AMCRNet: Adaptive Multi-Cycle Residual Modeling for Long-Term Time Series Forecasting

Jiannan Wan[1], Fei Chen[1], Xunxun Zeng[2*], Hang Cheng[2], Meiqing Wang[2]
[1]College of Computer and Data Science, Fuzhou University, Fuzhou, China
[2]School of Mathematics and Statistics, Fuzhou University, Fuzhou, China
Email: wanjiannan202@163.com, chenfei314@fzu.edu.cn, xunxun@fzu.edu.cn, hcheng@fzu.edu.cn, mqwang@fzu.edu.cn

***Abstract*—Time-series forecasting is fundamental to many real-world applications such as weather and transportation. In this paper, we propose AMCRNet, a long-term forecasting framework that integrates explicit multi-cycle modeling with correlation-aware residual learning in an end-to-end manner. Specifically, AMCRNet maintains multiple learnable cyclic prototypes associated with different cycle lengths. Periodic components are extracted via cyclic indexing, which enables direct and interpretable modeling of stable periodic patterns. A sample-wise gating fusion is applied to adaptively assign the explanatory contribution of each cycle to the input sequence. Meanwhile, the same cycle weights are reused to construct a consistent future-cycle compensation term for the forecasting horizon. This design prevents periodic information loss after decomposition. On the residual space, AMCRNet adopts a correlation-aware channel modeling to capture inter-variable relationships. A learnable fusion coefficient is introduced to adaptively balance the residual shortcut and correlation-enhanced representations. Extensive experiments on multiple benchmark datasets demonstrate that AMCRNet consistently achieves strong long-term forecasting performance with improved robustness under composite periodicities and non-stationary perturbations.**

***Index Terms*—multivariate time series forecasting, attention, periodic patterns, residual prediction.**

I. Introduction

Time series forecasting is critical in fields such as transportation, energy [1], weather [2] and economic [3]. In particular, long-term time series forecasting(LTSF) [4] [5] has attracted increasing attention. It requires models to simultaneously capture short-term abrupt patterns, long-term temporal dependencies, and inter-variable relationships in multivariate sequences.

Real-world time series data often exhibit multiple periodic patterns with different frequencies and amplitudes, which may further interact through superposition and modulation. For instance, in traffic flow [6], the daily cycle is mainly characterized by morning and evening commuting peaks and nighttime troughs. The weekly cycle arises from heterogeneous travel purposes between weekdays and weekends and leads to systematic variations in overall traffic levels and peak profiles within a week. Moreover, seasonal and holiday effects (e.g., summer vacations, public holidays, and adverse weather such as rain or snow) modulate both travel demand and roadway capacity, further altering the amplitude and duration of intra-day peaks and valleys. The interaction across these multi-scale periodicities yields complex periodic structures in traffic data. Therefore, it is necessary to model multiple cycles to improve forecast accuracy.

* Corresponding author

Although recent long-term forecasting methods [7] [8] [9]have achieved encouraging progress in extracting latent long-term patterns, most of them either rely on implicit decomposition or focus on a single dominant periodic structure. As a result, fine-grained multi-cycle interactions are often underexplored. Furthermore, many existing approaches [10] [11] [12] treat residual components independently across variables, which limits their ability to fully exploit inter-variable correlations under non-stationary.

Inspired by CycleNet [13] and TQNet [14], we propose AMCRNet, an adaptive multi-cycle residual modeling framework for long-term time series forecasting. By constructing a long-term forecasting framework, AMCRNet unifies explicit multi-cycle modeling, correlation-aware residual forecasting, and patch-based representation learning within an end-to-end training. We achieve good forecasting accuracy while enhancing robustness against non-stationarity and noise perturbations.

Our contributions can be summarized as follows:

- Adaptive multi-cycle decomposition module is proposed to explicitly model multi-scale periodicities via multiple learnable cyclic prototypes and a sample-wise gating fusion, together with a consistent future-cycle compensation for the forecasting horizon.
- Unified forecasting framework is proposed to integrate explicit modeling with correlation-aware residual learning and patch-based representation modeling in an end-to-end training.
- Comprehensive experiments are conducted on multiple benchmark datasets, and the results demonstrate that AMCRNet consistently achieves stable and competitive performance on long-term forecasting tasks.

II. Related Works

A. Transformer-based Models

With the aid of the self-attention [15], transformer-based models have demonstrated strong capability in capturing long-term temporal dependencies and entangled multivariate correlations. FEDformer [16] enhances the series decomposition block with multiple kernels moving average. Pyraformer [17] constructs a multiresolution tree and develops a pyramidal attention mechanism, in which every node can only attend to its neighboring, adjacent, and children nodes. These models effectively capture long-range dependencies in time series data through self-attention mechanisms, which is crucial for LTSF tasks. However, despite their strong performance, these models typically incur high computational costs, especially when dealing with long-term time series data. Recent studies have shown that competitive results can also be achieved by processing the data in series-wise approaches. For example, TimeXer [18] focuses on forecasting with exogenous variables by separately leveraging patch-level and series-level representations of endogenous and exogenous variables. TQNet [14] introduces a hybrid attention mechanism, where the Query is instantiated by globally shared learnable parameters with periodic shift to characterize global correlations, while the Key is generated from the raw local input to preserve sample-specific dependencies. Although this type of model shows impressive performance, it still suffers from insufficient robustness in forecasting accuracy.

