multiple periodicities. These lightweight models, despite the risk of oversimplifying complex temporal patterns, achieve high forecasting accuracy, raising questions about the effectiveness of Transformers for LSTF. In this paper, we address this issue by enhancing the Transformer with Periodic-Nested Group Attention, achieving state-of-the-art performance.

2.3 Language Model v.s. Time Series

Transformers have demonstrated remarkable success in Large Language Models (LLMs) (Yang, 2019). Recently, several works have explored leveraging large

language models for time series forecasting (Liu et al., 2024; Ekambaram et al., 2024; Liu et al., 2024; Cao et al., 2024). UniTime (Liu et al., 2024) proposes a language-empowered unified model for cross-domain time series forecasting. TEMPO (Cao et al., 2024) introduces prompt-based generative pre-trained Transformers for time series. These methods focus on leveraging LLM representations for TSF. However, unlike natural language, time series data exhibits a distinct periodic nature (Xu et al., 2024; Fan et al., 2025; Wu et al., 2021; Wen et al., 2022). To bridge the gap, we extend the ALiBi bias (Press et al., 2021) by incorporating periodicity, enabling the model to capture repetitive seasonal trends[2], making it fundamentally different from the LLM-based approaches.

3 PRELIMINARIES

Long-term time series forecasting (LTSF) involves predicting future values of a time series based on historical observations. Formally, a multi-variate time series $X = [x_{t-L+1}, \cdots, x_t] \in \mathbb{R}^{L\times C}$ represents a sequence of observations over L timesteps, each with C variates (a.k.a channels). The objective is to predict the future sequence $\overline{X} = [x_{t+1}, \cdots, x_{t+H}] \in \mathbb{R}^{H\times C}$, where H denotes the forecasting horizon.

4 PROPOSED METHODOLOGY

We introduce *PENGUIN*, a novel periodic-nested group attention mechanism for accurate long-term time series forecasting. As illustrated in Figure 1, our model first converts time series into channel-independent patch representations via patch embedding. The encoder then leverages our enhanced self-attention with periodic-nested group attention to effectively capture periodic and temporal dependencies, followed by a linear projection to produce predictions. The model is optimized using mean squared error loss.

In the rest of this section, we will detail the proposed attention architecture, while leaving the details about input processing to Appendix A.

4.1 Channel Independent Encoder

The encoder processes the time series in a channel-independent manner. For the c-th channel, the input is first normalized using Revertible Instance Normalization (Nie et al., 2022) and projected into patch embeddings, yielding $x_e^{(c)} \in \mathbb{R}^{N\times d}$, where N is the number of patches. The periodic information $\mathcal{P}_S$ (detailed in the next subsection) is incorporated as input and shared uniformly across all channels. Suppose we have E encoder layers. The overall equations for the l-th layer are summarized as:

$$\begin{aligned} x_e^{(c,l,1)} &= x_e^{(c,l-1)} + \mathrm{Norm}\big(\mathrm{PENGUIN}(x_e^{(c,l-1)}; \mathcal{P}_S)\big) \ , \\ x_e^{(c,l,2)} &= x_e^{(c,l,1)} + \mathrm{Norm}\big(\mathrm{FeedForward}(x_e^{(c,l,1)})\big) \ , \end{aligned} \quad (1)$$

where $x_e^{(c,l)} = x_e^{(c,l,2)}$ and $x_e^{(c,0)} = x_e^{(c)}$. Here, Norm denotes the root mean square layer normalization (Zhang and Sennrich, 2019), i.e.,

$$\mathrm{Norm}(x) = \frac{x}{\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2 + \epsilon}} \cdot \gamma \ , \quad (2)$$

where d is the dimensionality of x, and $\gamma \in \mathbb{R}^d$ denotes learnable weights. FeedForward is an MLP module, i.e.,

$$\mathrm{FeedForward}(X) = \mathrm{ReLU}(XW_f^1 + b_f^1)W_f^2 + b_f^2 \ , \quad (3)$$

where $W_f^1 \in \mathbb{R}^{d\times d_{\mathrm{ff}}}$,$W_f^2 \in \mathbb{R}^{d_{\mathrm{ff}}\times d}$,$b_f^1 \in \mathbb{R}^{d_{\mathrm{ff}}}$ and $b_f^2 \in \mathbb{R}^d$ are learnable parameters, and d_{ff} is the intermediate dimensionality of the feed-forward network. PENGUIN is the proposed periodic-nested group attention.

4.2 Periodic-Nested Group Attention

PENGUIN is an attention mechanism for capturing multiple periodicities within time series data. Specifically, given the input representation $X \in \mathbb{R}^{N\times d}$, we calculate self-attention in a group-query attention manner (Ainslie et al., 2023). Assuming we have h attention heads and g attention groups. Additionally, we assume that g is a factor of h. Therefore, the self-attention module applies linear projections to compute the query, key, and value matrices $Q \in \mathbb{R}^{N\times h\times d_h}$, $K \in \mathbb{R}^{N\times g\times d_h}$, and $V \in \mathbb{R}^{N\times g\times d_h}$, where $d_h = d/h$ denotes the dimensionality per head. Specifically, $Q = XW_Q$, $K = XW_K$, $V = XW_V$, where $W_Q \in \mathbb{R}^{d\times d}$, $W_K, W_V \in \mathbb{R}^{d\times(g\times d_h)}$ are trainable weight matrices. For each group, keys and values are shared, i.e., for each query within the r-th group, the group-specific key and value are denoted as $k^r \in \mathbb{R}^{d_h}$ and $v^r \in \mathbb{R}^{d_h}$ (Shazeer, 2019).

Generally speaking, we denote $n = h/g$ as the number of heads per group. Each attention group calculates a multi-query attention with a group specified relative attention bias $\mathcal{B}_r \in \mathbb{R}^{n\times N\times N}$, $\mathcal{B}_r^{(k)}$ stands for the attention bias for the k-th attention head within the group. Here, the attention bias is an additive term applied to the attention scores before softmax to encode

[2]Although originally proposed to address input length extrapolation in natural language processing (Press et al., 2021), ALiBi stands out for its simplicity, robustness, and strong performance even without extrapolation (i.e., the sequence length within the train dataset is equivalent to that in the test dataset), outperforming sinusoidal embedding (Vaswani, 2017). These advantages motivate our choice to extend ALiBi for time series data.

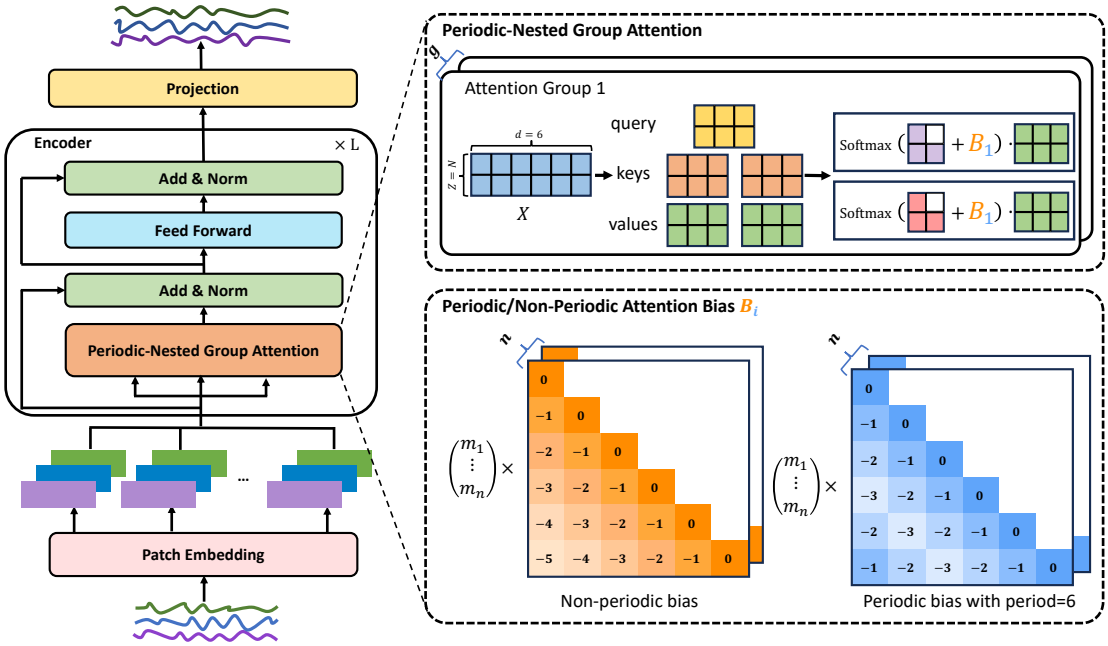

Figure 1: Illustration of *PENGUIN*, a novel Transformer model with periodic-nested group attention tailored for LTSF. We first transform time series into patch embeddings in a channel-independent manner. Next, the encoder module comprises the proposed periodic-nested group attention and a feed-forward network. The right part of the figure illustrates the periodic-nested group attention with periodic/non-periodic attention bias.

structural priors such as temporal proximity and periodic alignment. Finally, for the r-th attention group with n keys and values, the attention mechanism is formulated as:

$$\begin{aligned} \text{head}_\ell &= \text{Softmax}\left(\frac{Q^\ell (k^r)^\top}{\sqrt{d_h}} + \mathcal{B}_r^{(k)}\right) v^r \ , \\ \text{PENGUIN}_r(X) &= \text{Output}([\text{head}_{\tau+1} \| \dots \| \text{head}_{\tau+n}]) \ , \end{aligned} \tag{4}$$

where $\tau = (r-1) * n$, $r \in \{1, \dots, g\}$, ℓ denotes the head index, $k = \ell \bmod n + 1$, $[\cdot\|\cdot]$ indicates the concatenation operation, and $\text{Output}(\cdot)$ denotes the linear projection. The final output of the attention module is obtained by concatenating the outputs from all g groups, i.e., $\text{PENGUIN}(X) = [\text{PENGUIN}_1(X) \| \cdots \| \text{PENGUIN}_g(X)]$.

