# Sonnet: Spectral Operator Neural Network for Multivariable Time Series Forecasting

**Yuxuan Shu, Vasileios Lampos**
Centre for Artificial Intelligence
Department of Computer Science
University College London
{yuxuan.shu.22, v.lampos}@ucl.ac.uk

**Abstract**

Multivariable time series forecasting methods can integrate information from exogenous variables, leading to significant prediction accuracy gains. The transformer architecture has been widely applied in various time series forecasting models due to its ability to capture long-range sequential dependencies. However, a naïve application of transformers often struggles to effectively model complex relationships among variables over time. To mitigate against this, we propose a novel architecture, termed **S**pectral **O**perator **N**eural **Net**work (**Sonnet**). Sonnet applies learnable wavelet transformations to the input and incorporates spectral analysis using the Koopman operator. Its predictive skill relies on the **M**ulti**v**ariable **C**oherence **A**ttention (**MVCA**), an operation that leverages spectral coherence to model variable dependencies. Our empirical analysis shows that Sonnet yields the best performance on 34 out of 47 forecasting tasks with an average mean absolute error (MAE) reduction of $2.2\%$ against the most competitive baseline. We further show that MVCA can remedy the deficiencies of naïve attention in various deep learning models, reducing MAE by $10.7\%$ on average in the most challenging forecasting tasks.

**Code & data** — https://github.com/ClaudiaShu/Sonnet
**Appendices** — https://arxiv.org/pdf/2505.15312

## 1 Introduction

Multivariable time series (MTS) forecasting methods learn from multiple input variables to predict a single target variable (Hidalgo and Goodman 2013). MTS models are deployed in many real-life applications, such as financial modelling (Niemira and Saaty 2004; Antunes et al. 2018; Taylor and Buizza 2003; Witt and Witt 1995), numerical weather prediction (Fildes and Kourentzes 2011; Young 2018; Dalton and Bekker 2022), and computational epidemiology (Dugas et al. 2013; Shaman and Karspeck 2012; da Silva et al. 2020; Morris et al. 2023). Recent machine learning models for time series forecasting tend to emphasise multivariate formulations (Nie et al. 2023; Zeng et al. 2023; Liu et al. 2023; Ansari et al. 2024; Das et al. 2024). While such models provide important insights for integrating traditional time series forecasting into deep learning techniques, e.g. capturing seasonality (Lin et al. 2024), frequency-domain methods (Zhou et al. 2022a,b), and autoregression (Zeng et al. 2023; Nie et al. 2023), they often overlook the benefits of leveraging external information through exogenous indicators.

In fact, empirical evidence suggests that capturing dependencies from external variables can increase the risk of overfitting for complex models (Nie et al. 2023). Moreover, existing methods that capture inter-variable dependencies (Zhang and Yan 2023; Wang et al. 2024a) have, in certain forecasting tasks, been outperformed by models that do not (Lin et al. 2024; Luo and Wang 2024a). However, this can be attributed to the choice of forecasting benchmarks, which, in many occasions, hold strong seasonal patterns (Lin et al. 2024; see also Appendix E), that can be effectively modelled using simpler autoregressive methods (Zeng et al. 2023; Lin et al. 2024). Additionally, models that do not capture inter-variable dependencies benefit from learning using longer look-back windows without significantly increasing modelling complexity and computational cost (Han, Ye, and Zhan 2023), making them less likely to overfit. Nevertheless, these empirical outcomes are not necessarily valid for multivariable forecasting scenarios, especially for tasks where exogenous variables have strong predictive power. In such cases, explicitly modelling interactions across variables facilitates better performance (Wang et al. 2024b; Shu and Lampos 2025).

Prior work has explored various ways in capturing inter-variable dependencies for multivariate or MTS forecasting tasks. Methods applying attention (Vaswani et al. 2017) across variables (Liu et al. 2024; Ilbert et al. 2024) can capture nonlinear dependencies, but disrupt temporal information by embedding sequences along the time dimension. Crossformer (Zhang and Yan 2023) attempts to rectify this by proposing a modified transformer structure where input is split into time series patches allowing to capture dependencies across both the time and variable dimensions at the sub-sequence level. ModernTCN (Luo and Wang 2024b), on the other hand, uses a convolutional kernel over both the variable and time dimension to jointly capture the inter- and intra-variable dependencies. However, both Crossformer and ModernTCN suffer from GPU computational overhead as the number of exogenous variables or the length of the input time series increases (Shu and Lampos 2025; Zhou et al. 2024). TimeXer (Wang et al. 2024b) models dependencies between exogenous variables and the target variable separately, and then joins their learned embeddings with a cross-attention

module. DeformTime (Shu and Lampos 2025) yields superior results by using deformable attention to incorporate information from exogenous variables at different time steps. However, it only captures inter-variable dependencies within the reception field of a convolutional kernel, which limits the exploration of a wider range of exogenous variables.

Frequency-based analysis techniques, including those using Fourier (Bochner 1953; Sorensen et al. 1987) or wavelet transform (Varanini et al. 1997; Zhang et al. 2003; Arneodo, Grasseau, and Holschneider 1988; Farge et al. 1992; Yu, Guo, and Sano 2024), has been widely used in statistical approaches for identifying periodic patterns (Priestley 2018) as well as in machine learning methods for forecasting (Lange, Brunton, and Kutz 2021; Zhou et al. 2022a,b; Piao et al. 2024). These methods compress temporal information (due to the Fourier transform) while some of them focus on capturing intra-variable dependencies (Zhou et al. 2022a,b). Wavelet transform, albeit dependent on the chosen mother wavelet (De Moortel, Munday, and Hood 2004; Ngui et al. 2013), can maintain both time and frequency information. In the frequency domain, spectral coherence (White and Boashash 1990) serves as a powerful tool for capturing the correlation between variables at different frequencies (Stein, French, and Holden 1972; Mima and Hallett 1999). To improve input projections into the frequency domain, Lange, Brunton, and Kutz (2021) used a Koopman operator (Mezić 2005; Rowley et al. 2009; Avila and Mezić 2020; Liu et al. 2023), i.e. a spectral function that enables linear modelling of nonlinear changes to better address nonlinearities. More recently, AdaWaveNet (Yu, Guo, and Sano 2024) decomposed the input series into seasonal and long-term components, and used the wavelet transform to capture periodic information. This method accounts for inter-variable dependencies only over the seasonal components.

Motivated by the aforementioned remarks, we propose **S**pectral **O**perator **N**eural **Net**work (Sonnet), a model that captures MTS dependencies in the spectral domain using a learnable wavelet transform. Sonnet captures both intra- and inter-variable dependencies using a novel frequency-domain **M**ulti**v**ariable **C**oherence **A**ttention (MVCA) layer. Furthermore, it deploys a learnable Koopman operator for linearised transitions of temporal states (Li et al. 2020). MVCA demonstrates stand-alone effectiveness when integrated into existing architectures, outperforming naïve attention and other modified attention mechanisms. Our key contributions are:

1. We propose **Sonnet**, a novel neural network architecture for MTS forecasting that captures inter-variable dependencies via adaptable time-frequency spectral operators while enforcing stability through learnable Koopman dynamics.
2. We introduce **MVCA**, an attention mechanism designed to model interactions between variables by leveraging their spectral coherence, a frequency-domain measurement of dependency. Unlike conventional self-attention, which computes pairwise similarity via dot products, MVCA captures temporal relationships through their cross-spectral density with the inclusion of frequency information from all variables, enhancing variable dependency modelling for MTS tasks.
3. We assess forecasting accuracy on carefully curated MTS data sets, including established benchmarks complemented by tasks on weather forecasting, influenza prevalence, electricity consumption, and energy prices. The weather prediction and influenza prevalence tasks support a more thorough evaluation as they include a substantial amount of exogenous variables, multiple years ($\geq 10$) for training, and multiple test periods ($\geq 3$).
4. Sonnet reduces mean absolute error (MAE) by $2.2\%$ on average compared to the most performant baseline model (stat. sig., $p < 10^{-3}$). In the more challenging tasks of influenza and weather modelling, MAE is reduced by $3.5\%$ and $2\%$, respectively. Performance gains persist as the forecasting horizon increases, demonstrating the effectiveness of Sonnet in longer-term forecasting.