B. Multi-layer perceptron (MLP) based Models

The study of DLinear [4] demonstrates that simple linear models can achieve favorable forecast performance with high computational efficiency. Subsequently, further research has investigated and validated the effectiveness of linear models for time series forecasting. SparseTSF [11] adopts Cross-Period Sparse Forecasting to decouple the periodicity and trend components of time series data. This approach downsamples the original sequence with a period and focuses on trend forecasting between periods. It effectively extracts periodic characteristics while minimizing both the complexity of the model and the number of parameters. TSMixer [12] contains interleaving time-mixing and feature-mixing MLPs to extract information from different perspectives. FreTS [19] investigates the learned patterns of frequency-domain MLPs, which operate on both inter-series and intra-series scales to capture channelwise and time-wise dependencies in multivariate data.

III. METHODOLOGY

A. Overall Architecture

Time series forecasting is to predict future horizons H steps ahead based on past L observations, mathematically represented as $f : \mathbf{X} = \{X_1^t, \ldots, X_C^t\}_{t=1}^{L} \in \mathbb{R}^{L\times C} \rightarrow \hat{\mathbf{X}} = \{\hat{X}_1^t, \ldots, \hat{X}_C^t\}_{t=L+1}^{L+H} \in \mathbb{R}^{H\times C}$, where C is the number of variables and X_i^t is the value of the i_{th}variate at the t_{th} time step. Our goal is to approximate the ground truth as closely as possible with predictions $\hat{\mathbf{X}}$.

The architecture of the proposed AMCRNet mainly consists of four components: Adaptive Multi-Cycle Residual decomposition (AMCRD), CQ Channel Aggregation (CQCA), Patch Embedding and Positional Embedding, and Forecasting Head and Future-Cycle Compensation, as illustrated in Fig. 1.

We employ RevIN [20] for both preprocessing and postprocessing steps to mitigate the adverse impact of non-stationarity on training stability.

B. Normalization and De-Normalization

Real-world time series often exhibit pronounced [21] and distribution shift, which can lead to substantial performance degradation when models trained on historical observations are deployed for future forecasting. To stabilize the subsequent periodic modeling and channel-correlation learning, we adopt a parameter-free RevIN [20] module to normalize each input instance, thereby enforcing a more consistent input distribution for the backbone network, formulated as follows.

$$\tilde{X}_i^t = \frac{X_i^t - \mu}{\sqrt{\sigma}}, \tag{1}$$

$$\hat{X}_i^t = \overline{X}_i^t \sqrt{\sigma + \epsilon} + \mu, \tag{2}$$

where μ and σ represent the mean and standard deviation of the $\mathbf{X}_i$, respectively, and ϵ is a small constant for numerical stability. $\overline{X}^t$represents the prediction output before the denormalization step.

C. Adaptive Multi-Cycle Residual Decomposition

Inspired by CycleNet [13], we employ multiple learnable recurrent cycles to decompose the superposed periodicities present in the data as shown in Fig. 1(a). Compared with explicit Fourier-based decomposition, our approach uses learnable periodic prototype tables to directly fit diverse periodic patterns. It allows the model to adaptively absorb stable periodic components during training. Moreover, parameter reuse is achieved via modulo indexing, which makes the learned periodic modes more robust when shared across samples.

Given a multivariate time series with C channels, we specify N distinct cycle length set $\mathcal{W} = \{W_1, W_2, \ldots, W_N\}$. For each W_i, we construct a learnable recurrent cycle $\mathbf{O}_i \in \mathbb{R}^{W_i\times C}$, which is initialized to zero. These recurrent cycles are intended to model distinct periodic components within the same sequence (e.g., hourly, daily, weekly, or monthly patterns). Since the channels are treated independently, each recurrent cycle can be cyclically repeated to produce a periodic component $\mathbf{S}_i \in \{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_N\}$, whose length matches that of the input sequence. For a given periodic component, $\mathbf{O}_i^t$ denotes the cyclic segment extracted from the learnable prototype $\mathbf{O}_i$ starting at position t mod W_i, where t is the index provided during data preprocessing. We then cyclically repeat $\mathbf{O}_i^t$ to construct the corresponding periodic sequence $\mathbf{S}_i \in \mathbb{R}^{L\times C}$, as formulated below:

$$\mathbf{S}_i = Concat(\underbrace{\mathbf{O}_i^t, \ldots, \mathbf{O}_i^t}_{T_L}, \mathbf{O}_i^{t'}), \tag{3}$$

where $T_L = \lfloor \frac{L}{W_i} \rfloor$ denotes the number of $\mathbf{O}_i^t$, and $\mathbf{O}_i^{t'}$ denotes the subsequence extracted from $\mathbf{O}_i^t$, starting at index 0 with a length of L mod W_i. In particular, these cycles $\mathcal{O} = \{\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_N\}$ are optimized jointly with the backbone via gradient-based end-to-end training. These learned informative representations reveal the underlying multi-periodic structure of the data.

