arXiv:2609.32384v2 [cs.LG] 14 Oct 2026

TimeES: Probabilistic and Deterministic Time Series Forecasting via Evolutionary Spectra

Weiwei Ye **Renhe Jiang*** **Hangchen Liu** **Dongyuan Li** **Yoshihide Sekimoto**
The University of Tokyo *Corresponding author

Abstract

Real-world time series are inherently non-stationary, with trends, periodic patterns, and uncertainty evolving over time. While the Fourier domain offers a natural lens to model time series, current deep learning approaches do not explicitly model evolution and randomness in the Fourier spectra, which limits their ability to accurately predict both the expected trajectory and its uncertainty in non-stationary time series. Motivated by Evolutionary Spectra (ES) theory, we propose **TimeES**, a general framework that enables probabilistic and deterministic forecasting via the evolutionary spectra theory. Specifically, we derive a parameterizable evolutionary spectra formulation, recasting non-stationary random process modeling as learning an evolving representation modulated by random variables. Furthermore, we reduce the complexity of the estimated spectra from $\mathcal{O}(NM)$ to $\mathcal{O}(NK)$, where $K \ll M/2$, by exploiting Hermitian symmetry and spectral energy sparsity for frequency selection. Based on a simple linear backbone, our proposed TimeES achieves consistent state-of-the-art performance across both deterministic and probabilistic forecasting tasks, with high efficiency and interpretability. Code is available at: `https://github.com/wwy155/TimeES`.

1 Introduction

General time series forecasting involves both deterministic forecasting and uncertainty estimation, and plays a critical role in various domains, such as traffic [21, 36] and finance [20]. The frequency domain offers a natural way to model trend and periodic patterns; however, under non-stationarity, these patterns often evolve rapidly and are entangled with uncertainty [39], posing significant challenges.

Regarding deterministic forecasting, current frequency-domain approaches typically rely on analytically derived Fourier features (e.g., Fourier transform (FT) [14] or short-time Fourier transform (STFT)), and then apply models to learn mappings from these features to predictions [71, 75]. However, because the Fourier features are obtained analytically, these methods remain bound by the intrinsic assumptions of a locally invariant spectrum [50]. Consequently, they inherently fail to capture the spectra evolution that characterizes non-stationary time series [73].

In parallel, many existing probabilistic forecasting methods inherit implicit stationarity or i.i.d. assumptions from deep generative frameworks originally designed for computer vision [61], while few works explore the fundamental theoretical framework specifically designed for modeling non-stationary random processes in a natural and principled manner. In real-world time series, data often exhibit structured deterministic dynamics, such as fixed periodic patterns [23], that are highly predictable and should not be absorbed into aleatoric uncertainty [33]. By conflating these systematic components with noise, such methods may misattribute the spectral component to random fluctuations, potentially failing to distinguish deterministic patterns from stochastic variations [10].

To illustrate this, we present a case study on the ETTm1 dataset. As shown in Figure 1, the signal exhibits time-varying frequency components and substantial stochastic fluctuations. However, the FFT computed over the entire sequence fails to capture time-varying spectral contents; besides, the

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

STFT suffers from spectral leakage and limited frequency resolution, resulting in spikes between the trend and daily spectrum and a blurred spectrogram (as seen in Figure 1 (b)); for probabilistic forecasting, even recent state-of-the-art methods such as TMDM [34] and NsDiff [74] erroneously treat periodic components as stochastic part, leading to poor performance.

To address these limitations, we turn to evolutionary spectra (ES) theory [50], a cornerstone of non-stationary stochastic time series analysis that theoretically defines a time-varying spectrum but has remained underexploited in practice due to the lack of reliable methods for estimating the evolutionary spectra [5]. In particular, we derive a parameterized formulation of the evolutionary spectra, which can be interpreted as both a decompositable and generative model and thus the evolutionary spectra can be effectively estimated as a representation learning problem. Building upon the formulation, we propose TimeES, a foundational forecasting framework that offers both physical interpretable decomposition and probabilistic uncertainty quantification capabilities. As shown in Figure 1, TimeES yields a sharp and temporally resolved spectral estimate (Figure 1 (c)) and simultaneously captures the intrinsic randomness embedded in periodic patterns (Figure 1 (f)).

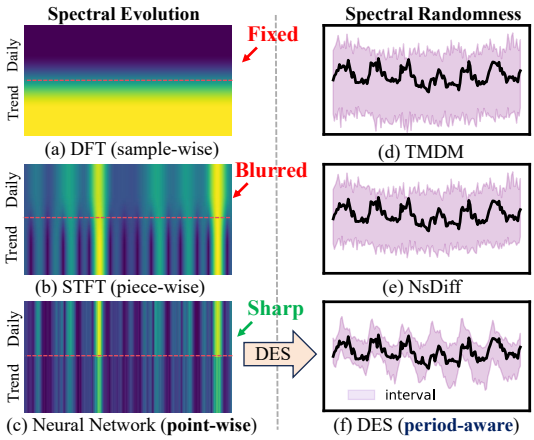


Figure 1: A case study on the ETTm1 using 7 days samples with daily and trend spectrum (0 and 1 frequency). (a-c): spectra obtained by different methods; (d-f): probabilistic forecasting results of different methods. TimeES produces point-wise spectra via neural network and quantify period-aware uncertainty interval.

In summary, our contributions are: (1) Motivated by the evolutionary spectra theory, we formulate **D**iscrete **E**volutionary **S**pectra (**DES**), a parameterized point-wise generative model for time series, in which the time-varying spectral density is explicitly represented and thus can be estimated directly from data. (2) We propose **TimeES**, a framework that estimates and predicts evolutionary spectra using DES which supports both probabilistic generation and interpretable decomposition, thereby enabling both deterministic and probabilistic forecasting. We reduce the complexity of the estimation of the evolutionary spectra from $\mathcal{O}(NM)$ to $\mathcal{O}(NK)$, where $K \ll M/2$ (Section 4.2) by exploiting Hermitian symmetry and spectral energy sparsity. (3) Through extensive experiments, we demonstrate that TimeES achieves consistent state-of-the-art performance in probabilistic and deterministic forecasting tasks. Notably, on the most challenging probabilistic forecasting benchmarks, TimeES outperforms recent baselines **17.23%** on average with high efficiency, highlighting its superior ability to forecast time series.

2 Related work

2.1 Evolutionary Spectra for Non-stationary Process

To formally account for non-stationarity, Priestley proposed evolutionary spectra (ES) theory [49, 50]. However, the ES is fundamentally non-identifiable due to infinite many valid solutions. There exist some techniques to estimate the ES, though these depend on very limiting conditions such as local stationarity or narrow-bandness [55, 59, 60, 64]. Recent approaches rely on perturbation-based optimization methods [5], requiring a strict initialization. Our proposed method estimates the ES directly from the data, by formulating the ES estimation as a representation learning problem.

2.2 Deterministic Time Series Forecasting

Recent years have witnessed significant advances in deep learning for deterministic time series forecasting, architectures such as RNNs [19] and Transformers [82] have demonstrated strong performance by capturing temporal dependencies through sequential or attention-based mechanisms [28, 41, 69, 70]. To capture the inherent periodic patterns, recent frequency-aware models [71, 75, 83] explicitly incorporate spectral components to improve forecasting performance and

efficiency. To further address non-stationarity, recent works explicitly model distributional or frequency shifts [31, 73]. To model complex dynamics beyond deterministic assumptions, Koopman operator theory [40, 65] and Dynamic Mode Decomposition (DMD) [56] offer powerful alternatives by providing linear representations of nonlinear dynamical systems in lifted feature spaces. Koopman models also learn continuous spectra [42] and admit probabilistic formulations [44, 47, 81], but their spectra decompose an evolution operator in a lifted space; evolutionary spectra [49] instead model the time-varying power spectra of the observed non-stationary random process itself, making them well-suited for data with non-stationary stochasticity (Appendix I).

2.3 Probabilistic Time Series Forecasting

Early efforts built on GANs [46, 78] or VAEs [16] to capture the temporal structure, but their training instability has motivated a shift toward other alternative generative paradigms. Specifically, diffusion-based models, have recently been adapted to time series due to their strong performance in modeling complex distributions [11, 52, 57, 58, 62, 79]. However, most of these methods assume stationary distributions. To address this, NsDiff [74] allows a varying endpoint distribution, thereby better capture non-stationarity. Nevertheless, nearly all existing approaches rely on methods originally developed for computer vision, which assume that data are generated i.i.d. In contrast, TimeES is explicitly grounded in the premise that data arise from a non-stationary random process.

We discussed related works concerning major methodology; refer to Appendix A for more details.

3 Preliminary

Throughout this paper, we use uppercase letters (e.g., X) to denote random variables, lowercase letters for scalars, and lowercase boldface letters for vectors. Specifically, we denote an input sample as $\mathbf{x} \in \mathbb{R}^N$ and the corresponding output sample as $\mathbf{y} \in \mathbb{R}^O$. We begin by considering a continuous-parameter stochastic process $\{X_t : t \in \mathbb{R}\}$, and review key preliminaries from non-stationary stochastic process and the evolutionary spectra theory.

In the stationary case, consider a continuous-time stochastic process $\{X_t\}$ (e.g., white noise). Since realizations of such a process are typically not absolutely integrable over $(-\infty, \infty)$, the classical Fourier integral does not exist. The following theorem provides a general extension by introducing a random spectral measure defined as the followings:

Theorem 3.1 (Spectrum Representation Theorem)**.** *The spectrum representation of a zero-mean, continuous, stationary process $\{X_t\}$ is:*

$$X_t = \int_{-\infty}^{\infty} e^{i\omega t} dZ(\omega), \tag{1}$$

where $Z(\omega)$ is a *complex random measure* satisfying $dZ(\omega) = \mathcal{O}(\sqrt{d\omega})$. *Particularly, X_t admits a white noise process $\mathcal{N}_t(\mathbf{0}, \mathbf{I})$ if we sample $dZ(\omega)$ from a standard complex Gaussian $\mathcal{CN}(\mathbf{0}, \mathbf{I})$.* The transform expressed is called the Fourier–Stieltjes transform, as opposed to the classical Fourier transform. To facilitate subsequent results, we provide a brief proof and illustration of Theorem 1 in Appendix B.1 (see Priestley [50], Sections 4.7–4.11, for detailed mathematical treatment).

In real-world settings, data-generating processes are often non-stationary, as a process is deemed non-stationary when its statistical and spectral properties vary over time. To address this, Priestley [50] proposed evolutionary spectral theory, which extends Equation (1) to the non-stationary regime. Specifically, for a non-stationary process with time-varying distributions and frequencies, it is tempting to define a varying quantity as proposed as:

Definition 3.2 (Evolutionary Spectra)**.** *Given a non-stationary random process $\{X_t\}$, assuming a deterministic square-integrable function $A(t, \omega)e^{i\omega t}$ and a random measure $dZ(\omega)$, X_t is defined as:*

$$X_t = \int_{-\infty}^{\infty} A(t, \omega) e^{i\omega t} dZ(\omega), \tag{2}$$

where $A(t, \omega)$ is the complex amplitude and the energy spectral density is defined as $dS(t, \omega) = |A(t, \omega)|^2$. The representation in Definition 3.2 provides a principled foundation for modeling non-stationary random process through time-varying spectral characteristics. Identifying evolutionary

spectra has significant practical importance [26], and is critical for effective forecasting, as they reflect the non-stationary random structures across the Fourier spectra with random consideration. However, in most scenarios, the true evolutionary spectra are unknown and admit infinite amounts of solutions, which poses a major challenge [59]. In this paper, we propose a tractable formulation of this equation so the ES can be estimated through representation learning.

4 Methodology

In this section, we present the proposed framework, TimeES. We first discuss the main theoretical result underlying the proposed generative model, DES, which reformulates evolutionary spectra estimation as a tractable representation learning problem. Next, we describe how this model effectively estimates the evolutionary spectra and leverages it for general time series forecasting tasks. Then, we discuss how TimeES can degenerate to the DFT and STFT. An overview of the TimeES framework is provided in Figure 2.

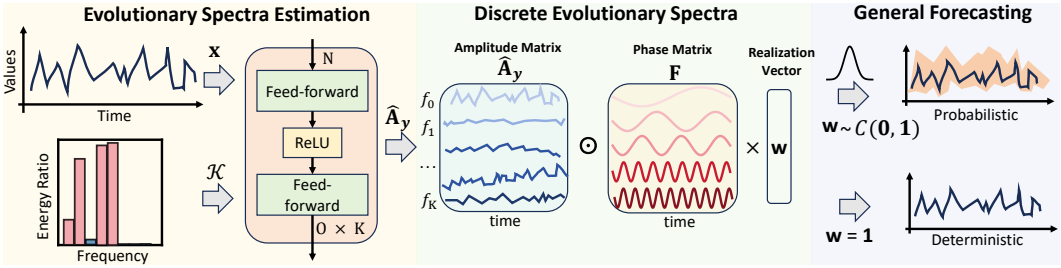


Figure 2: The overview of TimeES. TimeES estimates the interpretable amplitude matrix $\mathbf{A}$ by a neural network; then the DES is used to synthesize time series with a physical phase matrix $\mathbf{F}$ and a probabilistic realization vector $\mathbf{w}$, enabling general time series forecasting.

4.1 Discrete Evolutionary Spectra

Definition 3.2 establishes a direct relationship between the instantaneous amplitude function $A(t, \omega)$ and the continuous-time process X_t at each time t. Motivated by this connection, we formulate Definition 3.2 as a parameterized generative model, which is the main theorem of this paper:

Theorem 4.1 (Discrete Evolutionary Spectra)**.** *Given discretely sampled data* $\{X_n\}_{n=1}^N$*, define a set of discrete frequencies* $\omega_k = \frac{2\pi k}{M}, \quad k = 0, 1, \ldots, M-1$*. Then* X_n *admits:*

$$X_n = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} A(n, \omega_k)\, W_k\, e^{i2\pi k n/M}, \tag{3}$$

where $W_k \sim \mathcal{CN}(\mu_{W_k}, \sigma_{W_k})$ are independent complex Gaussian random variables, and $A(n, \omega_k)$ denotes the evolutionary amplitude spectrum at time n and frequency ω_k. We provide the proof in Appendix B.2. The parameter M can be seen as the frequency length in FFT or STFT, we thus follow this convention in our experiments to use $M = N$ to cover the full discrete frequencies. Note that, this equation describes a single realization of the stochastic process $\{X_n\}$ corresponding to a particular draw of the complex random variables W_k. By sampling W_k multiple times, we can perform generative modeling of the signal. *In particular, in the special case* $A = 1$*,* X_n *is a random variable following a standard normal distribution as stated in Definition 3.2.*

Importantly, unlike many prior approaches that adopt score-based generative models from computer vision which typically assume that data are drawn from instance-independent distributions, DES instead assumes that the observed time series is sampled from an oscillatory non-stationary process. This formulation is therefore far better suited to capturing the inherent non-stationary distribution in time series data. Presenting Theorem 4.1 in matrix form will be more convenient for later text:

Proposition 4.2. *Theorem 4.1 can be equivalently expressed as*

$$\mathbf{x} = \frac{1}{\sqrt{M}} \left(\mathbf{A} \odot \mathbf{F} \right) \mathbf{w}, \tag{4}$$

where $\mathbf{x} = [x_0, \ldots, x_{N-1}]^\top \in \mathbb{C}^N$, $\mathbf{w} = [W_0, \ldots, W_{M-1}]^\top \in \mathbb{C}^M$, $\mathbf{A}_{n,k} = A(n, \omega_k)$, $\mathbf{F}_{n,k} = e^{i2\pi kn/M}$. In Equation (4), $\odot$ denotes the Hadamard product. To this end, we establish a generative model for non-stationary time series where $\mathbf{F}$ represents the physical frequency basis, $\mathbf{A}$ assigns evolutionary amplitude to each of these components; $\mathbf{w}$ encodes the probabilistic nature of the signal, so that any observed realization is a sample path of the underlying random process.

4.2 Probabilistic and Deterministic Forecasting via Evolutionary Spectra

Currently, there is no analytic or numerical way to determine evolutionary spectra [5]. Given that Theorem 4.1 establishes a point-wise and tractable relationship between the evolutionary spectrum and the sampled time series X_n, we consider a natural learning strategy: train a neural network to estimate $\mathbf{A}$ by matching the reconstructed signal with observed data. In principle, this approach requires a network parameterization with an output size of $\mathcal{O}(NM)$; and to capture all possible spectral components present in the data, one would set $M \geq N$ (e.g., $M = N$ for a full DFT basis), resulting in an output size of at least $\mathcal{O}(N^2)$. However, we show that by exploiting two inherent properties of real-world time series (Hermitian symmetry and spectral sparsity) the computational complexity can be reduced to $\mathcal{O}(NK)$, where $K \ll M/2$.

Hermitian Symmetry. Since most real-world time series are inherently real-valued, this requirement imposes a structural constraint on the complex amplitude spectrum $A(n, \omega_k)$ in Theorem 4.1. Specifically, *Hermitian symmetry* [53] must hold across the frequency dimension to ensure that the inverse spectral synthesis yields a real-valued sequence, which can be defined as:

Theorem 4.3. *Let the signal X_n be synthesized as Equation* (3) *and define $B(n, k) = A(n, \omega_k)W_k$. Then $X_n \in \mathbb{R}$ for all n if and only if the following Hermitian symmetry condition holds for all n and all $k = 0, 1, \ldots, M-1$:*

$$B(n, \omega_k) = \overline{B(n, \omega_{(M-k) \bmod M})}. \tag{5}$$

See the proof in Appendix B.3. In particular: The DC component ($k = 0$) must be real: $A(n, \omega_0) \in \mathbb{R}$. If M is even, the Nyquist component ($k = M/2$) must also be real. Consequently, the entire spectrum is uniquely determined by the amplitudes at the non-redundant frequency indices $\{0, 1, \ldots, \lfloor \frac{M}{2} \rfloor\}$, with total $M_h = \lfloor \frac{M}{2} \rfloor + 1$ frequencies. By exploiting Hermitian symmetry in Theorem 4.1, we only need to estimate half of the frequency components.

Spectral energy sparsity. Moreover, in time series, not all frequency components contain useful information [18]. While many prior methods exploit this property by selecting frequencies based on fixed thresholds [71, 83] or a pre-specified number of top components [70, 73], the number of frequencies K is often heuristic and lacks principled guidance. Motivated by the observation that real-world signals typically concentrate the majority of their energy $\|\mathbf{x}\|^2$ in only a few frequencies, we propose an *energy-based frequency selection strategy*. Specifically, we first compute the average periodogram then sort all frequencies by their energy $I(k)$ in descending order and select the smallest subset of frequencies $\mathcal{K}$ such that the cumulative energy accounts for at least r of the total:

$$I(k) = \mathbb{E}_{x \sim \mathcal{D}^{\text{train}}} \left[|\hat{X}_k|^2 \right], \quad \frac{\sum_{k \in \mathcal{K}} I(k)}{\sum_{k=0}^{M-1} I(k)} \geq r, \tag{6}$$

where $\hat{X}_k$ are the Fourier coefficients obtained by using Fourier Transform on each training sample, $I(k)$ is the periodogram at frequency k, $r \in (0, 1)$ is the intended energy ratio. This subset $\mathcal{K}$ captures the dominant oscillatory components of the data. In practice, for sequences of Fourier length $M = 96$, this typically yields $K = 2$ to 5 dominant frequencies across datasets as presented in Appendix C.3. Compared to previous works [71, 75, 83] choosing this set $\mathcal{K}$ heuristically on each dataset, we show that a fixed ratio r can adapt to data signal-noise-ratio in Appendix B.5.