## 2 MTS Forecasting Task Definition

We focus exclusively on multivariable time series (MTS) forecasting, whereby multiple input variables are used to predict a single target variable. We note that although for some baseline models multiple output variables may be present (multivariate forecasting), our evaluation is restricted to the prediction of one specific output. All models are trained and evaluated under a rolling window setup, with a look-back window and a forecasting horizon of $L$ and $H$ time steps, respectively. At each time step $t$, the $C$ observed exogenous variables over $L$ past time steps, $\{t-L+1,\ldots,t\}$, are captured in an input matrix $\mathbf{X}_t \in \mathbb{R}^{L\times C}$. The autoregressive signal for the target (endogenous) variable is denoted by $\mathbf{y}_{t-\delta} \in \mathbb{R}^L$; this encompasses time steps $\{t-\delta-L+1,\ldots,t-\delta\}$, where $\delta \in \mathbb{N}_0$ is an optional delay applied when the target variable is observed with a temporal lag. We capture the exogenous and endogenous input variables in $\mathbf{Z}_t = [\mathbf{X}_t, \mathbf{y}_{t-\delta}] \in \mathbb{R}^{L\times(C+1)}$. The goal is to predict the target variable at time step $t+H$, where $H$ denotes the forecasting horizon. Hence, the output of a forecasting model is denoted by $\mathbf{y}_{t+H} \in \mathbb{R}^H$ and holds forecasts for time steps $\{t+1,\ldots,t+H-1,t+H\}$. For models that conduct multivariate forecasting, the output includes predictions for all covariates and hence is denoted by $\mathbf{Y}_{t+H} = [\mathbf{X}_{t+H}, \mathbf{y}_{t+H}] \in \mathbb{R}^{H\times(C+1)}$. The forecasting task is to learn $f : \mathbf{Z}_t \to \mathbf{y}_{t+H}$ or $\mathbf{Y}_{t+H}$. Performance is measured based on the endogenous forecast at time step $t+H$, i.e. the last (temporally) element of $\mathbf{y}_{t+H}$, $y_{t+H} \in \mathbb{R}$. For notational simplicity, we omit temporal subscripts and use $\mathbf{Z}$ for $\mathbf{Z}_t$, $\mathbf{y}$ for $\mathbf{y}_{t-\delta}$, and $\mathbf{X}$ for $\mathbf{X}_t$.

## 3 Spectral Coherence with Sonnet

In this section, we provide a detailed description of our proposed model, Sonnet. It operates in the spectral domain via a learnable wavelet transform and introduces a **M**ulti**v**ariable **C**oherence **A**ttention (MVCA) module to capture both inter- and intra-variable dependencies. We further use a Koopman projection layer that enables stable temporal evolution via a learned linear operator.

### 3.1 Joint Embedding of Input Variables

Given an input matrix $\mathbf{Z} \in \mathbb{R}^{L\times(C+1)}$ that consists of the exogenous variables $\mathbf{X} \in \mathbb{R}^{L\times C}$ and endogenous variable $\mathbf{y} \in \mathbb{R}^L$, we first obtain the embeddings of $\mathbf{X}$ and $\mathbf{y}$ independently.

These are denoted by $\mathbf{E}_x \in \mathbb{R}^{L\times \alpha d}$ and $\mathbf{E}_y \in \mathbb{R}^{L\times(1-\alpha)d}$. They are derived using learnable weight matrices $\mathbf{W}_x \in \mathbb{R}^{C\times \alpha d}$ and $\mathbf{W}_y \in \mathbb{R}^{1\times(1-\alpha)d}$, as in $\mathbf{E}_x = \mathbf{X}\mathbf{W}_x$, where $d$ is the embedding dimension and $\alpha \in [0,1]$ a hyperparameter that controls the projected dimensionality of $\mathbf{X}$ and $\mathbf{y}$ in the final embedding.[1] By concatenating along the feature dimension, we obtain the final embedding $\mathbf{E} = [\mathbf{E}_x, \mathbf{E}_y] \in \mathbb{R}^{L\times d}$.

### 3.2 Learnable Wavelet Transform

We then transform the time series embedding into the wavelet space that contains both time and frequency information, to capture both fine-grained details and overall trends across various temporal resolutions (Mallat 1989; Daubechies 1990). Specifically, after obtaining the input time series embedding $\mathbf{E}$, we first define a set of $K$ learnable wavelet transformations (also referred to as atoms), with the $k$-th atom held in a matrix $\mathbf{M}_k \in \mathbb{R}^{d\times L}$ derived by

$$\mathbf{M}_k = \exp\left(-\mathbf{w}_\alpha \mathbf{t}^2\right) \times \cos\left(\mathbf{w}_\beta \mathbf{t} + \mathbf{w}_\gamma \mathbf{t}^2\right), \qquad (1)$$

where $\mathbf{t} \in \mathbb{R}^L$ is a row vector capturing normalised time steps with $\mathbf{t}_i = i/(L-1)$ for $i = 0,\ldots,L-1$, and $\mathbf{w}_\alpha$, $\mathbf{w}_\beta$, and $\mathbf{w}_\gamma \in \mathbb{R}^d$ are learnable weight vectors that control the shape of the wavelet, each initialised randomly from a normal distribution. In particular, $\mathbf{w}_\alpha$ controls the width of the Gaussian envelope, and $\mathbf{w}_\beta$, $\mathbf{w}_\gamma$ respectively determine the linear or quadratic frequency modulation of the generated cosine waveforms. This formulation enables the atoms to adapt to localised time-frequency structures in the data. The time series embedding $\mathbf{E} \in \mathbb{R}^{L\times d}$ is then transformed into the wavelet space by projecting it onto each of the wavelet atoms $\mathbf{M}_k \in \mathbb{R}^{d\times L}$ using $\mathbf{P}_k = \mathbf{E} \odot \mathbf{M}_k^\top$, where $\mathbf{P}_k \in \mathbb{R}^{L\times d}$ denotes the embedding's projection for the $k$-th atom, and $\odot$ is element-wise multiplication. The transformed wavelet across all $K$ atoms is denoted by $\mathbf{P} \in \mathbb{R}^{K\times L\times d}$. The aforementioned steps help to preserve temporal structure while decomposing the input into multi-resolution time-frequency components using adaptive wavelet transforms that can capture both short- and long-term patterns in the data.

### 3.3 Multivariable Coherence Attention (MVCA)

In MTS forecasting, input variables can be both auto-correlated (to their own past values) and cross-correlated (to each other). To improve learning from these correlations, we propose MVCA, a module that supplements the standard attention. MVCA can capture inter- and intra-variable dependencies within the frequency domain using spectral density coherence. Its premise is that variables with a higher spectral coherence should contribute more to the attention output. MVCA can be used in place of any naïve attention module used in a forecasting method.

Given the obtained input embedding matrix in wavelet space $\mathbf{P} \in \mathbb{R}^{K\times L\times d}$, where $L$ and $d$ are the time and variable dimension, $K$ is the number of wavelets after transformation, we first linearly project it to query, key and value embeddings, denoted as $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V} \in \mathbb{R}^{K\times L\times d}$, using weight matrices $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v \in \mathbb{R}^{d\times d}$ respectively, as in $\mathbf{Q} = \mathbf{P}\mathbf{W}_q$. We then consider the embeddings from different wavelet transformations (indexed by $k$) as separate attention heads. Specifically, for each attention head, we obtain the sub-tensor of $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, denoted as $\mathbf{Q}_h$, $\mathbf{K}_h$, and $\mathbf{V}_h \in \mathbb{R}^{L\times d}$, as the query, key, and value embeddings. Therefore, each head captures a distinct subspace of the original embedding, where subspaces correspond to different wavelet transformations of the input. Including a multi-head structure enables the model to learn diverse dependencies in parallel (Vaswani et al. 2017).

We then apply a Fast Fourier Transform (FFT) along the variable dimension (Dudgeon and Mersereau 1984) of each transformer head to transfer the query and key embeddings to the frequency domain, i.e. $\mathbf{Q}_f = \mathrm{FFT}(\mathbf{Q}_h)$ and $\mathbf{K}_f = \mathrm{FFT}(\mathbf{K}_h)$, with both $\mathbf{Q}_f$ and $\mathbf{K}_f \in \mathbb{C}^{L\times \ell}$, where $\ell = \lfloor \frac{d}{2} \rfloor + 1$. Each frequency bin in the transformed data ($\mathbf{Q}_f$, $\mathbf{K}_f$) contains information from all original inputs to the FFT (Lee-Thorp et al. 2022), capturing both fine-grained (high-frequency) and global (low-frequency) patterns. The cross-spectral density $\mathbf{P}_{qk} \in \mathbb{C}^{L\times \ell}$ and power-spectral densities $\mathbf{P}_{qq}$, $\mathbf{P}_{kk} \in \mathbb{R}^{L\times \ell}$ are obtained as follows:

$$\mathbf{P}_{qk} = \mathbf{Q}_f \odot \mathbf{K}_f^*,\ \mathbf{P}_{qq} = \mathbf{Q}_f \odot \mathbf{Q}_f^*,\ \mathbf{P}_{kk} = \mathbf{K}_f \odot \mathbf{K}_f^*, \quad (2)$$

where '$*$' denotes the complex conjugate. We average along the second dimension to obtain $\overline{\mathbf{P}}_{qk} \in \mathbb{C}^L$, $\overline{\mathbf{P}}_{qq}$, and $\overline{\mathbf{P}}_{kk} \in \mathbb{R}^L$ (see also Appendix B). The normalised spectral coherence $\mathbf{C}_{qk} \in \mathbb{R}^L$ is then computed using

$$\mathbf{C}_{qk} = |\overline{\mathbf{P}}_{qk}|^2 / \left(\overline{\mathbf{P}}_{qq} \cdot \overline{\mathbf{P}}_{kk} + \epsilon\right), \qquad (3)$$

where $\epsilon = 10^{-6}$ mitigates division by 0. $\mathbf{C}_{qk}$ captures the linear dependency between sequences across frequency bands. Therefore, the coherence here assigns higher importance to the time steps where the query and key hold more similar averaged values across multiple frequencies.