For a periodic sequence $\mathbf{S}_i$, to estimate each component that contributes to the original input across different channels,

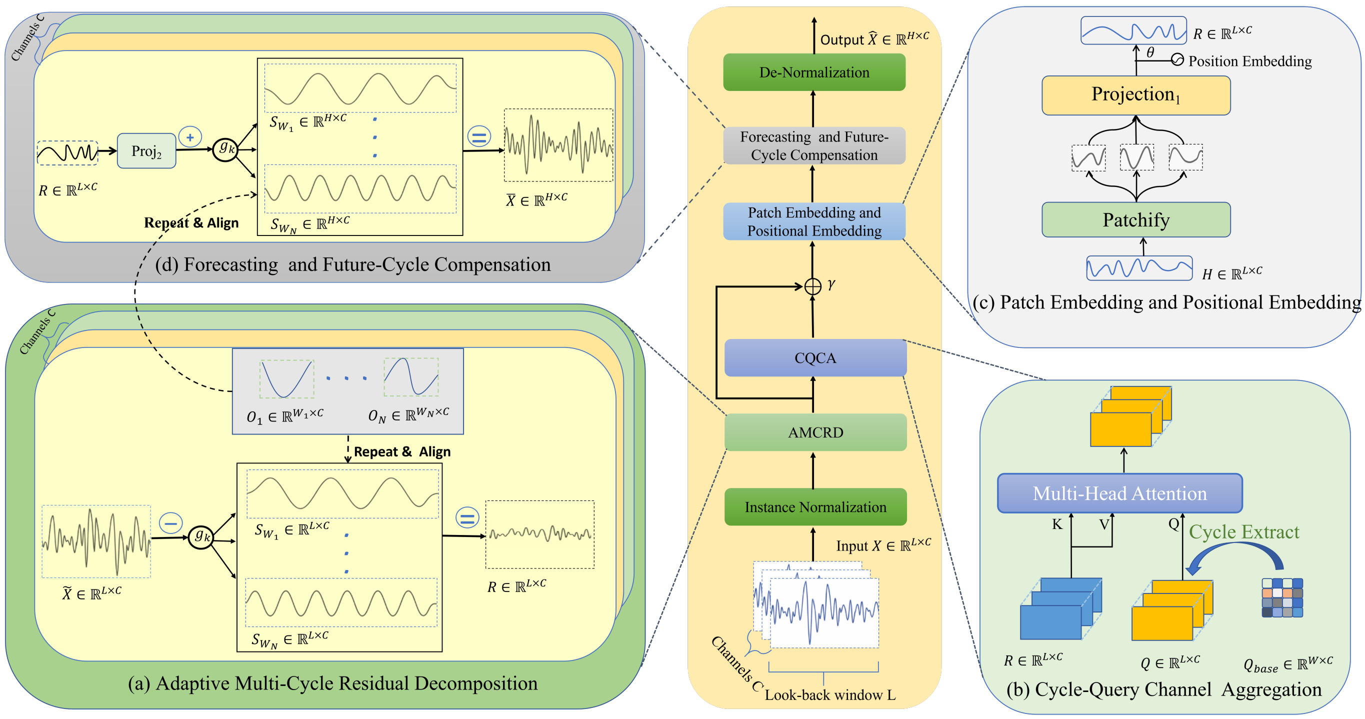


Fig. 1: The overall architecture of AMCRNet. (a) AMCRD removes multi-periodic components; (b) CQCA learns inter-channel correlations; (c) The Patch Embedding module captures short-term temporal dependencies; (d) The Forecasting Head and Future-Cycle Compensation module produces the residual forecasts and compensates for the removed periodic components.

we introduce a gating-based fusion mechanism to adaptively assign the "explanatory power" of each cycle at the sample level. At each time step, we compute a scalar gating signal by performing channel-weighted pooling over the normalized input $\tilde{\mathbf{X}}$ and have:

$$\mathbf{w} = softmax(\frac{\mathbf{E}}{\tau}), \tag{4}$$

where $\mathbf{E} \in \mathbb{R}^C$ is a learnable channel-importance vector. The $softmax$ normalized weights $\mathbf{w} \in \mathbb{R}^C$ determine the relative contribution of each channel, while τ controls the sharpness of the channel selection.