4.2.1 Attention Bias for the Non-Periodic Case

Time series data may contain instances where periodic information is absent, or the dataset itself lacks inherent periodicity. In such cases, prior methods such as CycleNet (Xu et al., 2024) often struggle to make accurate predictions and tend to degrade into simple linear predictors. Moreover, when time series lack inherent periodicity, frequency decomposition methods, such as Autoformer (Wu et al., 2021) and FEDformer (Zhou et al., 2022), often produce highly inconsistent spectra

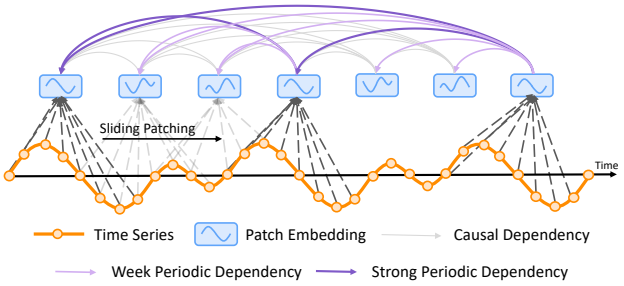

Figure 2: Illustration of periodic attention modeling with an overlapping patching technique. Given a time series with a period of 12, the patching length P and stride S are set to 8 and 4, respectively. Every three patch embeddings share the same sub-region of the original time series, leading to a period after patching of $\mathcal{P}_S = [3]$.

across different windowed segments, undermining the stability and effectiveness of frequency-domain modeling. On the contrary, for the non-periodic case, we can define an effective linear bias dependent on the relative positional distance[3], i.e.,

$$\mathcal{B}_{ij}^{(k)} = -m_k \cdot |i - j| \ , \tag{5}$$

[3]For periodic data, such attention bias can still be beneficial. Therefore, we can also leave one "expert" or one attention group that does not rely on periodic information.

Table 1: Summary of datasets for long-term time series forecasting.

Dataset	ETTh1 & ETTh2	ETTm1 & ETTm2	Electricity	Exchange	Weather	Solar	Traffic
Channels	7	7	321	8	21	137	862
Sampling Frequency	1 hour	15 mins	1 hour	1 Day	10 mins	10 mins	1 hour
Natural Periodicity	Daily	Daily	Daily & Weekly	-	Daily	Daily	Daily & Weekly
Period Lengths $\mathcal{P}$	$\{24\}$	$\{96\}$	$\{24, 168\}$	$\emptyset$	$\{144\}$	$\{144\}$	$\{24, 168\}$

where $i, j \in [1, N]$, $m = [m_1, ..., m_n]$ is a head-specific slope fixed before training, with $m_k = 2^{-\frac{8}{k}}$ and $k \in [1, n]$ One key benefit of the attention bias is that it encourages the model to focus more on local context while still maintaining access to long-range information.

4.2.2 Attention Bias for the Periodic Case

Time series data exhibits periodic characteristics, distinguishing it from natural language data. To bridge the gap, we adopt a periodic-nested attention bias to capture periodic attributes from input embeddings adeptly. In this study, we assume that the periodic information or cycle lengths are known a priori for the dataset. Previous work (Xu et al., 2024) highlights that such periodic information could be obtained through autocorrelation functions (ACF) (Cryer, 1986). Formally speaking, let the dataset be associated with multiple periods denoted as $\mathcal{P} = \{\mathcal{P}_1, ..., \mathcal{P}_p\}$, where p denotes the number of periods. For instance, the Traffic dataset is collected hourly and exhibits both daily and weekly periodicity, corresponding to $\mathcal{P} = \{24, 168\}$.

Since the time series is transformed into a sequence of tokens through patching to reduce computational costs, the periodicity of the patch embeddings becomes dependent on the stride S. As illustrated in Figure 2, the periods after applying sliding patches are defined as $\mathcal{P}_S = \{\frac{\mathcal{P}_1}{S}, ..., \frac{\mathcal{P}_p}{S}\}$. Throughout this paper, we assume S to be a factor of each element in $\mathcal{P}$.

Additionally, each attention group is specified for one particular period length. Let the period length after patching for the r-th attention group be defined as $\mathcal{P}_S^{(r)}$. Therefore, we can define a periodic-nested attention bias as:

$$\mathcal{B}_{ij}^{(k)} = -m_k \cdot \widehat{\mathcal{B}}_{ij}^{(k)} \ ,$$
$$\widehat{\mathcal{B}}_{ij}^{(k)} = \begin{cases} u, & \text{if } u < \frac{\mathcal{P}_S^{(r)}}{2}, \\ \mathcal{P}_S^{(r)} - u, & \text{if } u \geq \frac{\mathcal{P}_S^{(r)}}{2}, \end{cases}$$
$$\text{where } u = |i - j| \text{ Mod } \mathcal{P}_S^{(r)} \ . \tag{6}$$

The modulo operation ensures that the bias reflects periodic variations aligned with the corresponding period value. Additionally, the bias slope vector, i.e., m, is shared across group, and we define $m = [m_1, ..., m_n]$, with $m_k = 2^{-\frac{8}{k}}$ and $k \in [1, n]$.

4.3 Output Projection and Loss Function

We apply a linear projection to obtain predictions $\widetilde{X} = [\widetilde{x}_{t+1}, \ldots, \widetilde{x}_{t+H}] \in \mathbb{R}^{H \times C}$. Following recent LSTF work, we use Mean Squared Error (MSE) loss. Let $\overline{X} = [x_{t+1}, \ldots, x_{t+H}] \in \mathbb{R}^{H \times C}$ be the ground truth, the loss function is calculated as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{H} \sum_{i=1}^{H} \|\widetilde{x}_{t+i} - \overline{x}_{t+i}\|_2^2. \tag{7}$$

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 Datasets

To evaluate the effectiveness of *PENGUIN*, we conducted experiments on nine widely used datasets in the field of long-term time series forecasting. These datasets include ETT (comprising ETTh1, ETTh2, ETTm1, and ETTm2), Electricity, Exchange, Weather, Solar, and Traffic (Wu et al., 2021; Zhou et al., 2021; Xu et al., 2024). A detailed overview of these datasets is provided in Table 1.

5.1.2 Baselines

We selected the current state-of-the-art and representative works in this field as baselines. These include linear-based or MLP-based models (e.g., DLinear (Zeng et al., 2023), SparseTSF (Lin et al., 2024), CycleNet (Xu et al., 2024), MoLE (Ni et al., 2024), MTLinear (Nochumsohn et al., 2025)), Transformer-based models (e.g., Autoformer (Wu et al., 2021), FEDformer (Zhou et al., 2022), PatchTST (Nie et al., 2022), iTransformer (Liu et al., 2023), CATS (Kim et al., 2024)), and CNN-based models (e.g., TimesNet (Wu et al., 2022)).

Table 2: Multivariate long-term forecasting results. The look-back length L is fixed as 336 and the results are averaged over prediction horizons of $H \in \{96, 192, 336, 720\}$. ETT indicates the averaged results of ETTh1, ETTh2, ETTm1, and ETTm2. The best results are in **bold** and the second best are underlined.

Dataset	ETT		Electricity		Exchange		Weather		Solar		Traffic	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Autoformer	0.451	0.461	0.229	0.340	0.732	0.634	0.348	0.391	0.913	0.703	0.679	0.422
FEDformer	0.408	0.439	0.251	0.355	0.769	0.643	0.333	0.382	0.324	0.404	0.622	0.386
DLinear	0.376	0.404	0.166	0.263	0.376	0.444	0.245	0.299	0.253	0.315	0.434	0.295
TimesNet	0.396	0.415	0.249	0.338	0.416	0.443	0.267	0.298	0.236	0.279	0.634	0.354
MoLE	0.380	0.404	0.166	0.264	0.557	0.512	0.244	0.298	0.253	0.314	0.434	0.295
MTLinear	0.377	0.404	0.166	0.263	**0.285**	**0.380**	0.246	0.300	0.252	0.314	0.434	0.295
PatchTST	0.356	0.390	0.179	0.278	0.448	0.443	0.234	0.270	0.203	0.265	0.405	0.282
iTransformer	0.371	0.400	0.163	0.258	0.459	0.457	0.236	0.272	0.230	0.286	0.390	0.275
CATS	0.357	0.392	0.173	0.259	0.425	0.436	**0.227**	**0.266**	0.206	**0.248**	0.411	0.276
SparseTSF	0.356	0.384	0.174	0.266	0.397	0.427	0.250	0.282	0.266	0.281	0.436	0.290
CycleNet	0.351	0.384	**0.160**	**0.253**	0.400	0.422	0.246	0.279	0.223	0.277	0.423	0.288
PENGUIN	**0.344**	**0.379**	0.165	0.261	0.340	0.403	0.228	0.267	**0.200**	0.253	**0.388**	**0.262**

Table 3: Experiment results of different choices of periodic length. The look-back length L is fixed as 336 and the prediction horizons of $H \in \{96, 192, 336, 720\}$. Throughout this experiment, we set the number of attention heads to 12 to ensure that the number of groups g is a factor of h.

Case		No Bias		Non-Periodic		Periodic						Both					
$\mathcal{P}$		$\emptyset$		$\emptyset$		{24}		{168}		{24, 168}		{24}		{168}		{24, 168}	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	96	0.382	0.266	0.364	0.252	0.361	0.250	0.360	0.246	0.358	0.244	0.356	0.244	0.356	0.243	0.357	0.247
	192	0.401	0.270	0.385	0.256	0.378	0.254	0.380	0.259	0.377	0.255	0.378	0.256	0.378	0.254	0.376	0.253
	336	0.411	0.280	0.401	0.274	0.396	0.269	0.399	0.273	0.395	0.266	0.390	0.265	0.393	0.270	0.390	0.262
	720	0.437	0.290	0.434	0.286	0.433	0.289	0.429	0.289	0.433	0.288	0.428	0.285	0.429	0.284	0.428	0.286
	avg.	0.407	0.277	0.396	0.267	0.392	0.265	0.392	0.266	0.390	0.263	0.388	0.262	0.389	0.263	**0.387**	**0.262**
Electricity	96	0.143	0.246	0.144	0.247	0.138	0.236	0.136	0.235	0.137	0.236	0.139	0.237	0.146	0.249	0.138	0.247
	192	0.159	0.260	0.151	0.245	0.149	0.243	0.149	0.244	0.147	0.243	0.150	0.245	0.149	0.244	0.149	0.245
	336	0.176	0.276	0.175	0.274	0.170	0.267	0.171	0.269	0.169	0.266	0.173	0.261	0.172	0.269	0.171	0.268
	720	0.208	0.299	0.209	0.300	0.208	0.299	0.208	0.298	0.207	0.298	0.209	0.300	0.210	0.302	0.206	0.295
	avg.	0.171	0.270	0.169	0.266	0.166	0.261	0.166	0.261	**0.165**	**0.260**	0.167	0.260	0.169	0.266	0.166	0.263

5.2 Main Results

To demonstrate the effectiveness of *PENGUIN* in capturing long-range periodic patterns, we evaluate its performance across various forecast horizons while keeping the input sequence length fixed at 336. Table 2 presents a comparison of *PENGUIN* against other models on multivariate long-term series forecasting (LTSF) tasks. [4] A detailed result can be found in Table 10. Moreover, we provide a comprehensive evaluation using the input sequence length of 96 as shown in Table 11. From the results, we can draw the following four conclusions:

(i) State-of-the-art performance: *PENGUIN* achieves state-of-the-art overall results. Specifically, compared to the state-of-the-art MLP-based model CycleNet (Xu et al., 2024), *PENGUIN* achieves an overall improvement of **5.3%** (0.317 → **0.300**) in terms of MSE. Moreover, when compared to the leading Transformer-based model CATS (Kim et al., 2024), *PENGUIN* delivers an overall MSE improvement of **6.0%** (0.319 → **0.300**).