A Simple Linear Backbone. Together, rather than model the entire matrix with $O(NM)$ complexity, we select only a small set of frequencies $\{\omega_1, \omega_2, \ldots, \omega_K\}$ from $\{\omega_0, \omega_1, \ldots, \omega_{\lfloor M/2 \rfloor}\}$. Then, we directly estimate the ES of the forecasting series $\mathbf{y}$ by constructing an estimation $\hat{\mathbf{A}}_\mathbf{y}$:

$$[\hat{\mathbf{A}}^r_\mathbf{y}, \hat{\mathbf{A}}^i_\mathbf{y}] = \boldsymbol{\Theta}_2 \, \mathrm{ReLU}\big(\boldsymbol{\Theta}_1 \, \mathbf{x} + \mathbf{b}_1\big) + \mathbf{b}_2 \in \mathbb{R}^{N \times 2K}, \tag{7}$$

where $\hat{\mathbf{A}}^r_\mathbf{y}, \hat{\mathbf{A}}^i_\mathbf{y} \in \mathbb{R}^{N \times K}$ denote the real and imaginary parts of the amplitude matrix, respectively, and the complex-valued representation is given by $\hat{\mathbf{A}}_\mathbf{y} = \hat{\mathbf{A}}^r_\mathbf{y} + i\hat{\mathbf{A}}^i_\mathbf{y}$. Here $\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \mathbf{b}_1, \mathbf{b}_2$ are

network parameters, and the output dimension is $2NK$ rather than $2NM$: the j-th column of $\hat{\mathbf{A}}_{\mathbf{y}}$ corresponds to the j-th selected frequency ω_{k_j}, and $\mathbf{F}$ in Equation (4) is restricted to the same columns, $\mathbf{F}_{n,j} = e^{i\omega_{k_j} n}$, so frequencies outside $\mathcal{K}$ carry zero amplitude by construction. Our experiments show that this simple backbone is sufficient to achieve state-of-the-art performance with high efficiency; see Appendix D for more experimental details.

Probabilistic and Deterministic Time Series Forecasting. Notably, Equation (3) admits two interpretations depending on the treatment of the latent variables W_k, enabling deterministic and probabilistic modeling across tasks. If we assume that the entire sequence is generated by a set of fixed oscillatory components (e.g., by fixing $W_k = 1$ for all k), the representation reduces to a deterministic time–frequency decomposition of the given signal. In contrast, when $W_k \sim \mathcal{CN}(0, 1)$ are treated as random variables, the same formulation becomes a generative model for non-stationary stochastic processes. We summarize these two perspectives in Table 1.

In practice, μ_{W_k}, σ_{W_k} are learnable constants for each selected frequency and channel, shared across the forecast horizon, so that the horizon dependence of the predictive distribution enters only through $\hat{\mathbf{A}}$ (Appendix J.2). For probabilistic forecasting, sampling from a distribution incurs a higher computational cost due to the need to sample and store latent variables $\mathbf{w}$. In contrast, the deterministic decomposition avoids this overhead. It enables efficient, lightweight inference while providing a physically interpretable representation that directly encodes time-varying spectral energy through the amplitude $\mathbf{A}$. It is worth mentioning that, the estimated evolutionary spectrum $\hat{\mathbf{A}}$ itself serves as a physically interpretable representation for various downstream tasks. We explain and evaluate this capability via an interpretability analysis in Section 5.4 and a time series classification task in Appendix C.7.

Table 1: Deterministic vs. Probabilistic Interpretation.

Interpretation	$\mathbf{y}$	$\mathbf{w}$
Deterministic	$\frac{1}{\sqrt{M}}(\hat{\mathbf{A}}_{\mathbf{y}} \odot \mathbf{F})\mathbf{1}$	$W_k = 1,\ \forall k$
Probabilistic	$\frac{1}{\sqrt{M}}(\hat{\mathbf{A}}_{\mathbf{y}} \odot \mathbf{F})\mathbf{w}$	$W_k \sim \mathcal{CN}(\mu_{W_k}, \sigma_{W_k})$

4.3 Connections with DFT and STFT

The proposed framework TimeES subsumes classical Fourier methods as special cases. When the amplitude spectrum is time-invariant, i.e., $A(n, \omega_k) = \widehat{X}_k$ for all n, DES reduces to the inverse DFT. When the signal is partitioned into non-overlapping blocks of length L and $A(n, \omega_k)$ is constant within each block (i.e., piecewise time-invariant), DES recovers the inverse STFT with a rectangular window. Thus, DFT and STFT correspond to fixed, analytically computed amplitude spectra with deterministic weights ($W_k = 1$), whereas DES learns $A(n, \omega_k)$ and supports stochastic synthesis, we summarize how DES can degenerate to DFT and STFT.

As summarized in Table 2, TimeES, DFT, and STFT are closely related. In fact, under the local stationarity assumption and with a normalized window, the continuous-time STFT approximates the evolutionary spectrum: $X_{\mathrm{STFT}}(t, \omega) \approx A(t, \omega)$, providing a localized estimate of its instantaneous spectral content. A theoretical justification is given in Appendix B.4.

Table 2: Special cases of the $A(n, \omega_k)$ and W_k

Method	$A(n, \omega_k)$	W_k
iDFT	$(\mathbf{F}\mathbf{x})_k$	1
iSTFT	$(\mathbf{F}\mathbf{x}^{(m)})_k,\ m = \lfloor n/L \rfloor$	1
DES (ours)	$\mathrm{MLP}_{n,k}(\mathbf{x})$	$\mathcal{CN}(\mu_{W_k}, \sigma_{W_k})$

5 Experiments

To verify the performance of TimeES as both probabilistic and deterministic forecasting framework, we conduct experiments on three types of tasks: multivariate probabilistic forecasting, univariate and multivariate deterministic time series forecasting. We provide details of the experiment in Appendix C. Among these, probabilistic forecasting is particularly challenging due to the lack of generative models specifically designed for time series; as a result, current strong baselines are all based on architectures originally developed for computer vision (e.g., diffusion models) [74], which may not adequately account for non-stationary time series.

5.1 Probabilistic Time Series Forecasting

Setup. We follow previous work [74] to assess the effectiveness of our proposed method, TimeES, in multivariate probabilistic forecasting task. We conduct comprehensive experiments across nine widely used real-world benchmarks including ETT{m1,m2,h1,h2}, ECL, EXG, Traffic, and Solar with input length 168 and forecast horizon 192 (36 on ILI). We assess both uncertainty quantification (CRPS, QICE) and forecasting accuracy (MSE and MAE) and compare against recent models including TimeGrad [52], CSDI [62], TimeDiff [57], DiffusionTS [79], TMDM [34], and NsDiff [74]. We set M equal to the input length N to cover all frequencies and set $r = 0.9$ to capture the main energy frequency components. All other experiments follow this setting. For training, we sample 100 times and calculate the mean values to match the ground truth. We provide a detailed hyperparameter sensitivity analysis in Appendix E.

Table 3: Experimental results on nine real-world datasets. **Bold face** indicates the best result, and underline indicates the second-best result.

Models	Datasets	ETTh1	ETTh2	ETTm1	ETTm2	ECL	EXG	ILI	Solar	Traffic
TimeGrad	CRPS	0.606	1.212	0.647	0.775	0.397	0.826	1.140	0.293	0.407
(2021)	MSE	1.062	3.462	1.218	1.690	0.505	1.567	4.197	0.475	0.983
CSDI	CRPS	0.492	0.647	0.524	0.817	0.577	0.855	1.244	0.432	1.418
(2022)	MSE	0.949	1.226	1.002	1.723	1.007	1.701	4.515	0.763	1.731
TimeDiff	CRPS	0.465	0.471	0.464	0.316	0.750	0.433	1.153	0.700	0.771
(2023)	MSE	0.517	0.456	0.537	0.268	0.879	0.402	3.958	0.821	1.350
DiffusionTS	CRPS	0.603	1.168	0.574	1.035	0.633	1.251	1.612	0.470	0.668
(2024)	MSE	1.089	3.273	1.030	2.372	1.072	3.628	6.053	0.749	1.473
TMDM	CRPS	0.452	0.383	0.375	0.289	0.461	0.336	0.967	0.350	0.557
(2024)	MSE	0.696	0.512	0.494	0.315	0.257	0.334	3.636	0.250	0.679
NsDiff	CRPS	0.392	0.358	0.346	0.256	0.290	0.324	**0.806**	0.300	0.378
(2025)	MSE	0.594	0.514	0.488	0.281	0.209	0.300	**2.846**	0.242	0.637
TimeES	CRPS	**0.349**	**0.320**	**0.310**	**0.230**	**0.205**	**0.270**	0.834	**0.193**	**0.245**
(ours)	MSE	**0.474**	**0.376**	**0.371**	**0.235**	**0.180**	**0.242**	3.052	**0.179**	**0.456**

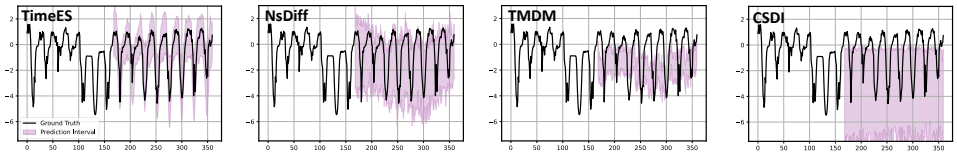


Figure 3: A probabilistic forecasting case of ETTh1 testing sample. We present 96% prediction intervals as the purple areas.

Results. We report CRPS and MSE in Table 3 and full results including QICE and MAE can be found in Appendix F. As shown in Table 3, TimeES achieves great performance, surpassing the recent diffusion-based approach NsDiff specifically designed for non-stationarity, achieving state-of-the-art results in both deterministic prediction (MSE) and uncertainty quantification (CRPS). On average, TimeES achieves a 17.23% reduction in CRPS and an 18.66% reduction in MSE across all datasets. The improvement is particularly pronounced on the Traffic dataset, which exhibits strong non-stationarity [73, 74] where TimeES reduces CRPS by 35.15% and MSE by 28.34%, demonstrating its ability to effectively model evolving spectral dynamics and non-stationary stochastic patterns. The only exception is the ILI dataset, where gains are marginal due to its limited sample size, which hinders reliable estimation of ES. Notably, TimeES achieves the highest efficiency in generating samples; refer to Appendix D for full efficiency analysis.

Sample Cases. To provide a clearer understanding of TimeES's capabilities, we visualize a representative prediction on the ETTh1 dataset. As shown in Figure 3, compared to other baselines, TimeES effectively captures both the time-varying uncertainty and the underlying frequency dynamics. Specifically, TMDM and CSDI fail to model the non-stationarity between the input and output, resulting in poorly calibrated predictive variances. Although NsDiff better captures the non-stationary patterns of uncertainty, it still fails to identify the deterministic spectral components. In contrast, TimeES is

the only model that accurately recovers both the deterministic frequency structure and the evolving uncertainty. We provide additional cases in Appendix G.

5.2 Deterministic Time Series Forecasting

Setup. We follow previous work [41] and evaluate on eight benchmarks: ETT{m1,m2,h1,h2}, ECL, EXG, Traffic, Weather, and Solar Energy. We use an input length of 96 and predict horizons of {96, 192, 336, 720}, reporting MSE and MAE. To fully assess our model's capability in modeling temporal dependency, we conduct an extra single variate forecasting experiments on each dataset by using only the first variable to predict its own future values. For multivariate forecasting, we train the model in a channel-independence setting [48] to independently predict each variate. We compare TimeES against the following representative baselines: Autoformer [69], FEDformer [83], TimesNet [70], FITS [71], FreTS [75], PatchTST [48], and iTransformer [41]. Among them, FEDformer, TimesNet, FITS, and FreTS are representative models that leverage FT or wavelet transforms for frequency modeling. We provide other additional baselines in Appendix C.5.

Table 4: Experimental results on forecasting, we report MSE. **Bold face** and underline indicate the best and second-best results. S.V denotes average MSE results on single forecasting and M.V. denotes average MSE results on multivariate forecasting.

Models	Autoformer		FEDformer		TimesNet		FITS		FreTS		PatchTST		iTransformer		TimeES (Ours)	
Datasets	S.V.	M.V.	S.V.	M.V.	S.V.	M.V.	S.V.	M.V.	S.V.	M.V.	S.V.	M.V.	S.V.	M.V.	S.V.	M.V.
ETTm1	1.176	0.580	0.840	0.448	0.811	0.400	0.793	0.418	0.755	0.414	0.763	**0.386**	0.778	0.407	**0.740**	0.390
ETTm2	0.498	0.327	0.495	0.305	0.480	0.291	0.478	0.285	0.452	0.323	0.462	0.281	0.487	0.288	**0.434**	**0.278**
ETTh1	0.925	0.496	0.882	**0.440**	0.929	0.458	1.155	0.534	1.010	0.504	0.970	0.469	0.968	0.454	**0.878**	**0.440**
ETTh2	0.700	0.450	0.713	0.437	0.659	0.414	0.656	0.396	0.596	0.500	0.633	0.387	0.682	0.383	**0.583**	**0.379**
ECL	0.679	0.227	0.610	0.214	0.591	0.193	0.587	0.328	0.530	0.205	0.543	0.216	0.601	**0.178**	**0.516**	0.197
EXG	0.827	0.613	0.765	0.519	0.711	0.416	0.762	0.389	0.517	0.422	0.693	0.367	0.713	0.360	**0.356**	**0.357**
Traffic	1.019	0.628	0.566	0.610	0.564	0.620	1.164	0.827	0.576	0.555	0.521	0.555	0.500	**0.428**	**0.431**	0.478
Weather	0.757	0.338	0.654	0.309	0.641	0.259	0.630	0.254	0.523	0.243	0.640	0.259	0.645	0.258	**0.513**	**0.239**
Solar	3.046	0.885	0.252	0.292	0.216	0.301	0.330	0.403	0.214	0.255	**0.191**	0.270	0.200	**0.233**	0.205	0.243

Main Results. We report MSE results in single variate and multivariate forecasting tasks in Table 4, full results across steps are in Appendix F. Specifically, TimeES achieves state-of-the-art results on 8 out of 9 datasets in single variate forecasting. Notably, FreTS, an MLP-based model that leverages the STFT, and FEDformer, an architecture based on wavelet decomposition that also explicitly models time-frequency variations, are both outperformed by TimeES. On average, TimeES improves upon FreTS by 19.38% and 16.00%, and surpasses FEDformer by 9.98% and 12.28%, respectively, in single variate and multivariate settings. These gains highlight the importance of modeling frequency components at a finer granularity beyond conventional representations. Also, TimeES achieves the highest efficiency in the inference step; refer to Appendix D for efficiency analysis. We provide an additional synthetic experiment on a chirp signal in Appendix C.6.

5.3 Ablation Study

Main Ablation. To analyze the effectiveness of TimeES'components, we conduct ablation studies across deterministic 720 steps forecasting, and probabilistic forecasting tasks on ETTh1 dataset. The experiment settings are consistent with Section 5.1 and 5.2. Specifically, we include two types of variants: (1) **w/o E** (without evolution): We enforce identical spectra at all forecasting steps to simulate a static spectral representation. (2) **w/o DES** (without DES): We replace the DES with a simple linear layer that directly outputs predictions. The results are presented in Table 5, where it shows that TimeES

Table 5: Experimental results on different variants of TimeES on ETTh1 dataset. × indicates that the variant lacks the DES and is therefore not generative. P.F. denotes probabilistic forecasting.

Variants	S.V. (MSE)	M.V. (MSE)	P.F. (CRPS)
w/o DES	0.974±0.014	0.554±0.006	×
w/o E	1.282±0.012	0.604±0.010	0.466±0.009
TimeES	**0.929±0.009**	**0.485±0.003**	**0.349±0.007**

achieves the best performance across all three tasks. Notably, although omitting DES increases prediction variance, it still yields a modest gain over w/o E: relative MSE improvements of 24.02% and 8.27% on single variate and multivariate long-term forecasting tasks, respectively. This further underscores the importance of modeling the evolutionary spectra. Full results are in Appendix C.4.

Plug-and-Play Testing. We further evaluate TimeES as a plug-in enhancement module within a decomposition framework by setting $\mathbf{w} = \mathbf{1}$ (see Table 1). Specifically, we replace the default backbone with PatchTST and iTransformer and compare their enhanced variants against the original models on the ETTh1 dataset in multivariate long-term forecasting. As shown in Table 6, TimeES consistently improves all backbone models. When integrated with iTransformer, the average MSE reductions become increasingly pronounced as the horizon grows: 0.56%, 1.88%, 2.44%, and 5.83% for horizons 96, 192, 336, and 720, respectively. Full ETT results are provided in Appendix F.

Table 6: Backbone enhancement on the ETTh1 dataset, TimeES is integrated as a plug-in module.

Model	PatchTST		+TimeES		Improvement		iTransformer		+TimeES		Improvement	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	0.414	0.419	**0.385**	**0.402**	6.89%	4.15%	0.386	0.405	**0.384**	**0.402**	0.56%	0.78%
192	0.460	0.445	**0.434**	**0.437**	5.65%	1.86%	0.441	0.436	**0.433**	**0.428**	1.88%	1.77%
336	0.501	0.466	**0.451**	**0.457**	9.99%	2.02%	0.487	0.458	**0.475**	**0.451**	2.44%	1.51%
720	0.500	0.488	**0.482**	**0.479**	3.68%	1.82%	0.503	0.491	**0.474**	**0.480**	5.83%	2.18%
Avg.	0.469	0.455	**0.438**	**0.444**	6.56%	2.42%	0.454	0.448	**0.441**	**0.440**	2.84%	1.59%

5.4 Interpretability Analysis

Our learned ES can be used to reconstruct signals by separately synthesizing components from different frequency bands (like DFT), which offers strong interpretability. To investigate the interpretability of TimeES, we reconstruct high- and low-frequency signals using the learned ES and examine whether each component captures the expected characteristics. Specifically, we conduct an experiment on the ETTh1 dataset, with results visualized in Figure 4. As shown, the high- and low-frequency components encode distinctly different information: the low-frequency component clearly captures the long-term trend, while the high-frequency component accurately reflects abrupt changes and fine-grained dynamics. We provide more results in Appendix H.

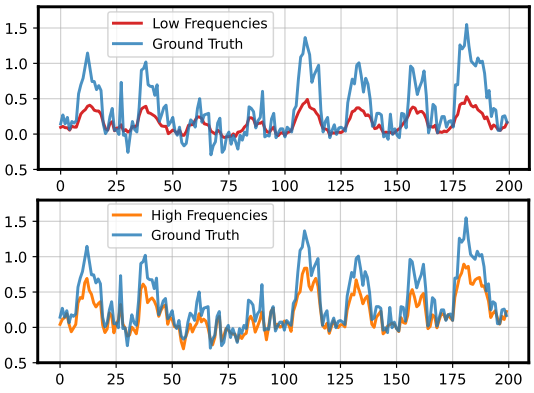


Figure 4: Low- and high-frequency components analysis on ETTh1 dataset.

