# KOONPRO: A VARIANCE-AWARE KOOPMAN PROBABILISTIC MODEL ENHANCED BY NEURAL PROCESS FOR TIME SERIES FORECASTING

**Ronghua Zheng, Hanru Bai & Weiyang Ding** *
Institute of Science and Technology for Brain-inspired Intelligence
Fudan University, Shanghai 200433, China
rhzheng23@m.fudan.edu.cn,hrbai23@m.fudan.edu.cn,dingwy@fudan.edu.cn

## ABSTRACT

The probabilistic forecasting of time series is a well-recognized challenge, particularly in disentangling correlations among interacting time series and addressing the complexities of distribution modeling. By treating time series as temporal dynamics, we introduce **KooNPro**, a novel probabilistic time series forecasting model that combines variance-aware deep **Koo**pman model with **N**eural **Pro**cess. KooNPro introduces a variance-aware continuous spectrum using Gaussian distributions to capture complex temporal dynamics with improved stability. It further integrates the Neural Process to capture fine dynamics, enabling enhanced dynamics capture and prediction. Extensive experiments on nine real-world datasets demonstrate that KooNPro consistently outperforms state-of-the-art baselines. Ablation studies highlight the importance of the Neural Process component and explore the impact of key hyperparameters. Overall, KooNPro presents a promising novel approach for probabilistic time series forecasting. Code is available at this repository: https://github.com/Rrh-Zheng/Koonpro.

## 1 INTRODUCTION

Time series forecasting is essential in supply chain management, finance, energy, and healthcare. Probabilistic forecasting quantifies uncertainty, supporting robust decision-making in tasks like inventory optimization and risk management. For instance, in supply chains, it helps businesses anticipate demand fluctuations, optimize planning, and improve efficiency. Traditional methods like Auto-Regressive Integrated Moving Average (ARIMA) (Said & Dickey, 1984) and the family of Kalman filters (Auger et al., 2013) struggle with the complexities of real-world time series. Deep learning approaches for time series forecasting have evolved from Recurrent Neural Networks (RNNs) (Salehinejad et al., 2017) to Long Short-Term Memory (LSTM) networks (Yu et al., 2019). The introduction of the attention mechanism (Vaswani et al., 2017) further advanced the field, leading to Transformer-based models such as Zhou et al. (2021), Wu et al. (2021), and Zhou et al. (2022). More recently, diffusion probabilistic models (Ho et al., 2020; Rasul et al., 2021; Li et al., 2022; Fan et al., 2024) have emerged as a promising paradigm for probabilistic forecasting, with works like Kollovieh et al. (2024) demonstrating their potential. Moreover, foundation models for time series forecasting, such as Ansari et al. (2024), have introduced the concept of learning a universal representation for diverse time series tasks. Despite these advancements, many deep learning models focus primarily on point estimation, which limits their ability to fully capture the uncertainty and dynamics inherent in complex systems.

State space models, such as Rangapuram et al. (2018a), Wang et al. (2019) and Paria et al. (2021) offer an alternative by modeling time series as low-dimensional latent decompositions. These methods excel at representing temporal dynamics in reduced spaces but often rely on fixed assumptions about state transitions, which may fail to generalize to highly nonlinear and non-stationary scenarios. Dynamic Mode Decomposition (DMD) is a powerful tool for modeling time series as dynamic systems (Kuttichira et al., 2017), aiming to identify the underlying dynamics and subsequently

*Corresponding author

utilize them for prediction (Kou & Zhang, 2019; Yuan et al., 2021). However, DMD's reliance on discrete eigenvalues (point spectrum) restricts its ability to describe the nonlinear patterns of complex systems. The Koopman theory offers a powerful framework for analyzing nonlinear systems by representing them in a linear but infinite-dimensional space. The ability to decompose complex dynamics into Koopman eigenfunctions has inspired many works to incorporate this theory into time series modeling. Combining the Koopman theory (Koopman, 1931) with deep learning (Lusch et al., 2018) enhances the ability to model temporal dynamics with greater sophistication, leading to more accurate predictions. For example, models such as Liu et al. (2023) and Wang et al. (2023) enhance predictive capabilities through deep Koopman theory by learning the continuous spectrum of dynamics. However, these models are often affected by spectrum pollution, leading to unstable convergence and reduced accuracy, especially in highly nonlinear systems (Colbrook & Townsend, 2024).

To address this, the concept of pseudospectra has been introduced. Pseudospectra accounts for the latent spaces where eigenvalues may appear under different perturbation conditions, which makes them more robust to noise and better suited for capturing continuous variations. Inspired by the idea of perturbations in pseudospectra and drawing on the notion that probability distributions also characterize variables within a neighborhood by measuring their overall dispersion (Colbrook et al., 2024), we propose a variance-aware continuous spectrum and model it using Gaussian distributions. In this way, the perturbation of eigenvalues is not only determined by their local perturbations but also influenced by the Gaussian distribution over global dynamics. Moreover, to better capture the global patterns of dynamic systems, we utilize the latent representation of point spectra as a condition to guide the learning of dynamics, thereby obtaining the final variance-aware continuous spectrum distribution.

To enhance the proposed variance-aware continuous spectrum distribution, we integrate it into the Neural Process (NP) framework (Garnelo et al., 2018b), leveraging its probabilistic inference capabilities to capture point spectra globally. Specifically, we introduce KooNPro, a novel probabilistic approach that merges the global latent feature representation of point spectra from NP with variance-aware continuous spectra modeling, thereby improving the global modeling capability of dynamic systems. We employ variational inference to solve this new modeling framework and derive a new Evidence Lower Bound (ELBO) to optimize the combined model. By incorporating the variance-aware continuous spectrum, KooNPro maintains high predictive accuracy and robustness in high-dimension and non-stationary tasks, achieving state-of-the-art performance. Our contributions are summarized as follows:

- We introduce KooNPro, a novel probabilistic prediction model that synergistically integrates the probabilistic Koopman model with NP. Drawing inspiration from perturbations in pseudospectra, KooNPro incorporates a variance-aware continuous spectrum to learn temporal dynamics effectively. Additionally, it utilizes NP to capture the underlying global dynamics that govern the entire time series, facilitating improved predictive capabilities.
- Extensive experiments conducted on diverse real-world datasets convincingly demonstrate the superiority of our proposed model. It consistently achieves state-of-the-art performance, significantly outperforming existing methods across two metrics.
- We perform comprehensive ablation studies demonstrating the ability of NP to capture temporal dynamics globally, thereby enhancing prediction performance. We also assess the impact of key hyperparameters on model efficacy. Additionally, we provide a detailed case study that visually illustrates the prediction performance of KooNPro, yielding intuitive and interpretable results.

## 2 Related work

This work engages with three key areas: probabilistic time series prediction, the probabilistic Koopman model, and Neural Process. While each of these fields has received considerable attention, we limit our discussion to the most pertinent studies to ensure brevity. A more extensive version can be found in Appendix A.

**Probabilistic time series prediction:** Recent developments in probabilistic forecasting for time series integrate deep learning, statistical approaches, and diffusion models. Rangapuram et al. (2018b),

Salinas et al. (2020) and Li et al. (2021) combine state space model with deep learning. Feng et al. (2023) and Tang & Matteson (2021) developed attention-based mechanism that enhance long-range dependencies for improved forecast accuracy. Gaussian Process (GP), combined with temporal decomposition, was used by Yan et al. (2021), Nguyen & Quanz (2021), and Salinas et al. (2019) to better model uncertainty in multivariate settings. The data distribution-based generative model proposed by (Gouttes et al., 2021) and the Kalman filter-based approach proposed by de Bézenac et al. (2020), blend probabilistic techniques for better scalability and robustness. Diffusion models, such as those proposed by Rasul et al. (2021), Li et al. (2022), and Fan et al. (2024), model forecasting as a denoising task, excelling in high-dimensional settings.

**Probabilistic Koopman model:** The Koopman theory has been developed in lots of fields, the most related work is the probabilistic Koopman model, originally proposed by Morton et al. (2019). Han et al. (2022) designed a stochastic Koopman neural network for control, in which the latent observables are represented through Gaussian distribution. Colbrook et al. (2024) integrates the concept of variance into the Koopman framework by introducing variance-pseudospectra, thereby ensuring convergence within the model. For time series tasks, Naiman et al. (2023) utilized Koopman theory to represent the latent conditional prior dynamics via a linear map, while Mallen et al. (2024) introduced a framework enabling probabilistic forecasting for systems with periodically varying uncertainty.

**Neural process:** Neural Process (NP), first proposed by Garnelo et al. (2018a) as Conditional Neural Process, bridges neural networks' scalability and Gaussian Process's ability to model uncertainty. Garnelo et al. (2018b) provide flexible, probabilistic function approximations, making them efficient and adaptable across tasks. Attentive Neural Process (Kim et al., 2019), Convolutional Conditional Neural Process (Gordon et al., 2019b), Gaussian Neural Process (Bruinsma et al., 2021) and Autoregressive Neural Process (Bruinsma et al., 2023) deploy different methods to probe the relation between input-output pairs. Other works extend NP to meta-learning, such as Meta-Learning Stationary Stochastic Process (Foong et al., 2020) and Meta-Learning Probabilistic Inference (Gordon et al., 2019a). A comprehensive survey by Jha et al. (2022) has organized these developments, exploring their wide applications in uncertainty-aware learning.

## 3 BACKGROUND

Due to space constraints, we discuss the background details in Appendix B and focus here on the core of Koopman theory and Neural Process, which form the foundation of KooNPro.

The spectrum decomposition of the Koopman operator is crucial for understanding dynamics. The point spectrum reveals modal contributions via eigenvalues and corresponding eigenfunctions, which can reveal how different modes contribute to the system's evolution. However, many complex systems exhibit a continuous spectrum, reflecting chaotic or highly irregular behavior. In such cases, the Koopman operator's spectrum forms a continuum rather than isolated eigenvalues and offers a sophisticated framework for capturing both regular and chaotic components of system dynamics. Althrough the Koopman operator is effective for predictive modeling but is prone to spectrum pollution, which can hinder convergence and stability. Drawing inspiration from pseudospectra perturbations and the idea that probability distributions capture variable dispersion, we propose a variance-aware continuous spectrum modeled by a Gaussian distribution.

Neural Process (NP) is a stochastic process $p(f)$ that describes the predictive distribution over the target set $(\boldsymbol{x}_D, \boldsymbol{y}_D) := (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i \in D}$ given the context set $(\boldsymbol{x}_C, \boldsymbol{y}_C) := (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i \in C}$. Garnelo et al. (2018b) prompts use the distribution of a high-dimensional random vector $\boldsymbol{S}$ to represent $p(f)$ as

$$p(\boldsymbol{y}_D|\boldsymbol{x}_D, \boldsymbol{x}_C, \boldsymbol{y}_C) := \int p(\boldsymbol{y}_D|\boldsymbol{x}_D, \boldsymbol{S}) p(\boldsymbol{S}|\boldsymbol{x}_C, \boldsymbol{y}_C) d\boldsymbol{S}. \tag{1}$$

NP consists of two components: an encoder that maps the input-output pairs of the context set $(\boldsymbol{x}_C, \boldsymbol{y}_C)$ to $\boldsymbol{S}$ for representing $p(f)$, and a decoder that combines $\boldsymbol{S}$ with $\boldsymbol{x}_D$ to generate $\boldsymbol{y}_D$. Due to the intractable log-likelihood, NP adopts amortized variational inference and maximizes the evidence lower bound (ELBO) of the log-likelihood as follows

$$\log\left(\boldsymbol{y}_D|\boldsymbol{x}_D, \boldsymbol{y}_C, \boldsymbol{x}_C\right) \geq \mathbb{E}_{\boldsymbol{S} \sim q(\boldsymbol{S}|\boldsymbol{x}_D, \boldsymbol{y}_D)}\left[\log p\left(\boldsymbol{y}_D|\boldsymbol{x}_D, \boldsymbol{S}\right)\right] - D_{KL}\left(q\left(\boldsymbol{S}|\boldsymbol{x}_D, \boldsymbol{y}_D\right) || p\left(\boldsymbol{S}|\boldsymbol{x}_C, \boldsymbol{y}_C\right)\right). \tag{2}$$

# 4 METHOD

This section offers a detailed overview of KooNPro, with the complete architecture illustrated in Fig.1. The central idea of KooNPro is to learn temporal dynamics for probabilistic future prediction by integrating Neural Process (NP) with a probabilistic deep Koopman model. Initially, NP globally captures the point spectra of dynamics to govern the entire time series, as indicated by the downward arrows in Fig.1. Additionally, inspired by the concept of pseudospectra, we utilize a probabilistic deep Koopman model to refine these dynamics, obtaining a variance-aware continuous spectrum for prediction as demonstrated by the shadowed box in Fig.1.

