DCNET: MULTIVARIATE TIME SERIES FORECASTING VIA GLOBAL AND LOCAL DUAL CONTEXT CONFERENCE SUBMISSIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time series forecasting is of great significance in fields such as energy scheduling, traffic management, and financial analysis. This paper proposes the Dual Context (DC) technique and reconstructs the retrieval space of the attention mechanism based on this technique; at the same time, combined with a novel multilayer perceptron, DCNet is constructed to fully exploit the coupling features between global and local contexts. DCNet not only avoids the isolated modeling state of global periodic patterns and local evolution dynamics in previous studies, but also effectively improves the problem of blind coupling of variables in existing inverted architectures under complex environments by introducing a global reference baseline. DCNet achieves overall leading accuracy on 12 widely used real-world datasets. In addition, further experiments show that the DC technique not only performs well on models based on the Transformer architecture, but also significantly improves linear models, proving that the technique has good universality and cross-architecture transferability. Our code is available at https://anonymous.4open.science/r/DCNet7218.

1 INTRODUCTION

Multivariate time series forecasting (MTSF) plays an irreplaceable role in many fields of modern society Lim & Zohren (2021), and is widely applied in complex real-world scenarios such as energy and power scheduling Zhou et al. (2021), traffic flow forecasting Yu et al. (2017), financial market analysis Sezer et al. (2020), and healthcare monitoring and meteorological forecasting Wu et al. (2021). Nowadays, the core task of MTSF has shifted from simple sequence mapping to how to deeply mine the intrinsic characteristics of time series.

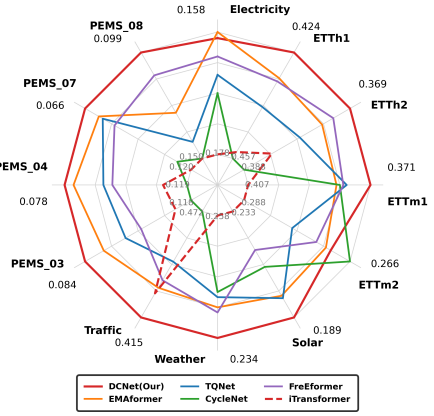

Figure 1: Performance of DCNet. All results are the average of the evaluation metric MSE across four tasks.

Early studies mainly modeled in the temporal dimension, embedding the observations at each time step as Token, such as Informer Zhou et al. (2021), Autoformer Wu et al. (2021), and FEDformer Zhou et al. (2022). However, such methods struggle to effectively distinguish the complex relationships among different variables, and the strict sequential order information of time series is easily lost within Transformer Vaswani et al. (2017) as positional encoding weakens. Subsequently, iTransformer Liu et al. (2024) maps the complete sequence of each variable to a Token, achieving explicit modeling of inter-variable correlations and effectively improving forecasting accuracy.

However, studies have found that calculating variable correlations based on raw numerical similarity is significantly limited in high-noise environments. Because variables are treated as homogeneous

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

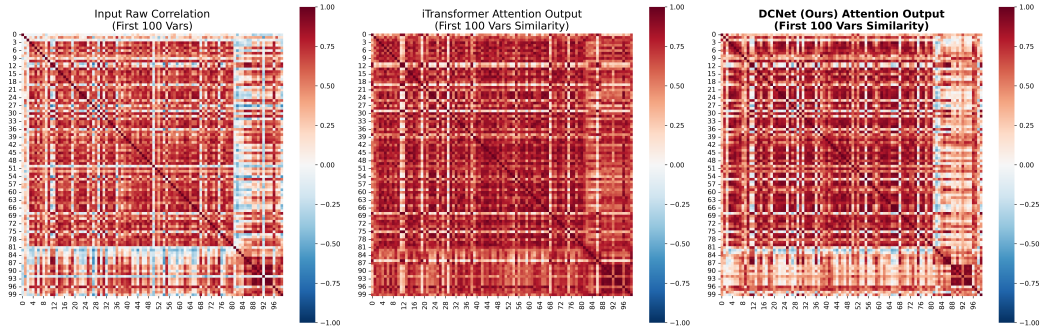

Figure 2: Correlation map of the first 100 variables in the first input sequence on the Electricity test set. The left panel shows the Pearson correlation coefficients of the raw data, the middle panel shows the variable correlations learned by iTransformer Liu et al. (2024), where the originally ordered variables are learned in a chaotic manner, and the right panel shows the variable correlations learned by DCNet, where the associations among variables become clear again.

Tokens and lack a stable baseline reference, the attention mechanism struggles to distinguish true physical associations from random couplings caused by noise, resulting in the capture of numerous spurious correlations. Such spurious correlations obscure the essential inter-variable logic, reflecting that when lacking global context guidance, the model finds it difficult to establish reliable variable interactions from limited local observations. This phenomenon can be confirmed from Figure. 2.

To this end, we propose the Dual Context (DC) technique and construct DCNet, aiming to mine the coupling features of different contexts through neural networks. In the temporal dimension, DC enables local dynamics to perceive in real time their temporal position within the global period, effectively eliminating the perceptual blind spots caused by the lack of global coordinate anchoring; in the variable dimension, this technique introduces a stable global periodic context as an interaction baseline, guiding the neural network to spontaneously capture the essential associations among variables, thereby effectively filtering out spurious correlations in high-noise environments. Our main contributions are as follows:

- We propose the DC technique, through which collaborative perception of global periodic patterns and local evolution dynamics is achieved, breaking the previous isolated state of the two in modeling context.
- Combining the DC technique with a novel multilayer perceptron (MLP), we design DCNet. As shown in Figure. 1, DCNet achieves overall leading accuracy on 12 widely used real-world datasets.
- Through extensive experiments, the universality and cross-architecture effectiveness of the DC technique are demonstrated. This technique not only achieves significant improvements on models based on the Transformer architecture, but also performs excellently on linear models based on the MLP structure.

2 Related Work

In multivariate time series forecasting, explicit modeling of inter-variable correlations has been a research hotspot in recent years. Early Transformer Vaswani et al. (2017) architectures, such as Informer Zhou et al. (2021), Autoformer Wu et al. (2021), FEDformer Zhou et al. (2022), ETSformer Woo et al. (2022), and Pyraformer Liu et al. (2021), generally adopted the time-step embedding paradigm, treating the observation vector at each time step as a Token. However, this approach has difficulty effectively distinguishing the complex relationships among different variables, and the strict sequential order information of time series is easily lost within the Transformer as positional encoding weakens. This leads to the fact that even though these models have huge numbers of parameters, their performance often struggles to surpass the extremely simple linear model DLinear Zeng et al. (2023).

Subsequently, PatchTST Nie et al. (2022) proposed the channel independence assumption, a strategy that separates the modeling of each variable and enables the model to forecast each channel independently. Works such as SegRNN Lin et al. (2025c), TiDE Das et al. (2023), N-HiTS Challu et al. (2023), FITS Xu et al. (2023), STID Shao et al. (2022), TimeMixer Wang et al. (2024a), PDF Dai et al. (2024), CATS Kim et al. (2024), SparseTSF Lin et al. (2025b), and CycleNet Lin et al. (2024) followed this practice. Nevertheless, the lack of cross-variable relationship modeling has become a performance bottleneck of this paradigm, limiting their performance in multivariate scenarios.

After that, iTransformer Liu et al. (2024) fundamentally changed the modeling perspective through Inverted Embedding, mapping the complete sequence of each variable to a Token, thereby achieving explicit learning of inter-variable correlations. Recently, works including Leddam Yu et al. (2024), CARD Xue et al. (2023), TimeXer Wang et al. (2024b), FreEformer Yue et al. (2025), TQNet Lin et al. (2025a), and DUET Qiu et al. (2025) have all followed this practice.

DCNet also belongs to the inverted embedding approach. On the basis of explicitly capturing variable correlations using this paradigm, DCNet further introduces the DC technique, aiming to address the limitation of blind variable coupling under existing inverted architectures and to achieve deep collaborative modeling of global periodic patterns and local evolution dynamics.