6 Conclusion

We introduce **TimeES**, a novel framework that bridges ES theory with deep learning for both probabilistic and deterministic forecasting. By exploiting Hermitian symmetry and spectra energy sparsity, we reduce complexity from $\mathcal{O}(NM)$ to $\mathcal{O}(NK)$ with $K \ll M/2$. To the best of our knowledge, TimeES represents the first work to harness evolutionary spectra theory and employ deep neural networks for modeling non-stationary random process.

Limitation & Future work Our frequency selection strategy may not be effective under high-noise conditions, as spectral energy may not concentrate at a few frequencies. Learning evolutionary spectra also requires sufficient data (Appendix J.4). Additionally, the current implementation employs a standard architecture. Future work can focus on noise-robust frequency selection, limited-data learning, non-Gaussian or mixture spectral measures, and specialized architectures that better capture the temporal dynamics of evolutionary spectra while maintaining computational efficiency.

References

[1] Ahmed Alaa, Alex James Chan, and Mihaela van der Schaar. Generative time-series modeling with Fourier flows. In *International Conference on Learning Representations*, 2021.

[2] Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Transactions on Machine Learning Research*, 2022.

[3] Jont B Allen and Lawrence R Rabiner. A unified approach to short-time Fourier analysis and synthesis. *Proceedings of the IEEE*, 65(11):1558–1564, 1977.

[4] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

[5] Brett A Benowitz, Michael D Shields, and George Deodatis. Determining evolutionary spectra from non-stationary autocorrelation functions. *Probabilistic Engineering Mechanics*, 41:73–88, 2015.

[6] E Oran Brigham. *The fast Fourier transform and its applications*. Prentice-Hall, Inc., 1988.

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[8] Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. Freeway performance measurement system: mining loop detector data. *Transportation Research Record*, 1748(1):96–102, 2001.

[9] Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Multi-scale transformers with adaptive pathways for time series forecasting. In *International Conference on Learning Representations*, 2024.

[10] Yifan Chen, Mark Goldstein, Mengjian Hua, Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Probabilistic forecasting with stochastic interpolants and Föllmer processes. In *International Conference on Machine Learning*, 2024.

[11] Yu Chen, Wei Deng, Shikai Fang, Fengpei Li, Nicole Tianjiao Yang, Yikai Zhang, Kashif Rasul, Shandian Zhe, Anderson Schneider, and Yuriy Nevmyvaka. Provably convergent Schrödinger bridge with applications to probabilistic time series imputation. In *International Conference on Machine Learning*, pages 4485–4513. PMLR, 2023.

[12] Dawei Cheng, Fangzhou Yang, Sheng Xiang, and Jin Liu. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition*, 121:108218, 2022.

[13] Razvan-Gabriel Cirstea, Darius-Valer Micu, Gabriel-Marcel Muresan, Chenjuan Guo, and Bin Yang. Correlated time series forecasting using multi-task deep neural networks. In *Proceedings of the 27th acm international conference on information and knowledge management*, pages 1527–1530, 2018.

[14] Leon Cohen. *Time-frequency analysis*. Prentice Hall PTR, Englewood Cliffs, NJ, 1995.

[15] Jerome T Connor, R Douglas Martin, and Les E Atlas. Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks*, 5(2):240–254, 1994.

[16] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. TimeVAE: A variational auto-encoder for multivariate time series generation. *arXiv preprint arXiv:2111.08095*, 2021.

[17] Zheng Dong, Renhe Jiang, Haotian Gao, Hangchen Liu, Jinliang Deng, Qingsong Wen, and Xuan Song. Heterogeneity-informed meta-parameter learning for spatiotemporal time series forecasting. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 631–641, 2024.

[18] David L. Donoho, Martin Vetterli, Ronald A. DeVore, and Ingrid Daubechies. Data compression and harmonic analysis. *IEEE transactions on information theory*, 44(6):2435–2476, 1998.

[19] Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. AdaRNN: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 402–411, 2021.

[20] Yitong Duan, Lei Wang, Qizhong Zhang, and Jian Li. FactorVAE: A probabilistic dynamic factor model based on variational autoencoder for predicting cross-sectional stock returns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4468–4476, 2022.

[21] Alireza Ermagun and David Levinson. Spatiotemporal traffic forecasting: review and proposed directions. *Transport Reviews*, 38(6):786–814, 2018.

[22] Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. Dish-TS: a general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 7522–7529, 2023.

[23] Wei Fan, Kun Yi, Hangting Ye, Zhiyuan Ning, Qi Zhang, and Ning An. Deep frequency derivative learning for non-stationary time series forecasting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024.

[24] Patrick Flandrin. *Time-frequency/time-scale analysis*, volume 10. Academic press, 1998.

[25] Haotian Gao, Renhe Jiang, Zheng Dong, Jinliang Deng, Yuxin Ma, and Xuan Song. Spatial-temporal-decoupled masked pre-training for spatiotemporal forecasting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024.

[26] Mircea Grigoriu. On the spectral representation method in simulation. *Probabilistic Engineering Mechanics*, 8(2):75–90, 1993.

[27] Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. CARD: Classification and regression diffusion models. *Advances in Neural Information Processing Systems*, 35:18100–18115, 2022.

[28] Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 8078–8086, 2023.

[29] Guangyin Jin, Yuxuan Liang, Yuchen Fang, Zezhi Shao, Jincai Huang, Junbo Zhang, and Yu Zheng. Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *IEEE transactions on knowledge and data engineering*, 36(10):5388–5408, 2023.

[30] Ming Jin, Yu Zheng, Yuan-Fang Li, Siheng Chen, Bin Yang, and Shirui Pan. Multivariate time series forecasting with dynamic graph neural ODEs. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9168–9180, 2022.

[31] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022.

[32] Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Bernie Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. *Advances in Neural Information Processing Systems*, 36, 2023.

[33] Qi Li, Zhenyu Zhang, Lei Yao, Zhaoxia Li, Tianyi Zhong, and Yong Zhang. Diffusion-based decoupled deterministic and uncertain framework for probabilistic multivariate time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025.

[34] Yuxin Li, Wenchao Chen, Xinyue Hu, Bo Chen, Baolin Sun, and Mingyuan Zhou. Transformer-modulated diffusion models for probabilistic multivariate time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

[35] Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. SegRNN: Segment recurrent neural network for long-term time-series forecasting. *IEEE Internet of Things Journal*, 13:9861–9871, 2026.

[36] Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer SOTA for traffic forecasting. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 4125–4129, 2023.

[37] Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. SCINet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022.

[38] Peiyuan Liu, Beiliang Wu, Yifan Hu, Naiqi Li, Tao Dai, Jigang Bao, and Shu-tao Xia. TimeBridge: Non-stationarity matters for long-term time series forecasting. In *International Conference on Machine Learning*, 2025.

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

[40] 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, 2023.

[41] 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*, 2024.

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

[43] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[44] Alex T. Mallen, Henning Lange, and J. Nathan Kutz. Deep probabilistic Koopman: Long-term time-series forecasting under periodic uncertainties. *International Journal of Forecasting*, 40: 859–868, 2024.

[45] James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

[46] Olof Mogren. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*, 2016.

[47] Ilan Naiman, N. Benjamin Erichson, Pu Ren, Michael W. Mahoney, and Omri Azencot. Generative modeling of regular and irregular time series data via Koopman VAEs. In *The Twelfth International Conference on Learning Representations*, 2024.

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

[49] Maurice B Priestley. Evolutionary spectra and non-stationary processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 27(2):204–229, 1965.

[50] Maurice Bertram Priestley. *Spectral Analysis and Time Series*. Academic Press, 1981.

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

[52] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pages 8857–8868. PMLR, 2021.

[53] Richard A Roberts and Clifford T Mullis. *Digital signal processing*. Addison-Wesley Longman Publishing Co., Inc., 1987.

[54] Attilio Sbrana, André Luis Debiaso Rossi, and Murilo Coelho Naldi. N-BEATS-RNN: deep learning for time series forecasting. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 765–768. IEEE, 2020.

[55] Dominik Schillinger and Vissarion Papadopoulos. Accurate estimation of evolutionary power spectra for strongly narrow-band random fields. *Computer Methods in Applied Mechanics and Engineering*, 199(17-20):947–960, 2010.

[56] Peter J Schmid. Dynamic mode decomposition and its variants. *Annual Review of Fluid Mechanics*, 54(1):225–254, 2022.

[57] Lifeng Shen and James Kwok. Non-autoregressive conditional diffusion models for time series prediction. In *International Conference on Machine Learning*, pages 31016–31029. PMLR, 2023.

[58] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion Schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36:62183–62223, 2023.

[59] Pol D Spanos and Giuseppe Failla. Evolutionary spectra estimation using wavelets. *Journal of Engineering Mechanics*, 130(8):952–960, 2004.

[60] Pol D Spanos, Jale Tezcan, and Petros Tratskas. Stochastic processes evolutionary spectrum estimation via harmonic wavelets. *Computer Methods in Applied Mechanics and Engineering*, 194(12-16):1367–1383, 2005.

[61] Chen Su, Zhengzhou Cai, Yuanhe Tian, Zhuochao Chang, Zihong Zheng, and Yan Song. Diffusion models for time series forecasting: A survey. *arXiv preprint arXiv:2507.14507*, 2025.

[62] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.

[63] Hristos Tyralis and Georgia Papacharalampous. A review of predictive uncertainty estimation with machine learning. *Artificial Intelligence Review*, 57(4):94, 2024.

[64] Rainer von Sachs and Kai Schneider. Wavelet smoothing of evolutionary spectra by nonlinear thresholding. *Applied and computational harmonic analysis*, 3(3):268–282, 1996.

[65] Rui Wang, Yihe Dong, Sercan Ö Arik, and Rose Yu. Koopman neural forecaster for time series with temporal distribution shifts. In *International Conference on Learning Representations*, 2023.

[66] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024.

[67] 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, 2024.

[68] Eugene Wigner. On the quantum correction for thermodynamic equilibrium. *Physical review*, 40(5):749, 1932.

[69] 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 (NeurIPS)*, 34:22419–22430, 2021.

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

[71] Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with $10k$ parameters. In *International Conference on Learning Representations*, 2024.

[72] Yiyuan Yang, Ming Jin, Haomin Wen, Chaoli Zhang, Yuxuan Liang, Lintao Ma, Yi Wang, Chenghao Liu, Bin Yang, Zenglin Xu, Shirui Pan, and Qingsong Wen. A survey on diffusion models for time series and spatio-temporal data. *ACM Computing Surveys*, 58(8):1–39, 2026.

[73] Weiwei Ye, Songgaojun Deng, Qiaosha Zou, and Ning Gui. Frequency adaptive normalization for non-stationary time series forecasting. *Advances in Neural Information Processing Systems*, 37, 2024.

[74] Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary diffusion for probabilistic time series forecasting. In *International Conference on Machine Learning*, 2025.

[75] 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, 2023.

[76] 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.

[77] Kun Yi, Qi Zhang, Wei Fan, Longbing Cao, Shoujin Wang, Hui He, Guodong Long, Liang Hu, Qingsong Wen, and Hui Xiong. A survey on deep learning based time series analysis with frequency transformation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6206–6215, 2025.

[78] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

[79] Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable diffusion for general time series generation. In *International Conference on Learning Representations*, 2024.

[80] Yitian Zhang, Liheng Ma, Antonios Valkanas, Boris N Oreshkin, and Mark Coates. SKOLR: Structured Koopman operator linear RNN for time-series forecasting. In *International Conference on Machine Learning*, 2025.

[81] Ronghua Zheng, Hanru Bai, and Weiyang Ding. KooNPro: A variance-aware Koopman probabilistic model enhanced by neural process for time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025.

[82] Haoyi Zhou, Shanghang Zhang, Jiehui Peng, Shuai Zhang, Jianxin Li, Hongyan Xiong, and Wengang Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, 2021.

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

A Full Related Works

A.1 Evolutionary Spectra for Non-stationary Time Series

Classical frequency analysis relies on the Fourier Transform (FT) [6], which assumes a time-invariant spectrum, making it unsuitable signals with evolving frequency content. Time–frequency methods such as STFT [3] and wavelet transforms [43] address this by localizing spectral estimation, but suffer from inherent trade-offs: STFT is limited by the Heisenberg uncertainty principle, while wavelets use fixed bases that lack adaptivity [24]. The Wigner–Ville distribution [68] provides high-resolution estimates, but can introduce severe cross-terms in multi-component signals. To formally account for non-stationarity, Priestley proposed evolutionary spectra theory [49, 50], which defines a point-wise spectrum for harmonizable stochastic processes. It is intractable to determine the ES since the evolutionary spectrum is fundamentally non-identifiable and admits infinitely many valid solutions. There exist some techniques to estimate the ES, though these depend on very limiting conditions such as local stationarity or narrow-bandness [55, 59, 60, 64]. Recent approaches rely on perturbation-based optimization methods [5], requiring a strict initialization. TimeES estimates evolutionary spectra directly from the data, without relying on strong parametric assumptions or constraints.

A.2 Deep Learning for Deterministic Time Series Forecasting

Time series is important in applications like traffic [17, 25, 29], and stock forecasting [12, 20]. Recent years have witnessed significant advances in deep learning for deterministic time series forecasting [13, 30, 72], spanning forecasting [82], classification [70], and anomaly detection [7], etc. Architectures such as RNNs [15, 35, 54], CNNs [37], and Transformers [69] have demonstrated strong empirical performance by capturing temporal dependencies through sequential or attention-based mechanisms [9, 28, 51, 41]. More recently, frequency-aware models like FEDformer [83] and FreTS [75] explicitly incorporate spectral components to improve long-term forecasting [76, 77]. To further address non-stationarity, recent works explicitly model distributional or frequency shifts [22, 31, 73, 38]. However, these approaches do not explicitly model spectral evolution, thus cannot effectively capture the frequency variations.

A.3 Deep Learning for Probabilistic Time Series Forecasting

Currently, most time series generation methods are based on deep generative models originally developed for computer vision tasks. Early approaches are predominantly based on Generative Adversarial Networks (GANs) [46, 78]. Nevertheless, the instability of adversarial training has motivated a shift toward other alternative deep generative paradigms. TimeVAE [16] incorporates an interpretable temporal prior into a Variational Autoencoder (VAE) framework, Fourier Flows [1] use a normalizing flow architecture and apply a cascade of spectral filters to enable exact likelihood optimization. Denoising Diffusion Probabilistic Models (DDPMs), have recently been adapted to time series due to their strong empirical performance in modeling complex distributions. CSDI [62] introduces a masking strategy that enables joint training for imputation and generation; DiffusionTS [79] combines the Fourier Transform with DDPMs to reconstruct and interpret time series in the spectral domain; and Alcaraz and Strodthoff [2] extends DDPMs with a structured spatial prior to capture long-term dependencies. Among generative tasks, probabilistic forecasting stands out as especially challenging due to the non-stationarity inherent in real-world time series. In this context, DDPMs have also shown great promise [63]: TimeGrad [52] integrates an autoregressive diffusion process guided by hidden states of RNN; TimeDiff [57] stabilizes generation through future mixup and autoregressive initialization; and Kollovieh et al. [32] propose a self-guiding strategy based on structured state-space models. Moreover, building on the Schrödinger bridge framework of Shi et al. [58], Chen et al. [11] provides theoretical convergence guaranties and algorithmic refinements for diffusion-based time series modeling. However, most of these methods assume stationary endpoint distributions, which are ill-suited for non-stationary temporal dynamics. To address this, NsDiff [74] incorporates a location-scale noise model into DDPM, allowing the endpoint distribution to vary over time and thereby better capture non-stationarity. Nevertheless, nearly all existing approaches are fundamentally built upon architectures originally designed for computer vision and therefore implicitly rely on the assumption of i.i.d. data [33]. In contrast, our proposed TimeES is grounded in the explicit assumption that the underlying data-generating process is non-stationary and stochastic.

B Proof of Theorems

B.1 Theorem 3.1

When analyzing the spectrum of a continuous-time stochastic process $\{X(t)\}_{t\in\mathbb{R}}$, the Fourier integral is not directly applicable. First, the Fourier series assumes that the signal is composed of a finite sum of periodic components, which is incompatible with general stochastic processes. Second, the Fourier integral requires that the sample paths be absolutely integrable over $(-\infty, \infty)$, i.e., $\int_{-\infty}^{\infty} |X(t)|\, dt < \infty$. This condition almost surely fails for stochastic processes, as their realizations typically do not decay at infinity and thus are not absolutely integrable.

To circumvent this, we consider a truncated version of the process as:

$$X_T(t) = \begin{cases} X(t), & -T \le t \le T, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Assuming that this function is continuous between $-T$ and T, we can express X_T in Fourier integral as:

$$X_T(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} G_T(\omega) e^{i\omega t}\, d\omega, \tag{9}$$

where

$$G_T(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-T}^{T} X(t) e^{-i\omega t}\, dt. \tag{10}$$

A natural idea is to take the limit $T \to \infty$ to recover a spectral representation of $X(t)$. However, since $T \to \infty$, $G_T(\omega)$ does not converge in the usual sense because $X(t)$ is not integrable. In fact, one can show that $|G_T(\omega)| = \mathcal{O}(\sqrt{T})$, implying divergence. Instead of focusing on energy (which is infinite), we consider *power*, defined via the normalized expected squared magnitude as:

$$h(\omega) := \lim_{T\to\infty} \mathbb{E}\left[\frac{|G_T(\omega)|^2}{2T}\right], \tag{11}$$

assuming that this limit exists. The function $h(\omega)$ is interpreted as the *spectral density* of the process.

Crucially, although $G_T(\omega)$ diverges as $T \to \infty$, its scaled increments remain well-behaved. Define the integrated Fourier transform as:

$$Z_T(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\omega} G_T(\theta)\, d\theta. \tag{12}$$

Then, over a small frequency interval $\delta\omega_n = \omega_{n+1} - \omega_n$, the increment satisfies

$$\Delta Z_T(\omega_n) = Z_T(\omega_{n+1}) - Z_T(\omega_n) \approx \frac{1}{\sqrt{2\pi}} G_T(\omega_n)\, \delta\omega_n. \tag{13}$$

Since $|G_T(\omega_n)| = \mathcal{O}(\sqrt{T})$ and $\delta\omega_n \sim 1/T$, it follows that $\Delta Z_T(\omega_n) = \mathcal{O}(1)$, i.e., the increments remain bounded as $T \to \infty$.

This observation motivates replacing the ill-defined Fourier integral with a *stochastic integral* with respect to a random measure. In the limit $T \to \infty$, the process $Z_T(\omega)$ converges (in an appropriate probabilistic sense) to a random function $Z(\omega)$ with orthogonal increments satisfying $\mathbb{E}[|dZ(\omega)|^2] = h(\omega)\, d\omega$.

Consequently, the original process admits the spectral representation:

$$X(t) = \int_{-\infty}^{\infty} e^{it\omega}\, dZ(\omega), \tag{14}$$

where $Z(\omega)$ is a complex-valued stochastic process with uncorrelated (or orthogonal) increments. This completes the justification of the spectral representation theorem.

