VCformer: Variable Correlation Transformer with Inherent Lagged Correlation for Multivariate Time Series Forecasting

Yingnan Yang[1] , **Qingling Zhu**[2] and **Jianyong Chen**[1,*]
[1]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
[2]National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen, China
csyyn1009@gmail.com, {zhuqingling, jychen}@szu.edu.cn

Abstract

Multivariate time series (MTS) forecasting has been extensively applied across diverse domains, such as weather prediction and energy consumption. However, current studies still rely on the vanilla point-wise self-attention mechanism to capture cross-variable dependencies, which is inadequate in extracting the intricate cross-correlation implied between variables. To fill this gap, we propose **V**ariable **C**orrelation Transformer (VCformer), which utilizes **V**ariable **C**orrelation **A**ttention (VCA) module to mine the correlations among variables. Specifically, based on the stochastic process theory, VCA calculates and integrates the cross-correlation scores corresponding to different lags between queries and keys, thereby enhancing its ability to uncover multivariate relationships. Additionally, inspired by Koopman dynamics theory, we also develop **K**oopman **T**emporal **D**etector (KTD) to better address the non-stationarity in time series. The two key components enable VCformer to extract both multivariate correlations and temporal dependencies. Our extensive experiments on eight real-world datasets demonstrate the effectiveness of VCformer, achieving top-tier performance compared to other state-of-the-art baseline models. Code is available at this repository: https://github.com/CSyyn/VCformer.

1 Introduction

Multivariate time series (MTS) forecasting is widely used in a range of applications, including energy consumption, weather, traffic, economics, and other fields [Shao *et al.*, 2022; Choi *et al.*, 2022; Wang *et al.*, 2023; Guo *et al.*, 2023; Castán-Lascorz *et al.*, 2022]. Unlike univariate time series, MTS involves multiple interrelated time-dependent variables, presenting unique challenges in capturing intricate inter-variable dependencies [Han *et al.*, 2023]. Consequently, MTS forecasting has always been a prominent research domain in both industry and academia.

The achievement of Transformer [Vaswani *et al.*, 2017] in natural language processing [Brown *et al.*, 2020] has led to the emergence of numerous Transformer variants for time series prediction tasks.

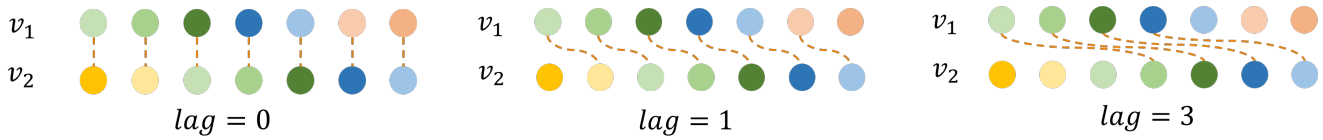
These models have developed various sophisticated attention mechanisms and enhancements to the Transformer architecture [Zhou *et al.*, 2022a; Wu *et al.*, 2021; Li *et al.*, 2019a; Zhou *et al.*, 2022b], which demonstrate a remarkable modelling ability for temporal dependencies in time series data [Wen *et al.*, 2023].

However, there is an ongoing academic discourse regarding their ability to effectively capture temporal dependencies [Zeng *et al.*, 2023] which typically embed each time step into a mix-channel token and apply attention mechanism on every token. Considering that these methods may overlook the valuable multivariate relationships, which is crucial for MTS forecasting, researchers have begun to focus on ensuring the channel independence and incorporating mutual information to explicitly model multivariate correlations. [Zhang and Yan, 2023; Nie *et al.*, 2023; Yu *et al.*, 2023; Liu *et al.*, 2023a].

Nevertheless, the traditional self-attention mechanism obtain the relationship between two variables via dot-product which can be approximately analogous to $attn(v_1, v_2)$ shown in Figure 1a. This approach aligns each time step of two variables ignoring the potential existence of different time delays between them, as shown in Figure 1b [John and Ferbinteanu, 2021; Chandereng and Gitter, 2020].

Addressing the limitations of vanilla variable point-wise attention, we introduce the **V**ariable **C**orrelation Transformer (VCformer) to fully exploit lagged correlation inherent in MTS through the **V**ariable **C**orrelation **A**ttention (VCA) module. The VCA module calculates the global strength of correlations between each query and key across different feature. Inspired by stochastic process theory [Chatfield and Xing, 2019; Blight and Chatfield, 1991], it not only computes auto-correlations akin to those in Autoformer [Wu *et al.*, 2021] but also extends this concept to determine lagged cross-correlations among various variates. The method employs a ROLL operation combined with Hadamard products to approximate these lagged correlations effectively. Furthermore, VCA adaptively aggregates lagged correlation over various lag lengths, thereby determining the comprehensive correlation for each variate. To enhance the model's capability

*Corresponding author



(a) Analogous to $attn(v_1, v_2)$ (b) Different lags inherent in time series

Figure 1: Illustration of the dot-product method to obtain correlations with different lags. For example, $lag = 3$ shows the similarity between v_1 and v_2 (points with the same color)

in addressing the non-stationary property in MTS, we design the **K**oopman **T**emporal **D**etector (KTD) module inspired by Koopman theory in dynamics [Koopman, 1931]. Experimentally, VCformer achieves state-of-the-art (SOTA) performance on eight real-world datasets. We also conduct experiments about VCA generality on other previous SOTA Transformer-based models, which demonstrates the powerful capability of modelling channel dependencies of VCA. In general, our contributions lie in the following three aspects:

- We propose a novel model for MTS forecasting, which is called VCformer. It learns both variable correlations and temporal dependencies of MTS.
- We design VCA mechanism to fully exploit lagged correlations among different variates. Additionally, we propose KTD inspired by Koopman theory in dynamics to effectively address non-stationarity in MTS forecasting.
- Experimentally, VCformer achieves top-tier performance on eight real-world datasets. To further evaluate the generality of VCA function, VCA is used in other Transformer-based models and gets better performance.

2 Related Work

Advancing beyond contemporaneous Temporal Convolutional Networks [Sen *et al.*, 2019] and RNN-based models [Salinas *et al.*, 2020; Lai *et al.*, 2018], Transformer variants have shown excellent capability in sequence modelling. All the modifications can be divided into two groups according to their focus: solely on modeling temporal dependencies and addressing both temporal and variable dependencies.

For the former, a series of sophisticated attention mechanisms has been developed which can be roughly classified into three categories. The first category is to remove redundant information by introducing sparse bias, thereby reducing the quadratic complexity of vanilla Transformer [Li *et al.*, 2019a; Zhou *et al.*, 2022a]. The second is to transfer the self-attention mechanism from time domain to frequency domain. This shift is facilitated by Fast Fourier Transform or other frequency analysis tools, enabling a more granular extraction for temporal dependencies at sub-series level [Wu *et al.*, 2021; Zhou *et al.*, 2022b]. The third category is related to tackling the distribution shift phenomenon in time series such as De-stationary attention [Liu *et al.*, 2022b]. Beyond these attention-focused innovations, the former also include methods that incorporate multi-resolution analysis of time series via hierarchical architectures [Liu *et al.*, 2022a].

With the primary focus on extracting temporal dependencies, these models designed various exquisite attention mechanisms and fancy architectures. However, a critical vulnerability in these models is the neglect of the rich cross-variable information, which is important for MTS forecasting tasks.

For the later in addressing MTS forecasting, two primary dimensions emerge in multivariate modeling: Channel-Independent (CI) and Channel-Dependent (CD). CI takes variates of time series independently and adopts the shared backbone. CD predicts future values by taking into account all the channels [Li *et al.*, 2023]. [Nie *et al.*, 2023] introduces patching and CI strategies, significantly enhancing its performance within Transformer-based architectures. Although CI structure is simple, its time-consuming training and inference has catalyzed the development of CD method for modeling multivariate relationships. For the CD method, [Zhang and Yan, 2023] employs temporal and variable attention serially to capture both cross-time and cross-dimension dependencies, while [Yu *et al.*, 2023] applies them in parallel. Moreover, iTransformer [Liu *et al.*, 2023a] revolutionizes the vanilla Transformer by inverting the duties of the traditional attention mechanism and the feed-forward network. They focus on capturing multivariate correlations and learning nonlinear representations respectively.

While these above works acknowledge the significance of modelling multivariate relationships, they adopt the classical self-attention mechanism based on point-wise method, which does not fully exploit the relationship among variable sequences. Despite the existing methods for analysis of lagged cross-correlations in time series [John and Ferbinteanu, 2021; Chandereng and Gitter, 2020; Shen, 2015], these time series Transformers in the literature have not leveraged them among variables, thereby limiting their predictive performance.

3 Method

In MTS forecasting, given historical observations $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\} \in \mathbb{R}^{T \times N}$ with T time steps and N variates, we predict the future H time steps $\mathbf{Y} = \{\mathbf{x}_{T+1}, \ldots, \mathbf{x}_{T+H}\} \in \mathbb{R}^{H \times N}$. To tackle this MTS forecasting task, we proposes **VCformer** which is shown in Figure 2.

3.1 Background

In this section, we first discuss the current limitation of vanilla variable attention in modelling feature-wise dependencies. This then motivates us to propose the variable cross-correlation attention mechanism, which operates across the feature channels for learning cross-correlation among variates.