$$u^t = \sum_{n=1}^{C} \tilde{X}_n^t w_n, \tag{5}$$

where u^t summarizes the multichannel information at time t into a single scalar. We then concatenate $\{u^t\}_{t=1}^{L}$ along the temporal dimension to obtain the vector $\mathbf{u} \in \mathbb{R}^L$:

$$\mathbf{u} = [u^1, u^2, \dots, u^L]. \tag{6}$$

To obtain the cycle weights $\mathbf{g} \in \mathbb{R}^N$, we adopt a one-layer fully connected neural network $Linear(\cdot)$ to produce cycle-selection logits, and have:

$$\mathbf{g} = Softmax(Linear(\mathbf{u})). \tag{7}$$

By $Softmax(\cdot)$ normalization, the weighted periodic sequences can be written as:

$$\mathbf{S}_{fusion} = \sum_{i=1}^{N} g_i \cdot \mathbf{S}_i, \tag{8}$$

where g_i quantifies sample-wise contribution of the i-th periodic component $\mathbf{S}_i$ in the fused periodic baseline $\mathbf{S}_{fusion} \in \mathbb{R}^{L\times C}$. Subsequently, this fused periodic component is removed from the original sequence to obtain the residual components $\mathbf{R}$:

$$\mathbf{R} = \mathbf{X} - \mathbf{S}_{fusion}. \tag{9}$$

D. CQ Channel Aggregation

After subtracting the periodic components, we proceed to model the residual term. As shown in Fig. 1(b), this module is designed to learn inter-channel dependencies within the residual sequence while simultaneously accounting for both global consistency and local, sample-specific characteristics, as done in TQNet [14]. Specifically, we first define a periodically shifted learnable query matrix $\mathbf{Q}_{base} \in \mathbb{R}^{W\times C}$, which is initialized to zeros. Given the index t, we cyclically retrieve the corresponding segment from $\mathbf{Q}_{base}$ and assemble it as the query for the attention mechanism. W is the predefined cycle length, and $\mathbf{Q}$ is constructed by cyclically sampling from $\mathbf{Q}_{base}$.

Next, we apply a multi-head attention mechanism, where $\mathbf{Q}$ serves as the query and the residual sequence $\mathbf{R}$ is used as the key and value. By jointly leveraging the cross-sample shared global prior encoded in $\mathbf{Q}$ and the local, sample-specific input, this design effectively mitigates the adverse impact of local noise perturbations. We have:

$$\mathbf{H}_j = Softmax(\frac{\mathbf{Q}_j \mathbf{K}_j^{\mathsf{T}}}{\sqrt{L}})\mathbf{V}_j. \tag{10}$$

Here, $\mathbf{Q}_j = \mathbf{Q}\mathbf{W}_j^Q, \mathbf{K}_j = \mathbf{R}\mathbf{W}_j^K, \mathbf{V}_j = \mathbf{R}\mathbf{W}_j^V$. After concatenating the outputs of all attention heads in the MHA module, we obtain the channel-fused feature representation:

$$\mathbf{H} = MHA(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Concat(\mathbf{H}_1, \dots, \mathbf{H}_h)\mathbf{W}^O, \tag{11}$$

where $MHA(\cdot)$ denotes the multi-head attention mechanism. $\mathbf{W}^O$ is the learnable output projection matrix that linearly maps the concatenated multi-head features back to the target embedding dimension. It helps head-wise information fusion into a unified representation.

To prevent overly strong attention fusion from causing information over-smoothing while fully preserving sample-specific cues, we introduce a learnable parameter γ to adaptively balance the channel-fused features and the original residual information. The fused representation $\mathbf{Z} \in \mathbb{R}^{L \times C}$ is given as:

$$\mathbf{Z} = \gamma \cdot \mathbf{R} + (1 - \gamma) \cdot \mathbf{H}. \tag{12}$$

E. Patch Embedding and Positional Embedding

Inspired by Vision Transformer [22], we adopt patch embedding to capture short-term temporal dependencies and mitigate overfitting. As shown in Fig. 1(c), we apply patch embedding to $\mathbf{Z}$ by partitioning it into M non-overlapping patches $\{\mathbf{P}_i\}_{i=1}^M \in \mathbb{R}^{P \times D}$ of length P, where $M = \lceil L/P \rceil$.

The position embedding is given as follows:

$$\mathbf{U} = \theta \cdot Proj_1(\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_M) + (1 - \theta) \cdot \mathbf{B}. \tag{13}$$

Here, $proj_1(\cdot)$ is implemented as a patch-wise MLP projector, where each patch is projected by a two-layer feed-forward neural network with interleaved LayerNorm and ReLU activations. The learnable coefficient θ is then used to adaptively fuse the resulting content representations with the positional information $\mathbf{B} \in \mathbb{R}^{M \times D}$ along the time axis.

F. Forecasting Head and Future-Cycle Compensation

Given $\mathbf{U}$, we concatenate the M patch embeddings along the patch axis for each variate and feed the resulting vector into $Proj_2(\cdot)$ to generate the residual forecasts, as shown in Fig. 1(d). We have:

$$\mathbf{R} = Proj_2(\mathbf{U}). \tag{14}$$

Here, $Proj_2(\cdot)$ is implemented as a patch-to-horizon projection head. We first flatten the patch dimension for each variate to obtain a compact representation. A two-layer MLP then maps this vector to the H-step residual forecast, with LayerNorm and ReLU to stabilize optimization and enhance nonlinearity.

