BayOTIDE: Bayesian Online Multivariate Time Series Imputation with Functional Decomposition

Shikai Fang [1] [†]   **Qingsong Wen** [2] [‡]   **Yingtao Luo** [3] [†]   **Shandian Zhe** [1]   **Liang Sun** [2]

Abstract

In real-world scenarios such as traffic and energy management, we frequently encounter large volumes of time-series data characterized by missing values, noise, and irregular sampling patterns. While numerous imputation methods have been proposed, the majority tend to operate within a local horizon, which involves dividing long sequences into batches of fixed-length segments for model training. This local horizon often leads to the overlooking of global trends and periodic patterns. More importantly, most methods assume the observations are sampled at regular timestamps, and fail to handle complex irregular sampled time series in various applications. Additionally, most existing methods are learned in an offline manner. Thus, it is not suitable for applications with rapidly arriving streaming data. To address these challenges, we propose *BayOTIDE* : Bayesian Online Multivariate Time series Imputation with functional decomposition. Our method conceptualizes multivariate time series as the weighted combination of groups of low-rank temporal factors with different patterns. We employ a suite of Gaussian Processes (GPs),each with a unique kernel, as functional priors to model these factors. For computational efficiency, we further convert the GPs into a state-space prior by constructing an equivalent stochastic differential equation (SDE), and developing a scalable algorithm for online inference. The proposed method can not only handle imputation over arbitrary timestamps, but also offer uncertainty quantification and interpretability for the downstream application. We evaluate our method

on both synthetic and real-world datasets. We release the code at `https://github.com/xuangu-fang/BayOTIDE`.

1. Introduction

Multivariate time series data are ubiquitous and generated quickly in many real-world applications (Wen et al., 2022; Esling & Agon, 2012; Zhang et al., 2024), such as traffic (Chen et al., 2024; Li et al., 2015) and energy (Zhu et al., 2023; Grabner et al., 2023). However, the collected data are often incomplete and noisy due to sensor failures, communication errors, or other reasons. The missing values in the time series data can lead to inaccurate downstream analysis. Therefore, it is essential to impute the missing values in the time series data in an efficient way.

Most early methods for time series imputation (Acuna & Rodriguez, 2004; Van Buuren & Groothuis-Oudshoorn, 2011; Durbin & Koopman, 2012) are based on statistical models. Deep learning based imputation methods have gotten boosted attention (Wang et al., 2024a; Fang & Wang, 2020) in recent years, for their ability to capture complex nonlinear patterns. Another remarkable direction is to apply diffusion models (Yang et al., 2024; Song et al., 2020; Ho et al., 2020) to handle probabilistic imputation, where filling the missing value can be modeled as a denoising process. Most recent work TIDER (LIU et al., 2023) proposed another imputation direction to apply the matrix factorization and decompose the series into disentangled representations.

Despite the success of the proposed methods, they are limited in several aspects. First, most DNN-based and diffusion-based methods are trained by splitting the long sequence into small patches. This local horizon can fail to capture the crucial global patterns (Alcaraz & Strodthoff, 2022; Woo et al., 2022), such as trends and periodicity (Chen et al., 2022; Wen et al., 2021), leading to less interpretability. Second, many methods assume the observations are sampled at regular time stamps, and always under-utilize or ignore the real timestamps. Thus, those models can only impute on fixed-step and discretized time points, instead of the arbitrary time stamps at the whole continuous field. Lastly, in real-world applications such as electricity load monitoring, massive time series data are generated quickly and collected

---

in a streaming manner (Liu et al., 2023). It is extremely costly or even impossible to retrain the model from scratch when new data arrives. Thus, to align with streaming data, the imputation model should work and update in an efficient online manner. However, to the best of our knowledge, all prior imputation methods are designed and optimized in an offline manner, i.e., go through all collected data several epochs for training, which is not suitable for streaming data scenarios.

To handle those limitations, we propose *BayOTIDE* : Bayesian Online multivariate Time series Imputation with functional DEcomposition. *BayOTIDE* treats the observed values of multivariate time series as the noisy samples from a continuous temporal function. Then, we decompose the function into groups of weighted functional factors, and each factor is aimed to capture a dynamic pattern. We apply Gaussian Processes (GPs) with smooth and periodic kernels as functional priors to fit the factors. Employing the SDE representation of GPs and moment-matching techniques, we develop an online algorithm to infer the running posterior of weights and factors efficiently. As it is a Bayesian model, *BayOTIDE* can offer uncertainty quantification and robustness against noise. The learned functional factors can provide not only interpretability but also imputation over arbitrary timestamps. We list the comparison of *BayOTIDE* and other main-stream imputation methods in Table 1. In summary, we highlight our contributions as follows:

- We propose *BayOTIDE* , a novel Bayesian method for multivariate time series imputation. *BayOTIDE* can explicitly learn the function factors representing various global patterns, which offer interpretability and uncertainty quantification. As *BayOTIDE* is a continuous model, it can utilize the irregularly sampled timestamps and impute over arbitrary timestamps naturally.

- To the best of our knowledge, *BayOTIDE* is the first online probabilistic imputation method of multivariate time series that could fit streaming data well. Furthermore, we develop a scalable inference algorithm with closed-form update and linear cost via moment-matching techniques.

- We extensively evaluate our method on synthetic and real-world datasets, and the results show that *BayOTIDE* outperforms the state-of-the-art methods in terms of accuracy and efficiency.

2. Related Work

Disentangled representations of time series. The most classical framework of decomposing time series into disentangled representations is the seasonal-trend decomposition(STL) (Cleveland et al., 1990) along with its following work (Wen et al., 2019; 2020; Abdollahi, 2020; Bandara et al., 2021), which are non-parametric method to decompose the univariate series into seasonal, trend and residual components. (Qiu et al., 2018) proposed the structural design to extend decomposition into multivariate and probabilistic cases. Recently, CoST (Woo et al., 2022) and TIDER (LIU et al., 2023) show the disentangled representations of multivariate series could get significant performance improvement in forecasting and imputation tasks, respectively, with bonus of interpretability. However, they are not flexible enough to handle the continuous time field and observation noise. Benavoli & Corani (2021) propose a similar idea to directly utilize the state-space GPs with mixture kernels to estimate the seasonal-trend factors, but is restricted in univariate series. Besides the imputation and forecasting, learning the disentangled representation of time series is a crucial tool for estimating causal-related latent variables in sequential data (Yao et al., 2021; 2022). A series of recent works on tensor-valued time series analysis (Fang et al., 2022; 2024; Wang et al., 2024b) also apply the idea of disentangled representations to learn temporal dynamics in latent spaces.

Bayesian imputation modeling. Bayesian methods are widely used in time series imputation tasks for robust modeling and uncertainty quantification. Early work directly applies a single Bayesian model like Gaussian Process (Roberts et al., 2013) and energy models (Brakel et al., 2013) to model the dynamics. With deep learning boosting in the past ten years, it is popular to utilize the probabilistic modules with various deep networks, such as RNN (Mulyadi et al., 2021), VAE (Fortuin et al., 2020) and GAN (Yoon et al., 2018). Adopting score-based generative models (SGMs) is another promising direction for probabilistic imputation, which could be used as autoregressive denoising (Rasul et al., 2021), conditional diffusion (Tashiro et al., 2021), Schrödinger Bridge (Chen et al., 2023) and state-space blocks (Alcaraz & Strodthoff, 2022). However, most of the above methods are trained in the offline and patching-sequence manner, which lacks interpretability and may not fit streaming scenarios.

3. Background

3.1. Multivariate Time Series Imputation

The classical multivariate time series imputation problem is formulated as follows. A N-step multivariate time series $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \in \mathbb{R}^{D \times N}$, where $\mathbf{x}_n \in \mathbb{R}^D$ is the D-size value at n-th step and $\mathbf{x}_n^d$ represents it's values at d-th channel. There is a mask matrix $\mathbf{M} \in \{0, 1\}^{D \times N}$, indicating whether the series value is observed or missing. The goal is to use the observed values, where $\mathbf{M}_{d,n} = 1$, to estimate the missing values $\mathbf{x}_n^d$, where $\mathbf{M}_{d,n} = 0$. In the above setting, the interval between two consecutive timestamps is

Properties / Methods	**BayOTIDE**	TIDER	Statistic-based	DNN-based	Diffusion-based
Uncertainty-aware	✓	✗	✗	✗*	✓
Interpretability	✓	✓	✓	✗	✗
Continuous modeling	✓	✗	✗	✗	✗*
Inference manner	**online**	offline	offline	offline	offline

Table 1: Comparison of *BayOTIDE* and main-stream multivariate time series imputation methods. ✗* means only partial models in the family have the property, or it's not clear in the original paper. For example, only deep models with probabilistic modules can offer uncertainty quantification, such as GP-VAE (Fortuin et al., 2020), but most deep models cannot. The diffusion-based CSDI (Tashiro et al., 2021) and CSBI (Chen et al., 2023) take timestamps as input, but the model is trained with discretized time embedding.

assumed to be constant by default. If the timestamps are irregularly sampled and continuous, the problem becomes more challenging and the exact timestamps $\{t_1, \ldots, t_N\}$ should be considered in the imputation model. In this paper, we aimed to learn a general function $\mathbf{X}(t) : t \to \mathbb{R}^D$ to impute the missing values at any time $t \in [t_1, t_N]$.