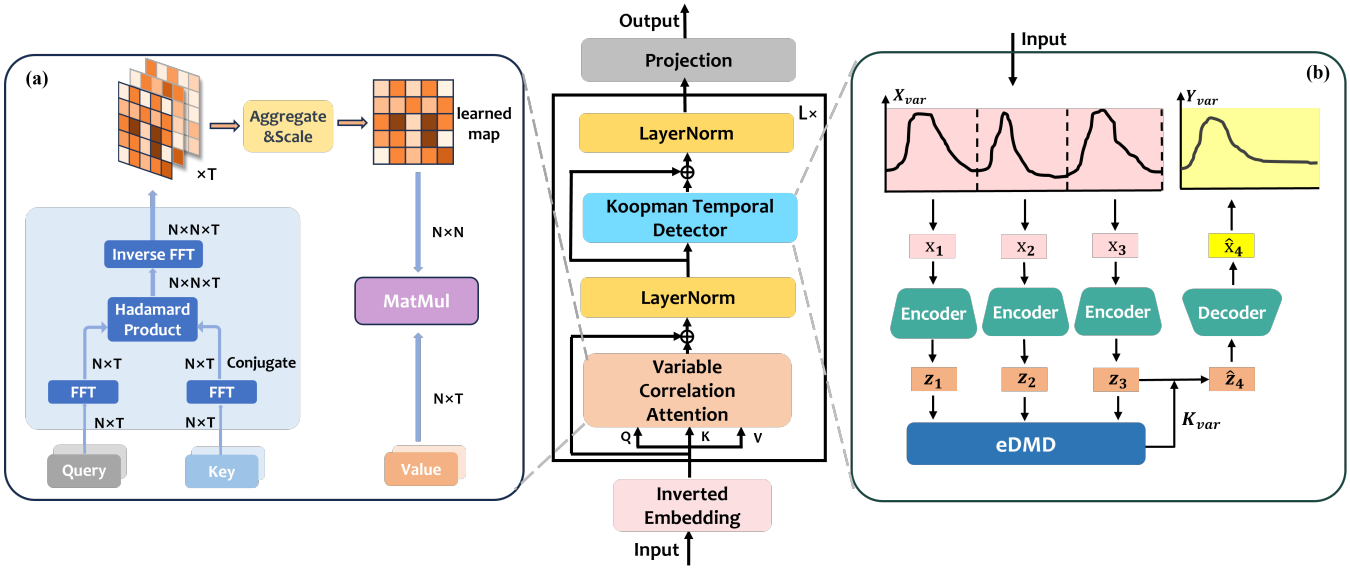


Figure 2: Overall structure of VCformer, VCA module (a) and KTD module (b)

Next, we review the Koopman theory and treat time series as dynamics. Based on this, we design the KTD module and combine it with the variable cross-correlation attention to learn both channels and time-steps dependencies.

Limitation of Vanilla Variable Attention

In the previous Transformer-based forecasters which adopted attention mechanism for facilitating the temporal dependencies, the self-attention module employs the linear projections to get queries, keys and values $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{T\times D}$, where D is the projected dimension. With the queries $Q = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T]^\top$ and keys $K = [\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_T]^\top$, the pre-Softmax attention score is the computation with $\mathbf{A}_{i,j} = \left(\mathbf{Q}\mathbf{K}^\top/\sqrt{D}\right)_{i,j} \propto \mathbf{q}_i^\top \mathbf{k}_j$. Nevertheless, feature-wise information, where each of the D features corresponds to an entry of $\mathbf{q}_i \in \mathbb{R}^{1\times D}$ or $\mathbf{k}_j \in \mathbb{R}^{1\times D}$, is absorbed into such inner-product representation. This thus makes such temporal attention unable to explicitly leverage the feature-wise information. iTransformer [Liu *et al.*, 2023a] considered the limitation of temporal attention and proposed the inverted Transformer to capture cross-variable dependencies that instead computes $K^\top Q \in \mathbb{R}^{D\times D}$. This simple design is suitable for capturing instantaneous cross-correlation, but it is insufficient for MTS data which is coupled with the intrinsic temporal dependencies. In particular, the variates of MTS data can be correlated with each other, yet with a lag interval. This phenomenon is referred to as lagged cross-correlation in MTS analysis [John and Ferbinteanu, 2021; Chandereng and Gitter, 2020; Shen, 2015]. Additionally, a variate in MTS data can even be correlated with the delayed copy of itself which is termed auto-correlation [Wu *et al.*, 2021]. With yet less-efficient modelling capabilities of cross-correlation, we hereby aim to derive a flexible and efficient correlated attention mechanism that can elevate existing Transformer-based models.

Non-linear Dynamics Tackled by Koopman Theory

Koopman theory [Koopman, 1931; Brunton *et al.*, 2022] shows that a linear dynamical system can be represented as an infinite-dimensional non-linear Koopman operator $\mathcal{K}$, which operates on a space of measurement functions g, such that:

$$\mathcal{K} \circ g\left(x_t\right) = g\left(\mathbf{F}\left(x_t\right)\right) = g\left(x_{t+1}\right) \tag{1}$$

Dynamic Mode Decomposition(DMD) [Schmid and Sesterhenn, 2008] seeks the best fitted matrix K to approximate infinite-dimensional operator $\mathcal{K}$ by collecting the observed system states (a.k.a *snapshots*). However, it is highly nontrivial to find appropriate measurement functions g as well as the Koopman operator $\mathcal{K}$. Therefore, by the universal approximation theorem [Hornik, 1991] of deep networks, many works employ DNNs to learn measurement functions in a data-driven way [Han *et al.*, 2020; Li *et al.*, 2019b; Morton *et al.*, 2019; Li and Jiang, 2021; Lusch *et al.*, 2018].

Koopman theory serves as a connection between finite-dimensional nonlinear dynamics and infinite-dimensional linear dynamics, enabling the use of spectral analysis tools for detailed examination. In this paper, we consider time series data $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ as observations of a series of dynamic system states, where $\mathbf{x}_i \in \mathbb{R}^N$ is the system state. Therefore, we design the KTD module which leverage Koopman-based approaches to tackle nonlinear dynamics.

3.2 Structure Overview

The proposed VCformer is shown in Figure 2. Following the same Encoder-only structure as iTransformer [Liu *et al.*, 2023a], we adopt the Inverted Embedding : $\mathbb{R}^T \mapsto \mathbb{R}^D$, which regards each univariate time series as the embedded token, instead of embedding multiple variates at the same time as the (temporal) token. By stacking L layers with VCA and KTD modules, the cross-variable relationships and temporal dependencies in time series can be captured. The final prediction is obtained by the Projection : $\mathbb{R}^D \mapsto \mathbb{R}^H$.

3.3 Variable Correlation Attention

Our VCA is comprised of lagged cross-correlation calculation and scores aggregation.

Lagged Cross-correlation Computing

Recall from stochastic process theory [Chatfield and Xing, 2019] that for any real discrete-time process $\{\mathcal{X}_t\}$, its auto-correlation $R_{\mathcal{X},\mathcal{X}}(\tau)$ can be computed as follows:

$$R_{\mathcal{X},\mathcal{X}}(\tau) = \lim_{L\to\infty} \frac{1}{L} \sum_{\tau=1}^{L} \mathcal{X}_t \mathcal{X}_{t-\tau} \tag{2}$$

Given the queries $Q = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N]$ and keys $K = [\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_N]$ expressed in feature-wise dimension where $\mathbf{q}_i, \mathbf{k}_j \in \mathbb{R}^{T\times 1}$, we make an approximation for the auto-correlation of variates i:

$$R_{\mathbf{q}_i,\mathbf{k}_i}(\tau) = \sum_{\tau=1}^{T} (\mathbf{q}_i)_t \cdot (\mathbf{k}_i)_{t-\tau} = \mathbf{q}_i \odot \mathrm{ROLL}(\mathbf{k}_i, \tau) \tag{3}$$

where $\mathrm{ROLL}(\mathbf{k}_i, \tau)$ denotes the elements of $\mathbf{k}_i$ shift along the time dimension and $\odot$ denotes the Hadamard product. This idea was also harnessed in Autoformer [Wu *et al.*, 2021]. Similarly, we can compute the cross-correlation between variate i and j by:

$$\text{LAGGED-COR}(\mathbf{q}_i, \mathbf{k}_j) = \mathbf{q}_i \odot \mathrm{ROLL}(\mathbf{k}_j, \tau) \tag{4}$$

where $\tau \in [1, T]$. Consequently, we calculate all the variates lagged cross-correlations with different lag lengths in this way.

Scores Aggregation

To obtain the total correlation of variate i and j, we aggregate different lags τ from 1 to T with learnable parameters $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_T]$ to more accurately calculate the effect of lagged correlation:

$$\mathrm{COR}(\mathbf{q}_i, \mathbf{k}_j) = \sum_{\tau=1}^{T} \lambda_i R_{\mathbf{q}_i,\mathbf{k}_j}(\tau) \tag{5}$$

Finally, the VCA performs softmax on the learned multivariate correlation map $\mathbf{A} \in \mathbb{R}^{N\times N}$ at each row and obtains the output via:

$$\mathrm{VCA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{SOFTMAX}(\mathrm{COR}(\mathbf{Q}, \mathbf{K}))\,\mathbf{V} \tag{6}$$

3.4 Koopman Temporal Detector

We employ the KTD to address the non-stationarity in the input series $\mathbf{X}_{var} \in \mathbb{R}^{N\times D}$ with multivariate correlation information. Remarkably, it is non-trivial to directly capture the non-stationarity in the entire series $\mathbf{X}_{var}$, but fortunately we identify that the localized time series exhibits weak stationarity, thereby aligning with the Koopman theory for nonlinear dynamics analysis. Consequently, we divide the input $\mathbf{X}_{var}$ into $\frac{D}{S}$ segments $\mathbf{x}_j$ of length S:

$$\mathbf{x}_j = \mathbf{X}_{\text{var}}[:, (j-1)S : jS], \quad j = 1, 2, \ldots, \frac{D}{S}. \tag{7}$$

where each segment can be served as a snapshot for the system. Subsequently, for every $\mathbf{x}_j \in \mathbb{R}^{N\times S}$, we leverage MLP-based Encoder : $\mathbb{R}^{N\times S} \mapsto \mathbb{R}^{M}$ to project it into a Koopman space embedding $\mathbf{z}_j \in \mathbb{R}^{M}$. According to eDMD [Williams *et al.*, 2015], these embeddings $Z = \left[z_1, z_2, \ldots, z_{\frac{D}{S}}\right] \in \mathbb{R}^{\frac{D}{S}\times M}$ are then utilized to calculate the fitted matrix K_{var}, facilitating an approximation of the infinite Koopman operator $\mathcal{K}$.