Following the common design of attention layers (Vaswani et al. 2017), we scale, normalise, and regularise $\mathbf{C}_{qk}$, i.e. $\mathbf{A}_h = \mathrm{Dropout}(\mathrm{Softmax}(\mathbf{C}_{qk}/\sqrt{d}))$. The attention weights $\mathbf{A}_h \in \mathbb{R}^L$ of each head are first broadcast along the feature dimension to form $\mathbf{A} \in \mathbb{R}^{L\times d}$, which is then multiplied with the value representations $\mathbf{V}_h$ (element-wise) to produce head-specific outputs, $\mathbf{O}_h \in \mathbb{R}^{L\times d} = \mathbf{A} \odot \mathbf{V}_h$. We concatenate these outputs across all heads to obtain $\mathbf{O}_r \in \mathbb{R}^{K\times L\times d}$. We then use a 2-layer perceptron (MLP, $d$-dimensional layers) with Gaussian Error Linear Unit (GELU) activation to further capture nonlinearities. This is connected to a residual layer to form the output $\mathbf{O}_m \in \mathbb{R}^{K\times L\times d} = \mathbf{O}_r + \mathrm{MLP}(\mathbf{O}_r)$. We then multiply $\mathbf{O}_m$ with a weight matrix $\mathbf{W}_{\text{out}} \in \mathbb{R}^{d\times d}$ to obtain the output of MVCA, $\mathbf{O} \in \mathbb{R}^{K\times L\times d} = \mathbf{O}_m \mathbf{W}_{\text{out}}$.

### 3.4 Koopman-Guided Spectrum Evolvement

Motivated by Koopman operator theory (Mezić 2005; Rowley et al. 2009; Avila and Mezić 2020), which offers a framework for modelling nonlinear dynamics, we introduce a layer to capture the temporal evolution of time-frequency patterns in

[1] We make sure that both products $\alpha d$ and $(1-\alpha)d \in \mathbb{N}$ to avoid dimension mismatch caused by rounding. For $\alpha = 0$, the forecasting is based on the historical values of the target variable only, collapsing to an autoregressive setting. For $\alpha = 1$, forecasting depends entirely on the exogenous variables.

wavelet space. We aim to learn a Koopman operator with $K$ dimensions in the transformed space. To obtain the operator $\mathbf{K} \in \mathbb{C}^{K \times K}$, we first initialise a learnable complex-valued matrix $\mathbf{S} \in \mathbb{C}^{K \times K}$. At each forward pass, we apply QR decomposition to $\mathbf{S}$, and retain only the resulting unitary matrix $\mathbf{U} \in \mathbb{C}^{K \times K}$, i.e. $\mathbf{U} \leftarrow \mathrm{QR}(\mathbf{S})$ s.t. $\mathbf{U}^\dagger \mathbf{U} = \mathbf{I}$, where $\mathbf{U}^\dagger$ is the conjugate transpose of $\mathbf{U}$. Multiplying with $\mathbf{U}$ therefore prevents data amplification or distortion.

We then initialise a learnable vector $\mathbf{p} \in \mathbb{R}^K$, where the $k$-th element, $p_k$, controls the temporal evolution (phase transformation angle) of the $k$-th transformation. All elements in the vector are then mapped into complex numbers, obtaining $\mathbf{v} \in \mathbb{C}^K$, where $v_k = e^{ip_k}$. We use $\mathbf{v}$ to form a diagonal matrix $\mathbf{D} \in \mathbb{C}^{K \times K} = \mathrm{diag}(\mathbf{v})$. The Koopman operator, $\mathbf{K}$, is then given by $\mathbf{K} = \mathbf{U}\mathbf{D}\mathbf{U}^\dagger$. We use $\mathbf{K}$ to model the temporal evolution of the MVCA embedding, $\mathbf{O}$, after converting it to a complex form, $\mathbf{O}_c \in \mathbb{C}^{K \times L \times d}$ (the imaginary part is set to $i0$ to perform complex-valued transformations without altering the original embedding), given by $\mathbf{O}_l = \mathbf{K} \times \mathbf{O}_c$, where $\mathbf{O}_l \in \mathbb{C}^{K \times L \times d}$ is the evolved embedding after multiplication with $\mathbf{K}$. A conventional Koopman framework models temporal evolution recursively at each time step (Lusch, Kutz, and Brunton 2018; Avila and Mezić 2020). We instead choose to apply Koopman transformation in the frequency domain with one forward pass, which is equivalent to learning a direct transformation from the input to the output. This global projection reduces the accumulation of sequential errors while maintaining robustness in training.

### 3.5 Sequence Reconstruction from Wavelets

The inverse transformation reconstructs the original sequence by aggregating weights from each wavelet atom. Given the evolved state $\mathbf{O}_l$, we first obtain its real part denoted as $\mathbf{O}_r \in \mathbb{R}^{K \times L \times d}$. For each wavelet atom indexed by $k$, let $\mathbf{O}_k \in \mathbb{R}^{L \times d}$ denote the corresponding slice of $\mathbf{O}_r$. We then multiply it with the wavelet atom $\mathbf{M}_k \in \mathbb{R}^{d \times L}$, i.e. $\mathbf{R}_k = \mathbf{O}_k \odot \mathbf{M}_k^\top$. $\mathbf{R}_k \in \mathbb{R}^{L \times d}$ denotes the series reconstruction from the $k$-th atom. This operation is conducted over all $K$ atoms. Joining all the reconstructed series forms a $K \times L \times d$ matrix. We sum over its first dimension to obtain the reconstructed embedding, denoted as $\mathbf{R} \in \mathbb{R}^{L \times d}$.

### 3.6 Convolutional Decoder

Finally, a 3-layer convolutional decoder transforms the learned representation. It comprises 3 1-dimensional convolutional layers with GELU activations between every 2 layers. Each layer uses kernel sizes $[5,3,3]$ and paddings $[2,1,1]$ respectively, followed by an adaptive average pooling layer at the end. The dimension of each layer is $[H \times 4, H \times 2, H]$, producing the final sequence representation $\mathbf{Z}_{\text{out}} \in \mathbb{R}^{H \times H}$, in accordance with the time steps of the target forecasting horizon. The result is then linearly projected using a weight vector $\mathbf{W}_z \in \mathbb{R}^H$ to generate the final output, as in $\hat{\mathbf{y}} \in \mathbb{R}^H = \mathbf{Z}_{\text{out}} \mathbf{W}_z$, in accordance with the dimensionality of the target variable.

## 4 Results

We assess forecasting accuracy using an expanded collection of data sets and tasks, to overcome potential biases present in the current literature. We first compare Sonnet against other competitive baseline models. We then investigate the role of attention mechanisms in time series forecasting models by removing or replacing naïve transformers with more advanced variants, including the proposed MVCA module. We also provide an ablation study of the key components of Sonnet and seed control in Appendix E.

### 4.1 Experiment Settings

We conduct experiments over 12 real-world data sets. This includes 2 established benchmarks from prior papers (Zhou et al. 2021; Zeng et al. 2023), specifically the ETTh1 and ETTh2 data sets, which contain hourly electricity transformer temperature forecasting. Oil temperature is our target variable, with the remaining indicators considered as exogenous variables, following Wang et al. (2024b). We also use 2 data sets from the Darts library (Herzen et al. 2022), predicting hourly energy prices (ENER) and electricity (low-voltage) consumption (ELEC). In addition, we form weather data sets extracted from the WeatherBench repository (Rasp et al. 2020) for 5 cities from diverse geographical locations, namely London (WEA-LD), New York (WEA-NY), Hong Kong (WEA-HK), Cape Town (WEA-CT), and Singapore (WEA-SG), to support a more inclusive analysis. For each city, we sample data from its nearest grid point and obtain 5 climate indicators. We provide spatial context by including data from its eight surrounding grid points ($3 \times 3$ grid) as additional exogenous variables. Following prior work on global climate forecasting (Verma, Heinonen, and Garg 2024), we resample the data to a temporal resolution of 6 hours. The forecasting target is the 850 hPa (T850) temperature, a key indicator for climate modelling (Scherrer et al. 2004; Hamill and Whitaker 2007). Finally, we include influenza-like illness (ILI) rate forecasting tasks (as in (Shu and Lampos 2025)) in 3 locations, England (ILI-ENG), and in U.S. Health & Human Services (HHS) Regions 2 (ILI-US2) and 9 (ILI-US9). In the ILI tasks, frequency time series of web searches are included as exogenous predictors. More information about the data sets is provided in Appendix A.

For the ETT tasks, we set the forecasting horizon ($H$) to $\{96, 192, 336, 720\}$ time steps, and use a single test set of consecutive unseen instances adopting the evaluation settings in prior work (Nie et al. 2023; Liu et al. 2024). For ENER we set $H = \{24, 48, 72, 168\}$ hours ahead and use 1 year (2018) for testing. For ELEC, we use the last 2 years (2020, '21) as 2 distinct test seasons, and set $H$ to $\{12, 24, 36\}$ hours. For the WEA tasks, we form 3 test sets (years 2016, '17, '18), and set $H$ to $\{4, 12, 28, 120\}$ time steps corresponding to $\{1, 3, 7, 30\}$ days. Following the same setup as in (Shu and Lampos 2025), for the ILI forecasting task, we test models on 4 consecutive influenza seasons (2015/16 to 2018/19), each time training a new model on data from previous seasons. We set $H = \{7, 14, 21, 28\}$ days (more details in Appendix D).