B.2 Theorem 4.1

First, for a discrete signal $\{X_t\}$, a generalized Cramér-type representation [14] can be written as a time-varying Fourier–Stieltjes integral [49] as:

$$X_n = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} e^{i\omega n} dZ_n(\omega) \tag{15}$$

where $dZ_n(\omega)$ is a complex-valued random measure with a covariance structure governed by the evolutionary spectral density. Under the evolutionary spectra framework defined in Theorem 3.1 and 3.2, $dZ_n(w)$ admits:

$$dZ_n(\omega) = A(n, \omega) dW(\omega) \tag{16}$$

where $A(n, \omega)$ is a complex amplitude modulation function, $dW(\omega)$ is a complex Gaussian white noise measure on $[-\pi, \pi)$ satisfying:

$$\mathbb{E}[dW(\omega)] = 0, \quad \mathbb{E}[dW(\omega)\overline{dW(\nu)}] = \delta(\omega - \nu) d\omega \tag{17}$$

where δ is the Dirac delta function that is zero everywhere except in $\nu = \nu_0$. Since numerical simulation cannot handle continuous integrals, we discretize the frequency axis into M equally spaced points over $[0, 2\pi)$:

$$\omega_k = \frac{2\pi k}{M}, \quad k = 0, 1, \ldots, M-1.$$

To align with the standard Fourier convention on $[-\pi, \pi)$, we may map high frequencies:

$$\omega_k \leftarrow \begin{cases} \omega_k & \text{if } \omega_k \leq \pi \\ \omega_k - 2\pi & \text{if } \omega_k > \pi \end{cases} \tag{18}$$

and define the frequency spacing as:

$$\Delta\omega = \frac{2\pi}{M}. \tag{19}$$

We approximate the stochastic integral Equation (1) by a Riemann sum:

$$X_n \approx \frac{1}{\sqrt{2\pi}} \sum_{k=0}^{M-1} A\left(n, \omega_k\right) e^{i\omega_k n} \cdot \Delta W_k, \tag{20}$$

where ΔW_k approximates the increment $dW(\omega)$ over the interval $[\omega_k, \omega_k + \Delta\omega)$.

For a complex Gaussian white noise measure, the increments satisfy:

$$\mathbb{E}\left[\Delta W_k\right] = 0, \quad \mathbb{E}\left[\Delta W_k \overline{\Delta W_\ell}\right] = \delta_{k\ell} \Delta\omega. \tag{21}$$

Thus, we can write:

$$\Delta W_k = \sqrt{\Delta\omega} \cdot W_k, \tag{22}$$

where $W_k \sim$ are i.i.d. standard complex normal random variables.

Substituting into Equation (2):

$$X_n \approx \frac{1}{\sqrt{2\pi}} \sum_{k=0}^{M-1} A\left(n, \omega_k\right) e^{i\omega_k n} \cdot \sqrt{\Delta\omega} \cdot W_k. \tag{23}$$

Recall $\Delta\omega = 2\pi/M$, so $\sqrt{\Delta\omega} = \sqrt{2\pi/M}$. Therefore:

$$X_n \approx \frac{1}{\sqrt{2\pi}} \cdot \sqrt{\frac{2\pi}{M}} \sum_{k=0}^{M-1} A\left(n, \omega_k\right) W_k e^{i\omega_k n} = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} A\left(n, \omega_k\right) W_k e^{i\omega_k n}. \tag{24}$$

Since $\omega_k = 2\pi k/M$, we have $e^{i\omega_k n} = e^{i2\pi kn/M}$, yielding the final discrete synthesis formula:

$$X_n \approx \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} A\left(n, \omega_k\right) W_k e^{i2\pi kn/M}, \tag{25}$$

which completes the proof.

B.3 Theorem 4.3

Proof. We prove that $X_n \in \mathbb{R}$ for all n if and only if the Hermitian symmetry condition (5) holds.

($\Rightarrow$) Suppose $X_n \in \mathbb{R}$ for all n. Then $X_n = \overline{X_n}$. Define $B(n,k) = A(n,\omega_k)W_k$. Then Equation (2) becomes

$$\overline{X_n} = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} \overline{B(n,\omega_k)}\, e^{-i\omega_k n}.$$

Since $\omega_k = 2\pi k/M$, we have $e^{-i\omega_k n} = e^{i(\omega_{(M-k) \bmod M})n}$. Reindexing the sum with $k' = (M - k) \bmod M$ (which is a permutation of $\{0, \ldots, M-1\}$), we obtain

$$\overline{X_n} = \frac{1}{\sqrt{M}} \sum_{k'=0}^{M-1} \overline{B(n,\omega_{(M-k') \bmod M})}\, e^{i\omega_{k'} n}.$$

Because $X_n = \overline{X_n}$ and the complex exponentials $\{e^{i\omega_k n}\}_{k=0}^{M-1}$ form an orthogonal basis over $n = 0, \ldots, M-1$, the coefficients in the two expansions must be equal. Hence,

$$B(n,\omega_k) = \overline{B(n,\omega_{(M-k) \bmod M})}, \quad \forall k,$$

which is precisely the Hermitian symmetry condition (5).

($\Leftarrow$) Conversely, assume (5) holds. Then

$$\begin{aligned}
\overline{X_n} &= \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} \overline{B(n,\omega_k)}\, e^{-i\omega_k n} \\
&= \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} B(n,\omega_{(M-k) \bmod M})\, e^{i(\omega_{(M-k) \bmod M})n} \quad \text{(by (5))} \\
&= \frac{1}{\sqrt{M}} \sum_{k'=0}^{M-1} B(n,\omega_{k'})\, e^{i\omega_{k'} n} \quad \text{(reindex with } k' = (M-k) \bmod M\text{)} \\
&= X_n.
\end{aligned}$$

Thus $X_n = \overline{X_n}$, so $X_n \in \mathbb{R}$ for all n.

Finally, for $k = 0$, condition (5) gives $B(n,\omega_0) = \overline{B(n,\omega_0)}$, so $B(n,\omega_0) \in \mathbb{R}$. If M is even, then $k = M/2$ satisfies $(M-k) \bmod M = k$, so the same argument implies $B(n,\omega_{M/2}) \in \mathbb{R}$. □

B.4 Continuous STFT Approximates ES under Local Stationarity

Consider a real- or complex-valued signal $x(u)$ that admits a generalized time–frequency representation of the form

$$x(u) = \int_{-\infty}^{\infty} A(u,\nu)\, e^{i\nu u}\, \frac{d\nu}{\sqrt{2\pi}}, \tag{26}$$

where $A(u,\nu)$ is a smooth complex-valued function interpreted as the *local spectral amplitude*. Such a representation is commonly used in the analysis of non-stationary signals under the assumption of *local stationarity*.

Let $w \in \{L^1(\mathbb{R}) \cap L^2(\mathbb{R})\}$ be a real-valued analysis window with effective support concentrated near the origin, and assume that it is normalized so that $\int_{-\infty}^{\infty} w(s)\, ds = 1$. The short-time Fourier transform (STFT) of x with respect to w is defined by

$$X_{\mathrm{STFT}}(t,\omega) = \int_{-\infty}^{\infty} x(u)\, w(u-t)\, e^{-i\omega u}\, du. \tag{27}$$

Substituting the representation (26) into (27) and interchanging the order of integration (justified, for instance, if $A \in L^2(\mathbb{R}^2)$ and w decays sufficiently fast), we obtain

$$\begin{aligned} X_{\mathrm{STFT}}(t,\omega) &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} A(u,\nu)\, e^{i\nu u}\, \frac{d\nu}{\sqrt{2\pi}} \right] w(u-t)\, e^{-i\omega u}\, du \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(u,\nu)\, w(u-t)\, e^{i(\nu-\omega)u}\, du\, \frac{d\nu}{\sqrt{2\pi}}. \end{aligned} \tag{28}$$

Introduce the change of variables $s = u - t$, i.e., $u = t + s$, so that $du = ds$. Then (28) becomes

$$\begin{aligned} X_{\mathrm{STFT}}(t,\omega) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(t+s,\nu)\, w(s)\, e^{i(\nu-\omega)(t+s)}\, ds\, \frac{d\nu}{\sqrt{2\pi}} \\ &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} w(s)\, e^{i(\nu-\omega)s}\, ds \right] A(t+s,\nu)\, e^{i(\nu-\omega)t}\, \frac{d\nu}{\sqrt{2\pi}}. \end{aligned} \tag{29}$$

We now invoke the local stationarity assumption: over the effective support of the window $w(s)$ (i.e., where $|s|$ is small compared to the time scale of variation of A), the function $A(t+s,\nu)$ varies slowly with s. Formally, we assume $A(t+s,\nu) \approx A(t,\nu)$ for all s such that $w(s)$ is non-negligible. Under this approximation, $A(t,\nu)$ can be treated as constant with respect to s and factored out of the inner integral:

$$\begin{aligned} X_{\mathrm{STFT}}(t,\omega) &\approx \int_{-\infty}^{\infty} A(t,\nu)\, e^{i(\nu-\omega)t} \left[\int_{-\infty}^{\infty} w(s)\, e^{i(\nu-\omega)s}\, ds \right] \frac{d\nu}{\sqrt{2\pi}} \\ &= \int_{-\infty}^{\infty} A(t,\nu)\, e^{i(\nu-\omega)t}\, \widehat{w}(\omega-\nu)\, \frac{d\nu}{\sqrt{2\pi}}, \end{aligned} \tag{30}$$

where $\widehat{w}(\xi) = \int_{-\infty}^{\infty} w(s) e^{-i\xi s}\, ds$ denotes the Fourier transform of the window. Equation (30) shows that, under local stationarity, the STFT is approximately a *frequency-smoothed* version of $A(t,\nu)$, modulated by a linear phase term. The smoothing kernel is precisely the spectral profile $\widehat{w}$ of the analysis window, which induces unavoidable *time–frequency smearing* (or leakage). If, in addition, $A(t,\nu)$ is approximately constant over the bandwidth of $\widehat{w}$, that is, over frequencies $|\nu - \omega| \lesssim B_w$, where B_w is the effective bandwidth of the window, then the convolution in Equation (30) is dominated by the value at $\nu = \omega$. In this case,

$$X_{\mathrm{STFT}}(t,\omega) \approx A(t,\omega)\, \widehat{w}(0)\, e^{i(\omega-\omega)t} = A(t,\omega),$$

since $\widehat{w}(0) = \int w(s)\, ds = 1$ by normalization. Thus, up to negligible smearing terms due to the finite support of w, the STFT recovers the local spectral amplitude, which completes the proof.

The approximation improves as the window narrows in time (better temporal localization) *and* $A(u,\nu)$ varies more slowly in u. However, due to the uncertainty principle, narrowing the window in time broadens $\widehat{w}$ in frequency, increasing spectral smearing. Hence, the accuracy of the approximation depends on a trade-off governed by the signal's intrinsic time–frequency structure and the choice of window.

B.5 Relationships between SNR, r and K

Suppose a additive noise model for the observed time series: $x_n = s_n + \varepsilon_n$, $n = 0, 1, \ldots, N-1$. where s_n is the latent clean signal, ε_n is zero-mean additive noise, assumed to be uncorrelated with s_n, and drawn from a standard normal distribution. Let $\hat{X}_k$, $\hat{S}_k$, and $\hat{E}_k$ denote the discrete Fourier transforms of x_n, s_n, and ε_n, respectively. Then:

$$\hat{X}_k = \hat{S}_k + \hat{E}_k.$$

The total signal energy (by Parseval's theorem [8]) is:

$$E_s = \|s\|_2^2 = \sum_{n=0}^{N-1} s_n^2 = \sum_{k=0}^{N-1} \frac{1}{N} |\hat{S}_k|^2.$$

The expected total noise energy is:

$$E_n = \mathbb{E}[\|\varepsilon\|_2^2] = N\sigma^2.$$

Thus, the signal-to-noise ratio is defined as:

$$\text{SNR} := \frac{E_s}{E_n} = \frac{\|s\|_2^2}{N\sigma^2}.$$

The periodogram at frequency bin k is:

$$I(k) = \frac{1}{N}|\hat{X}_k|^2 = \frac{1}{N}|\hat{S}_k + \hat{E}_k|^2.$$

Taking expectation, we obtain:

$$\mathbb{E}[I(k)] = \frac{1}{N}|\hat{S}_k|^2 + \frac{1}{N}\mathbb{E}[|\hat{E}_k|^2],$$

where the first term is the signal energy and the second is the noise energy, defined as E_s and E_n, respectively. Now consider selecting the smallest set of frequencies such that the cumulative observed energy reaches a fraction $r \in (0, 1)$ of the total:

$$\sum_{k\in\mathcal{K}} I(k) \geq r \sum_{k=0}^{N-1} I(k)$$

In expectation, and suppose that the selected frequencies exactly capture a fraction r of the total energy, this becomes:

$$\mathbb{E}\left[\sum_{k\in\mathcal{K}} I(k)\right] \approx r\,(E_s + E_n)$$

Then:

$$\mathbb{E}\left[\sum_{k\in\mathcal{K}} I(k)\right] \approx E_s + K\sigma^2,$$

since the signal energy E_s is concentrated in K dominant frequency bins, and each selected bin contributes σ^2 of noise energy in expectation (due to the flat spectrum of white noise). The total expected observed energy is $E_s + M\sigma^2 = E_s + E_n$, where $E_n = M\sigma^2$ is the total noise energy. Enforcing the energy retention criterion yields:

$$E_s + K\sigma^2 \approx r(E_s + M\sigma^2).$$

Solving for r, we obtain:

$$r \approx \frac{E_s + K\sigma^2}{E_s + M\sigma^2}.$$

we substitute $E_s = \text{SNR} \cdot M\sigma^2$ to get:

$$r \approx \frac{\text{SNR} \cdot M\sigma^2 + K\sigma^2}{\text{SNR} \cdot M\sigma^2 + M\sigma^2} = \frac{N \cdot \text{SNR} + K}{M(\text{SNR} + 1)}.$$

Equivalently, solving for SNR gives:

$$\text{SNR} \approx \frac{Mr - K}{M(1 - r)}.$$

This relationship reveals why fixing r leads to adaptive behavior, when the signal has a high SNR, only a few frequencies are needed; when noise dominates, requiring a larger set of frequencies to capture sufficient signal content. Hence, this eliminates the need for manual tuning like previous methods, requiring only a lower-bound of r to balance computational overhead and performance. We also provide analysis in Appendix C.3 to justify the selection of r. Thus, the energy-thresholding rule automatically adjusts the retained frequencies K to the data's SNR, without requiring prior knowledge about the frequency structure or the noise level.

C Experimental Details

Table 7 shows a summary of the benchmarks. As a supplementary evaluation, we perform a classification task using the learned representations to validate their effectiveness in capturing task-relevant features.

Table 7: Summary of experiment benchmarks.

Tasks	Benchmarks	Metrics	Input	Output
Long-term Forecasting (Single variate and Multivariate)	ETT{m1, m2, h1, h2}, ECL, EXG, Traffic, Weather, Solar	MSE, MAE	96	{96, 192, 336, 720}
Probabilistic Forecasting	ETT{m1, m2, h1, h2}, ECL, EXG, Traffic, ILI, Solar	CRPS, QICE, MSE, MAE	168	36 (ILI), 192 (others)
Classification	UEA (5 subsets)	Accuracy	315-2500	-

C.1 Datasets Details

ETT[1] (Electricity Transformer Temperature): The electricity transformer datasets contain power load information and oil temperature as the target variable. {ETTh1, ETTh2} are 1-hour-level datasets, and {ETTm1, ETTm2} are 15-minute-level datasets. Each data point consists of seven features: one "oil temperature" (OT) target and six power load features.

ECL[2] (Electricity Consuming Load): The electricity consumption dataset contains hourly electricity consumption (in Kwh) of 321 clients in two years.

EXG (Exchange-Rate): The Exchange-Rate dataset contains daily exchange rates of eight foreign countries including Australia, British, Canada, Switzerland, China, Japan, New Zealand and Singapore ranging from 1990 to 2016.

Traffic[3] (Traffic): The data contains 48 months (2015-2016) of hourly road occupancy rates (between 0 and 1) measured by 862 sensors on freeways in the San Francisco Bay area.

Weather[4] (Weather): The weather dataset contains 21 different meteorological features of the U.S. in 4 years, including visibility, wind speed, etc.

Solar[5] (Solar Energy): The solar power production dataset records the hourly energy output from 137 photovoltaic (PV) plants in Alabama over the year 2006.

ILI[6] (Influenza-like Illness): The ILI dataset contains weekly percentages of patients reported with influenza-like illness with multiple flu seasons containing 7 features.

UEA Classification[7]: The UEA & UCR Time Series Classification Archive [4] is a widely used benchmark suite for evaluating time series classification algorithms. It was introduced to unify and standardize evaluation across the field, combining datasets from the former UCR (University of California, Riverside) and UEA (University of East Anglia) archives.

C.2 Metrics

We adopt the following standard metrics to evaluate model performance:

Mean Absolute Error (MAE) measures the average magnitude of absolute prediction errors:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|,$$

[1] `https://github.com/zhouhaoyi/ETDataset`.
[2] `https://github.com/laiguokun/multivariate-time-series-data`.
[3] `http://pems.dot.ca.gov`.
[4] `https://www.ncei.noaa.gov/data/local-climatological-data`.
[5] `https://www.nrel.gov/grid/solar-power-data.html`.
[6] `https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html`.
[7] `https://timeseriesclassification.com`.

where y_i and $\hat{y}_i$ denote the ground-truth and predicted values for the i-th sample, respectively, and N is the total number of samples.

Mean Squared Error (MSE) quantifies the average squared deviation between predictions and ground truth, penalizing larger errors more heavily:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2.$$

Accuracy is used for classification tasks and represents the proportion of correctly classified samples:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i = y_i),$$

where $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if the prediction matches the true label and 0 otherwise.

CRPS: The continuous ranked probability score (CRPS) [45] measures the compatibility of a cumulative distribution function (CDF) F with an observation x as:

$$\text{CRPS}(F, x) = \int_{\mathbb{R}} (F(z) - \mathbb{I}\{x \le z\})^2 \, \mathrm{d}z, \tag{31}$$

where $\mathbb{I}_{z<q}$ denotes the indicator function. By approximating the predictive cumulative distribution function (CDF) F with its empirical counterpart, $\hat{F}(z) = \frac{1}{S}\sum_{s=1}^{S}\mathbb{I}\left\{x^{0,s} \le z\right\}$, constructed from S samples $x^{0,s} \sim F$, the CRPS can be directly estimated from the generated trajectories. In our experiments, we use samples $S = 100$ to approximate the distribution F.

QICE: The Quantile Interval Calibration Error (QICE) [27] measures the discrepancy between the empirical coverage of predicted quantile intervals (QIs) and their nominal target coverage. Specifically, given I non-overlapping quantile intervals that partition the predictive distribution into equal-probability regions (each with nominal coverage $1/I$), QICE computes the average absolute deviation between the observed proportion of ground-truth values falling within each interval and the ideal proportion $1/I$.