Specifically, given the Koopman embedding Z, we construct two matrices $Z_{\text{back}} = \left[z_1, z_2, \ldots, z_{\frac{D}{S}-1}\right] \in \mathbb{R}^{\left(\frac{D}{S}-1\right)\times M}$ and $Z_{\text{fore}} = \left[z_2, z_3, \ldots, z_{\frac{D}{S}}\right] \in \mathbb{R}^{\left(\frac{D}{S}-1\right)\times M}$, which respectively contain information of adjacent embeddings. After that, the fitted matrix $K_{\text{var}} \in \mathbb{R}^{D\times D}$ can be calculated as the following equation:

$$K_{\text{var}} = Z_{\text{fore}}\, Z_{\text{back}}^{\dagger} \tag{8}$$

where $Z_{\text{back}}^{\dagger}$ is the Moore-Penrose inverse of Z_{back} . Following the deviation of K_{var}, we iteratively apply it to predict $\frac{H}{S}$ Koopman embeddings as follows:

$$\hat{z}_{\frac{T}{S}+t} = \left(K_{\text{var}}\right)^t z_{\frac{T}{S}}, \quad t = 1, 2, \ldots, H/S. \tag{9}$$

In this way, a prediction of length H is obtained. Finally, to obtain the output of KTD, we adopt a Decoder : $\mathbb{R}^{M} \mapsto \mathbb{R}^{N\times S}$, which maps the predicted embeddings back, yielding Y_{var} as follows:

$$Y_{\text{var}} = \left[\hat{\mathrm{x}}_{\frac{T}{S}+1}, \ldots, \hat{\mathrm{x}}_{\frac{T}{S}+\frac{H}{S}}\right]^{\top} \tag{10}$$

3.5 Efficient Computation

For each vector pair $\mathbf{q}_i, \mathbf{k}_j \in \mathbb{R}^{T\times 1}$, the time complexity of the lag-correlation (Equation 5) is $\mathcal{O}\left(T^2\right)$. Therefore, calculating $\mathrm{COR}(q_i, \mathbf{K})$ demands $\mathcal{O}\left(NT^2\right)$ time. It leads the overall complexity of VCA to $\mathcal{O}\left(N^2T^2\right)$ in its current form. To alleviate the computational burden, we apply Fast Fourier Transforms (FFT) based on Wiener-Khinchin theorem [Wiener, 1930], thus reducing the complexity to $\mathcal{O}\left(N^2T\log T\right)$. Specifically, for computing the lag-correlation in Equation 4, given discrete time series $\{\mathcal{X}_t\}$ and $\{\mathcal{Y}_t\}$, the $R_{\mathcal{X}\mathcal{Y}}(\tau)$ can be calculated via FFT as follows:

$$\begin{aligned} \mathcal{S}_{\mathcal{X}\mathcal{Y}}(f) &= \mathcal{F}(\mathcal{X}_t)\,\mathcal{F}^{*}(\mathcal{Y}_t) \\ &= \int_{-\infty}^{+\infty} \mathcal{X}_t e^{-i2\pi tf} dt \overline{\int_{-\infty}^{+\infty} \mathcal{Y}_t e^{-i2\pi tf} dt} \\ R_{\mathcal{X}\mathcal{Y}}(\tau) &= \mathcal{F}^{-1}\left(\mathcal{S}_{\mathcal{X}\mathcal{Y}}(f)\right) = \int_{-\infty}^{+\infty} \mathcal{S}_{\mathcal{X}\mathcal{Y}}(f) e^{i2\pi fl} df, \end{aligned} \tag{11}$$

where $\tau \in [1, T]$. $\mathcal{F}$ and $\mathcal{F}^{-1}$ denotes the FFT and its inverse respectively, and $*$ is the conjugate operation. Specifically, we transform $\mathbf{Q}$ and $\mathbf{K}$ into frequency domain using FFT. Then element-wise multiplication ($a.k.a$ Hadamard Product) is applied to ith row of the $\mathcal{F}(\mathbf{Q})$ and $\mathcal{F}(\mathbf{K})$ to compute the $\text{LAGGED-COR}(\mathbf{q}_i, \mathbf{K})$. Extending this process to the entire matrix $\mathcal{F}(\mathbf{Q})$ and applying inverse FFTs to these products yield the complete lag-correlations between $\mathbf{Q}$ and $\mathbf{K}$. As FFT and inverse FFT each requires $\mathcal{O}(T\log T)$, the optimized VCA achieves a complexity of $\mathcal{O}\left(N^2T\log T\right)$.

Method		**VCformer**		iTransformer		PatchTST		DSformer		Koopa		Crossformer		TimesNet		DLinear		Stationary	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	0.171	0.220	0.174	0.214	0.177	0.218	**0.170**	**0.217**	0.177	0.226	0.185	0.248	0.172	0.220	0.196	0.255	0.205	0.265
	192	0.230	0.266	**0.221**	**0.254**	0.242	0.271	0.253	0.296	0.223	0.257	0.229	0.305	0.230	0.281	0.253	0.323	0.233	0.274
	336	0.280	0.299	**0.278**	**0.296**	0.290	0.305	0.285	0.310	0.281	0.299	0.323	0.285	0.335	0.311	0.325	0.390	0.296	0.317
	720	**0.352**	**0.344**	0.354	0.349	0.362	0.350	0.395	0.391	0.360	0.350	0.665	0.356	0.398	0.356	0.389	0.437	0.372	0.365
Electricity	96	**0.150**	**0.242**	0.154	0.245	0.188	0.280	0.164	0.261	0.174	0.273	0.153	0.250	0.182	0.285	0.195	0.276	0.172	0.275
	192	**0.167**	**0.255**	0.169	0.258	0.193	0.285	0.177	0.272	0.195	0.291	0.223	0.329	0.247	0.329	0.194	0.280	0.187	0.287
	336	**0.182**	**0.270**	0.185	0.275	0.211	0.302	0.201	0.294	0.216	0.310	0.191	0.291	0.256	0.338	0.207	0.295	0.208	0.307
	720	**0.221**	**0.302**	0.225	0.308	0.253	0.335	0.242	0.327	0.265	0.346	0.609	0.568	0.311	0.382	0.242	0.328	0.235	0.329
Traffic	96	**0.454**	**0.310**	0.717	0.466	0.475	0.303	0.546	0.352	0.539	0.368	0.530	0.285	0.592	0.315	0.640	0.388	0.732	0.418
	192	**0.468**	**0.315**	0.472	0.320	0.474	0.322	0.547	0.347	0.552	0.375	0.607	0.311	0.645	0.336	0.593	0.362	0.756	0.425
	336	**0.486**	**0.325**	0.488	0.330	0.489	0.332	0.562	0.352	0.573	0.383	0.642	0.324	0.659	0.347	0.600	0.365	1.172	0.680
	720	**0.524**	**0.348**	0.530	0.361	0.526	0.356	0.597	0.370	0.632	0.407	0.592	0.380	0.723	0.388	0.634	0.388	0.896	0.516
ETTh1	96	0.376	0.397	0.380	0.398	0.378	**0.396**	**0.373**	0.397	0.389	0.403	0.384	0.409	0.438	0.447	0.479	0.471	0.761	0.612
	192	0.431	0.427	0.433	0.428	0.433	0.427	**0.419**	**0.425**	0.438	0.431	0.461	0.459	0.488	0.472	0.448	0.443	0.746	0.599
	336	0.473	0.449	0.475	0.451	0.471	0.448	**0.457**	**0.446**	0.479	0.451	0.521	0.496	0.510	0.480	0.489	0.467	0.739	0.572
	720	0.476	0.474	0.486	0.480	**0.472**	**0.471**	0.499	0.497	0.486	0.474	0.627	0.586	0.511	0.497	0.511	0.509	0.757	0.612
ETTh2	96	**0.292**	**0.344**	0.292	0.344	0.292	0.345	0.296	0.351	0.306	0.355	0.596	0.532	0.337	0.375	0.299	0.351	0.477	0.462
	192	0.377	**0.396**	**0.375**	**0.396**	0.388	0.405	0.399	0.414	0.388	0.408	0.880	0.663	0.442	0.435	0.385	0.413	0.571	0.507
	336	**0.417**	**0.430**	0.418	0.430	0.427	0.436	0.434	0.443	0.430	0.443	1.988	1.097	0.476	0.468	0.511	0.490	0.608	0.534
	720	**0.423**	**0.443**	0.424	**0.443**	0.447	0.458	0.454	0.463	0.472	0.470	2.526	1.285	0.496	0.484	0.741	0.603	0.508	0.487
ETTm1	96	**0.319**	**0.359**	0.345	0.369	0.326	0.361	0.326	0.364	0.334	0.372	0.352	0.388	0.334	0.375	0.336	0.362	0.386	0.398
	192	0.364	0.382	0.386	0.391	0.372	**0.381**	**0.360**	0.382	0.374	0.391	0.409	0.436	0.385	0.401	0.378	0.389	0.459	0.434
	336	0.399	0.405	0.423	0.416	0.404	**0.403**	**0.394**	0.405	0.409	0.414	0.424	0.428	0.410	0.411	0.413	0.416	0.551	0.485
	720	**0.467**	0.442	0.491	0.445	**0.467**	**0.438**	0.474	0.451	0.473	0.448	0.569	0.528	0.513	0.473	0.475	0.454	0.585	0.516
ETTm2	96	**0.180**	**0.266**	0.190	0.276	0.193	0.280	0.201	0.286	0.187	0.271	0.297	0.370	0.185	0.267	0.188	0.284	0.240	0.320
	192	**0.245**	**0.306**	0.251	0.311	0.246	0.307	0.281	0.335	0.253	0.314	0.499	0.492	0.249	**0.306**	0.259	0.337	0.314	0.367
	336	**0.307**	**0.345**	0.315	0.352	0.314	0.351	0.336	0.367	0.323	0.358	0.597	0.684	0.314	0.346	0.334	0.389	0.340	0.371
	720	**0.406**	0.402	0.413	0.404	0.410	0.405	0.430	0.417	0.416	0.407	0.835	0.659	0.411	**0.399**	0.463	0.466	0.438	0.421
Exchange	96	**0.085**	**0.205**	0.090	0.211	0.100	0.231	0.092	0.216	0.092	0.217	0.139	0.265	0.108	0.244	0.110	0.266	0.154	0.297
	192	**0.176**	**0.299**	0.186	0.307	0.215	0.344	0.189	0.312	0.182	0.304	0.241	0.375	0.278	0.391	0.218	0.376	0.374	0.447
	336	**0.328**	**0.415**	0.339	0.424	0.403	0.473	0.348	0.430	0.349	0.432	0.392	0.468	0.523	0.556	0.387	0.497	0.548	0.563
	720	**0.830**	**0.688**	0.898	0.718	1.057	0.782	0.947	0.740	1.178	0.830	1.11	0.802	1.224	0.856	0.839	0.695	0.987	0.777

Table 1: Multivariate long-term time series forecasting results

4 Experiment

Dataset We conduct extensive experiments on eight widely-used real-world datasets [Zhou *et al.*, 2022a], including Electricity Transformer Temperature (ETT) with its four sub-datasets (ETTh1, ETTh2, ETTm1, ETTm2), Weather, Electricity, Traffic and Exchange. Following [Zhou *et al.*, 2022a], we adopt the same train/val/test sets with splits ratio 6:2:2. For the ETT datasets, we split the remaining four sub-datasets by the ratio of 7:1:2 following [Wu *et al.*, 2021].

Baselines We carefully select a range of SOTA methods as baselines to provide a comprehensive comparison with our proposed approach including (1) Transformer-based methods: Stationary [Liu *et al.*, 2022b], Crossformer [Zhang and Yan, 2023], DSformer [Zhang and Yan, 2023], iTransformer [Liu *et al.*, 2023a]; (2) MLP-based methods: DLinear [Zeng *et al.*, 2023], Koopa [Liu *et al.*, 2023b]; (3) TCN-based methods: TimesNet [Wu *et al.*, 2023].