To obtain the periodic component for the forecasting horizon, we start the indexing from $t+L \bmod W_i$ in $\mathbf{O}_i$ to retrieve $\mathbf{O}_i^{(t+L)}$, and then cyclically sample from it to construct the predicted periodic sequence $\hat{\mathbf{S}}_i$:

$$\hat{\mathbf{S}}_i = Concat(\underbrace{\mathbf{O}_i^{(t+L)}, \dots, \mathbf{O}_i^{(t+L)}}_{T_H}, \mathbf{O}_i^{(t+L)'}). \tag{15}$$

Similar to (3), $T_H = \lfloor \frac{H}{W_i} \rfloor$ denotes the number of $\mathbf{O}_i^{(t+L)}$, and $\mathbf{O}_i^{(t+L)'}$ denotes the subsequence extracted from $\mathbf{O}_i^t$, starting at index 0 with a length of $H \bmod W_i$.

By reusing the previously obtained cycle weights $\mathbf{g}$ to compute a weighted fusion of the corresponding periodic sequences, we obtain the periodic compensation term:

$$\mathbf{S} = \sum_{k=1}^{N} g_i \cdot \hat{\mathbf{S}}_i \in \mathbb{R}^{H \times C}. \tag{16}$$

The periodic components are added to the predicted residual $\mathbf{R}$ to obtain the final prediction result:

$$\overline{\mathbf{X}} = \mathbf{R} + \mathbf{S}. \tag{17}$$

IV. EXPERIMENTS

A. Datasets

To comprehensively evaluate the model performance, we select seven widely used and open-access benchmarks for multivariate time series forecasting, including ETT(ETTh1, ETTh2, ETTm1, ETTm2) [1], ECL, Weather, and Traffic [8].

Detailed dataset statistics are summarized in Table I.

TABLE I: STATISTICS OF THE DATASETS

Dataset	Channels	Timesteps	Frequency	Domain
ETTh1	7	17,420	1 hour	Electricity
ETTh2	7	17,420	1 hour	Electricity
ETTm1	7	69,680	15 mins	Electricity
ETTm2	7	69,680	15 mins	Electricity
Electricity	321	26,304	1 hour	Electricity
Traffic	862	17,544	1 hour	Transportation
Weather	21	52,696	10 mins	Weather

B. Baseline

We performed a thorough comparison of the AMCRNet model with the competing models. We carefully choose 10 state-of-the-art forecasting models as our baselines, including TQNet [14], TimeXer [18], CycleNet [13], iTransformer [23], TimesNet [24], PatchTST [10], Crossformer [25], DLinear [4], Autoformer [8]. To comprehensively evaluate the performance of MultiCycleNet, we employed two evaluation metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE), ensuring a robust assessment of both overall precision and the magnitude of deviations.

C. Main Results

Comprehensive long-term forecasting results are presented in Tables II. We fix the lookback window L at 96 and set the forecasting horizons $H \in \{96, 192, 336, 720\}$. Across a wide range of benchmarks, AMCRNet consistently outperforms

TABLE II: FULL RESULTS OF LONG-TERM MULTIVARIATE TIME SERIES FORECASTING TASKS. THE LOOKBACK HORIZON IS $L = 96$, AND THE PREDICTION LENGTHS $H \in \{96, 192, 336, 720\}$. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND THE SECOND BEST RESULTS ARE UNDERLINED.

