UFO: LEARNING UNIVERSAL FREQUENCY REPRESENTATION WITH NEURAL OPERATOR FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Frequency representations play a vital role in time series forecasting, with recent deep learning approaches increasingly modeling temporal dependencies in the Fourier basis. In parallel, operator learning frameworks such as the Fourier Neural Operator (FNO) demonstrate that complicated operators can be parameterized via linear transforms in frequency space coupled with local non-linearities. However, existing methods remain constrained by discretization errors and full-frequency mode representation, which inherently bottleneck their prediction capacity. To mitigate this limitation, we replace Fourier bases with adaptive basis functions and determine their rotation coefficients. Time translation is then converted into these coefficients acting on the basis functions. Mathematically, we prove that both the basis functions and the rotation coefficients can be universally approximated by neural networks. Grounded in this theoretical framework, we propose the Universal Frequency Operator (UFO) for long-term time-series forecasting (LSTF). Instead of relying on fixed Fourier or wavelet bases, UFO learns adaptive bases and the corresponding rotation coefficients, capturing temporal dynamics by combining them through complex multiplication. Extensive experiments on 8 real-world datasets demonstrate that UFO consistently outperforms state-of-the-art baselines. To demonstrate that UFO captures core temporal dynamics in the basis functions, we ablate the history input's high-frequency mode and observe that UFO consistently outperforms other methods across all high-frequency cutoff levels. Code is available at this repository: https://anonymous.4open.science/status/UFO-3D30.

1 INTRODUCTION

Time series prediction is a classical field with applications spanning climate forecasting to the stock market (Box et al., 2015). In long-term time-series forecasting (LSTF), deep learning demonstrates strong capability in leveraging diverse neural network backbones (Zeng et al., 2023; Liu et al., 2024; Wang et al., 2024b). Frequency transformations, such as Fourier and wavelet transforms, represent time series using bases and coefficients, enabling effective cascade analysis and prediction (Nussbaumer, 1981). Recently, a proliferation of research combines this traditional method with deep learning to improve time series prediction (Wu et al., 2022; Xu et al., 2024; Yi et al., 2024).

In parallel, Fourier Neural Operator (FNO) combines a linear operator learned in the Fourier domain with a neural network for capturing nonlinear dynamics, enabling mesh-free prediction (Li et al., 2020; Lu et al., 2021). Unlike conventional neural networks tied to fixed input and output dimensions, neural operators approximate the underlying function-to-function mapping using parameters shared across discretizations. Under appropriate assumptions, their discrete implementations converge to a well-defined operator as the discretization is refined (Berner et al., 2026).

The successes of frequency-based methods are rooted in the fact that a time series can be exactly represented in the frequency domain through a Fourier basis and its corresponding coefficients. Through this lens, these models effectively represent time series via the Fourier basis, and seek the optimal coefficients for prediction. However, the frequency-based models still face several challenges. For frequency transformations, although the Fourier transform ensures any time series can be represented losslessly in the Fourier basis, the generally used discret Fourier transform (DFT)

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

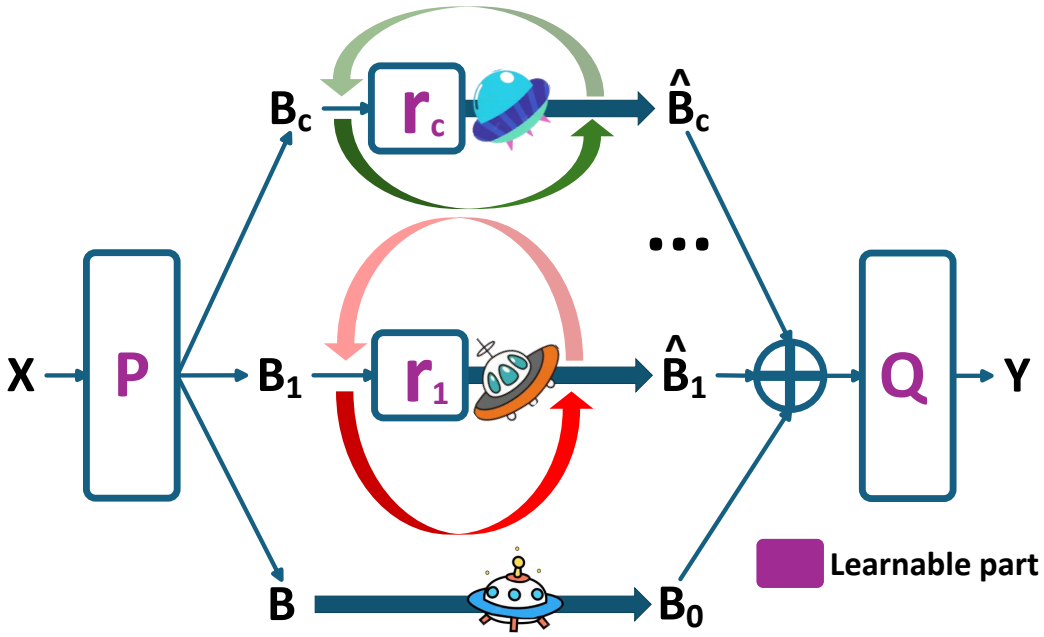


Figure 1: Illustration of a UFO. The lifting function P maps the input history $\mathbf{X}$ into a latent space. In this space, the basis functions $\mathbf{B}$ are split into c channels. For the i-th channel, the rotation function r_i takes $\mathbf{B}_i$ to determine a rotation coefficient. Channel-wise complex multiplication of $\mathbf{B}_i$ with its rotation coefficient yields time-shifted basis functions $\widehat{\mathbf{B}}_i$. This evolved term is added to the stationary term $\mathbf{B}$ and then projected by Q back to the observation space for predicting $\mathbf{Y}$.

naturally introduces discretization error. This representation error is further increased by the general truncation operation, since high-frequency modes often contain noise. For computational complexity, even the fast Fourier transform (FFT) reduces it to $\mathcal{O}(T \log T)$, it still burdens the training loop of deep learning.

Time series decomposition in the frequency domain suggests that deep learning seeks adaptive basis functions and learns their corresponding coefficients. With these basis functions and coefficients, the time increment can be expressed as a rotation of the basis functions driven by the coefficients. Building on this insight, we further prove that both the basis functions and their rotation coefficients can be universally approximated by neural networks. Grounded in this theoretical framework, we propose the Universal Frequency Operator (UFO), which learns the universal frequency representation in adaptive function space for time series forecasting. Extensive experiments in 8 real-world datasets suggest that UFO consistently outperforms state-of-the-art (SOTA) baselines. To demonstrate UFO's ability to capture core temporal dynamics in the basis function, we ablate the history input's high-frequency mode and observe that UFO consistently outperforms other methods across all high-frequency cutoff levels.

Our contributions are summarized as follows:

- The Fourier transform decomposes time series into basis functions and coefficients, converting time translation into a rotation of these basis functions driven by their coefficients. We prove that both the basis functions and the rotation coefficients can be universally approximated by neural networks.

- We introduce the Universal Frequency Operator (UFO) for LSTF. UFO employs neural networks to directly learn basis functions and their rotation coefficients, which are then combined through complex multiplication to model temporal dynamics, thereby circumventing the need for Fourier transforms.

- Experiments on 8 real-world datasets show UFO consistently outperforms SOTA baselines. Ablation of the history input's high-frequency modes further demonstrates that UFO captures core temporal dynamics.

2 PRELIMINARIES AND RELATED WORK

2.1 PROBLEM DEFINITION

Given a d-dimensional multivariate time series $\{x_t\}_{t\in\mathbb{Z}} \subset \mathbb{R}^d$, define the history input $\mathbf{X} \in \mathbb{R}^{d\times n}$ and prediction target $\mathbf{Y} \in \mathbb{R}^{d\times m}$ as

$$\mathbf{X} = [x_1, \cdots, x_n], \quad \mathbf{Y} = [x_{n+1}, \cdots, x_{n+m}]. \tag{1}$$

In frequency-based models, the learning paradigm can be described by the Fourier transform $\mathcal{F}$, its inverse $\mathcal{F}^{-1}$, and a deep learning model R_θ operating in the frequency domain like

$$\mathbf{Y} = \mathcal{F}^{-1} \circ R_\theta \circ \mathcal{F}(\mathbf{X}). \tag{2}$$

Frequency-based models primarily differ in their use of R_θ; accordingly, we categorize related work by how R_θ is employed.