Setups Following [Zhou *et al.*, 2022a], we normalize the train/val/test to zero-mean using the mean and standard deviation from the training set. The Mean Square Error (MSE) and Mean Absolute Error (MAE) are selected as evaluation metrics, consistent with previous works. All of models adopt the same prediction length $H = \{96, 192, 336, 720\}$. For the look-back window with length T, we follow the same setting as TimesNet [Wu *et al.*, 2023] which sets $T = 96$ for all the baselines to ensure the fairness.

4.1 Time Series Forecasting

Table 1 shows the comprehensive experimental results, where the lower MSE/MAE indicates the more accurate result. And we highlight the best in red and bold, while the second in blue and underlined. As we can see, Table 1 illustrate that VCformer consistently achieves top-tier performance across a range of datasets, outperforming other previous SOTA models. It can be attributed to the robust capability of VCA component in extracting correlations among multiple variables. Additionally, it is noteworthy that VCformer achieves the best results on the Exchange dataset which is characterized by high non-stationarity. This success can be owing to the KTD component which augments the power in capturing the non-stationarity from time series. Furthermore, in other datasets like ETT where VCformer does not attain the best benchmark, it still yield competitive results. We also find that the conventional Transformer-based models such as Non-stationary Transformer [Liu *et al.*, 2022b], only achieve the modest performance. It further substantiates the previously discussed limitations of the vanilla attention mechanism in tackling temporal dependencies.

4.2 VCA Generality

To further explore the effectiveness of VCA, we migrate the VCA module to several well-known Transformer variants:

Models		Nonstationary (2022)		Nonstationary (VCA)		Autoformer (2022)		Autoformer (VCA)		Informer (2021)		Informer (VCA)	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	0.172	0.275	0.163	0.262	0.201	0.317	0.170	0.273	0.274	0.368	0.195	0.301
	192	0.187	0.287	0.175	0.270	0.222	0.334	0.195	0.290	0.296	0.386	0.210	0.315
	336	0.208	0.307	0.195	0.286	0.231	0.338	0.200	0.295	0.300	0.394	0.231	0.339
	720	0.235	0.329	0.230	0.310	0.254	0.361	0.237	0.331	0.373	0.439	0.266	0.361
	Average	0.201	0.300	**0.191(5.1%)**	**0.282(6.0%)**	0.227	0.338	**0.201(11.7%)**	**0.297(12.1%)**	0.311	0.397	**0.226(27.5%)**	**0.329(17.1%)**
Exchange	96	0.154	0.297	0.100	0.235	0.197	0.323	0.124	0.278	0.847	0.752	0.301	0.414
	192	0.374	0.447	0.220	0.301	0.300	0.369	0.255	0.323	1.204	0.895	0.441	0.615
	336	0.548	0.563	0.405	0.479	0.509	0.524	0.443	0.501	1.672	1.036	0.573	0.729
	720	0.987	0.777	0.860	0.844	1.447	0.941	1.051	0.893	2.478	1.310	1.109	0.883
	Average	0.516	0.545	**0.396(23.2%)**	**0.465(14.7%)**	0.613	0.517	**0.468(23.6%)**	**0.499(3.5%)**	1.550	0.998	**0.606(60.9%)**	**0.660(33.9%)**
Traffic	96	0.612	0.338	0.540	0.321	0.613	0.388	0.559	0.357	0.719	0.391	0.590	0.371
	192	0.613	0.340	0.548	0.324	0.616	0.379	0.563	0.355	0.696	0.379	0.601	0.381
	336	0.618	0.328	0.554	0.331	0.622	0.337	0.570	0.366	0.777	0.420	0.595	0.382
	720	0.653	0.355	0.579	0.362	0.660	0.408	0.601	0.385	0.864	0.472	0.622	0.407
	Average	0.624	0.420	**0.555(11.6%)**	**0.334(20.4%)**	0.628	0.379	**0.573(8.8%)**	**0.365(3.5%)**	0.764	0.416	**0.602(21.2%)**	**0.385(7.4%)**

Table 2: VCA Generality and improvement for other Transformer-based models

Design	VCformer		Replace VCA		w/o VCA		Replace KTD		w/o KTD	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	**0.360**	**0.402**	0.390	0.445	0.419	0.454	0.425	0.480	0.440	0.506
Traffic	**0.483**	**0.324**	0.527	0.351	0.498	0.365	0.498	0.337	0.518	0.351
Electricity	**0.180**	**0.198**	0.290	0.445	0.184	0.288	0.184	0.288	0.190	0.280
Weather	**0.258**	**0.282**	0.265	0.285	0.264	0.290	0.264	0.290	0.269	0.291

Table 3: Ablations on VCformer. We conduct substitution and removal experiments on two key components (VCA & KTD) of VCformer respectively. For the substitution experiments, we replace the VCA and KTD modules with self-attention and FFN module respectively. The average results with all prediction lengths are presented in here.

Non-Stationary Transformer [Liu *et al.*, 2022b], Autoformer [Wu *et al.*, 2021] and Informer [Zhou *et al.*, 2022a]. Since these Transformer-based models have masked decoders in which the partial values of scores ($\mathbf{QK^T}$) are replaced with $-\infty$, the FFT in VCA module can not be used to quickly calculate the lag-correlations of queries and keys.

Therefore, we retain the original design of decoder and simply replace self-attention in encoder with VCA. The experimental results are shown in Table 2, where (VCA) represents the replaced model. We can see that the VCA module has significantly improved the performance of traditional Transformer-based models (Non-stationary Transformer, Autoformer and Informer), which yields an overall relative MSE reduction with **13.3%**, **14.7%** and **36.5%**.

4.3 Model Analysis

Ablation Study

In order to comprehensively understand the individual contributions of the key components in VCformer, we conduct ablation experiments covering experiments with both replacing components (Replace) and removing components (w/o), as shown in Table 3. From these results, we can conclude that both VCA and KTD are indispensable for the best performance of VCformer, which utilizes lag-correlation on variate dimension and Koopman detector on temporal dimension. After replacing or removing either one of them, the MSE/MAE

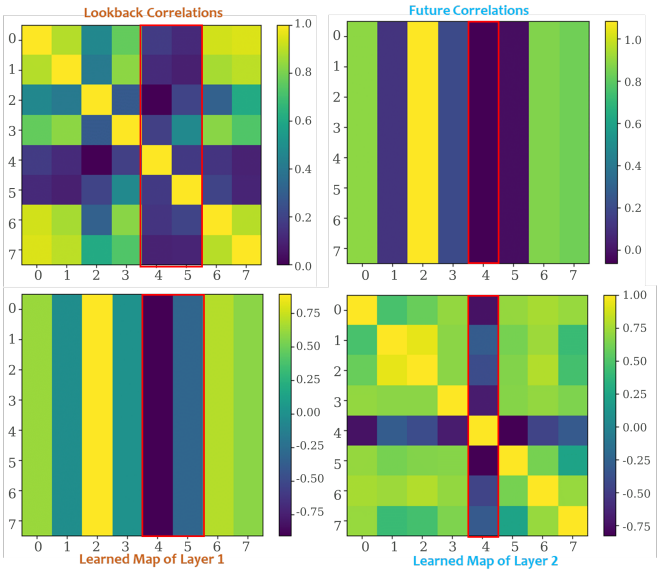


Figure 3: A case visualization for multivariate correlation analysis. The upper part is the multivariate correlation of past series and future series. The bottom part is the learned correlation maps in different layers.

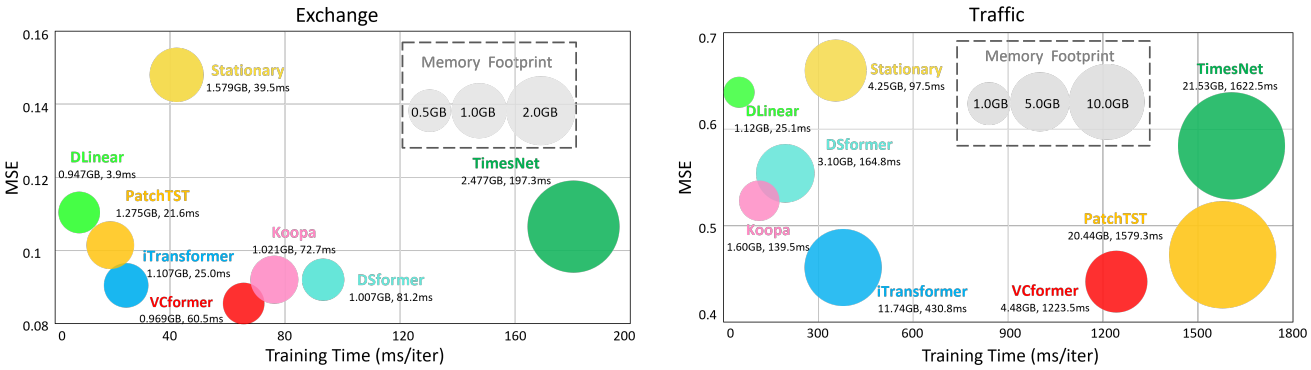


Figure 4: Model efficiency comparison. The running efficiency of eight models on the Exchange (left) and Traffic (right) dataset with the prediction length $H = 96$ and the batch size $B = 16$.

will increase, which validates their effectiveness. Notably, in the datasets with a large number of variates, specifically Traffic and Electricity, the replacement or removal of the VCA module incurs a remarkable increase in MSE/MAE, i.e., an averaged increase of 9% over the increase by KTD. This suggests that VCA plays a more critical role when dealing with a larger number of variates. On the other hand, within the Exchange dataset noted for its volatility, the substitution or removal of the KTD module, which can capture series non-stationarity, results in more noticeable performance drops than that of the VCA module. Conversely, pertaining to the Weather dataset, the experimental results indicate that the omission or replacement of either module doesn't lead to major variances in performance. These results show that the KTD module is good at learning features from non-stationary time series.

Multivariate Correlation Analysis

To enhance the interpretability of learned multivariate correlations by VCA, we provide a visualization case with random selection on series from Exchange. As demonstrated in Figure 3, the upper part presents the variate correlations inherent within the raw series including both input and prediction sequences. These correlations are calculated by Pearson Correlation coefficient as the following equation:

$$r_{xy} = \frac{\sum_{i=1}^{L}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{L}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{L}(y_i - \bar{y})^2}} \tag{12}$$

where $x_i, y_i \in \mathbb{R}$ run through all time points of the paired variates to be correlated. The lower part portrays the pre-Softmax score maps learned by VCA in both the first and the last layers.