Formally, let $\hat{y}_n^{\text{low}_m}$ and $\hat{y}_n^{\text{high}_m}$ denote the lower and upper bounds of the m-th quantile interval for the n-th sample. The empirical coverage of the m-th interval is:

$$r_m = \frac{1}{N}\sum_{n=1}^{N}\mathbb{I}\left\{\hat{y}_n^{\text{low}_m} \le y_n \le \hat{y}_n^{\text{high}_m}\right\},$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. QICE is then defined as:

$$\text{QICE} := \frac{1}{I}\sum_{m=1}^{I}\left|r_m - \frac{1}{I}\right|. \tag{32}$$

Under perfect calibration (i.e., when the predicted distribution matches the true data distribution), we expect $r_m \approx 1/I$ for all m, yielding QICE $\to 0$. Following Li et al. [34], we partition the probability range into $I = 10$ equal-sized decile-based intervals to compute QICE.

C.3 Spectral Energy Sparsity

A key observation from the spectral analysis of real-world time series is the phenomenon of spectral energy sparsity, which refers to the fact that most of the signal energy is concentrated in a small subset of frequency components, while the majority of frequencies contribute negligible power. This property is empirically validated across multiple benchmark datasets, as illustrated in Figure 5. As shown in Figure 5, for all tested datasets, including electricity demand (ETTm1, ETTm2), traffic flow, weather variables, exchange rates, and solar energy, the dominant spectral energy is highly localized in a few low-frequency bins. Specifically, we highlight the top frequencies accounting for 90% of total spectral energy using pink bars, revealing that this critical energy concentration often resides within only a handful of frequency indices. For instance, in ETTm1, 90% of the energy

is captured by just six frequencies; similarly, ETTm2 and ETTh1 require only two-five frequency components to retain 90% of signal energy. Even more strikingly, in the ExchangeRate dataset, over 80% of energy lies in a single dominant frequency, indicating strong periodicity. Across all datasets, the number of required frequencies grows slowly with increasing energy ratio,indicating that even at lower thresholds (e.g., 95%), only a small fraction of frequencies are responsible for most of the signal dynamics. Notably, even when capturing 99% of energy, the number of involved frequencies rarely exceeds 10–20 out of hundreds, demonstrating a high degree of frequency-wise energy concentration. The analysis together verify our choice of a fixed energy ratio of $r = 0.9$ is effective across the experiments.

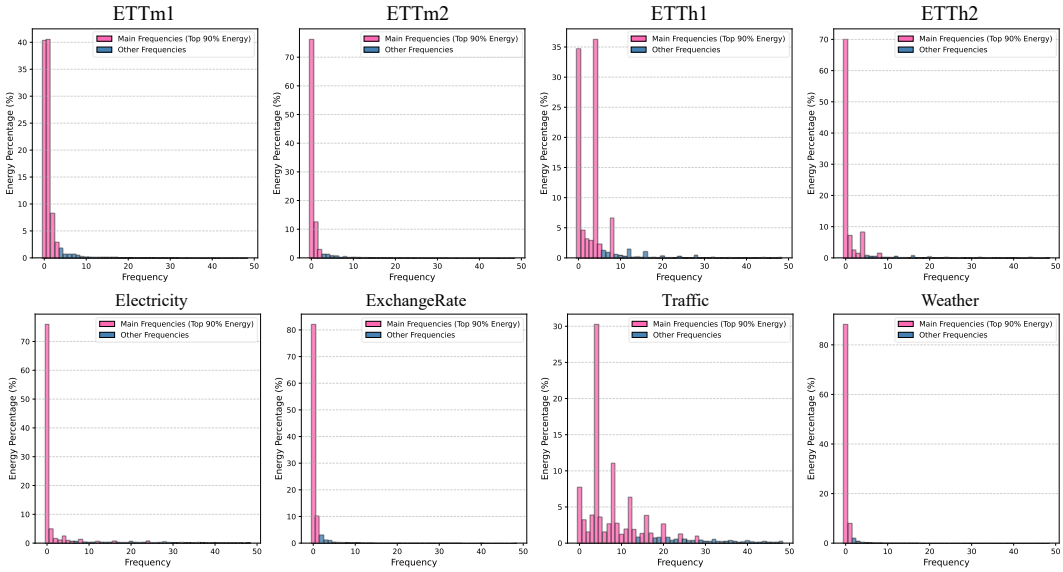


Figure 5: Energy distribution of different datasets. We use pink to highlight frequencies of top 90% energy.

C.4 Ablation Experiments

To investigate the individual contributions of key components in TimeES, we conduct ablation studies on the ETTs datasets, focusing on two core modules: (1) the Evolutionary Spectra (E) module, which models the time-varying spectral structure; and (2) the DES module, responsible for generative modeling and uncertainty quantification. As shown in Table 8, removing the evolutionary spectra modeling (w/o E) leads to a significant performance degradation across all tasks. For instance, in single-variate long-term forecasting on ETTm1, MSE increases from 0.893 to 1.306, indicating that the model fails to capture the non-stationary dynamics when spectral evolution is ignored. Similar drops are observed in multi-variate settings and probabilistic forecasting, where CRPS rises from 0.310 to 0.417 without E. This confirms that explicitly modeling the time-dependent frequency content is crucial for accurate long-term prediction under changing periodicities. In contrast, removing the DES module (w/o DES) results in only minor performance changes in deterministic forecasting tasks, e.g., MSE increases by less than 5% in most cases, suggesting that the core spectral representation remains effective even without the generative component. However, DES is essential for probabilistic forecasting: as shown in the bottom panel of Table 8, the model becomes non-generative without DES, rendering it incapable of producing predictive distributions or uncertainty estimates. Moreover, DES provides an interpretable frequency evolution matrix that enables fine-grained analysis of how dominant frequencies shift over time, a capability absent in standard Fourier-based methods. This not only enhances model transparency but also supports downstream applications such as anomaly detection and signal diagnosis.

As shown in Table 8, modeling the evolutionary spectra yields performance gains even without the DES. Notably, DES is a novel generative module; without it, the model cannot perform generative tasks such as probabilistic forecasting. Furthermore, DES provides an interpretable frequency evolu-

tion matrix that enables effective spectral analysis. These results collectively verify the importance of explicitly modeling the spectra at a finer-grained level.

Table 8: Ablation results on ETTs datasets. We report MSE, MAE for 720-steps single- and multi-variate long-term forecasting and report CRPS, MSE for 192 steps multivariate probabilistic forecasting. The best result is denoted in **bold** characters.

Single-Variate Long-term Forecasting								
Datasets	ETTm1		ETTm2		ETTh1		ETTh2	
Variants	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TimeES	**0.893**	**0.619**	**0.590**	**0.553**	**0.929**	**0.670**	**0.618**	**0.585**
w/o E	1.306	0.792	0.673	0.587	1.282	0.840	0.785	0.676
w/o DES	0.966	0.662	0.639	0.569	0.974	0.699	0.744	0.646
Multi-Variate Long-term Forecasting								
Datasets	ETTm1		ETTm2		ETTh1		ETTh2	
Variants	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TimeES	**0.465**	**0.445**	**0.396**	**0.395**	**0.485**	**0.479**	**0.426**	**0.443**
w/o E	0.601	0.510	0.414	0.407	0.604	0.538	0.439	0.452
w/o DES	0.467	0.447	0.411	0.410	0.554	0.508	0.432	0.449
Multi-Variate Probabilistic Forecasting								
Datasets	ETTm1		ETTm2		ETTh1		ETTh2	
Variants	CRPS	MSE	CRPS	MSE	CRPS	MSE	CRPS	MSE
TimeES	**0.310**	**0.371**	**0.230**	**0.235**	**0.349**	**0.474**	**0.320**	**0.376**
w/o E	0.417	0.378	0.289	0.250	0.466	0.592	0.363	0.387
w/o DES	The model is not generative without DES.							

C.5 Additional Baseline Experiments

To provide a comprehensive evaluation of our proposed method, we conduct extensive experiments against six state-of-the-art baseline models spanning different architectural paradigms in time series forecasting. These include: KNF [65], a Koopman neural operator-based approach; Koopa [40], which leverages Koopman spectrum analysis; SKOLR [80], a structured Koopman learning framework; TimeBridge [38], a bridge-based temporal modeling architecture; TimeMixer [66], a channel-independent mixing strategy; and TimeXer [67], a Transformer-based temporal encoder. This diverse set of baselines enables rigorous comparison across spectral methods, neural operators, and attention-based architectures, ensuring a thorough assessment of our method's effectiveness in multivariate time series forecasting tasks.

C.6 Synthetic Experiments

To evaluate the ability of TimeES to capture *spectral evolution*, a key challenge in non-stationary time series, we conduct a synthetic experiment on a chirp signal, which exhibits continuously varying frequency and amplitude over time. The signal is defined as:

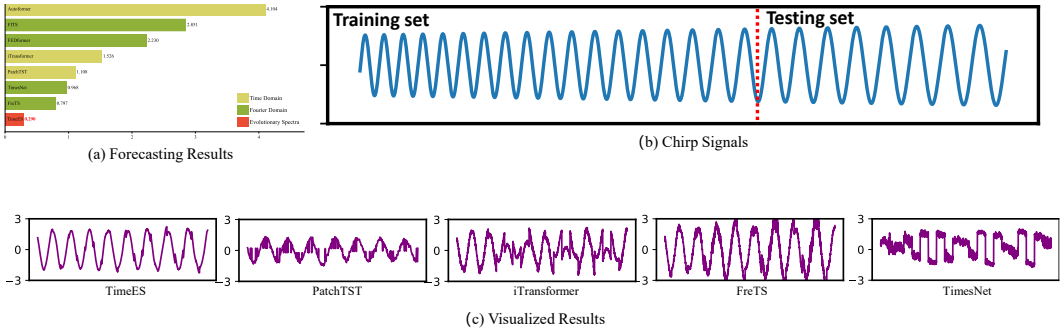
$$x(t) = A(t) \cdot \sin(2\pi f(t)t),$$

where $A(t)$ and $f(t)$ are slowly increasing functions, simulating real-world phenomena such as radar signals or mechanical vibrations with changing dynamics, we let the periodicity change from 64 to 128, the magnitude change from 1 to 3, and other experiment settings are consistent with the deterministic forecasting experiment 96 step forecasting. As shown in Figure 6(b), the training set contains only the initial portion of the chirp signal, where the frequency increases gradually. The testing set extends into a region with higher frequencies not seen during training, thus forming a challenging extrapolation task that requires modeling the *evolution* of spectral components rather than mere interpolation. Figure 6(c) visualizes the forecasting results across different models. While most methods (e.g., PatchTST, iTransformer, TimesNet) fail to track the increasing frequency and produce oscillations with fixed periodicity, FreTS shows partial success due to its frequency-aware design

Table 9: Mean Squared Error (MSE) results on deterministic multivariate forecasting tasks across different datasets and prediction horizons. **Bold** values indicate the best performance for each setting. The last row shows the number of times each model achieves the best performance.

Dataset	Horizon	KNF	Koopa	SKOLR	TimeBridge	TimeMixer	TimeXer	TimeES
ETTm1	96	0.691	0.331	0.342	**0.324**	0.344	0.387	0.328
	192	0.707	0.377	0.397	0.365	0.368	0.441	**0.364**
	336	0.718	0.402	0.445	0.395	**0.394**	0.514	0.402
	720	0.742	0.466	0.536	0.457	**0.455**	0.592	0.465
ETTm2	96	0.230	0.182	0.194	0.178	0.178	0.364	**0.176**
	192	0.274	0.245	0.256	0.245	0.246	0.526	**0.240**
	336	0.385	0.301	0.321	0.307	0.305	0.870	**0.300**
	720	0.479	0.402	0.423	0.411	0.402	0.902	**0.396**
ETTh1	96	0.700	0.394	0.427	0.378	0.376	0.438	**0.374**
	192	0.717	0.442	0.470	0.431	**0.415**	0.497	0.430
	336	0.723	0.489	0.526	0.478	**0.446**	0.528	0.470
	720	0.716	0.482	0.531	0.533	**0.472**	0.639	0.485
ETTh2	96	0.362	0.305	0.330	0.296	**0.287**	0.583	0.293
	192	0.430	0.392	0.419	0.376	0.372	0.590	**0.371**
	336	0.478	0.426	0.443	0.417	0.429	0.612	**0.427**
	720	0.506	0.447	0.463	0.427	0.447	0.652	**0.426**
1st Count		0	0	0	1	5	0	9

but still suffers from phase drift. In contrast, TimeES (NES) accurately captures both the evolving frequency and amplitude, producing a smooth and faithful reconstruction of the true signal. The quantitative comparison in Figure 6(a) further confirms this advantage: TimeES achieves the lowest MSE (0.290), significantly outperforming all baselines. Notably, even Fourier-domain methods like FreTS, despite their explicit frequency modeling, struggle to maintain accuracy under spectral evolution, highlighting the limitation of static spectral bases. This synthetic experiment demonstrates that TimeES's learned evolutionary spectra can effectively model non-stationary dynamics by adapting to changing frequency content, providing strong empirical support for its capability in capturing long-term spectral trends.



(a) Forecasting Results

(b) Chirp Signals

(c) Visualized Results

Figure 6: (a-f) Predictive uncertainty quantification by TimeES across different datasets. We plot the last 500 predicted time steps of the first four variables in each dataset. The orange line shows the ground truth, the blue line represents the predicted mean, and the gray shaded region indicates the 96% prediction interval.

C.7 Time Series Classification

We evaluate the discriminative power of the learned spectral representation $\mathbf{A}$ on five multivariate time series classification tasks from the UEA archive [4], as shown in Table 10 and summarized in Figure 7. Our method, TimeES, achieves an average accuracy of 62.96%, outperforming the previous best model, TimesNet, by 2.85%, demonstrating the effectiveness of modeling non-stationary spectral evolution for general-purpose time series representation. Notably, TimeES achieves state-of-the-art performance across multiple datasets: it reaches 92.83% on `SelfRegulationSCP1`, a dataset with complex brainwave dynamics, and 36.81% on EthanolConcentration, where fine-grained frequency

patterns are critical. On UWaveGestureLibrary, our method performs competitively (86.50%) despite being slightly behind iTransformer (87.50%), suggesting that while temporal context is important, evolutionary spectral modeling captures complementary information. While other frequency-domain models like FreTS (59.04%) and FEDformer (56.17%) show improved performance over pure time-domain methods, they still fall short of TimeES. This suggests that simply applying frequency analysis is insufficient; the key lies in modeling how spectral support evolves over time, especially under non-stationary conditions such as changing periodicity. The performance of TimeES confirms that learning an adaptive and temporally coherent spectral representation not only benefits long-term forecasting but also enables more robust and transferable features for downstream tasks.

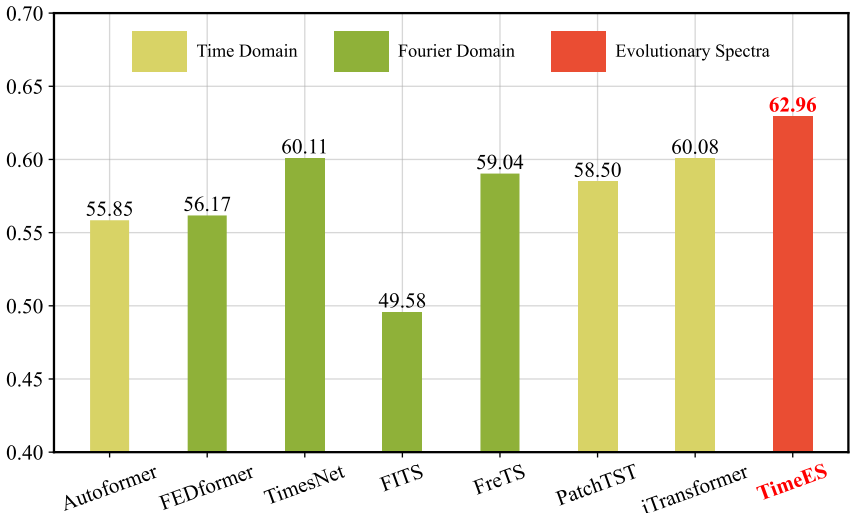


Figure 7: Average accuracy across datasets.

Table 10: Classification accuracy on five datasets. **Bold** and underline indicate best and second-best results respectively.

	Autoformer	FEDformer	TimesNet	FITS	FreTS	PatchTST	iTransformer	TimeES
EthanolConcentration	31.60	31.20	35.70	25.10	24.79	36.42	31.60	**36.81**
MotorImagery	48.40	45.00	50.40	49.60	50.40	52.80	**55.60**	52.00
SelfRegulationSCP1	84.00	88.70	91.80	72.70	89.56	81.78	88.39	**92.83**
StandWalkJump	29.33	30.67	37.33	38.67	45.33	41.33	37.33	**46.67**
UWaveGestureLibrary	85.90	85.30	85.30	61.81	85.10	80.19	**87.50**	86.50
Avg.	55.85	56.17	60.11	49.58	59.04	58.50	60.08	**62.96**

D Efficiency Analysis

To evaluate the computational and memory efficiency of TimeES, we conduct an efficiency analysis on the ETTh1 dataset probabilistic forecasting and 720-step long-term multivariate forecasting. As shown in Figure 8, TimeES achieves the highest efficiency among all compared methods in both generative and deterministic forecasting tasks. Notably, our probabilistic forecasting significantly outperforms prior diffusion-based approaches, which typically require hundreds or thousands of iterative denoising steps to generate predictions. In contrast, TimeES produces accurate samples in a single forward pass. Moreover, unlike frequency-domain models such as FITS that rely on explicit operations (e.g., DFT and iDFT) to compute spectral representations, TimeES directly learns time-series dynamics through a compact neural network without any explicit spectral transformation, which enables TimeES to surpass even highly optimized frequency-based models in deterministic forecasting speed, demonstrating its high efficiency.