3 DCNET

Problem statement: Given the historical observation sequence $X = \{x_1, \ldots, x_L\} \in \mathbb{R}^{L \times N}$, where L denotes the look-back window length and N denotes the number of variables (channels), which typically correspond to observation indicators of different dimensions. The goal of MTSF is to establish a mapping model to predict the numerical sequence of the future S time steps $Y = \{x_{L+1}, \ldots, x_{L+S}\} \in \mathbb{R}^{S \times N}$ based on the historical observation sequence X.

3.1 OVERALL ARCHITECTURE

This paper proposes an innovative architecture for MTSF. As shown in Figure. 3, the model first handles data non-stationarity through normalization, and then constructs high-dimensional representations through a linear layer. Inside the architecture, we use the DC technique to reconstruct the retrieval space of the attention mechanism to enhance the robustness of the model; at the same time, we introduce a novel MLP based on SwiGLU Shazeer (2020), which enhances the nonlinear representation capability of the model through dual-branch interaction. Finally, the model outputs forecasting results through linear projection and restores them to the original scale through denormalization.

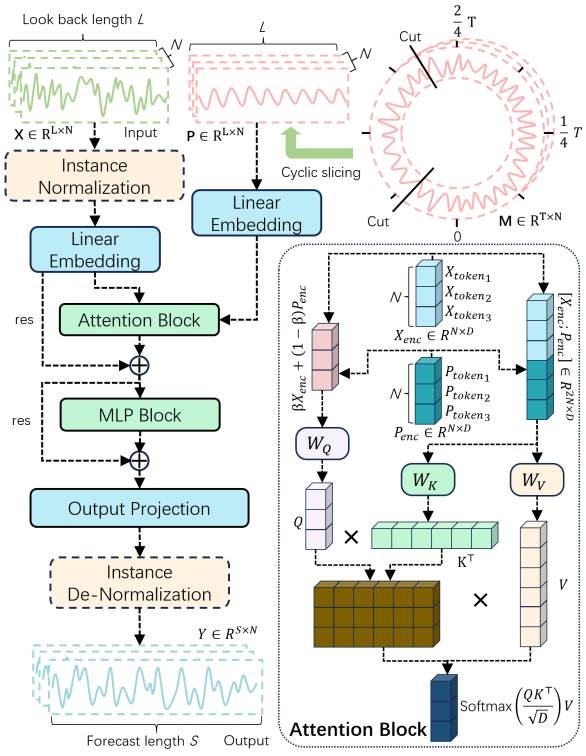

Figure 3: Overview of DCNet, where the bottom-right part shows the DC-improved attention mechanism.

3.2 INPUT AND PROCESSING

To capture global long-term dependencies, we introduce a periodically shifted learnable vector from CycleNet Lin et al. (2024), denoted as $M \in \mathbb{R}^{T \times N}$, where T represents the period length. For a given historical observation sequence X, the model retrieves the corresponding periodic pattern $P \in \mathbb{R}^{L \times N}$ from M according to the current time index to form the

global periodic context (GPC), representing the idealized trajectory of multivariate variables at this time scale. Meanwhile, the original historical observations X are defined as the local evolution context (LEC). Both are mapped, through an inverted embedding manner similar to iTransformer Liu et al. (2024) (without timestamp token), from the observation sequence over the entire look-back window L to feature vectors of dimension D, where "variables" are "Tokens", yielding the representation of LEC $X_{enc} \in \mathbb{R}^{N\times D}$ and the representation of GPC $P_{enc} \in \mathbb{R}^{N\times D}$:

$$X_{enc} = \text{Linear}_{Local}\left(X^{\top}\right), \quad P_{enc} = \text{Linear}_{Global}\left(P^{\top}\right) \tag{1}$$

3.3 DC-Improved Attention Mechanism

DC technique is the core of DCNet, aiming to achieve deep collaborative modeling of local evolution information and global periodic patterns. In the traditional self-attention mechanism, the query vector (Q), key vector (K), and value vector (V) are usually derived from X_{enc}. However, real-world time series data often contain non-stationary interference such as random noise or extreme outliers. Since these vectors depend entirely on the raw observation data, such interference causes the allocation of attention weights to shift, making the model over-capture transient noise while ignoring the essential long-term patterns, thereby reducing the reliability of forecasting results. To this end, we propose the DC technique. As shown in Figure. 3 (bottom-right), we use this technique to reconstruct the retrieval space of the attention mechanism to enhance the robustness of the model.

First, by introducing a weighting factor β to fuse X_{enc} and P_{enc}, a composite query basis $H_Q \in \mathbb{R}^{N\times D}$ is generated:

$$H_Q = \beta X_{enc} + (1-\beta)\, P_{enc} \tag{2}$$

Second, the GPC representation is introduced to expand the retrieval space originally based on the LEC representation. We concatenate X_{enc} and P_{enc} along the variable dimension to construct a composite context representation of dimension $2N \times D$:

$$H_{KV} = [X_{enc}; P_{enc}] \in \mathbb{R}^{2N\times D} \tag{3}$$

The formulas for computing the query $Q \in \mathbb{R}^{N\times D}$, key $K \in \mathbb{R}^{2N\times D}$, and value $V \in \mathbb{R}^{2N\times D}$ are as follows:

$$Q = H_Q W_Q, \quad K = H_{KV} W_K, \quad V = H_{KV} W_V \tag{4}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{D\times D}$ are linear projection matrices.

This design broadens the retrieval scope of the attention mechanism from a single local representation space to a dual context that includes global representations, and the token scale is expanded from N to 2N. Since local dynamics and global patterns are juxtaposed and decoupled along the variable dimension, this mechanism fully preserves local observation details while providing a global reference baseline unaffected by transient interference for the retrieval process, thereby enhancing the model's ability to extract underlying patterns. Finally, the resulting attention output $H_{out} \in \mathbb{R}^{N\times D}$:

$$H_{out} = \text{Attention}\,(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{D}}\right) V \tag{5}$$

Under this architecture, the attention weights reflect the model's dynamic trade-off between the local observation track (the first N entries of the key matrix K) and the global empirical track (the last N entries of the key matrix K). By performing unified computation and aggregation within the expanded space, the DC technique can adaptively adjust the contextual contribution according to the degree of data non-stationarity, effectively decoupling noise from patterns.

In addition, this design maintains morphological consistency while expanding the representation: although the retrieval space increases to $2N$ tokens, since Q remains at the scale of N variables, the output matrix remains $N \times D$. This non-invasive design ensures structural compatibility for

multi-layer stacking, and significantly improves the information density of feature representations without increasing architectural complexity.

3.4 A Novel MLP Based on SwiGLU

Given the outstanding feature extraction capability demonstrated by SwiGLU Shazeer (2020) in large language models such as Llama Touvron et al. (2023) and PaLMChowdhery et al. (2023) , we introduce it into MTSF to handle complex nonlinear dynamic characteristics in the data. This structure abandons the traditional "fully connected + ReLU" combination and constructs a dual-branch interaction mechanism through three learnable matrices W_1, W_2, W_3. One branch utilizes the smoothness of the SiLU activation function to dynamically filter and enhance the information from the other branch through element-wise product. For the input feature representation x, its computation process is as follows:

$$\mathrm{SwiGLU}(x) = (\mathrm{SiLU}(xW_1) \otimes xW_3)W_2 \tag{6}$$

where $\otimes$ denotes element-wise multiplication.

3.5 Instance Normalization and Output

The model extracts deep features by stacking multiple encoder layers. Each layer consists of the improved attention module and FFN, supplemented by residual connections and layer normalization to ensure the stability of feature extraction. After processing through multiple encoder layers, the final hidden representation $H_{final} \in \mathbb{R}^{N \times D}$ is obtained. Following the design of state-of-the-art models such as iTransformer Liu et al. (2024) and TQNet Lin et al. (2025a), the output layer utilizes a linear projection to map the hidden representations to the prediction dimension. By projecting the feature vectors along the temporal dimension, the predicted sequence $\hat{Y} \in \mathbb{R}^{S \times N}$ is derived:

$$\hat{Y} = \left(\mathrm{Linear}_{out}\left(H_{final}\right)\right)^T \tag{7}$$

To address the distribution shift problem in multivariate time series, the model implements instance normalization RevIN Kim et al. (2021) (without learnable affine parameters) at the input.