3.2. Gaussian Process (GP) and State-Space Model

Gaussian Process (GP) (Rasmussen & Williams, 2006)s is a powerful Bayesian prior for functional approximation, always denoted as $f \sim \mathcal{GP}(0, \kappa(\mathbf{x}, \mathbf{x}'))$. As a non-parametric model, it's characterized by a mean function, here assumed to be zero, and a covariance function or kernel $\kappa(\mathbf{x}, \mathbf{x}')$, which is a positive definite function that measures the similarity of inputs. The choice of the kernel is crucial as it determines the types of functions the GP can model. For instance, the Matérn kernel

$$\kappa_{\text{Matérn}} = \sigma^2 \frac{\left(\frac{\sqrt{2\nu}}{l}\alpha(\mathbf{x}, \mathbf{x}')\right)^\nu}{\Gamma(\nu)2^{\nu-1}} K_\nu \left(\frac{\sqrt{2\nu}}{l}\alpha(\mathbf{x}, \mathbf{x}')\right),$$

(1)

and periodic kernel:

$$\kappa_{\text{periodic}} = \sigma^2 exp\left(-2\sin^2(\pi\alpha(\mathbf{x}, \mathbf{x}')/p)/l^2\right) \quad (2)$$

are versatile choices to model functions with non-linear and cyclical patterns, respectively. $\{\sigma^2, l, \nu, p\}$ are hyperparameters determining the variance, length-scale, smoothness, and periodicity of the function. $\alpha(\cdot, \cdot)$ is the Euclidean distance, and K_ν is the modified Bessel function, $\Gamma(\cdot)$ is the Gamma function.

Despite the flexibility and capacity, full GP is a computationally expensive model with $\mathcal{O}(n^3)$ inference cost while handlining n observation data, which is not feasible in practice. To sidestep expensive kernel matrix computation, (Hartikainen & Särkkä, 2010; Särkkä, 2013) applied the spectral analysis and worked out a crucial statement: a temporal GP with a stationary kernel is equivalent to a linear time-invariant stochastic differential equation (LTI-SDE). Specifically, given $f(t) \sim \mathcal{GP}(0, \kappa(t, t'))$, we can define a vector-valued companion form: $\mathbf{z}(t) =$

$\left(f(t), \frac{\mathrm{d}f(t)}{\mathrm{d}t}, \ldots, \frac{\mathrm{d}f^m(t)}{\mathrm{d}t}\right)^\top : t \to \mathcal{R}^{m+1}$, where m is the order of the derivative. Then, the GP is equivalent to the solution of the LTI-SDE with canonical form:

$$\frac{\mathrm{d}\mathbf{z}(t)}{\mathrm{d}t} = \mathbf{F}\mathbf{z}(t) + \mathbf{L}w(t), \quad (3)$$

where $\mathbf{F}$ is a $(m+1) \times (m+1)$ matrix, $\mathbf{L}$ is a $(m+1) \times 1$ vector, and $w(t)$ is a white noise process with spectral density $q_{\mathbf{s}}$. On arbitrary collection of timestamps $\{t_1, \ldots, t_N\}$, the LTI-SDE (3) can be further discretized as the Markov model with Gaussian transition, known as the state-space model (SSM):

$$\begin{aligned} p(\mathbf{z}(t_1)) &= \mathcal{N}(\mathbf{z}(t_1)|\mathbf{0}, \mathbf{P}_\infty), \\ p(\mathbf{z}(t_{n+1})|\mathbf{z}(t_n)) &= \mathcal{N}(\mathbf{z}(t_{n+1})|\mathbf{A}_n\mathbf{z}(t_n), \mathbf{Q}_n), \end{aligned} \quad (4)$$

where $\mathbf{A}_n = \exp(\mathbf{F}\Delta_n)$, $\mathbf{Q}_n = \int_{t_n}^{t_{n+1}} \mathbf{A}_n\mathbf{L}\mathbf{L}^\top\mathbf{A}_n^\top q_{\mathbf{s}}\mathrm{d}t$, $\Delta_n = t_{n+1} - t_n$, and $\mathbf{P}_\infty$ is the steady-state covariance matrix, which can be obtained by solving the Lyapunov equation $\mathbf{F}\mathbf{P}_\infty + \mathbf{P}_\infty\mathbf{F}^\top + \mathbf{L}\mathbf{L}^\top q_{\mathbf{s}} = 0$ (Lancaster & Rodman, 1995). SSM (4) can be efficiently solved with linear cost by classical methods like the Kalman filter (Kalman, 1960). After inference over $\mathbf{z}(t)$, we can easily obtain $f(t)$ by a simple projection: $f(t) = [1, 0, \ldots 0]\mathbf{z}(t)$.

We highlight that all the parameters of the LTI-SDE (3) and its SSM (4): $\{m, \mathbf{F}, \mathbf{L}, q_{\mathbf{s}}, \mathbf{P}_\infty\}$ are time-invariant constant and can be derived from the given stationary kernel function. In practice, stationary kernels are a common choice for GP, which requires the kernel to be a function of the distance between two inputs. For example, the Matérn and periodic kernels are stationary kernels, and we can work out their closed-form formulas of LTI-SDE and SSM. We omit the specific formulas here and refer the readers to the appendix.

4. Method

4.1. Functional Decomposition of Time Series

The motivation of *BayOTIDE* is based on the fact that the different channels of real-world multivariate time series are always correlated, and there may exist shared patterns across channels. Thus, we propose to decompose the series into a set of functional basis (factors) and channel-wise weights.

Inspired by the classic seasonal-trend decomposition (Cleveland et al., 1990) and TIDER (LIU et al., 2023), we assume there are two groups of factors representing different temporal patterns. The first group of factors is supposed to capture the nonlinear and long-term patterns, and the second represents the periodic parts, namely, trends and seasonalities. Thus, we decompose the function $\mathbf{X}(t) : t \rightarrow \mathbb{R}^D$ as the weighted combination of two groups of functional factors. Specifically, assume there are D_r trend factors and D_s seasonality factors, then we have the following decomposition:

$$\mathbf{X}(t) = \mathbf{U}\mathbf{V}(t) = [\mathbf{U}_{\text{trend}}, \mathbf{U}_{\text{season}}] \left[\begin{array}{c} \mathbf{v}_{\text{trend}}(t), \\ \mathbf{v}_{\text{season}}(t) \end{array} \right], \quad (5)$$

where $\mathbf{U} = [\mathbf{U}_{\text{trend}} \in \mathbb{R}^{D \times D_r}, \mathbf{U}_{\text{season}} \in \mathbb{R}^{D \times D_s}]$ are the weights of the combination. The trends factor group $\mathbf{v}_{\text{season}}(t) : t \rightarrow \mathbb{R}^{D_s}$, and seasonality factor groups $\mathbf{v}_{\text{trend}}(t) : t \rightarrow \mathbb{R}^{D_r}$ are the concatenation of independent temporal factors over each dimension:

$$\begin{aligned} \mathbf{v}_{\text{trend}}(t) &= \text{concat}[v_{\text{trend}}^i(t)]_{i=1\ldots D_r}, \\ \mathbf{v}_{\text{season}}(t) &= \text{concat}[v_{\text{season}}^j(t)]_{j=1\ldots D_s}, \end{aligned} \quad (6)$$

where $v_{\text{trend}}^i(t), v_{\text{season}}^j(t)$ are the 1-d temporal function factor, aimed to model the i-th trend and j-th seasonality latent pattern respectively.