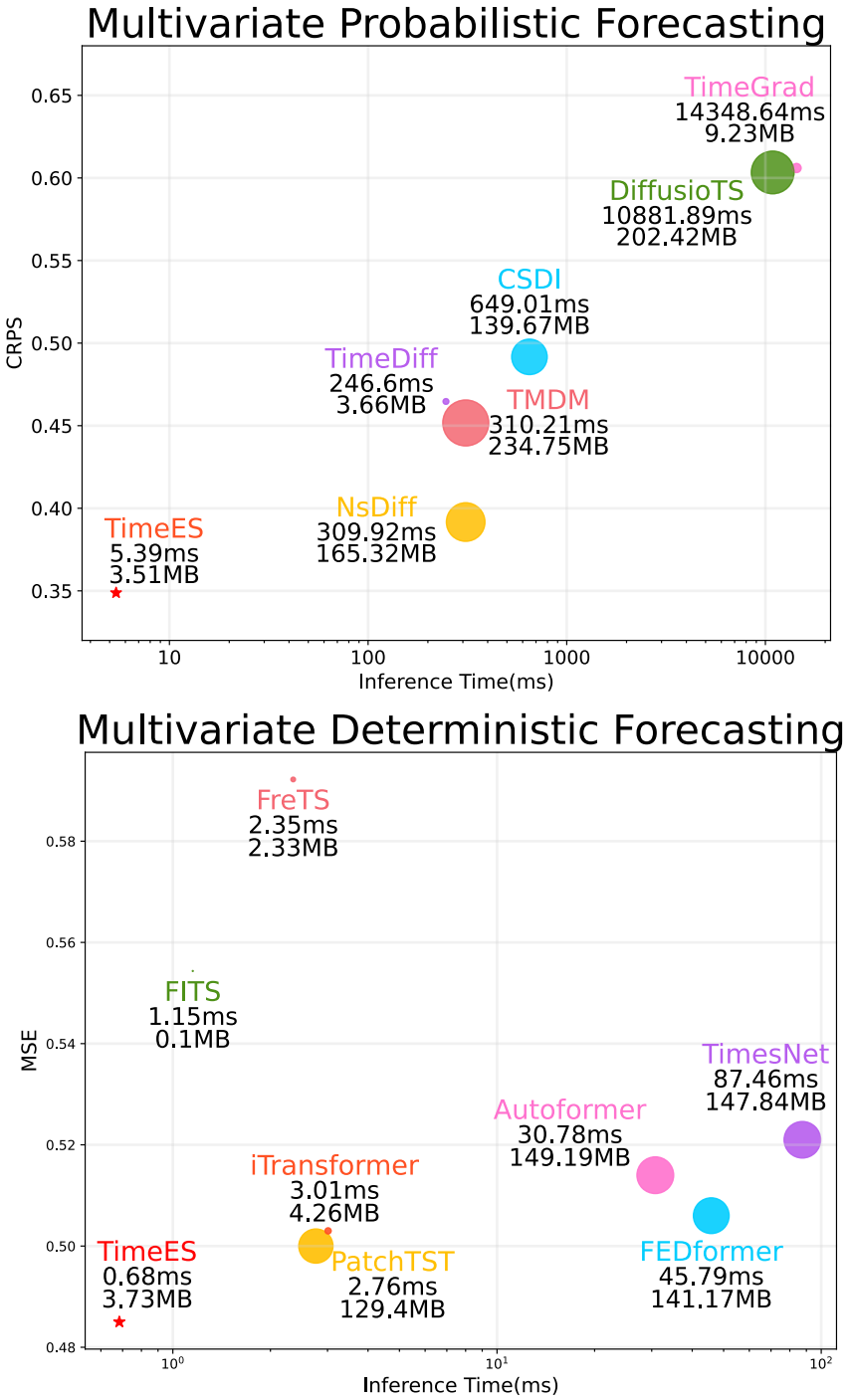
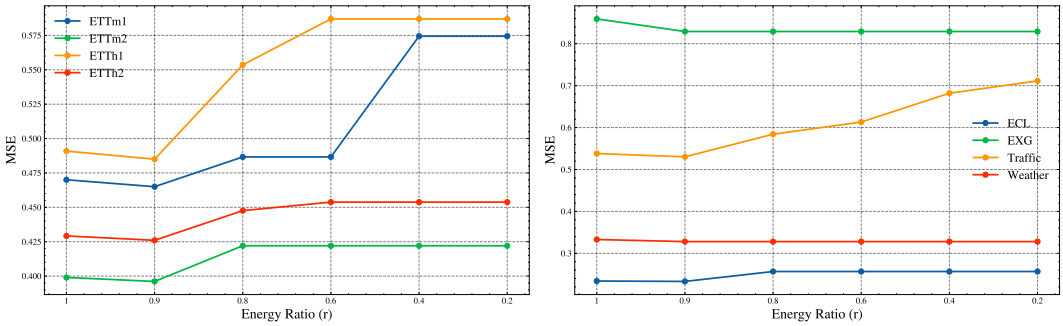


Figure 8: Efficiency analysis on ETTh1 dataset.

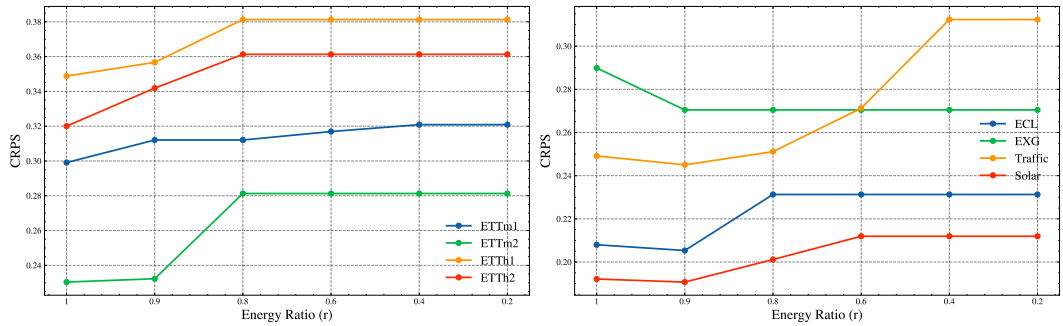
E Hyperparameter Sensitivity

We introduce a hyper-parameter r to retain the frequency components that collectively account for the top r ratio of total energy. The sensitivity analysis of this hyper-parameter is provided in Figure 9. We observe that our empirically selected value $r = 0.9$ consistently yields stable performance across various forecasting tasks. Notably, on the Traffic dataset, aggressive frequency pruning leads to significant performance degradation, highlighting the importance of modeling multiple frequency components, which is consistent with the broad energy distribution observed in Figure 5. Moreover, our energy-based selection strategy reliably captures the dominant frequencies, in contrast to naive

top-k approaches that may inadvertently miss critical spectral components due to rigid cardinality constraints. Importantly, on several datasets like ETTs multivariate in forecasting, appropriately selecting frequencies not only avoids performance loss but actually improves forecasting accuracy, further demonstrating that effective frequency selection can enhance model performance to a certain extent.



(a) Hyperparameter Sensitivity on Multivariate Forecasting



(b) Hyperparameter Sensitivity on Probabilistic Forecasting

Figure 9: We conduct a sensitivity analysis of the hyper-parameter r in multivariate long-term forecasting and probabilistic forecasting tasks. Specifically, we set the prediction horizon to 720 time steps. We report the Mean Squared Error (MSE) for deterministic forecasting and the Continuous Ranked Probability Score (CRPS) for probabilistic forecasting, evaluated on the ETT (ETTm1, ETTm2, ETTh1, ETTh2), ECL, EXG, Traffic, Weather, and Solar datasets.

F Additional Experimental Results

Due to page limits, we place some full results of our experiments in the following: classification in Table 10, probabilistic forecasting in Table 11 and Table 12, single variable deterministic forecasting in Table 13, multivariate deterministic forecasting results in Table 14, backbone enhancement results in Table 15. The statistical significance is within 3%.

Table 11: Probabilistic forecasting performance (CRPS, QICE) across nine datasets.

Models	Datasets	ETTh1	ETTh2	ETTm1	ETTm2	ECL	EXG	ILI	Solar	Traffic
TimeGrad	CRPS	0.606	1.212	0.647	0.775	0.397	0.826	1.140	0.293	0.407
(2021)	QICE	6.731	9.488	6.693	6.962	7.118	9.464	6.519	7.378	4.581
CSDI	CRPS	0.492	0.647	0.524	0.817	0.577	0.855	1.244	0.432	1.418
(2022)	QICE	3.107	5.331	2.828	8.106	7.506	7.864	7.693	9.957	13.613
TimeDiff	CRPS	0.465	0.471	0.464	0.316	0.750	0.433	1.153	0.700	0.771
(2023)	QICE	14.931	14.813	14.795	13.385	15.466	14.556	14.942	14.914	15.439
DiffusionTS	CRPS	0.603	1.168	0.574	1.035	0.633	1.251	1.612	0.470	0.668
(2024)	QICE	6.423	9.577	5.605	9.959	8.205	10.411	10.090	6.627	5.958
TMDM	CRPS	0.452	0.383	0.375	0.289	0.461	0.336	0.967	0.350	0.557
(2024)	QICE	2.821	4.471	2.567	2.610	10.562	6.393	6.217	9.342	10.676
NsDiff	CRPS	0.392	0.358	0.346	0.256	0.290	0.324	**0.806**	0.300	0.378
(2025)	QICE	**1.470**	**2.074**	**2.041**	2.030	6.685	5.930	**5.598**	6.820	3.601
TimeES	CRPS	**0.349**	**0.320**	**0.310**	**0.230**	**0.205**	**0.270**	0.834	**0.193**	**0.245**
(ours)	QICE	1.921	2.479	2.247	**1.984**	**1.941**	**4.302**	6.930	**4.768**	**2.177**

Table 12: Deterministic forecasting performance (MSE, MAE) across nine datasets.

Models	Datasets	ETTh1	ETTh2	ETTm1	ETTm2	ECL	EXG	ILI	Solar	Traffic
TimeGrad	MAE	0.813	1.496	0.831	0.967	0.504	1.058	1.414	0.446	0.535
(2021)	MSE	1.062	3.462	1.218	1.690	0.505	1.567	4.197	0.475	0.983
CSDI	MAE	0.708	0.900	0.752	1.069	0.822	1.081	1.481	0.675	0.925
(2022)	MSE	0.949	1.226	1.002	1.723	1.007	1.701	4.515	0.763	1.731
TimeDiff	MAE	0.479	0.485	0.477	0.333	0.764	0.446	1.169	0.713	0.784
(2023)	MSE	0.517	0.456	0.537	0.268	0.879	0.402	3.958	0.821	1.350
DiffusionTS	MAE	0.774	1.411	0.744	1.232	0.856	1.564	1.788	0.740	0.815
(2024)	MSE	1.089	3.273	1.030	2.372	1.072	3.628	6.053	0.749	1.473
TMDM	MAE	0.607	0.490	0.455	0.395	0.359	0.430	1.175	0.316	0.425
(2024)	MSE	0.696	0.512	0.494	0.315	0.257	0.334	3.636	0.250	0.679
NsDiff	MAE	0.523	0.490	0.455	0.352	0.306	0.412	**0.985**	0.307	0.373
(2025)	MSE	0.594	0.514	0.488	0.281	0.209	0.300	**2.846**	0.242	0.637
TimeES	MAE	**0.455**	**0.408**	**0.403**	**0.292**	**0.263**	**0.347**	1.043	**0.234**	**0.304**
(ours)	MSE	**0.474**	**0.376**	**0.371**	**0.235**	**0.180**	**0.242**	3.052	**0.179**	**0.456**

Table 13: Full results of the single variate forecasting task. We compare extensive competitive models under different prediction lengths following the setting of iTransformer [41]. The input sequence length is set to 96 for all baselines. The statistical significance is within 1%.

Model		Autoformer		FEDformer		TimesNet		FITS		FreTS		PatchTST		iTransformer		TimeES	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.886	0.636	0.713	0.598	0.682	0.551	0.643	0.511	**0.600**	0.508	0.624	0.508	0.622	0.517	0.607	**0.505**
	192	1.252	0.798	0.770	0.614	0.748	0.568	0.745	0.545	0.710	0.553	0.710	0.545	0.730	0.554	**0.690**	**0.534**
	336	0.974	0.681	0.886	0.664	0.810	0.592	0.811	0.573	0.788	0.592	0.786	0.585	0.829	0.609	**0.770**	**0.568**
	720	1.594	0.884	0.992	0.705	1.006	0.689	0.974	0.633	0.921	0.649	0.931	0.646	0.931	0.642	**0.893**	**0.619**
	Avg	1.177	0.750	0.840	0.645	0.812	0.600	0.793	0.566	0.755	0.576	0.763	0.571	0.778	0.581	**0.740**	**0.556**
ETTm2	96	0.324	0.390	0.298	0.385	0.292	**0.368**	0.313	0.396	0.288	0.391	0.292	0.382	0.328	0.401	**0.282**	0.375
	192	0.445	0.459	0.449	0.463	0.458	0.465	0.425	0.454	0.390	0.440	0.411	0.441	0.408	0.438	**0.379**	**0.435**
	336	0.549	0.516	0.556	0.519	0.557	0.524	0.521	0.505	0.514	0.503	0.503	0.496	0.544	0.520	**0.485**	**0.490**
	720	0.673	0.582	0.675	0.589	0.615	0.734	0.653	0.570	0.616	0.558	0.641	0.573	0.667	0.581	**0.590**	**0.553**
	Avg	0.498	0.487	0.494	0.489	0.480	0.523	0.478	0.481	0.452	0.473	0.462	0.473	0.487	0.485	**0.434**	**0.463**
ETTh1	96	0.871	0.658	**0.735**	0.582	0.800	0.619	1.064	0.748	0.836	0.624	0.812	0.616	0.916	0.672	0.749	**0.580**
	192	0.934	0.691	**0.849**	0.623	0.931	0.667	1.162	0.778	0.960	0.672	0.938	0.665	0.893	0.654	0.863	**0.621**
	336	**0.942**	0.694	0.957	**0.669**	1.020	0.692	1.233	0.798	1.103	0.733	1.064	0.703	1.054	0.708	0.972	0.673
	720	0.952	0.729	0.989	0.744	0.965	0.710	1.161	0.808	1.143	0.755	1.065	0.733	1.007	0.730	**0.929**	**0.670**
	Avg	0.925	0.693	0.882	0.655	0.929	0.672	1.155	0.783	1.010	0.696	0.970	0.679	0.968	0.691	**0.878**	**0.636**
ETTh2	96	0.587	0.562	0.579	0.552	0.550	0.532	0.494	0.503	0.484	0.499	**0.468**	0.491	0.497	0.507	0.478	**0.484**
	192	0.666	0.602	0.696	0.620	0.664	0.585	0.621	0.569	0.594	0.553	0.627	0.576	0.660	0.589	**0.576**	**0.541**
	336	0.773	0.664	0.828	0.697	0.769	0.641	0.761	0.647	0.679	0.594	0.714	0.632	0.763	0.657	**0.659**	**0.583**
	720	0.774	0.677	0.749	0.665	0.651	0.625	0.749	0.659	0.629	0.590	0.724	0.662	0.807	0.692	**0.618**	**0.585**
	Avg	0.700	0.626	0.713	0.634	0.659	0.596	0.656	0.595	0.597	0.559	0.633	0.590	0.682	0.611	**0.583**	**0.548**
ECL	96	0.492	0.493	0.390	0.358	0.367	0.309	0.339	0.294	0.370	0.374	0.357	0.321	0.349	0.306	**0.335**	**0.292**
	192	0.511	0.441	0.458	0.374	0.442	0.346	0.427	0.398	**0.414**	0.324	0.430	0.373	0.462	0.355	0.416	**0.323**
	336	0.603	0.465	0.588	0.428	0.590	0.409	0.557	**0.382**	0.544	0.470	0.550	0.440	0.568	0.392	**0.543**	0.386
	720	1.110	0.680	1.003	0.592	0.964	0.599	1.026	**0.573**	0.794	0.614	0.833	0.617	1.026	0.579	**0.771**	0.618
	Avg	0.679	0.520	0.610	0.438	0.591	0.416	0.587	0.412	0.530	0.446	0.542	0.438	0.601	0.408	**0.516**	**0.405**
EXG	96	0.214	0.356	0.147	0.281	0.137	0.272	0.193	0.328	0.170	0.311	0.131	**0.262**	0.131	0.267	**0.121**	0.276
	192	0.390	0.490	0.352	0.455	0.337	0.442	0.433	0.321	0.447	0.374	0.409	0.302	0.408	**0.291**	**0.226**	0.372
	336	0.807	0.722	0.658	0.641	0.662	0.631	0.628	0.637	0.605	0.618	0.600	0.609	0.588	0.588	**0.384**	**0.497**
	720	1.898	1.207	1.901	1.204	1.707	1.119	1.795	1.166	0.845	0.733	1.632	1.078	1.723	1.112	**0.694**	**0.693**
	Avg	0.827	0.694	0.764	0.645	0.711	0.616	0.762	0.613	0.517	0.509	0.693	0.563	0.712	0.565	**0.356**	**0.460**
Traffic	96	0.790	0.407	0.505	0.335	0.460	0.291	1.134	0.582	0.504	0.334	0.472	0.301	0.437	**0.274**	**0.432**	0.285
	192	0.992	0.518	0.520	0.338	0.548	0.335	1.122	0.589	0.524	0.332	0.488	0.287	**0.435**	**0.281**	0.445	0.287
	336	1.103	0.567	0.563	0.351	0.589	0.351	1.175	0.607	0.592	0.351	0.543	0.322	0.549	0.331	**0.424**	**0.295**
	720	1.190	0.599	0.674	0.402	0.658	0.376	1.224	0.604	0.682	0.384	0.583	0.321	0.580	0.349	**0.423**	**0.287**
	Avg	1.019	0.523	0.566	0.357	0.564	0.338	1.164	0.595	0.576	0.350	0.522	0.308	0.500	0.309	**0.431**	**0.288**
Weather	96	0.254	0.333	0.114	0.230	0.110	0.216	0.115	0.231	0.095	0.209	0.100	0.206	0.102	0.211	**0.093**	**0.202**
	192	0.533	0.524	0.381	0.429	0.404	0.419	0.386	0.424	0.316	0.379	0.329	0.383	0.348	0.399	**0.310**	**0.370**
	336	0.906	0.679	0.837	0.641	0.825	0.610	0.778	0.609	0.636	0.530	0.815	0.609	0.828	0.620	**0.630**	**0.526**
	720	1.335	0.865	1.286	0.836	1.226	0.814	1.241	0.821	1.045	0.713	1.316	0.831	1.300	0.839	**1.020**	**0.707**
	Avg	0.757	0.600	0.654	0.534	0.641	0.515	0.630	0.521	0.523	0.458	0.640	0.507	0.645	0.517	**0.513**	**0.451**
Solar	96	0.950	0.769	0.230	0.369	0.176	0.267	0.299	0.378	0.187	0.275	**0.174**	0.259	0.179	0.256	0.181	**0.253**
	192	4.459	1.776	0.242	0.363	0.209	0.306	0.327	0.397	0.209	0.290	0.183	0.271	**0.170**	**0.258**	0.205	0.271
	336	3.319	1.454	0.268	0.387	0.240	0.348	0.350	0.404	0.230	0.303	**0.194**	0.283	0.218	0.289	0.218	**0.279**
	720	3.455	1.450	0.269	0.377	0.239	0.312	0.346	0.391	0.232	0.304	**0.213**	0.289	0.235	0.297	0.217	**0.275**
	Avg	3.046	1.362	0.252	0.374	0.216	0.308	0.331	0.393	0.215	0.293	**0.191**	0.276	0.201	0.275	0.205	**0.270**
1st Count		1	0	2	1	0	1	0	2	2	0	4	1	2	4	**25**	**27**

Table 14: Full results of the multivariate forecasting task. We compare extensive competitive models under different prediction lengths following the setting of iTransformer [41]. The input sequence length is set to 96 for all baselines. The statistical significance is within 1%.