4 Experiments

4.1 Setup

Datasets. To comprehensively evaluate the performance of DCNet, we conducted extensive experiments on 12 widely used real-world datasets, including ETT (4 subsets), electricity, weather, Solar-Energy, traffic, and PEMS (four subsets). These datasets vary in scale, channels (the number of variables), sampling intervals, and domains. Table 1 summarizes the detailed information of these datasets.

Baseline. We compare DCNet with several strong representative models from recent years, including EMAformer Zhang et al. (2026), FerEformer Yue et al. (2025), TQNet Lin et al. (2025a), DUET Qiu et al. (2025), FilterTS Wang et al. (2025), CycleNet Lin et al. (2024), Leddam Yu et al. (2024), TimeXer Wang et al. (2024b), and iTransformer Liu et al. (2024).

Experimental setup. The long-term forecasting task follows the benchmark configuration of Informer Zhou et al. (2021), SCINet Liu et al. (2022), and Autoformer Wu et al. (2021): the lookback window is fixed as $L = 96$, the prediction lengths for the PEMS dataset are $S \in \{12, 24, 48, 96\}$, and the prediction lengths for other datasets are $S \in \{96, 192, 336, 720\}$. Performance is evaluated by mean squared error (MSE) and mean absolute error (MAE); smaller values indicate better performance. All experimental results are the average under 6 seeds: 2021, 2022, 2023, 2024, 2025, and 2026.

Table 1: Detailed information of all datasets. "Channels" denotes the number of variables, "Timesteps" represents the number of time steps, and "Interval" is the sampling interval. Hyperparameter T is configured to remain consistent with the stable period length of each dataset, following the settings in CycleNet Lin et al. (2024). "Domain" refers to the application area. The final column provides the split ratios for the training, validation, and test sets, following the protocols established in Informer Zhou et al. (2021), SCINet Liu et al. (2022), and Autoformer Wu et al. (2021).

Dataset	Channels	Timesteps	Interval	HParam. T	Domain	Split ratio
ETTh1	7	14,400	1 hour	24	Electricity	6:2:2
ETTh2	7	14,400	1 hour	24	Electricity	6:2:2
ETTm1	7	57,600	15 mins	96	Electricity	6:2:2
ETTm2	7	57,600	15 mins	96	Electricity	6:2:2
Electricity	321	26,304	1 hour	168	Electricity	7:1:2
Solar	137	52,560	10 mins	144	Energy	7:1:2
Traffic	862	17,544	1 hour	168	Transportation	7:1:2
Weather	21	52,696	10 mins	144	Weather	7:1:2
PEMS03	358	26,208	5 mins	288	Transportation	6:2:2
PEMS04	307	16,992	5 mins	288	Transportation	6:2:2
PEMS07	883	28,224	5 mins	288	Transportation	6:2:2
PEMS08	170	17,856	5 mins	288	Transportation	6:2:2

Table 2: Comparison of MTSF results across 12 real-world datasets. Summarized results are presented as the average over four forecast horizons; full results are provided in Table8. The look-back window L is fixed at 96. **Red-bold** indicates the best performance, while blue-underlined represents the second best. The Count row indicates the number of times each model ranks within the top 2.

Model	DCNet (Ours)		EMAformer (2026)		FreEformer (2025)		TQNet (2025a)		DUET (2025)		FilterTS (2025)		CycleNet (2024)		TimeXer (2024b)		Leddam (2024)		iTransformer (2024)	
Metric	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Electricity	0.159	0.248	**0.158**	**0.247**	0.162	0.251	0.165	0.259	0.172	0.259	0.180	0.272	0.168	0.259	0.171	0.270	0.169	0.263	0.178	0.270
ETTh1	**0.424**	0.429	0.432	**0.425**	0.433	0.434	0.441	0.434	0.443	0.436	0.433	0.436	0.457	0.441	0.437	0.437	0.431	0.430	0.455	0.447
ETTh2	**0.369**	0.394	0.374	0.395	0.372	**0.393**	0.378	0.402	0.372	0.398	0.373	0.396	0.388	0.409	**0.369**	0.396	0.373	0.399	0.383	0.407
ETTm1	**0.371**	**0.390**	0.381	0.393	0.379	0.393	0.378	0.392	0.390	0.393	0.385	0.396	0.379	0.396	0.382	0.397	0.386	0.397	0.407	0.410
ETTm2	0.270	**0.313**	0.271	**0.313**	0.273	0.314	0.278	0.323	0.280	0.324	0.276	0.337	**0.266**	0.314	0.274	0.322	0.281	0.325	0.288	0.332
Solar	**0.189**	0.249	0.198	0.224	0.217	**0.219**	0.197	0.256	0.237	0.268	0.259	0.292	0.210	0.261	0.237	0.302	0.230	0.264	0.233	0.262
Weather	**0.234**	**0.256**	0.240	0.262	0.239	0.260	0.242	0.269	0.251	0.273	0.245	0.274	0.243	0.271	0.241	0.271	0.242	0.272	0.258	0.278
Traffic	**0.415**	0.272	0.431	0.255	0.435	**0.252**	0.445	0.276	0.451	0.279	0.471	0.315	0.472	0.301	0.466	0.287	0.467	0.294	0.428	0.282
PEMS_03	**0.084**	**0.187**	0.090	0.190	0.102	0.206	0.097	0.203	0.105	0.210	0.180	0.279	0.118	0.226	0.112	0.214	0.107	0.210	0.113	0.222
PEMS_04	**0.078**	**0.179**	0.081	0.181	0.094	0.196	0.091	0.196	0.109	0.216	0.209	0.308	0.119	0.232	0.105	0.210	0.103	0.210	0.111	0.221
PEMS_07	**0.066**	**0.158**	0.073	0.162	0.081	0.168	0.075	0.171	0.125	0.183	0.161	0.263	0.113	0.214	0.086	0.182	0.084	0.180	0.120	0.204
PEMS_08	**0.099**	**0.183**	0.128	0.209	0.110	0.195	0.142	0.230	0.125	0.205	0.254	0.320	0.150	0.246	0.175	0.251	0.122	0.211	0.150	0.226
Count	22		12		7		3		0		0		1		1		1		1	

4.2 Main Results

Table 2 summarizes the forecasting errors of our model and the baseline models on 12 real-world benchmark datasets. From the average of the results for 4 prediction lengths, DCNet ranks in the top two for 22 out of a total of 24 forecasting metrics, demonstrating excellent overall forecasting accuracy. Notably, DCNet performs particularly well in high-dimensional scenarios, such as PEMS datasets with an extremely large number of variables. In the average MSE metric, compared with the previous best model, DCNet achieves relative performance improvements of 6.6%, 9.5%, and 10.0% on the PEMS_03, PEMS_07, and PEMS_08 datasets, respectively, indicating that DCNet has a great advantage in handling complex multivariate interaction modeling.

4.3 Ablation Study and Analysis

To evaluate the specific contributions of each component of DCNet, we conducted ablation experiments on the Electricity and PEMS datasets. In addition, to verify that the DC technique is not only effective for the Transformer architecture, we introduced this mechanism into the widely known DLinear Zeng et al. (2023) and CycleNet Lin et al. (2024), and found that it also contributes significantly to linear structures. The experimental results are shown in Table 3.

Table 3: Ablation study results of different components and generality analysis results of the DC technique, full results are provided in Table 6.

Model	DCNet (Ours)		DCNet W/o DC		DCNet W/o MLP		DCNet W/o Both		iTransformer (2024)		DLinear W DC		DLinear (2023)		CycleNet W DC		CycleNet (2024)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	**0.159**	**0.248**	0.173	0.265	0.169	0.263	0.179	0.270	0.178	0.270	**0.165**	**0.263**	0.212	0.300	**0.165**	**0.258**	0.168	0.259
PEMS_03	**0.084**	**0.187**	0.103	0.211	0.089	0.192	0.106	0.215	0.113	0.222	**0.132**	**0.237**	0.278	0.375	**0.101**	**0.205**	0.118	0.226
PEMS_04	**0.078**	**0.179**	0.106	0.212	0.082	0.186	0.107	0.214	0.111	0.221	**0.130**	**0.240**	0.295	0.388	**0.100**	**0.208**	0.119	0.232
PEMS_07	**0.066**	**0.158**	0.082	0.177	0.069	0.163	0.088	0.188	0.120	0.204	**0.126**	**0.226**	0.329	0.396	**0.094**	**0.191**	0.113	0.214
PEMS_08	**0.099**	**0.183**	0.129	0.214	0.104	0.188	0.132	0.219	0.150	0.226	**0.176**	**0.256**	0.379	0.416	**0.127**	**0.217**	0.150	0.246

The control groups are set as follows: 1. **W/o DC**: remove GPC and degenerate to the traditional self-attention mechanism. 2. **W/o MLP**: replace the MLP structure with a stack of traditional fully connected layers and activation functions. 3. **W/o Both (Base)**: this group ablates the first two groups simultaneously. 4. **W DC DLinear**: DLinear extended with GPC. 5. **W DC CycleNet**: CycleNet extended with GPC. It is worth noting that the latter two groups do not contain an attention mechanism, and only extend the single linear layer within them, whose essence is still a linear model based on the MLP structure.