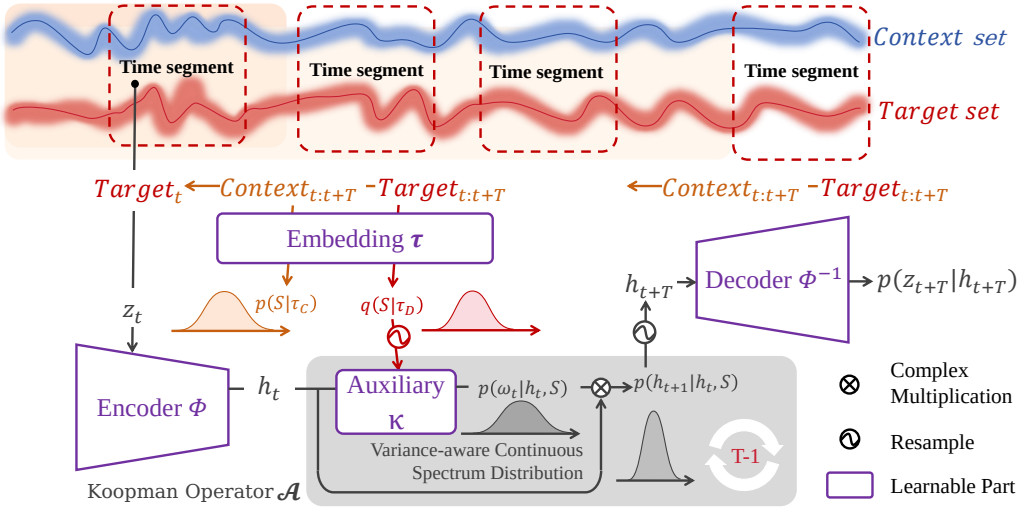


Figure 1: **The complete architecture of KooNPro:** $p(\boldsymbol{S}|\tau_D)$ and $p(\boldsymbol{S}|\tau_C)$ are the latent representations of DMD mean over the target set and the context set respectively. Given the first time step $\boldsymbol{z}_1$ in the target set, the encoder $\phi$ generates the latent variable $\boldsymbol{h}_1$. We apply the Koopman operator $\mathcal{A}$ to $\boldsymbol{h}_1$ for $T-1$ steps then generate the distribution of $\boldsymbol{h}_{2:T}$. Finally, the decoder $\phi^{-1}$ maps $\boldsymbol{h}_{2:T}$ back to the original space to generate the distribution of $\boldsymbol{z}_{2:T}$.

## 4.1 CAPTURE TEMPORAL DYNAMICS BY NEURAL PROCESS

Dynamic Mode Decomposition (DMD) is a classical method that elucidates the relationship between time series data and corresponding dynamics. However, DMD's reliance on point spectra limits its capacity to capture the nonlinear patterns inherent in complex systems, thereby constraining its predictive power. To address these limitations, KooNPro integrates Neural Process (NP) with DMD to generate a stochastic process that captures the dynamics of the whole time series more effectively. Specifically, KooNPro identifies the distribution of the latent variable $\boldsymbol{S} \in \mathbb{R}^s$, which integrates the underlying dynamics of a time series in Eq.1, as shown in the left part of Fig.1.

We utilize Takens' theory like Yuan et al. (2021), define a time series $\boldsymbol{z}_{1:T} \in \mathbb{R}^{T\times d\times k}$, where $T$ denotes the time length, $d$ denotes the feature dimension and $k$ denotes the delay-embedded length, and let $\boldsymbol{x} = \boldsymbol{z}_{1:T-1}$, $\boldsymbol{y} = \boldsymbol{z}_{2:T}$. To estimate the dynamics governing time series in the context set, KooNPro employs a neural network $\boldsymbol{\tau}$ to embed $\hat{\boldsymbol{x}} \in \mathbb{R}^{(T-1)\times d\times 1}$ and $\hat{\boldsymbol{y}} \in \mathbb{R}^{(T-1)\times d\times 1}$ as the initial components of delay embedding like

$$\tau_C := \boldsymbol{\tau}(\boldsymbol{x}_C, \boldsymbol{y}_C) = \boldsymbol{\psi}\left(\frac{1}{c}\sum_{i=1}^{c} vec\left(\hat{\boldsymbol{x}}^{\dagger}\hat{\boldsymbol{y}}\right)\right), \tag{3}$$

where $\boldsymbol{\psi}$ denotes a learnable Multi-Layer Perceptron (MLP), $c$ denotes the number of items in the context set, and the eigenvalues of $\hat{\boldsymbol{x}}^{\dagger}\hat{\boldsymbol{y}}$ can approximate the point spectrum of the corresponding time series dynamic. We model the distribution of $\boldsymbol{S}$ by a factorized Gaussian distribution parametrized by $\tau_C$, i.e.,

$$p(\boldsymbol{S}|\boldsymbol{x}_C, \boldsymbol{y}_C) = \mathcal{N}(\boldsymbol{S}; \mu(\tau_C), \sigma(\tau_C)). \tag{4}$$

Denote $p(\boldsymbol{S}|\tau_C) := p(\boldsymbol{S}|x_C, y_C)$ for any set $C$, thus $p(\boldsymbol{S})$ represents the distribution of underlying dynamics according to different time series sets. The ELBO in Eq.2 can be reformulated as follows

$$\log(\boldsymbol{y}_D|\boldsymbol{x}_D, \boldsymbol{y}_C, \boldsymbol{x}_C) \geq \mathbb{E}_{\boldsymbol{S}\sim q_{(\boldsymbol{S}|\tau_D)}}[\log p(\boldsymbol{y}_D|\boldsymbol{x}_D, \boldsymbol{S})] - D_{KL}(q(\boldsymbol{S}|\tau_D)||p(\boldsymbol{S}|\tau_C)). \tag{5}$$

The second term on the right of Eq.5 captures the difference in the distribution of $\boldsymbol{S}$, reflecting the diversity of temporal dynamics across different time series. Although we employ $\boldsymbol{\tau}$ to embed the point spectrum that characterizes temporal dynamics, it provides only a coarse approximation. For the first term on the right side of Eq.5, we maximize it using a probabilistic deep Koopman model, which provides a more nuanced representation of temporal dynamics through the lens of the pseudospectra.

### 4.2 Probabilistic deep Koopman model

The proposed probabilistic deep Koopman model shown in the shadowed box of Fig.1 concentrates on explaining local characteristics of time series with a variance-aware continuous spectrum. This approach offers a deeper understanding of temporal dynamics behavior over time, leading to improved predictive capabilities. In essence, estimating the true temporal dynamics is equivalent to maximizing the expectation $\mathbb{E}_{q_{(\boldsymbol{S}|\tau_D)}}[\log p(\boldsymbol{y}_D|\boldsymbol{x}_D, \boldsymbol{S})]$.

For simplicity, we discuss only the data in the target set and omit the index $D$. Following Lusch et al. (2018), we compose KooNPro by three MLPs: an **encoder** $\boldsymbol{\phi} : \mathbb{R}^{d\times k} \to \mathbb{R}^n$ and a **decoder** $\boldsymbol{\phi}^{-1} : \mathbb{R}^n \to \mathbb{R}^{d\times k}$ to identify an appropriate linear space, along with an **auxiliary** $\boldsymbol{\kappa} : \mathbb{R}^{n+s} \to \mathbb{R}^n$ to learn the continuous spectra of the corresponding temporal dynamics in linear space. Firstly, KooNPro vectorized the delay-embedded time series $\boldsymbol{x} = \boldsymbol{z}_{1:T-1}$ from size $(T-1) \times d \times k$ to $(T-1) \times (d \cdot k) \times 1$, then applies the encoder as $\boldsymbol{h}_{1:T-1} = \boldsymbol{\phi}(\boldsymbol{x})$. We hypothesize the latent space created by $\boldsymbol{\phi}$ possesses linear characteristics, thus the dynamics in such space can be described as follows

$$\boldsymbol{h}_{t+1} = e^{\boldsymbol{\lambda}_t \Delta t}\boldsymbol{h}_t. \tag{6}$$

We assume $\boldsymbol{h}_t$ represents the eigenvalue of the dynamics, consisting of a pair of conjugate numbers, and $\boldsymbol{\lambda}_t$ to be pure imaginary like Lange et al. (2021), let $\boldsymbol{\lambda}_t = j\boldsymbol{\omega}_t$, thus we have

$$\mathbf{Re}(\boldsymbol{h}_{t+1}) = \mathbf{Re}(\boldsymbol{h}_t) \odot \cos(\boldsymbol{\omega}_t \Delta t) - \mathbf{Im}(\boldsymbol{h}_t) \odot \sin(\boldsymbol{\omega}_t \Delta t), \tag{7}$$

$$\mathbf{Im}(\boldsymbol{h}_{t+1}) = \mathbf{Re}(\boldsymbol{h}_t) \odot \sin(\boldsymbol{\omega}_t \Delta t) + \mathbf{Im}(\boldsymbol{h}_t) \odot \cos(\boldsymbol{\omega}_t \Delta t), \tag{8}$$

where $\odot$ is the element-wise product. We can simplify Eq.7 and Eq.8 to

$$\boldsymbol{h}_{t+1} := \mathcal{A}(\boldsymbol{h}_t, \boldsymbol{\omega}_t). \tag{9}$$

In order to learn the Koopman operator $\mathcal{A}$, which is conditioned on the global temporal dynamics representation $\boldsymbol{S}$, KooNPro utilizes the auxiliary $\boldsymbol{\kappa}$ to project the concatenation of $\boldsymbol{h}_t$ and $\boldsymbol{S}$ to $\kappa_t$,

$$\boldsymbol{\kappa} : \begin{bmatrix} \boldsymbol{h}_t \\ \boldsymbol{S} \end{bmatrix} \mapsto \kappa_t. \tag{10}$$

Drawing on the theory of pseudospectra, KooNPro aims to learn dynamic information not only from the eigenvalues but also from their neighborhoods. To facilitate this, KooNPro employs the auxiliary network to generate the corresponding Gaussian distribution of $\boldsymbol{\omega}_t$ parameterized by $\kappa_t$ as

$$p(\boldsymbol{\omega}_t|\boldsymbol{h}_t, \boldsymbol{S}) = \mathcal{N}(\boldsymbol{\omega}_t; \mu(\kappa_t), \sigma(\kappa_t)). \tag{11}$$

To simplify KooNPro, we assume the independence of $\boldsymbol{h}_{1:T-1}$ on the linear space. Consequently, we can describe the uncertain dynamics as

$$p(\boldsymbol{\omega}_{1:T-1}|\boldsymbol{h}_{1:T-1}, \boldsymbol{S}) = \prod_{t=1}^{T-1} \mathcal{N}(\boldsymbol{\omega}_t; \mu(\kappa_t), \sigma(\kappa_t)). \tag{12}$$

Since $\boldsymbol{h}_t = \mathcal{A}(\boldsymbol{h}_{t-1}, \boldsymbol{\omega}_{t-1})$, it yields that $p(\boldsymbol{h}_T|\boldsymbol{\omega}_{1:T-1}, \boldsymbol{h}_{1:T-1}, \boldsymbol{S}) = 1$. Thus, the variance-aware continuous spectrum in the linear space can be characterized within the framework of the Gaussian distribution like

$$p(\boldsymbol{h}_{2:T}|\boldsymbol{h}_{1:T-1}, \boldsymbol{S}) = p(\boldsymbol{h}_T|\boldsymbol{\omega}_{1:T-1}, \boldsymbol{h}_{1:T-1}, \boldsymbol{S})p(\boldsymbol{\omega}_{1:T-1}|\boldsymbol{h}_{1:T-1}, \boldsymbol{S}) \tag{13}$$

$$= \prod_{t=1}^{T-1} \mathcal{N}(\boldsymbol{\omega}_t; \mu(\kappa_t), \sigma(\kappa_t)). \tag{14}$$

Utilizing the decoder $\boldsymbol{\phi}^{-1}$, KooNPro maps the temporal dynamic evolution from the linear space back to the original space, as follows

$$p(\boldsymbol{z}_{2:T}|\boldsymbol{h}_{2:T}) = \prod_{t=2}^{T} \mathcal{N}(\boldsymbol{z}_t; \mu(\boldsymbol{\phi}^{-1}(\boldsymbol{h}_t)), \sigma(\boldsymbol{\phi}^{-1}(\boldsymbol{h}_t))). \tag{15}$$