Model		Autoformer		FEDformer		TimesNet		FITS		FreTS		PatchTST		iTransformer		TimeES	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.475	0.505	0.379	0.419	0.338	0.375	0.353	0.378	0.352	0.385	0.329	0.367	0.334	0.368	**0.328**	**0.366**
	192	0.553	0.496	0.426	0.441	0.374	0.387	0.389	0.400	0.390	0.407	0.367	0.385	0.377	0.391	**0.364**	**0.384**
	336	0.621	0.537	0.445	0.459	0.410	0.411	0.447	0.441	0.421	0.423	0.410	**0.399**	0.426	0.420	**0.402**	0.402
	720	0.671	0.561	0.543	0.490	0.478	0.450	0.482	**0.445**	0.495	0.475	**0.439**	0.454	0.491	0.459	0.465	**0.445**
	Avg	0.580	0.525	0.448	0.452	0.400	0.406	0.418	0.416	0.414	0.423	**0.386**	0.401	0.407	0.410	0.390	**0.400**
ETTm2	96	0.255	0.339	0.203	0.287	0.187	0.267	0.183	0.264	0.190	0.280	**0.175**	0.259	0.180	0.264	0.176	**0.258**
	192	0.281	0.340	0.269	0.328	0.249	0.309	0.246	0.303	0.267	0.335	0.241	0.302	0.250	0.309	**0.240**	**0.300**
	336	0.339	0.372	0.325	0.366	0.321	0.351	0.305	0.340	0.339	0.378	0.305	0.343	0.311	0.348	**0.300**	**0.338**
	720	0.433	0.432	0.421	0.415	0.408	0.403	0.406	0.396	0.495	0.476	0.402	0.400	0.412	0.407	**0.396**	**0.395**
	Avg	0.327	0.371	0.305	0.349	0.291	0.333	0.285	0.326	0.323	0.367	0.281	0.326	0.288	0.332	**0.278**	**0.323**
ETTh1	96	0.449	0.459	0.376	0.419	0.384	0.402	0.490	0.469	0.425	0.435	0.414	0.419	0.386	0.405	**0.374**	**0.397**
	192	0.500	0.482	**0.420**	0.448	0.436	0.429	0.531	0.491	0.478	0.469	0.460	0.445	0.441	0.436	0.430	**0.426**
	336	0.521	0.496	**0.459**	0.465	0.491	0.469	0.561	0.506	0.520	0.485	0.501	0.466	0.487	0.458	0.470	**0.446**
	720	0.514	0.512	0.506	0.507	0.521	0.500	0.554	0.519	0.592	0.544	0.500	0.488	0.503	0.491	**0.485**	**0.479**
	Avg	0.496	0.487	**0.440**	0.460	0.458	0.450	0.534	0.496	0.504	0.483	0.469	0.455	0.454	0.448	**0.440**	**0.437**
ETTh2	96	0.346	0.388	0.358	0.397	**0.340**	0.374	0.312	0.355	0.353	0.400	0.302	0.348	0.297	0.349	**0.293**	**0.344**
	192	0.456	0.452	0.429	0.439	0.402	0.414	0.399	0.406	0.462	0.462	0.388	0.400	0.380	0.400	**0.371**	**0.395**
	336	0.482	0.486	0.496	0.487	0.452	0.452	0.435	0.438	0.500	0.487	**0.426**	0.433	0.428	**0.432**	0.427	0.440
	720	0.515	0.511	0.463	0.474	0.462	0.468	0.437	0.449	0.684	0.590	0.431	0.446	0.427	0.445	**0.426**	**0.443**
	Avg	0.450	0.459	0.437	0.449	0.414	0.427	0.396	0.412	0.500	0.485	0.387	0.407	0.383	0.407	**0.379**	**0.405**
ECL	96	0.201	0.317	0.193	0.308	0.168	0.272	0.312	0.405	0.181	0.270	0.195	0.285	**0.148**	**0.240**	0.175	0.264
	192	0.222	0.334	0.201	0.315	0.184	0.289	0.313	0.406	0.188	0.277	0.199	0.289	**0.162**	**0.253**	0.182	0.272
	336	0.231	0.338	0.214	0.329	0.198	0.300	0.326	0.414	0.208	0.298	0.215	0.305	**0.178**	**0.269**	0.197	0.288
	720	0.254	0.361	0.246	0.355	**0.220**	0.320	0.362	0.435	0.243	0.330	0.256	0.337	0.225	**0.317**	0.233	0.320
	Avg	0.227	0.338	0.214	0.327	0.193	0.295	0.328	0.415	0.205	0.294	0.216	0.304	**0.178**	**0.270**	0.197	0.286
EXG	96	0.197	0.323	0.148	0.278	0.107	0.234	0.104	0.229	0.094	0.223	0.088	0.205	0.086	0.206	**0.082**	**0.200**
	192	0.300	0.369	0.271	0.315	0.226	0.344	0.200	0.322	0.196	0.332	**0.176**	0.299	0.177	0.299	**0.176**	**0.298**
	336	0.509	0.524	0.460	0.427	0.367	0.448	0.352	0.433	0.363	0.453	**0.301**	**0.397**	0.331	0.417	0.338	0.421
	720	1.447	0.941	1.195	0.695	0.964	0.746	0.900	0.721	1.033	0.755	0.901	0.714	0.847	0.691	**0.829**	**0.684**
	Avg	0.613	0.539	0.519	0.429	0.416	0.443	0.389	0.426	0.422	0.441	0.367	0.404	0.360	0.403	**0.357**	**0.401**
Traffic	96	0.613	0.941	0.587	0.366	0.593	0.321	0.831	0.520	0.531	0.340	0.544	0.359	**0.395**	**0.268**	0.446	0.290
	192	0.616	0.382	0.604	0.373	0.617	0.336	0.808	0.513	0.534	0.338	0.540	0.354	**0.417**	**0.276**	0.462	0.295
	336	0.622	0.337	0.621	0.383	0.629	0.336	0.817	0.515	0.553	0.345	0.551	0.358	**0.433**	**0.283**	0.475	0.306
	720	0.660	0.408	0.626	0.382	0.640	0.350	0.852	0.527	0.602	0.366	0.586	0.375	**0.467**	**0.302**	0.530	0.318
	Avg	0.628	0.517	0.610	0.376	0.620	0.336	0.827	0.519	0.555	0.347	0.555	0.362	**0.428**	**0.282**	0.478	0.302
Weather	96	0.266	0.336	0.217	0.296	0.172	0.220	0.171	0.219	0.169	0.227	0.177	0.218	0.174	**0.214**	**0.164**	0.221
	192	0.307	0.367	0.276	0.336	0.219	0.261	0.218	0.260	0.212	0.272	0.225	0.259	0.221	**0.254**	**0.205**	0.262
	336	0.359	0.395	0.339	0.380	0.280	0.306	0.274	0.299	0.261	0.307	0.278	0.297	0.278	**0.296**	**0.257**	0.307
	720	0.419	0.428	0.403	0.428	0.365	0.359	0.353	0.349	0.329	0.356	0.354	**0.348**	0.358	0.349	**0.328**	0.358
	Avg	0.338	0.382	0.309	0.360	0.259	0.287	0.254	0.282	0.243	0.291	0.259	0.281	0.258	**0.278**	**0.239**	0.287
Solar	96	0.884	0.711	0.242	0.342	0.250	0.292	0.351	0.382	0.224	0.297	0.234	0.286	**0.203**	**0.237**	0.226	0.278
	192	0.834	0.692	0.285	0.380	0.296	0.318	0.397	0.413	0.254	0.312	0.267	0.310	**0.233**	**0.261**	0.248	0.291
	336	0.941	0.723	0.282	0.376	0.319	0.330	0.434	0.424	0.274	0.326	0.290	0.315	**0.248**	**0.273**	0.253	0.292
	720	0.882	0.717	0.357	0.427	0.338	0.337	0.430	0.411	0.269	0.320	0.289	0.317	0.249	**0.275**	**0.245**	0.288
	Avg	0.885	0.711	0.292	0.381	0.301	0.319	0.403	0.407	0.255	0.314	0.270	0.307	**0.233**	0.262	0.243	**0.287**
1st Count		0	0	2	2	2	0	0	0	0	1	5	2	9	15	**22**	**16**

Table 15: We evaluate TimeES as a decomposition framework on long-term single variate and multivariate forecasting tasks in ETTs datasets, the best results are denote in **bold** characters.

Single-Variates

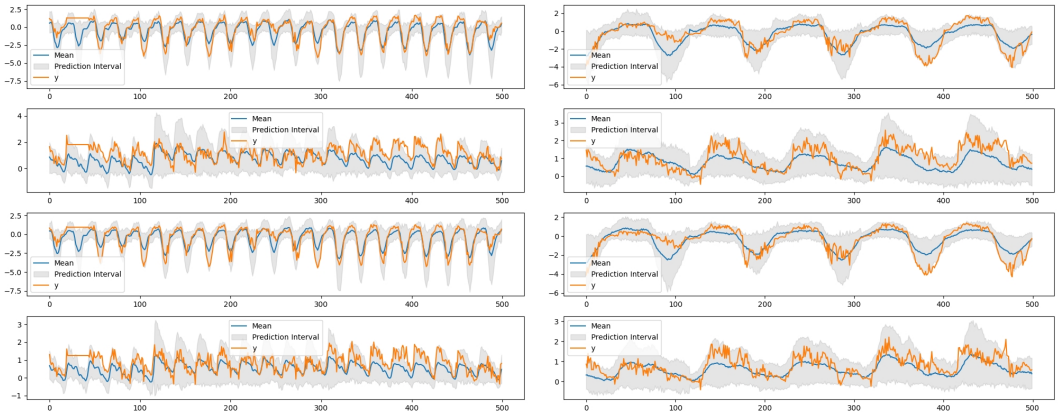
Model		PatchTST		+TimeES		Improvements		iTransformer		+TimeES		Improvements	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.624	0.508	**0.603**	**0.505**	3.37%	0.63%	0.622	0.517	**0.612**	**0.496**	1.56%	4.02%
	192	0.710	0.545	**0.704**	**0.533**	0.88%	2.35%	0.730	0.554	**0.721**	**0.544**	1.35%	1.77%
	336	0.786	0.585	**0.780**	**0.574**	0.70%	1.80%	0.829	0.609	**0.787**	**0.576**	5.07%	5.39%
	720	0.931	0.646	**0.921**	**0.635**	1.15%	1.63%	0.931	0.642	**0.923**	**0.634**	0.86%	1.23%
	Avg	0.763	0.571	**0.752**	**0.562**	1.42%	1.62%	0.778	0.580	**0.760**	**0.563**	2.24%	3.07%
ETTm2	96	0.292	0.382	**0.289**	**0.370**	1.01%	3.30%	0.328	0.401	**0.296**	**0.374**	9.90%	6.59%
	192	0.411	0.441	**0.411**	**0.440**	0.08%	0.10%	0.408	**0.438**	**0.407**	**0.438**	0.31%	-0.09%
	336	0.503	0.496	**0.492**	**0.491**	2.24%	1.08%	0.544	0.520	**0.504**	**0.493**	7.48%	5.16%
	720	0.641	0.573	**0.625**	**0.565**	2.42%	1.42%	0.667	0.581	**0.634**	**0.559**	4.94%	3.71%
	Avg	0.462	0.473	**0.454**	**0.466**	1.63%	1.40%	0.487	0.485	**0.460**	**0.466**	5.51%	3.84%
ETTh1	96	0.812	0.616	**0.808**	**0.604**	0.55%	1.94%	0.916	0.672	**0.786**	**0.619**	14.18%	7.88%
	192	0.938	0.665	**0.893**	**0.622**	4.74%	6.44%	0.893	0.654	**0.822**	**0.572**	7.91%	12.53%
	336	1.064	0.703	**1.043**	**0.701**	2.03%	0.33%	1.054	0.708	**1.021**	**0.706**	3.20%	0.28%
	720	1.065	0.733	**0.945**	**0.716**	11.23%	2.34%	1.007	0.730	**0.951**	**0.724**	5.50%	0.82%
	Avg	0.970	0.679	**0.922**	**0.661**	4.90%	2.73%	0.968	0.691	**0.895**	**0.655**	7.48%	5.16%
	Avg	0.469	0.455	**0.444**	**0.442**	5.19%	2.73%	0.968	0.691	**0.895**	**0.655**	7.48%	5.16%
ETTh2	96	0.468	0.491	**0.454**	**0.484**	3.14%	1.47%	0.497	0.507	**0.453**	**0.479**	8.92%	5.51%
	192	0.627	0.576	**0.597**	**0.565**	4.77%	1.99%	0.660	0.589	**0.586**	**0.552**	11.20%	6.20%
	336	0.714	0.632	**0.713**	**0.631**	0.13%	0.10%	0.763	0.657	**0.704**	**0.618**	7.70%	5.92%
	720	0.724	0.662	**0.718**	**0.658**	0.91%	0.68%	0.807	0.692	**0.695**	**0.639**	13.80%	7.65%
	Avg	0.633	0.590	**0.620**	**0.584**	2.06%	1.01%	0.682	0.611	**0.609**	**0.572**	10.57%	6.39%

Multivariate

Model		PatchTST		+TimeES		Improvements		iTransformer		+TimeES		Improvements	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.329	0.367	**0.324**	**0.362**	1.43%	1.37%	0.334	0.368	**0.333**	**0.367**	0.23%	0.14%
	192	**0.367**	**0.385**	0.369	0.388	-0.44%	-0.79%	**0.377**	**0.391**	0.378	0.392	-0.33%	-0.21%
	336	**0.410**	**0.399**	0.412	0.410	-0.43%	-2.73%	0.426	0.420	**0.412**	**0.413**	3.34%	1.63%
	720	**0.439**	0.454	0.461	**0.443**	-5.01%	2.36%	0.491	0.459	**0.480**	**0.452**	2.22%	1.49%
	Avg	**0.386**	**0.401**	0.391	**0.401**	-1.34%	0.11%	0.407	0.410	**0.401**	**0.406**	1.52%	0.82%
ETTm2	96	**0.175**	**0.259**	0.182	0.268	-3.79%	-3.31%	**0.180**	**0.264**	0.185	0.266	-2.53%	-0.81%
	192	**0.241**	**0.302**	0.246	0.310	-2.25%	-2.71%	0.250	0.309	**0.248**	**0.306**	0.73%	0.87%
	336	0.305	**0.343**	**0.304**	0.346	0.42%	-0.97%	0.311	0.348	**0.308**	**0.344**	0.82%	1.13%
	720	**0.402**	**0.400**	**0.402**	**0.400**	0.00%	0.00%	0.412	0.407	**0.407**	**0.401**	1.17%	1.49%
	Avg	**0.281**	**0.326**	0.283	0.331	-0.96%	-1.54%	0.288	0.332	**0.287**	**0.329**	0.41%	0.79%
ETTh1	96	0.414	0.419	**0.385**	**0.402**	6.89%	4.15%	0.386	0.405	**0.384**	**0.402**	0.56%	0.78%
	192	0.460	0.445	**0.434**	**0.437**	5.65%	1.86%	0.441	0.436	**0.433**	**0.428**	1.88%	1.77%
	336	0.501	0.466	**0.451**	**0.457**	9.99%	2.02%	0.487	0.458	**0.475**	**0.451**	2.44%	1.51%
	720	0.500	0.488	**0.482**	**0.479**	3.68%	1.82%	0.503	0.491	**0.474**	**0.480**	5.83%	2.18%
	Avg	0.469	0.455	**0.438**	**0.444**	6.56%	2.42%	0.454	0.448	**0.441**	**0.440**	2.84%	1.59%
ETTh2	96	0.302	0.348	**0.289**	**0.344**	4.19%	1.03%	0.297	0.349	**0.297**	**0.345**	-0.01%	1.09%
	192	0.388	0.400	**0.365**	**0.390**	5.81%	2.43%	0.380	0.400	**0.376**	**0.395**	1.17%	1.34%
	336	0.426	0.433	**0.406**	**0.421**	4.61%	2.69%	0.428	0.432	**0.418**	**0.429**	2.43%	0.62%
	720	0.431	0.446	**0.421**	**0.441**	2.32%	1.17%	0.427	0.445	**0.424**	**0.444**	0.77%	0.28%
	Avg	0.387	0.407	**0.371**	**0.399**	4.19%	1.85%	0.383	0.407	**0.378**	**0.403**	1.18%	0.81%

G Additional Showcases

We present other probabilistic forecasting showcases in Figure 10.



(a) ETTh1 (b) ETTm1

(c) ETTh2 (d) Electricity

(e) ExchangeRate (f) ILI

Figure 10: (a-f) Predictive uncertainty quantification by TimeES across different datasets. We plot the last 500 predicted time steps of the first four variables in each dataset. The orange line shows the ground truth, the blue line represents the predicted mean, and the gray shaded region indicates the 96% prediction interval.

H Additional Interpretability Analysis

We present additional results of the spectral decomposition analysis in Figure 11.

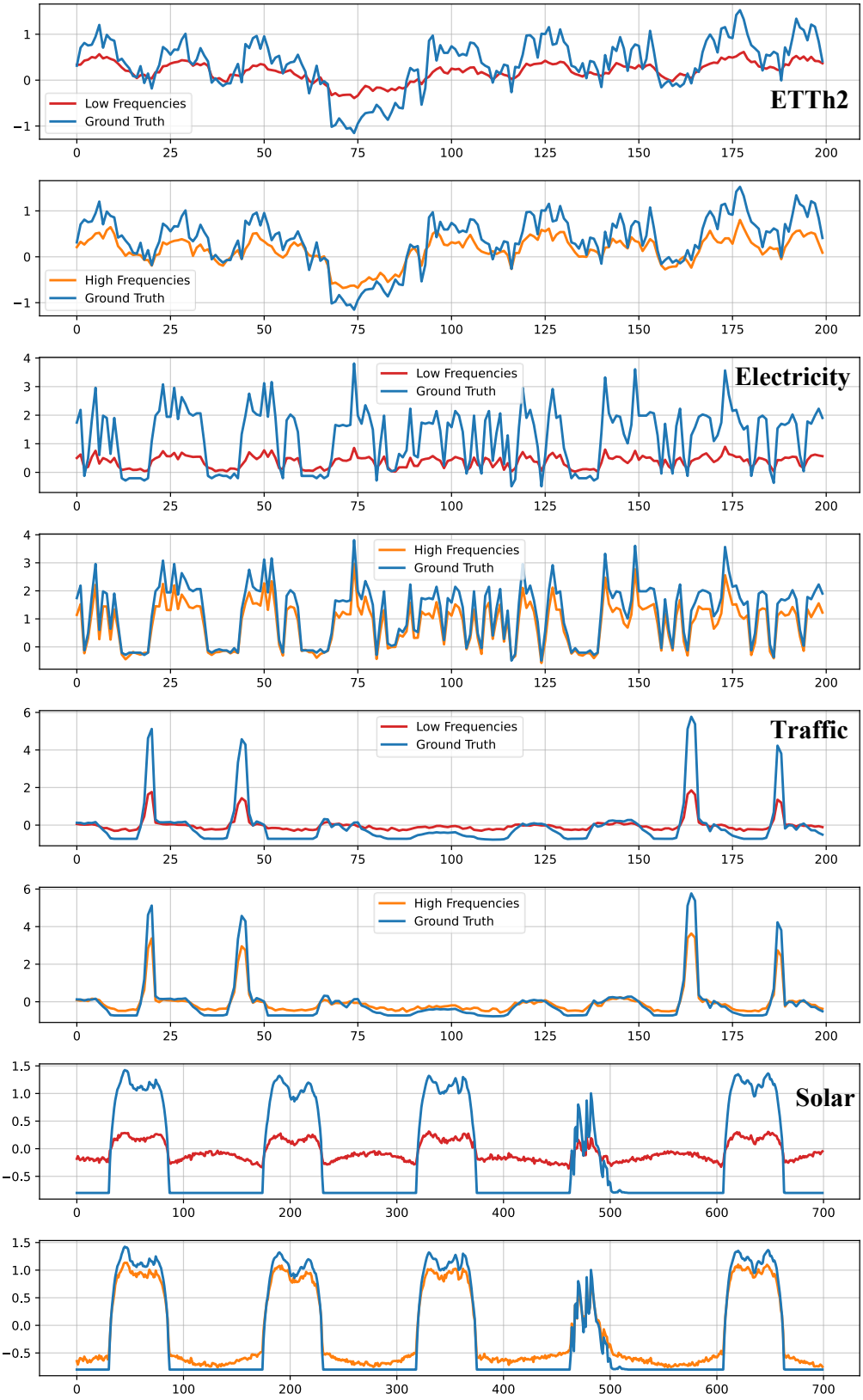


Figure 11: Spectral decomposition analysis on four dataset. We visualize the high and low frequencies.