The fully ablated version (**W/o Both**) is structurally similar to iTransformer Liu et al. (2024), and the experimental results on the Electricity dataset also confirm this. The slight difference between the two stems from DCNet removing timestamp tokens. We believe that timestamps and raw variables exhibit significant heterogeneity in physical attributes and variation characteristics, and forcing interaction between them often introduces redundant weight interference. It is worth noting that instance normalization RevIN Kim et al. (2021) is not adopted on the PEMS dataset, which is consistent with mainstream models such as TQNet Lin et al. (2025a), CycleNet Lin et al. (2024), and EMAformer Zhang et al. (2026).

Result analysis. Experiments show that both the DC technique and the novel MLP make positive contributions to model forecasting performance, among which the improvement of the former is significantly greater than that of the latter. This difference stems from the fact that the DC mechanism solves the collaborative perception problem of LEC and GPC from the underlying architecture. By providing a global reference for local fluctuations, it eliminates the model's "perceptual blind spots" under complex spatiotemporal backgrounds, so its improvement is fundamental; while the improved MLP mainly enhances feature expression by optimizing nonlinear mapping, which belongs to enhancement at the representation level. In addition, we transfer the DC technique to benchmark models with non-Transformer architectures. Experiments show that CycleNet and DLinear both achieve substantial performance leaps after incorporating this mechanism, which strongly demonstrates that the DC technique has strong universality and cross-architecture transferability in capturing multivariate temporal patterns.

4.4 Additional Experimental Results

Efficiency analysis. To further comprehensively evaluate model performance, we analyze the trade-off relationship of DCNet and mainstream models across three dimensions: forecasting accuracy, training time, and number of parameters. The results are shown in Figure. 4(a). Experimental results show that DCNet not only achieves second-best forecasting error, but also maintains extremely high training efficiency. In contrast, although EMAformer exhibits competitive accuracy, its number of parameters is significantly larger than that of other models, and its training time is longer. This indicates that EMAformer relies on a huge number of parameters in exchange for performance. When facing high-dimensional large-scale datasets, its computational overhead and storage pressure are enormous, resulting in low practical deployment efficiency. In comparison, by virtue of the DC mechanism, DCNet significantly reduces the parameter scale and training time while still accurately capturing sequence features, achieving the best balance between model accuracy and computational efficiency.

Hyperparameter T analysis. This section investigates the impact of period length T on forecasting performance on the Electricity dataset, with results shown in Fig. 4(b).

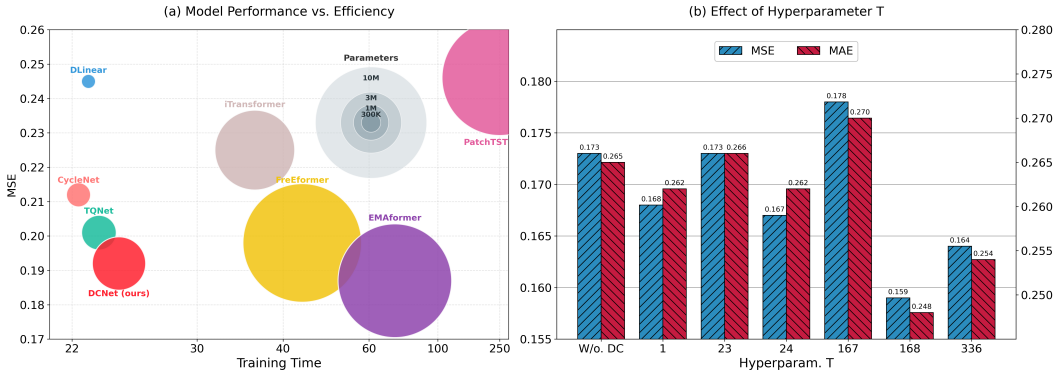

Figure 4: (a) Efficiency analysis on Electricity ($S = 720$), and full results are provided in Table 7. (b) Impact of hyperparameter T on Electricity, averaged over $S \in \{96, 192, 336, 720\}$.

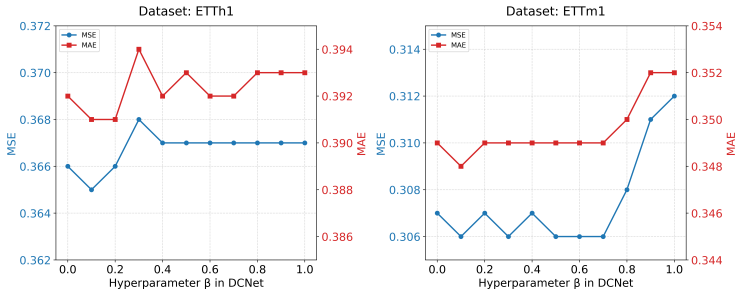

Figure 5: The impact of hyperparameter β on model accuracy, with both look-back window length and prediction length set to 96.

When $T = 24$, although the period is set to one day, the model still performs well because it captures partial periodic features. When $T = 336$ (i.e., twice the true period), the model achieves second-best performance. The essential reason is that when T is an integer multiple of the period, the training samples assigned to each parameter unit of the learnable matrix M still maintain consistent evolutionary patterns. Although the sample distribution density is diluted, the semantic logic of GPC remains intact.

When T does not match the true physical period (e.g., $T = 23$ or 167), a semantically inconsistent GPC is introduced, biasing the guidance of the DC technique and causing a significant drop in accuracy. This confirms that accurate GPC is key to collaborative global-local modeling.

In addition, $T = 1$ exhibits a special state. Although the model loses the ability to capture periodic features, M remains independent in the variable dimension, functionally degenerating into a weakened channel identifier that still provides basic variable identity information, thus outperforming the case of no reference.

Hyperparameter β analysis. This section discusses the impact of the mixing factor β on model forecasting performance. We performed a parameter sweep over $\beta \in \{0.0, 0.1, ..., 0.9, 1.0\}$ on the ETTh1 and ETTm1 datasets, and the results are shown in Figure. 5. Experimental results show that the model has a certain degree of robustness to the value of β, with only slight performance fluctuations at the boundary values. This is mainly because under extreme values, the query matrix Q is driven only by a single local evolution context or global periodic context, causing a slight shift in the representation capability of the retrieval space. Nevertheless, such fluctuations are within an acceptable range, and even under the most unfavorable parameter configuration, the forecasting accuracy of the model is still consistently better than that of existing baseline models.

Adversarial noise analysis. Table 4 presents the comparison results between DCNet and iTransformer on the Electricity dataset when the prediction length is 96 after injecting Gaussian white

Table 4: Performance comparison across SNR levels.

Model	Level	MSE	MAE	ΔMSE%	ΔMAE%
iTransformer	Raw	0.15201	0.24448	-	-
DCNet	Raw	0.13187	0.22680	-	-
iTransformer	SNR=40dB	0.15263	0.24478	0.41	0.12
DCNet	SNR=40dB	0.13188	0.22680	0.01	0.00
iTransformer	SNR=30dB	0.15284	0.24524	0.54	0.31
DCNet	SNR=30dB	0.13203	0.22719	0.12	0.17
iTransformer	SNR=20dB	0.15490	0.24920	1.90	1.93
DCNet	SNR=20dB	0.13357	0.23062	1.29	1.68
iTransformer	SNR=10dB	0.17637	0.28146	16.02	15.13
DCNet	SNR=10dB	0.14895	0.25756	12.95	13.56
iTransformer	SNR=5dB	0.22960	0.34122	51.04	39.57
DCNet	SNR=5dB	0.18588	0.30603	40.95	34.93

noise of different intensities. Noise intensity is measured by SNR, and a smaller SNR indicates stronger noise injection. The results show that under the same noise interference, DCNet not only has lower forecasting error, but also exhibits a smaller degree of performance degradation, indicating that it has stronger robustness to noise.