For the imputation task, if we can estimate the $\mathbf{U}$ and $\mathbf{V}(t)$ well, we can impute the missing values of $\mathbf{X}(t)$ by $\mathbf{U}\mathbf{V}(t)$ for any t. As TIDER (LIU et al., 2023) proposed a low-rank decomposition similar to (5), our model can be seen as a generalization of TIDER to the continuous-time and functional field with the Bayesian framework.

4.2. GP Prior and Joint Probability of Our Model

We assume $\mathbf{X}(t)$ is partially observed with missing values and noise on timestamps $\{t_1, \ldots t_N\}$. We denote the observed values as $\mathbf{Y} = \{\mathbf{y}_n\}_{n=1}^N$, where $\mathbf{y}_n \in \mathbb{R}^D$, and its value at d-th channel is denoted as y_n^d. $\mathbf{M} \in \{0, 1\}^{D \times N}$ is the mask matrix, where 1 for observed values and 0 for missing values. The noise level is assumed to be the same for all the channels. Thus, we set the Gaussian likelihood for the observed values as:

$$p(\mathbf{Y}|\mathbf{U}, \mathbf{V}(t), \tau) = \prod_{(d,n) \in \Omega} \mathcal{N}(y_n^d \mid \mathbf{u}^d \mathbf{v}(t_n), \tau^{-1}), \quad (7)$$

where τ is the inverse of the noise level. Ω is the collection of observed values' location, namely $\Omega = \{(d, n) \mid \mathbf{M}_{d,n} = 1\}$. $\mathbf{u}^d \in \mathbb{R}^{1 \times (D_r + D_s)}$ is the d-th row of $\mathbf{U}$, and $\mathbf{v}(t_n) \in \mathbb{R}^{(D_r + D_s) \times 1}$ is the concatenation of $\mathbf{v}_{\text{trend}}(t_n)$ and $\mathbf{v}_{\text{season}}(t_n)$.

As $\mathbf{v}_{\text{season}}(t)$ and $\mathbf{v}_{\text{trend}}(t)$ are supposed to capture different temporal patterns, we adopt Gaussian Processes (GPs) with

different kernels to model them. Specifically, we use the Matérn kernel to model the trend factors, and the periodic kernel to model the seasonality factors:

$$v_{\text{trend}}^i(t) \sim \mathcal{GP}(0, \kappa_{\text{Matérn}}), v_{\text{season}}^j(t) \sim \mathcal{GP}(0, \kappa_{\text{periodic}}). \quad (8)$$

We further assume that Gaussian prior for $\mathbf{u}^d$ and Gamma prior for τ. Then, the joint probability model is:

$$p(\mathbf{Y}, \mathbf{V}(t), \mathbf{U}, \tau) = \text{Gam}(\tau \mid a_0, b_0) \prod_{d=1}^D \mathcal{N}(\mathbf{u}^d \mid \mathbf{0}, \mathbf{I})$$

$$\cdot \prod_{j=1}^{D_s} \mathcal{GP}(0, \kappa_{\text{periodic}}) \prod_{i=1}^{D_r} \mathcal{GP}(0, \kappa_{\text{Matérn}}) \cdot p(\mathbf{Y}|\Theta), \quad (9)$$

where $\Theta = \{\mathbf{U}, \mathbf{V}(t), \tau\}$ denotes all model random variables for compactness. As each GP prior term corresponds to a companion form $\mathbf{z}(t)$ (see section(3.2)), we define the concatenation of all factors' companion forms as $\mathbf{Z}(t)$ and we have $p(\mathbf{V}(t)) = p(\mathbf{Z}(t)) = P(\mathbf{Z}(t_1)) \prod_{i=1}^{N-1} P(\mathbf{Z}(t_{n+1})|\mathbf{Z}(t_n))$.

4.3. Online Inference

With the joint probability (9), we further propose an online inference algorithm to estimate the running posterior of Θ. We denote all observations up to time t_n as $\mathcal{D}_{t_n}$, i.e. $\mathcal{D}_{t_n} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$. When a new observation $\mathbf{y}_{n+1}$ arrives at t_{n+1}, we aimed to update the posterior distribution $p(\Theta \mid \mathcal{D}_{t_n} \cup \mathbf{y}_{n+1})$ without reusing the previous observations $\mathcal{D}_{t_n}$. The general principle for online inference is the incremental version of Bayes'rule, which is:

$$p(\Theta \mid \mathcal{D}_{t_n} \cup \mathbf{y}_{n+1}) \propto p(\mathbf{y}_{n+1} \mid \Theta, \mathcal{D}_{t_n})p(\Theta \mid \mathcal{D}_{t_n}). \quad (10)$$

However, the exact posterior is not tractable. Thus, we first apply the mean-field factorization to approximate the posterior. Specifically, we approximate the exact posterior as:

$$p(\Theta \mid \mathcal{D}_{t_n}) \approx q(\tau \mid \mathcal{D}_{t_n}) \prod_{d=1}^D q(\mathbf{u}^d \mid \mathcal{D}_{t_n})q(\mathbf{Z}(t) \mid \mathcal{D}_{t_n}), \quad (11)$$

where $q(\mathbf{u}^d \mid \mathcal{D}_{t_n}) = \mathcal{N}(\mathbf{m}_n^d, \mathbf{V}_n^d)$ and $q(\tau \mid \mathcal{D}_{t_n}) = \text{Gamma}(\tau \mid a_n, b_n)$ are the approx. distributions for $\mathbf{U}$ and τ at time t_n respectively. For $\mathbf{Z}(t)$, we design the approximated posterior as $q(\mathbf{Z}(t) \mid \mathcal{D}_{t_n}) = \prod_{i=1}^n q(\mathbf{Z}(t_i))$, where $q(\mathbf{Z}(t_i))$ are the concatenation of $q(\mathbf{z}(t_i)) = \mathcal{N}(\mu_i, \mathbf{S}_i)$ across all factors. $\{\{\mathbf{m}_n^d, \mathbf{V}_n^d\}, \{\mu_i, \mathbf{S}_i\}, a_n, b_n, \}$ are the variational parameters of the approximated posterior to be estimated.

With mean-field approximation, (10) is still not feasible. It's because the multiple factors and weights interweave in the

likelihood, and R.H.S of (10) is unnormalized. To solve it, we propose a novel online approach to update $q(\Theta \mid \mathcal{D}_{t_n})$ with a closed form by adopting conditional Expectation Propagation(CEP) (Wang & Zhe, 2019) and chain structure of $\mathbf{Z}(t)$. Specifically, with current $q(\Theta \mid \mathcal{D}_{t_n})$ and prior $p(\Theta)$, we can approximate each likelihood term of the new-arriving observation y_{n+1} as several messages factors:

$$p(y_{n+1}^d \mid \Theta) \approx \mathcal{Z} f_{n+1}^d(\mathbf{Z}(t_{n+1})) f_{n+1}^d(\mathbf{u}_d) f_{n+1}^d(\tau), \quad (12)$$

where $\mathcal{Z}$ is the normalization constant, $f_{n+1}^d(\mathbf{u}_d) = \mathcal{N}(\mathbf{u}_d \mid \hat{\mathbf{m}}_{n+1}^d, \hat{\mathbf{V}}_{n+1}^d)$ and $f_{n+1}^d(\tau) = \mathrm{Gamma}(\tau \mid \hat{a}_{n+1}, \hat{b}_{n+1})$, $f_{n+1}^d(\mathbf{Z}(t_{n+1})) = \mathrm{concat}[\mathcal{N}(\hat{\mu}_i, \hat{S}_i)]$ are the approximated message factors of $\mathbf{u}_d$, τ, and $\mathbf{Z}(t_{n+1})$ respectively.

Then, we merge all the message factors of $\mathbf{U}$ and τ and follow the variational form of (10), and will obtain the update equations:

$$q(\tau|\mathcal{D}_{t_{n+1}}) = q(\tau|\mathcal{D}_{t_n}) \prod_{d=1}^{D} f_{n+1}^d(\tau), \quad (13)$$

$$q(\mathbf{u}^d|\mathcal{D}_{t_{n+1}}) = q(\mathbf{u}^d|\mathcal{D}_{t_n}) f_{n+1}^d(\mathbf{u}^d). \quad (14)$$

The message approximation (12) and the message merging procedure (13)(14) are based on the conditional moment-matching technique, which can be done in parallel for all channels with closed-form update. We omit the derivation and exact formulas and refer the readers to the appendix.