The likelihood $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{S})$ in Eq.5, where $\boldsymbol{x} = \boldsymbol{z}_{1:T-1}$ and $\boldsymbol{y} = \boldsymbol{z}_{2:T}$, can be derived as

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{S}) = p(\boldsymbol{z}_{2:T}|\boldsymbol{h}_{1:T-1}, \boldsymbol{S}) = p(\boldsymbol{z}_{2:T}|\boldsymbol{h}_{2:T})p(\boldsymbol{h}_{2:T}|\boldsymbol{h}_{1:T-1}, \boldsymbol{S}), \tag{16}$$

given that the encoder $\boldsymbol{\phi}$ is a deterministic function. Insert Eq.14 and Eq.15 to Eq.16, we can draw the expectation as

$$\mathbb{E}_{\boldsymbol{S}\sim q_{(\boldsymbol{S}|\tau_D)}}[\log p(\boldsymbol{y}_D|\boldsymbol{x}_D, \boldsymbol{S})] = \mathbb{E}_{\boldsymbol{S}\sim q_{(\boldsymbol{S}|\tau_D)}}[\sum_{t=2}^{T} \log(p(\boldsymbol{z}_t|\boldsymbol{h}_t)) + \sum_{t=1}^{T-1} \log(p(\boldsymbol{h}_{t+1}|\boldsymbol{h}_t, \boldsymbol{S}))]. \tag{17}$$

The first term can be interpreted as the prediction loss, while the second term represents the linear loss, as described in Lusch et al. (2018). By maximizing the ELBO in Eq.5, we derive $\boldsymbol{S}_C$ approximates the dynamics for each time series present during the training phase. Consequently, our predictions in the test phase commence with the initial $\boldsymbol{z}_1 \in \mathbb{R}^{1\times d\times k}$ like

$$\boldsymbol{z}_{\tilde{T}} = \boldsymbol{\phi}^{-1}(\mathcal{A}^{\tilde{T}-1}(\boldsymbol{\phi}(\boldsymbol{z}_1), \boldsymbol{S}_C)). \tag{18}$$

It is crucial to note that the time length $T$ during the training stage is distinct from the prediction horizon $\tilde{T}$, with the input history length solely determined by the delay-embedded length $k$. This enables us to train KooNPro once and then forecast future values of arbitrary length.

## 5 EXPERIMENTS

In this section, we conduct comprehensive experiments on nine real-world datasets to evaluate the performance of KooNPro against state-of-the-art baselines. We demonstrate Neural Process's ability to enhance KooNPro's predictive performance and investigate the impact of key hyperparameters through ablation studies. Finally, we visualize the prediction results for the `Solar` dataset and analyze the relationship between prediction error and variance.

### 5.1 SETTINGS

**Datasets.** We consider nine real-world datasets characterized by a range of temporal dynamics, namely `ETTs`, `Solar`, `Electricity`, `Traffic`, `Taxi`, and `KDD-cup`. The data is recorded at intervals of 15 minutes, 30 minutes, 1 hour, or 1 day frequencies. Refer to Appendix C for details. All datasets are split chronologically and adopt the same train/validation/test ratios, *i.e.*, 7:1:2.

**Evaluation Metrics.** Following previous work (Fan et al., 2024), we assess our model and all baselines using $\mathbf{CRPS_{sum}}$ (Continuous Ranked Probability Score), a widely used metric for probabilistic time series forecasting, as well as $\mathbf{NRMSE_{sum}}$ (Normalized Root Mean Squared Error). The details of metrics are shown in Appendix D.

**Baselines.** We assess the predictive performance of KooNPro in comparison with multivariate time series forecasting models, including GP-Copula (Salinas et al., 2019), Transformer-MAF (Rasul et al., 2021), TimeGrad (Rasul et al., 2021), TACTiS (Ashok et al., 2023), $D^3$VAE (Li et al., 2023), DPK (Mallen et al., 2024), and MG-TSD (Fan et al., 2024). The more details of baselines can be found in Appendix E.

**Implementation details.** The training process is early stopped within 5 epochs using the Adam optimizer with a fixed learning rate of $10^{-5}$. The mini-batch size is set to 128. Additional hyperparameters, such as time length $T$, delay-embedded length $k$, and layers of MLPs are detailed in Appendix F. All models are trained and tested on a single NVIDIA RTX4070Ti 12GB GPU.

### 5.2 RESULTS

In this section, we compare the performance of KooNPro with baselines using $\mathbf{CRPS_{sum}}$ and $\mathbf{NRMSE_{sum}}$. More comprehensive results, including long-term predictions (Appendix G), performance in different noisy scenarios (Appendix H), and the relationship between performance and memory consumption (Appendix I), can be found in the appendix.

Table 1: Comparison of $\mathbf{CRPS_{sum}}$ (denoted as C-s, smaller is better) and $\mathbf{NRMSE_{sum}}$ (denoted as N-s, smaller is better) across nine real-world datasets. The means and standard errors are based on 10 independent runs of retraining and evaluation. The best performances are highlighted in red and the second are in blue.

| Model | Metric | ETTh1 | ETTh2 | ETTm1 | ETTm2 | Solar | Electricity | Traffic | Taxi | Cup |
|---|---|---|---|---|---|---|---|---|---|---|
| GP-Copula | C-s | $0.537_{\pm 0.019}$ | $0.264_{\pm 0.023}$ | $0.241_{\pm 0.043}$ | $0.147_{\pm 0.019}$ | $0.305_{\pm 0.024}$ | $0.078_{\pm 0.035}$ | $0.199_{\pm 0.008}$ | $0.286_{\pm 0.066}$ | $0.217_{\pm 0.000}$ |
| | N-s | $0.835_{\pm 0.032}$ | $0.383_{\pm 0.032}$ | $0.424_{\pm 0.088}$ | $0.172_{\pm 0.027}$ | $0.671_{\pm 0.034}$ | $0.122_{\pm 0.058}$ | $0.293_{\pm 0.018}$ | $0.412_{\pm 0.115}$ | $0.346_{\pm 0.010}$ |
| Trans-MAF | C-s | $0.800_{\pm 0.049}$ | $0.223_{\pm 0.012}$ | $0.379_{\pm 0.029}$ | $0.228_{\pm 0.040}$ | $0.964_{\pm 0.013}$ | $0.144_{\pm 0.046}$ | $0.477_{\pm 0.021}$ | $0.403_{\pm 0.046}$ | $0.257_{\pm 0.013}$ |
| | N-s | $1.285_{\pm 0.188}$ | $0.315_{\pm 0.024}$ | $0.577_{\pm 0.082}$ | $0.356_{\pm 0.065}$ | $1.665_{\pm 0.022}$ | $0.245_{\pm 0.079}$ | $0.762_{\pm 0.031}$ | $0.598_{\pm 0.029}$ | $0.390_{\pm 0.010}$ |
| Timegrid | C-s | $0.547_{\pm 0.022}$ | $0.241_{\pm 0.002}$ | $0.227_{\pm 0.029}$ | $0.212_{\pm 0.015}$ | $0.594_{\pm 0.011}$ | $0.044_{\pm 0.010}$ | $0.455_{\pm 0.020}$ | $0.327_{\pm 0.058}$ | $0.271_{\pm 0.087}$ |
| | N-s | $0.889_{\pm 0.039}$ | $0.325_{\pm 0.000}$ | $0.363_{\pm 0.072}$ | $0.291_{\pm 0.018}$ | $1.081_{\pm 0.015}$ | $0.072_{\pm 0.014}$ | $0.560_{\pm 0.016}$ | $0.498_{\pm 0.095}$ | $0.341_{\pm 0.097}$ |
| TACTIS | C-s | $0.601_{\pm 0.004}$ | $0.208_{\pm 0.002}$ | $0.634_{\pm 0.009}$ | $0.142_{\pm 0.015}$ | $1.871_{\pm 0.022}$ | $0.254_{\pm 0.012}$ | $0.456_{\pm 0.003}$ | $0.981_{\pm 0.013}$ | $0.276_{\pm 0.008}$ |
| | N-s | $0.907_{\pm 0.010}$ | $0.320_{\pm 0.006}$ | $1.013_{\pm 0.010}$ | $0.320_{\pm 0.006}$ | $2.309_{\pm 0.528}$ | $0.391_{\pm 0.016}$ | $1.871_{\pm 0.003}$ | $1.170_{\pm 0.013}$ | $0.403_{\pm 0.013}$ |
| $D^3$VAE | C-s | $0.445_{\pm 0.023}$ | $0.266_{\pm 0.016}$ | $0.219_{\pm 0.009}$ | $0.177_{\pm 0.017}$ | $0.312_{\pm 0.035}$ | $0.198_{\pm 0.020}$ | $0.265_{\pm 0.027}$ | $0.257_{\pm 0.008}$ | $0.243_{\pm 0.035}$ |
| | N-s | $0.662_{\pm 0.029}$ | $0.479_{\pm 0.027}$ | $0.257_{\pm 0.058}$ | $0.263_{\pm 0.007}$ | $0.642_{\pm 0.070}$ | $0.253_{\pm 0.104}$ | $0.926_{\pm 0.103}$ | $0.391_{\pm 0.038}$ | $0.501_{\pm 0.005}$ |
| DPK | C-s | $0.718_{\pm 0.011}$ | $0.471_{\pm 0.024}$ | $0.556_{\pm 0.018}$ | $0.341_{\pm 0.062}$ | $0.753_{\pm 0.035}$ | $0.784_{\pm 0.008}$ | $0.827_{\pm 0.007}$ | $0.843_{\pm 0.009}$ | $0.728_{\pm 0.029}$ |
| | N-s | $1.026_{\pm 0.012}$ | $0.725_{\pm 0.040}$ | $0.887_{\pm 0.023}$ | $0.391_{\pm 0.157}$ | $1.130_{\pm 0.040}$ | $1.062_{\pm 0.005}$ | $1.160_{\pm 0.007}$ | $1.165_{\pm 0.010}$ | $1.147_{\pm 0.056}$ |
| MG-TSD | C-s | $0.430_{\pm 0.038}$ | $0.174_{\pm 0.009}$ | $0.254_{\pm 0.054}$ | $0.129_{\pm 0.009}$ | $0.298_{\pm 0.025}$ | $0.107_{\pm 0.055}$ | $0.528_{\pm 0.057}$ | $0.250_{\pm 0.073}$ | $0.323_{\pm 0.015}$ |
| | N-s | $0.693_{\pm 0.083}$ | $0.220_{\pm 0.017}$ | $0.394_{\pm 0.076}$ | $0.292_{\pm 0.046}$ | $0.623_{\pm 0.026}$ | $0.155_{\pm 0.063}$ | $0.710_{\pm 0.058}$ | $0.347_{\pm 0.078}$ | $0.638_{\pm 0.056}$ |
| KooNPro | C-s | $0.328_{\pm 0.037}$ | $0.149_{\pm 0.051}$ | $0.165_{\pm 0.057}$ | $0.081_{\pm 0.020}$ | $0.211_{\pm 0.033}$ | $0.057_{\pm 0.006}$ | $0.184_{\pm 0.022}$ | $0.226_{\pm 0.041}$ | $0.204_{\pm 0.017}$ |
| | N-s | $0.520_{\pm 0.045}$ | $0.224_{\pm 0.065}$ | $0.225_{\pm 0.028}$ | $0.122_{\pm 0.034}$ | $0.313_{\pm 0.044}$ | $0.095_{\pm 0.012}$ | $0.289_{\pm 0.025}$ | $0.330_{\pm 0.078}$ | $0.308_{\pm 0.030}$ |

Tab.1 presents the $\mathbf{CRPS_{sum}}$ and $\mathbf{NRMSE_{sum}}$ values, averaged over 10 independent runs. In comparison, traditional baseline models (e.g., GP-Copula and Timegrid) have higher $\mathbf{CRPS_{sum}}$ values on most datasets, which indicates their difficulty in handling complex time series data. Compared to modern deep generative models (e.g., $D^3$VAE and TACTIS), KooNPro achieves a better balance between predictive stability and accuracy. Although MG-TSD shows a slight advantage in $\mathbf{NRMSE_{sum}}$ on certain datasets (e.g., `ETTs`), its $\mathbf{CRPS_{sum}}$ fluctuates significantly, suggesting an inadequate characterization of distributional uncertainty. KooNPro introduces the variance-aware spectrum to enhance global pattern capturing, avoiding spectrum instability caused by local perturbations, thus achieving higher stability and accuracy in handling complex nonlinear patterns. This demonstrates its effectiveness in high-dimensional time series forecasting.