2.2 DEEP FREQUENCY MODEL

Deep frequency models exploit spectral representations to capture temporal dependencies. To enhance representation capacity, Autoformer (Wu et al., 2021) introduces an FFT-based autocorrelation mechanism that aggregates temporal patterns. TimesNet (Wu et al., 2022), DEPTS (Fan et al., 2022), PDF (Dai et al., 2024), and SparseTSF (Lin et al., 2024) exploit inherent periodicity through periodic-pattern extraction or period-based decomposition, facilitating the modeling of recurring temporal structures. To capture multivariate temporal dependencies, FEDformer (Zhou et al., 2022b) employs frequency-enhanced attention to model spectral interactions, while FreTS (Yi et al., 2023) uses frequency-domain multi-layer perceptrons (MLPs) to learn channel and temporal dependencies. To suppress noise, FiLM (Zhou et al., 2022a) combines historical information preservation with Fourier-domain filtering, while FITS (Xu et al., 2024) applies a low-pass filter followed by a complex-valued linear mapping in the frequency domain. To handle Fourier bases and coefficients, FilterNet (Yi et al., 2024) combines signal processing with MLPs, DERITS (Fan et al., 2024) uses an order-adaptive Fourier convolution network to exploit the full frequency spectrum, and FR-Net (Zhang et al., 2024b) rotates frequency-domain bases to model temporal increments. Although these methods aim to capture informative global and periodic structures for accurate forecasting, most are limited to fixed discretizations. This motivates the use of neural operators, which learn mappings between function spaces and generalize across resolutions.

2.3 NEURAL OPERATOR

Neural operators extend neural networks from finite-dimensional vector mappings to mappings between function spaces. Let $\mathcal{A}$ and $\mathcal{U}$ denote suitable input and output function spaces. A neural operator learns a nonlinear mapping $\mathcal{G}_\theta : \mathcal{A} \to \mathcal{U}$, which can generally be expressed as

$$\mathcal{G}_\theta = Q \circ \mathcal{V}_L \circ \cdots \circ \mathcal{V}_1 \circ P. \tag{3}$$

Here, the lifting operator P maps the input function pointwise to a higher-dimensional latent representation. Subsequently, a sequence of hidden operator layers $\{\mathcal{V}_l\}_{l=1}^{L}$ updates this representation by combining local linear transformations with global integral kernels and nonlinear activations, thereby modeling long-range dependencies. Finally, the projection operator Q maps the final latent representation pointwise to the output space.

Several neural operator architectures have been proposed based on distinct parameterizations of their hidden operators to learn discretization-independent, function-to-function mappings. Foundational approaches like DeepONet (Lu et al., 2021) utilize separate branch and trunk networks, whereas the Fourier Neural Operator (FNO) (Li et al., 2020), Graph Neural Operator (GNO) (Kovachki et al., 2023), and Wavelet Neural Operator (WNO) (Tripura & Chakraborty, 2022) parameterize integral kernels via the Fourier domain, graph-based message passing, and wavelet transforms, respectively.

Recent advancements primarily enrich the latent transformation through efficient spectral mixing in AFNO (Guibas et al., 2021), multi-resolution hierarchies in U-NO (Rahman et al., 2022), and attention mechanisms in HAMLET (Bryutkin et al., 2024). Conversely, NOMAD (Seidman et al., 2022) target the decoding stage with a nonlinear manifold decoder. Further extensions explicitly incorporate structured temporal evolution via neural Taylor expansions (Wu et al., 2026) or linearized Koopman dynamics (Xiong et al., 2024), and explore alternative transform domains such as the Laplace domain (Cao et al., 2024). Grounded by theoretical frameworks like frame theory (Bartolucci et al., 2023), which analyze the mathematical validity of discretization consistency across scales.

2.4 KOOPMAN OPERATOR

By sharing the learning linear operator in a latent space with the neural operator, modern Koopman theory shows that nonlinear dynamics can be described by a linear operator acting in an infinite-dimensional space (Brunton et al., 2021). Recent work uses autoencoders to learn basis function spaces, mapping inputs into nonlinear latent representations governed by linear evolution operators (Takeishi et al., 2017; Lusch et al., 2018; Azencot et al., 2020). The commute between a nonlinear operator in the observation space and a linear operator in the latent space is used in time series prediction (Liu et al., 2023; Zheng et al., 2025). Building on Takens' theorem (Takens, 2006), DeepEDM (Majeedi et al., 2025) uses empirical dynamic modeling to approximate the underlying dynamics. SKOLR (Zhang et al., 2025) integrates Koopman theory into recurrent neural networks (RNNs) by stacking parallel RNNs for prediction.

3 ADAPTIVE FREQUENCY BASIS

Equation 2 can typically be viewed as an encoder-decoder architecture (Berner et al., 2026), where the Fourier transform $\mathcal{F}$ and inverse Fourier transform $\mathcal{F}^{-1}$ represent the encoder and decoder, respectively. $\mathcal{F}$ and $\mathcal{F}^{-1}$ determine a fixed Fourier basis. Although any time series can be represented perfectly in the continuous Fourier basis, discretization and truncation errors inevitably exist in practice. To alleviate this discretization error, Neural Operators (NO) employ a lifting function P that maps data from the observation space to a high-dimensional space, combined with the projection function Q to map back to the original space. These lifting and projection functions can be viewed as a universal approximation to $\mathcal{F}$ and $\mathcal{F}^{-1}$ (Cybenko, 1989; Hornik, 1991).

Theorem 1 (Universal approximation of the Fourier transform). *Let V be a d-dimensional real subspace of $L^2(\mathbb{R};\mathbb{R})$, and let $T : \mathbb{R}^d \to V$ be a fixed linear isomorphism associating each observation vector $\mathbf{X}$ with a signal $f_{\mathbf{X}} = T\mathbf{X}$. Let $\mathscr{F}$ denote the continuous Fourier transform in L^2, and set $\widehat{V} = \mathscr{F}(V)$. Fix an injective real-linear coordinate map $C : \widehat{V} \to \mathbb{R}^D$ with $D \geq 2d$, and define the Fourier encoding map by*

$$\mathcal{F} = C \circ \mathscr{F} \circ T : \mathbb{R}^d \to \mathbb{R}^D, \tag{4}$$

with inverse $\mathcal{F}^{-1}$ defined on $\mathcal{F}(\mathbb{R}^d)$. Let $\mathcal{K} \subset \mathbb{R}^d$ be nonempty and compact. Assuming the class of multilayer perceptrons satisfies the uniform universal approximation property on compact subsets of finite-dimensional Euclidean spaces, then for every $\epsilon > 0$, there exist a lifting network $P : \mathbb{R}^d \to \mathbb{R}^D$ and a projection network $Q : \mathbb{R}^D \to \mathbb{R}^d$ such that

$$\sup_{\mathbf{X}\in\mathcal{K}} \|P(\mathbf{X}) - \mathcal{F}(\mathbf{X})\|_{\mathrm{F}} < \epsilon \tag{5}$$

and

$$\sup_{\mathbf{Z}\in\mathcal{F}(\mathcal{K})} \|Q(\mathbf{Z}) - \mathcal{F}^{-1}(\mathbf{Z})\|_{\mathrm{F}} < \epsilon, \tag{6}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm with vectors formatted as column matrices.

The proof is detailed in Appendix B.1.

Remark. Theorem 1 applies to globally acting networks operating on a fixed, finite-dimensional signal representation. Under this coordinate representation, the operator composition in Equation 3 is well-defined provided that $\mathcal{G}_\theta(\mathcal{F}(\mathcal{K})) \subseteq \mathcal{F}(\mathbb{R}^d)$. Furthermore, establishing an end-to-end approximation bound requires two key conditions: (i) the spectral operator $\mathcal{G}_\theta$ must be well-controlled

over a neighborhood containing both the true spectral domain $\mathcal{F}(\mathcal{K})$ and the perturbed latent domain $P(\mathcal{K})$, and (ii) Q must approximate a continuous extension of the inverse Fourier transform $\mathcal{F}^{-1}$ over the resulting latent representations. Consequently, the choice of approximation domains and the structural design of the pair (P, Q) should be tailored to both the properties of $\mathcal{G}_\theta$ and the specific dataset.