Then, we present the online update of $\mathbf{Z}(t)$. With the chain structure of $q(\mathbf{Z}(t) \mid \mathcal{D}_{t_n})$ and $p(\mathbf{Z}(t))$, we found the update can be conducted sequentially. Specifically:

$$q(\mathbf{Z}(t_{n+1})) =$$
$$q(\mathbf{Z}(t_n)) p(\mathbf{Z}(t_{n+1})|\mathbf{Z}(t_n)) \prod_{d=1}^{} f_{n+1}^d(\mathbf{Z}(t_{n+1})), \quad (15)$$

where $p(\mathbf{Z}(t_{n+1})|\mathbf{Z}(t_n))$ is the concatenation of all factors' transition given in (4). If we treat $\prod_d f_{n+1}^d(\mathbf{Z}(t_{n+1}))$ as the observation of the state space, (15) is the Kalman filter model (Kalman, 1960). Thus, we can obtain the closed-form update of $q(\mathbf{Z}(t)| \mid \mathcal{D}_{t_n})$, which is the running posterior $q(\mathbf{Z}_n|\mathbf{y}_{1:n})$. After going through all observations, we run Rauch-Tung-Striebel (RTS) smoother (Rauch et al., 1965) to efficiently compute the full posterior of each state $q(\mathbf{Z}_n|\mathbf{y}_{1:N})$ from backward.

The online algorithm is summarized in Algorithm table 1: we go through all the timestamps, approximate the message factors with moment-matching, and run Kalman filter and message merging and update sequentially. For each timestamp, we can run moment-matching and posterior update steps iteratively several times with damping trick (Minka, 2001a) for better approximation. The algorithm is very efficient as the message approximation (12) can be parallel for

---

Algorithm 1 *BayOTIDE*

Input: observation $\mathbf{Y} = \{\mathbf{y}_n\}_{n=1}^N$ over $\{t_n\}_{n=1}^N$, D_s, D_r, the kernel hyperparameters.
Initialize $q(\tau), q(\mathcal{W}), \{q(\mathbf{Z}(t_n))\}_{n=1}^N$.
for $t = 1$ **to** N **do**
    Approximate messages by (12) for all observed channels in parallel.
    Update posterior of τ and $\mathbf{U}$ by (13) and (14) for all observed channels in parallel.
    Update posteriors of $\mathbf{Z}(t)$ using Kalman filter by (15).
end for
Run RTS smoother to obtain the full posterior of $\mathbf{Z}(t)$.
Return: $q(\tau), q(\mathcal{W}), \{q(\mathbf{Z}(t_n))\}_{n=1}^N$

---

different channels, and all the updates are closed-form. The algorithm is with time cost $\mathcal{O}(N(D_s + D_r))$ and space cost $\mathcal{O}(\sum_{k=1}^{D_r+D_s} N(m_k + m_k^2) + D(D_s + D_r))$ where N is the number of timestamps, D is the number of channel of original time series, D_r, D_s are number of trends and seasonality factors respectively, m_k is the order of the companion form of k-th factor's GP prior determined by the kernel types.

4.4. Probabilistic Imputation at Arbitrary Timestamp

With the current posterior $\{q(\mathbf{z}(t_1)) \ldots q(\mathbf{z}(t_N))\}$ over the observed timestamps, the functional and chain property of GP priors allow us to infer the prediction distribution, namely do probabilistic interpolation at arbitrary time stamps. Specifically, for a never-seen timestamp $t^\star \in (t_1, \mathbf{t}_N)$, we can identify the closest neighbor of $t^\star$ observed in training, i.e, $t_k < t^\star < t_{k+1}$, where $t_k, t_{k+1} \in \{t_1 \ldots t_N\}$. Then the predictive distribution at $t^\star$ is given by $q(\mathbf{z}(t^\star)) = \mathcal{N}(\mathbf{z}^\star \mid \mathbf{m}^\star, \mathbf{V}^\star)$. $\{\mathbf{m}^\star, \mathbf{V}^\star\}$ are defined as:

$$\mathbf{V}^\star = (\mathcal{Q}_1^{-1} + \mathcal{A}_2^\top \mathcal{Q}_1^{-1} \mathcal{A}_2)^{-1},$$
$$\mathbf{m}^\star = \mathbf{V}^\star (\mathcal{Q}_1^{-1} \mathcal{A}_1 \mathbf{m}_k + \mathcal{A}_2^\top \mathcal{Q}_2^{-1} \mathbf{m}_{k+1}), \quad (16)$$

where $\mathbf{m}_k, \mathbf{m}_{k+1}$ are the predictive mean of $q(\mathbf{z}(t_k)), q(\mathbf{z}(t_{k+1}))$, $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{Q}_1, \mathcal{Q}_2\}$ are transition matrices and covariance matrices based on the forward-backward transitions $p(\mathbf{z}(t^\star)|\mathbf{z}(t_k)), p(\mathbf{z}(t_{k+1})|\mathbf{z}(t^\star))$ respectively. Eq.(16) offers continuous modeling of the time series. We give a detailed derivation in the appendix.

5. Experiments

5.1. Synthetic data

We first evaluate *BayOTIDE* on a synthetic task. We first set four temporal functions and a weight matrix, defined as

follows:

$$\mathbf{U} = \begin{pmatrix} 1 & 1 & -2 & -2 \\ 0.4 & 1 & 2 & -1 \\ -0.3 & 2 & 1 & -1 \\ -1 & 1 & 1 & 0.5 \end{pmatrix},$$

$$\mathbf{V}(t) = \begin{pmatrix} 10t, \\ \sin(20\pi t), \\ \cos(40\pi t), \\ \sin(60\pi t) \end{pmatrix}.$$

Then, the four-channel time series is defined as $\mathbf{X}(t) = \mathbf{UV}(t)$, and each channel is a mixture of multiscale trend and seasonality factors. We collected 2000 data points over the 500 irregularly sampled timestamps from $[0, 1]$. We randomly set only 20% of the data as observed values, and the rest as missing for evaluation. We further add Gaussian noise with a standard deviation 0.1 to the observed data. We use the Matérn kernel with $\nu = 3/2$ as the trend kernel and the periodic kernel with period 20π as the seasonality kernel. We set $D_r = 1, D_s = 3$. We highlight that evaluation could be taken on the timestamps that never appear in the training set, known as the all-channel-missing imputation, but we can apply (16) to handle such hard cases easily.

The imputation results are shown in Figure 1a. We can see that *BayOTIDE* recovers the series well, and the estimated uncertainty is reasonable. We also show the channel-wise estimated factors in Figure 1b,1c,1d, and 1e. We can see that the estimated factors are close to the real factors, which indicates that *BayOTIDE* can capture the underlying multiscale patterns of the data.

5.2. Real-world Applications

Datasets We evaluate *BayOTIDE* on three real-world datasets, *Traffic-Guangzhou*(Chen et al.): traffic speed records in Guangzhou with 214 channels and 500 timestamps. *Solar-Power*(https://www.nrel.gov/grid/solar-power-data.html) : 137 channels and 52560 timestamps, which records the solar power generation of 137 PV plants. *Uber-Move*(https://movement.uber.com/): 7489 channels and 744 timestamps, recording the average movement of Uber cars along with the road segments in London, Jan 2020. For each dataset, we randomly sample {70%, 50%} of the available data points as observations for model training, and the rest for evaluation. The data process and split strategy are aligned with TIDER (LIU et al., 2023).

Baselines and Settings To the best of our knowledge, there are no online algorithms for multivariate time series imputation. Thus, we set several popular deterministic and probabilistic offline imputation approaches as baselines. The deterministic group includes: (1) *SimpleMean* (Acuna & Rodriguez, 2004), impute with column-wise mean values.

(2) *BRITS* (Cao et al., 2018), the RNN-based model for imputation with time decay (3) *NAOMI*(Liu et al., 2019), a Bidirectional RNN model build with adversarial training (4) *SAITS*(Du et al., 2023), a transformer-based model which adopts the self-attention mechanism. (5) *TIDER*(LIU et al., 2023). State of art deterministic imputation model based on disentangled temporal representations.

The probabilistic group includes: (1) *Multi-Task GP*(Bonilla et al., 2008), the classical multi-output Gaussian process model (2) *GP-VAE*(Fortuin et al., 2020), a deep generative model which combines Gaussian Processes(GP) and variational autoencoder(VAE) for imputation (3) *CSDI*(Tashiro et al., 2021) Famous probabilistic approach which apply conditional diffusion model to capture the temporal dependency. (4)*CSBI* Advanced diffusion method that models the imputation task as a conditional Schrödinger Bridge(SB)(Chen et al., 2023). We also set *BayOTIDE-fix-wight* by fixing all weight values as one and *BayOTIDE-trend-only*, and only using trend factor, respectively for *BayOTIDE* .