(ii) Superior improvement over existing decomposition approaches: *PENGUIN* significantly outperforms Autoformer (Wu et al., 2021) and FEDformer (Zhou et al., 2022) by over **45%** in MSE, underscoring the importance of explicitly modeling periodic information to boost performance.

(iii) Consistent improvement over Transformer-based approaches: *PENGUIN* outperforms PatchTST (Nie et al., 2022) and CATS (Kim et al., 2024) across almost all datasets, except for the Electricity dataset,

[4] To ensure fair comparison, we reproduce baseline results using the code and scripts from TimesLib. We set the input length to 336 to align with our experimental setting and avoid the "drop-last" issue that discards samples in the final batch during evaluation.

Table 4: Experiment results of MHA compared to GQA. The look-back length L is fixed as 336 and the results are averaged over prediction horizons of $H \in \{96, 192, 336, 720\}$. Time denotes the average duration of a single training iteration (seconds/iter).

Dataset	Traffic			Electricity		
Metric	Time	MSE	MAE	Time	MSE	MAE
MHA	0.166	0.389	0.263	0.239	0.167	0.264
GQA	**0.145**	**0.387**	**0.262**	**0.224**	**0.165**	**0.260**
Improve	+12.7%	+0.51%	+0.38%	+6.28%	+1.20%	+1.51%

Table 5: Experiment results of incorporating *PENGUIN* with RCF technique. The look-back length L is fixed as 336 and the prediction horizons of $H \in \{96, 192, 336, 720\}$.

Dataset	Traffic		Solar		Electricity	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
CycleNet	0.423	0.288	0.223	0.277	**0.160**	**0.265**
PENGUIN	**0.388**	**0.262**	0.200	**0.253**	0.165	0.261
+ RCF	0.401	0.268	**0.198**	0.254	0.163	0.258
Improve	-3.35%	-2.29%	+1.00%	-0.39%	+1.21%	+1.15%

where CATS achieves better performance. We hypothesize that this is due to the Electricity dataset being more affected by non-stationarity (Shao et al., 2024; Liu et al., 2024), which *PENGUIN* is not specifically designed to handle. In conclusion, these results highlight the robustness of *PENGUIN* and its superior capacity compared to existing Transformer-based approaches.

(iv) Effective periodic modeling technique: Both *PENGUIN* and CycleNet (Xu et al., 2024) aim to capture periodic patterns in time series data. The superior performance of *PENGUIN* over CycleNet demonstrates its effectiveness and offers novel insights into modeling periodic patterns for time series data.

5.3 Ablation Study

5.3.1 Ablation Study of Attention Bias

In this paper, we propose a simple yet effective attention mechanism, termed *PENGUIN*, specifically designed for LTSF. To validate its effectiveness, we conduct a comprehensive ablation study on the Traffic and Electricity datasets, as shown in Table 3. Notably, both datasets exhibit multiple periodic patterns (Xu et al., 2024), including daily and weekly cycles. We conduct four groups of experiments: (i) No Bias, where we adopt multi-query attention with a single attention group (Shazeer, 2019); (ii) Non-Periodic, where we also adopt multi-query attention with a single attention group, but with a non-periodic attention bias; (iii) Periodic, where we adopt a group-query attention with various predefined periodic information; (iv) Both, where we combine both types of biases by assigning one attention group to non-periodic bias and the remaining groups to periodic biases. From this table, we can draw the following four conclusions:

(i) The attention mechanism without any bias yields the poorest performance, underscoring the importance of incorporating attention bias in LTSF. For instance, a non-periodic bias improves the performance by 2.77% (0.407 → 0.396) on the Traffic dataset.

(ii) Results from the periodic groups consistently outperform their non-periodic counterparts, highlighting the critical role of periodic information in long-term forecasting. For instance, a multiple-periodic bias further improves the performance by 2.32% (0.396 → 0.387) on the Traffic dataset.

(iii) When multiple periodicities are present in the dataset, leveraging these patterns leads to an overall 0.5% performance improvement on both datasets.

(iv) Periodic attention bias captures recurring patterns, while non-periodic bias focuses on preserving temporal dependencies. Our empirical study shows that combining both types of attention bias yields improved performance on the Traffic dataset.

5.3.2 Ablative Study of Attention Structure

To improve *PENGUIN*'s efficiency, we replace standard multi-head attention (MHA) with group-query attention (GQA), where queries remain head-specific but keys and values are shared within each attention group. As reported in Table 4, GQA lowers computational overhead, cutting time cost by 13% on the Traffic dataset, and consistently yields performance gains across multiple benchmarks. These results highlight GQA as an effective trade-off between efficiency and accuracy in long-term time series forecasting.

5.3.3 Ablation of Periodic Patterns Modeling

Both *PENGUIN* and the Residual Cycle Forecasting (RCF) technique (Xu et al., 2024) are designed to capture periodic patterns in time series data. While *PENGUIN* focuses on modeling complex and implicit periodic dependencies, RCF decomposes the time series into a learnable recurrent cycle component and a residual component, where the cycle component explicitly captures periodic patterns by learning a fixed-length repeating template. To assess their complementary effectiveness, we integrate RCF into *PENGUIN* and report the results in Table 5, comparing the original *PENGUIN*, the combined *PENGUIN* +RCF (Xu et al., 2024), and CycleNet. The results show only

Table 6: The effectiveness of *PENGUIN* for Decoder architecture. The look-back length L is fixed as 336 and the results are averaged over prediction horizons of $H \in \{96, 192, 336, 720\}$.

Dataset	ETTh1		ETTm1		Weather	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
Decoder	0.478	0.465	0.448	0.439	0.228	0.261
+ *PENGUIN*	**0.430**	**0.440**	**0.361**	**0.392**	**0.226**	**0.258**
Improve	+10.00%	+5.43%	+19.36%	+10.60%	+1.23%	+1.04%
EncDec	0.495	0.473	0.520	0.469	0.236	0.267
+ *PENGUIN*	**0.419**	**0.436**	**0.354**	**0.388**	**0.231**	**0.265**
Improve	+15.42%	+7.88%	+31.96%	+17.33%	+2.04%	+0.68%

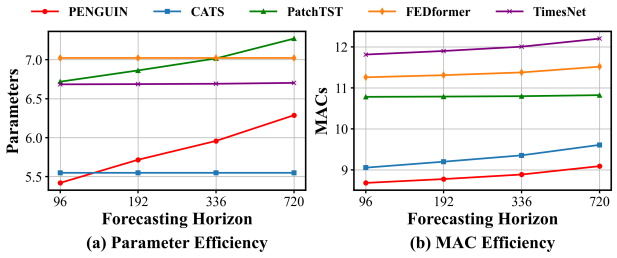

Figure 3: The efficiency analysis of five models (i.e., PatchTST (Nie et al., 2022), CATS (Kim et al., 2024), TimesNet (Wu et al., 2022), FEDformer (Zhou et al., 2022), and *PENGUIN*) on ETTm1 dataset. The experiments are conducted with a look-back length L of 336. Both the number of parameters and MACs are plotted on a logarithmic scale.

marginal performance gains from incorporating RCF with *PENGUIN*. This suggests that the Transformer-based *PENGUIN* is sufficiently capable of modeling periodic dependencies without additional reliance on the RCF components.

5.4 Extendability Study

The Transformer consists of encoder and decoder modules (Vaswani, 2017). We extend *PENGUIN* to the decoder, with a key difference in the non-periodic attention bias: $\mathcal{B}^{(k)}_{ij} = m_k \cdot |(i+N)-j|$, replacing Eq. (5). Here, N is the length of keys and values in the encoder architecture. [5] A similar modification is applied to Eq. (9) for the periodic case. The experimental results, presented in Table 6, reveal that *PENGUIN* delivers significant improvements for both the decoder structure and the encoder-decoder (EncDec) models. These findings suggest that *PENGUIN* has strong potential as a plugin for Transformer-based models in time series modeling.

[5] In cross-attention forecasting, keys and values come from the input series of length N, so M decoding tokens align with positions $N+1, \ldots, N+M$ (Kim et al., 2024).

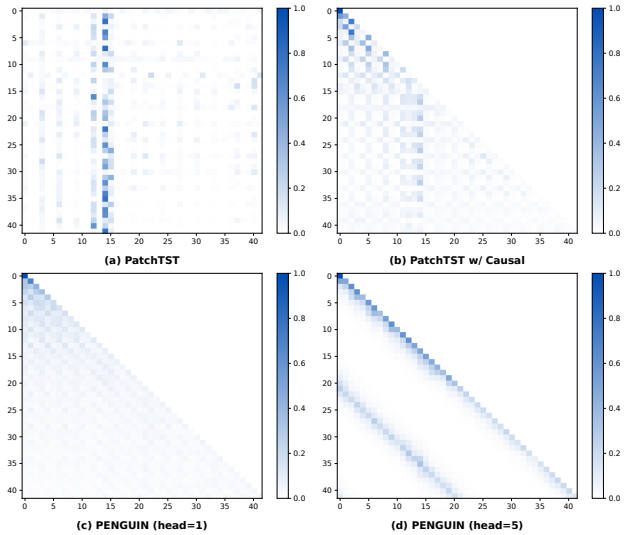

Figure 4: Visualization of *PENGUIN* and PatchTST on the Traffic dataset, using $\mathcal{P} = \{24, 168\}$ with a patch length $P = 16$ and stride $S = 8$. For PatchTST, we train two variants with different masking strategies (i.e., full attention and attention with a causal mask). For *PENGUIN*, we visualize two attention heads, each belonging to a different attention group.

5.5 Efficiency Study

To assess the efficiency of *PENGUIN*, we consider two key metrics: the number of parameters and Multiply-Accumulate Operations (MACs). The parameter count measures the total number of learnable weights, while MACs quantify the computational cost of inference. The results, evaluated on the ETTm1 dataset with a fixed batch size of 32 across all models, are presented in Figure 3. For comparison, we include four representative baselines designed to model temporal dependencies in time series data. Notably, *PENGUIN* achieves the highest prediction accuracy, outperforming the second-best baseline, CATS, by 1.5%, while simultaneously requiring the lowest computational cost among all evaluated approaches. In particular, *PENGUIN* demonstrates a substantial advantage over TimesNet in terms of MAC efficiency, underscoring the inefficiency of decomposing time series into multiple temporal sequences.