Building upon Theorem 1, the lifting function P maps an observation time series $\mathbf{X} \in \mathbb{R}^{d\times n}$ into the latent basis functions $\mathbf{B} \in \mathbb{R}^{k\times c}$:

$$P(\mathbf{X}) = \mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_c], \tag{7}$$

where $\mathbf{B}$ comprises c channel vectors $\{\mathbf{b}_i\}_{i=1}^{c} \subset \mathbb{R}^k$, with k denoting the number of frequency modes. Unlike the classical Fourier transform, which predetermines the frequency modes k and channels c, our lifting function P learns a set of adaptive basis functions tailored to a specific dataset. Leveraging this dimensional flexibility, the parameterized pair (P, Q) constructs a data-adaptive latent space optimized for capturing complex temporal dynamics beyond rigid trigonometric components.

4 Learning temporal dynamics in adaptive basis

In classical Fourier analysis, the temporal translation Δt applied to a continuous signal $x(t)$ is equivalent to a phase shift in its frequency spectrum $X(\omega) = \mathcal{F}\{x(t)\}$. Specifically, the Fourier transform converts time increments into complex rotations

$$\mathcal{F}\{x(t + \Delta t)\} = X(\omega)e^{i\omega\Delta t}. \tag{8}$$

Since the phase factor $|e^{i\omega\Delta t}| = 1$, spectral dynamics are confined to the unit circle, meaning time propagation is a pure rotation in the complex plane.

Analogous to the Fourier frequency-domain representation of real-valued signals as conjugate-symmetric spectra, we extend the complex rotation to the basis functions $\mathbf{B}$ spanning the latent space generated by $P(\mathbf{X})$. Rather than relying on a predetermined grid of frequencies, we introduce a learnable frequency matrix $\mathbf{\Omega} \in \mathbb{R}^{\frac{k}{2}\times c}$ representing the rotational coefficients. To ensure dimensional matching and enforce conjugate pairing across the k latent dimensions, we construct the full rotation matrix by block-concatenating $\mathbf{\Omega}$ with its negative counterpart, forming $\widetilde{\mathbf{\Omega}} = [\mathbf{\Omega}^\top, -\mathbf{\Omega}^\top]^\top \in \mathbb{R}^{k\times c}$.

We define the temporal evolution of the adaptive basis as a rotation

$$\mathbf{B}(\Delta t) = \mathbf{B}(0) \odot \exp_\odot\left(i\widetilde{\mathbf{\Omega}}\Delta t\right), \tag{9}$$

where $\mathbf{B}(0) = P(\mathbf{X})$, $i = \sqrt{-1}$, $\odot$ denotes the Hadamard product, and $\exp_\odot$ denotes the element-wise complex exponential. This formulation naturally pairs each learned frequency with its conjugate, governing the pairwise complex evolution of $\mathbf{B}$.

To parameterize this continuous temporal evolution computationally, we perform a Taylor expansion on the adaptive basis $\mathbf{B}(\Delta t)$ around $\Delta t = 0$:

$$\mathbf{B}(\Delta t) = \sum_{j=0}^{\infty} \frac{(\Delta t)^j}{j!} \frac{\partial^j \mathbf{B}}{\partial t^j}(0). \tag{10}$$

Given the rotational model in Equation equation 9, the j-th order temporal derivative of the latent state evaluates to

$$\frac{\partial^j \mathbf{B}}{\partial t^j}(0) = \mathbf{B}(0) \odot \left(i\widetilde{\mathbf{\Omega}}\right)^{\odot j}. \tag{11}$$

Substituting this derivative back into Equation equation 10 yields the infinite-order Taylor propagation operator acting on the adaptive basis:

$$\mathbf{B}(\Delta t) = \mathbf{B}(0) \odot \sum_{j=0}^{\infty} \frac{(\Delta t)^j}{j!} \left(i\widetilde{\mathbf{\Omega}}\right)^{\odot j}. \tag{12}$$

The following theorem demonstrates that this exact continuous Taylor expansion in the adaptive basis can be universally approximated by neural networks, enabling the model to learn complex temporal dynamics without relying on fixed discretization grids (Chen & Chen, 1995; Chen et al., 2018).

Theorem 2 (Universal approximation of latent dynamics evolution). *Let $c \geq 1$ and $k \geq 2$ be integers, where k is even. Let $\Omega, \Delta t > 0$ be finite real constants. Let $\mathcal{K}_B \subset \mathbb{C}^{k\times c}$ and $\mathcal{K}_\Omega \subseteq [-\Omega, \Omega]^{\frac{k}{2}\times c}$ be nonempty compact sets. Define the domain*

$$\mathcal{D} = \mathcal{K}_B \times \mathcal{K}_\Omega \times [-\Delta t, \Delta t] \tag{13}$$

and the block-expanded frequency matrix $\widetilde{\mathbf{\Omega}} = [\mathbf{\Omega}^\top, -\mathbf{\Omega}^\top]^\top \in \mathbb{R}^{k\times c}$. We define the continuous latent space evolution function $T : \mathcal{D} \to \mathbb{C}^{k\times c}$ as

$$\begin{aligned} T(\mathbf{B}, \mathbf{\Omega}, \Delta t) &= \mathbf{B} \odot \sum_{j=0}^{\infty} \frac{(\Delta t)^j}{j!} (i\widetilde{\mathbf{\Omega}})^{\odot j} \\ &= \mathbf{B} \odot \exp_\odot(i\widetilde{\mathbf{\Omega}}\Delta t). \end{aligned} \tag{14}$$

Then the defining series converges uniformly on $\mathcal{D}$, guaranteeing that T is uniformly continuous on this domain.

Assuming a class of real-valued multilayer perceptrons possessing the universal approximation property on compact subsets of finite-dimensional Euclidean spaces, any such continuous evolution function in latent space can be universally approximated. Specifically, by identifying each complex matrix with its real and imaginary components, for every $\epsilon > 0$, there exists a neural map

$$N_\phi : \mathbb{C}^{k\times c} \times \mathbb{R}^{\frac{k}{2}\times c} \times \mathbb{R} \to \mathbb{C}^{k\times c}, \tag{15}$$

implemented by such a real-valued multilayer perceptron, such that

$$\sup_{(\mathbf{B},\mathbf{\Omega},\Delta t)\in\mathcal{D}} \|N_\phi(\mathbf{B}, \mathbf{\Omega}, \Delta t) - T(\mathbf{B}, \mathbf{\Omega}, \Delta t)\|_{\mathrm{F}} < \epsilon. \tag{16}$$

Here, the complex Frobenius norm is defined by $\|\mathbf{Z}\|_{\mathrm{F}}^2 = \sum_{a=1}^{k}\sum_{b=1}^{c} |Z_{a,b}|^2$.

The proof is detailed in Appendix B.2.

5 Universal Frequency Operator

Building on Theorems 1 and 2, which demonstrate that time translation can be transformed into learnable rotational coefficients acting on an adaptive latent space, we introduce the Universal Frequency Operator (UFO) for time series forecasting. As Figure 1 illustrates, UFO integrates a lifting function P, a projection function Q, and a set of channel-wise rotation functions $R = \{r_1, \ldots, r_c\}$, all of them implemented via a single linear layer.