For most baselines, we use the released implementation provided by the authors. We partially use the results of deterministic methods reported in *TIDER*(LIU et al., 2023), as the setting is aligned. To avoid the out-of-memory problem of diffusion-based and deep-based baselines, we follow the preprocess of original papers, which split the whole sequence into small patches and subsample the channels for those methods.

For *BayOTIDE* , we implemented it by Pytorch and fine-tuned the D_s, D_r, and the kernel hyperparameters to obtain optimal results. Detailed information on hyperparameter settings is provided at Table 5 in the appendix. For the metrics, we use the mean absolute error (MAE) and the root mean squared error (RMSE) as the deterministic evaluation metrics for all methods. We adopt the continuous ranked probability score (CRPS) and the negative log-likelihood (NLLK), for *BayOTIDE* and all probabilistic baselines. We use 50 samples from the posterior to compute CRPS and NLLK. We repeat all the experiments 5 times and report the average results. For the CEP step at each timestamp in *BayOTIDE* , we use the *damping trick* (Minka, 2001a) in several inner epochs to avoid numerical instability. For the online imputation, we use the online update equations to obtain the imputation results and run RTS smoother at every evaluation timestamp.

Deterministic and Probabilistic performance Table 2 and table 3 show the RMSE, MAE, and CRPS scores of imputation on three datasets with observed ratio = 50% and = 70% respectively. We can see that *BayOTIDE* , an online method that only processes data once, beats the offline baselines and performs best in most cases. *TIDER* is the sub-optimal method for most cases. For probabilistic approaches, diffusion-based *CSDI* and *CSBI* obtain fair per-
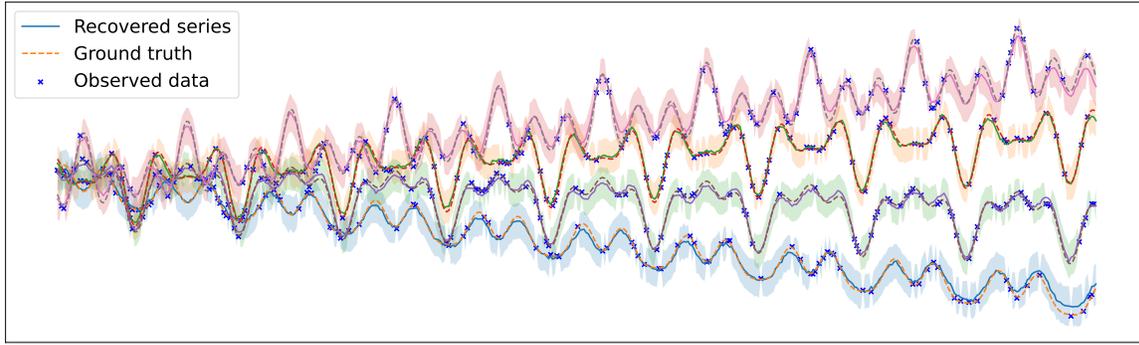
(a) Imputation results of the four-channel synthetic time series.
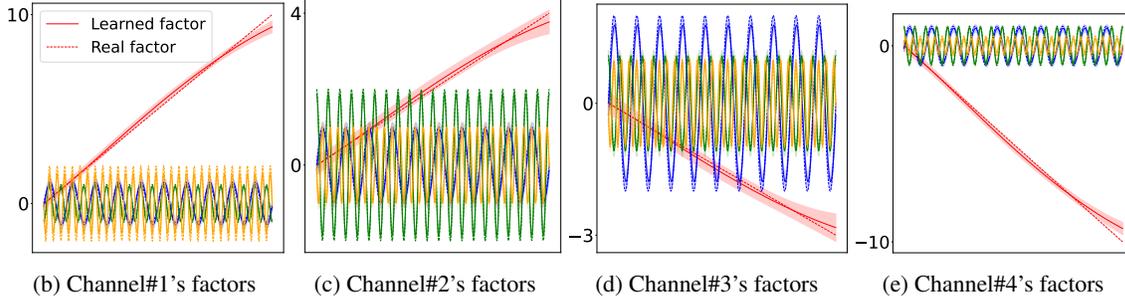
(b) Channel#1's factors (c) Channel#2's factors (d) Channel#3's factors (e) Channel#4's factors

Figure 1: (a): The multivariate time series recovered from observations. The shaded region indicates two posterior standard deviations. (b)-(e): The weighted trend-seasonality factors learned by *BayOTIDE* of each channel.

Observed-ratio=50%	*Traffic-GuangZhou*			*Solar-Power*			*Uber-Move*		
Metrics	RMSE	MAE	CRPS	RMSE	MAE	CRPS	RMSE	MAE	CRPS
Deterministic & Offline									
SimpleMean	9.852	7.791	-	3.213	2.212	-	5.183	4.129	-
BRITS	4.874	3.335	-	2.842	1.985	-	2.180	1.527	-
NAOMI	5.986	4.543	-	2.918	2.112	-	2.343	1.658	-
SAITS	4.839	3.391	-	2.791	1.827	-	1.998	1.453	-
TIDER	4.708	3.469	-	**1.679**	0.838	-	1.959	1.422	-
Probabilistic & Offline									
Multi-Task GP	4.887	3.530	0.092	2.847	1.706	0.203	3.625	2.365	0.121
GP-VAE	4.844	3.419	0.084	3.720	1.810	0.368	5.399	3.622	0.203
CSDI	4.813	3.202	0.076	2.276	0.804	0.166	1.982	1.437	0.072
CSBI	4.790	3.182	0.074	2.097	1.033	0.153	1.985	1.441	0.075
Probabilistic & Online									
BayOTIDE-fix weight	11.032	9.294	0.728	5.245	2.153	0.374	5.950	4.863	0.209
BayOTIDE-trend only	4.188	2.875	0.059	1.789	0.791	0.132	2.052	1.464	0.067
BayOTIDE	**3.820**	**2.687**	**0.055**	1.699	**0.734**	**0.122**	**1.901**	**1.361**	**0.062**

Table 2: RMSE, MAE and CRPS scores of imputation results of all methods on three datasets with observed ratio $= 50\%$.

Observed-ratio=70%	*Traffic-GuangZhou*			*Solar-Power*			*UberLondon*		
Metrics	RMSE	MAE	CRPS	RMSE	MAE	CRPS	RMSE	MAE	CRPS
Deterministic & Offline									
SimpleMean	10.141	8.132	-	3.156	2.319	-	5.323	4.256	-
BRITS	4.416	3.003	-	2.617	1.861	-	2.154	1.488	-
NAOMI	5.173	4.013	-	2.702	2.003	-	2.139	1.423	-
SAITS	4.407	3.025	-	2.359	1.575	-	1.893	1.366	-
TIDER	4.168	3.098	-	1.676	0.874	-	1.867	1.354	-
Probabilistic & Offline									
Multi-Task GP	4.471	3.223	0.082	2.618	1.418	0.189	3.159	2.126	0.108
GP-VAE	4.373	3.156	0.075	3.561	1.723	0.331	3.133	2.005	0.625
CSDI	4.301	2.991	0.069	2.132	1.045	0.153	1.886	1.361	0.068
CSBI	4.201	2.955	0.064	1.987	0.926	0.138	1.899	1.353	0.070
Probabilistic & Online									
BayOTIDE-fix weight	13.319	9.29	0.677	5.238	2.026	0.388	5.889	4.849	0.208
BayOTIDE-trend only	4.002	2.759	0.056	1.651	0.712	0.124	2.015	1.438	0.065
BayOTIDE	**3.724**	**2.611**	**0.053**	**1.621**	**0.709**	**0.116**	**1.832**	**1.323**	**0.061**

Table 3: RMSE, MAE and CRPS scores of imputation results of all methods on three datasets with observed ratio = 70%.