From the red box in Figure 3, we can observe that the multivariate correlations learned by shallow layer of VCA are more similar to correlations of the input raw series. And as we explore at a deeper layer of VCA, we find that the learned multivariate correlations are closer to the forecasting sequences. This observation indicates that the focus of learned multivariate correlations shifts progressively from input series to prediction sequences. It also enhances interpretability of VCA which aggregates different lag-correlations to represent these variable relationships.

Model Efficiency Analysis

As shown in the Figure 4, we conduct a comparative study of the VCformer's efficiency with seven baselines. Our assessment considers three aspects: training speed, memory consumption and prediction performance. It can be observed that the time complexity of VCformer is $\mathcal{O}\left(N^2 T \log(T)\right)$. However, the coefficient N^2 does not significantly impact the training time when handling datasets with a small number of variates like Exchange. Notably, for the Traffic dataset which contains a large number of variates, the actual computational consumption is not as large as expected. It even outperforms PatchTST and TimesNet, which can be largely attributed to all the required calculations based on matrix operations. And these operations are well parallelized in built-in library.

5 Conclusion

In this paper, we address the limitations of the conventional dot product attention mechanism in extracting multivariate correlations. Then we propose VCformer which contains two effective modules. The VCA module can not only mine the lagged cross-correlation implicit in MTS, but also seamlessly integrate into other Transformer-based models. The KTD module employs MLP modules to derive Koopman embeddings and generates Koopman operator to enhance the capability for capturing non-stationarity in MTS. Extensive experiments shows that VCformer achieves SOTA forecasting performance and its VCA module is general enough to improve performance of various Transformer-based models.

Ethical Statement

There are no ethical issues.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants U2013201 and 62203308.

References

[Blight and Chatfield, 1991] P. A. Blight and C. Chatfield. The analysis of time series: An introduction. 4th edn. *Applied Statistics*, 40(1):178, Jan 1991.

[Brown *et al.*, 2020] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc.

[Brunton *et al.*, 2022] Steven L. Brunton, Marko Budišić, Eurika Kaiser, and J. Nathan Kutz. Modern koopman theory for dynamical systems. *SIAM Review*, page 229–340, May 2022.

[Castán-Lascorz *et al.*, 2022] M.A. Castán-Lascorz, P. Jiménez-Herrera, A. Troncoso, and G. Asencio-Cortés. A new hybrid method for predicting univariate and multivariate time series based on pattern forecasting. *Inf. Sci.*, 586(C):611–627, mar 2022.

[Chandereng and Gitter, 2020] Thevaa Chandereng and Anthony Gitter. Lag penalized weighted correlation for time series clustering. *BMC bioinformatics*, 21:1–15, 2020.

[Chatfield and Xing, 2019] Chris Chatfield and Haipeng Xing. *The analysis of time series: an introduction with R*. CRC press, 2019.

[Choi *et al.*, 2022] Taesung Choi, Dongkun Lee, Yuchae Jung, and Ho-Jin Choi. Multivariate time-series anomaly detection using seqvae-cnn hybrid model. In *2022 International Conference on Information Networking (ICOIN)*, pages 250–253, Jan 2022.

[Guo *et al.*, 2023] Wei Guo, Chang Meng, Enming Yuan, Zhicheng He, Huifeng Guo, Yingxue Zhang, Bo Chen, Yaochen Hu, Ruiming Tang, Xiu Li, et al. Compressed interaction graph based framework for multi-behavior recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 960–970, 2023.

[Han *et al.*, 2020] Yiqiang Han, Wenjian Hao, and Umesh Vaidya. Deep learning of koopman representation for control. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 1890–1895. IEEE, 2020.

[Han *et al.*, 2023] Lu Han, Han-Jia Ye, and De-Chuan Zhan. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *arXiv preprint arXiv:2304.05206*, 2023.

[Hornik, 1991] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, page 251–257, Jan 1991.

[John and Ferbinteanu, 2021] Majnu John and Janina Ferbinteanu. Detecting time lag between a pair of time series using visibility graph algorithm. *Communications in Statistics: Case Studies, Data Analysis and Applications*, 7(3):315–343, Jul 2021.

[Koopman, 1931] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

[Lai *et al.*, 2018] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

[Li and Jiang, 2021] Mengnan Li and Lijian Jiang. Deep learning nonlinear multiscale dynamic problems using koopman operator. *Journal of Computational Physics*, page 110660, Dec 2021.

[Li *et al.*, 2019a] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yuxiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *arXiv: Learning,arXiv: Learning*, Jun 2019.

[Li *et al.*, 2019b] Yunzhu Li, Hao He, Jiajun Wu, Dina Katabi, and Antonio Torralba. Learning compositional koopman operators for model-based control. *arXiv preprint arXiv:1910.08264*, 2019.

[Li *et al.*, 2023] Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*, 2023.

[Liu *et al.*, 2022a] 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*, 2022.

[Liu *et al.*, 2022b] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35:9881–9893, 2022.

[Liu *et al.*, 2023a] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

[Liu *et al.*, 2023b] Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[Lusch *et al.*, 2018] Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, Nov 2018.

[Morton *et al.*, 2019] Jeremy Morton, Freddie D Witherden, and Mykel J Kochenderfer. Deep variational koopman

models: Inferring koopman observations for uncertainty-aware dynamics modeling and control. *arXiv preprint arXiv:1902.09742*, 2019.

[Nie *et al.*, 2023] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

[Salinas *et al.*, 2020] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[Schmid and Sesterhenn, 2008] PeterJ. Schmid and Jörn Sesterhenn. Dynamic mode decomposition of numerical and experimental data. *Bulletin of the American Physical Society,Bulletin of the American Physical Society*, Nov 2008.

[Sen *et al.*, 2019] Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in neural information processing systems*, 32, 2019.

[Shao *et al.*, 2022] Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and ChristianS. Jensen. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *Proceedings of the VLDB Endowment*, 15(11):2733–2746, Jul 2022.

[Shen, 2015] Chenhua Shen. Analysis of detrended time-lagged cross-correlation between two nonstationary time series. *Physics Letters A*, 379(7):680–687, Mar 2015.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[Wang *et al.*, 2023] Fei Wang, Di Yao, Yong Li, Tao Sun, and Zhao Zhang. Ai-enhanced spatial-temporal data-mining technology: New chance for next-generation urban computing. *The Innovation*, 4(2):100405, Mar 2023.

[Wen *et al.*, 2023] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 6778–6786, August 2023.

[Wiener, 1930] Norbert Wiener. Generalized harmonic analysis. *Acta mathematica*, 55(1):117–258, 1930.

[Williams *et al.*, 2015] Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, page 1307–1346, Dec 2015.

[Wu *et al.*, 2021] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pages 22419–22430, 2021.

[Wu *et al.*, 2023] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023.

[Yu *et al.*, 2023] Chengqing Yu, Fei Wang, Zezhi Shao, Tao Sun, Lin Wu, and Yongjun Xu. Dsformer: A double sampling transformer for multivariate time series long-term prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3062–3072, 2023.

[Zeng *et al.*, 2023] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, Jun. 2023.

[Zhang and Yan, 2023] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023.

[Zhou *et al.*, 2022a] 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. *Proceedings of the AAAI Conference on Artificial Intelligence*, page 11106–11115, Sep 2022.

[Zhou *et al.*, 2022b] 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*, pages 27268–27286. PMLR, 2022.

A Experiment Details

A.1 Datasets

We conduct extensive experiments on eight real-world datasets including Weather, Electricity, Traffic, Exchange and four ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2). These datasets are widely used in time series forecasting. Here are more details about various datasets as follows:

- **Weather** [1] contains 21 meteorological measurements such as temperature, precipitation, wind speed and humidity, which is recorded every 10 minutes in German for 2020 whole year.
- **Electricity** [2] records hourly electricity consumption data for 321 clients from 2012 to 2014.
- **Traffic** [3] encompasses 862 traffic-related measurements such as vehicle counts, speed and congestion levels collected from sensors or cameras on road networks, which spans from 2015 to 2016 in the San Francisco Bay area.
- **Exchange** [4] collects the panel data of daily exchange rates from 8 countries from 1990 to 2016.
- **ETT** [5] (Electricity Transformer Temperature) consists of two years of data from two separate counties in China, with subsets created for different levels of granularity on Time series forecasting problem, including ETTh1 and ETTh2 for 1-hour-level data and ETTm1 for 15-minute-level data.

Table 4 includes detailed statistics of these datasets. *Timesteps* denotes the total number of time points in dataset. *Sample Frequency* denotes the sampling interval of time points. *Dimension* denotes the number of variates included in dataset.

Datasets	Timesteps	Sample Frequency	Dimension
Weather	52696	10 min	21
Electricity	26304	1 hour	321
Traffic	17544	1 hour	862
Exchange	7207	1 day	8
ETTh1	17420	1 hour	7
ETTh2	17420	1 hour	7
ETTm1	69680	15 min	7
ETTm2	69680	15 min	7

Table 4: The detail statistics of datasets

A.2 Baselines

The descriptions of selected baseline methods is given as follows:

- iTransformer [Liu *et al.*, 2023a] inverts the vanilla Transformer backbone for time series forecasting, applying the self-attention mechanism on learning multivariate correlations and encoding series representations by FFN. The source code is available at https://github.com/thuml/iTransformer.
- DSformer [Yu *et al.*, 2023] proposes double sampling (DS) block and the temporal variable attention (TVA) block to mine the global and local information as well as variable correlations. The source code is available at https://github.com/ChengqingYu/DSformer.
- PatchTST [Nie *et al.*, 2023] is a Transformer-based approach for MTS forecasting that utilizes patching and channel-independence strategies. The source code is available at https://github.com/yuqinie98/patchtst.
- Crossformer [Zhang and Yan, 2023] is the first Transformer-based model that employed the two-stage attention to capture both cross-dim and cross-time dependencies.The source code is available at https://github.com/Thinklab-SJTU/Crossformer.
- Koopa [Liu *et al.*, 2023b] is a MLP-based model that disentangle time series into time-invariant and time-variant dynamics. The source code is available at https://github.com/thuml/Koopa.
- TimesNet [Wu *et al.*, 2023] transforms 1D time series into 2D tensors based on Fourier Transform and applies inception blocks on 2D tensor to capture intricate temporal variations. The source code is available at https://github.com/thuml/TimesNet.
- DLinear [Zeng *et al.*, 2023] questions the efficiency of Transformer-based forecasters and leverage a simple one-layer linear model achieving superior performance on multiple datasets. The source code is available at https://github.com/curelab/LTSF-Linear.
- Nonstationary Transformer [Liu *et al.*, 2022b] focuses on the over-stationarization problem, designing the De-stationary attention mechanism to tackle it. The source code is available at https://github.com/thuml/Nonstationary_Transformers.