5 Limitations and future directions

Although the DCNet proposed in this paper demonstrates significant advantages in multiple benchmark tasks, it still has the following limitations:

First, the model has a certain dependence on the inherent periodic characteristics of the data. The core logic of the DC technique lies in using GPC as a guiding baseline for local dynamics. For datasets that lack obvious periodic patterns and exhibit pure random walk or monotonic non-stationary characteristics, the preset periodic reference baseline may fail to provide effective reference information, and may even introduce inappropriate inductive bias, thereby limiting the model's performance in non-periodic scenarios.

Second, the computational overhead in large-scale variable scenarios remains high. Since the model adopts the inverted embedding paradigm and retains the standard attention mechanism, its computational complexity has a quadratic relationship with the variable dimension N ($O(N^2)$). Although it performs excellently on conventional MTSF datasets, when dealing with extremely large-scale scenarios where the variable dimension reaches tens of thousands, the quadratic complexity will bring heavy GPU memory pressure and computational load. Future research may consider introducing sparse attention to further optimize the model's performance in high-dimensional spaces.

6 Conclusion

This paper addresses the problem of isolated modeling of global patterns and local dynamics in MTSF, proposes the DC technique, and constructs the DCNet model based on it. By reconstructing the attention retrieval space, the DC mechanism achieves deep collaboration between GPC and local LEC, enabling local dynamics to be anchored to global coordinates in real time and effectively eliminating perceptual blind spots during the evolution process. At the same time, this technique provides a stable global reference baseline for inter-variable interactions, significantly improving the blind coupling problem of existing inverted architectures in complex environments.

Experimental results show that DCNet achieves leading forecasting accuracy on 12 real-world datasets, and exhibits excellent robustness and computational efficiency. In addition, further transfer experiments prove that the DC technique can not only significantly enhance the performance of models based on the Transformer architecture, but also perform excellently on linear models such as DLinear. In summary, the dual-context collaborative modeling paradigm proposed in this paper provides a robust and efficient new strategy for handling complex non-stationary time series data.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with required disclosure:

Assisting with translation — We used DeepSeek to translate selected passages from non-English literature into English for our literature review. The translated content was subsequently reviewed and verified against the original sources by all authors. We confirm that all interpretations and citations in the manuscript accurately reflect the original literature.

We have not used generative AI tools for generating synthetic datasets, formulating mathematical claims, designing research methodology, implementing methods, analyzing results, or any other tasks requiring mandatory disclosure; these tasks are not applicable to this work.

Additionally, we used generative AI tools for the following tasks with recommended disclosure:

Editing LaTeX formatting — We used DeepSeek to assist in editing the LaTeX formatting of the manuscript, including table alignment, reference formatting, and overall layout adjustments. All formatting edits were reviewed and verified by the authors to ensure compliance with the ICLR 2027 style requirements.

We have reviewed all AI-assisted work. The translated content was checked against original sources by all authors. The LaTeX formatting edits were verified by the first author. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All datasets used in our experiments are publicly available. We follow the standard preprocessing pipelines provided by established benchmarking frameworks, which are widely used in the time series forecasting community. Therefore, no custom data cleaning or splitting procedures were introduced beyond these standard settings.

An anonymized link to our source code is provided in the abstract. The anonymous repository contains the complete implementation, the full hyperparameter configurations for all main experiments, and a reproducible environment image (Docker) that captures the exact software dependencies and library versions used in our runs.

In the appendix, we list the key hyperparameters for the main experiments . The complete set of hyperparameters, including those for baselines and ablation studies, is available in the anonymous code repository. All reported results are averaged over 6 random seeds, and the code to reproduce each table and figure is included in the repository.

REFERENCES

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 6989–6997, 2023.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of machine learning research*, 24(240):1–113, 2023.

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. Periodicity decoupling framework for long-term series forecasting. In *The twelfth international conference on learning representations*, 2024.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

Dongbin Kim, Jinseong Park, Jaewook Lee, and Hoki Kim. Are self-attentions effective for time series forecasting? *Advances in Neural Information Processing Systems*, 37:114180–114209, 2024.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical transactions of the royal society a: mathematical, physical and engineering sciences*, 379(2194), 2021.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. *Advances in Neural Information Processing Systems*, 37:106315–106345, 2024.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. *arXiv preprint arXiv:2505.12917*, 2025a.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and CL Philip Chen. Sparsetsf: Lightweight and robust time series forecasting via sparse modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. Segrnn: Segment recurrent neural network for long-term time series forecasting. *IEEE Internet of Things Journal*, 2025c.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in neural information processing systems*, 35:5816–5828, 2022.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=JePfAI8fah`.

Henrik Madsen. *Time series analysis*. Chapman and Hall/CRC, 2007.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1185–1196, 2025.

Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90:106181, 2020.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 4454–4458, 2022.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024a.

Yulong Wang, Yushuo Liu, Xiaoyi Duan, and Kai Wang. Filterts: Comprehensive frequency filtering for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21375–21383, 2025.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024b.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. *arXiv preprint arXiv:2307.03756*, 2023.

Wang Xue, Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, and Rong Jin. Card: Channel aligned robust blend transformer for time series forecasting. *arXiv preprint arXiv:2305.12095*, 2023.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

Guoqi Yu, Jing Zou, Xiaowei Hu, Angelica I Aviles-Rivero, Jing Qin, and Shujun Wang. Revitalizing multivariate time series forecasting: Learnable decomposition with inter-series dependencies and intra-series variations modeling. *arXiv preprint arXiv:2402.12694*, 2024.

Wenzhen Yue, Yong Liu, Xianghua Ying, Bowei Xing, Ruohao Guo, and Ji Shi. Freeformer: Frequency enhanced transformer for multivariate time series forecasting. *arXiv preprint arXiv:2501.13989*, 2025.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Zhiwei Zhang, Xinyi Du, Xuanchi Guo, Weihao Wang, and Wenjuan Han. Emaformer: Enhancing transformer through embedding armor for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28636–28644, 2026.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

A Related work

This section explains in detail how to construct the GPC using the learnable vector M from CycleNet Lin et al. (2024). For each input sequence $X = \{x_1, \ldots, x_L\} \in R^{L \times N}$, each timestamped data point $x_i \in \mathbb{R}^{1 \times N}$ in the middle is assigned an index during data loading: $i \in [0,\ T-1]$. Through this index, the i-th row is extracted from the periodic component matrix $M \in \mathbb{R}^{T \times N}$.

For each input sequence, the steps for the model to extract features from M are as follows:

1. Determine the starting position: when reading the data, according to the starting time point of the current sampling window in the original sequence, calculate its corresponding starting index within the period T.

2. Extract the feature segment: taking this index as the starting point, continuously extract L rows of features from the parameter matrix M to form the global periodic context $P \in \mathbb{R}^{L \times N}$. If the extraction range exceeds the maximum length T of the matrix, loop back to the first row of the matrix and continue extracting, so as to ensure the periodic continuity of the features in the temporal dimension.

In this design, the hyperparameter T determines the length of the learnable vector and the interval of periodic shifting. Its value should be consistent with the inherent periodicity of the dataset. In practice, an appropriate value can be determined through domain knowledge of the data or calculation of the autocorrelation function (ACF) Madsen (2007). If you are very interested in this content, CycleNet Lin et al. (2024) provides a detailed usage scheme for this technique.

B Experimental details

B.1 Overall algorithm

Algorithm 1 illustrates the workflow of our model. Key points are as follows: N, L, S, D: correspond respectively to the number of variables, lookback length, prediction length, and model hidden dimension.

Algorithm 1 DCNet applied to multivariate time series.