formance, but are costly in memory and need to split the long sequence into small patches for training. *BayOTIDE-fix-wight* is with poor performance, which indicates that the weighted bases mechanism is crucial. *BayOTIDE-trend-only* is slightly worse than *BayOTIDE* , showing the modeling of periodic factor is necessary. The results of NLLK score over three datasets can be found at table 6 in the appendix.
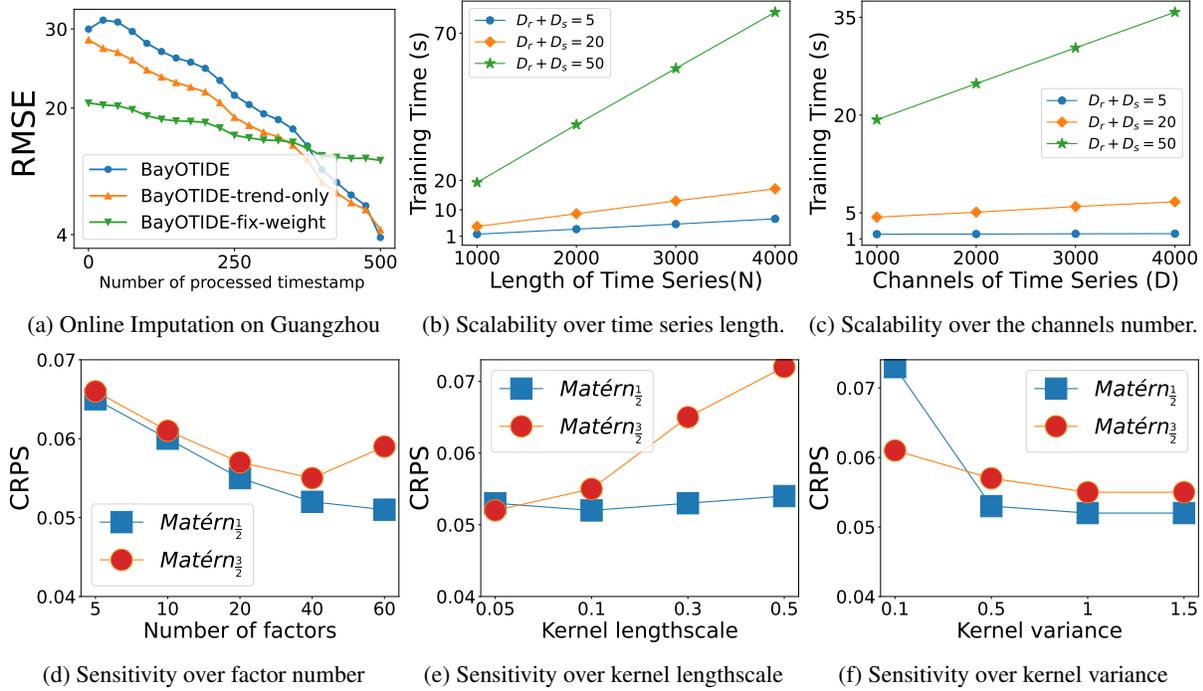
Online Imputation We demonstrate the online imputation performance of *BayOTIDE* on three datasets with observed ratio 50%. Whenever a group of observations at new timestamps have been sequentially fed into the model, we evaluate the test RMSE of the model with the updated weights and temporal factor. We compare the performance of *BayOTIDE* with the *BayOTIDE-fix-wight*. The online result on *Traffic-Guangzhou* is shown in Figure 2a. We can see that *BayOTIDE* shows the reasonable results that the evaluation error drops gradually when more timestamps are processed, meaning the model can continuously learn and improve. The performance of *BayOTIDE-fix-wight* is very poor. It indicates the trivial usage of the GP-SS model for multivariate time series imputation may not be feasible. The online results for the other two datasets can be found in the appendix.

Scalability and Sensitivity We evaluate the scalability of *BayOTIDE* over data size and factor numbers. We set three different factor numbers: $D_r + D_s = \{5, 20, 50\}$. As for the scalability over series length N, We make synthetic data with channel number $D = 1000$, increase the N from 1000 to 4000, and measure the training time. The result is shown in Figure 2b. Similarly, we fix the series length $N = 1000$, increase the series channel D from 1000 to 4000, and then measure the training time. The result is shown in Figure 2c. As we can see, the running time of *BayOTIDE* grows linearly in both channel and length size, and the factor number determines the slope. Therefore, *BayOTIDE* enjoys the linear scalability to the data size, which is suitable for large-scale applications.

We further examine the sensitivity of *BayOTIDE* with different hyperparameters. We build the model with Matérn kernel with different smoothness $\nu = \{1/2, 3/2\}$ on the *Traffic-Guangzhou* with observed ratio 70%. We vary three crucial hyperparameters: the number of factors $D_r + D_s$, the kernel length scale, and the kernel variance, and check how imputation performance (CRPS) changes. The results are shown in Figure 2d, Figure 2e, and Figure 2f, respectively. We found more factors that will result in better performance in general for both kernel types. We can find that the performance is relatively stable over different hyperparameters for Matérn kernel with smoothness $\nu = \{1/2\}$. For the Matérn kernel with smoothness $\nu = \{3/2\}$, the performance is more sensitive, especially to the kernel lengthscale. The smaller kernel lengthscale will result in better performance. The kernel variance is also sensitive, but the performance is relatively stable when the variance is large enough. More results on the sensitivity of *BayOTIDE* over different combinations of latent space dimensions (D_r, D_s) are shown in Table 7 in the appendix.

Imputation on Irregular and All-Channel-Missing Timestamps We further show *BayOTIDE* can work well with irregulate timestamps with functional and continuous design, and therefore can handle the challenging case of all-channel-missing case. We select the observations at $\{50\%, 70\%\}$ randomly sampled irregulate timestamps for

(a) Online Imputation on Guangzhou (b) Scalability over time series length. (c) Scalability over the channels number.

(d) Sensitivity over factor number (e) Sensitivity over kernel lengthscale (f) Sensitivity over kernel variance

Figure 2: Online performance, scalability and sensitivity of *BayOTIDE*

training, and evaluate the model on the left all-channel-missing timestamps. Such timestamps are never seen in the training, and the model needs to do the interpolation. We highlight that most existing advanced imputation methods cannot handle this hard case well. It's because they are based on the regular-time-interval setting, which assumes there is at least one observation at every timestamp during the training. However, *BayOTIDE* can apply the Eq. (16) and give probabilistic imputation at the arbitrary continuous timestamp. Thus, we only list the results of *BayOTIDE* on three datasets in Table 4. We can see the performance is closed or even better than the standard imputation setting shown in Table 2.

Dataset	GuangZhou		Solar-Power		Uber-Move	
Obs.	50%	70%	50%	70%	50%	70%
RMSE	3.625	3.383	1.624	1.442	3.017	2.931
MAE	2.524	2.401	0.706	0.614	1.199	1.154
CRPS	0.051	0.046	0.121	0.113	0.311	0.302
NLLK	2.708	2.634	1.861	1.857	2.137	2.138

Table 4: The imputation results of *BayOTIDE* with settings of irregulate and all-channel-missing timestamps on three datasets with observed ratio $= \{50\%, 70\%\}$.

6. Conclusion

We proposed *BayOTIDE* , a novel Bayesian model for online multivariate time series imputations. We decompose the multivariate time series into a temporal function basis and channel-wise weights, and apply a group of GPs to fit the temporal function basis. An efficient online inference algorithm is developed based on the SDE representation of GPs and moment-matching techniques. Results on both synthetic and real-world datasets show that *BayOTIDE* outperforms the state-of-the-art methods in terms of both imputation accuracy and uncertainty quantification. We also evaluated the scalability and robustness of *BayOTIDE* on large-scale real-world datasets with different settings of hyperparameters and missing patterns. . In the future, we plan to extend *BayOTIDE* to handle more complex patterns and more challenging tasks, such as long-term forecasting with non-stationary patterns.

Impact Statement

This paper is dedicated to innovating time series imputation techniques to push the boundaries of time series analysis further. While our primary objective is to enhance imputation accuracy and computational efficiency, we are also mindful of the broader ethical considerations that accompany technological progress in this area. While immediate societal impacts may not be apparent, we acknowledge the importance of ongoing vigilance regarding the ethical use of these advancements. It is essential to continuously assess and address potential implications to ensure responsible development and application in various scenarios.

References

Abdollahi, H. A novel hybrid model for forecasting crude oil price based on time series decomposition. Applied energy, 267:115035, 2020.

Acuna, E. and Rodriguez, C. The treatment of missing values and its effect on classifier accuracy. In Classification, Clustering, and Data Mining Applications: Proceedings of the Meeting of the International Federation of Classification Societies (IFCS), Illinois Institute of Technology, Chicago, 15–18 July 2004, pp. 639–647. Springer, 2004.

Alcaraz, J. M. L. and Strodthoff, N. Diffusion-based time series imputation and forecasting with structured state space models. arXiv preprint arXiv:2208.09399, 2022.

Bandara, K., Hyndman, R. J., and Bergmeir, C. Mstl: A seasonal-trend decomposition algorithm for time series with multiple seasonal patterns. arXiv preprint arXiv:2107.13462, 2021.