Per-channel uncertainty attribution. The variance decomposition in Equation (33) attributes the predictive uncertainty of each channel to individual frequencies. Since the learned σ_{W_k} are nearly constant (Appendix J.2), the structure of the uncertainty lies in the amplitude, and the quantity to read is the share of each frequency in the predictive variance, $dS(n, \omega_k) / \sum_{k'} dS(n, \omega_{k'})$. We compute these shares on ETTh1 ($M = 168$, i.e., 85 frequencies, for which a uniform share would be 1.18%), averaged over all forecast steps of 672 test windows. Table 16 reports the shares at the trend (DC), weekly, daily, 12-hour, and 8-hour frequencies, and Figure 12 shows all 85 frequencies. For the load channels, the trend carries 1.4 to 2.2 times the uniform share, with the largest values for HULL and MULL, whereas OT (oil temperature) places more uncertainty in the weekly band than any load channel (1.51% vs. 0.44% to 1.05%), consistent with oil temperature being a slowly varying thermal state.

Table 16: Share (%) of the predictive variance contributed by selected frequencies on ETTh1, averaged over 672 test windows. A uniform share would be 1.18%. The largest value in each column is in **bold**.

Channel	Trend (DC)	Weekly (168h)	Daily (24h)	12h	8h
HUFL (high useful load)	1.95	0.46	1.60	0.80	0.95
HULL (high useless load)	2.55	0.83	1.34	0.71	0.78
MUFL (middle useful load)	2.02	0.44	1.57	0.79	0.94
MULL (middle useless load)	**2.56**	0.84	1.32	0.70	0.81
LUFL (low useful load)	1.63	0.84	**1.81**	**0.94**	0.96
LULL (low useless load)	2.07	1.05	1.34	0.83	1.10
OT (oil temperature)	1.73	**1.51**	1.54	0.69	**1.14**

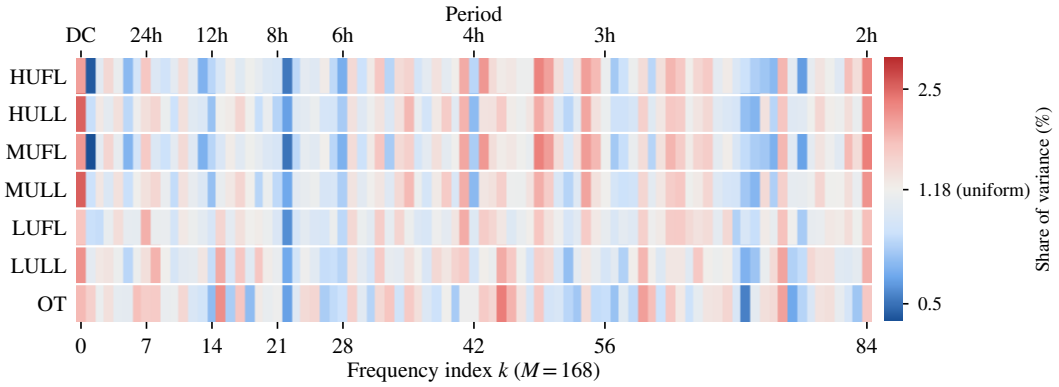


Figure 12: Share of the predictive variance contributed by each frequency on ETTh1 for each channel ($M = 168$, 85 frequencies). Red and blue cells indicate shares above and below the uniform share of 1.18%, respectively; $k = 1$ corresponds to the weekly period.

I Relation to Koopman Operators and Dynamic Mode Decomposition

Koopman operator theory represents nonlinear dynamics by a linear operator acting on a lifted space of observables. Besides deterministic forecasters such as KNF [65], Koopa [40], and SKOLR [80], which we compare with in Appendix C.5, this line of work also learns continuous spectra [42] and includes probabilistic formulations such as Deep Probabilistic Koopman [44], Koopman VAE [47], and KooNPro [81]. Both Koopman models and TimeES describe time series through a spectrum, but they treat the spectrum differently. Taking KooNPro as a representative probabilistic Koopman model, we compare the two in terms of the randomness in the spectrum, the spectral decomposition, and the parameterization of non-stationarity, and then discuss the forecasting procedure, the relation to DMD, and an empirical comparison.

I.1 Randomness in the Spectrum

Both models place a Gaussian distribution on a spectral quantity, but on different ones. KooNPro places it on the frequency, $\omega_t \sim \mathcal{N}(\mu(\kappa_t), \sigma(\kappa_t))$, so its variance is a variance *of frequencies* and expresses uncertainty about which dynamics generated the data. In evolutionary spectra, the spectrum *is* the variance of the observation, resolved by frequency. Since the W_k in Theorem 4.1 are independent, the predictive variance decomposes additively over the retained frequencies:

$$dS(n, \omega_k) = \frac{1}{M} |A(n, \omega_k)|^2 \, \sigma^2_{W_k}, \qquad \sum_{k \in \mathcal{K}} dS(n, \omega_k) = \mathrm{Var}[X_n], \tag{33}$$

where $\sigma^2_{W_k} = \mathbb{E}|W_k - \mu_{W_k}|^2$.[8] We verify this identity with the trained ETTh1 probabilistic model on one test batch (32 windows, 7 channels, and 192 steps), comparing the analytic sum with the empirical variance of 4,000 sampled paths. Averaged over all steps, the analytic sum is 1.310 and the empirical variance is 1.307, with a mean relative error of 1.8% (maximum 9.3%), which matches the sampling error expected from 4,000 Gaussian samples (a mean absolute relative error of $\sqrt{4/(4000\pi)} \approx 1.8\%$). The randomness modeled by TimeES is therefore the stochastic excitation of the process itself, i.e., the discrete counterpart of $\mathbb{E}|dZ(\omega)|^2$ in Definition 3.2.

I.2 Spectral Decomposition

The two spectra decompose different objects. A Koopman spectrum decomposes the *evolution operator* in a lifted space: the modes live in the latent state h and reach the observation only through a decoder ϕ^{-1}, where ϕ denotes the encoder. Since ϕ^{-1} is nonlinear, $\phi^{-1}\big(\sum_j h^{(j)}\big) \neq \sum_j \phi^{-1}\big(h^{(j)}\big)$, so individual modes are not separable signal components. An evolutionary spectrum decomposes the *observation itself*, and it does so additively:

$$x_n = \sum_{k \in \mathcal{K}} x_n^{(k)}, \qquad x_n^{(k)} = \frac{1}{\sqrt{M}} A(n, \omega_k) \, W_k \, e^{i\omega_k n}. \tag{34}$$

Any sub-band of frequencies therefore yields a valid time-domain component on its own, without a decoder. This additivity underlies the band-wise reconstruction in Section 5.4 and the per-channel uncertainty attribution in Appendix H.

I.3 Parameterization of Non-stationarity

KooNPro moves the frequencies (ω_t is re-drawn at every step), which is a form of frequency modulation. DES fixes the frequencies on the DFT grid of the observation and lets their amplitudes evolve, which is amplitude modulation, the form prescribed by Priestley's construction [49]. This costs no generality: since $A(n, \omega_k)$ is an unconstrained function of n, an off-grid component $c\, e^{i\omega^* n}$ is represented exactly on a grid frequency ω_k by $A(n, \omega_k) = c\, e^{i(\omega^* - \omega_k)n}$, so a time-varying amplitude already subsumes a shifted frequency. Fixing the grid also keeps the synthesis basis orthogonal, without which $|A|^2$ would no longer be a power in the Parseval sense.

Table 17: CRPS (mean±std over 3 seeds) of TimeES with frequencies fixed on the DFT grid or learned jointly with the amplitudes.

Dataset	Fixed grid	Learnable
ETTh1	**0.363±0.006**	0.372±0.008
Off-grid synthetic	**0.156±0.018**	0.157±0.017

Fixed grid vs. learnable frequencies. To test whether the fixed grid restricts the model, we make the frequencies learnable (initialized at the selected bins and trained jointly with the amplitudes) and compare against the fixed grid on ETTh1 and on a synthetic series whose periods deliberately miss the grid. The synthetic series is a sum of two sinusoids with periods of 30 and 45 steps plus Gaussian noise; with $M = 168$, these correspond to 5.60 and 3.73 cycles per window, both between DFT

[8] For real-valued series synthesized under Hermitian symmetry (Theorem 4.3), each retained frequency is paired with its conjugate, which changes the constant $1/M$ but not the additive decomposition. The check below uses the exact constant of our implementation.

bins. Both variants use the same training settings and seeds. As shown in Table 17, the accuracy is unchanged and the frequencies stay on the grid: the largest movement of any frequency is 0.065 of a bin (0.02 to 0.03 on average). On the off-grid series, reaching the true components would require the nearest selected bins, 6 and 4, to move by 0.40 and 0.27 of a bin, roughly six and four times further than the optimizer actually moves them. The freedom to leave the grid is therefore available but unused, as the above argument predicts.

I.4 Recursive vs. Closed-form Forecasting

KooNPro draws a global dynamics representation S from the neural-process context, which is frozen as S_C at test time, and then applies its Koopman operator $\mathcal{U}$ recursively. DES instead predicts the whole amplitude field at once, so the time index enters only through a fixed Fourier phase and the entire horizon is a single matrix product:

$$z_H = \phi^{-1}\big(\mathcal{U}^{H-1}(\phi(z_1), S_C)\big) \qquad \text{vs.} \qquad \mathbf{y} = \frac{1}{\sqrt{M}}\big(\hat{\mathbf{A}} \odot \mathbf{F}\big)\mathbf{w}, \tag{35}$$

where H denotes the forecast horizon. The left side requires $H-1$ sequential applications of $\mathcal{U}$, and since the frequency sampled at each step feeds into the next one, perturbations propagate rather than average out. The right side involves no recursion: lengthening the horizon adds terms to a sum rather than links to a chain, and a single draw of $\{W_k\}$ keeps each sampled path spectrally coherent.

I.5 Relation to Dynamic Mode Decomposition

DMD [56] fits a linear operator $\mathbf{G}$ between consecutive snapshot matrices, $\mathbf{X}_1 \approx \mathbf{G}\mathbf{X}_0$, and reconstructs the series from its eigen-decomposition as $\mathbf{x}_n = \sum_j b_j \boldsymbol{\psi}_j e^{(\gamma_j + i\nu_j)n}$, where $\boldsymbol{\psi}_j$ is a mode with amplitude b_j, frequency ν_j, and growth rate γ_j. Each DMD mode thus oscillates at a single frequency under an exponential envelope, and both are fixed once the operator is fitted. In the notation of DES, a DMD reconstruction corresponds to deterministic weights $W_k = 1$ and amplitudes of the restricted form $A(n, \omega_k) \propto e^{\gamma_j n} e^{i(\nu_j - \omega_k)n}$, following the off-grid argument in Appendix I.3. DES removes this restriction: the amplitudes are arbitrary functions of n predicted from the lookback window, the whole horizon is produced at once instead of by iterating a one-step operator, and the random weights W_k turn the decomposition into a generative model.

I.6 Empirical Comparison with KooNPro

The experimental protocols of KooNPro and TimeES differ in both history length and horizon: KooNPro uses a history of $L = 10$ steps on the ETT datasets and a horizon of $H = 24$, whereas our benchmark uses $L = 168$ and $H = 192$. We therefore ran the official implementation of KooNPro under both history lengths and at both horizons, with 3 seeds for each setting, and report the CRPS in Table 18. TimeES outperforms KooNPro in all settings. At $H = 24$, it reduces the CRPS by 27% to 54% with $L = 168$, and by 33% to 42% with $L = 10$, except on ETTm1, where 10 steps of 15-minute data cover only 2.5 hours and the margin narrows to 5%. At the horizon of our benchmark ($H = 192$), the CRPS of KooNPro increases by roughly an order of magnitude, and 83% to 88% of the test points fall below its 10th predictive percentile, which is consistent with the error accumulation of recursive forecasting discussed above.

Table 18: CRPS comparison with KooNPro on the ETT datasets (3 seeds per setting). L denotes the history length and H the forecast horizon; the original setting of KooNPro is $L = 10$ and $H = 24$, and the TimeES results for $L = 168$, $H = 192$ are from Table 3. **Bold** indicates the better result.

Setting	Model	ETTh1	ETTh2	ETTm1	ETTm2
$L = 10$, $H = 24$	KooNPro	0.527	0.351	0.395	0.277
	TimeES	**0.355**	**0.202**	**0.375**	**0.175**
$L = 168$, $H = 24$	KooNPro	0.398	0.380	0.352	0.290
	TimeES	**0.292**	**0.206**	**0.175**	**0.132**
$L = 168$, $H = 192$	KooNPro	3.079	2.922	3.465	3.656
	TimeES	**0.349**	**0.320**	**0.310**	**0.230**

J Further Discussion and Analysis

J.1 Frequency Support and Evolving Spectra

The energy-based selection in Section 4.2 fixes a set of frequencies $\mathcal{K}$, whereas the premise of TimeES is that spectra evolve. The two are compatible because they concern different objects. In DES (Theorem 4.1), non-stationarity is carried by the time-varying complex amplitude $A(n, \omega_k)$, not by the support $\mathcal{K}$ of the retained frequencies. $\mathcal{K}$ only determines which columns of the complete DFT grid are modeled, and a time-varying amplitude on a fixed grid can represent evolving spectral content, including frequency drift, since amplitude modulation between neighboring grid frequencies is equivalent to a varying instantaneous frequency (Appendix I.3). The chirp experiment in Appendix C.6 illustrates this: the test region contains frequencies never seen during training, yet TimeES tracks both the evolving frequency and the evolving amplitude. Moreover, $\mathcal{K}$ is estimated from the average periodogram of the entire training split rather than from individual lookback windows, so it reflects the spectral support of the dataset rather than a snapshot of a single window.

To quantify how well this support transfers to unseen data, we select $\mathcal{K}$ on the training split only ($M = 96$ and $r = 0.9$) and measure the fraction of the spectral energy of the test split captured by these frequencies. As shown in Table 19, the train-selected support captures 92.3% of the test energy on average, close to the target ratio $r = 0.9$, while the evolution of the spectrum within this support is modeled by $A(n, \omega_k)$.

Table 19: Fraction of the spectral energy of the test split captured by the frequencies selected on the training split ($M = 96$, $r = 0.9$).

Dataset	ETTh1	ETTh2	ETTm1	ETTm2	ECL	EXG	Traffic	Weather	Solar	ILI	Avg.
Coverage	0.885	0.862	0.914	0.900	0.915	0.992	0.904	0.965	0.928	0.963	0.923

J.2 The Spectral Random Variables

Parameterization. The parameters μ_{W_k} and σ_{W_k} are learnable constants, one pair for each selected frequency and each channel ($K \times C$ pairs), shared across the forecast horizon. The horizon dependence of the predictive distribution therefore enters entirely through $A(n, \omega_k)$, consistent with the ES construction, in which the randomness is carried by the spectral measure $dZ(\omega)$ with $\mathbb{E}|dZ(\omega)|^2 = d\omega$ and all non-stationary structure is carried by $A(t, \omega)$. Training recovers this structure: on ETTh1, the learned σ_{W_k} stay within about 4% of their initial value $\sqrt{2}$ (unit variance for both the real and imaginary parts), ranging from 1.36 to 1.42 over all 595 frequency-channel pairs. The spectral variance of $\mathbf{w}$ is thus nearly flat across frequencies, and the modulation of uncertainty over time and frequency is expressed by $|A(n, \omega_k)|$, with the per-step predictive variance given by Equation (33).

Expressiveness. The synthesis is linear in $\mathbf{w}$, but its coefficients are not: $A(n, \omega_k)$ is a nonlinear function of the input, so DES is a conditional location-scale model whose mean and full predictive covariance are both predicted nonlinearly from the history. This yields two properties that models with independent per-step noise lack. First, a single draw of $\{W_k\}$ is shared by the entire horizon, so a sampled trajectory stays spectrally coherent instead of fluctuating independently at each step. Second, the covariance between any two forecast steps is available in closed form rather than only through sampling; for the complex synthesis in Theorem 4.1, $\mathrm{Cov}[X_n, X_m] = \frac{1}{M} \sum_k A(n, \omega_k) \overline{A(m, \omega_k)}\, \sigma^2_{W_k} e^{i\omega_k (n-m)}$. Empirically, the Gaussian assumption does not appear restrictive: TimeES attains the best QICE on 5 of 9 datasets and the second best on 3 (Table 11), against diffusion-based baselines with unconstrained generators. Its cost is limited flexibility in the tails of the per-step distribution, and extending DES with non-Gaussian or mixture spectral measures is a natural direction for future work.

J.3 Sensitivity to Input Length

We examine the effect of the input length N on probabilistic forecasting, setting $M = N$ and keeping all other settings of Section 5.1 (horizon 192, 3 seeds). As shown in Table 20, TimeES is relatively robust to the input length. On ETTh1, even a 24-step input (one day) stays within 2% of the default setting. On ETTm1 (15-minute granularity), only the 24-step input degrades markedly (+28%), since

6 hours do not cover a daily cycle; from 72 steps on, the CRPS stays within 4% of the default setting, and the 96- and 336-step inputs improve on it.

Table 20: CRPS of TimeES with different input lengths (horizon 192, mean over 3 seeds). The default input length is 168. **Bold** indicates the best result.

Input length	24	72	96	168 (default)	336
ETTh1	0.356	**0.336**	0.339	0.349	0.369
ETTm1	0.398	0.321	**0.298**	0.310	0.299

J.4 Operating Range under Limited Data

TimeES estimates the evolutionary spectrum directly from data rather than assuming an analytic form, which underlies its gains in Table 3: it achieves the best CRPS on 8 of 9 datasets and the best QICE on 5 of 9 (Table 11). The same property also sets its boundary, since estimating a full time-frequency amplitude field requires sufficient samples. ILI is the extreme case in our benchmark: its 966 time steps yield only 473 training windows, compared with 17,420 time steps for ETTh1, and it is also where the variance across seeds is largest (a CRPS standard deviation of 0.172 over 3 seeds, compared with less than 0.01 on ETTh1). NsDiff [74] is less affected in this regime because its time-varying variance is estimated from a trailing sliding window over the observed series, which remains well defined with few samples and adapts freely to asymmetric residuals; the same flexibility explains its better decile-level QICE on ETTh1, ETTh2, and ETTm1. Learning evolutionary spectra under limited data is therefore an important direction for future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The assumptions and proofs are provided at Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information needed to reproduce the main experimental results at Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the link to code in the abstract.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experimental settings/details are discussed in Section 5 including all the information to reproduce the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We state in Appendix F about statistical significance. The statistical significance of probabilistic and deterministic is under 3%/1%, respectively.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The results are run in NVIDIA A6000 GTX GPU. Section D shows the computational cost of various methods.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses both potential positive societal impacts and negative societal impacts of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper has no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the model code in the abstract.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: the paper does not involve crowd-sourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.