We compare Sonnet to 8 competitive forecasting models that, to the best of our knowledge, form the current SOTA methods: DLinear (Zeng et al. 2023), Crossformer (Zhang and Yan 2023), iTransformer (Liu et al. 2024), PatchTST (Nie et al. 2023), TimeXer (Wang et al. 2024b), Samformer (Ilbert et al. 2024), ModernTCN (Luo and Wang 2024b), and

| Task | | Sonnet | | DeformTime | | ModernTCN | | Samformer | | TimeXer | | PatchTST | | iTransformer | | Crossformer | | DLinear | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $H$ | MAE | $\epsilon$% | MAE | $\epsilon$% | MAE | $\epsilon$% | MAE | $\epsilon$% | MAE | $\epsilon$% | MAE | $\epsilon$% | MAE | $\epsilon$% | MAE | $\epsilon$% | MAE | $\epsilon$% |
| ELEC | 12 | **0.1040** | **24.95** | 0.1162 | 26.68 | 0.1596 | 36.81 | 0.2336 | 53.84 | 0.1287 | 28.45 | 0.1419 | 30.76 | 0.1468 | 31.91 | 0.1513 | 30.35 | 0.3307 | 71.51 |
| | 24 | **0.1203** | **27.70** | 0.1359 | 30.11 | 0.1806 | 39.70 | 0.2520 | 56.39 | 0.1586 | 33.20 | 0.1545 | 34.04 | 0.1588 | 35.40 | 0.1873 | 35.60 | 0.3369 | 73.23 |
| | 36 | **0.1389** | **30.21** | 0.1729 | 35.81 | 0.2065 | 43.79 | 0.2866 | 62.40 | 0.1785 | 35.33 | 0.1659 | 33.85 | 0.1791 | 38.18 | 0.2273 | 39.36 | 0.3908 | 82.38 |
| ENER | 24 | **0.3621** | **65.67** | 0.3717 | 66.92 | 0.3752 | 66.53 | 0.3703 | 66.01 | 0.3733 | 69.59 | 0.3717 | 65.89 | 0.3838 | 68.36 | 0.3716 | 67.44 | 0.4307 | 76.49 |
| | 48 | **0.4120** | **71.07** | 0.4299 | 74.59 | 0.4154 | 71.74 | 0.4294 | 72.34 | 0.4404 | 77.61 | 0.4467 | 75.60 | 0.4386 | 75.63 | 0.4647 | 81.79 | 0.5138 | 89.79 |
| | 72 | **0.4036** | **70.52** | 0.4217 | 74.50 | 0.4089 | 70.84 | 0.4353 | 73.40 | 0.4246 | 74.70 | 0.4303 | 73.90 | 0.4276 | 74.19 | 0.4473 | 78.62 | 0.5124 | 89.26 |
| | 168 | **0.3980** | **68.63** | 0.4399 | 78.98 | 0.4497 | 74.86 | 0.4243 | 72.28 | 0.4493 | 76.76 | 0.4872 | 81.34 | 0.4352 | 74.11 | 0.4168 | 76.76 | 0.5070 | 87.87 |
| ETTh1 | 96 | 0.2145 | 16.01 | **0.1941** | **14.96** | 0.2047 | 15.66 | 0.2047 | 15.73 | 0.2135 | 16.03 | 0.2017 | 15.41 | 0.2052 | 15.46 | 0.2126 | 16.52 | 0.2599 | 20.82 |
| | 192 | 0.2330 | 17.61 | **0.2116** | **16.08** | 0.2417 | 18.32 | 0.2307 | 17.64 | 0.2322 | 16.96 | 0.2409 | 18.29 | 0.2429 | 18.13 | 0.2820 | 21.63 | 0.3798 | 31.78 |
| | 336 | 0.2392 | 18.91 | **0.2158** | **16.27** | 0.2415 | 18.52 | 0.2523 | 18.94 | 0.2414 | 17.34 | 0.2559 | 19.29 | 0.2593 | 19.11 | 0.2947 | 22.65 | 0.6328 | 58.34 |
| | 720 | 0.2768 | 19.73 | 0.2862 | 21.81 | 0.2785 | 20.44 | 0.3026 | 23.21 | **0.2617** | **18.58** | 0.3087 | 23.89 | 0.2886 | 22.05 | 0.3350 | 24.84 | 0.7563 | 69.52 |
| ETTh2 | 96 | **0.3098** | **33.04** | 0.3121 | 40.07 | 0.3199 | 40.68 | 0.3312 | 40.16 | 0.3346 | 41.04 | 0.3145 | 39.25 | 0.3420 | 42.41 | 0.3486 | 40.71 | 0.3349 | 41.68 |
| | 192 | 0.3742 | 39.58 | **0.3281** | **37.90** | 0.3887 | 47.08 | 0.3874 | 45.37 | 0.4154 | 47.07 | 0.3839 | 45.45 | 0.4233 | 47.44 | 0.4035 | 43.16 | 0.4084 | 50.67 |
| | 336 | 0.3689 | 39.48 | **0.3450** | **37.00** | 0.3904 | 50.54 | 0.4083 | 46.41 | 0.4041 | 42.26 | 0.4018 | 46.77 | 0.4332 | 45.95 | 0.4487 | 49.44 | 0.4710 | 55.53 |
| | 720 | 0.4335 | 51.62 | **0.3640** | **34.99** | 0.5728 | 63.04 | 0.5198 | 58.44 | 0.5135 | 56.17 | 0.4960 | 55.27 | 0.4565 | 45.40 | 0.5832 | 61.45 | 0.7981 | 94.67 |
| ILI-ENG | 7 | **1.4791** | **21.84** | 1.6417 | 28.60 | 1.9489 | 28.27 | 2.3475 | 28.31 | 2.8084 | 33.66 | 2.3115 | 27.61 | 2.3084 | 26.38 | 1.8698 | 25.70 | 2.8214 | 43.02 |
| | 14 | **1.9225** | **25.77** | 2.2308 | 33.98 | 2.7050 | 36.01 | 3.0290 | 36.63 | 3.4937 | 41.88 | 3.2547 | 37.76 | 3.2301 | 36.67 | 2.6543 | 30.97 | 3.7922 | 55.29 |
| | 21 | **2.5101** | 36.53 | 2.6500 | **32.70** | 3.0400 | 40.02 | 4.4980 | 54.41 | 4.3337 | 51.57 | 4.3192 | 51.11 | 4.2347 | 48.93 | 3.0014 | 40.57 | 4.4739 | 61.25 |
| | 28 | 2.7481 | **36.95** | **2.7228** | 40.44 | 3.3611 | 47.87 | 5.1598 | 60.78 | 4.9013 | 61.60 | 4.9964 | 59.60 | 4.8125 | 55.35 | 3.1983 | 46.15 | 5.0347 | 67.75 |
| ILI-US2 | 7 | **0.3806** | **14.89** | 0.4122 | 16.01 | 0.4398 | 16.55 | 0.6495 | 24.21 | 0.6083 | 23.38 | 0.7097 | 24.51 | 0.6507 | 23.24 | 0.4400 | 16.46 | 0.7355 | 27.94 |
| | 14 | **0.4491** | 18.38 | 0.4752 | **17.73** | 0.5279 | 20.22 | 0.7696 | 30.16 | 0.7725 | 29.07 | 0.8635 | 30.11 | 0.7896 | 28.17 | 0.5852 | 20.98 | 0.8435 | 32.22 |