Benavoli, A. and Corani, G. State space approximation of gaussian processes for time series forecasting. In Advanced Analytics and Learning on Temporal Data: 6th ECML PKDD Workshop, AALTD 2021, Bilbao, Spain, September 13, 2021, Revised Selected Papers 6, pp. 21–35. Springer, 2021.

Bonilla, E., Chai, K. M., and Williams, C. Multi-task Gaussian process prediction. In Advances in Neural Information Processing Systems 20. 2008.

Brakel, P., Stroobandt, D., and Schrauwen, B. Training energy-based models for time-series imputation. The Journal of Machine Learning Research, 14(1):2771–2797, 2013.

Cao, W., Wang, D., Li, J., Zhou, H., Li, L., and Li, Y. Brits: Bidirectional recurrent imputation for time series. Advances in neural information processing systems, 31, 2018.

Chen, W., Wang, W., Peng, B., Wen, Q., Zhou, T., and Sun, L. Learning to rotate: Quaternion transformer for complicated periodical time series forecasting. In Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, pp. 146–156, 2022.

Chen, W., Liang, Y., Zhu, Y., Chang, Y., Luo, K., Wen, H., Li, L., Yu, Y., Wen, Q., Chen, C., et al. Deep learning for trajectory data management and mining: A survey and beyond. arXiv preprint arXiv:2403.14151, 2024.

Chen, X., Chen, Y., and He, Z. Urban traffic speed dataset of guangzhou, china, march 2018. URL https://doi.org/10.5281/zenodo, 1205229.

Chen, Y., Deng, W., Fang, S., Li, F., Yang, N. T., Zhang, Y., Rasul, K., Zhe, S., Schneider, A., and Nevmyvaka, Y. Provably convergent schr\" odinger bridge with applications to probabilistic time series imputation. arXiv preprint arXiv:2305.07247, 2023.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., and Terpenning, I. Stl: A seasonal-trend decomposition. J. Off. Stat, 6(1):3–73, 1990.

Du, W., Côté, D., and Liu, Y. Saits: Self-attention-based imputation for time series. Expert Systems with Applications, 219:119619, 2023.

Durbin, J. and Koopman, S. J. Time series analysis by state space methods, volume 38. OUP Oxford, 2012.

Esling, P. and Agon, C. Time-series data mining. ACM Computing Surveys (CSUR), 45(1):1–34, 2012.

Fang, C. and Wang, C. Time series data imputation: A survey on deep learning approaches. arXiv preprint arXiv:2011.11347, 2020.

Fang, S., Narayan, A., Kirby, R., and Zhe, S. Bayesian continuous-time tucker decomposition. In International Conference on Machine Learning, pp. 6235–6245. PMLR, 2022.

Fang, S., Yu, X., Li, S., Wang, Z., Kirby, M., and Zhe, S. Streaming factor trajectory learning for temporal tensor decomposition. Advances in Neural Information Processing Systems, 36, 2024.

Fortuin, V., Baranchuk, D., Rätsch, G., and Mandt, S. Gp-vae: Deep probabilistic time series imputation. In International conference on artificial intelligence and statistics, pp. 1651–1661. PMLR, 2020.

Grabner, M., Wang, Y., Wen, Q., Blavzivc, B., and vStruc, V. A global modeling framework for load forecasting in distribution networks. IEEE Transactions on Smart Grid, 2023.

Hartikainen, J. and Särkkä, S. Kalman filtering and smoothing solutions to temporal gaussian process regression models. In 2010 IEEE international workshop on machine learning for signal processing, pp. 379–384. IEEE, 2010.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020.

Kalman, R. E. A new approach to linear filtering and prediction problems. 1960.

Lancaster, P. and Rodman, L. Algebraic riccati equations. Clarendon press, 1995.

Li, L., Su, X., Zhang, Y., Lin, Y., and Li, Z. Trend modeling for traffic time series analysis: An integrated study. IEEE Transactions on Intelligent Transportation Systems, 16 (6):3430–3439, 2015.

Liu, H., Ma, Z., Yang, L., Zhou, T., Xia, R., Wang, Y., Wen, Q., and Sun, L. Sadi: A self-adaptive decomposed interpretable framework for electric load forecasting under extreme events. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE, 2023.

LIU, S., Li, X., Cong, G., Chen, Y., and JIANG, Y. Multivariate time-series imputation with disentangled temporal representations. In The Eleventh International Conference on Learning Representations, 2023.

Liu, Y., Yu, R., Zheng, S., Zhan, E., and Yue, Y. Naomi: Non-autoregressive multiresolution sequence imputation. Advances in neural information processing systems, 32, 2019.

Minka, T. P. Expectation propagation for approximate bayesian inference. In Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence, pp. 362–369, 2001a.

Minka, T. P. A family of algorithms for approximate Bayesian inference. PhD thesis, Massachusetts Institute of Technology, 2001b.

Mulyadi, A. W., Jun, E., and Suk, H.-I. Uncertainty-aware variational-recurrent imputation network for clinical time series. IEEE Transactions on Cybernetics, 52(9):9684–9694, 2021.

Qiu, J., Jammalamadaka, S. R., and Ning, N. Multivariate bayesian structural time series model. J. Mach. Learn. Res., 19(1):2744–2776, 2018.

Rasmussen, C. E. and Williams, C. K. I. Gaussian Processes for Machine Learning. MIT Press, 2006.

Rasul, K., Seward, C., Schuster, I., and Vollgraf, R. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In International Conference on Machine Learning, pp. 8857–8868. PMLR, 2021.

Rauch, H. E., Tung, F., and Striebel, C. T. Maximum likelihood estimates of linear dynamic systems. AIAA journal, 3(8):1445–1450, 1965.

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., and Aigrain, S. Gaussian processes for time-series modelling, 2013.

Särkkä, S. Bayesian filtering and smoothing. Number 3. Cambridge University Press, 2013.

Solin, A. and Särkkä, S. Explicit link between periodic covariance functions and state space models. In Artificial Intelligence and Statistics, pp. 904–912. PMLR, 2014.

Solin, A. et al. Stochastic differential equation methods for spatio-temporal gaussian process regression. 2016.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456, 2020.

Tashiro, Y., Song, J., Song, Y., and Ermon, S. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. Advances in Neural Information Processing Systems, 34:24804–24816, 2021.

Van Buuren, S. and Groothuis-Oudshoorn, K. mice: Multivariate imputation by chained equations in r. Journal of statistical software, 45:1–67, 2011.

Wang, J., Du, W., Cao, W., Zhang, K., Wang, W., Liang, Y., and Wen, Q. Deep learning for multivariate time series imputation: A survey. arXiv preprint arXiv:2402.04059, 2024a.

Wang, Z. and Zhe, S. Conditional expectation propagation. In UAI, pp. 6, 2019.

Wang, Z., Fang*, S., Li, S., and Zhe, S. Dynamic tensor decomposition via neural diffusion-reaction processes. Advances in Neural Information Processing Systems, 36, 2024b.

Wen, Q., Gao, J., Song, X., Sun, L., Xu, H., and Zhu, S. Robuststl: A robust seasonal-trend decomposition algorithm for long time series. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pp. 5409–5416, 2019.

Wen, Q., Zhang, Z., Li, Y., and Sun, L. Fast robuststl: Efficient and robust seasonal-trend decomposition for time series with complex patterns. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2203–2213, 2020.

Wen, Q., He, K., Sun, L., Zhang, Y., Ke, M., and Xu, H. Robustperiod: Robust time-frequency mining for multiple periodicity detection. In Proceedings of the 2021 international conference on management of data, pp. 2328–2337, 2021.

Wen, Q., Yang, L., Zhou, T., and Sun, L. Robust time series analysis and applications: An industrial perspective. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 4836–4837, 2022.

Woo, G., Liu, C., Sahoo, D., Kumar, A., and Hoi, S. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. arXiv preprint arXiv:2202.01575, 2022.

Yang, Y., Jin, M., Wen, H., Zhang, C., Liang, Y., Ma, L., Wang, Y., Liu, C., Yang, B., Xu, Z., et al. A survey on diffusion models for time series and spatio-temporal data. arXiv preprint arXiv:2404.18886, 2024.

Yao, W., Sun, Y., Ho, A., Sun, C., and Zhang, K. Learning temporally causal latent processes from general temporal data. arXiv preprint arXiv:2110.05428, 2021.

Yao, W., Chen, G., and Zhang, K. Temporally disentangled representation learning. Advances in Neural Information Processing Systems, 35:26492–26503, 2022.