Models	Metric	AMCRNet		TQNet		TimeXer		CycleNet		iTransformer		TimesNet		PatchTST		Crossformer		DLinear		Autoformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.368**	**0.393**	0.371	**0.393**	0.382	0.403	0.375	0.395	0.386	0.405	0.384	0.402	0.414	0.419	0.423	0.448	0.386	0.400	0.449	0.459
	192	**0.416**	**0.423**	0.428	0.426	0.429	0.435	0.436	0.428	0.441	0.436	0.436	0.429	0.460	0.445	0.471	0.474	0.437	0.432	0.500	0.482
	336	**0.466**	**0.442**	0.476	0.446	0.468	0.448	0.496	0.455	0.487	0.458	0.491	0.469	0.501	0.466	0.570	0.546	0.481	0.459	0.521	0.496
	720	**0.464**	**0.459**	0.487	0.470	0.469	0.461	0.520	0.484	0.503	0.491	0.521	0.500	0.500	0.488	0.653	0.621	0.519	0.516	0.514	0.512
	Avg	**0.429**	**0.429**	0.441	0.434	0.437	0.437	0.457	0.441	0.454	0.448	0.458	0.450	0.469	0.455	0.529	0.522	0.456	0.452	0.496	0.487
ETTh2	96	**0.282**	**0.336**	0.295	0.343	0.286	0.338	0.298	0.344	0.297	0.349	0.340	0.374	0.302	0.348	0.745	0.584	0.333	0.387	0.346	0.388
	192	**0.356**	**0.386**	0.367	0.393	0.363	0.389	0.372	0.396	0.380	0.400	0.402	0.414	0.388	0.400	0.877	0.656	0.477	0.476	0.456	0.452
	336	**0.410**	**0.421**	0.417	0.427	0.414	0.423	0.431	0.439	0.428	0.432	0.452	0.452	0.426	0.433	1.043	0.731	0.594	0.541	0.482	0.486
	720	0.411	0.434	0.433	0.446	**0.408**	**0.432**	0.450	0.458	0.427	0.445	0.462	0.468	0.431	0.446	1.104	0.763	0.831	0.657	0.515	0.511
	Avg	**0.365**	**0.394**	0.378	0.402	0.368	0.396	0.388	0.409	0.383	0.407	0.414	0.427	0.387	0.407	0.942	0.684	0.559	0.515	0.450	0.459
ETTm1	96	**0.304**	**0.349**	0.311	0.353	0.318	0.356	0.319	0.360	0.334	0.368	0.338	0.375	0.329	0.367	0.404	0.426	0.345	0.372	0.505	0.475
	192	**0.349**	**0.375**	0.356	0.378	0.362	0.383	0.360	0.381	0.377	0.391	0.374	0.387	0.367	0.385	0.450	0.451	0.380	0.389	0.553	0.496
	336	**0.377**	**0.398**	0.390	0.401	0.395	0.407	0.389	0.403	0.426	0.420	0.410	0.411	0.399	0.410	0.532	0.515	0.413	0.413	0.621	0.537
	720	**0.437**	**0.435**	0.452	0.440	0.452	0.441	0.447	0.441	0.491	0.459	0.478	0.450	0.454	0.439	0.666	0.589	0.474	0.453	0.671	0.561
	Avg	**0.367**	**0.389**	0.377	0.393	0.382	0.397	0.379	0.396	0.407	0.410	0.400	0.406	0.387	0.400	0.513	0.495	0.403	0.407	0.588	0.517
ETTm2	96	0.169	0.251	0.173	0.256	0.171	0.256	**0.163**	**0.246**	0.180	0.264	0.187	0.267	0.175	0.259	0.287	0.366	0.193	0.292	0.255	0.339
	192	0.234	0.295	0.238	0.298	0.237	0.299	**0.229**	**0.290**	0.250	0.309	0.249	0.309	0.241	0.302	0.414	0.492	0.284	0.362	0.281	0.340
	336	0.293	0.333	0.301	0.340	0.296	0.338	**0.284**	**0.327**	0.311	0.348	0.321	0.351	0.305	0.343	0.597	0.542	0.369	0.427	0.339	0.372
	720	**0.386**	**0.391**	0.397	0.396	0.392	0.394	0.389	**0.391**	0.412	0.407	0.408	0.403	0.402	0.400	1.730	1.042	0.554	0.522	0.433	0.432
	Avg	0.271	0.318	0.277	0.323	0.274	0.322	**0.266**	**0.314**	0.288	0.332	0.291	0.333	0.281	0.326	0.757	0.611	0.350	0.401	0.327	0.371
Weather	96	**0.152**	**0.199**	0.157	0.200	0.157	0.205	0.158	0.203	0.174	0.214	0.172	0.220	0.177	0.210	0.158	0.230	0.196	0.255	0.266	0.336
	192	**0.200**	**0.244**	0.206	0.245	0.204	0.247	0.207	0.247	0.221	0.254	0.219	0.261	0.225	0.250	0.206	0.277	0.237	0.296	0.307	0.367
	336	**0.255**	**0.285**	0.262	0.287	0.261	0.290	0.262	0.289	0.278	0.296	0.280	0.306	0.278	0.290	0.272	0.335	0.283	0.335	0.359	0.395
	720	**0.339**	0.339	0.344	0.342	0.340	0.341	0.344	0.344	0.358	0.349	0.365	0.359	0.354	0.340	0.398	0.418	0.345	0.381	0.419	0.428
	Avg	**0.237**	**0.267**	0.242	0.269	0.241	0.271	0.243	0.271	0.258	0.278	0.259	0.287	0.259	0.273	0.259	0.315	0.265	0.317	0.338	0.382
Electricity	96	**0.130**	**0.225**	0.134	0.229	0.140	0.242	0.136	0.229	0.148	0.240	0.168	0.272	0.181	0.270	0.219	0.314	0.197	0.282	0.201	0.317
	192	**0.150**	**0.243**	0.154	0.247	0.157	0.256	0.152	0.244	0.162	0.253	0.184	0.289	0.188	0.274	0.231	0.322	0.196	0.285	0.222	0.334
	336	**0.167**	**0.261**	0.169	0.264	0.176	0.275	0.170	0.264	0.178	0.269	0.198	0.300	0.204	0.293	0.246	0.337	0.209	0.301	0.231	0.338
	720	**0.200**	**0.292**	0.201	0.294	0.211	0.306	0.212	0.299	0.225	0.317	0.220	0.320	0.246	0.324	0.280	0.363	0.245	0.333	0.254	0.361
	Avg	**0.162**	**0.255**	0.164	0.259	0.171	0.270	0.168	0.259	0.178	0.270	0.193	0.295	0.205	0.290	0.244	0.334	0.212	0.300	0.227	0.338
Traffic	96	0.434	0.264	0.413	**0.261**	0.428	0.271	0.458	0.296	**0.395**	0.268	0.593	0.321	0.462	0.290	0.522	0.290	0.650	0.396	0.613	0.388
	192	0.453	0.272	0.432	**0.271**	0.448	0.282	0.457	0.294	**0.417**	0.276	0.617	0.336	0.466	0.290	0.530	0.293	0.598	0.370	0.616	0.382
	336	0.475	0.279	0.450	**0.277**	0.473	0.289	0.470	0.299	**0.433**	0.283	0.629	0.336	0.482	0.300	0.558	0.305	0.605	0.373	0.622	0.337
	720	0.526	0.302	0.486	**0.295**	0.516	0.307	0.502	0.314	**0.467**	0.302	0.640	0.350	0.514	0.320	0.589	0.328	0.645	0.394	0.660	0.408
	Avg	0.472	0.279	0.445	**0.276**	0.466	0.287	0.472	0.301	**0.428**	0.282	0.620	0.336	0.481	0.300	0.550	0.304	0.625	0.383	0.628	0.379