### 5.3 Ablation Study

In this section, we conduct an ablation study to identify the factors contributing to KooNPro's success in prediction tasks. We first examine the role of the Neural Process (NP), which primarily learns and represents the underlying dynamics across whole time series within $\boldsymbol{S}$. We track the KL divergence between $\boldsymbol{S}_D$ and $\boldsymbol{S}_C$ during the training phase (Fig. 2, upper) on the `ETTh1` dataset. It decreases over time, converging with the validation set around the 70th epoch, indicating effective learning of time series dynamics. To assess generalization, we compute the KL divergence between $\boldsymbol{S}_C$ and 150 test time series (Fig. 2, lower). The low divergence suggests $\boldsymbol{S}_C$ reliably captures temporal dynamics, even for previously unseen data.

To evaluate the contribution of NP further, we compare different methods for generating $\boldsymbol{S}_C$, including cases without it. For clarity, we label the methods as follows: **With-ANP** (using Attention Neural Process), **With-GP** (using GP), and **Without-NP** (no NP). Implementation details are provided in Appendix J. As shown in Tab.2, the overall performance of **With-ANP** is comparable to KooNPro, with predictive performance improving as data dimensionality increases. Notably, on the `taxi` dataset, the highest-dimensional dataset, **With-ANP** outperforms KooNPro on both metrics. However, ANP incurs higher computational complexity, increasing from O(n+m) for NP to O(n(n+m)) (Kim et al., 2019). **With-GP** demonstrates significantly degraded performance, even worse than **Without-NP**. This phenomenon may be attributed to the critical influence of prior knowledge in defining

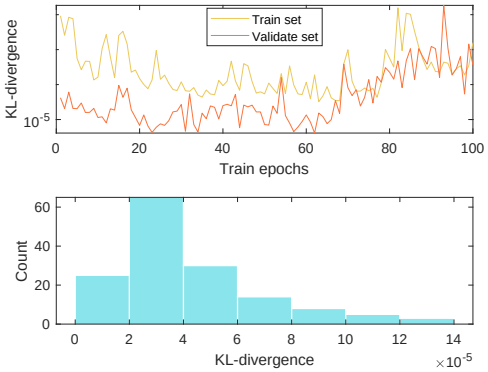


Figure 2: The upper section illustrates the KL-divergence between $\boldsymbol{S}_D$ and $\boldsymbol{S}_C$ during the training phase, with epochs of training stages on the x-axis and an exponential scale on the y-axis. The lower section analyzes 150 test dataset time segments, displaying their KL-divergence with $\boldsymbol{S}_C$, with the x-axis using an exponential scale and the y-axis showing time series counts.

Table 2: Prediction performance when replacing Neural Process (NP) with Attention Neural Process (ANP), Gaussian Process (GP), or removing NP version (denoted as with-ANP, with-GP, and without-NP, respectively).

| Model | Metric | ETTh1 | ETTh2 | ETTm1 | ETTm2 | Solar | Elec. | Traffic | Taxi | Cup |
|---|---|---|---|---|---|---|---|---|---|---|
| KooNPro | C-s | $0.328_{\pm 0.037}$ | $0.149_{\pm 0.051}$ | $0.165_{\pm 0.057}$ | $0.081_{\pm 0.020}$ | $0.211_{\pm 0.033}$ | $0.057_{\pm 0.006}$ | $0.184_{\pm 0.022}$ | $0.226_{\pm 0.041}$ | $0.204_{\pm 0.017}$ |
| | N-s | $0.520_{\pm 0.045}$ | $0.224_{\pm 0.065}$ | $0.225_{\pm 0.028}$ | $0.122_{\pm 0.034}$ | $0.313_{\pm 0.044}$ | $0.095_{\pm 0.012}$ | $0.289_{\pm 0.025}$ | $0.330_{\pm 0.078}$ | $0.308_{\pm 0.030}$ |
| with-ANP | C-s | $0.334_{\pm 0.061}$ | $0.194_{\pm 0.063}$ | $0.195_{\pm 0.029}$ | $0.107_{\pm 0.019}$ | $0.201_{\pm 0.062}$ | $0.067_{\pm 0.022}$ | $0.197_{\pm 0.016}$ | $0.213_{\pm 0.056}$ | $0.306_{\pm 0.008}$ |
| | N-s | $0.541_{\pm 0.084}$ | $0.300_{\pm 0.093}$ | $0.350_{\pm 0.061}$ | $0.182_{\pm 0.033}$ | $0.371_{\pm 0.052}$ | $0.127_{\pm 0.033}$ | $0.308_{\pm 0.030}$ | $0.314_{\pm 0.081}$ | $0.497_{\pm 0.008}$ |
| with-GP | C-s | $0.912_{\pm 0.067}$ | $0.613_{\pm 0.056}$ | $0.579_{\pm 0.023}$ | $0.438_{\pm 0.067}$ | $0.304_{\pm 0.052}$ | $0.125_{\pm 0.033}$ | $0.231_{\pm 0.023}$ | $0.349_{\pm 0.029}$ | $0.413_{\pm 0.159}$ |
| | N-s | $1.210_{\pm 0.061}$ | $0.791_{\pm 0.058}$ | $0.861_{\pm 0.029}$ | $0.622_{\pm 0.075}$ | $0.467_{\pm 0.067}$ | $0.205_{\pm 0.052}$ | $0.388_{\pm 0.016}$ | $0.501_{\pm 0.033}$ | $0.655_{\pm 0.252}$ |
| without-NP | C-s | $0.390_{\pm 0.083}$ | $0.218_{\pm 0.120}$ | $0.332_{\pm 0.068}$ | $0.155_{\pm 0.042}$ | $0.341_{\pm 0.059}$ | $0.093_{\pm 0.027}$ | $0.313_{\pm 0.051}$ | $0.252_{\pm 0.068}$ | $0.674_{\pm 0.224}$ |
| | N-s | $0.609_{\pm 0.117}$ | $0.318_{\pm 0.169}$ | $0.511_{\pm 0.103}$ | $0.243_{\pm 0.056}$ | $0.531_{\pm 0.065}$ | $0.144_{\pm 0.036}$ | $0.590_{\pm 0.095}$ | $0.380_{\pm 0.092}$ | $1.058_{\pm 0.341}$ |

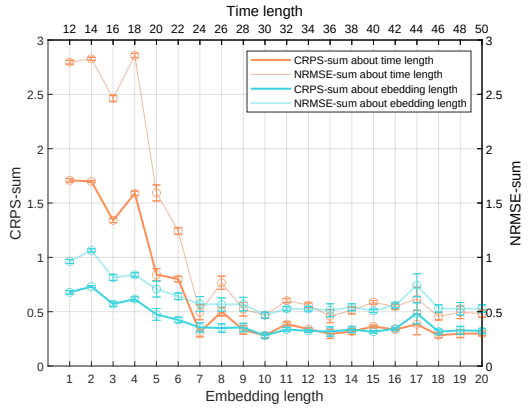


Figure 3: The top axis shows the increase in time length $T$, while the bottom axis indicates the increase in delay-embedded length $k$. The orange and transparent orange curves represent the $\mathbf{CRPS_{sum}}$ and $\mathbf{NRMSE_{sum}}$ for increasing $T$. The blue and transparent blue curves correspond to these metrics for increasing $k$. Error bars reflect the standard error from 10 independent retraining and evaluation runs.

the kernel function. Furthermore, the non-learnable parameters may impede $\boldsymbol{S}_D$ from adequately capturing the temporal dynamics governing the entire time series.

Next, we investigate the impact of the hyperparameter time length $T$ and delay-embedded length $k$ on the prediction performance. In Fig.3, we analyze the impact of hyperparameters on the `ETTh1` dataset. The orange and transparent orange curves illustrate changes in $\mathbf{CRPS_{sum}}$ and $\mathbf{NRMSE_{sum}}$ as $T$ increases. We observe that KooNPro's prediction accuracy improves significantly with $T$ until $T = 30$, beyond which further increases yield no additional benefits despite higher computational costs. A similar trend is seen when varying $k$, as shown by the blue and transparent blue curves. This suggests that temporal dynamics in time series cannot be fully captured simply by extending the learning time segment. Additional ablation studies across various datasets are presented in Appendix K.

### 5.4 Case Study

The predictive capability of KooNPro arises from its ability to capture the underlying dynamics of time series data effectively, we illustrate this using the `Solar` dataset, which comprises hourly measurements from 137 solar plants located in Alabama state. The dataset exhibits a clear diurnal pattern: values are non-zero from 6:00 to 18:00 and zero at night. In probabilistic prediction, large errors often correspond to high variance, indicating uncertainty. We assess this by analyzing the correlation between prediction variance and accuracy, measured by the mean absolute error (MAE) of the predicted mean relative to the true value. A scatter plot of MAE versus prediction variance (Fig.4, left) shows a correlation of $0.94$ with a significance level near zero, confirming that KooNPro effectively captures temporal dynamics and ensures reliable predictions. The right panel visualizes the ground truth and predictions for the first eight plants over 24 hours. The results demonstrate that KooNPro accurately captures the temporal fluctuations in `Solar` energy generation: The prediction mean approaches zero during the transition from nighttime to daylight and vice versa for each plant. Furthermore, the prediction intervals expand during sunlight and night periods, aligning with the peaks and troughs of `Solar` fluctuations, indicating the reliability of the predictions.

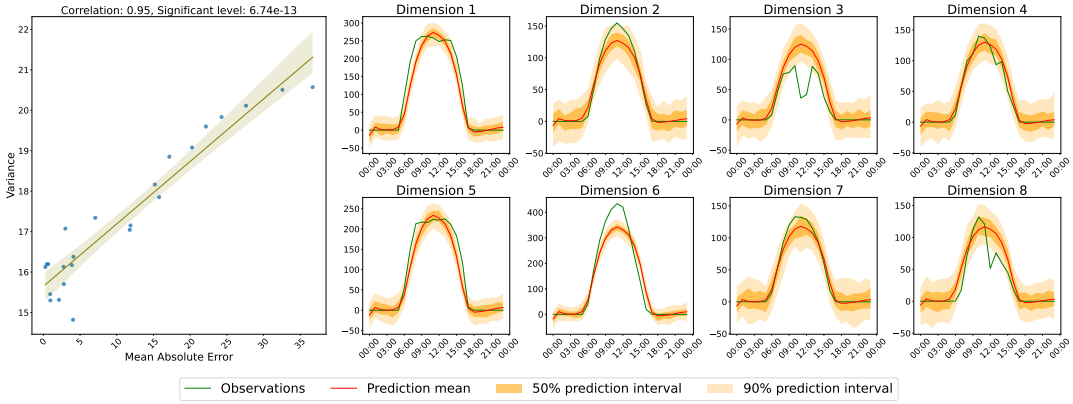


Figure 4: The left panel illustrates the correlation between the MAE of the prediction means relative to the true values and the prediction variance. The title indicates that the correlation achieves $0.95$ and the significant level is $6.74 \times 10^{-13}$. The right panel visualizes the changes in the first eight dimensions of the Solar dataset over 24 hours.

## 6 Conclusion

This paper presents KooNPro, a novel probabilistic forecasting model that treats time series as temporal dynamics. KooNPro integrates Neural Process for global dynamics modeling and a variance-aware continuous spectrum inspired by pseudospectra, enhancing learning capabilities. Extensive experiments on nine real-world datasets demonstrate KooNPro's superior performance compared to state-of-the-art methods. Comprehensive ablation studies explore the origins of KooNPro's predictive power, while visualizations of `Solar` dataset predictions showcase its accuracy and reliability.

## 7 ACKNOWLEDGEMENTS

This work was partially supported by the National Natural Science Foundation of China (No. 12471481, U24A2001) and the Science and Technology Commission of Shanghai Municipality (No. 23ZR1403000).

## REFERENCES


Ian Abraham, Gerardo De La Torre, and Todd D Murphey. Model-based control using koopman operators. *arXiv preprint arXiv:1709.01568*, 2017.

Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, et al. Gluonts: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116):1–6, 2020.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Nicolas Chapados, and Alexandre Drouin. Tactis-2: Better, faster, simpler attentional copulas for multivariate time series. *arXiv preprint arXiv:2310.01327*, 2023.