A.3 Experiment Setting and Hyperparameter

Our experiments, except for the Traffic dataset, are conducted with Pytorch 1.11.0 on single NVIDIA Tesla P100 GPU, which is equipped with 16GB CUDA memory. For the Traffic dataset, we run the experiments on multiple GPUs to facilitate the process. Following [Wu *et al.*, 2023], we set lookback window length $T = 96$ for all the experiments to ensure fairness and various prediction horizons $H \in \{96, 192, 336, 720\}$. All the experiments are trained by Adam using L2 loss and repeated three times to avoid accidents. All the baselines are reproduced by their official implementations with recommended hyperparameters. The batch size is set to 16 for most baselines, while Crossformer is set to 8 due to its large memory consumption. The learning rate is initialized to 0.5 and decays exponentially as training epochs grow. We set the number of encoder layers $L \in \{1, 2, 3\}$, the dimension of Koopman embedding $M \in \{256, 512, 1024\}$, the projection space of inverted embedding $D \in \{128, 256, 512\}$ and the length of

[1] https://www.bgc-jena.mpg.de/wetter/

[2] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

[3] https://pems.dot.ca.gov/

[4] https://github.com/laiguokun/multivariate-time-series-data

[5] https://github.com/zhouhaoyi/ETDataset

Algorithm 1 VCformer - Overall Architecture

Require: Input past time series $\mathbf{X}_{in} \in \mathbb{R}^{T \times N}$; Input length T; Prediction length H; Number of variates N; VCformer encoder block number L; Token dimension D; Koopman embedding dimension M; Koopman segment length S.

$\mathbf{X}_{in} = \mathbf{X}_{in}.\text{transpose}$ ▷ $\mathbf{X} \in \mathbb{R}^{N \times T}$

▷ Multi-layer Perceptron works on the last dimension to embed series into variate tokens.

$\mathbf{X}_{en}^{0} = \text{MLP}(\mathbf{X}_{in})$ ▷ $\mathbf{X}_{en}^{0} \in \mathbb{R}^{N \times D}$

for l **in** $\{1, \ldots, \text{L}\}$ **do** ▷ Run through VCformer blocks.

  ▷ VCA layer is applied on variate tokens to learn multivariate correlations.

  $\mathbf{X}_{en}^{l,1} = \text{LayerNorm}(\mathbf{X}_{en}^{l-1} + \text{VCA}(\mathbf{X}_{en}^{l-1}))$ ▷ $\mathbf{X}_{en}^{l,1} \in \mathbb{R}^{N \times D}$

  ▷ KTD layer is applied on temporal tokens to capture non-stationarity in time series.

  $\mathbf{X}_{en}^{l,2} = \text{LayerNorm}(\mathbf{X}_{en}^{l,1} + \text{KTD}(\mathbf{X}_{en}^{l,1}))$ ▷ $\mathbf{X}_{en}^{l,1} \in \mathbb{R}^{N \times H}$

  ▷ LayerNorm is adopted on series representations to reduce variates discrepancies.

  $\mathbf{X}_{en}^{l} = \mathbf{X}_{en}^{l,2}$ ▷ $\mathbf{X}_{en}^{l} \in \mathbb{R}^{N \times H}$

end for

$\hat{\mathbf{Y}} = \text{MLP}(X_{en}^{L})$ ▷ Project tokens from the output of Encoder, $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times H}$

$\hat{\mathbf{Y}} = \hat{\mathbf{Y}}.\text{transpose}$ ▷ $\mathbf{Y} \in \mathbb{R}^{H \times N}$

return $\hat{\mathbf{Y}}$ ▷ Return the prediction result $\hat{\mathbf{Y}}$

Koopman segment $S = 32$. And we also provide the pseudo-code of VCformer in Algorithm 1.

B Proof of Efficient Computation

Given series $\{\mathbf{k}_j\}$ that lags behind $\{\mathbf{q}_i\}$ by τ steps, we denote $\mathcal{K}_f$ and $\mathcal{Q}_f$ as the respective frequency components. According to stochastic process theory, the cross-correlation between them can be defined by:

$$R_{\mathbf{q}_i,\mathbf{k}_j}(\tau) = \frac{1}{T}\sum_{t=1}^{T} \mathbf{k}_j(t-\tau)\mathbf{q}_i(t)$$

With $\mathbf{k}_j(t-\tau)$ denoted as $\check{\mathbf{k}}_j(\tau - t)$ and $(\tau - t)$ denoted as τ', we derive the Fourier Transform of $\left\{\sum_{t=1}^{T} \mathbf{q}_i(t)\check{\mathbf{k}}_j(\tau - t)\right\}_{\tau=1}^{T}$ as follows:

$$\begin{aligned}
&\mathcal{F}\left(\sum_{t=1}^{T} \mathbf{q}_i(t)\check{\mathbf{k}}_j(\tau - t)\right)_f \\
&= \sum_{\tau=1}^{T}\left(\sum_{t=1}^{T} \mathbf{q}_i(t)\check{\mathbf{k}}_j(\tau - t)\right) e^{-i2\pi f\tau} \\
&= \sum_{\tau=1}^{T} \mathbf{q}_i(t)\left(\sum_{t=1}^{T} \check{\mathbf{k}}_j(\tau - t) e^{-i2\pi f\tau}\right) \\
&= \sum_{\tau=1}^{T} \mathbf{q}_i(t) e^{-i2\pi ft}\left(\sum_{t=1}^{T} \check{\mathbf{k}}_j(\tau - t) e^{-i2\pi f(\tau - t)}\right) \\
&= \mathcal{Q}_f\left(\sum_{\tau=1}^{T} \check{\mathbf{k}}_j\left(\tau'\right) e^{-i2\pi f\tau'}\right).
\end{aligned}$$

Assuming $\check{\mathbf{k}}_j$ is T-periodic, we have

$$\begin{aligned}
&\sum_{\tau'=1-t}^{T-t} \check{\mathbf{k}}_j\left(\tau'\right) e^{-i2\pi f\tau'} \\
&= \sum_{\tau'=1-t}^{0} \check{\mathbf{k}}_j\left(\tau'\right) \cdot e^{-i2\pi f\left(\tau' + T\right)} + \sum_{\tau'=1}^{T-t} \check{\mathbf{k}}_j\left(\tau'\right) \cdot e^{-i2\pi f\tau'} \\
&= \sum_{\tau'=1-t}^{0} \check{\mathbf{k}}_j\left(\tau' + T\right) \cdot e^{-i2\pi f\left(\tau' + T\right)} + \sum_{\tau'=1}^{T-t} \check{\mathbf{k}}_j\left(\tau'\right) \cdot e^{-i2\pi f\tau'} \\
&= \sum_{\tau'=T-t+1}^{T} \check{\mathbf{k}}_j\left(\tau'\right) \cdot e^{-i2\pi f\tau'} + \sum_{\tau'=1}^{T-t} \check{\mathbf{k}}_j\left(\tau'\right) \cdot e^{-i2\pi f\tau'} \\
&= \sum_{\tau'=1}^{T} \check{\mathbf{k}}_j\left(\tau'\right) \cdot e^{-i2\pi f\tau'} \\
&= \mathcal{F}(\check{\mathbf{k}}_j)_f.
\end{aligned}$$

Due to the conjugate symmetry of the DFT on real-valued signals, we have

$$\mathcal{F}(\check{\mathbf{k}}_j)_f = \overline{\mathcal{F}(\mathbf{k}_j)}_f = \overline{\mathcal{K}_f},$$

where the bar is the conjugate operation. Thereby, we can obtain

$$\mathcal{F}\left(\sum_{t=1}^{T} \mathbf{q}_i(t)\check{\mathbf{k}}_j(\tau - t)\right)_f = \mathcal{Q}_f\overline{K_f}$$

Finally, we can obtain $R_{\mathbf{q}_i,\mathbf{k}_j}(\tau)$ as:

$$R_{\mathbf{q}_i,\mathbf{k}_j}(\tau) = \frac{1}{T}\mathcal{F}^{-1}\left(\mathcal{F}\left(\mathbf{q}_i\right) \odot \overline{\mathcal{F}\left(\mathbf{k}_j\right)}\right)_\tau$$

Note that $R_{\mathbf{q}_i,\mathbf{k}_j}(\tau) \in [-1, 1]$ when $\mathbf{q}_i$ and $\mathbf{k}_j$ have been normalized.

C Hyperparameter Sensitivity

We conduct experiments about the hyperparameter sensitivity of VCformer as shown in Figure 5, which include three factors: the number of encoder layers L, the dimension D of inverted embeddings and the dimension M of Koopman embeddings. Based on the experimental results, we find that as the hyperparameter values increasing, the performance on most datasets will have an improvement except for Exchange dataset. It can be attributed to the overfitting problem which is caused by the high volatility of Exchange dataset and the increasing parameters of model. Moreover, compared to other datasets, the Electricity and ETT datasets exhibit low sensitivity to changes in the hyperparameter.

D Full Results

D.1 Full Ablation Results

Due to the limited pages, we list the overall ablation study results on the effect of VCA and KTD in VCformer as shown in Table 5. The detailed ablations contain two type of experiments denoted as removing components (w/o) and replacing components (replace).

In Table 5, among different architecture designs, VCformer utilizes the lagged-correlation inherent between variates by the VCA module and captures the non-stationarity in time series by the KTD module. It thus exhibits top-tier performance (with the average results in bold). Specifically, the replacement of VCA achieves inferior performance which indicates the deficiency of vanilla point-wise self-attention mechanism on learning the multivariate correlations. It can be attributed to the neglect of existence of lagged-correlations. With the increasing number of variates, the deterioration in performance becomes increasingly evident. It implies that the importance of capturing multivariate correlations is ever more highlighted. Besides, the replacement of KTD by FFN also gets the worse performance especially on the Exchange dataset which is noted for the non-stationarity. This phenomenon indicates the effectiveness of KTD module for the capability to mine the complex temporal dependencies.

D.2 Full VCA Generality Results

In Table 6, we apply the VCA module to six Transformer-based models and set the better average results in bold. "Replace" denotes that the VCA module is used to substitute the self-attention mechanism in these Transformer variants. From the results, we can find that the number of bold average results for "Replace" (count=52) is much more than "Origin" (count=8). Due to the capability to learn the multivariate correlations, the replacements by VCA module significantly improves the performance of these Transformer-based methods.