5.6 Visualization Study

We compare the attention matrices of *PENGUIN* and PatchTST (Nie et al., 2022), as shown in Figure 4. In the figure, the i-th row denotes the attention distribution over all tokens conditioned on the i-th query token. As illustrated in Figures 4(c)–(d), *PENGUIN* effectively captures periodic temporal correlations, with

Table 7: Experiment results of missing or incorrect periodic information. The look-back length L is fixed as 336 and the prediction horizons of $H \in \{96, 192, 336, 720\}$.

Case		Existing Approaches				*PENGUIN*											
Model / $\mathcal{P}$		SparseTSF		CATS		$\emptyset$		{24}		{72}		{96}		{144}		{168}	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	96	0.415	0.281	0.382	0.260	0.364	0.252	0.361	0.250	0.366	0.254	0.363	0.250	0.361	0.251	0.360	0.246
	192	0.426	0.283	0.401	0.268	0.385	0.256	0.378	0.254	0.383	0.258	0.383	0.262	0.382	0.261	0.380	0.259
	336	0.438	0.289	0.414	0.275	0.401	0.274	0.396	0.269	0.400	0.274	0.402	0.274	0.401	0.276	0.399	0.273
	720	0.465	0.306	0.447	0.300	0.434	0.286	0.433	0.289	0.433	0.292	0.433	0.290	0.432	0.291	0.429	0.289
	avg.	0.436	0.290	0.411	0.276	0.396	0.267	**0.392**	**0.266**	0.396	0.270	0.395	0.269	0.394	0.270	0.392	0.267
ETTm1	96	0.306	0.347	0.297	0.352	0.290	0.341	0.288	0.341	0.290	0.341	0.290	0.341	0.290	0.344	0.290	0.342
	192	0.339	0.366	0.323	0.369	0.330	0.370	0.328	0.365	0.329	0.367	0.319	0.361	0.329	0.368	0.328	0.366
	336	0.374	0.386	0.357	0.389	0.361	0.390	0.359	0.389	0.363	0.391	0.355	0.387	0.362	0.391	0.360	0.390
	720	0.428	0.416	0.413	0.421	0.414	0.420	0.411	0.423	0.415	0.427	0.409	0.419	0.414	0.421	0.413	0.420
	avg.	0.362	0.379	0.348	0.383	0.349	0.380	0.347	0.380	0.349	0.382	**0.343**	**0.377**	0.349	0.381	0.348	0.380

different attention heads or groups learning distinct periodic patterns. In contrast, PatchTST produces less interpretable attention maps, while its causal-attention variant reveals some structural patterns but still falls short of *PENGUIN*. We hypothesize that the causal attention mechanism is essential for preserving temporal causality and modeling the auto-regressive generative nature of time series data. Supporting this claim, our empirical study shows that removing the causal mask from *PENGUIN* leads to performance drops of 3.5% and 4.8% on ETTh1 and ETTh2, respectively. Together, these results underscore the importance of explicitly modeling temporal dependencies and highlight the effectiveness of *PENGUIN* in capturing both causal and periodic structures for accurate time series forecasting.

5.7 Robustness Towards Missing and Incorrect Periodic Information

In real-world time series forecasting tasks, it is common for periodic information to be either missing or incorrectly specified due to noisy data or inaccurate prior knowledge. To evaluate *PENGUIN*'s robustness in such scenarios, we conducted experiments by intentionally introducing missing or incorrect periodic information into the datasets (Xu et al., 2024). Specifically, we select various periodic lengths from [24,72,96,144,168]. The results, shown in Table 7, compare *PENGUIN* with representative approaches, i.e., SparseTSF and CATS. We can draw the following two conclusions:

(i) *PENGUIN* maintains comparable, and sometimes state-of-the-art, performance against strong baselines even when periodic information is missing, highlighting its ability to capture temporal dependencies without relying on precise periodic cues.

(ii) When periodic information is incorrect, performance degrades noticeably as expected (e.g., 1.7% on the ETTm1 dataset), since incorrect patterns hinder model predictions. However, even with incorrect periodicity, *PENGUIN* outperforms many existing models (e.g., SparseTSF), demonstrating its ability to extract useful temporal features despite periodic discrepancies.

6 CONCLUSION

Long-term time series forecasting (LSTF) is a fundamental task with broad real-world applications. In this paper, we revisit the effectiveness of self-attention mechanisms and demonstrate that incorporating a simple attention bias can significantly enhance their capability for LSTF. To this end, we propose a novel periodic-nested group attention mechanism, *PENGUIN*, which combines a periodic linear bias with a grouped query attention structure. This design enables the model to capture diverse periodic patterns while maintaining temporal causality. Extensive experiments on nine benchmark datasets show that *PENGUIN* consistently achieves state-of-the-art performance across a wide range of LSTF tasks.

References

Ainslie, J., J. Lee-Thorp, M. De Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai (2023). Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.

Bai, J., S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, et al. (2023). Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sas-

try, A. Askell, et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems 33*, 1877–1901.

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

Chen, C., L. Zhao, J. Bian, C. Xing, and T.-Y. Liu (2019). Investment behaviors can tell what inside: Exploring stock intrinsic properties for stock trend prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2376–2384.

Chen, P., Y. Zhang, Y. Cheng, Y. Shu, Y. Wang, Q. Wen, B. Yang, and C. Guo (2024). Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. *arXiv preprint arXiv:2402.05956*.

Chen, Y., K. Ren, K. Song, Y. Wang, Y. Wang, D. Li, and L. Qiu (2024). Eegformer: Towards transferable and interpretable large-scale eeg foundation model. *arXiv preprint arXiv:2401.10278*.

Chen, Y., H. Zhang, W. Sun, and B. Zheng (2023). Rntrajrec: Road network enhanced trajectory recovery with spatial-temporal transformer. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 829–842. IEEE.

Cryer, J. D. (1986). *Time series analysis*, Volume 286. Duxbury Press Boston.

Ekambaram, V., A. Jati, N. Nguyen, P. Dayama, C. Reddy, W. Gifford, and J. Kalagnanam (2024). Tiny time mixers (TTMs): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Elfeky, M. G., W. G. Aref, and A. K. Elmagarmid (2005). Periodicity detection in time series databases. *IEEE Transactions on Knowledge and Data Engineering 17*(7), 875–887.

Fan, J., W. Weng, Q. Chen, H. Wu, and J. Wu (2025). Pdg2seq: Periodic dynamic graph to sequence model for traffic flow prediction. *Neural Networks 183*, 106941.

Kim, D., J. Park, J. Lee, and H. Kim (2024). Are self-attentions effective for time series forecasting? *arXiv preprint arXiv:2405.16877*.

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

Li, L., J. Yan, Y. Zhang, J. Zhang, J. Bao, Y. Jin, and X. Yang (2022). Learning generative rnn-ode for collaborative time-series and event sequence forecasting. *IEEE Transactions on Knowledge and Data Engineering 35*(7), 7118–7137.

Lin, S., W. Lin, W. Wu, H. Chen, and J. Yang (2024). Sparsetsf: Modeling long-term time series forecasting with 1k parameters. *arXiv preprint arXiv:2405.00946*.

Liu, P., B. Wu, Y. Hu, N. Li, T. Dai, J. Bao, and S.-t. Xia (2024). Timebridge: Non-stationarity matters for long-term time series forecasting. *arXiv preprint arXiv:2410.04442*.

Liu, Q., Y. Fang, P. Jiang, and G. Li (2024). Dgcformer: Deep graph clustering transformer for multivariate time series forecasting. *arXiv preprint arXiv:2405.08440*.

Liu, S., H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar (2022). Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Liu, X., J. Hu, Y. Li, S. Diao, Y. Liang, B. Hooi, and R. Zimmermann (2024). Unitime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM Web Conference (WWW)*.

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

Liu, Y., G. Qin, X. Huang, J. Wang, and M. Long (2024). Autotimes: Autoregressive time series forecasters via large language models. *Advances in Neural Information Processing Systems 37*, 122154–122184.

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

Liu, Z. (2025). FreqMoE: Enhancing time series forecasting through frequency decomposition mixture of experts. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Ni, R., Z. Lin, S. Wang, and G. Fanti (2024). Mixture-of-linear-experts for long-term time series forecasting. In *International Conference on Artificial Intelligence and Statistics*, pp. 4672–4680. PMLR.

Nie, Y., N. H. Nguyen, P. Sinthong, and J. Kalagnanam (2022). A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

Nochumsohn, L., H. Zisling, and O. Azencot (2025). A multi-task learning approach to linear multivariate forecasting. *arXiv preprint arXiv:2502.03571*.

Press, O., N. A. Smith, and M. Lewis (2021). Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Qin, J., Y. Jia, Y. Tong, H. Chai, Y. Ding, X. Wang, B. Fang, and Q. Liao (2024). Muse-net: Disentangling multi-periodicity for traffic flow forecasting. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 1282–1295. IEEE.

Qiu, X., X. Wu, Y. Lin, C. Guo, J. Hu, and B. Yang (2024). Duet: Dual clustering enhanced multivariate time series forecasting. *arXiv preprint arXiv:2412.10859*.

Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research 21*(140), 1–67.

Shao, Z., F. Wang, Y. Xu, W. Wei, C. Yu, Z. Zhang, D. Yao, T. Sun, G. Jin, X. Cao, et al. (2024). Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Transactions on Knowledge and Data Engineering*.

Shazeer, N. (2019). Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*.

Sturua, S., I. Mohr, M. K. Akram, M. Günther, B. Wang, M. Krimmel, F. Wang, G. Mastrapas, A. Koukounas, N. Wang, et al. (2024). jina-embeddings-v3: Multilingual embeddings with task lora. *arXiv preprint arXiv:2409.10173*.

Sun, T., Y. Chen, B. Zheng, and W. Sun (2025). Learning spatio-temporal dynamics for trajectory recovery via time-aware transformer. *IEEE Transactions on Intelligent Transportation Systems*.

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

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

Wang, S., J. Li, X. Shi, Z. Ye, B. Mo, W. Lin, S. Ju, Z. Chu, and M. Jin (2024). Timemixer++: A general time series pattern machine for universal predictive analysis. *arXiv preprint arXiv:2410.16032*.

Wang, S., H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. Zhou (2024). Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*.

Wen, Q., T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun (2022). Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*.