TABLE III: FULL RESULTS OF THE ABLATION ON AMCRNET, WITH AN INPUT SIZE T= 96. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND THE SECOND BEST RESULTS ARE UNDERLINED.

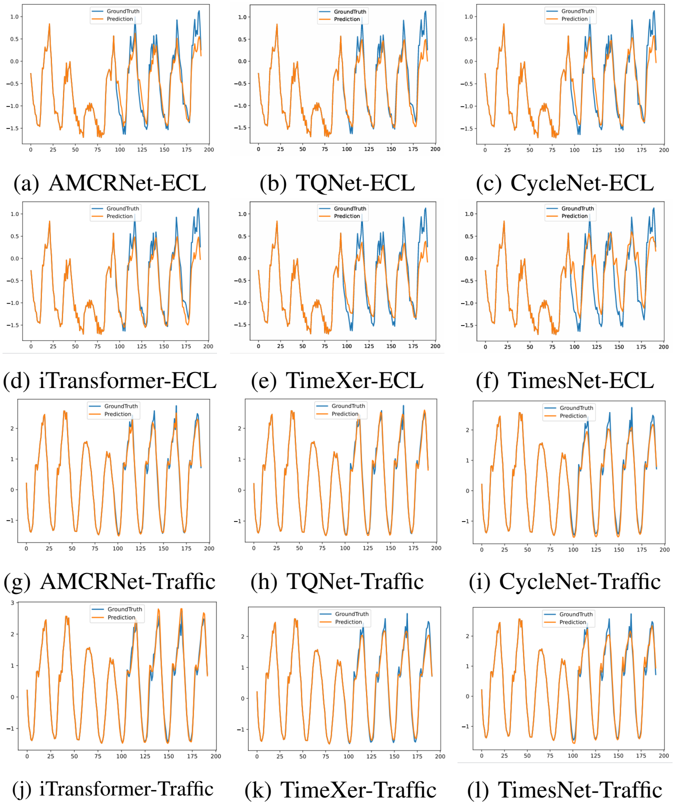
Models	Metric	w/o MultiCycle		w/o CQ		w/o CQ&MultiCycle		w/o γ & θ		AMCRNet	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.373	0.395	0.369	0.394	0.373	0.395	0.374	0.394	**0.368**	**0.393**
	192	0.440	0.434	0.417	0.424	0.440	0.433	0.428	0.431	**0.416**	**0.423**
	336	0.484	0.452	0.466	0.443	0.484	0.452	0.468	0.444	**0.466**	**0.442**
	720	0.515	0.483	0.466	0.460	0.516	0.483	0.501	0.486	**0.464**	**0.459**
	Avg	0.453	0.441	0.430	0.430	0.453	0.441	0.443	0.439	**0.429**	**0.429**
ETTm1	96	0.315	0.357	0.306	0.351	0.320	0.360	0.309	0.352	**0.304**	**0.349**
	192	0.351	0.377	0.352	0.377	0.354	0.379	0.358	0.382	**0.349**	**0.375**
	336	0.378	0.399	0.381	0.399	0.381	0.400	0.382	0.399	**0.377**	**0.398**
	720	0.438	0.437	0.439	0.436	0.441	0.437	0.443	0.437	**0.437**	**0.435**
	Avg	0.371	0.393	0.370	0.391	0.374	0.394	0.373	0.393	**0.367**	**0.389**

state-of-the-art Transformer-based and linear-based forecasters. In the ETT dataset, AMCRNet showed approximately 4% improvements in MAE and 2% improvement in MSE compared to TQNet. In the Weather and Electricity datasets, AMCRNet showed approximately 2% improvements in both MAE and MSE compared to TQNet. These results indicate that AMCRNet exhibits advantages in capturing periodic patterns and modeling inter-variable dependencies.

As shown in Fig. 2, we also provide visual comparisons among different models on Electricity dataset and Traffic dataset under input-96-prediect-96 setting. Among various deep learning based models, AMCRNet demonstrates outstanding performance in handling time series data prediction.