The forecasting procedure consists of three main stages:

1. **Lifting:** The lifting function P maps the input time series $\mathbf{X} \in \mathbb{R}^{d\times n}$ (with data dimension d and history input length n) into a multi-channel latent space as
$$\mathbf{B} = P(\mathbf{X}) \in \mathbb{R}^{d\times k\times c}, \tag{17}$$
where $d \times k$ denotes the number of adaptive frequency modes and c denotes the latent channel dimension.
2. **Rotation:** Following Equation 7, we slice the latent state $\mathbf{B}$ along the channel dimension as $\mathbf{B} = [\mathbf{B}_1, \ldots, \mathbf{B}_c]$ with $\mathbf{B}_i \in \mathbb{R}^{d\times k}$. For the i-th channel, the rotation function r_i receives $\mathbf{B}_i$ as input to generate the rotational frequencies
$$\mathbf{\Omega}_i = r_i(\mathbf{B}_i) \in \mathbb{R}^{d\times \frac{k}{2}}, \quad \mathbf{\Omega} = [\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_c] \in \mathbb{R}^{d\times \frac{k}{2}\times c}. \tag{18}$$
The full rotation tensor $\widetilde{\mathbf{\Omega}} \in \mathbb{R}^{d\times k\times c}$ is formed by concatenating $\mathbf{\Omega}$ with its negative counterpart along the mode dimension, $\widetilde{\mathbf{\Omega}} = [\mathbf{\Omega}^\top, -\mathbf{\Omega}^\top]^\top$. Applying Equation 9, temporal evolution is realized via complex state rotation
$$\widehat{\mathbf{B}} = \mathbf{B} \odot \exp_\odot\left(i\widetilde{\mathbf{\Omega}}\Delta t\right). \tag{19}$$

We constrain the phase dynamics to the complex unit circle $\mathbb{S}^1 = \{z \in \mathbb{C} : |z| = 1\}$, which guarantees energy conservation by modeling the state evolution as a system of linear Hamiltonian harmonic oscillators (Zhang et al., 2024a).

3. **Projection:** Treating $\mathbf{B}$ as the stationary state at $\Delta t = 0$, we combine it with the evolved state $\widehat{\mathbf{B}}$ via a ResNet-style residual connection (He et al., 2016) and sum across channels to obtain $\widetilde{\mathbf{B}} \in \mathbb{R}^{d \times k}$. The projection function Q maps $\widetilde{\mathbf{B}}$ back to the original observation space to yield the forecast

$$\widehat{\mathbf{Y}} = Q(\widetilde{\mathbf{B}}) \in \mathbb{R}^{d \times m}, \tag{20}$$

where m denotes the prediction horizon.

The entire framework is trained end-to-end by minimizing the Mean Squared Error (MSE) between the forecasted output $\widehat{\mathbf{Y}}$ and the ground-truth future series $\mathbf{Y} \in \mathbb{R}^{d \times m}$.

Computational complexity analysis. Deep frequency models (e.g., FreTS (Yi et al., 2023), FilterNet (Yi et al., 2024)) rely on FFT to map signals into the spectral domain. For an input sequence of length n and a prediction length m, the forward FFT and subsequent inverse FFT (iFFT) operations impose a computational complexity of $\mathcal{O}(n \log n + m \log m)$. In contrast, UFO bypasses the standard orthogonal basis transformation. By utilizing a learnable lifting function to project the temporal sequence directly into a latent space, UFO reduces the computational overhead to linear time $\mathcal{O}(n + m)$. By removing the logarithmic factor, this asymptotic improvement makes UFO inherently more scalable and computationally efficient for processing long sequences.

6 Experiments

6.1 Dataset.

To evaluate the effectiveness of UFO, we conduct experiments on widely used multivariate forecasting benchmarks, including `ETT` (`ETTh1`, `ETTh2`, `ETTm1`, `ETTm2`), `Weather`, `Solar`, `Electricity`, and `Traffic`. More details on these datasets can be found in Informer (Zhou et al., 2020) and Autoformer (Wu et al., 2021). These datasets are standard in long-term forecasting evaluation and cover diverse fields, including periodic industrial signals, meteorological observations, solar power production, electricity consumption, and transportation measurements.

6.2 Baselines.

We compare UFO's prediction performance against three categories of state-of-the-art (SOTA) baselines:

(1) **Deep frequency model**: TimesNet (Wu et al., 2022), FreTS (Yi et al., 2023) and FilterNet (Yi et al., 2024);

(2) **Operator learning model**: Koopa (Liu et al., 2023), DeepEDM (Majeedi et al., 2025), SKOLR (Zhang et al., 2025);

(3) **Other backbones**: DLinear Zeng et al. (2023), iTransformer (Liu et al., 2024), TimeMixer (Wang et al., 2024a) and TimeXer (Wang et al., 2024b).

The introduction of each baseline is detailed in Appendix A.

6.3 Implementation Details.

UFO was trained for 50 epochs using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 1×10^{-3}. Baselines were trained using their official settings. We selected two common metrics in time series forecasting: Mean Absolute Error (MAE) and Mean Squared Error (MSE). Experiments were conducted on 1 NVIDIA RTX 4070 Ti 12 GB GPU.

Table 1: Long-term forecasting results. The input length is 96, and results are averaged over prediction lengths of [96, 192, 336, 720]. Lower MSE and MAE values signify better predictions. Red indicates the best result, Blue indicates the second best.