| | 21 | **0.5326** | **20.64** | 0.5425 | 22.12 | 0.5781 | 23.85 | 0.8374 | 31.42 | 0.8243 | 31.46 | 1.0286 | 36.70 | 0.8042 | 30.03 | 0.6245 | 22.29 | 0.9124 | 34.93 |
| | 28 | 0.5788 | **21.15** | **0.5538** | 22.25 | 0.5710 | 23.66 | 0.9389 | 36.80 | 0.9074 | 34.72 | 1.1525 | 42.61 | 0.9619 | 36.75 | 0.6512 | 23.47 | 0.9999 | 38.46 |
| ILI-US9 | 7 | 0.2668 | 12.98 | **0.2622** | **12.26** | 0.2899 | 14.17 | 0.4025 | 19.39 | 0.3813 | 18.21 | 0.4116 | 19.34 | 0.4057 | 18.57 | 0.3149 | 14.44 | 0.4675 | 23.47 |
| | 14 | **0.2806** | **13.10** | 0.3084 | 13.80 | 0.3417 | 15.29 | 0.5257 | 24.50 | 0.4665 | 22.14 | 0.5020 | 24.09 | 0.4702 | 22.44 | 0.3571 | 17.23 | 0.5467 | 27.35 |
| | 21 | **0.3179** | **14.11** | 0.3179 | 14.23 | 0.3710 | 15.43 | 0.5415 | 24.22 | 0.5715 | 27.43 | 0.5935 | 29.40 | 0.5106 | 24.11 | 0.3418 | 15.90 | 0.6001 | 29.66 |
| | 28 | 0.3675 | 16.53 | **0.3532** | **15.75** | 0.3940 | 17.19 | 0.6050 | 27.95 | 0.6555 | 31.32 | 0.6665 | 33.35 | 0.6498 | 31.05 | 0.3747 | 16.44 | 0.6564 | 32.16 |
| WEA-CT | 4 | **1.6240** | 9.63 | 1.7600 | 10.41 | 1.8752 | 11.05 | 2.2265 | 13.09 | 2.2376 | 13.14 | 3.5004 | 20.17 | 2.1906 | 12.83 | 1.6382 | **9.62** | 3.1483 | 18.22 |
| | 12 | **3.5432** | **20.38** | 3.5681 | 20.46 | 3.7761 | 21.67 | 4.0444 | 23.14 | 3.7241 | 21.31 | 4.1910 | 23.99 | 3.9741 | 22.83 | 3.5932 | 20.62 | 4.0265 | 22.94 |
| | 28 | **3.7277** | **21.32** | 3.7601 | 21.49 | 3.9399 | 22.36 | 3.9239 | 22.37 | 3.8325 | 21.87 | 3.9405 | 22.48 | 3.9584 | 22.60 | 3.8061 | 21.72 | 3.9254 | 22.37 |
| | 120 | **3.7373** | **21.35** | 3.8040 | 21.67 | 4.0889 | 23.54 | 3.9018 | 22.30 | 3.8412 | 21.88 | 4.0547 | 23.02 | 3.9473 | 22.49 | 3.7659 | 21.49 | 4.0570 | 23.05 |
| WEA-HK | 4 | **0.6389** | **4.05** | 0.6804 | 4.32 | 0.7004 | 4.42 | 0.8097 | 5.10 | 0.8648 | 5.52 | 1.1488 | 7.15 | 0.8048 | 5.08 | 0.6488 | 4.11 | 0.9898 | 6.21 |
| | 12 | **1.2355** | **7.74** | 1.2786 | 7.99 | 1.3555 | 8.43 | 1.4006 | 8.70 | 1.3027 | 8.13 | 1.5825 | 9.77 | 1.4225 | 8.86 | 1.2896 | 8.08 | 1.5464 | 9.62 |
| | 28 | **1.4135** | **8.83** | 1.4746 | 9.16 | 1.5866 | 9.82 | 1.6226 | 10.05 | 1.5115 | 9.39 | 1.6441 | 10.16 | 1.6356 | 10.08 | 1.5282 | 9.45 | 1.6232 | 10.06 |
| | 120 | 1.5469 | 9.58 | **1.5399** | **9.45** | 1.6326 | 10.22 | 1.8843 | 12.06 | 1.7092 | 10.55 | 2.0084 | 12.84 | 1.6745 | 10.45 | 1.5931 | 9.73 | 1.8329 | 11.49 |
| WEA-LD | 4 | **1.7231** | **15.16** | 1.8753 | 16.31 | 1.9456 | 16.88 | 2.1537 | 18.53 | 2.2628 | 19.25 | 2.7602 | 22.93 | 2.1509 | 18.55 | 1.7447 | 15.26 | 2.5065 | 20.94 |
| | 12 | **2.9589** | **23.88** | 3.0214 | 24.15 | 3.2056 | 25.58 | 3.3070 | 26.61 | 3.1625 | 25.25 | 3.5406 | 28.33 | 3.3622 | 27.36 | 3.0492 | 24.35 | 3.3927 | 26.70 |
| | 28 | **3.2161** | **25.49** | 3.2724 | 25.84 | 3.5067 | 27.48 | 3.5841 | 27.97 | 3.4672 | 27.23 | 3.7365 | 29.42 | 3.6884 | 29.14 | 3.3048 | 25.88 | 3.6073 | 28.16 |
| | 120 | **3.2464** | **25.82** | 3.2973 | 26.17 | 3.8434 | 29.92 | 3.8420 | 30.32 | 3.6557 | 28.52 | 4.2344 | 32.40 | 3.8518 | 30.36 | 3.3935 | 26.75 | 3.9640 | 30.35 |
| WEA-NY | 4 | **1.2716** | **11.94** | 1.4028 | 13.03 | 1.4154 | 12.95 | 1.6003 | 14.44 | 1.7290 | 15.57 | 2.1644 | 19.24 | 1.6066 | 14.40 | 1.2935 | 12.24 | 1.9782 | 17.88 |
| | 12 | 2.4476 | 21.23 | **2.4453** | **21.04** | 2.6221 | 22.93 | 2.7069 | 23.57 | 2.6537 | 22.85 | 2.8592 | 24.81 | 2.6609 | 23.09 | 2.4494 | 21.11 | 2.9507 | 25.29 |
| | 28 | **2.6744** | **23.10** | 2.7450 | 23.20 | 2.9336 | 25.37 | 3.0347 | 26.18 | 2.8775 | 24.48 | 3.0956 | 27.01 | 3.0204 | 25.30 | 2.7830 | 23.66 | 3.2099 | 27.07 |
| | 120 | **2.7135** | **23.15** | 2.8224 | 23.86 | 3.3000 | 28.05 | 3.5289 | 33.01 | 3.1501 | 26.55 | 3.4086 | 28.78 | 3.1029 | 27.74 | 2.9615 | 24.69 | 3.6129 | 29.75 |
| WEA-SG | 4 | **0.3444** | **1.25** | 0.3557 | 1.30 | 0.3624 | 1.32 | 0.3925 | 1.43 | 0.3801 | 1.38 | 0.4238 | 1.54 | 0.3868 | 1.41 | 0.3532 | 1.29 | 0.4048 | 1.47 |
| | 12 | **0.4160** | **1.51** | 0.4256 | 1.55 | 0.4493 | 1.64 | 0.4784 | 1.74 | 0.4447 | 1.62 | 0.4992 | 1.82 | 0.4662 | 1.70 | 0.4359 | 1.59 | 0.4764 | 1.73 |
| | 28 | **0.4653** | **1.69** | 0.4875 | 1.77 | 0.5196 | 1.89 | 0.5421 | 1.97 | 0.4974 | 1.81 | 0.5542 | 2.02 | 0.5307 | 1.93 | 0.5003 | 1.82 | 0.5284 | 1.92 |
| | 120 | **0.4830** | **1.76** | 0.5050 | 1.83 | 0.5321 | 1.94 | 0.5221 | 1.90 | 0.5215 | 1.90 | 0.5357 | 1.95 | 0.5233 | 1.90 | 0.5212 | 1.89 | 0.5328 | 1.94 |