E Visualization of MTS Forecasting

To illustrate the prediction performance of VCformer more intuitively, we list several prediction showcases of three datasets in Figure 6-8 given by VCformer, iTransformer [Liu *et al.*, 2023a], DSformer [Yu *et al.*, 2023], PatchTST [Nie *et al.*, 2023], DLinear [Zeng *et al.*, 2023] and Koopa [Liu *et al.*, 2023b]. The look-back window and prediction length are both set to 96 for all models. From the visualization, VCformer exhibit precise prediction to the ground truth and thus achieve superior performance.

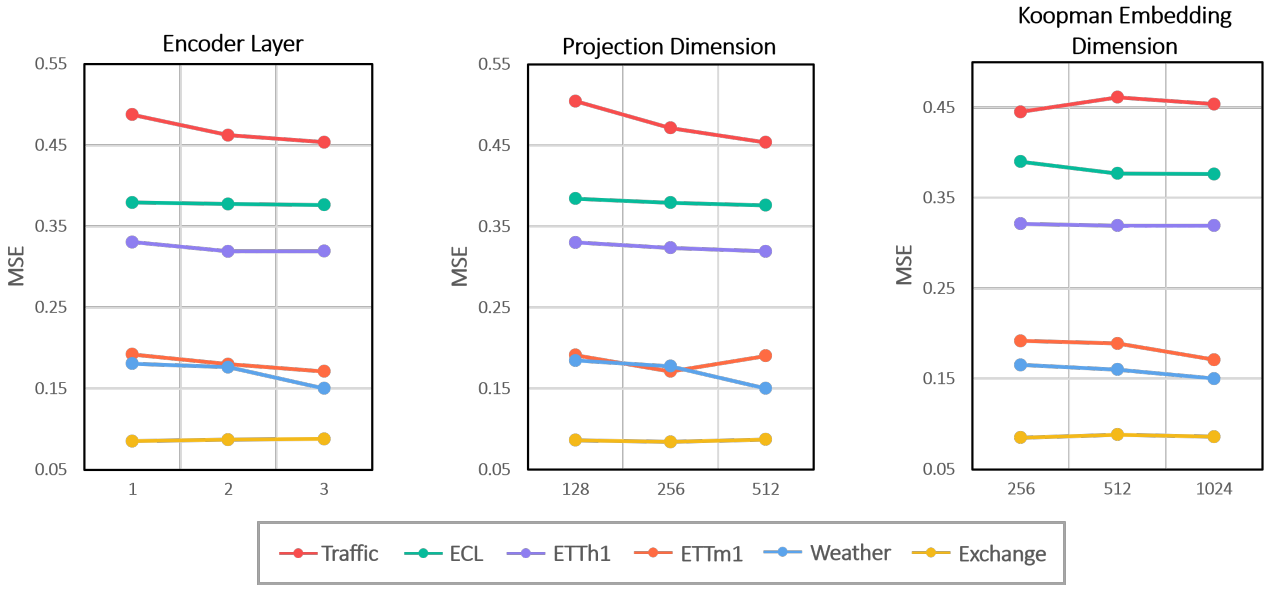


Figure 5: Hyperparameter sensitivity with respect to the encoder layer, the Koopman embedding dimension and the projection dimension of variate tokens.

Design	Variate	Temporal	Prediction	Exchange		Traffic		Electricity		Weather		ETTh2	
			Lengths	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
VCformer	VCA	KTD	96	0.085	0.205	0.454	0.310	0.150	0.242	0.171	0.220	0.292	0.344
			192	0.176	0.195	0.468	0.315	0.167	0.255	0.230	0.266	0.377	0.396
			336	0.328	0.328	0.486	0.325	0.182	0.270	0.280	0.299	0.417	0.430
			720	0.850	0.914	0.524	0.348	0.221	0.302	0.352	0.344	0.423	0.443
			Avg	**0.360**	**0.402**	**0.483**	**0.325**	**0.180**	**0.267**	**0.258**	**0.282**	0.377	0.403
Replace	Attention	KTD	96	0.100	0.235	0.490	0.235	0.162	0.259	0.173	0.218	0.292	0.345
			192	0.195	0.331	0.513	0.331	0.180	0.275	0.235	0.273	0.376	0.395
			336	0.350	0.485	0.529	0.485	0.203	0.294	0.290	0.301	0.419	0.432
			720	0.914	0.735	0.577	0.735	0.245	0.330	0.363	0.349	0.422	0.441
			Avg	0.390	0.447	0.527	0.351	0.198	0.290	0.265	0.285	**0.376**	**0.401**
	VCA	FFN	96	0.110	0.265	0.469	0.321	0.155	0.246	0.175	0.223	0.295	0.346
			192	0.219	0.377	0.481	0.327	0.170	0.259	0.234	0.270	0.381	0.401
			336	0.389	0.500	0.490	0.340	0.186	0.276	0.289	0.304	0.423	0.436
			720	0.982	0.779	0.551	0.359	0.226	0.310	0.357	0.351	0.429	0.452
			Avg	0.425	0.480	0.498	0.337	0.184	0.273	0.264	0.287	0.382	0.409
w/o	w/o	KTD	96	0.092	0.216	0.501	0.342	0.165	0.262	0.177	0.225	0.301	0.357
			192	0.201	0.356	0.520	0.358	0.179	0.272	0.236	0.271	0.385	0.406
			336	0.357	0.493	0.535	0.371	0.199	0.290	0.291	0.302	0.433	0.448
			720	1.025	0.751	0.596	0.389	0.247	0.329	0.381	0.362	0.457	0.469
			Avg	0.419	0.454	0.538	0.365	0.198	0.288	0.271	0.290	0.394	0.420
	VCA	w/o	96	0.188	0.270	0.501	0.333	0.160	0.253	0.176	0.225	0.305	0.360
			192	0.215	0.389	0.520	0.341	0.174	0.261	0.235	0.273	0.387	0.403
			336	0.403	0.573	0.512	0.356	0.192	0.283	0.293	0.309	0.436	0.447
			720	1.025	0.792	0.577	0.375	0.234	0.323	0.371	0.357	0.461	0.470
			Avg	0.440	0.506	0.518	0.351	0.190	0.280	0.269	0.291	0.397	0.420

Table 5: Full results of the ablation on VCformer. We conduct substitution and removal experiments on two key components (VCA & KTD) of VCformer respectively on the dimensions they represent (Variate & Temporal).

Models			iTransformer		DSformer		Crossformer		Stationary		Autoformer		Informer	
Metric			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	Original	96	0.154	0.245	0.164	0.261	0.153	0.250	0.172	0.275	0.201	0.317	0.274	0.368
		192	0.169	0.258	0.177	0.272	0.223	0.329	0.187	0.287	0.222	0.334	0.296	0.386
		336	0.185	0.275	0.201	0.294	0.191	0.291	0.208	0.307	0.231	0.338	0.300	0.394
		720	0.225	0.308	0.242	0.327	0.609	0.568	0.235	0.329	0.254	0.361	0.373	0.439
		Avg	0.183	0.272	0.196	0.288	0.294	0.359	0.200	0.299	0.227	0.338	0.311	0.397
	Replace	96	0.151	0.243	0.159	0.252	0.153	0.251	0.163	0.262	0.170	0.273	0.195	0.301
		192	0.168	0.256	0.172	0.263	0.219	0.328	0.175	0.270	0.195	0.290	0.210	0.315
		336	0.183	0.272	0.190	0.285	0.190	0.288	0.195	0.286	0.200	0.295	0.231	0.339
		720	0.223	0.305	0.235	0.315	0.610	0.566	0.230	0.310	0.237	0.331	0.266	0.361
		Avg	**0.181**	**0.269**	**0.189**	**0.279**	**0.293**	**0.358**	**0.191**	**0.282**	**0.200**	**0.297**	**0.226**	**0.329**
Exchange	Original	96	0.090	0.211	0.092	0.216	0.139	0.265	0.154	0.297	0.197	0.323	0.847	0.752
		192	0.186	0.307	0.189	0.312	0.241	0.375	0.374	0.447	0.300	0.369	1.204	0.895
		336	0.339	0.424	0.348	0.430	0.392	0.468	0.548	0.563	0.509	0.524	1.672	1.036
		720	0.898	0.718	0.947	0.740	1.110	0.802	0.987	0.777	1.447	0.941	2.478	1.310
		Avg	0.378	0.415	**0.394**	**0.424**	0.471	0.478	0.516	0.521	0.613	0.539	1.550	0.998
	Replace	96	0.088	0.207	0.095	0.217	0.097	0.225	0.100	0.235	0.124	0.278	0.301	0.414
		192	0.183	0.302	0.192	0.320	0.197	0.332	0.220	0.301	0.255	0.323	0.441	0.615
		336	0.334	0.420	0.349	0.435	0.350	0.447	0.405	0.479	0.443	0.501	0.573	0.729
		720	0.866	0.695	0.960	0.745	0.973	0.755	0.860	0.844	1.501	0.893	1.109	0.883
		Avg	**0.368**	**0.406**	0.399	0.429	**0.404**	**0.440**	**0.396**	**0.465**	**0.581**	**0.499**	**0.606**	**0.660**
Traffic	Original	96	0.717	0.466	0.546	0.352	0.530	0.285	0.612	0.338	0.613	0.388	0.719	0.391
		192	0.472	0.320	0.547	0.347	0.607	0.311	0.613	0.340	0.616	0.379	0.696	0.379
		336	0.488	0.330	0.562	0.352	0.642	0.324	0.618	0.328	0.622	0.337	0.777	0.420
		720	0.530	0.361	0.597	0.370	0.592	0.380	0.653	0.355	0.660	0.408	0.864	0.472
		Avg	0.552	0.369	0.563	0.355	0.593	0.325	0.624	0.340	0.628	0.378	0.764	0.416
	Replace	96	0.495	0.334	0.519	0.341	0.527	0.283	0.540	0.321	0.559	0.357	0.590	0.371
		192	0.470	0.319	0.525	0.343	0.565	0.299	0.548	0.324	0.563	0.355	0.601	0.381
		336	0.487	0.328	0.541	0.349	0.583	0.325	0.554	0.331	0.570	0.366	0.595	0.382
		720	0.526	0.359	0.568	0.363	0.591	0.379	0.579	0.362	0.601	0.385	0.622	0.407
		Avg	**0.495**	**0.335**	**0.538**	**0.349**	**0.567**	**0.322**	**0.555**	**0.335**	**0.573**	**0.366**	**0.602**	**0.385**
Weather	Original	96	0.174	0.214	0.170	0.217	0.185	0.248	0.205	0.265	0.266	0.336	0.300	0.384
		192	0.221	0.254	0.253	0.296	0.229	0.305	0.233	0.274	0.336	0.367	0.598	0.544
		336	0.278	0.296	0.285	0.310	0.323	0.285	0.296	0.317	0.359	0.395	0.578	0.523
		720	0.354	0.349	0.395	0.391	0.665	0.356	0.372	0.365	0.419	0.428	1.059	0.741
		Avg	**0.257**	**0.278**	0.276	0.304	0.351	0.299	0.276	0.305	0.345	0.382	0.634	0.548
	Replace	96	0.175	0.215	0.175	0.100	0.186	0.250	0.193	0.260	0.244	0.329	0.269	0.372
		192	0.223	0.260	0.244	0.288	0.237	0.310	0.230	0.269	0.319	0.359	0.493	0.469
		336	0.285	0.303	0.269	0.295	0.301	0.279	0.293	0.315	0.343	0.387	0.489	0.475
		720	0.356	0.352	0.379	0.380	0.490	0.417	0.365	0.361	0.401	0.392	0.882	0.693
		Avg	0.260	0.283	**0.267**	**0.293**	**0.304**	**0.314**	**0.270**	**0.301**	**0.327**	**0.367**	**0.500**	**0.508**
ETTh2	Original	96	0.292	0.344	0.296	0.351	0.745	0.584	0.477	0.462	0.346	0.388	3.755	1.525
		192	0.375	0.396	0.399	0.414	0.877	0.656	0.571	0.507	0.456	0.452	5.602	1.931
		336	0.418	0.430	0.434	0.443	1.043	0.731	0.608	0.534	0.482	0.486	4.721	1.835
		720	0.424	0.443	0.454	0.463	1.104	0.763	0.508	0.487	0.515	0.511	3.647	1.625
		Avg	**0.377**	**0.403**	**0.396**	**0.418**	0.942	0.684	0.541	0.498	0.450	0.459	1.301	3.874
	Replace	96	0.293	0.350	0.300	0.359	0.379	0.402	0.430	0.445	0.344	0.373	1.293	0.925
		192	0.380	0.399	0.399	0.417	0.410	0.433	0.523	0.483	0.435	0.441	1.595	0.957
		336	0.420	0.431	0.434	0.445	0.455	0.489	0.557	0.519	0.472	0.469	2.014	1.133
		720	0.424	0.439	0.459	0.465	0.829	0.693	0.492	0.480	0.499	0.497	2.355	1.294
		Avg	0.379	0.405	0.398	0.422	**0.518**	**0.504**	**0.501**	**0.482**	**0.438**	**0.445**	**0.675**	**1.722**