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

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

Xu, S., Z. Ma, Y. Huang, H. Lee, and J. Chai (2024). Cyclenet: Rethinking cycle consistency in text-guided diffusion for image manipulation. *Advances in Neural Information Processing Systems 36*.

Xu, Z., A. Zeng, and Q. Xu (2023). Fits: Modeling time series with $10k$ parameters. *arXiv preprint arXiv:2307.03756*.

Yang, Z. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Yi, K., Y. Wang, K. Ren, and D. Li (2024). Learning topology-agnostic eeg representations with geometry-aware modeling. *Advances in Neural Information Processing Systems 36*.

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

Zhang, B. and R. Sennrich (2019). Root mean square layer normalization. *Advances in Neural Information Processing Systems 32*.

Zhang, H., H. Wu, W. Sun, and B. Zheng (2018). Deeptravel: a neural network based travel time estimation model with auxiliary supervision. *arXiv preprint arXiv:1802.02147*.

Zhang, J., J. Wang, X. Shen, and W. Qiang (2025). Enhancing time series forecasting via logic-inspired regularization. *arXiv preprint arXiv:2503.06867*.

Zhang, L., C. Aggarwal, and G.-J. Qi (2017). Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 2141–2149.

Zhang, Y., M. Liu, S. Zhou, and J. Yan (2024). Up2me: Univariate pre-training to multivariate fine-tuning as a general-purpose framework for multivariate time series analysis. In *Forty-first International Conference on Machine Learning*.

Zhang, Y. and J. Yan (2023). Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*.

Zheng, C., X. Fan, C. Wang, and J. Qi (2020). Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Volume 34, pp. 1234–1241.

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

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

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Not Applicable]
   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Enhancing Transformer with Periodic-Nested Group Attention for Long-term Time Series Forecasting - Supplementary Materials

A Details about Input Processing

A.1 Revertible Instance Normalization

Time series data often encounter challenges related to distributional shifts in terms of statistical properties such as mean and variance over time (Nie et al., 2022; Liu et al., 2023). To address the challenge, a standard normalization strategy, namely reversible instance normalization (Revin) (Kim et al., 2021), demonstrates a great enhancement for LTSF. Mathematically, given the input $X \in \mathbb{R}^{L \times C}$, Revin normalizes the data by removing its mean and variance:

$$x^{(c)} = \frac{x^{(c)} - \mu^{(c)}}{\sqrt{\sigma^{(c)} + \epsilon}}, c \in \{1, \ldots, C\} \ , \tag{8}$$

where $\mu^{(c)}$ and $\sigma^{(c)}$ are the mean and standard deviation across the c-th channel input sequence, and ϵ is a small constant added for numerical stability. After the model processes the data, the original mean and variance are restored, mitigating the impact of non-stationary components. This approach improves the model's robustness and accuracy.

A.2 Input Patching

Long-term time series possess extensive look-back windows, necessitating models that can adeptly capture both local and long-term information. This enhancement through patching has been substantiated in numerous studies (Nie et al., 2022). This approach involves dividing each univariate input time series $x^{(c)} \in \mathbb{R}^L$, where c represents the c-th channel, into patches that serve as either overlapping or non-overlapping input tokens. Formally, given an input sequence of length L, the input time series sequence is divided into non-overlapping subsequences of length P with a stride of S, resulting in $N = \lfloor \frac{(L-P)}{S} \rfloor + 2$ patches as $x_p^{(c)} \in \mathbb{R}^{N \times P}$. Subsequently, we embed each patch into a higher-dimensional space, $\tilde{x}_p^{(c)} = x_p^{(c)} W_p + b_p$, where $W_p \in \mathbb{R}^{P \times d}$ and $b_p \in \mathbb{R}^d$ denote the learnable embedding matrix and bias vector, d represents the size of hidden dimension. To incorporate the sequential order of the patches, a position embedding PE is added to enhance the model's ability to understand the absolute position of each patch within the sequence. This process is mathematically represented as $x_e^{(c)} = \tilde{x}_p^{(c)} + PE$, where $x_e^{(c)} \in \mathbb{R}^{N \times d}$ represents the final input to the Transformer encoder and PE is the position embedding vector that encodes the position information.

B Details about applying *PENGUIN* for Decoder Structure

In the vanilla Transformer, the decoder network contains a self-attention sub-module and a cross-attention sub-module. To incorporate *PENGUIN* for the decoder structure, we extend the proposed attention mechanism for the cross-attention sub-module. Furthermore, following the cross-attention architecture of CATS (Kim et al., 2024), the input to the decoder model consists of learnable queries. Since the time series is transformed to a patch embedding corresponding to the patch stride of S, the number of learning queries is $M = \frac{H}{S}$ and the input for the c-th channel is $x_d^{(c)} \in \mathbb{R}^{M \times d}$. After processing through the decoder model, the output $\tilde{x}_d^{(c)} \in \mathbb{R}^{M \times d}$ is transformed to the output via linear transformation, i.e., $\widetilde{x}_d^{(c)} = \tilde{x}_d^{(c)} W_d^T + b_d$, where $W_d \in \mathbb{R}^{d \times S}$ and $b_d \in \mathbb{R}^S$. Afterward, the transformed output $\widetilde{x}_d^{(c)} \in \mathbb{R}^{M \times S}$ is then concatenated to reconstruct the final sequence $\widetilde{x}^{(c)} \in \mathbb{R}^H$.

Attention Bias for the Periodic Case for Cross Attention Unlike the encoder, the number of tokens for queries and keys can be different. Specifically, given queries $Q \in \mathbb{R}^{M \times d}$ and keys $K \in \mathbb{R}^{N \times d}$, the periodic bias in Equation (9) should be modified as follows:

$$\mathcal{B}_{ij}^{(k)} = -m_k \cdot \widehat{\mathcal{B}}_{ij}^{(k)} \ ,$$
$$\widehat{\mathcal{B}}_{ij}^{(k)} = \begin{cases} u, & \text{if } u < \frac{\mathcal{P}_S^{(r)}}{2}, \\ \mathcal{P}_S^{(r)} - u, & \text{if } u \geq \frac{\mathcal{P}_S^{(r)}}{2}, \end{cases}$$
$$\text{where } u = |(i+N) - j| \text{ Mod } \mathcal{P}_S^{(r)} \ . \tag{9}$$

Since there are N tokens corresponding to the keys (i.e., the past history), the first decoding token — that is, the first query — should be assigned an index of $N+1$, the second $N+2$, and so on. Consequently, the positional bias between the i-th query and the j-th key should be defined as $|(i+N)-j|$ instead of $|i-j|$, reflecting the fact that queries represent future tokens while keys represent past tokens.

Furthermore, the decoder's objective is to predict future horizons, with each learnable query token responsible for forecasting S future values. As a result, causal masking is no longer necessary in the cross-attention mechanism, since future tokens are allowed unrestricted access to information from past tokens.

C Details about Experimental Setting

C.1 Dataset Split and Hyper-parameter Settings

Following the settings of Nie et al. (2022) and Xu et al. (2024), we split the ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2) into training, validation, and testing sets using a 6:2:2 ratio, while the other datasets are divided using a 7:1:2 ratio. For our model, we adopt fixed positional embeddings for the input data (Vaswani, 2017). To address GPU memory limitations during training, we adapt the batch size according to the dataset characteristics: larger batch sizes are used for datasets with fewer feature channels, while smaller batch sizes are applied to higher-dimensional datasets to ensure stable optimization (Lin et al., 2024). A comprehensive overview of the hyperparameter configurations and training setups is provided in Table 8.

Dataset	Batch Size	Hidden Dimension	Patch Length	Patch Stride	Epochs	Patience
ETTh1	64	16	1	1	30	6
ETTh2	32	16	12	2	30	6
ETTm1	64	16	4	2	30	6
ETTm2	32	16	2	2	30	6
Electricity	12	128	12	2	30	6
Weather	8	128	4	1	30	6
Solar	4	512	16	1	30	10
Traffic	4	512	16	8	10	6

Table 8: Experimental setting of various time series forecasting datasets.

C.2 Hyper-Parameter Search Space

We observe that patch stride, patch length, and hidden dimensions influence the performance. Therefore, we conduct the following search space as shown in Table 9.

Table 9: Hyperparameter search space of *PENGUIN*.

Hyperparameter	Values
Patch Stride	[1, 2, 4, 8]
Patch Length	[1, 2, 4, 8, 12, 16]
Hidden Dimensions	[16, 64, 128, 512]

To avoid the OOM issue during hyperparameter search, we set the batch size according to the GPU limitation when using the largest hyperparameter.

C.3 Compared Baselines

We selected the current state-of-the-art and representative works in this field as baselines. These include linear-based or MLP-based models, including:

- DLinear: A combination of a decomposition scheme (Wu et al., 2021; Zhou et al., 2022) with linear layers.
- SparseTSF (Lin et al., 2024): A linear approach that simplifies the forecasting task by decoupling the periodicity and trend in time series data.
- CycleNet (Xu et al., 2024): The state-of-the-art MLP-based approach for LTSF, modeling time series through residual cycle forecasting (RCF) to capture inherent periodic patterns[6].
- MoLE (Ni et al., 2024): A Mixture-of-Linear-Experts approach that employs channel-wise MoE with a linear-based model for TSF.
- MTLinear (Nochumsohn et al., 2025): A multi-task learning approach for TSF where similar variates are grouped together, and each group forms a separate task.

Additionally, we also compare the Transformer-based models, including:

- Autoformer (Wu et al., 2021): A novel decomposition architecture incorporating an Auto-Correlation mechanism, which empowers Autoformer with progressive decomposition capabilities for modeling complex time series.
- FEDformer (Zhou et al., 2022): An efficient Transformer design that segments time series into patches and applies shared embeddings and weights across univariate series.
- PatchTST (Nie et al., 2022): An approach that first divides time series into sub-level patches, followed by a Transformer network.
- iTransformer (Liu et al., 2023) (short for iTrans.): An approach that applies the attention and feed-forward network on the inverted dimensions to capture multivariate correlations.
- CATS (Kim et al., 2024): The state-of-the-art Transformer-based approach for LTSF that eliminates self-attention, leveraging cross-attention mechanisms instead.

Furthermore, we compare *PENGUIN* with a convolution-based approach, TimesNet (Wu et al., 2022), an approach that transforms time series data into 2D tensors and captures multi-periodic patterns through an inception block.

C.4 Metrics and Environments

Consistent with prevalent methodologies (Nie et al., 2022; Xu et al., 2024; Zeng et al., 2023), we elect the traditional Mean Squared Error (MSE) and Mean Absolute Error (MAE) as the core metrics for evaluation.

C.5 Environments

All experiments are conducted on a single NVIDIA RTX 3090 GPU with 24GB of memory.