François Auger, Mickael Hilairet, Josep M Guerrero, Eric Monmasson, Teresa Orlowska-Kowalska, and Seiichiro Katsura. Industrial applications of the kalman filter: A review. *IEEE Transactions on Industrial Electronics*, 60(12):5458–5471, 2013.

Omri Azencot, Wotao Yin, and Andrea Bertozzi. Consistent dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 18(3):1565–1585, 2019.

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

Wessel P. Bruinsma, James Requeima, Andrew Y. K. Foong, Jonathan Gordon, and Richard E. Turner. The Gaussian Neural Process, January 2021.

Wessel P Bruinsma, Stratis Markou, James Requiema, Andrew YK Foong, Tom R Andersson, Anna Vaughan, Anthony Buonomo, J Scott Hosking, and Richard E Turner. Autoregressive conditional neural processes. *arXiv preprint arXiv:2303.14468*, 2023.

Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control. *PLOS ONE*, 11(2):e0150171, February 2016. ISSN 1932-6203. doi: 10.1371/journal.pone.0150171.

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

Matthew J Colbrook and Alex Townsend. Rigorous data-driven computation of spectral properties of koopman operators for dynamical systems. *Communications on Pure and Applied Mathematics*, 77(1):221–283, 2024.

Matthew J Colbrook, Qin Li, Ryan V Raut, and Alex Townsend. Beyond expectations: residual dynamic mode decomposition and variance for stochastic dynamical systems. *Nonlinear Dynamics*, 112(3):2037–2061, 2024.

Emmanuel de Bézenac, Syama Sundar Rangapuram, Konstantinos Benidis, Michael Bohlke-Schneider, Richard Kurle, Lorenzo Stella, Hilaf Hasson, Patrick Gallinari, and Tim Januschowski. Normalizing Kalman Filters for Multivariate Time Series Analysis. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2995–3007. Curran Associates, Inc., 2020.

Akshunna S Dogra and William Redman. Optimizing neural networks via koopman operator theory. *Advances in Neural Information Processing Systems*, 33:2087–2097, 2020.

Xinyao Fan, Yueying Wu, Chang Xu, Yuhao Huang, Weiqing Liu, and Jiang Bian. MG-TSD: Multi-Granularity Time Series Diffusion Models with Guided Learning Process, March 2024.

Shibo Feng, Chunyan Miao, Ke Xu, Jiaxiang Wu, Pengcheng Wu, Yang Zhang, and Peilin Zhao. Multi-scale Attention Flow for Probabilistic Time Series Forecasting, July 2023.

Andrew Foong, Wessel Bruinsma, Jonathan Gordon, Yann Dubois, James Requeima, and Richard Turner. Meta-learning stationary stochastic process prediction with convolutional neural processes. *Advances in Neural Information Processing Systems*, 33:8284–8295, 2020.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional Neural Processes. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1704–1713. PMLR, July 2018a.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural Processes, July 2018b.

Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard E. Turner. Meta-Learning Probabilistic Inference For Prediction, August 2019a.

Jonathan Gordon, Wessel P Bruinsma, Andrew YK Foong, James Requeima, Yann Dubois, and Richard E Turner. Convolutional conditional neural processes. *arXiv preprint arXiv:1910.13556*, 2019b.

Adèle Gouttes, Kashif Rasul, Mateusz Koren, Johannes Stephan, and Tofigh Naghibi. Probabilistic Time Series Forecasting with Implicit Quantile Networks, July 2021.

Zongyu Guo, Cuiling Lan, Zhizheng Zhang, Yan Lu, and Zhibo Chen. Versatile neural processes for learning implicit neural representations. *arXiv preprint arXiv:2301.08883*, 2023.

Minghao Han, Jacob Euler-Rolle, and Robert K Katzschmann. DESKO: STABILITY-ASSURED ROBUST CONTROL OF NONLINEAR SYSTEMS WITH A DEEP STOCHASTIC KOOPMAN OPERATOR. 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Saurav Jha, Dong Gong, Xuesong Wang, Richard E Turner, and Lina Yao. The neural process family: Survey, applications and perspectives. *arXiv preprint arXiv:2209.00517*, 2022.

Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of koopman eigenfunctions for control. *Machine Learning: Science and Technology*, 2(3):035023, 2021.

Makoto Kawano, Wataru Kumagai, Akiyoshi Sannai, Yusuke Iwasawa, and Yutaka Matsuo. Group equivariant conditional neural processes. *arXiv preprint arXiv:2102.08759*, 2021.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive Neural Processes, July 2019.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL `http://arxiv.org/abs/1312.6114`.

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, 2024.

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

Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.

Jiaqing Kou and Weiwei Zhang. Dynamic mode decomposition with exogenous input for data-driven modeling of unsteady flows. *Physics of fluids*, 31(5), 2019.

Deepthi Praveenlal Kuttichira, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Stock price prediction using dynamic mode decomposition. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 55–60. IEEE, 2017.

Henning Lange, Steven L Brunton, and J Nathan Kutz. From fourier to koopman: Spectral methods for long-term time series prediction. *Journal of Machine Learning Research*, 22(41):1–38, 2021.

Hyungi Lee, Eunggu Yun, Giung Nam, Edwin Fong, and Juho Lee. Martingale posterior neural processes. *arXiv preprint arXiv:2304.09431*, 2023.

Longyuan Li, Junchi Yan, Xiaokang Yang, and Yaohui Jin. Learning Interpretable Deep State Space Model for Probabilistic Time Series Forecasting, January 2021.

Yan Li, Xinjiang Lu, Yaqing Wang, and Dejing Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. *Advances in Neural Information Processing Systems*, 35: 23009–23022, 2022.

Yanhong Li, Jack Xu, and David C. Anastasiu. An Extreme-Adaptive Time Series Prediction Model Based on Probability-Enhanced LSTM Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(7):8684–8691, June 2023. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v37i7.26045.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning Non-stationary Time Series Dynamics with Koopman Predictors.

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

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, November 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-07210-0.

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(3): 859–868, 2024.

Iva Manojlović, Maria Fonoberova, Ryan Mohr, Aleksandr Andrejčuk, Zlatko Drmač, Yannis Kevrekidis, and Igor Mezić. Applications of koopman mode analysis to neural networks. *arXiv preprint arXiv:2006.11765*, 2020.

Peiman Mohseni and Nick Duffield. Spectral convolutional conditional neural processes. *arXiv preprint arXiv:2404.13182*, 2024.

Jeremy Morton, Freddie D Witherden, and Mykel J Kochenderfer. Deep variational koopman models: Inferring koopman observations for uncertainty-aware dynamics modeling and control. *arXiv preprint arXiv:1902.09742*, 2019.

Ilan Naiman and Omri Azencot. A koopman approach to understanding sequence neural models. *arXiv preprint arXiv:2102.07824*, 2021.

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. *arXiv preprint arXiv:2310.02619*, 2023.

Abhinav Narasingam, Sang Hwan Son, and Joseph Sang-Il Kwon. Data-driven feedback stabilisation of nonlinear systems: Koopman-based model predictive control. *International Journal of Control*, 96(3):770–781, 2023.

Nam Nguyen and Brian Quanz. Temporal Latent Auto-Encoder: A Method for Probabilistic Multivariate Time Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35 (10):9117–9125, May 2021. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v35i10.17101.

Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. *arXiv preprint arXiv:2207.04179*, 2022.

Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

Biswajit Paria, Rajat Sen, Amr Ahmed, and Abhimanyu Das. Hierarchically regularized deep forecasting. *arXiv preprint arXiv:2106.07630*, 2021.

Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31, 2018a.

Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep State Space Models for Time Series Forecasting. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018b.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8857–8868. PMLR, July 2021.

William T Redman, Maria Fonoberova, Ryan Mohr, Ioannis G Kevrekidis, and Igor Mezic. An operator theoretic view on pruning deep neural networks. *arXiv preprint arXiv:2110.14856*, 2021.

Said E Said and David A Dickey. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71(3):599–607, 1984.

Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*, 2017.

David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank Gaussian Copula Processes. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, July 2020. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2019.07.001.

Lifeng Shen and James Kwok. Non-autoregressive Conditional Diffusion Models for Time Series Prediction. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 31016–31029. PMLR, July 2023.

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

Binh Tang and David S Matteson. Probabilistic Transformer For Time Series Analysis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 23592–23608. Curran Associates, Inc., 2021.

Kshitij Tayal, Arvind Renganathan, Rahul Ghosh, Xiaowei Jia, and Vipin Kumar. Koopman invertible autoencoder: Leveraging forward and backward dynamics for temporal modeling. In *2023 IEEE International Conference on Data Mining (ICDM)*, pp. 588–597. IEEE, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Rui Wang, Yihe Dong, Sercan Ö Arik, and Rose Yu. Koopman Neural Forecaster for Time Series with Temporal Distribution Shifts, February 2023.

Yuyang Wang, Alex Smola, Danielle Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep factors for forecasting. In *International conference on machine learning*, pp. 6607–6617. PMLR, 2019.

Haomin Wen, Youfang Lin, Yutong Xia, Huaiyu Wan, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. DiffSTG: Probabilistic Spatio-Temporal Graph Forecasting with Denoising Diffusion Models, March 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430. Curran Associates, Inc., 2021.

Tijin Yan, Hongwei Zhang, Tong Zhou, Yufeng Zhan, and Yuanqing Xia. ScoreGrad: Multivariate Probabilistic Time Series Forecasting with Continuous Energy-based Generative Models, June 2021.

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

Yuan Yuan, Kaiwen Zhou, Wenwu Zhou, Xin Wen, and Yingzheng Liu. Flow prediction using dynamic mode decomposition with time-delay embedding based on local measurement. *Physics of Fluids*, 33(9), 2021.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, May 2021. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v35i12.17325.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 27268–27286. PMLR, June 2022.

# A APPENDIX: RELATED WORK

This work engages with three key areas: probabilistic time series prediction, the probabilistic Koopman model, and Neural Process. While each of these fields has received considerable attention in recent literature. Here we talk only about the most related works:

**Probabilistic time series prediction:** Recent developments in probabilistic forecasting for time series integrate deep learning, statistical approaches, and diffusion models. State space models have been significantly enhanced by deep learning, as seen in (Rangapuram et al. (2018b)), (Salinas et al. (2020)) and (Li et al. (2021)), where they effectively capture temporal dynamics and handle missing data. (Feng et al. (2023)) and (Tang & Matteson (2021)) developed attention-based mechanism that enhance long-range dependencies for improved forecast accuracy. Gaussian Process (GP), combined with temporal decomposition, were used by (Yan et al. (2021)) and (Nguyen & Quanz (2021)) to better model uncertainty in multivariate settings. (Salinas et al. (2019)) proposed a Gaussian copula process, and (Ashok et al. (2023)) combines Gaussian copula and the attention mechanism for capturing dependencies in high-dimensional multivariate forecasting. The data distribution-based generative model proposed by (Gouttes et al. (2021)) and the Kalman filter-based approach of (de Bézenac et al. (2020)), blend probabilistic technique for better scalability and robustness. Diffusion models, such as those proposed by (Rasul et al. (2021)), (Li et al. (2022)), and (Fan et al. (2024)), model forecasting as a denoising task, excelling in high-dimensional settings. For faster training and prediction, (Shen & Kwok (2023)) introduced non-autoregressive diffusion model. (Wen et al. (2024)) introduced spatio-temporal diffusion model, extending these methods to spatial dependencies.

**Probabilistic Koopman model:** Over the past two decades, Koopman techniques have garnered substantial attention, with applications spanning analysis (Brunton et al. (2016)), (Takeishi et al. (2017)), (Lusch et al. (2018)), (Azencot et al. (2019)); control Abraham et al. (2017), Korda & Mezić (2018), Kaiser et al. (2021), (Narasingam et al. (2023)); optimization (Dogra & Redman (2020)), (Manojlović et al. (2020)), (Naiman & Azencot (2021)), (Redman et al. (2021)), and forecasting Azencot et al. (2020), Lange et al. (2021), (Wang et al. (2023)), (Liu et al.), (Tayal et al. (2023)). (Brunton et al. (2021)) comprehensively discusses these advances and highlights future research directions. The most closely related work to ours is the probabilistic Koopman model, originally proposed by (Morton et al. (2019)). (Han et al. (2022)) designed a stochastic Koopman neural network for control, in which the latent observables are represented through Gaussian distribution. (Colbrook et al., 2024) integrates the concept of variance into the Koopman framework by introducing variance-pseudospectra, thereby ensuring convergence within the model. For time series tasks, (Naiman et al. (2023)) utilized Koopman theory to represent the latent conditional prior dynamics via a linear map, while (Mallen et al. (2024)) introduced a framework enabling probabilistic forecasting for systems with periodically varying uncertainty.