Table 1: Forecasting accuracy results across all tasks, methods, and forecasting horizons ($H$). For the ELEC/ILI/WEA tasks, we report the average performance across 2, 4, and 3 test sets, respectively (detailed breakdowns in Appendix E.1). $\epsilon$% denotes sMAPE. Best results are **bolded**, second best underlined. Grey background denotes the model does not outperform persistence.

DeformTime (Shu and Lampos 2025), with TimeXer and DeformTime being designed for multivariable forecasting. We also include a naïve baseline which can either be a seasonal or a standard persistence model, depending on the presence of strong seasonality. For all tasks (except ETT), we conduct hyperparameter tuning for all models. For the ETT tasks, we adopt settings from the official repositories of methods (except for Crossformer, which did not provide this). Appendix C has further details about the baseline models.

## 4.2 Forecasting Accuracy of Sonnet

Prediction accuracy for all tasks is enumerated in Table 1, using Mean Absolute Error (MAE) and symmetric Mean Absolute Percentage Error (sMAPE or $\epsilon$%) as the main evaluation

| | Attention | $H=7$ days | | | $H=14$ days | | | $H=21$ days | | | $H=28$ days | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **ENG** | **US2** | **US9** | **ENG** | **US2** | **US9** | **ENG** | **US2** | **US9** | **ENG** | **US2** | **US9** |
| iTransformer | — | 2.3084 | 0.6507 | 0.4057 | 3.2301 | 0.7896 | 0.4702 | 4.2347 | 0.8042 | 0.5106 | 4.8125 | 0.9619 | 0.6498 |
| | ¬ Attn | 2.3011 | 0.7242 | 0.4475 | 3.2557 | 0.8696 | 0.5344 | 4.4729 | 1.0105 | 0.6207 | 5.1864 | 1.1667 | 0.7179 |
| | FNet | 2.5534 | 0.7665 | 0.4741 | 3.5337 | 0.8746 | 0.5550 | 4.6970 | 1.0006 | 0.6066 | 5.3153 | 1.1118 | 0.7139 |
| | FED | 3.6150 | 0.9286 | 0.5613 | 4.3098 | 1.0750 | 0.6527 | 5.8596 | 1.4140 | 0.8669 | 6.4196 | 1.5167 | 0.9397 |
| | VDAB | 2.4287 | 0.6772 | 0.4126 | 3.3093 | 0.8274 | 0.4982 | **3.9406** | 0.8485 | 0.4992 | **4.4975** | 0.9068 | 0.5790 |
| | MVCA | **2.2707** | **0.5483** | **0.3581** | **3.1233** | **0.7267** | **0.4454** | 4.1780 | **0.7578** | **0.4880** | 4.8705 | **0.8885** | **0.5409** |
| SamFormer | — | 2.3475 | 0.6495 | 0.4025 | 3.0290 | 0.7696 | 0.5257 | 4.4980 | 0.8374 | 0.5415 | 5.1598 | 0.9389 | 0.6050 |
| | ¬ Attn | 2.5010 | 0.8245 | 0.5131 | 3.3015 | 0.9679 | 0.6047 | 4.1830 | 1.1396 | 0.7389 | 5.0136 | 1.2391 | 0.8347 |
| | FNet | 2.4812 | 0.7765 | 0.4699 | 3.2054 | 0.9202 | 0.5865 | 4.3909 | 0.9266 | 0.6443 | 5.0592 | 1.0983 | 0.7149 |
| | FED | 3.5217 | 0.9140 | 0.5599 | 4.2109 | 1.0519 | 0.6508 | 5.7800 | 1.3439 | 0.8519 | 5.9809 | 1.4348 | 0.9190 |
| | VDAB | **2.0406** | 0.5755 | 0.3651 | **2.9525** | 0.7281 | 0.4505 | 4.1080 | 0.7883 | **0.4861** | 4.9015 | 0.8617 | 0.5605 |
| | MVCA | 2.1365 | **0.5514** | **0.3609** | 3.2046 | **0.6935** | **0.4379** | **3.8033** | **0.7294** | 0.4975 | **4.7438** | **0.8159** | **0.5345** |
| PatchTST | — | 2.3115 | 0.7097 | 0.4116 | 3.2547 | 0.8635 | 0.5020 | 4.3192 | 1.0286 | 0.5935 | 4.9964 | 1.1525 | 0.6665 |
| | ¬ Attn | 2.4723 | 0.7702 | 0.4585 | 3.6415 | 0.9017 | 0.5551 | 4.2998 | 1.0657 | 0.6472 | 4.8538 | 1.1704 | 0.7313 |
| | FNet | 2.8097 | 0.8033 | 0.4591 | 4.0342 | 0.9322 | 0.5327 | 4.7975 | 1.0341 | 0.5430 | 4.9718 | 1.1087 | 0.6279 |
| | FED | 3.6158 | 0.9292 | 0.5614 | 4.6572 | 1.0765 | 0.6542 | 5.8725 | 1.4043 | 0.8671 | 6.3728 | 1.4953 | 0.9306 |
| | VDAB | 2.0799 | 0.5925 | 0.3820 | **3.0211** | 0.7722 | 0.4413 | **3.8164** | 0.8009 | 0.5159 | 4.6044 | **0.8518** | 0.5801 |
| | MVCA | **2.0054** | **0.5824** | **0.3705** | 3.0411 | **0.7406** | **0.4291** | 3.8627 | **0.7871** | **0.4746** | **4.5712** | 0.8765 | **0.5491** |

Table 2: Performance (average MAE across 4 test seasons) of iTransformer, Samformer, and PatchTST on the ILI forecasting tasks (ILI-ENG/US2/US9) with different modifications to the naïve attention mechanism. '¬ Attn' denotes the removal of the residual attention module, and FNet / FED / VDAB refer to using the attention modules proposed in FNet (2025), FEDformer (2022a), and DeformTime (2025), respectively. Best results are **bolded**, second best underlined.

metrics. For WEA tasks, sMAPE is modified to avoid small values due to Kelvin temperature units (see Appendix D). For ETT and ENER tasks, results are based on a single forecasting season (i.e., one train / test split per forecasting horizon), while we average over multiple seasons for other tasks.

Sonnet exhibits the overall best performance across the explored MTS forecasting tasks. Comparing Sonnet's performance to the best-performing baseline model for each task and forecasting horizon yields an MAE reduction of $1.1\%$ on average. Ranking-wise, Sonnet is the best-performing model on $34$, and the second-best on $9$ out of $47$ forecasting tasks. Sonnet shows relatively inferior performance on the ETT tasks. However, the ETT data sets have only $6$ exogenous variables and cover a limited training time span ($2$ years). Consequently, they offer a limited spectrum to gain from exploring variable dependencies, as well as insufficient historical context, both of which are elements that Sonnet leverages from. Aside from the ETT tasks, Sonnet reduces MAE by $3.3\%$ on average. Therefore, Sonnet is more effective with more covariates and longer time spans for training. Compared to a specific forecasting model, Sonnet consistently improves accuracy, reducing MAE from $2.2\%$ for the best-performing baseline (DeformTime) to $31.1\%$ for the worst-performing one (DLinear). We note that the MAE performance gain over DeformTime is statistically significant based on a paired t-test across all the $47$ forecasting tasks ($p=5e-4$). This further highlights the model's forecasting capacity amongst a wide range of applications.

Focusing on the more challenging forecasting tasks (ILI and WEA), Sonnet outperforms baselines $25$ or $26$ (out of $32$ tasks in total) based on sMAPE or MAE, respectively. The MAE reduction is $3.5\%$ on average for ILI tasks and $2\%$ for WEA tasks. Within these tasks, we also observe consistent comparative performance patterns in the baseline models. Models that do not capture inter-variable dependencies (DLinear and PatchTST) generally offer lower predictability (Sonnet reduces their MAE by $28.8\%$ and $29.8\%$), and in most ILI tasks, they cannot surpass the performance of a persistence model. In contrast, baseline models that capture inter-variable dependencies tend to perform better. Among these, those that embed the covariates along the temporal dimension, namely Samformer, TimeXer, and iTransformer, exhibit inferior performance (Sonnet reduces their MAE by $23.9\%$, $22.1\%$, and $22.9\%$), whereas models that preserve temporal order, namely DeformTime, ModernTCN, and Crossformer, achieve stronger results (Sonnet reduces their MAE by $3.4\%$, $11.1\%$, and $8\%$). Hence, in tasks with more informative covariates, preserving temporal structure while modelling inter-variable dependencies is a desirable property. We provide ablation study results in Appendix E.3 that quantify the contribution of different modules to the overall forecasting accuracy of Sonnet. Additional performance evaluation results over the entire output sequence are provided in Appendix E.8.

### 4.3 Effectiveness of Different Attention Modules in MTS Forecasting

We evaluate the effectiveness of the proposed attention-driven module, MVCA, by integrating it into existing forecasting models (we refer to them as base models) that originally deploy naïve transformer attention. Experiments are conducted on the ILI tasks as these are the most representative of the MTS class we are exploring: they not only contain many exogenous predictors, but also frame hard practical epidemiological modelling problems when $H=21$ or $28$ days ahead. We use Samformer (Ilbert et al. 2024), iTransformer (Liu et al. 2024), and PatchTST (Nie et al. 2023) as base models. For each one of them, we evaluate $5$ attention configurations: removing the attention module altogether (¬ Attn), attention with Fourier transformer proposed by FNet (Lee-Thorp