D Additional Experimental Results

D.1 Main Results

To demonstrate the effectiveness of *PENGUIN* in capturing long-range periodic patterns, we evaluate its performance across various forecast horizons while keeping the input sequence length fixed at 336. Table 10 presents

[6]For datasets without periodicity (e.g., Exchange), we remove the RCF technique from CycleNet, resulting in a model with a single linear layer and reversible instance normalization.

a comparison of *PENGUIN* against other models on multivariate long-term series forecasting (LTSF) tasks. As reported in the table, *PENGUIN* achieves state-of-the-art results, attaining the highest top-1 count and the lowest average error across benchmarks. Moreover, table 11 summarizes the experimental results using the input sequence length fixed at 96, a standard experimental setting for most research on long-term time series forecasting. Notably, the results of CycleNet Xu et al. (2024), iTransformer (Liu et al., 2023), and CATS (Kim et al., 2024) are directly copied from the original paper. The results of PatchTST (Nie et al., 2022) are copied from the iTransformer (Liu et al., 2023) paper to avoid the drop-last issue of time series evaluation. The results show that PENGUIN achieves state-of-the-art performance as compared to recent strong baselines.

Table 10: Multivariate long-term forecasting results using a fixed random seed across benchmarks and compared models. The look-back length L is fixed as 336 and the results are averaged over prediction horizons of $H \in \{96, 192, 336, 720\}$. The best results are in **bold** and the second best are underlined.

Model		ETTh1		ETTh2		ETTm1		ETTm2		Electricity		Exchange		Weather		Solar		Traffic		**Avg.**	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PENGUIN	96	0.375	0.397	0.283	**0.343**	**0.290**	**0.341**	0.168	0.258	0.137	0.236	0.093	0.218	**0.150**	0.202	**0.180**	0.244	**0.357**	**0.247**		
	192	0.420	0.426	0.351	**0.387**	**0.319**	**0.361**	0.227	**0.296**	0.147	0.243	0.190	0.316	0.194	**0.243**	**0.200**	0.256	**0.376**	**0.253**	**0.300**	**0.330**
	336	0.436	0.439	0.376	**0.409**	**0.355**	0.387	0.282	**0.333**	0.169	0.266	0.313	**0.408**	0.245	0.284	**0.205**	**0.254**	**0.390**	**0.262**		
	720	**0.448**	**0.461**	**0.403**	0.438	**0.409**	0.419	**0.374**	**0.390**	0.207	0.298	0.763	0.668	0.323	0.339	0.214	0.259	0.428	**0.286**		
CATS	96	0.381	0.399	0.294	0.350	0.297	0.352	0.197	0.289	**0.130**	**0.218**	0.101	0.222	0.151	0.203	0.191	**0.237**	0.382	0.260		
	192	0.421	0.432	0.356	0.391	0.323	0.369	0.252	0.325	0.147	**0.236**	**0.182**	**0.305**	**0.193**	0.243	0.203	**0.246**	0.401	0.268	0.319	0.339
	336	**0.432**	0.444	0.387	0.416	0.357	0.389	0.304	0.357	0.165	**0.253**	0.342	0.427	**0.244**	**0.282**	0.214	0.256	0.414	0.275		
	720	0.495	0.486	0.415	0.443	0.413	0.421	0.395	0.411	0.251	0.327	1.074	0.790	**0.319**	**0.335**	0.214	**0.251**	0.447	0.300		
CycleNet	96	0.382	0.403	**0.281**	0.345	0.301	0.344	0.172	0.263	0.130	0.226	0.091	0.211	0.174	0.225	0.203	0.287	0.399	0.276		
	192	0.414	0.422	**0.349**	0.392	0.335	0.364	**0.226**	0.299	**0.146**	0.241	0.192	0.310	0.216	0.259	0.224	0.297	0.413	0.282		
	336	0.438	**0.437**	0.373	0.415	0.369	**0.384**	**0.280**	0.335	**0.162**	0.257	0.385	0.449	0.263	0.293	0.227	0.261	0.426	0.288	0.317	0.339
	720	0.455	0.468	0.428	0.453	0.424	**0.414**	0.381	0.395	0.200	**0.289**	0.930	0.719	0.330	0.339	0.238	0.264	0.453	0.305		
SparseTSF	96	0.425	0.424	0.290	0.345	0.306	0.347	0.174	0.264	0.149	0.243	0.101	0.226	0.177	0.227	0.234	0.264	0.415	0.281		
	192	0.438	0.431	0.352	0.387	0.339	0.366	0.231	0.302	0.160	0.253	0.189	0.313	0.220	0.262	0.263	0.279	0.426	0.283		
	336	0.441	0.438	**0.372**	0.413	0.374	0.386	0.280	0.333	0.175	0.268	0.334	0.420	0.267	0.297	0.284	0.291	0.438	0.289	0.327	0.343
	720	0.462	0.474	0.404	**0.434**	0.428	0.416	0.375	0.390	0.213	0.300	0.963	0.749	0.335	0.343	0.283	0.290	0.465	0.306		
iTrans.	96	0.404	0.420	0.301	0.358	0.302	0.357	0.173	0.266	0.133	0.228	0.103	0.230	0.159	0.209	0.201	0.256	0.365	0.261		
	192	0.451	0.450	0.376	0.405	0.350	0.385	0.247	0.314	0.156	0.251	0.212	0.334	0.203	0.251	0.223	0.278	0.378	0.269		
	336	0.476	0.467	0.398	0.423	0.380	0.402	0.297	0.345	0.167	0.265	0.369	0.448	0.254	0.289	0.243	0.298	0.392	0.277	0.329	0.350
	720	0.538	0.523	0.416	0.444	0.446	0.442	0.379	0.398	**0.195**	0.289	1.153	0.815	0.327	0.340	0.254	0.312	**0.424**	0.293		
TimesNet	96	0.384	0.402	0.393	0.418	0.342	0.387	0.198	0.280	0.209	0.312	0.107	0.234	0.179	0.230	0.223	0.268	0.607	0.344		
	192	0.436	0.429	0.416	0.441	0.374	0.407	0.297	0.340	0.242	0.331	0.226	0.344	0.241	0.283	0.242	0.294	0.625	0.349		
	336	0.491	0.469	0.403	0.439	0.417	0.428	0.327	0.365	0.257	0.343	0.367	0.448	0.297	0.322	0.236	0.281	0.641	0.356	0.376	0.375
	720	0.521	0.500	0.460	0.475	0.469	0.460	0.408	0.403	0.288	0.367	0.964	0.746	0.350	0.356	0.242	0.273	0.663	0.367		
PatchTST	96	0.380	0.406	0.288	0.346	0.290	0.345	0.174	0.265	0.142	0.245	0.100	0.227	0.151	**0.199**	0.193	0.253	0.377	0.270		
	192	0.422	0.435	0.358	0.390	0.340	0.379	0.229	0.301	0.178	0.280	0.185	0.313	0.200	0.249	0.200	0.264	0.394	0.275		
	336	0.473	0.464	0.381	0.412	0.374	0.400	0.288	0.337	0.186	0.288	0.392	0.451	0.252	0.287	0.209	0.274	0.407	0.283	0.321	0.344
	720	0.483	0.489	0.405	0.437	0.435	0.439	0.378	0.391	0.208	0.299	1.113	0.780	0.334	0.344	**0.210**	0.269	0.442	0.299		
MTLinear	96	**0.371**	**0.392**	0.309	0.368	0.301	0.345	0.170	0.264	0.140	0.237	**0.086**	**0.208**	0.178	0.243	0.222	0.291	0.410	0.282		
	192	**0.408**	**0.416**	0.352	0.396	0.336	0.368	0.235	0.315	0.154	0.250	0.182	0.308	0.216	0.275	0.249	0.312	0.423	0.287		
	336	0.455	0.454	0.423	0.446	0.381	0.402	0.303	0.365	0.169	0.267	**0.306**	0.416	0.265	0.319	0.269	0.326	0.436	0.296	0.321	0.352
	720	0.500	0.510	0.634	0.565	0.447	0.443	0.407	0.424	0.203	0.300	**0.565**	**0.586**	0.325	0.363	0.271	0.328	0.466	0.315		
MoLE	96	0.371	0.393	0.291	0.354	0.299	0.344	0.167	0.262	0.140	0.238	0.088	0.213	0.175	0.237	0.222	0.291	0.410	0.282		
	192	0.409	0.420	0.357	0.396	0.336	0.367	0.236	0.318	0.154	0.250	0.488	0.491	0.216	0.275	0.250	0.312	0.423	0.288		
	336	0.446	0.447	0.441	0.458	0.374	0.393	0.304	0.360	0.169	0.268	0.599	0.576	0.260	0.310	0.268	0.325	0.437	0.297	0.353	0.367
	720	0.495	0.506	0.721	0.604	0.439	0.438	0.396	0.416	0.203	0.301	1.053	0.770	0.327	0.369	0.271	0.329	0.466	0.315		
DLinear	96	0.374	0.398	0.281	0.347	0.307	0.350	**0.165**	**0.250**	0.140	0.237	0.110	0.252	0.174	0.235	0.222	0.292	0.410	0.282		
	192	0.430	0.440	0.367	0.404	0.340	0.373	0.227	0.307	0.153	0.250	0.208	0.349	0.219	0.281	0.249	0.313	0.423	0.288		
	336	0.442	0.445	0.438	0.454	0.377	0.397	0.304	0.362	0.169	0.267	0.410	0.501	0.264	0.317	0.268	0.327	0.436	0.296	0.331	0.359
	720	0.497	0.507	0.598	0.549	0.433	0.433	0.431	0.441	0.203	0.299	0.774	0.675	0.324	0.363	0.271	0.326	0.466	0.315		
FEDformer	96	0.387	0.433	0.383	0.430	0.382	0.428	0.256	0.333	0.225	0.334	0.375	0.454	0.276	0.345	0.275	0.374	0.606	0.380		
	192	0.435	0.461	0.420	0.457	0.404	0.441	0.290	0.353	0.232	0.342	0.490	0.524	0.309	0.370	0.308	0.382	0.620	0.383		
	336	0.463	0.474	0.430	0.468	0.479	0.474	0.334	0.380	0.259	0.361	0.715	0.645	0.353	0.395	0.376	0.446	0.623	0.386	0.437	0.436
	720	0.492	0.503	0.479	0.493	0.471	0.466	0.428	0.436	0.286	0.382	1.494	0.948	0.395	0.419	0.337	0.414	0.640	0.396		
Autoformer	96	0.551	0.499	0.381	0.427	0.459	0.463	0.266	0.341	0.208	0.323	0.351	0.445	0.294	0.369	1.019	0.753	0.679	0.419		
	192	0.514	0.505	0.398	0.437	0.550	0.512	0.298	0.361	0.210	0.327	0.542	0.559	0.341	0.388	0.904	0.706	0.677	0.417		
	336	0.501	0.511	0.393	0.437	0.608	0.548	0.331	0.378	0.218	0.333	0.661	0.624	0.372	0.411	0.964	0.710	0.663	0.415	0.523	0.481
	720	0.513	0.505	0.448	0.475	0.590	0.549	0.414	0.428	0.280	0.376	1.373	0.908	0.384	0.394	0.763	0.642	0.697	0.437		

Discussion To ensure a fair and rigorous comparison with CycleNet (Xu et al., 2024), we conduct comprehensive experiments, with results presented in Table 12. The table clearly shows that *PENGUIN* outperforms both variants of CycleNet by a considerable margin.