**Neural process:** Neural Process (NP), first proposed by (Garnelo et al. (2018a)) as Conditional Neural Process, bridges neural networks' scalability and Gaussian Process's ability to model uncertainty. (Garnelo et al. (2018b)) provide flexible, probabilistic function approximation, making them efficient and adaptable across tasks, but struggled with long-range dependencies and flexible function distributions. Various models have since emerged to address these limitations. Attentive Neural Process (Kim et al. (2019)) introduced the attention mechanism to handle long-range dependencies better. Convolutional Conditional Neural Process (Gordon et al. (2019b)) adapted convolutional layers to process images and time series data more effectively. Gaussian Neural Process (Bruinsma et al. (2021)) combined NP with Gaussian inference, improving performance in regression tasks by leveraging Gaussian uncertainty. (Lee et al. (2023)) refined uncertainty quantification of NP, providing better posterior estimation. Other works extend NP to meta-learning, such as Meta-Learning Stationary Stochastic Process (Foong et al. (2020)) and Meta-Learning Probabilistic Inference (Gordon et al. (2019a)), both addressing generalization to unseen tasks. Recent works focus on expanding NP applications to complex data structures. Group Equivariant Conditional Neural Process (Kawano et al. (2021)) and Versatile Neural Process (Guo et al. (2023)) improved the handling of symmetry and representation, enhancing NP's capacity to model equivariant and implicit functions. To improve generation ability further, (Nguyen & Grover (2022)) and (Mohseni & Duffield (2024)) combine NP with Transformer and Neural Operator respectively. A comprehensive survey by (Jha et al. (2022)) has organized these developments, exploring their wide applications in uncertainty-aware learning.

# B BACKGROUND

## B.1 KOOPMAN THEORY AND PSEUDOSPECTRA

The spectrum decomposition of the Koopman operator is a powerful tool for understanding the underlying dynamics. If the operator has a discrete spectrum, it can be expressed in terms of its eigenvalues $\lambda_j$ and eigenfunctions $\phi_j$, where $\mathcal{K}\phi_j = \lambda_j \phi_j$. This decomposition reveals how different modes contribute to the system's evolution. However, many complex systems exhibit a continuous spectrum, indicating chaotic or highly irregular behavior. In such cases, the Koopman operator's spectrum is not composed of isolated eigenvalues but rather a continuum, requiring a more nuanced analysis. The operator can be represented through a spectrum measure $\sigma$ and an integral over the continuous spectrum: $\mathcal{K}g(\mathbf{x}) = \int_\sigma e^{i\omega t} dE(\omega) g(\mathbf{x})$, where $E(\omega)$ is a projection-valued measure. This spectrum approach provides a comprehensive framework for capturing both regular and chaotic components of a system's dynamics. While the Koopman operator effectively models dynamics for predictive purposes, it is prone to spectrum pollution, which can impede convergence and stability. Inspired by perturbations in pseudospectra and the notion that probability distributions characterize variables through overall dispersion, we propose a variance-aware continuous spectrum and model it using Gaussian distribution. This method ensures that eigenvalue perturbations are influenced not only by local disturbances but also by the Gaussian distribution over the dynamics.

## B.2 NEURAL PROCESS

A stochastic process can be viewed as a random function $\mathcal{F} : \mathcal{X} \to \mathcal{Y}$ where inputs can be regarded as indexing the output random variables. With a relaxed use of notation, we employ $p(f)$ in denoting a stochastic process, where $f$ map inputs $x \in \mathcal{X}$ to $y \in \mathcal{Y}$. When fulfilling exchangeability and consistency as stated by Kolmogorov's extension theorem (Oksendal, 2013), Neural Process (NP) cited by Garnelo et al. (2018a) is a stochastic process that describes the predictive distribution over the target set $(\boldsymbol{x}_D, \boldsymbol{y}_D) := (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i \in D}$ given the context set $(\boldsymbol{x}_C, \boldsymbol{y}_C) := (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i \in C}$. Garnelo et al. (2018b) prompts use the distribution of a high-dimensional random vector $\boldsymbol{S}$ to represent $p(f)$ as

$$p(\boldsymbol{y}_D|\boldsymbol{x}_D, \boldsymbol{x}_C, \boldsymbol{y}_C) := \int p(\boldsymbol{y}_D|\boldsymbol{x}_D, \boldsymbol{S}) p(\boldsymbol{S}|\boldsymbol{x}_C, \boldsymbol{y}_C) d\boldsymbol{S}. \tag{19}$$

NP consists of two components: an encoder that maps the input-output pairs of the context set $(\boldsymbol{x}_C, \boldsymbol{y}_C)$ to $\boldsymbol{S}$ for representing $p(f)$, and a decoder that combines $\boldsymbol{S}$ with $\boldsymbol{x}_D$ to generate $\boldsymbol{y}_D$. Due to the intractable log-likelihood, NPs adopt amortized variational inference as Kingma & Welling (2014) and maximize the evidence lower bound (ELBO) of the log-likelihood as follows

$$\log\left(\boldsymbol{y}_D|\boldsymbol{x}_D, \boldsymbol{y}_C, \boldsymbol{x}_C\right) \geq \mathbb{E}_{\boldsymbol{S} \sim q_{(\boldsymbol{S}|\boldsymbol{x}_D, \boldsymbol{y}_D)}}\left[\log p\left(\boldsymbol{y}_D|\boldsymbol{x}_D, \boldsymbol{S}\right)\right] - D_{KL}\left(q\left(\boldsymbol{S}|\boldsymbol{x}_D, \boldsymbol{y}_D\right) || p\left(\boldsymbol{S}|\boldsymbol{x}_C, \boldsymbol{y}_C\right)\right). \tag{20}$$

# C BENCHMARK DATASETS

For our experiments, we use `ETTs`, `Solar`, `Electricity`, `Traffic`, `Taxi`, and `KDD-cup` open-source datasets, with their properties listed in Tab.C. The dataset can be obtained through the links below.

(i) `ETTs`: https://github.com/zhouhaoyi/ETDataset

(ii) `Solar`: https://www.nrel.gov/grid/solar-power-data.html

(iii) `Electricity`: https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014

(iv) `Traffic`: https://pems.dot.ca.gov

(v) `Taxi`: https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

(vi) `KDD-cup`: https://www.kdd.org/kdd2018/kdd-cup

Table 3: Datasets detail

| Name | Frequency | Dimensions | Context length | Prediction length |
|---|---|---|---|---|
| ETTh1 | 1 hour | 7 | 10 | 24 |
| ETTh2 | 1 hour | 7 | 10 | 24 |
| ETTm1 | 15 min | 7 | 10 | 24 |
| ETTm2 | 15 min | 7 | 10 | 24 |
| Solar | 1 hour | 137 | 15 | 24 |
| Electricity | 1 hour | 370 | 10 | 24 |
| Traffic | 1 hour | 862 | 10 | 24 |
| Taxi | 30 min | 1214 | 15 | 24 |
| KDD-cup | 1 hour | 270 | 10 | 48 |

# D APPENDIX: EVALUATION METRIC

We consider two metrics: $\mathbf{CRPS_{sum}}$ and $\mathbf{NRMSE_{sum}}$, the first one can describe the predictive distribution, and the second can describe the distance between truth value and prediction mean, more details can be found in Gluonts documentation (Alexandrov et al. (2020)).

$\mathbf{CRPS_{sum}}$: $\mathbf{CRPS}$ is a univariate, strictly proper scoring rule that quantifies the compatibility of a cumulative distribution function $F$ with an observed value $x \in \mathbb{R}$ as:

$$\mathbf{CRPS} = \int_{\mathbb{R}} (\mathbf{F(y)} - \mathbb{I}(\mathbf{x} \leq \mathbf{y}))^{\mathbf{2}} \mathbf{dy}, \tag{21}$$

where $\mathbb{I}(x \leq y)$ denotes the indicator function. The CRPS achieves the minimum value when predictive prediction $F$ same as the data distribution. $\mathbf{CRPS}$ can be extend to $\mathbf{CPRS_{sum}}$ to evaluate multivariate distribution:

$$\mathbf{CRPS_{sum}} = \mathbb{E}_t[\mathbf{CPRS}(F^{-1}_{sum}, \sum_i x^i_t)], \tag{22}$$

where $F^{-1}_{sum}$ is computed by aggregating samples across dimensions and subsequently sorting them to obtain quantiles. A smaller $\mathbf{CRPS_{sum}}$ indicates more accurate predictions.

$\mathbf{NRMSE_{sum}}$: $\mathbf{NRMSE_{sum}}$ is an adaptation of the Root Mean Squared Error (RMSE) that accounts for the scale of the target values. It is defined as follows:

$$\mathbf{NRMSE_{sum}} = \sqrt{\frac{\mathrm{mean}((\hat{\mathbf{Y}} - \mathbf{Y})^{\mathbf{2}})}{\mathrm{mean}(|\mathbf{Y}|)}}, \tag{23}$$

where $\hat{Y}$ represents the predicted time series, and $Y$ represents the true target time series. $\mathbf{NRMSE_{sum}}$ quantifies the average squared difference between predictions and true values across all dimensions, normalized by the mean absolute magnitude of the target values. A smaller $\mathbf{NRMSE_{sum}}$ indicates more accurate predictions.

# E DETAILS ON TRAINING BASELINES

We train baselines by open code which is reported in corresponding papers, and follow the default setting. The code for the baseline methods is obtained from the following sources.

- GP-Copula: https://github.com/mbohlkeschneider/gluon-ts/tree/mvrelease
- Transfomer-MAF:https://github.com/zalandoresearch/pytorch-ts/tree/master/pts/model/transformertempflow
- Timegrad: https://github.com/zalandoresearch/pytorch-ts
- TACTIS: https://github.com/servicenow/tactis

- $D^3$VAE: https://github.com/ramber1836/d3vae
- DPK: https://github.com/AlexTMallen/koopman-forecasting
- MG-TSD: https://github.com/Hundredl/MG-TSD

1.GP-Copula (2019): A method combining Gaussian Process with Copulas to model complex dependencies between variables in multivariate time series, providing uncertainty estimates and capturing nonlinear correlations.

2. Transformer-MAF (2021) Combines Transformers with Masked Autoregressive Flow (MAF) to model long-term dependencies and the conditional probability distribution of time series data effectively.

3. TimeGrad (2021) A diffusion model-based approach for time series forecasting, which progressively generates samples to capture complex dynamics and uncertainty in the data.

4. TACTiS (2023) A probabilistic autoregressive model leveraging Transformers to handle non-stationary time series, focusing on dynamic structures and probabilistic predictions.

5. D3VAE (2023) A deep variational autoencoder (VAE)-based model designed for time series, featuring a dynamic decoder to effectively capture and predict complex temporal structures.

6. DPK (2024) Dynamic Probabilistic Kernel (DPK) models probabilistic dependencies in time series using a dynamic kernel-based approach, balancing flexibility and efficiency for multivariate data.

7. MG-TSD (2024) Multi-Granularity Time Series Decomposition (MG-TSD) decomposes time series into components of varying frequencies or trends, modeling each with a probabilistic framework to capture multi-scale patterns.

# F APPENDIX: IMPLEMENTATION DETAILS

In Tab.4, we show the hyperparameters of KooNPro include time length $T$, delay-embedded length $k$, layers of the encoder $\boldsymbol{\phi}$, layers of the decoder $\boldsymbol{\phi}^{-1}$, layers of the auxiliary $\boldsymbol{\kappa}$, layers of $\boldsymbol{\psi}$ shown in Eq.3. Note that the choice of $T$ and $k$ is based on the ablation study showcased in Appendix K.

Table 4: Hyperparameters of KooNPro

| Name | $T$ | $k$ | $\boldsymbol{\phi}$ | $\boldsymbol{\phi}^{-1}$ | $\boldsymbol{\kappa}$ | $\boldsymbol{\psi}$ |
|---|---|---|---|---|---|---|
| ETTh1 | 30 | 10 | 4 | 4 | 4 | 2 |
| ETTh2 | 20 | 10 | 4 | 4 | 4 | 2 |
| ETTm1 | 20 | 10 | 4 | 4 | 4 | 2 |
| ETTm1 | 20 | 10 | 4 | 4 | 4 | 2 |
| Solar | 30 | 15 | 8 | 8 | 8 | 3 |
| Electricity | 30 | 10 | 4 | 4 | 4 | 2 |
| Traffic | 20 | 10 | 4 | 4 | 4 | 2 |
| Taxi | 30 | 15 | 6 | 6 | 6 | 3 |
| KDD-cup | 10 | 10 | 4 | 4 | 4 | 2 |

# G APPENDIX: LONG-TERM PREDICTION

To assess KooNPro's predictive capability in capturing hidden temporal dynamics within time series, we evaluate its performance under extended prediction length, as detailed in Tab.5. According to the results in Tab.6, KooNPro exhibits the least degradation across both metrics compared to baseline methods. This outcome highlights the effectiveness of KooNPro in learning and leveraging the temporal dynamics of time series for accurate prediction.