| | | Sonnet | | VDAB | | FED | | FNet | | Naïve | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $H$ | MAE | $\epsilon$% | MAE | $\epsilon$% | MAE | $\epsilon$% | MAE | $\epsilon$% | MAE | $\epsilon$% |
| ILI-ENG | 7 | 1.479 | 21.8 | **1.433** | **20.3** | 5.991 | 77.7 | 1.591 | 27.7 | 1.717 | 25.3 |
| | 14 | **1.923** | **25.8** | 1.973 | 29.4 | 6.031 | 78.1 | 2.084 | 32.8 | 2.068 | 31.4 |
| | 21 | **2.510** | 36.5 | 2.774 | **35.2** | 6.045 | 78.2 | 2.765 | 39.3 | 2.781 | 37.2 |
| | 28 | **2.748** | **37.0** | 3.060 | 44.3 | 6.051 | 78.4 | 3.240 | 44.5 | 3.078 | 50.0 |
| ILI-US2 | 7 | **0.381** | **14.9** | 0.429 | 16.6 | 1.168 | 45.0 | 0.415 | 18.5 | 0.436 | 17.5 |
| | 14 | **0.449** | **18.4** | 0.500 | 19.1 | 1.179 | 45.5 | 0.558 | 23.2 | 0.532 | 20.4 |
| | 21 | **0.533** | 20.6 | 0.575 | **20.5** | 1.195 | 46.2 | 0.659 | 25.3 | 0.613 | 23.5 |
| | 28 | **0.579** | **21.1** | 0.643 | 24.3 | 1.193 | 46.1 | 0.705 | 29.1 | 0.651 | 26.1 |
| ILI-US9 | 7 | **0.267** | **13.0** | 0.277 | 13.4 | 0.856 | 41.0 | 0.286 | 14.4 | 0.301 | 13.9 |
| | 14 | **0.281** | **13.1** | 0.324 | 15.8 | 0.856 | 41.0 | 0.351 | 16.1 | 0.340 | 14.9 |
| | 21 | **0.318** | **14.1** | 0.319 | 14.5 | 0.857 | 41.0 | 0.379 | 17.8 | 0.404 | 19.2 |
| | 28 | **0.367** | **16.5** | 0.375 | 16.8 | 0.857 | 41.0 | 0.379 | 18.2 | 0.411 | 19.1 |
| WEA-CT | 4 | **1.624** | **9.6** | 1.714 | 10.1 | 4.359 | 24.6 | 1.629 | 9.6 | 1.726 | 10.2 |
| | 12 | **3.543** | **20.4** | 3.563 | 20.5 | 4.359 | 24.7 | 3.588 | 20.6 | 3.558 | 20.5 |
| | 28 | **3.728** | **21.3** | 3.760 | 21.5 | 4.359 | 24.6 | 3.916 | 22.3 | 3.758 | 21.5 |
| | 120 | 3.737 | 21.4 | **3.734** | **21.3** | 4.356 | 24.6 | 3.815 | 21.7 | 3.763 | 21.5 |
| WEA-HK | 4 | **0.639** | **4.1** | 0.645 | 4.1 | 2.801 | 17.7 | 0.639 | 4.1 | 0.676 | 4.3 |
| | 12 | **1.235** | **7.7** | 1.279 | 8.0 | 2.801 | 17.7 | 1.325 | 8.3 | 1.281 | 8.1 |
| | 28 | **1.413** | **8.8** | 1.463 | 9.1 | 2.808 | 17.8 | 1.494 | 9.3 | 1.475 | 9.2 |
| | 120 | **1.547** | **9.6** | 1.547 | 9.6 | 2.810 | 17.8 | 1.639 | 10.1 | 1.621 | 10.0 |
| WEA-LD | 4 | **1.723** | **15.2** | 1.814 | 15.8 | 4.668 | 34.9 | 1.740 | 15.3 | 1.811 | 16.0 |
| | 12 | **2.959** | **23.9** | 2.986 | 24.1 | 4.661 | 34.9 | 3.008 | 24.1 | 2.973 | 24.0 |
| | 28 | **3.216** | **25.5** | 3.271 | 25.8 | 4.659 | 34.9 | 3.378 | 26.5 | 3.268 | 25.8 |
| | 120 | **3.246** | **25.8** | 3.319 | 26.2 | 4.668 | 34.9 | 3.667 | 28.5 | 3.407 | 26.7 |
| WEA-NY | 4 | 1.272 | 11.9 | 1.331 | 12.4 | 5.386 | 40.9 | **1.268** | **11.8** | 1.366 | 12.9 |
| | 12 | 2.448 | **21.2** | 2.474 | 21.6 | 5.384 | 40.9 | 2.510 | 21.8 | **2.438** | 21.5 |
| | 28 | **2.674** | **23.1** | 2.708 | 23.4 | 5.387 | 40.9 | 2.797 | 23.8 | 2.724 | 23.1 |
| | 120 | **2.713** | **23.2** | 2.788 | 23.5 | 5.389 | 40.9 | 3.063 | 25.6 | 2.911 | 24.6 |
| WEA-SG | 4 | 0.344 | 1.3 | 0.346 | 1.3 | 0.700 | 2.5 | **0.341** | **1.2** | 0.346 | 1.3 |
| | 12 | **0.416** | **1.5** | 0.421 | 1.5 | 0.698 | 2.5 | 0.424 | 1.5 | 0.425 | 1.5 |
| | 28 | **0.465** | **1.7** | 0.473 | 1.7 | 0.698 | 2.5 | 0.495 | 1.8 | 0.471 | 1.7 |
| | 120 | **0.483** | **1.8** | 0.496 | 1.8 | 0.697 | 2.5 | 0.512 | 1.9 | 0.485 | 1.8 |

Table 3: Performance of Sonnet across the ILI and WEA tasks with different attention modules. Best results are **bolded**, second best underlined. MAE values are rounded to 3 decimals, sMAPE to 1 for spacing.

et al. 2022), the Frequency Enhanced Decomposed (FED) attention proposed in FEDformer (Zhou et al. 2022a), the variable deformable attention (VDAB) proposed by DeformTime (Shu and Lampos 2025), and ours (MVCA). Further details are provided in Appendices C and D.

Table 2 enumerates MAEs averaged across the $4$ test seasons for all forecasting horizons (sMAPEs in Appendix E, Table S9). Notably, for the base models, removing the attention module does not lead to significant performance degradation, with some cases even showing improved results, e.g. improved accuracy is reported for some horizons for the ILI-ENG task. This indicates that the attention modules in the base models do not always capture useful information.

By replacing naïve attention with the proposed MVCA module, we obtain significant performance gains across all ILI tasks, reducing MAE by $10.7\%$ on average across all base models. The average MAE reduction for PatchTST is greater ($15.1\%$) compared to the other base models ($10.4\%$ for Samformer and $6.7\%$ for iTransformer), indicating that the application of MVCA yields enhanced performance gains for the model that does not capture inter-variable dependencies. When compared to other modified attention modules, MVCA shows competitive performance, reducing MAE by $2.8\%$ on average. Compared to the best-performing baseline (VDAB from DeformTime), it reduces MAE and sMAPE by $3.5\%$ and $3\%$, respectively. While the gains in the ENG region are less pronounced, MVCA achieves consistently the best results across the US regions, and VDAB is the second best; both methods capture inter-variable dependencies while others do not. Comparably, replacing naïve attention with the Fourier attention modules from FEDformer and FNet results in worse performance.

### 4.4 Sonnet with Different Attention Variants

To understand how different attention mechanisms affect the forecasting accuracy of Sonnet, we replace the MVCA module with the aforementioned modified attention mechanisms (see section 4.3). Experiments are conducted on both the ILI and WEA forecasting tasks across all locations, forecasting horizons, and test periods. Hyperparameters are re-tuned for each attention variant using the same validation procedure. Appendix C provides further details.

Results are enumerated in Table 3. Overall, using MVCA yields the best accuracy compared to other attention modules, outperforming them in 26 out of the 32 tasks (MAE). On average, it reduces the MAE and sMAPE scores by $2.9\%$ and $2\%$, respectively. The most competitive baseline module is VDAB (DeformTime). MVCA outperforms it by $3.6\%$ and $3\%$ on average w.r.t. MAE and sMAPE, respectively. The worst-performing attention module is FED (FEDformer). MVCA reduces its MAE by $51.5\%$. FED focuses on capturing seasonality information using frequency transformation without considering the dependencies between variables. This shows that modelling seasonality alone is insufficient for MTS forecasting, particularly when key predictive information comes from exogenous variables. The performance obtained by using either naïve attention or FNet is similar. MVCA reduces their MAE by $7\%$ and $6.6\%$ on average, respectively. This is expected as FNet does not introduce learnable parameters to the naïve attention.

## 5 Conclusion

In this paper, we present Sonnet, a novel model for multivariable time series forecasting tasks. Sonnet operates in the spectral dimension using learnable wavelet transforms and captures variable dependencies with frequency-based coherence. It then predicts future temporal dynamics with the Koopman operator. Experiments are conducted on 12 data sets from different application domains, with 9 of them containing multiple test seasons for a more comprehensive analysis. Sonnet yields the best performance in 34 out of 47 tasks, reducing MAE by $2.2\%$ and $1.1\%$ on average compared to the most performant baseline overall (DeformTime) or per task (can vary), respectively. Additionally, our experiments highlight that replacing vanilla attention, present in various forecasting models, with the proposed MVCA module improves prediction accuracy by a large margin.

## Acknowledgements

The authors would like to thank the RCGP for providing ILI rates for England. V. Lampos would like to acknowledge all levels of support from the EPSRC grant EP/X031276/1.

## References

Ansari, A. F.; Stella, L.; Turkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S. S.; Arango, S. P.; Kapoor, S.; Zschiegner, J.; Maddix, D. C.; Mahoney, M. W.; Torkkola, K.; Wilson, A. G.; Bohlke-Schneider, M.; and Wang, Y. 2024. Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research*.

Antunes, A.; Bonfim, D.; Monteiro, N.; and Rodrigues, P. M. 2018. Forecasting banking crises with dynamic panel probit models. *International Journal of Forecasting*, 34(2): 249–275.

Arneodo, A.; Grasseau, G.; and Holschneider, M. 1988. Wavelet Transform of Multifractals. *Phys. Rev. Lett.*, 61: 2281–2284.

Avila, A. M.; and Mezić, I. 2020. Data-driven analysis and forecasting of highway traffic dynamics. *Nature Communications*, 11(1): 2090.

Bochner, S. 1953. Fourier Transforms of Time Series. *Proceedings of the National Academy of Sciences*, 39(4): 302–307.

da Silva, R. G.; Ribeiro, M. H. D. M.; Mariani, V. C.; and Coelho, L. D. S. 2020. Forecasting Brazilian and American COVID-19 cases based on artificial intelligence coupled with climatic exogenous variables. *Chaos, Solitons & Fractals*, 139: 110027.

Dalton, A.; and Bekker, B. 2022. Exogenous atmospheric variables as wind speed predictors in machine learning. *Applied Energy*, 319: 119257.