D.2 Results on M4 dataset

In the previous sections, we have comprehensively evaluated *PENGUIN* on benchmark datasets of time series forecasting. However, these datasets span an excessively short time horizon (e.g., the weather dataset covers

Table 11: Multivariate long-term forecasting results. The look-back length L is fixed as 96 and the results are averaged over prediction horizons of $H \in \{96, 192, 336, 720\}$. The best results are in **bold**. * indicates results not reported in the original paper, and the results are reproduced.

Model	*PENGUIN*		CycleNet/Linear		CycleNet/MLP		CATS		PatchTST		iTransformer	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.426**	0.433	0.432	**0.427**	0.457	0.441	0.454	0.447	0.469	0.454	0.454	0.447
ETTh2	**0.378**	**0.402**	0.383	0.404	0.388	0.409	0.383	0.407	0.387	0.407	0.383	0.407
ETTm1	**0.378**	**0.394**	0.386	0.395	0.379	0.396	0.407	0.410	0.387	0.400	0.407	0.410
ETTm2	0.286	0.332	0.272	0.315	**0.266**	**0.314**	0.288	0.332	0.281	0.326	0.288	0.332
Exchange	**0.355**	**0.401**	0.383*	0.414*	0.378*	0.410*	0.374*	0.409*	0.367	0.404	0.360	0.403
Traffic	0.440	**0.278**	0.485	0.313	0.472	0.301	**0.428**	0.282	0.481	0.304	**0.428**	0.282
Weather	0.246	0.274	0.254	0.279	**0.243**	**0.271**	0.258	0.278	0.259	0.281	0.258	0.278
Electricity	0.176	0.266	0.170	0.260	**0.168**	**0.259**	0.178	0.270	0.205	0.290	0.178	0.270
Solar	0.224	**0.261**	0.235	0.270	**0.210**	**0.261**	0.233	0.262	0.270	0.307	0.233	0.262
avg.	**0.323**	**0.337**	0.333	0.341	0.329	0.340	0.333	0.344	0.345	0.352	0.332	0.343

Table 12: Rigorous comparison with CycleNet. The look-back length L is fixed as 336 and the results are averaged over prediction horizons of $H \in \{96, 192, 336, 720\}$. * indicates results that are reproduced, while + indicates results from the original paper (Table 8 of CycleNet paper (Xu et al., 2024)).

Model	*PENGUIN*		CycleNet/Linear+		CycleNet/MLP+		CycleNet/Linear*		CycleNet/MLP*	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.419	0.430	0.415	0.426	0.437	0.440	0.422	0.432	0.432	0.437
ETTh2	0.353	0.394	0.355	0.398	0.371	0.407	0.357	0.401	0.373	0.410
ETTm1	0.343	0.377	0.355	0.379	0.361	0.390	0.357	0.376	0.362	0.389
ETTm2	0.263	0.319	0.251	0.309	0.271	0.324	0.264	0.323	0.271	0.324
Traffic	0.388	0.262	0.421	0.289	0.413	0.281	0.423	0.288	0.414	0.283
Weather	0.228	0.267	0.242	0.278	0.226	0.266	0.246	0.279	0.230	0.269
Electricity	0.165	0.261	0.158	0.250	0.157	0.251	0.160	0.253	0.158	0.252
Solar	0.200	0.253	0.226	0.265	0.194	0.255	0.223	0.277	0.197	0.260
avg.	**0.294**	**0.320**	0.302	0.324	0.303	0.326	0.306	0.328	0.304	0.328

Table 13: Forecasting results on the M4 dataset across different time granularities under SMAPE, MASE, and OWA metrics. Smaller values indicate better performance. **Bold** values denote the best average performance.

	Model	*PENGUIN*	CATS	CycleNet/MLP	CycleNet/Linear	PatchTST
Yearly	SMAPE	13.364 ± 0.012	13.575 ± 0.007	13.534 ± 0.013	14.129 ± 0.002	13.529 ± 0.032
	MASE	3.003 ± 0.006	3.065 ± 0.008	3.056 ± 0.009	3.050 ± 0.001	3.042 ± 0.011
	OWA	0.786 ± 0.001	0.801 ± 0.002	0.798 ± 0.002	0.816 ± 0.000	0.796 ± 0.002
Quarterly	SMAPE	10.202 ± 0.012	10.342 ± 0.013	10.310 ± 0.016	10.762 ± 0.001	10.843 ± 0.071
	MASE	1.201 ± 0.004	1.222 ± 0.008	1.215 ± 0.003	1.305 ± 0.002	1.292 ± 0.005
	OWA	0.902 ± 0.002	0.915 ± 0.001	0.911 ± 0.002	0.964 ± 0.001	0.963 ± 0.005
Monthly	SMAPE	12.872 ± 0.019	13.750 ± 0.101	13.011 ± 0.018	13.346 ± 0.000	18.176 ± 0.482
	MASE	0.958 ± 0.003	1.071 ± 0.013	0.969 ± 0.002	1.015 ± 0.000	1.626 ± 0.110
	OWA	0.897 ± 0.002	0.980 ± 0.010	0.907 ± 0.001	0.940 ± 0.000	1.420 ± 0.209
Weekly	SMAPE	9.791 ± 0.050	10.116 ± 0.243	12.025 ± 0.252	12.436 ± 0.267	9.821 ± 0.222
	MASE	3.166 ± 0.022	3.174 ± 0.102	3.887 ± 0.158	4.496 ± 0.108	3.157 ± 0.096
	OWA	1.079 ± 0.035	1.104 ± 0.029	1.356 ± 0.083	1.488 ± 0.049	1.134 ± 0.025
Daily	SMAPE	3.063 ± 0.012	3.161 ± 0.119	3.113 ± 0.012	3.415 ± 0.000	3.322 ± 0.041
	MASE	3.286 ± 0.014	3.385 ± 0.122	3.354 ± 0.016	3.744 ± 0.000	3.532 ± 0.043
	OWA	1.004 ± 0.003	1.035 ± 0.038	1.023 ± 0.004	1.132 ± 0.000	1.0795 ± 0.014
Hourly	SMAPE	18.802 ± 0.002	20.374 ± 0.125	18.335 ± 0.158	21.361 ± 0.000	21.734 ± 0.542
	MASE	2.769 ± 0.001	3.834 ± 0.137	2.939 ± 0.065	3.392 ± 0.000	4.065 ± 0.334
	OWA	1.090 ± 0.001	1.525 ± 0.045	1.168 ± 0.021	1.289 ± 0.000	1.439 ± 0.090
Average	SMAPE	**11.360**	11.886	11.721	12.574	12.904
	MASE	**2.422**	2.625	2.570	2.833	2.785
	OWA	**0.972**	1.060	1.027	1.104	1.138

only a 1-year period, and ETT datasets cover only 2 years), rendering them inadequate for evaluating models specifically designed to capture seasonal patterns. To this end, we extend our evaluation to the M4 datasets from `https://www.kaggle.com/datasets/yogesh94/m4-forecasting-competition-dataset`, which contains time series data collected over years. The M4 dataset is divided into Yearly, Quarterly, Monthly, Weekly, Daily, and Hourly. The experimental results are reported in Table 13 using a significant test with five random seeds in {2021,2022,2023,2024,2025}. We have noticed that *PENGUIN* outperforms CycleNet on all six frequencies of the M4 dataset, with an overall improvement of 3.17% (11.721 $\rightarrow$ 11.360) over CycleNet/MLP and 10.68% (12.574 $\rightarrow$ 11.360) over CycleNet/Linear under the SMAPE metric. Moreover, *PENGUIN* achieves significant improvements on the yearly, quarterly, monthly, and hourly datasets.

D.3 Ablation Study on More Dataset

We conduct an additional ablation study on the Weather and ETTm2 dataset and show the results in the table 14. We can draw three conclusions. First, incorporating attention bias consistently improves forecasting performance. Second, integrating periodic information also yields significant gains. Finally, using multiple periodic components is beneficial even in datasets without clear multi-periodicity, as fine-grained patterns (e.g., hourly) capture variations that coarser patterns (e.g., daily) miss.

Table 14: More experiment results of different choices of periodic length. The look-back length L is fixed as 336 and the prediction horizons of $H \in \{96, 192, 336, 720\}$. Throughout this experiment, we set the number of attention heads to 12 to ensure that the number of groups g is a factor of h.

Weather Dataset																
Case	No Bias		Non-Periodic		Periodic						Both					
$\mathcal{P}$	$\emptyset$		$\emptyset$		{6}		{144}		{6,144}		{6}		{144}		{6,144}	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	0.151	0.199	0.151	0.200	0.151	0.200	0.150	0.202	0.149	0.201	0.152	0.201	0.151	0.202	0.150	0.200
192	0.200	0.249	0.196	0.245	0.193	0.242	0.194	0.243	0.193	0.240	0.195	0.243	0.192	0.240	0.194	0.241
336	0.252	0.287	0.250	0.284	0.246	0.282	0.245	0.284	0.244	0.281	0.243	0.281	0.243	0.282	0.243	0.282
720	0.334	0.344	0.326	0.340	0.322	0.337	0.323	0.339	0.322	0.342	0.326	0.340	0.321	0.337	0.318	0.334
avg.	0.234	0.270	0.230	0.267	0.228	0.265	0.228	0.267	0.227	0.266	0.229	0.266	0.227	0.265	0.226	0.264
ETTm2 Dataset																
Case	No Bias		Non-Periodic		Periodic						Both					
$\mathcal{P}$	$\emptyset$		$\emptyset$		{4}		{96}		{4,96}		{4}		{96}		{4,96}	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	0.174	0.265	0.172	0.262	0.167	0.257	0.168	0.258	0.168	0.257	0.167	0.257	0.172	0.261	0.171	0.260
192	0.229	0.301	0.226	0.298	0.223	0.294	0.227	0.296	0.224	0.294	0.224	0.294	0.224	0.294	0.224	0.295
336	0.288	0.337	0.285	0.334	0.279	0.333	0.282	0.333	0.279	0.333	0.278	0.332	0.278	0.331	0.282	0.332
720	0.378	0.391	0.373	0.390	0.370	0.388	0.374	0.390	0.370	0.388	0.370	0.389	0.370	0.390	0.371	0.391
avg.	0.267	0.324	0.264	0.321	0.260	0.318	0.263	0.319	0.260	0.318	0.260	0.318	0.261	0.319	0.262	0.320

D.4 Ablation Study of Patching Stride

To elaborate on the performance of various patching strides in *PENGUIN*, we conduct comprehensive experiments of different patching lengths P and stride S illustrated in Table 15. The results consistently demonstrate that reducing the stride S enhances prediction accuracy across multiple datasets and metrics. For instance, in the ETTh1 dataset, a smaller stride significantly improves performance, with the average MSE decreasing from 0.426 at $P = 8, S = 4$ to 0.419 at $P = 1, S = 1$, with 1.64% decrease. Similarly, the trend is also evident for the Weather dataset, where the smallest stride configuration ($P = 4, S = 1$) achieves the lowest MSE. Overall, shorter strides allow *PENGUIN* to process a greater number of overlapping patches, enabling finer-grained feature extraction and better long-range dependency modeling.