Table 5: Datasets detail

| Name | Frequency | Dimensions | Context length | Prediction length |
|---|---|---|---|---|
| ETTh1 | 1 hour | 7 | 10 | 48 |
| ETTh2 | 1 hour | 7 | 10 | 48 |
| ETTm1 | 15 min | 7 | 10 | 48 |
| ETTm2 | 15 min | 7 | 10 | 48 |
| Solar | 1 hour | 137 | 15 | 48 |
| Electricity | 1 hour | 370 | 10 | 48 |
| Traffic | 1 hour | 862 | 10 | 48 |
| Taxi | 30 min | 1214 | 15 | 48 |
| KDD-cup | 1 hour | 270 | 10 | 72 |

Table 6: Comparison of $\mathbf{CRPS_{sum}}$ (denoted as C-s, smaller is better) and $\mathbf{NRMSE_{sum}}$ (denoted as N-s, smaller is better) across nine real-world datasets. The means and standard errors are based on 10 independent runs of retraining and evaluation. The best performances are highlighted in red and the second are in blue. A block marked with '-' denotes a numerical issue encountered during model training with longer prediction lengths.

| Model | Metric | ETTh1 | ETTh2 | ETTm1 | ETTm2 | Solar | Electricity | Traffic | Taxi | Cup |
|---|---|---|---|---|---|---|---|---|---|---|
| GP-Copula | C-s | $0.611_{\pm 0.031}$ | $0.381_{\pm 0.034}$ | $0.646_{\pm 0.056}$ | $0.558_{\pm 0.054}$ | $0.465_{\pm 0.089}$ | $0.234_{\pm 0.047}$ | $0.529_{\pm 0.006}$ | $1.007_{\pm 0.025}$ | $0.731_{\pm 0.025}$ |
| | N-s | $0.909_{\pm 0.033}$ | $0.610_{\pm 0.062}$ | $0.762_{\pm 0.138}$ | $0.844_{\pm 0.093}$ | $0.709_{\pm 0.085}$ | $0.382_{\pm 0.103}$ | $0.713_{\pm 0.014}$ | $1.464_{\pm 0.057}$ | $1.073_{\pm 0.056}$ |
| Trans-MAF | C-s | $1.271_{\pm 0.051}$ | $0.507_{\pm 0.014}$ | $1.045_{\pm 0.085}$ | $0.279_{\pm 0.007}$ | - | $0.198_{\pm 0.069}$ | $0.596_{\pm 0.031}$ | $0.769_{\pm 0.046}$ | $0.410_{\pm 0.073}$ |
| | N-s | $1.571_{\pm 0.144}$ | $0.780_{\pm 0.027}$ | $1.797_{\pm 0.122}$ | $0.485_{\pm 0.127}$ | - | $0.397_{\pm 0.127}$ | $0.872_{\pm 0.046}$ | $0.936_{\pm 0.010}$ | $0.515_{\pm 0.083}$ |
| Timegrid | C-s | $0.796_{\pm 0.084}$ | $0.477_{\pm 0.007}$ | $0.458_{\pm 0.059}$ | $0.346_{\pm 0.010}$ | $0.886_{\pm 0.036}$ | $0.263_{\pm 0.028}$ | $0.726_{\pm 0.050}$ | $0.791_{\pm 0.021}$ | $0.421_{\pm 0.059}$ |
| | N-s | $0.953_{\pm 0.102}$ | $0.697_{\pm 0.009}$ | $0.588_{\pm 0.104}$ | $0.455_{\pm 0.014}$ | $1.243_{\pm 0.056}$ | $0.423_{\pm 0.077}$ | $0.932_{\pm 0.055}$ | $0.971_{\pm 0.195}$ | $0.522_{\pm 0.098}$ |
| TACTIS | C-s | $0.752_{\pm 0.004}$ | $0.401_{\pm 0.001}$ | $1.331_{\pm 0.013}$ | $0.261_{\pm 0.023}$ | $3.786_{\pm 1.708}$ | $0.360_{\pm 0.004}$ | $0.552_{\pm 0.067}$ | $1.368_{\pm 0.014}$ | $0.390_{\pm 0.018}$ |
| | N-s | $0.943_{\pm 0.004}$ | $0.522_{\pm 0.002}$ | $1.853_{\pm 0.030}$ | $0.422_{\pm 0.023}$ | $5.615_{\pm 2.168}$ | $0.498_{\pm 0.002}$ | $0.751_{\pm 0.023}$ | $1.591_{\pm 0.020}$ | $0.505_{\pm 0.022}$ |
| $D^3$VAE | C-s | $0.916_{\pm 0.036}$ | $0.626_{\pm 0.044}$ | $0.598_{\pm 0.021}$ | $0.737_{\pm 0.064}$ | $0.725_{\pm 0.064}$ | $0.408_{\pm 0.048}$ | $0.704_{\pm 0.069}$ | $0.814_{\pm 0.035}$ | - |
| | N-s | $1.265_{\pm 0.078}$ | $0.861_{\pm 0.041}$ | $0.768_{\pm 0.067}$ | $0.951_{\pm 0.031}$ | $0.919_{\pm 0.141}$ | $0.601_{\pm 0.051}$ | $1.126_{\pm 0.139}$ | $1.308_{\pm 0.164}$ | - |
| DPK | C-s | $0.891_{\pm 0.027}$ | $0.744_{\pm 0.071}$ | $0.824_{\pm 0.040}$ | $0.519_{\pm 0.092}$ | $0.938_{\pm 0.004}$ | $0.997_{\pm 0.012}$ | $1.131_{\pm 0.002}$ | $0.969_{\pm 0.006}$ | $0.900_{\pm 0.015}$ |
| | N-s | $1.262_{\pm 0.032}$ | $0.998_{\pm 0.138}$ | $1.349_{\pm 0.070}$ | $0.592_{\pm 0.202}$ | $1.301_{\pm 0.004}$ | $1.263_{\pm 0.013}$ | $1.478_{\pm 0.005}$ | $1.213_{\pm 0.005}$ | $1.309_{\pm 0.024}$ |
| MG-TSD | C-s | $0.619_{\pm 0.056}$ | $0.435_{\pm 0.099}$ | $0.371_{\pm 0.085}$ | $0.269_{\pm 0.005}$ | $1.000_{\pm 0.001}$ | $0.174_{\pm 0.027}$ | $0.617_{\pm 0.045}$ | $0.409_{\pm 0.051}$ | $0.590_{\pm 0.091}$ |
| | N-s | $0.967_{\pm 0.071}$ | $0.627_{\pm 0.117}$ | $0.539_{\pm 0.109}$ | $0.318_{\pm 0.050}$ | $1.610_{\pm 1.286}$ | $0.283_{\pm 0.035}$ | $0.882_{\pm 0.053}$ | $0.621_{\pm 0.068}$ | $0.767_{\pm 0.073}$ |
| KooNPro | C-s | $0.488_{\pm 0.025}$ | $0.376_{\pm 0.025}$ | $0.365_{\pm 0.018}$ | $0.227_{\pm 0.002}$ | $0.417_{\pm 0.021}$ | $0.165_{\pm 0.017}$ | $0.401_{\pm 0.014}$ | $0.396_{\pm 0.023}$ | $0.327_{\pm 0.014}$ |
| | N-s | $0.746_{\pm 0.044}$ | $0.541_{\pm 0.035}$ | $0.564_{\pm 0.023}$ | $0.343_{\pm 0.034}$ | $0.657_{\pm 0.040}$ | $0.287_{\pm 0.017}$ | $0.715_{\pm 0.030}$ | $0.658_{\pm 0.042}$ | $0.457_{\pm 0.021}$ |

## H APPENDIX: ROBUSTNESS OF PERFORMANCE

To assess the robustness of KooNPro's predictive performance, we evaluate it under varying Signal-to-Noise Ratio (SNR) conditions (20dB, 40dB, and 60dB). During training, KooNPro is trained with ground truth data. In the testing phase, Gaussian noise is added to the input data, and predictions are compared to the ground truth. As shown in Tab.7, the performance degradation with decreasing SNR remains within acceptable limits, demonstrating KooNPro's robust predictive capability across different noise levels.

Table 7: KooNPro represents robust performance across various noisy-add scenarios, with SNR ranging from 20dB to 60dB. Note that KooNPro is trained once on noise-free data and then tested on different noisy-add data.

| Model | Metric | ETTh1 | ETTh2 | ETTm1 | ETTm2 | Solar | Elec. | Traffic | Taxi | Cup |
|---|---|---|---|---|---|---|---|---|---|---|
| KooNPro | C-s | $0.328_{\pm 0.037}$ | $0.149_{\pm 0.051}$ | $0.165_{\pm 0.057}$ | $0.081_{\pm 0.020}$ | $0.211_{\pm 0.033}$ | $0.057_{\pm 0.006}$ | $0.184_{\pm 0.022}$ | $0.226_{\pm 0.041}$ | $0.204_{\pm 0.017}$ |
| | N-s | $0.520_{\pm 0.045}$ | $0.224_{\pm 0.065}$ | $0.225_{\pm 0.028}$ | $0.122_{\pm 0.034}$ | $0.313_{\pm 0.044}$ | $0.095_{\pm 0.012}$ | $0.289_{\pm 0.025}$ | $0.330_{\pm 0.078}$ | $0.308_{\pm 0.030}$ |
| 60dB | C-s | $0.367_{\pm 0.023}$ | $0.195_{\pm 0.029}$ | $0.183_{\pm 0.017}$ | $0.111_{\pm 0.012}$ | $0.229_{\pm 0.032}$ | $0.074_{\pm 0.008}$ | $0.192_{\pm 0.023}$ | $0.248_{\pm 0.017}$ | $0.217_{\pm 0.055}$ |
| | N-s | $0.586_{\pm 0.023}$ | $0.284_{\pm 0.032}$ | $0.380_{\pm 0.021}$ | $0.132_{\pm 0.015}$ | $0.323_{\pm 0.042}$ | $0.118_{\pm 0.014}$ | $0.333_{\pm 0.029}$ | $0.337_{\pm 0.021}$ | $0.339_{\pm 0.105}$ |
| 40dB | C-s | $0.385_{\pm 0.028}$ | $0.221_{\pm 0.021}$ | $0.271_{\pm 0.033}$ | $0.152_{\pm 0.001}$ | $0.243_{\pm 0.039}$ | $0.192_{\pm 0.052}$ | $0.213_{\pm 0.019}$ | $0.274_{\pm 0.015}$ | $0.229_{\pm 0.019}$ |
| | N-s | $0.608_{\pm 0.049}$ | $0.317_{\pm 0.024}$ | $0.565_{\pm 0.048}$ | $0.162_{\pm 0.003}$ | $0.359_{\pm 0.038}$ | $0.283_{\pm 0.061}$ | $0.344_{\pm 0.029}$ | $0.380_{\pm 0.012}$ | $0.352_{\pm 0.031}$ |
| 20dB | C-s | $0.432_{\pm 0.021}$ | $0.251_{\pm 0.010}$ | $0.311_{\pm 0.018}$ | $0.194_{\pm 0.017}$ | $0.269_{\pm 0.047}$ | $0.287_{\pm 0.058}$ | $0.268_{\pm 0.016}$ | $0.285_{\pm 0.026}$ | $0.249_{\pm 0.020}$ |
| | N-s | $0.687_{\pm 0.034}$ | $0.331_{\pm 0.028}$ | $0.491_{\pm 0.026}$ | $0.245_{\pm 0.019}$ | $0.385_{\pm 0.056}$ | $0.415_{\pm 0.049}$ | $0.369_{\pm 0.024}$ | $0.384_{\pm 0.012}$ | $0.395_{\pm 0.036}$ |

## I APPENDIX: THE PARAMETERS OF LEARNED MODELS

The backbone of KooNPro is based on MLPs, with the number of parameters determined by the input data dimensions, as well as the depth and width of the MLP. Tab. 8 compares the memory consumption of KooNPro to baseline methods. Furthermore, Fig.5 illustrates the relationship between memory consumption and predictive performance for each model on the `ETTm2` dataset. The results indicate that KooNPro achieves the best performance with the fewest memory consumption.