D. Ablation Studiesand Analysis

To verify the rationality and effectiveness of each module in AMCRNet, we conduct a comprehensive ablation study by progressively removing model components. The results are summarized in Table III. Ablation results indicate that each component of AMCRNet contributes materially to performance. Removing the multi-cycle decomposition compromises explicit modeling of superposed periodicities, leaving residual



(a) AMCRNet-ECL (b) TQNet-ECL (c) CycleNet-ECL

(d) iTransformer-ECL (e) TimeXer-ECL (f) TimesNet-ECL

(g) AMCRNet-Traffic (h) TQNet-Traffic (i) CycleNet-Traffic

(j) iTransformer-Traffic (k) TimeXer-Traffic (l) TimesNet-Traffic

Fig. 2: Visualization comparison of input-96-predict-96 results on the Electricity and Traffic datasets.

periodic interference and degrading long-term accuracy. Replacing the learnable Q_{base} with X_{input} as the query weakens the cross-sample global prior, making correlation modeling more sensitive to local noise. Disabling both MultiCycle and CQ removes periodic alignment and the stabilized correlation prior, resulting in a clear performance drop. Finally, removing the learnable fusion coefficients reduces adaptive feature balancing and may exacerbate attention-induced over-smoothing, thereby harming robustness and generalization.

V. CONCLUSION

The proposed AMCRNet improves the accuracy and robustness of long-term time series forecasting by effectively capturing periodic patterns in complex data and modeling residual inter-variable dependencies from both global and local perspectives. Extensive experiments across multiple datasets demonstrate that AMCRNet consistently outperforms existing methods on LTSF tasks. However, when the target series exhibits no pronounced periodicity, the effectiveness of the Adaptive Multi-Cycle Residual Decomposition module may be limited. In future work, We plan to incorporate graph-based relational modeling to better capture inter-variable dependencies and enhance generalization across diverse domains.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (62471141).

REFERENCES

[1] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *AAAI*, vol. 35, pp. 11106–11115, 2021.

[2] H. Wu, H. Zhou, M. Long, and J. Wang, "Interpretable weather forecasting for worldwide stations with a unified deep model," *Nature Machine Intelligence*, vol. 5, no. 6, pp. 602–611, 2023.

[3] C. W. J. Granger and P. Newbold, *Forecasting economic time series*. Academic press, 2014.

[4] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?," in *AAAI*, vol. 37, pp. 11121–11128, 2023.

[5] T. Dai, B. Wu, P. Liu, N. Li, J. Bao, Y. Jiang, and S.-T. Xia, "Periodicity decoupling framework for long-term series forecasting," in *ICLR*, 2024.

[6] M. Cools, E. Moons, and G. Wets, "Investigating effect of holidays on daily traffic counts: time series approach," *Transportation research record*, vol. 2019, no. 1, pp. 22–31, 2007.

[7] D. Luo and X. Wang, "Moderntcn: A modern pure convolution structure for general time series analysis," in *ICLR*, pp. 1–43, 2024.

[8] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *NeurIPS*, vol. 34, pp. 22419–22430, 2021.

[9] Z. Wu, C. Lian, X. Chen, T. Gong, X. Peng, and F. Chen, "Muticyclenet: Long-term time series forecasting using multi-cycle and residual prediction," in *ICBASE*, pp. 26–31, 2025.

[10] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *ICLR*, 2023.

[11] S. Lin, W. Lin, W. Wu, H. Chen, and J. Yang, "Sparsetsf: Modeling long-term time series forecasting with 1k parameters," in *ICML*, 2024.

[12] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. ZHOU, "Timemixer: Decomposable multiscale mixing for time series forecasting," in *ICLR*, 2024.

[13] S. Lin, W. Lin, X. Hu, W. Wu, R. Mo, and H. Zhong, "Cyclenet: Enhancing time series forecasting through modeling periodic patterns," *NeurIPS*, vol. 37, pp. 106315–106345, 2024.

[14] S. Lin, H. Chen, H. Wu, C. Qiu, and W. Lin, "Temporal query network for efficient multivariate time series forecasting," in *ICML*, 2025.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, vol. 30, 2017.

[16] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *ICML*, pp. 27268–27286, PMLR, 2022.

[17] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *ICLR*, 2022.

[18] Y. Wang, H. Wu, J. Dong, G. Qin, H. Zhang, Y. Liu, Y. Qiu, J. Wang, and M. Long, "Timexer: Empowering transformers for time series forecasting with exogenous variables," *NeurIPS*, 2024.

[19] K. Yi, Q. Zhang, W. Fan, S. Wang, P. Wang, H. He, N. An, D. Lian, L. Cao, and Z. Niu, "Frequency-domain mlps are more effective learners in time series forecasting," *NeurIPS*, vol. 36, pp. 76656–76679, 2023.

[20] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *ICLR*, 2021.

[21] Z. Liu, M. Cheng, Z. Li, Z. Huang, Q. Liu, Y. Xie, and E. Chen, "Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective," *NeurIPS*, vol. 36, pp. 14273–14292, 2023.

[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[23] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," in *ICLR*, 2023.

[24] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *ICLR*, 2023.

[25] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *ICLR*, 2023.