P/S		16/8		8/8		8/4		4/4		4/2		2/2		4/1		2/1		1/1	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.382	0.401	0.382	0.400	0.380	0.397	0.381	0.398	0.379	0.397	0.378	0.397	0.379	0.398	0.378	0.396	0.380	0.400
	192	0.416	0.421	0.416	0.419	0.417	0.418	0.418	0.418	0.416	0.418	0.414	0.417	0.422	0.423	0.416	0.418	0.418	0.424
	336	0.435	0.434	0.437	0.435	0.437	0.433	0.434	0.429	0.441	0.438	0.432	0.430	0.459	0.455	0.435	0.434	0.431	0.436
	720	0.459	0.474	0.466	0.478	0.466	0.478	0.458	0.470	0.454	0.466	0.455	0.467	0.446	0.460	0.463	0.472	0.431	0.436
	avg.	0.423	0.433	0.425	0.433	0.426	0.432	0.423	0.429	0.423	0.430	0.420	0.428	0.427	0.434	0.423	0.430	**0.419**	**0.430**
ETTh2	96	0.287	0.344	0.289	0.345	0.290	0.346	0.288	0.344	0.287	0.345	0.287	0.343	0.291	0.347	0.291	0.347	0.290	0.347
	192	0.354	0.387	0.355	0.387	0.357	0.389	0.354	0.388	0.355	0.389	0.362	0.396	0.359	0.393	0.359	0.393	0.354	0.389
	336	0.380	0.410	0.380	0.410	0.379	0.410	0.379	0.412	0.380	0.413	0.378	0.410	0.396	0.425	0.396	0.425	0.379	0.412
	720	0.397	0.433	0.399	0.436	0.398	0.434	0.394	0.432	0.399	0.437	0.400	0.438	0.392	0.430	0.392	0.430	0.397	0.435
	avg.	0.355	0.394	0.356	0.395	0.356	0.395	**0.354**	**0.394**	0.355	0.396	0.357	0.397	0.360	0.399	0.360	0.399	0.355	0.396
ETTm1	96	0.293	0.344	0.301	0.345	0.286	0.339	0.293	0.345	0.281	0.337	0.291	0.342	0.293	0.348	0.295	0.347	0.299	0.351
	192	0.332	0.370	0.335	0.371	0.329	0.367	0.327	0.365	0.319	0.363	0.327	0.365	0.322	0.365	0.334	0.372	0.332	0.372
	336	0.362	0.387	0.361	0.388	0.357	0.387	0.356	0.387	0.353	0.385	0.355	0.387	0.353	0.386	0.354	0.386	0.356	0.389
	720	0.416	0.418	0.419	0.421	0.405	0.418	0.404	0.416	0.403	0.416	0.401	0.416	0.406	0.418	0.399	0.415	0.407	0.420
	avg.	0.351	0.380	0.354	0.381	0.344	0.378	0.345	0.378	**0.339**	**0.375**	0.344	0.378	0.344	0.379	0.346	0.380	0.349	0.383
Weather	96	0.150	0.198	0.152	0.203	0.150	0.199	0.152	0.202	0.152	0.203	0.151	0.202	0.148	0.200	0.200	0.195	0.157	0.207
	192	0.195	0.242	0.197	0.244	0.193	0.240	0.195	0.244	0.193	0.243	0.198	0.245	0.194	0.241	0.242	0.245	0.197	0.245
	336	0.248	0.282	0.247	0.281	0.246	0.281	0.246	0.283	0.246	0.282	0.249	0.283	0.243	0.282	0.282	0.319	0.249	0.285
	720	0.324	0.334	0.321	0.333	0.323	0.337	0.320	0.334	0.320	0.334	0.319	0.333	0.320	0.337	0.319	0.333	0.321	0.336
	avg.	0.229	0.264	0.321	0.333	0.228	0.264	0.228	0.266	0.228	0.266	0.229	0.266	**0.226**	0.265	0.227	**0.264**	0.231	0.268

Table 15: Additional experiment results of the impact of various patching strides. The look-back length L is fixed as 336 and the prediction horizons of $H \in \{96, 192, 336, 720\}$. In the table, P indicates the patching length, while S indicates the stride.

E Additional Robustness Study

E.1 Robustness Towards Varied Look-Back Window

To assess *PENGUIN*'s robustness, we conducted experiments on the ETTm1 and Traffic datasets, varying the look-back window length, denoted as L, across a range of values: $[48, 96, 192, 264, 336, 528]$. As shown in the experimental results in Figure 5, *PENGUIN* consistently outperforms baseline models across these different window lengths when the input length is larger than 96. We hypothesize that an input length of 48 is insignificant for capturing temporal patterns. These results validate *PENGUIN*'s ability to effectively handle varying temporal scales and long-range dependencies in time series data.

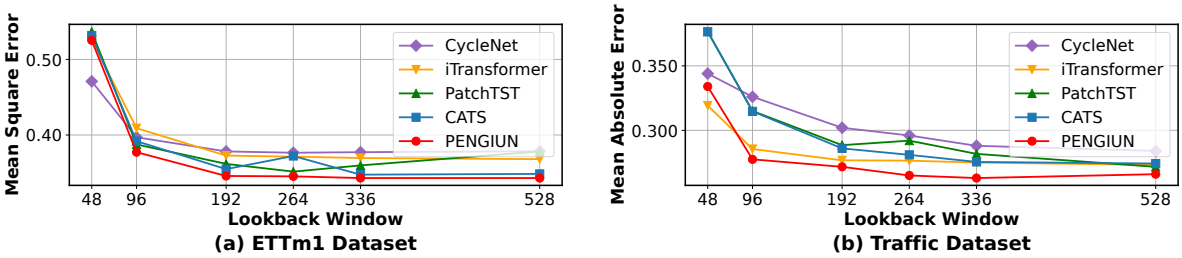

Figure 5: The experimental results of various models with diverse lookback windows L on the ETTm1 and Traffic dataset. The experiments are averaged over the predicted horizon of $H \in \{96, 192, 336, 720\}$.

E.2 Robustness Towards Random Seeds

In addition, we conducted robustness tests used in the main comparison experiments. To validate the stability of our approach against random parameter initialization, we evaluated *PENGUIN* with five different random seeds {2021,2022,2023,2024,2025}. Table 16 reports the standard deviation of MSE and MAE for different methods across these seeds. Results show that *PENGUIN* achieves statistically significant improvements over strong baselines on these datasets.

Table 16: Robustness analysis with five random seeds compared to baselines. The look-back length L is fixed as 336 and the results are averaged over prediction horizons of $H \in \{96, 192, 336, 720\}$.

Model		*PENGUIN*		CycleNet/Linear		PatchTST	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
Traffic	96	0.357 ± 0.000	0.247 ± 0.000	0.399 ± 0.000	0.276 ± 0.000	0.378 ± 0.002	0.270 ± 0.002
	192	0.376 ± 0.000	0.253 ± 0.000	0.413 ± 0.000	0.282 ± 0.000	0.394 ± 0.002	0.274 ± 0.003
	336	0.390 ± 0.000	0.262 ± 0.000	0.426 ± 0.000	0.288 ± 0.000	0.406 ± 0.003	0.283 ± 0.004
	720	0.428 ± 0.000	0.286 ± 0.000	0.453 ± 0.000	0.305 ± 0.000	0.443 ± 0.004	0.298 ± 0.004
Solar	96	0.180 ± 0.002	0.245 ± 0.002	0.202 ± 0.001	0.286 ± 0.001	0.195 ± 0.004	0.252 ± 0.003
	192	0.202 ± 0.005	0.257 ± 0.006	0.225 ± 0.001	0.297 ± 0.000	0.201 ± 0.003	0.263 ± 0.002
	336	0.203 ± 0.006	0.252 ± 0.003	0.227 ± 0.001	0.261 ± 0.000	0.208 ± 0.005	0.274 ± 0.004
	720	0.213 ± 0.005	0.257 ± 0.003	0.237 ± 0.001	0.264 ± 0.000	0.211 ± 0.004	0.268 ± 0.003
ETTm1	96	0.290 ± 0.001	0.341 ± 0.000	0.301 ± 0.000	0.344 ± 0.000	0.289 ± 0.001	0.345 ± 0.001
	192	0.319 ± 0.000	0.361 ± 0.000	0.335 ± 0.000	0.364 ± 0.000	0.339 ± 0.002	0.378 ± 0.002
	336	0.356 ± 0.001	0.387 ± 0.000	0.369 ± 0.000	0.384 ± 0.000	0.373 ± 0.002	0.400 ± 0.001
	720	0.409 ± 0.000	0.418 ± 0.001	0.424 ± 0.000	0.414 ± 0.000	0.434 ± 0.004	0.438 ± 0.003

F Limitation and Future Work

PENGUIN demonstrates its effectiveness in LTSF with the proposed periodic causal attention. However, there are several potential limitations of *PENGUIN* that warrant discussion here. Firstly, *PENGUIN* might not be suitable for datasets with varied periodic lengths over time, such as electrocardiogram (ECG) data, since *PENGUIN* employs a predefined set of periodic lengths. Secondly, *PENGUIN*, as well as other Transformer-based methods, typically requires more GPU memory and computation cost compared to linear models, because the attention matrix is square in relation to the input length. In future work, we plan to develop flash attention to conserve GPU memory and explore linear attention approaches to reduce computational costs.