Models		UFO		Koopa		DeepEDM		SKOLR		TimesNet		FreTS		FilterNet		DLinear		iTransfomer		TimeMixer		TimeXer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.370	0.389	0.392	0.409	0.383	0.402	0.411	0.426	0.436	0.445	0.400	0.425	0.377	0.397	0.412	0.442	0.378	0.397	0.383	0.399	0.390	0.400
	192	0.426	0.422	0.441	0.436	0.442	0.437	0.458	0.455	0.473	0.484	0.451	0.443	0.477	0.446	0.484	0.466	0.435	0.433	0.441	0.430	0.445	0.432
	336	0.469	0.443	0.491	0.460	0.500	0.460	0.503	0.480	0.525	0.488	0.509	0.481	0.526	0.476	0.518	0.487	0.476	0.451	0.495	0.458	0.482	0.452
	720	0.462	0.469	0.505	0.481	0.480	0.466	0.483	0.478	0.528	0.502	0.559	0.536	0.609	0.532	0.540	0.527	0.511	0.490	0.506	0.490	0.501	0.491
	Avg	0.432	0.431	0.452	0.454	0.451	0.441	0.464	0.460	0.491	0.480	0.480	0.471	0.497	0.463	0.496	0.481	0.450	0.443	0.449	0.439	0.455	0.444
ETTh2	96	0.284	0.338	0.304	0.352	0.294	0.342	0.324	0.364	0.330	0.368	0.342	0.397	0.292	0.343	0.341	0.395	0.294	0.345	0.294	0.347	0.296	0.343
	192	0.360	0.388	0.390	0.408	0.374	0.395	0.408	0.414	0.395	0.409	0.440	0.449	0.374	0.394	0.482	0.323	0.379	0.399	0.391	0.404	0.379	0.401
	336	0.409	0.424	0.435	0.446	0.411	0.428	0.424	0.433	0.457	0.456	0.538	0.510	0.422	0.431	0.593	0.542	0.434	0.436	0.423	0.436	0.415	0.425
	720	0.440	0.448	0.461	0.465	0.432	0.454	0.437	0.450	0.451	0.459	0.796	0.644	0.430	0.446	0.840	0.661	0.471	0.469	0.448	0.454	0.441	0.450
	Avg	0.373	0.400	0.398	0.418	0.378	0.405	0.398	0.415	0.408	0.423	0.529	0.500	0.380	0.404	0.564	0.480	0.395	0.412	0.389	0.410	0.383	0.405
ETTm1	96	0.320	0.357	0.328	0.365	0.331	0.368	0.334	0.372	0.341	0.378	0.340	0.375	0.319	0.359	0.346	0.374	0.340	0.370	0.320	0.360	0.330	0.367
	192	0.364	0.385	0.373	0.389	0.365	0.386	0.379	0.398	0.382	0.400	0.383	0.399	0.365	0.384	0.382	0.391	0.399	0.401	0.367	0.388	0.380	0.394
	336	0.394	0.404	0.404	0.411	0.391	0.406	0.427	0.424	0.415	0.422	0.420	0.425	0.393	0.405	0.415	0.415	0.436	0.427	0.391	0.404	0.406	0.414
	720	0.453	0.441	0.469	0.448	0.460	0.448	0.533	0.477	0.482	0.457	0.497	0.474	0.458	0.444	0.473	0.451	0.506	0.465	0.470	0.452	0.458	0.444
	Avg	0.383	0.397	0.394	0.403	0.387	0.402	0.418	0.418	0.405	0.414	0.410	0.418	0.384	0.398	0.404	0.408	0.420	0.416	0.387	0.401	0.394	0.405
ETTm2	96	0.173	0.256	0.181	0.261	0.180	0.266	0.183	0.261	0.187	0.265	0.195	0.287	0.175	0.257	0.193	0.293	0.185	0.266	0.176	0.259	0.177	0.258
	192	0.237	0.297	0.243	0.303	0.254	0.315	0.252	0.307	0.251	0.308	0.277	0.350	0.240	0.300	0.293	0.361	0.253	0.311	0.238	0.298	0.247	0.305
	336	0.294	0.336	0.301	0.341	0.312	0.352	0.310	0.342	0.322	0.349	0.371	0.405	0.298	0.339	0.385	0.429	0.325	0.358	0.292	0.335	0.303	0.343
	720	0.393	0.394	0.409	0.406	0.409	0.400	0.425	0.407	0.417	0.404	0.561	0.519	0.397	0.394	0.556	0.523	0.418	0.409	0.392	0.395	0.410	0.403
	Avg	0.274	0.321	0.284	0.328	0.289	0.333	0.293	0.329	0.294	0.332	0.351	0.390	0.278	0.322	0.357	0.402	0.295	0.336	0.275	0.322	0.284	0.327
Weather	96	0.162	0.208	0.160	0.206	0.197	0.232	0.176	0.225	0.172	0.223	0.184	0.239	0.164	0.211	0.196	0.255	0.172	0.211	0.161	0.209	0.159	0.206
	192	0.207	0.252	0.210	0.251	0.238	0.266	0.222	0.260	0.229	0.269	0.222	0.273	0.211	0.253	0.237	0.296	0.227	0.259	0.208	0.253	0.207	0.250
	336	0.260	0.291	0.268	0.293	0.288	0.302	0.284	0.306	0.277	0.301	0.272	0.318	0.270	0.297	0.283	0.335	0.281	0.299	0.264	0.294	0.262	0.290
	720	0.342	0.343	0.352	0.347	0.361	0.349	0.363	0.356	0.358	0.353	0.354	0.354	0.387	0.354	0.349	0.355	0.385	0.357	0.349	0.344	0.346	0.348
	Avg	0.243	0.274	0.248	0.274	0.271	0.287	0.261	0.287	0.259	0.287	0.258	0.304	0.250	0.278	0.268	0.318	0.259	0.280	0.244	0.275	0.244	0.274
Solar	96	0.191	0.241	0.229	0.266	0.277	0.316	0.226	0.261	0.246	0.271	0.236	0.299	0.201	0.240	0.283	0.374	0.208	0.247	0.199	0.245	0.250	0.300
	192	0.228	0.264	0.265	0.290	0.305	0.331	0.305	0.324	0.288	0.295	0.266	0.321	0.232	0.264	0.312	0.394	0.238	0.265	0.230	0.268	0.287	0.321
	336	0.227	0.283	0.294	0.309	0.329	0.345	0.285	0.293	0.375	0.364	0.285	0.331	0.249	0.278	0.347	0.417	0.249	0.274	0.248	0.280	0.334	0.343
	720	0.231	0.294	0.293	0.306	0.319	0.333	0.358	0.348	0.318	0.317	0.285	0.325	0.252	0.281	0.352	0.409	0.253	0.278	0.249	0.282	0.346	0.346
	Avg	0.219	0.271	0.270	0.293	0.308	0.331	0.294	0.307	0.307	0.312	0.268	0.319	0.234	0.266	0.324	0.399	0.237	0.266	0.232	0.269	0.304	0.328
ECL	96	0.152	0.244	0.167	0.268	0.175	0.261	0.165	0.268	0.169	0.274	0.188	0.275	0.151	0.245	0.224	0.308	0.150	0.242	0.156	0.247	0.140	0.242
	192	0.163	0.254	0.182	0.281	0.179	0.265	0.176	0.278	0.187	0.291	0.189	0.278	0.163	0.256	0.222	0.310	0.166	0.257	0.170	0.260	0.157	0.256
	336	0.178	0.267	0.201	0.298	0.196	0.283	0.250	0.334	0.200	0.301	0.205	0.296	0.179	0.272	0.242	0.327	0.181	0.273	0.187	0.278	0.172	0.269
	720	0.218	0.305	0.241	0.329	0.250	0.321	0.302	0.377	0.273	0.355	0.244	0.332	0.219	0.310	0.270	0.352	0.214	0.302	0.228	0.312	0.211	0.307
	Avg	0.178	0.268	0.198	0.294	0.200	0.283	0.223	0.314	0.207	0.305	0.207	0.295	0.178	0.271	0.240	0.324	0.178	0.269	0.185	0.274	0.170	0.269
Traffic	96	0.411	0.265	0.444	0.320	0.485	0.319	0.586	0.315	0.589	0.315	0.565	0.368	0.429	0.292	0.696	0.429	0.397	0.271	0.463	0.293	0.447	0.270
	192	0.416	0.270	0.440	0.321	0.467	0.303	0.612	0.323	0.618	0.323	0.568	0.365	0.449	0.296	0.646	0.407	0.416	0.279	0.507	0.301	0.465	0.279
	336	0.428	0.274	0.448	0.326	0.549	0.339	0.638	0.344	0.649	0.352	0.600	0.374	0.464	0.302	0.653	0.409	0.432	0.287	0.526	0.308	0.483	0.289
	720	0.451	0.290	0.480	0.340	0.584	0.358	0.673	0.357	0.678	0.367	0.661	0.398	0.496	0.320	0.724	0.442	0.465	0.304	0.565	0.324	0.524	0.314
	Avg	0.427	0.275	0.453	0.327	0.521	0.330	0.627	0.335	0.634	0.339	0.599	0.376	0.460	0.303	0.680	0.422	0.428	0.285	0.515	0.307	0.480	0.288

6.4 Prediction Results

We present the forecasting results of UFO compared to several SOTA baselines on 8 benchmarks with various prediction lengths in Table 1. The average improvement of UFO over all baseline models is statistically significant at the 95% confidence level. Overall, UFO shows consistent and robust gains across datasets, reducing MSE by over 4.96% and MAE by over 2.23% on average. Specifically, on representative datasets, UFO achieves clear improvements: for `ETTh1` (dimension=7), it reduces MSE by 3.79% and MAE by 1.82%; for `Solar` (dimension=137), it reduces MSE by 5.60% and MAE by 1.12%; and for `Traffic` (dimension=862), it reduces MSE by 0.23% and MAE by 3.51%. These results demonstrate UFO's strong generalization across time series with different characteristics and scales. Collectively, UFO significantly improves prediction accuracy by learning the universal frequency representation.

6.5 Frequency Ablation

Leveraging UFO's ability to capture core temporal dynamics through latent basis functions, we demonstrate its adaptive prediction performance for high-frequency cutoff time series. We compare several models trained on full-spectrum `ETTh1`. Specifically, within the FFT, the original input length of 96 is mapped to 48 positive-frequency modes. As the cutoff is lowered from 48 (full spectrum) to 24, 12, and then 6, the high-frequency energy is increasingly suppressed, leaving only the lower-frequency modes. Figure 2 illustrates spectrum amplitude in the lower panel and prediction performance in the upper panel. At every frequency cutoff, UFO consistently outperforms other models, including Koopa (operator learning), FilterNet (deep frequency model), iTransformer (Transformer model), and DLinear (linear layer), exhibiting lower MSE (bold line) and MAE (dashed line). We further visualize UFO's prediction results for each frequency cutoff in Figure 3.