Table 8: Comparison of memory consumption for different models' parameters across nine datasets (in MB).

| Model | ETTh1 | ETTh2 | ETTm1 | ETTm2 | Solar | Elec. | Traffic | Taxi | Cup |
|---|---|---|---|---|---|---|---|---|---|
| Trans-MAF | 9.69 | 9.69 | 9.69 | 9.69 | 12.21 | 16.73 | 26.26 | 33.08 | 14.79 |
| Timegrid | 4.22 | 4.22 | 4.22 | 4.22 | 25.93 | 29.32 | 37.83 | 45.06 | 27.82 |
| TACTIS | 7.48 | 7.48 | 7.48 | 7.48 | 7.49 | 7.49 | 7.50 | 7.51 | 7.49 |
| $D^3$VAE | 58.41 | 58.41 | 58.41 | 58.41 | 58.72 | 59.29 | 60.50 | 61.36 | 59.05 |
| DPK | 0.17 | 0.17 | 0.17 | 0.17 | 0.25 | 0.40 | 0.70 | 0.92 | 0.34 |
| MG-TSD | 2.41 | 2.41 | 2.41 | 2.41 | 3.05 | 6.33 | 14.60 | 21.65 | 5.87 |
| KooNPro | 1.73 | 1.73 | 1.73 | 1.73 | 10.23 | 7.18 | 14.42 | 17.50 | 5.60 |

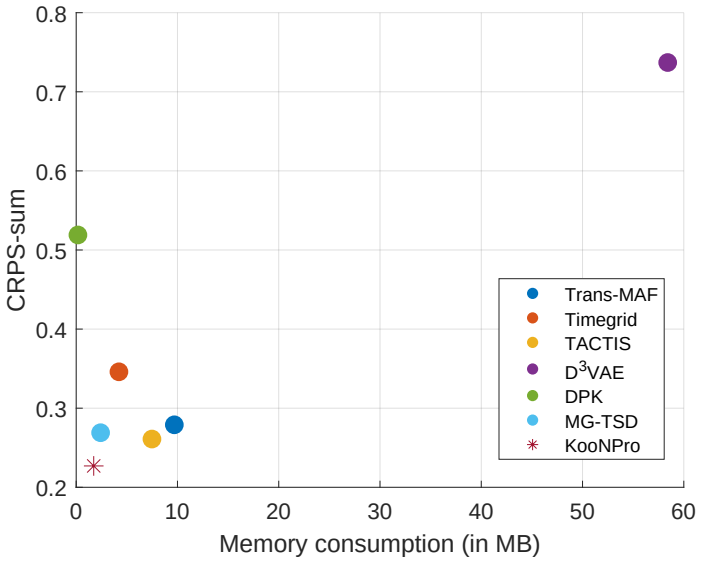


Figure 5: The plot illustrates the relationship between memory consumption and prediction performance of each model on `ETTm2`. The x-axis indicates memory consumption (in MB), and the y-axis represents $\mathbf{CRPS_{sum}}$. Optimal performance is achieved as the marker approaches the bottom-left corner.

## J Implementation of Neural Process through different ways

This section details the implementation of various methods for generating $\boldsymbol{S}_D$, represented the point spectra globally and learned by Neural Process. We introduce three KooNPro variants: **Without-NP**, **With-ANP**, and **With-GP**.

- **Without-NP** closely follows the KooNPro procedure. To isolate the impact of $\boldsymbol{S}_D$, the training process adheres to Sec. 4.2, omitting $\boldsymbol{S}$ generated in Sec.4.1. The test procedure is outlined as follows

$$\boldsymbol{z}_{\tilde{T}} = \boldsymbol{\phi}^{-1}(\mathcal{A}^{\tilde{T}-1}(\boldsymbol{\phi}(\boldsymbol{z}_1))). \tag{24}$$

- **With-ANP**, following the Attention Neural Process (ANP) (Kim et al., 2019) and integrating it with our work, employs the attention mechanism to generate $\boldsymbol{S}_D$, which governs the temporal dynamics of the entire time series. In this process, we set the Key $K$ as $\boldsymbol{x}_C$, the Value $V$ as $\boldsymbol{S}_C$, and the Query $Q$ as $\boldsymbol{x}_D$, then compute $\boldsymbol{S}_D$.

- **With-GP** employs the Gaussian process to predict $\boldsymbol{S}_D$ at the target set $\boldsymbol{x}_D$ using context set $\boldsymbol{x}_C$ and their corresponding outputs $\boldsymbol{S}_C$ by modeling a joint Gaussian distribution. The procedure can be summarized as follows

$$\mathbf{K}_C = \mathrm{k}(\boldsymbol{z}_C, \boldsymbol{z}_C) + \sigma_n^2, \tag{25}$$
$$\mathbf{K}_D = \mathrm{k}(\boldsymbol{z}_D, \boldsymbol{z}_D), \tag{26}$$
$$\mathbf{K}_{CD} = \mathrm{k}(\boldsymbol{z}_C, \boldsymbol{z}_D), \tag{27}$$

  where $\mathrm{k}$ we choose the radial basis function kernel. Consequently, the distribution of $\boldsymbol{S}_D \sim \mathcal{N}(\mu, \Sigma)$ can be calculated by

$$\mu = K_{CD}^\top K_C^{-1} \boldsymbol{S}_C, \tag{28}$$
$$\Sigma = K_D - K_{CD}^\top K_C^{-1} K_{CD}, \tag{29}$$

  where $\mu$ is mean and $\Sigma$ is covariance.

# K APPENDIX: ABLATION STUDY

We conduct ablation studies on the time length $T$ and delay-embedded length $k$ across several datasets. The results indicate that merely extending the learning time segment is insufficient to reveal the temporal dynamics underlying the time series, as discussed in Sec.5.3. The choices for $T$ and $k$ presented in Tab.4 are guided by the results of this ablation study.

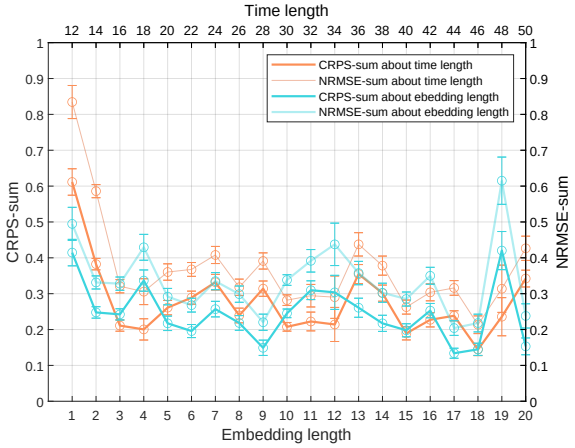


Figure 6: Ablation study on `ETTh2` dataset.

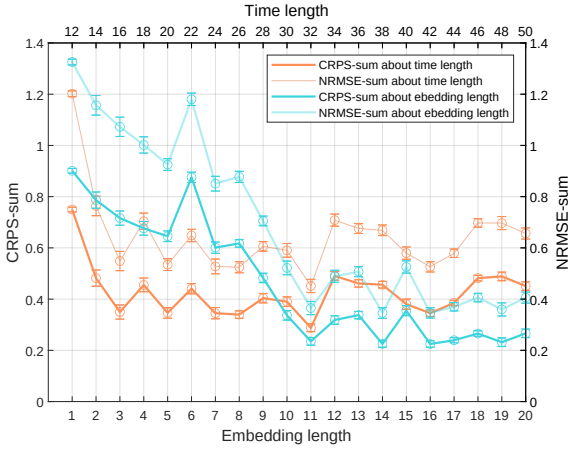


Figure 7: Ablation study on `ETTm1` dataset.

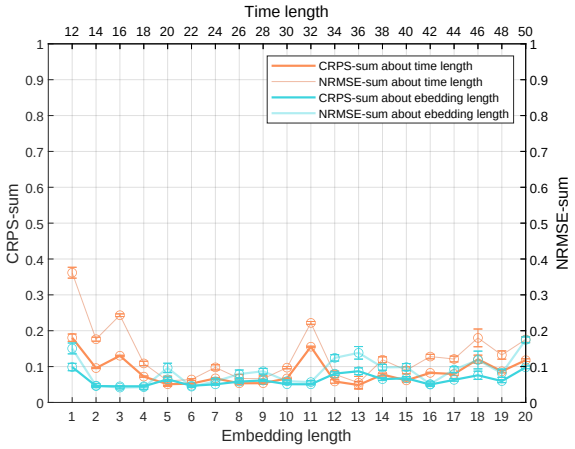


Figure 8: Ablation study on `ETTm2` dataset.

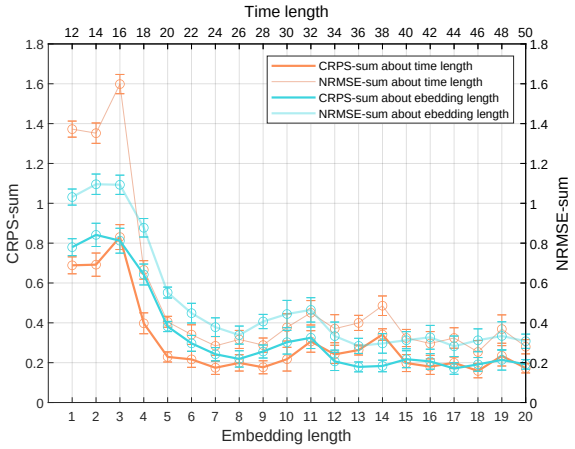


Figure 9: Ablation study on `Solar` dataset.

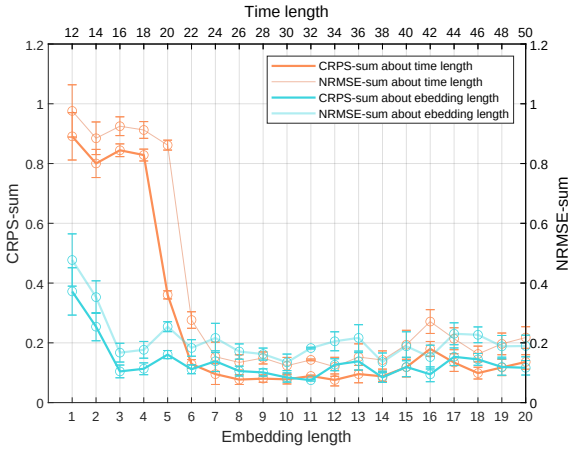


Figure 10: Ablation study on `Electricity` dataset.

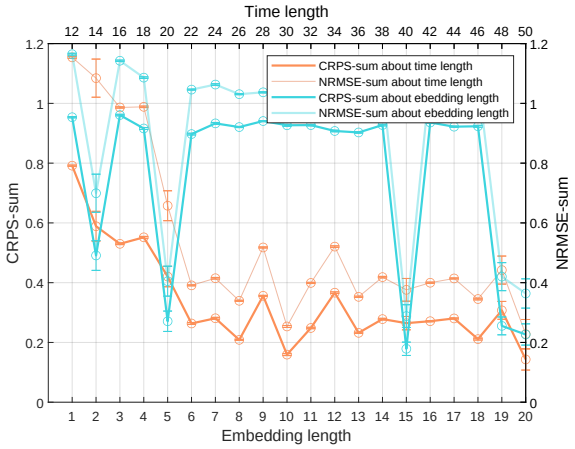


Figure 11: Ablation study on `Taxi` dataset.

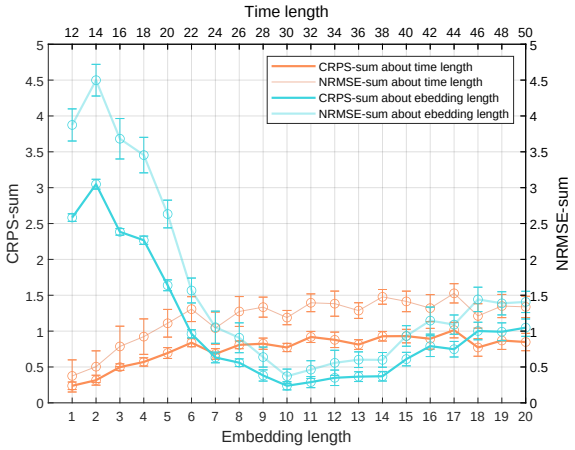


Figure 12: Ablation study on `KDD-cup` dataset.