Require: Look back window $\mathbf{X} \in \mathbb{R}^{L \times N}$, Cycle Index τ, Cycle Matrix $\mathbf{M}_{cycle} \in \mathbb{R}^{T \times N}$;

if use_norm: $\mathbf{X} = \text{Instance_Norm}(\mathbf{X})$ ▷ Optional: Normalize $\mathbf{X} \in \mathbb{R}^{L \times N}$
$\mathbf{X} = \mathbf{X}.\text{transpose}$ ▷$\mathbf{X} \in \mathbb{R}^{N \times L}$
$\mathbf{X}_{enc} = \text{Linear}_{data}(\mathbf{X})$ ▷ Get series embedding, $\mathbf{X}_{enc} \in \mathbb{R}^{N \times D}$
$\mathbf{P} = \text{Extract}(\mathbf{M}_{cycle}, \tau, L)$ ▷ Extract cycle segment, $\mathbf{P} \in \mathbb{R}^{N \times L}$
$\mathbf{P}_{enc} = \text{Linear}_{cycle}(\mathbf{P})$ ▷ Get cycle embedding, $\mathbf{P}_{enc} \in \mathbb{R}^{N \times D}$
for $e = 1 \ldots E$ **do**
  $\mathbf{Q}_{in} = \beta \mathbf{X}_{enc} + (1-\beta)\mathbf{P}_{enc}$ ▷ Mixed query input (β is bias)
  $\mathbf{KV}_{in} = \text{Concat}(\mathbf{X}_{enc}, \mathbf{P}_{enc})$ ▷ Concatenated key/value input
  $\mathbf{Score} = \mathbf{QK}^T / \sqrt{d_k}$ ▷ Attention score
  $\mathbf{H} = \text{Softmax}(\mathbf{Score})\mathbf{V} + \mathbf{X}_{enc}$ ▷ Multi-head attention with residual
  $\mathbf{X}_{enc} = \text{SwiGLU}(\mathbf{H}) + \mathbf{H}$ ▷ Gated FFN with residual, $\mathbf{X}_{enc} \in \mathbb{R}^{N \times D}$
end for
$\mathbf{Y} = \text{Linear}_{out}(\mathbf{X}_{enc})$ ▷ Project to predicted length, $\mathbf{Y} \in \mathbb{R}^{N \times S}$
$\mathbf{Y} = \mathbf{Y}.\text{transpose}$ ▷ Final prediction, $\mathbf{Y} \in \mathbb{R}^{S \times N}$
if use_norm: $\mathbf{Y} = \text{De-Instance_Norm}(\mathbf{Y})$ ▷ Optional: De-Normalize $\mathbf{Y}$
return $\mathbf{Y}$

B.2 Additional experimental details

All experiments in this paper are implemented using PyTorch Paszke et al. (2019), trained with the Adam Kingma & Ba (2014) optimizer, and executed on a single NVIDIA GeForce RTX 4070 Ti Super GPU with 16 GB of VRAM. Each task is trained for 30 epochs using an early stopping strategy with a patience of 5. More hyperparameters are shown in Table 5.

Table 5: Hyperparameter settings for different datasets.

Metric	batch_size	learning_rate	e_layers	d_model (D)	d_ff	β
Traffic	16	0.001	2	512	512	1.0
PEMS	32	0.003	2	512	512	0.0
ETT	32	0.0001	1/2	512	512	0.0/0.5/1.0
Weather	32	0.0001	1/2	256/512	512	0.0/0.5/1.0
Electricity	32	0.001	1/2	512	512	0.0
Solar	32	0.0005	1/2	512	512	0.0/0.5

To accurately measure the discrepancy between model predictions and ground truth values during the training process and guide effective parameter optimization, we adopt Mean Squared Error (MSE) as the objective function of the model. For multivariate time series forecasting tasks, the MSE loss function is defined as the mean of the squared differences between the predicted sequence and the actual observations. Its mathematical expression is as follows:

$$\mathcal{L}_{\mathcal{MSE}} = \frac{1}{S \times N} \sum_{i=1}^{S} \sum_{j=1}^{N} (y_{i,j} - \widehat{y_{i,j}})^2 \tag{8}$$

In our experiments, to address the non-stationarity of time series, the model includes an optional instance normalization RevIN Kim et al. (2021) before input, but without learnable affine parameters. $\mathrm{Mean}(\cdot)$ and $\mathrm{Var}(\cdot)$ are calculated over the lookback window L, respectively. The model performs de-normalization at the output end to map the predicted values back to the original scale. For the input sequence X:

$$\hat{X} = \frac{X - \mathrm{Mean}\,(X)}{\sqrt{\mathrm{Var}\,(X) + \epsilon}} \tag{9}$$

Experimental details, source code, and parameter settings are available via our code repository. In addition, we provide dual environment support, including both a directly clonable Conda environment and a detailed dependency list, to ensure the rapid setup of the experimental environment.

C More experimental results

Table 6: The results of the ablation study, with the look-back length fixed at 96. **Red** indicates the best performance, and Blue indicates the second best.

Model		DCNet (Ours)		DCNet W/o DC		DCNet W/o MLP		DCNet W/o Both		iTransformer (2024)		DLinear W DC		DLinear (2023)		CycleNet W DC		CycleNet (2024)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	**0.131**	**0.221**	0.145	0.237	0.134	0.229	0.145	0.238	0.148	0.240	**0.139**	**0.237**	0.197	0.282	**0.135**	**0.228**	0.136	0.229
	192	**0.150**	**0.238**	0.163	0.254	0.152	0.245	0.163	0.256	0.162	0.253	**0.153**	**0.250**	0.196	0.285	**0.151**	**0.244**	0.152	0.244
	336	**0.165**	**0.254**	0.177	0.269	0.168	0.265	0.178	0.273	0.178	0.269	**0.168**	**0.267**	0.209	0.301	**0.168**	**0.263**	0.170	0.264
	720	**0.192**	**0.279**	0.207	0.298	0.221	0.311	0.228	0.312	0.225	0.317	**0.201**	**0.298**	0.245	0.333	**0.206**	**0.296**	0.212	0.299
	avg	**0.159**	**0.248**	0.173	0.265	0.169	0.263	0.179	0.270	0.178	0.270	**0.165**	**0.263**	0.212	0.300	**0.165**	**0.258**	0.168	0.259
PEMS_03	12	**0.056**	**0.155**	0.064	0.166	0.058	0.159	0.064	0.167	0.071	0.174	**0.072**	**0.180**	0.122	0.243	**0.061**	**0.164**	0.066	0.172
	24	**0.068**	**0.171**	0.081	0.188	0.069	0.174	0.082	0.190	0.093	0.201	**0.104**	**0.214**	0.201	0.317	**0.078**	**0.185**	0.089	0.201
	48	**0.089**	**0.196**	0.114	0.225	0.091	0.198	0.114	0.225	0.125	0.236	**0.153**	**0.259**	0.333	0.425	**0.113**	**0.219**	0.136	0.247
	96	**0.126**	**0.229**	0.154	0.265	0.136	0.235	0.162	0.276	0.164	0.275	**0.198**	**0.296**	0.457	0.515	**0.151**	**0.250**	0.182	0.282
	avg	**0.084**	**0.187**	0.103	0.211	0.089	0.192	0.106	0.215	0.113	0.222	**0.132**	**0.237**	0.278	0.375	**0.101**	**0.205**	0.118	0.226
PEMS_04	12	**0.063**	**0.161**	0.073	0.176	0.065	0.165	0.074	0.176	0.078	0.183	**0.081**	**0.189**	0.148	0.272	**0.070**	**0.173**	0.078	0.186
	24	**0.072**	**0.171**	0.089	0.195	0.074	0.175	0.089	0.195	0.095	0.205	**0.107**	**0.219**	0.224	0.340	**0.084**	**0.192**	0.099	0.212
	48	**0.083**	**0.185**	0.115	0.223	0.086	0.196	0.117	0.227	0.120	0.233	**0.149**	**0.260**	0.355	0.437	**0.106**	**0.216**	0.133	0.248
	96	**0.097**	**0.201**	0.145	0.252	0.104	0.209	0.147	0.256	0.150	0.262	**0.182**	**0.291**	0.452	0.504	**0.139**	**0.249**	0.167	0.281
	avg	**0.078**	**0.179**	0.106	0.212	0.082	0.186	0.107	0.214	0.111	0.221	**0.130**	**0.240**	0.295	0.388	**0.100**	**0.208**	0.119	0.232
PEMS_07	12	**0.050**	**0.140**	0.056	0.148	0.052	0.143	0.057	0.151	0.067	0.165	**0.066**	**0.169**	0.115	0.242	**0.056**	**0.152**	0.062	0.162
	24	**0.058**	**0.150**	0.069	0.166	0.059	0.152	0.073	0.173	0.088	0.190	**0.096**	**0.204**	0.210	0.329	**0.072**	**0.173**	0.086	0.192
	48	**0.070**	**0.163**	0.091	0.189	0.073	0.168	0.097	0.201	0.110	0.215	**0.146**	**0.248**	0.398	0.458	**0.103**	**0.204**	0.128	0.234
	96	**0.087**	**0.180**	0.113	0.204	0.091	0.187	0.123	0.227	0.215	0.245	**0.196**	**0.284**	0.594	0.553	**0.143**	**0.236**	0.176	0.268
	avg	**0.066**	**0.158**	0.082	0.177	0.069	0.163	0.088	0.188	0.120	0.204	**0.126**	**0.226**	0.329	0.396	**0.094**	**0.191**	0.113	0.214
PEMS_08	12	**0.067**	**0.162**	0.077	0.171	0.069	0.165	0.078	0.174	0.079	0.182	**0.090**	**0.191**	0.154	0.276	**0.072**	**0.173**	0.082	0.185
	24	**0.083**	**0.174**	0.105	0.197	0.083	0.174	0.109	0.202	0.115	0.219	**0.131**	**0.229**	0.248	0.353	**0.097**	**0.197**	0.117	0.226
	48	**0.109**	**0.191**	0.153	0.231	0.115	0.198	0.154	0.238	0.186	0.235	**0.203**	**0.278**	0.440	0.470	**0.142**	**0.232**	0.169	0.268
	96	**0.138**	**0.205**	0.181	0.256	0.147	0.216	0.188	0.263	0.221	0.267	**0.279**	**0.325**	0.674	0.565	**0.198**	**0.266**	0.233	0.306
	avg	**0.099**	**0.183**	0.129	0.214	0.104	0.188	0.132	0.219	0.150	0.226	**0.176**	**0.256**	0.379	0.416	**0.127**	**0.217**	0.150	0.246