Das, A.; Kong, W.; Sen, R.; and Zhou, Y. 2024. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*.

Daubechies, I. 1990. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5): 961–1005.

De Moortel, I.; Munday, S. A.; and Hood, A. W. 2004. Wavelet Analysis: the effect of varying basic wavelet parameters. *Solar Physics*, 222(2): 203–228.

Dudgeon, D.; and Mersereau, R. 1984. *Multidimensional Digital Signal Processing*. Prentice-Hall. ISBN 9780136049593.

Dugas, A. F.; Jalalpour, M.; Gel, Y.; Levin, S.; Torcaso, F.; Igusa, T.; and Rothman, R. E. 2013. Influenza forecasting with Google Flu Trends. *PLoS One*, 8(2): e56176.

Farge, M.; et al. 1992. Wavelet transforms and their applications to turbulence. *Annual review of fluid mechanics*, 24(1): 395–458.

Fildes, R.; and Kourentzes, N. 2011. Validation and forecasting accuracy in models of climate change. *International Journal of Forecasting*, 27(4): 968–995.

Hamill, T. M.; and Whitaker, J. S. 2007. Ensemble calibration of 500-hPa geopotential height and 850-hPa and 2-m temperatures using reforecasts. *Monthly Weather Review*, 135(9): 3273–3280.

Han, L.; Ye, H.; and Zhan, D. 2023. The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting. arXiv:2304.05206.

Herzen, J.; Lässig, F.; Piazzetta, S. G.; Neuer, T.; Tafti, L.; Raille, G.; Van Pottelbergh, T.; Pasieka, M.; Skrodzki, A.; Huguenin, N.; Dumonal, M.; Kościsz, J.; Bader, D.; Gusset, F.; Benheddi, M.; Williamson, C.; Kosinski, M.; Petrik, M.; and Grosch, G. 2022. Darts: User-Friendly Modern Machine Learning for Time Series. *Journal of Machine Learning Research*, 23(124): 1–6.

Hidalgo, B.; and Goodman, M. 2013. Multivariate or multivariable regression? *American journal of public health*, 103(1): 39–40.

Ilbert, R.; Odonnat, A.; Feofanov, V.; Virmaux, A.; Paolo, G.; Palpanas, T.; and Redko, I. 2024. SAMformer: Unlocking the Potential of Transformers in Time Series Forecasting with Sharpness-Aware Minimization and Channel-Wise Attention. In *International Conference on Machine Learning*, volume 235. PMLR.

Lange, H.; Brunton, S. L.; and Kutz, J. N. 2021. From Fourier to Koopman: Spectral Methods for Long-term Time Series Prediction. *Journal of Machine Learning Research*, 22(41): 1–38.

Lee-Thorp, J.; Ainslie, J.; Eckstein, I.; and Ontanon, S. 2022. FNet: Mixing Tokens with Fourier Transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4296–4313.

Li, Y.; He, H.; Wu, J.; Katabi, D.; and Torralba, A. 2020. Learning Compositional Koopman Operators for Model-Based Control. In *International Conference on Learning Representations*.

Lin, S.; Lin, W.; HU, X.; Wu, W.; Mo, R.; and Zhong, H. 2024. CycleNet: Enhancing Time Series Forecasting through Modeling Periodic Patterns. In *The Annual Conference on Neural Information Processing Systems*.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations*.

Liu, Y.; Li, C.; Wang, J.; and Long, M. 2023. Koopa: Learning Non-stationary Time Series Dynamics with Koopman Predictors. In *Conference on Neural Information Processing Systems*.

Luo, D.; and Wang, X. 2024a. DeformableTST: Transformer for Time Series Forecasting without Over-reliance on Patching. In *The Annual Conference on Neural Information Processing Systems*.

Luo, D.; and Wang, X. 2024b. ModernTCN: A Modern Pure Convolution Structure for General Time Series Analysis. In *International Conference on Learning Representations*.

Lusch, B.; Kutz, J. N.; and Brunton, S. L. 2018. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1): 4950.

Mallat, S. 1989. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7): 674–693.

Mezić, I. 2005. Spectral Properties of Dynamical Systems, Model Reduction and Decompositions. *Nonlinear Dynamics*, 41(1): 309–325.

Mima, T.; and Hallett, M. 1999. Corticomuscular Coherence: A Review. *Journal of Clinical Neurophysiology*, 16(6).

Morris, M.; Hayes, P.; Cox, I. J.; and Lampos, V. 2023. Neural network models for influenza forecasting with associated uncertainty using Web search activity trends. *PLoS Computational Biology*, 19(8): e1011392.

Ngui, W. K.; Leong, M. S.; Hee, L. M.; and Abdelrhman, A. M. 2013. Wavelet analysis: mother wavelet selection methods. *Applied mechanics and materials*, 393: 953–958.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations*.

Niemira, M. P.; and Saaty, T. L. 2004. An Analytic Network Process model for financial-crisis forecasting. *International Journal of Forecasting*, 20(4): 573–587.

Piao, X.; Chen, Z.; Murayama, T.; Matsubara, Y.; and Sakurai, Y. 2024. Fredformer: Frequency Debiased Transformer for Time Series Forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2400–2410.

Priestley, M. B. 2018. The Spectral Analysis of Time Series. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 151(3): 573–574.

Rasp, S.; Dueben, P. D.; Scher, S.; Weyn, J. A.; Mouatadid, S.; and Thuerey, N. 2020. WeatherBench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11): e2020MS002203.

Rowley, C. W.; Mezić, I.; Bagheri, S.; Schlatter, P.; and Henningson, D. S. 2009. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641: 115–127.

Scherrer, S. C.; Appenzeller, C.; Eckert, P.; and Cattani, D. 2004. Analysis of the spread–skill relations using the ECMWF ensemble prediction system over Europe. *Weather and Forecasting*, 19(3): 552–565.

Shaman, J.; and Karspeck, A. 2012. Forecasting seasonal outbreaks of influenza. *PNAS*, 109(50): 20425–20430.

Shu, Y.; and Lampos, V. 2025. DEFORMTIME: Capturing Variable Dependencies with Deformable Attention for Time Series Forecasting. *Transactions on Machine Learning Research*.

Sorensen, H.; Jones, D.; Heideman, M.; and Burrus, C. 1987. Real-valued fast Fourier transform algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(6): 849–863.

Stein, R. B.; French, A. S.; and Holden, A. V. 1972. The Frequency Response, Coherence, and Information Capacity of Two Neuronal Models. *Biophysical Journal*, 12(3): 295–322.

Taylor, J. W.; and Buizza, R. 2003. Using weather ensemble predictions in electricity demand forecasting. *International Journal of Forecasting*, 19(1): 57–70.

Varanini, M.; De Paolis, G.; Emdin, M.; Macerata, A.; Pola, S.; Cipriani, M.; and Marchesi, C. 1997. Spectral analysis of cardiovascular time series by the S-transform. In *Computers in Cardiology 1997*, 383–386.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*.

Verma, Y.; Heinonen, M.; and Garg, V. 2024. ClimODE: Climate and Weather Forecasting with Physics-informed Neural ODEs. In *The Twelfth International Conference on Learning Representations*.

Wang, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J. Y.; and Zhou, J. 2024a. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *International Conference on Learning Representations*.

Wang, Y.; Wu, H.; Dong, J.; Qin, G.; Zhang, H.; Liu, Y.; Qiu, Y.; Wang, J.; and Long, M. 2024b. TimeXer: Empowering Transformers for Time Series Forecasting with Exogenous Variables. In *The Annual Conference on Neural Information Processing Systems*.

White, L.; and Boashash, B. 1990. Cross spectral analysis of non-stationary processes. *IEEE Transactions on Information Theory*, 36(4): 830–835.

Witt, S. F.; and Witt, C. A. 1995. Forecasting tourism demand: A review of empirical research. *International Journal of Forecasting*, 11(3): 447–475.

Young, P. C. 2018. Data-based mechanistic modelling and forecasting globally averaged surface temperature. *International Journal of Forecasting*, 34(2): 314–335.

Yu, H.; Guo, P.; and Sano, A. 2024. AdaWaveNet: Adaptive Wavelet Network for Time Series Analysis. *Transactions on Machine Learning Research*.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are Transformers Effective for Time Series Forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11121–11128.

Zhang, J.; Tsui, F.-C.; Wagner, M. M.; and Hogan, W. R. 2003. Detection of outbreaks from time series data using wavelet transform. *AMIA Annu Symp Proc*, 2003: 748–752.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *International Conference on Learning Representations*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022a. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, 27268–27286.

Zhou, T.; Ma, Z.; xue wang; Wen, Q.; Sun, L.; Yao, T.; Yin, W.; and Jin, R. 2022b. FiLM: Frequency improved Legendre Memory Model for Long-term Time Series Forecasting. In *Advances in Neural Information Processing Systems*.

Zhou, X.; Ye, J.; Zhao, S.; Jin, M.; Yang, C.; Wen, Y.; and Yuan, X. 2024. EffiCANet: Efficient Time Series Forecasting with Convolutional Attention. arXiv:2411.04669.