Table 6: Full results of VCA generality experiments on six Transformer-based models.

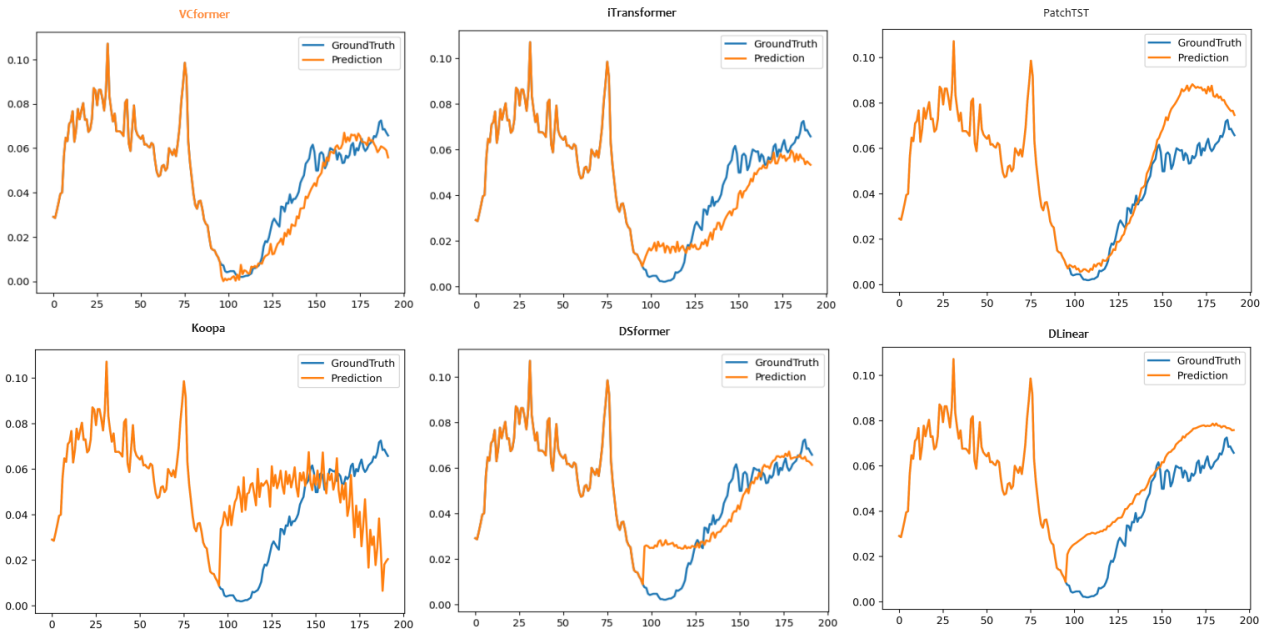


Figure 6: Visualization of input-96-predict-96 results on the Weather dataset.

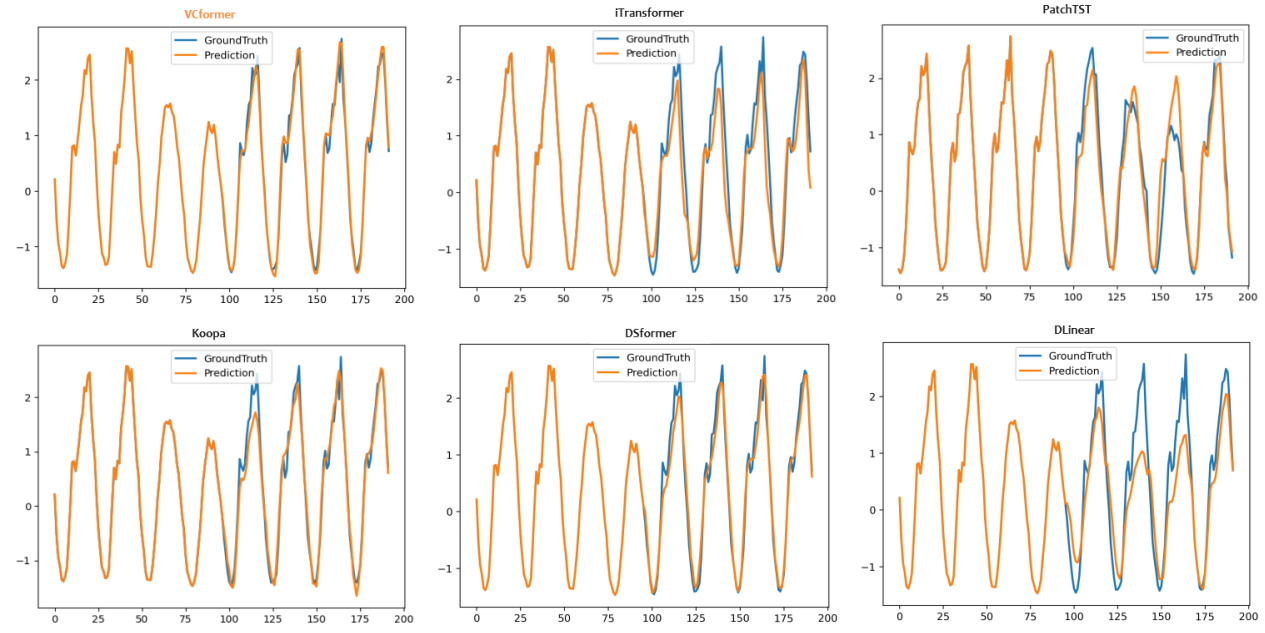


Figure 7: Visualization of input-96-predict-96 results on the Traffic dataset.

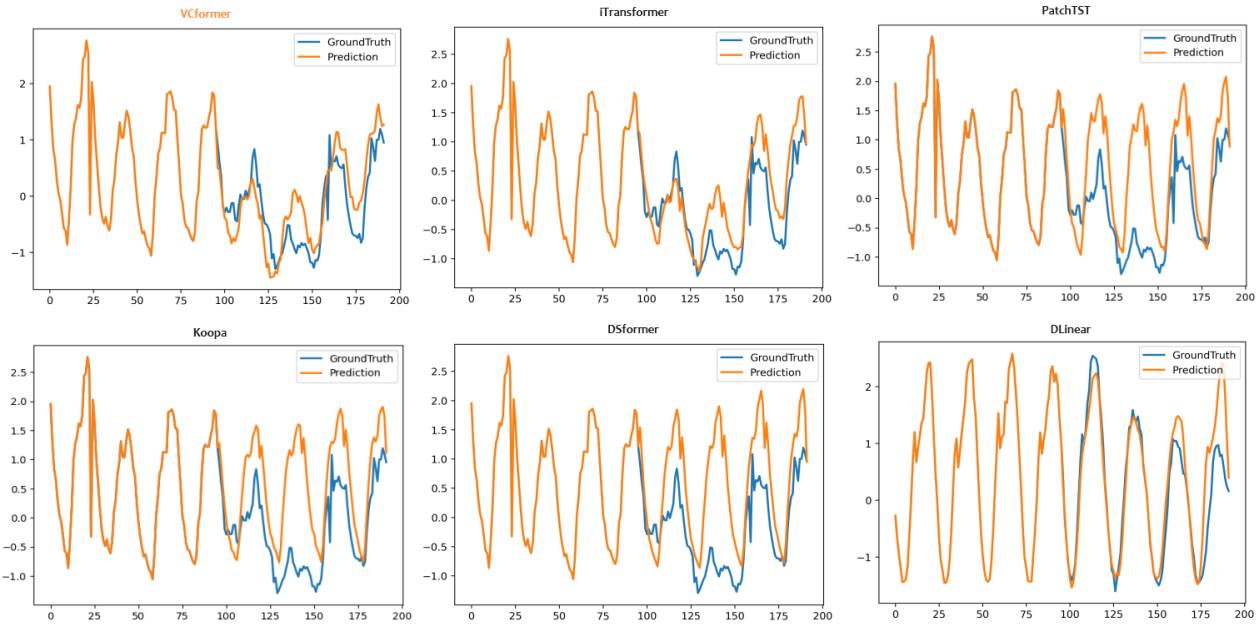


Figure 8: Visualization of input-96-predict-96 results on the Electricity dataset.