Table 7: Full results of the efficiency analysis in Figure 4(a), where MACs is a metric for measuring inference speed, we generally consider 1 MACs = 2 FLOPs.

Model	Parameters	MACs	Training time (s)
Informer	12.53M	3.97G	78.1s
Autoformer	12.22M	4.41G	113.7s
Dlinear	139.62K	44.91M	22.8s
PatchTST	10.74M	25.87G	249.1s
iTransformer	5.15M	1.65G	36.0s
CycleNet	472.93K	134.84M	22.3s
TQNet	966.96K	309.65M	23.3s
FreeEformer	11.37M	3.04G	43.2s
EMAformer	10.45M	3.35G	70.59s
DCNet	2.30M	0.90G	24.4s

Table 8: Comparison of MTSF results across 12 real-world datasets. The look-back window L is fixed at 96. **Red-bold** indicates the best performance, while blue-underlined represents the second best.

Model		DCNet (Ours)		EMAformer (2026)		FreEformer (2025)		TQNet (2025a)		DUET (2025)		FilterTS (2025)		CycleNet (2024)		TimeXer (2024b)		Leddam (2024)		iTransformer (2024)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	0.131	0.221	**0.130**	**0.220**	0.133	0.223	0.134	0.229	0.145	0.233	0.151	0.245	0.136	0.229	0.140	0.242	0.141	0.235	0.148	0.240
	192	**0.150**	**0.238**	**0.150**	**0.238**	0.152	0.240	0.154	0.247	0.163	0.248	0.163	0.256	0.152	0.244	0.157	0.256	0.159	0.252	0.162	0.253
	336	**0.165**	**0.254**	0.166	0.255	**0.165**	0.256	0.169	0.264	0.175	0.262	0.180	0.274	0.170	0.264	0.176	0.275	0.173	0.268	0.178	0.269
	720	0.192	0.279	**0.187**	**0.274**	0.198	0.286	0.201	0.294	0.204	0.291	0.224	0.311	0.212	0.299	0.211	0.306	0.201	0.295	0.225	0.317
	avg	0.159	0.248	**0.158**	**0.247**	0.162	0.251	0.165	0.259	0.172	0.259	0.180	0.272	0.168	0.259	0.171	0.270	0.169	0.263	0.178	0.270
ETTh1	96	**0.366**	0.392	0.374	**0.390**	0.371	0.392	0.371	0.393	0.377	0.393	0.374	0.397	0.375	0.395	0.382	0.403	0.377	0.394	0.386	0.405
	192	**0.413**	0.421	0.428	**0.419**	0.424	0.422	0.428	0.426	0.429	0.425	0.424	0.429	0.436	0.428	0.429	0.435	0.424	0.422	0.442	0.428
	336	**0.459**	0.441	0.469	**0.439**	0.466	0.443	0.476	0.446	0.471	0.446	0.464	0.451	0.496	0.455	0.468	0.448	**0.459**	0.442	0.487	0.465
	720	0.459	0.464	**0.456**	**0.450**	0.471	0.477	0.487	0.470	0.496	0.480	0.470	0.466	0.520	0.484	0.469	0.461	0.463	0.461	0.503	0.490
	avg	**0.424**	0.429	0.432	**0.425**	0.433	0.434	0.441	0.434	0.443	0.436	0.433	0.436	0.457	0.441	0.437	0.437	0.431	0.430	0.455	0.447
ETTh2	96	0.291	0.343	0.290	**0.335**	**0.286**	**0.335**	0.295	0.343	0.296	0.345	0.290	0.338	0.298	0.344	**0.286**	0.338	0.293	0.343	0.297	0.349
	192	0.367	0.390	0.367	0.385	**0.363**	**0.381**	0.367	0.393	0.368	0.389	0.374	0.390	0.372	0.396	0.366	0.389	0.367	0.389	0.380	0.400
	336	**0.408**	**0.419**	0.414	0.422	0.416	0.420	0.417	0.427	0.411	0.422	**0.408**	0.420	0.431	0.439	0.414	0.423	0.412	0.424	0.428	0.432
	720	0.413	**0.426**	0.424	0.439	0.422	0.436	0.433	0.446	0.412	0.434	0.418	0.437	0.450	0.458	**0.408**	0.432	0.419	0.438	0.427	0.445
	avg	0.369	0.394	0.374	0.395	0.372	**0.393**	0.378	0.402	0.372	0.398	0.373	0.396	0.388	0.409	**0.369**	0.396	0.373	0.399	0.383	0.407
ETTm1	96	0.306	0.349	**0.304**	**0.346**	0.306	0.349	0.311	0.353	0.324	0.354	0.321	0.360	0.319	0.360	0.318	0.356	0.319	0.359	0.334	0.368
	192	**0.355**	0.378	0.359	**0.375**	0.359	0.377	0.357	0.378	0.369	0.379	0.363	0.382	0.360	0.381	0.362	0.383	0.369	0.383	0.377	0.391
	336	**0.385**	**0.398**	0.393	**0.398**	0.392	0.401	0.390	0.399	0.404	0.402	0.395	0.403	0.389	0.403	0.395	0.407	0.394	0.402	0.426	0.420
	720	**0.440**	**0.435**	0.467	0.451	0.458	0.445	0.452	0.438	0.463	0.437	0.462	0.438	0.447	0.441	0.452	0.441	0.460	0.442	0.491	0.459
	avg	**0.371**	**0.390**	0.381	0.393	0.379	0.393	0.378	0.392	0.390	0.393	0.385	0.396	0.379	0.396	0.382	0.397	0.386	0.397	0.407	0.410
ETTm2	96	0.168	0.247	0.167	**0.246**	0.169	0.248	0.173	0.256	0.174	0.255	0.172	0.255	**0.163**	**0.246**	0.171	0.256	0.176	0.257	0.180	0.264
	192	0.232	**0.289**	0.233	0.290	0.233	**0.289**	0.242	0.300	0.243	0.302	0.237	0.299	**0.229**	0.290	0.237	0.299	0.243	0.303	0.250	0.309
	336	0.291	**0.327**	0.293	0.329	0.292	0.328	0.301	0.340	0.304	0.341	0.299	0.398	**0.284**	**0.327**	0.296	0.338	0.303	0.341	0.311	0.348
	720	**0.389**	0.389	0.390	**0.387**	0.396	0.389	0.397	0.396	0.399	0.397	0.397	0.394	**0.389**	0.391	0.392	0.394	0.400	0.398	0.412	0.407
	avg	0.270	**0.313**	0.271	**0.313**	0.273	0.314	0.278	0.323	0.280	0.324	0.276	0.337	**0.266**	0.314	0.274	0.322	0.281	0.325	0.288	0.332
Solar	96	**0.165**	0.231	0.169	0.200	0.180	**0.191**	0.173	0.233	0.200	0.233	0.221	0.269	0.190	0.247	0.215	0.295	0.197	0.241	0.203	0.237