7 Conclusion

In this work, we introduced the Universal Frequency Operator (UFO), which learns a universal frequency representation for time series forecasting. Given that the Fourier transform decomposes time series into basis functions and coefficients, we gain insight into how neural networks can universally approximate these basis functions and coefficients. Mathematically, we prove that a neural network

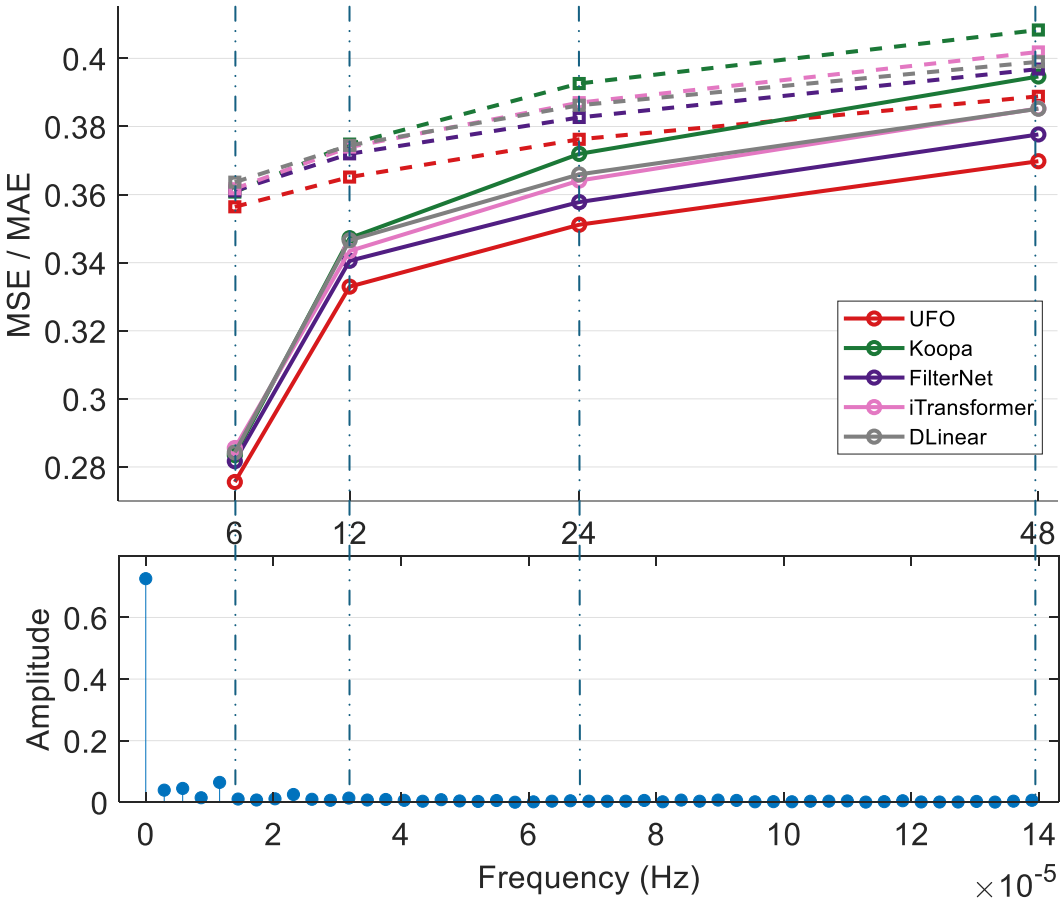


Figure 2: High-frequency cutoff prediction results in `ETTh1`. The upper panel compares model performance using MSE (bold lines) and MAE (dashed lines), while the lower panel shows frequency amplitude with cutoffs reduced from 48 (full) to 24, 12, and 6.

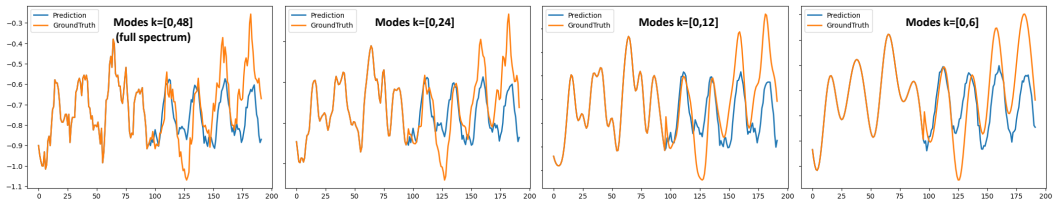


Figure 3: Illustration of prediction results on `ETTh1`. By truncating the high-frequency modes and keeping only the low-frequency modes $k = 0 \sim 6,\ 12,\ 24,\ 48$, where 48 denotes the full spectrum.

can approximate the Fourier basis, so UFO's basis functions are not constrained to fixed frequency modes or channels. Within such basis functions. Using Taylor's expansion, we prove that neural networks can learn rotation coefficients, enabling UFO to capture temporal dynamics. Experimental results indicate that UFO consistently outperforms SOTA baselines on 8 real-world datasets. We further ablate the history input's high-frequency mode, demonstrating UFO's capture of core temporal dynamics. For model backbone selection, several promising directions remain for future work. UFO uses linear layers for lifting, projection, and rotation; lifting and projection could be extended to more advanced modules, and the rotation coefficients may benefit from generative models.

REFERENCES

Omri Azencot, N Benjamin Erichson, Vanessa Lin, and Michael Mahoney. Forecasting sequential data using consistent koopman autoencoders. In *International conference on machine learning*, pp. 475–485. PMLR, 2020.

Francesca Bartolucci, Emmanuel De Bézenac, Bogdan Raonic, Roberto Molinaro, Siddhartha Mishra, and Rima Alaifari. Are neural operators really neural operators? frame theory meets operator learning. *arXiv preprint arXiv:2305.19913*, 2023.

Julius Berner, Miguel Liu-Schiaffini, Jean Kossaifi, Valentin Duruisseaux, Boris Bonev, Kamyar Azizzadenesheli, and Anima Anandkumar. Principled approaches for extending neural architectures to function spaces for operator learning. *Nature Machine Intelligence*, pp. 1–9, 2026.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Steven L Brunton, Marko Budišić, Eurika Kaiser, and J Nathan Kutz. Modern koopman theory for dynamical systems. *arXiv preprint arXiv:2102.12086*, 2021.

Andrey Bryutkin, Jiahao Huang, Zhongying Deng, Guang Yang, Carola-Bibiane Schönlieb, and Angelica Aviles-Rivero. Hamlet: Graph transformer neural operator for partial differential equations. *arXiv preprint arXiv:2402.03541*, 2024.

Qianying Cao, Somdatta Goswami, and George Em Karniadakis. Laplace neural operator for solving differential equations. *Nature Machine Intelligence*, 6(6):631–640, 2024.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE transactions on neural networks*, 6(4):911–917, 1995.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. Periodicity decoupling framework for long-term series forecasting. In *International conference on learning representations*, volume 2024, pp. 48067–48078, 2024.

Wei Fan, Shun Zheng, Xiaohan Yi, Wei Cao, Yanjie Fu, Jiang Bian, and Tie-Yan Liu. Depts: Deep expansion learning for periodic time series forecasting. *arXiv preprint arXiv:2203.07681*, 2022.

Wei Fan, Kun Yi, Hangting Ye, Zhiyuan Ning, Qi Zhang, and Ning An. Deep frequency derivative learning for non-stationary time series forecasting. *arXiv preprint arXiv:2407.00502*, 2024.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4 (2):251–257, 1991.

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

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. Sparsetsf: Modeling long-term time series forecasting with 1k parameters. *arXiv preprint arXiv:2405.00946*, 2024.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. *Advances in neural information processing systems*, 36:12271–12290, 2023.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

Abrar Majeedi, Viswanatha Reddy Gajjala, Satya Sai Srinath Namburi Gnvv, Nada Magdi Elkordi, and Yin Li. Lets forecast: Learning embedology for time series forecasting. *arXiv preprint arXiv:2506.06454*, 2025.

Henri J Nussbaumer. The fast fourier transform. In *Fast Fourier transform and convolution algorithms*, pp. 80–111. Springer, 1981.

Md Ashiqur Rahman, Zachary E Ross, and Kamyar Azizzadenesheli. U-no: U-shaped neural operators. *arXiv preprint arXiv:2204.11127*, 2022.

Jacob Seidman, Georgios Kissas, Paris Perdikaris, and George J Pappas. Nomad: Nonlinear manifold decoders for operator learning. *Advances in Neural Information Processing Systems*, 35: 5601–5613, 2022.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. *Advances in neural information processing systems*, 30, 2017.

Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, pp. 366–381. Springer, 2006.

Tapas Tripura and Souvik Chakraborty. Wavelet neural operator: a neural operator for parametric partial differential equations. *arXiv preprint arXiv:2205.02191*, 2022.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 38626–38652, 2024a.

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

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

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Zhikai Wu, Sifan Wang, Shiyang Zhang, Sizhuang He, Min Zhu, Anran Jiao, Lu Lu, and David van Dijk. Tante: Time-adaptive operator learning via neural taylor expansion. *Journal of Computational Physics*, pp. 115041, 2026.

Wei Xiong, Xiaomeng Huang, Ziyang Zhang, Ruixuan Deng, Pei Sun, and Yang Tian. Koopman neural operator as a mesh-free solver of non-linear partial differential equations. *Journal of Computational Physics*, 513:113194, 2024.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. In *International Conference on Learning Representations*, volume 2024, pp. 26295–26318, 2024.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in neural information processing systems*, 36:76656–76679, 2023.

Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. Filternet: Harnessing frequency filters for time series forecasting. *Advances in neural information processing systems*, 37:55115–55140, 2024.

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

Jingdong Zhang, Qunxi Zhu, and Wei Lin. Learning hamiltonian neural koopman operator and simultaneously sustaining and discovering conservation laws. *Physical Review Research*, 6(1): L012031, 2024a.

Xinyu Zhang, Shanshan Feng, Jianghong Ma, Huiwei Lin, Xutao Li, Yunming Ye, Fan Li, and Yew Soon Ong. Frnet: Frequency-based rotation network for long-term time series forecasting. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 3586–3597, 2024b.

Yitian Zhang, Liheng Ma, Antonios Valkanas, Boris N Oreshkin, and Mark Coates. Skolr: structured koopman operator linear rnn for time-series forecasting. *arXiv preprint arXiv:2506.14113*, 2025.

Ronghua Zheng, Hanru Bai, and Weiyang Ding. Koonpro: a variance-aware koopman probabilistic model enhanced by neural process for time series forecasting. 2025:29457–29481, 2025.

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. *arXiv preprint arXiv:2012.07436*, 2020.

Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in neural information processing systems*, 35:12677–12690, 2022a.

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

A BASELINES

We choose SOTA baselines grouped into three categories and briefly summarize the core idea of each method as follows:

- **Deep frequency model:**
  1. TimesNet (Wu et al., 2022): Employs a 2D kernel to uncover intra-period and inter-period variations in the frequency domain, and further refines and fuses these variations through an adaptive aggregation module to enhance forecasting.
  2. FreTS (Yi et al., 2023): Operates directly in the frequency domain via Discrete Fourier Transform (DFT), employing real- and imaginary-valued MLPs for cross-time and cross-variable frequency learning to lower complexity and reduce noise.
  3. FilterNet (Yi et al., 2024): Applies adaptive frequency-domain filtering through learnable passbands and attenutation weights to explicitly preserve relevant seasonal frequencies with handling high-frequency modes.
- **Operator learning model:**
  1. Koopa (Liu et al., 2023): Leverages Koopman operator theory to map non-linear time series dynamics into a linear latent space, disentangling complex time-varying patterns into linear dynamics for accurate long-term forecasting.
  2. DeepEDM (Majeedi et al., 2025): Combines empirical dynamic modeling (EDM) with deep neural networks to construct differential operators that learn non-linear attractor dynamics from complex, non-stationary multivariate time series.
  3. SKOLR (Zhang et al., 2025): Incorporates spectral Koopman operator learning with low-rank matrix representations to capture dominant operator modes while maintaining computational scalability across high-dimensional time series.
- **Other backbones:**
  1. DLinear Zeng et al. (2023): Decomposes the series into trend and seasonality components and feeds both into a single linear layer for forecasting.
  2. iTransformer (Liu et al., 2024): Employs an inverted Transformer that treats each channel (variable) as a token to primarily capture cross-variable correlations, and then applies a feed-forward network over these tokens to improve multivariate forecasting efficiency.
  3. TimeMixer (Wang et al., 2024a): Integrates past-decomposable mixing and future-multipredictor mixing to capture temporal patterns relevant to both the observed history and the prediction horizon, and further models these patterns at multiple time granularities to improve forecasting.
  4. TimeXer (Wang et al., 2024b): Builds an exogenous module that explicitly performs channel-wise attention for each channel, and integrates it with an endogenous module to learn global patterns for forecasting the exogenous variables.

B PROOF

B.1 PROOF OF THEOREM 1

Proof. The proof relies on the continuity of discrete linear operators on compact domains and the classical Universal Approximation Theorem (UAT) for neural networks. Let $\mathcal{K} \subset \mathbb{R}^d$ be a compact set of inputs. Because $\mathcal{F}$ represents a linear mapping between finite-dimensional spaces, it is bounded and thus Lipschitz continuous with some constant $L_{\mathcal{F}} > 0$. By the Continuous Mapping Theorem, the image $\mathcal{F}(\mathcal{K})$ is also compact. Similarly, the inverse transform $\mathcal{F}^{-1}$ is a bounded linear operator and thus Lipschitz continuous with constant $L_{\mathcal{F}^{-1}} > 0$.

By UAT, any continuous function defined on a compact domain can be uniformly approximated to arbitrary precision by a feedforward neural network with non-polynomial activation functions. Applying UAT to the continuous operator $\mathcal{F}$ on $\mathcal{K}$, there exists a lifting network P such that $\sup_{\mathbf{X}\in\mathcal{K}} \|P(\mathbf{X}) - \mathcal{F}(\mathbf{X})\|_{\mathrm{F}} < \epsilon$. In the same manner, applying UAT to the continuous inverse $\mathcal{F}^{-1}$ on the compact set $\mathcal{F}(\mathcal{K})$ yields a projection network Q such that $\sup_{\mathbf{Z}\in\mathcal{F}(\mathcal{K})} \|Q(\mathbf{Z}) - \mathcal{F}^{-1}(\mathbf{Z})\|_{\mathrm{F}} < \epsilon$.

Furthermore, this approximation establishes a robust end-to-end operator bound. Let $\mathcal{G}_\theta$ be an intermediate spectral operator with Lipschitz constant $L_{\mathcal{G}}$ on $\mathcal{F}(\mathcal{K})$. Comparing the exact operator $\mathbf{Y} = (\mathcal{F}^{-1} \circ \mathcal{G}_\theta \circ \mathcal{F})(\mathbf{X})$ and the neural approximation $\hat{\mathbf{Y}} = (Q \circ \mathcal{G}_\theta \circ P)(\mathbf{X})$, the error can be bounded using the triangle inequality:

$$\begin{aligned}\|\hat{\mathbf{Y}} - \mathbf{Y}\|_{\mathrm{F}} &= \|Q(\mathcal{G}_\theta(P(\mathbf{X}))) - \mathcal{F}^{-1}(\mathcal{G}_\theta(\mathcal{F}(\mathbf{X})))\|_{\mathrm{F}} \\ &\le \underbrace{\|Q(\mathcal{G}_\theta(P(\mathbf{X}))) - \mathcal{F}^{-1}(\mathcal{G}_\theta(P(\mathbf{X})))\|_{\mathrm{F}}}_{\text{Projection error } E_Q} + \underbrace{\|\mathcal{F}^{-1}(\mathcal{G}_\theta(P(\mathbf{X}))) - \mathcal{F}^{-1}(\mathcal{G}_\theta(\mathcal{F}(\mathbf{X})))\|_{\mathrm{F}}}_{\text{Propagated lifting error } E_P}.\end{aligned} \tag{21}$$

Using the bounds derived from UAT and the Lipschitz property of $\mathcal{F}^{-1}$ and $\mathcal{G}_\theta$, the projection error is strictly bounded by $E_Q < \epsilon$. The propagated lifting error is bounded by $E_P \le L_{\mathcal{F}^{-1}} L_{\mathcal{G}} \|P(\mathbf{X}) - \mathcal{F}(\mathbf{X})\|_{\mathrm{F}} < L_{\mathcal{F}^{-1}} L_{\mathcal{G}} \epsilon$. Combining both terms yields $\sup_{\mathbf{X}\in\mathcal{K}} \|\hat{\mathbf{Y}} - \mathbf{Y}\|_{\mathrm{F}} < (1 + L_{\mathcal{F}^{-1}} L_{\mathcal{G}})\, \epsilon$. Since $(1 + L_{\mathcal{F}^{-1}} L_{\mathcal{G}})$ is constant and $\epsilon > 0$ can be made arbitrarily small by expanding the capacity of P and Q, this confirms that the lifting and projection networks enable the universal approximation of the end-to-end operator on $\mathcal{K}$. ■

B.2 PROOF OF THEOREM 2

Proof. The proof decomposes the total approximation error into two components: the Taylor truncation error of the infinite series operator and the universal neural network approximation error on the truncated polynomial map.