	192	**0.189**	0.250	**0.189**	0.220	0.213	**0.215**	0.199	0.257	0.228	0.254	0.255	0.290	0.210	0.266	0.236	0.301	0.231	0.264	0.233	0.261
	336	**0.198**	0.256	0.224	**0.232**	0.233	**0.232**	0.208	0.263	0.262	0.298	0.280	0.304	0.217	0.266	0.252	0.307	0.241	0.268	0.248	0.273
	720	**0.205**	0.261	0.208	0.243	0.241	**0.237**	0.209	0.270	0.258	0.287	0.278	0.303	0.223	0.266	0.244	0.305	0.250	0.281	0.249	0.275
	avg	**0.189**	0.249	0.198	0.224	0.217	**0.219**	0.197	0.256	0.237	0.268	0.259	0.292	0.210	0.261	0.237	0.302	0.230	0.264	0.233	0.262
Weather	96	**0.153**	**0.189**	**0.153**	0.190	**0.153**	**0.189**	0.157	0.200	0.163	0.202	0.162	0.207	0.158	0.203	0.157	0.205	0.156	0.202	0.174	0.214
	192	**0.199**	**0.234**	0.204	0.239	0.201	0.238	0.206	0.245	0.218	0.252	0.209	0.252	0.207	0.247	0.204	0.247	0.207	0.250	0.221	0.254
	336	**0.255**	**0.277**	0.261	0.282	0.261	0.282	0.262	0.287	0.274	0.294	0.263	0.292	0.262	0.289	0.261	0.290	0.262	0.291	0.278	0.296
	720	**0.329**	**0.324**	0.343	0.336	0.341	0.334	0.344	0.342	0.349	0.343	0.344	0.344	0.344	0.344	0.340	0.341	0.343	0.343	0.358	0.347
	avg	**0.234**	**0.256**	0.240	0.262	0.239	0.260	0.242	0.269	0.251	0.273	0.245	0.274	0.243	0.271	0.241	0.271	0.242	0.272	0.258	0.278
Traffic	96	**0.384**	0.257	0.397	0.238	0.395	**0.233**	0.413	0.261	0.407	0.262	0.448	0.309	0.458	0.296	0.428	0.271	0.426	0.276	0.395	0.268
	192	**0.407**	0.269	0.418	0.249	0.423	**0.245**	0.432	0.271	0.431	0.272	0.455	0.307	0.457	0.294	0.448	0.282	0.458	0.289	0.417	0.276
	336	**0.420**	0.274	0.435	0.256	0.443	**0.254**	0.450	0.277	0.456	0.279	0.472	0.313	0.470	0.299	0.473	0.289	0.486	0.297	0.433	0.283
	720	**0.450**	0.291	0.472	0.276	0.480	**0.274**	0.486	0.295	0.509	0.302	0.508	0.332	0.502	0.314	0.516	0.307	0.498	0.313	0.467	0.302
	avg	**0.415**	0.272	0.431	0.255	0.435	**0.252**	0.445	0.276	0.451	0.279	0.471	0.315	0.472	0.301	0.466	0.287	0.467	0.294	0.428	0.282
PEMS_03	12	**0.056**	**0.155**	0.058	0.157	0.060	0.160	0.060	0.161	0.064	0.166	0.076	0.186	0.066	0.172	0.070	0.173	0.063	0.164	0.071	0.174
	24	**0.068**	**0.171**	0.076	0.176	0.077	0.181	0.077	0.182	0.081	0.186	0.111	0.224	0.089	0.201	0.092	0.194	0.080	0.185	0.093	0.201
	48	**0.089**	**0.196**	0.095	0.198	0.112	0.218	0.104	0.215	0.114	0.222	0.200	0.306	0.136	0.247	0.129	0.229	0.124	0.226	0.125	0.236
	96	**0.126**	**0.229**	0.130	**0.229**	0.159	0.265	0.148	0.253	0.161	0.266	0.333	0.401	0.182	0.282	0.157	0.261	0.160	0.275	0.164	0.275
	avg	**0.084**	**0.187**	0.090	0.190	0.102	0.206	0.097	0.203	0.105	0.210	0.180	0.279	0.118	0.226	0.112	0.214	0.107	0.210	0.113	0.222
PEMS_04	12	**0.063**	**0.161**	0.064	**0.161**	0.068	0.164	0.067	0.166	0.079	0.181	0.094	0.205	0.078	0.186	0.074	0.178	0.071	0.172	0.078	0.183
	24	**0.072**	**0.171**	**0.072**	0.172	0.079	0.179	0.077	0.181	0.096	0.203	0.136	0.253	0.099	0.212	0.087	0.195	0.087	0.193	0.095	0.205
	48	**0.083**	**0.185**	0.088	0.186	0.099	0.204	0.097	0.206	0.114	0.226	0.226	0.332	0.133	0.248	0.110	0.214	0.113	0.222	0.120	0.233
	96	**0.097**	**0.201**	0.103	0.205	0.129	0.238	0.123	0.233	0.148	0.253	0.378	0.442	0.167	0.281	0.148	0.251	0.141	0.252	0.150	0.262
	avg	**0.078**	**0.179**	0.081	0.181	0.094	0.196	0.091	0.197	0.109	0.216	0.209	0.308	0.119	0.232	0.105	0.210	0.103	0.210	0.111	0.221
PEMS_07	12	**0.050**	0.140	0.051	0.139	0.053	**0.138**	0.051	0.143	0.226	0.156	0.071	0.176	0.062	0.162	0.057	0.152	0.055	0.145	0.067	0.165
	24	**0.058**	0.150	0.060	**0.149**	0.067	0.153	0.063	0.159	0.073	0.172	0.103	0.214	0.086	0.192	0.079	0.179	0.070	0.164	0.088	0.190
	48	**0.070**	**0.163**	0.078	0.165	0.087	0.174	0.081	0.179	0.096	0.201	0.173	0.284	0.128	0.234	0.099	0.191	0.094	0.192	0.110	0.215
	96	**0.087**	**0.180**	0.104	0.194	0.115	0.205	0.103	0.203	0.104	0.202	0.296	0.378	0.176	0.268	0.107	0.205	0.117	0.217	0.215	0.245
	avg	**0.066**	**0.158**	0.073	0.162	0.081	0.168	0.075	0.171	0.125	0.183	0.161	0.263	0.113	0.214	0.086	0.182	0.084	0.180	0.120	0.204
PEMS_08	12	0.067	0.162	0.072	0.170	**0.066**	**0.157**	0.071	0.170	0.072	0.168	0.092	0.201	0.082	0.185	0.075	0.176	0.071	0.171	0.079	0.182
	24	**0.083**	**0.174**	0.099	0.199	0.085	0.176	0.096	0.196	0.093	0.191	0.144	0.254	0.117	0.226	0.102	0.201	0.091	0.189	0.115	0.219
	48	**0.109**	**0.191**	0.127	0.212	0.119	0.207	0.149	0.244	0.123	0.207	0.267	0.350	0.169	0.268	0.158	0.248	0.128	0.219	0.186	0.235
	96	**0.138**	**0.205**	0.212	0.254	0.169	0.238	0.253	0.309	0.211	0.253	0.513	0.476	0.233	0.306	0.366	0.377	0.198	0.266	0.221	0.267
	avg	**0.099**	**0.183**	0.128	0.209	0.110	0.195	0.142	0.230	0.125	0.205	0.254	0.320	0.150	0.246	0.175	0.251	0.122	0.211	0.150	0.226