Step 1: Uniform convergence and series truncation. By the compactness of $\mathcal{K}_B$, the supremum

$$B_* = \max_{\mathbf{B}\in\mathcal{K}_B} \|\mathbf{B}\|_{\mathrm{F}} \tag{22}$$

is well-defined and finite. Define $A = \Omega \Delta t$. Because $\mathbf{\Omega} \in \mathcal{K}_\Omega \subseteq [-\Omega, \Omega]^{\frac{k}{2}\times c}$, every entry of the block-expanded matrix satisfies $|\widetilde{\Omega}_{a,b}| \le \Omega$ for all $a \in \{1, \dots, k\}$ and $b \in \{1, \dots, c\}$. Consequently, for any $(\mathbf{B}, \mathbf{\Omega}, \Delta t) \in \mathcal{D}$, the element-wise powers satisfy $|(i\widetilde{\mathbf{\Omega}})^{\odot j}_{a,b}| \le \Omega^j$. Using the compatibility of the Hadamard product with the Frobenius norm under entry-wise bounds, the j-th term of the series satisfies

$$\left\| \mathbf{B} \odot \frac{(\Delta t)^j}{j!} (i\widetilde{\mathbf{\Omega}})^{\odot j} \right\|_{\mathrm{F}} \le \|\mathbf{B}\|_{\mathrm{F}} \frac{|\Delta t|^j}{j!} \max_{a,b} \left|\widetilde{\Omega}_{a,b}\right|^j \le B_* \frac{A^j}{j!}. \tag{23}$$

Because the scalar exponential series dominates the term-wise norm bound,

$$\sum_{j=0}^{\infty} B_* \frac{A^j}{j!} = B_* e^A < \infty, \tag{24}$$

the Weierstrass M-test establishes that the defining series for $T(\mathbf{B}, \mathbf{\Omega}, \Delta t)$ converges uniformly on the compact domain $\mathcal{D}$.

Now consider the sequence of truncated partial sums $T_J : \mathcal{D} \to \mathbb{C}^{k\times c}$, defined by

$$T_J(\mathbf{B}, \mathbf{\Omega}, \Delta t) = \mathbf{B} \odot \sum_{j=0}^{J} \frac{(\Delta t)^j}{j!} (i\widetilde{\mathbf{\Omega}})^{\odot j}. \tag{25}$$

As a finite sum of continuous functions, each T_J is continuous on $\mathcal{D}$. Because uniform limits of continuous functions preserve continuity, T is continuous on $\mathcal{D}$. By the Heine–Cantor theorem, since $\mathcal{D}$ is compact, T is also uniformly continuous on $\mathcal{D}$. Furthermore, the truncation remainder satisfies

$$\sup_{\mathcal{D}} \|T - T_J\|_{\mathrm{F}} \leq B_* \sum_{j=J+1}^{\infty} \frac{A^j}{j!}. \tag{26}$$

Since the tail of the exponential series vanishes as $J \to \infty$, for any $\epsilon > 0$, we can select a sufficiently large integer $J \geq 0$ such that $\sup_{\mathcal{D}} \|T - T_J\|_{\mathrm{F}} < \frac{\epsilon}{2}$.

Step 2: Neural network approximation. Let $\rho : \mathbb{C}^{k\times c} \to \mathbb{R}^{2kc}$ denote the canonical isometric vectorization bijection mapping a complex matrix to a real vector:

$$\rho(\mathbf{Z}) = \big(\mathrm{vec}(\mathrm{Re}\,\mathbf{Z})^\top, \mathrm{vec}(\mathrm{Im}\,\mathbf{Z})^\top\big)^\top \in \mathbb{R}^{2kc}. \tag{27}$$

This mapping preserves Euclidean norms: $\|\rho(\mathbf{Z})\|_2 = \|\mathbf{Z}\|_{\mathrm{F}}$. Under ρ, the compact domain $\mathcal{D}$ maps bijectively to a compact real subset

$$\widetilde{\mathcal{D}} = \Big\{ \big(\rho(\mathbf{B})^\top, \mathrm{vec}(\mathbf{\Omega})^\top, \Delta t\big)^\top : (\mathbf{B}, \mathbf{\Omega}, \Delta t) \in \mathcal{D} \Big\} \subset \mathbb{R}^{\frac{5kc}{2}+1}, \tag{28}$$

where $\rho(\mathbf{B}) \in \mathbb{R}^{2kc}$, $\mathrm{vec}(\mathbf{\Omega}) \in \mathbb{R}^{\frac{kc}{2}}$, and $\Delta t \in \mathbb{R}$. In these real coordinates, T_J acts as a continuous polynomial map from $\widetilde{\mathcal{D}} \subset \mathbb{R}^{\frac{5kc}{2}+1}$ into $\mathbb{R}^{2kc}$.

By the Universal Approximation Theorem for multilayer perceptrons on compact Euclidean subsets, there exists a real-valued neural network map $\widetilde{N}_\phi : \mathbb{R}^{\frac{5kc}{2}+1} \to \mathbb{R}^{2kc}$ such that

$$\sup_{\mathcal{D}} \big\| \widetilde{N}_\phi\big(\rho(\mathbf{B}), \mathrm{vec}(\mathbf{\Omega}), \Delta t\big) - \rho\big(T_J(\mathbf{B}, \mathbf{\Omega}, \Delta t)\big) \big\|_2 < \frac{\epsilon}{2}. \tag{29}$$

Constructing the complex-valued neural operator $N_\phi(\mathbf{B}, \mathbf{\Omega}, \Delta t) = \rho^{-1}\Big[\widetilde{N}_\phi\big(\rho(\mathbf{B}), \mathrm{vec}(\mathbf{\Omega}), \Delta t\big)\Big]$, the isometric property of ρ ensures that

$$\sup_{\mathcal{D}} \|N_\phi - T_J\|_{\mathrm{F}} = \sup_{\mathcal{D}} \big\| \widetilde{N}_\phi - \rho(T_J) \big\|_2 < \frac{\epsilon}{2}. \tag{30}$$

Step 3: Error aggregation. Applying the triangle inequality to combine the operator truncation error and the neural approximation error yields

$$\begin{aligned} \sup_{\mathcal{D}} \|N_\phi - T\|_{\mathrm{F}} &\leq \sup_{\mathcal{D}} \|N_\phi - T_J\|_{\mathrm{F}} + \sup_{\mathcal{D}} \|T_J - T\|_{\mathrm{F}} \\ &< \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon. \end{aligned} \tag{31}$$

This completes the proof. ■

C GENERATIVE AI DISCLOSURE

In this work, we used generative AI tools for the following tasks (all fall under the category of **recommended disclosure**):

- **Drafting paper text** (e.g., generating initial wording and rephrasing, followed by authors' manual revision and confirmation)
- **Summarizing literature** (extracting key points, followed by authors' verification against the original sources)
- **Finding related work and identifying research gaps** (using AI-assisted suggestions to guide search and synthesis, followed by authors' validation)
- **Improving readability and editing** (language polishing and structural/wording edits)
- **Formatting references** (formatting reference entries and bibliographic style; final citation details were checked by the authors)
- **Suggesting titles and keywords** (generating candidate options; final selection was made by the authors)

We did **not** use generative AI tools for the **required disclosure** tasks, including (but not limited to) generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating/assisting with mathematical claims or proofs, proposing/refining hypotheses, designing or providing feedback on methodology/experiments, implementing methods, translation, qualitative/thematic data analysis, or interpreting results.