Yoon, J., Jordon, J., and Schaar, M. Gain: Missing data imputation using generative adversarial nets. In International conference on machine learning, pp. 5689–5698. PMLR, 2018.

Zhang, K., Wen, Q., Zhang, C., Cai, R., Jin, M., Liu, Y., Zhang, J. Y., Liang, Y., Pang, G., Song, D., et al. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

Zhu, Z., Chen, W., Xia, R., Zhou, T., Niu, P., Peng, B., Wang, W., Liu, H., Ma, Z., Wen, Q., et al. eforecaster: Unifying electricity forecasting with robust, flexible, and explainable machine learning algorithms. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pp. 15630–15638, 2023.

Appendix

.1. LTI-SDE representation of GP with Matérn kernel and periodic kernel

.1.1. CONNECT GP WITH LTI-SDE BY SPECTRAL ANALYSIS

We take the Matérn kernel as an example to show how to connect GP with LTI-SDE. The Matérn kernel is defined as:

$$\kappa_\nu(t, t') = a \frac{(\frac{\sqrt{2\nu}}{\rho}\Delta)^\nu}{\Gamma(\nu)2^{\nu-1}} K_\nu(\frac{\sqrt{2\nu}}{\rho}\Delta) \qquad (17)$$

where $\Delta = |t - t'|$, $\Gamma(\cdot)$ is the Gamma function, $a > 0$ and $\rho > 0$ are the amplitude and length-scale parameters respectively, K_ν is the modified Bessel function of the second kind, and $\nu > 0$ controls the smoothness of sample paths from the GP prior $f(t) \sim \mathcal{GP}(0, \kappa_\nu(t, t'))$.

For a stationary Matérn kernel $\kappa_\nu(t, t') = \kappa_\nu(t - t')$, the energy spectrum density of $f(t)$ can be obtained via the Wiener-Khinchin theorem by taking the Fourier transform of $\kappa_\nu(\Delta)$:

$$S(\omega) = \frac{\sigma^2}{(\alpha^2 + \omega^2)^{m+1}} \qquad (18)$$

where ω is the frequency, $\alpha = \frac{\sqrt{2\nu}}{\rho}$, and we take $\nu = m + \frac{1}{2}$ for $m \in \{0, 1, 2, ...\}$.

Expanding the polynomial gives:

$$(\alpha + i\omega)^{m+1} = \sum_{k=0}^m c_k (i\omega)^k + (i\omega)^{m+1} \qquad (19)$$

where c_k are coefficients. This allows constructing an equivalent frequency domain system:

$$\sum_{k=1}^m c_k (i\omega)^k \widehat{f}(\omega) + (i\omega)^{m+1} \widehat{f}(\omega) = \widehat{\beta}(\omega) \qquad (20)$$

where $\widehat{f}(\omega)$ and $\widehat{\beta}(\omega)$ are Fourier transforms of $f(t)$ and white noise $w(t)$ with spectral density $q_{\mathbf{s}}$ respectively.

Taking the inverse Fourier transform yields the time domain SDE:

$$\sum_{k=1}^m c_k \frac{d^k f}{dt^k} + \frac{d^{m+1} f}{dt^{m+1}} = w(t) \qquad (21)$$

We can further construct a new state $\mathbf{z} = (f, f^{(1)}, \ldots, f^{(m)})^\top$ (where each $f^{(k)} \triangleq \mathrm{d}^k f / \mathrm{d}t^k$) and convert (21) into a linear time-invariant (LTI) SDE,

$$\frac{\mathrm{d}\mathbf{z}(t)}{\mathrm{d}t} = \mathbf{F}\mathbf{z}(t) + \mathbf{L}w(t) \qquad (22)$$

where

$$\mathbf{F} = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -c_0 & \ldots & -c_{m-1} & -c_m \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

The LTI-SDE is particularly useful in that its finite set of states follows a Gauss-Markov chain, namely the state-space prior. Specifically, given arbitrary $t_1 < \ldots < t_L$, we have

$$p(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_L)) = p(\mathbf{z}(t_1)) \prod_{k=1}^{L-1} p(\mathbf{z}(t_{k+1})|\mathbf{z}(t_k))$$

where

$$p(\mathbf{z}(t_1)) = \mathcal{N}(\mathbf{z}(t_1)|\mathbf{0}, \mathbf{P}_\infty),$$
$$p(\mathbf{z}(t_{n+1})|\mathbf{z}(t_n)) = \mathcal{N}(\mathbf{z}(t_{n+1})|\mathbf{A}_n\mathbf{z}(t_n), \mathbf{Q}_n) \qquad (23)$$

where $\mathbf{A}_n = \exp(\mathbf{F}\Delta_n)$, $\mathbf{Q}_n = \int_{t_n}^{t_{n+1}} \mathbf{A}_n \mathbf{L}\mathbf{L}^\top \mathbf{A}_n^\top q_\mathbf{s} \mathrm{d}t$, $\Delta_n = t_{n+1} - t_n$, and $\mathbf{P}_\infty$ is the steady-state covariance matrix of the LTI-SDE 3, which can be obtained by solving the Lyapunov equation $\mathbf{F}\mathbf{P}_\infty + \mathbf{P}_\infty \mathbf{F}^\top + \mathbf{L}\mathbf{L}^\top q_\mathbf{s} = 0$ (Lancaster & Rodman, 1995), as we claimed in the main paper.

Note that for other types of stationary kernel functions, such as the periodic kernels, we can approximate the inverse spectral density $1/S(\omega)$ with a polynomial of ω^2 with negative roots (Solin & Särkkä, 2014), and follow the same way to construct an LTI-SDE and state-space prior.

.1.2. THE CLOSED-FORM OF LTI-SDE AND STATE SPACE PRIOR WITH MATÉRN KERNEL AND PERIODIC KERNEL

With the canonical form of LTI-SDE (22) and state space prior(23) and above derivation, we can work out the closed-form of LTI-SDE and state space prior for Matérn kernel and periodic kernel. We present the results in the following.

For Matérn kernel with $m = 0$, indicating the smoothness is $\nu = 0 + \frac{1}{2}$, it becomes the exponential covariance function:

$$\kappa_{\exp}(\tau) = \sigma^2 \exp\left(-\frac{\tau}{\ell}\right) \qquad (24)$$

Then the parameters of the LTI-SDE and state space prior are: $\{m = 0, \mathbf{F} = -1/l, \mathbf{L} = 1, q_\mathbf{s} = 2\sigma^2/l, \mathbf{P}_\infty = \sigma^2\}$

For Matérn kernel with $m = 1$, indicating the smoothness is $\nu = 1 + \frac{1}{2} = 3/2$, the kernel becomes the Matérn 3/2

covariance function:

$$\kappa_{\text{Mat.}}(\tau) = \sigma^2 \left(1 + \frac{\sqrt{3}\tau}{\ell}\right) \exp\left(-\frac{\sqrt{3}\tau}{\ell}\right) \qquad (25)$$

and the parameters of the LTI-SDE and state space prior are: $m = 1$, $\mathbf{F} = \begin{pmatrix} 0 & 1 \\ -\lambda^2 & -2\lambda \end{pmatrix}$, $\mathbf{L} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $\mathbf{P}_\infty = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \lambda^2\sigma^2 \end{pmatrix}$, $q_\mathbf{s} = 4\lambda^3\sigma^2$, where $\lambda = \sqrt{3}/\ell$.

For the periodic kernel:

$$\kappa_{\text{periodic}}(t, t') = \sigma^2 \exp\left(-\frac{2\sin^2(\pi\Delta/p)}{l^2}\right) \qquad (26)$$

with preset periodicity p, (Solin & Särkkä, 2014) construct corresponding SDE by a sum of n two-dimensional SDE models(m=1) of the following parameters:

$$\mathbf{F}_j = \begin{pmatrix} 0 & -\frac{2\pi}{p}j \\ \frac{2\pi}{p}j & 0 \end{pmatrix}, \mathbf{L}_j = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad (27)$$

$\mathbf{P}_{\infty,j} = q_j^2 \mathbf{I}_2$, where $q_j^2 = 2\mathrm{I}_j\left(\ell^{-2}\right)/\exp\left(\ell^{-2}\right)$, for $j = 1, 2, \ldots, n$ and $q_0^2 = \mathrm{I}_0\left(\ell^{-2}\right)/\exp\left(\ell^{-2}\right)$ (Solin et al., 2016)

.2. Derivative of online update equations by conditional moment matching

.2.1. BRIEF INTRODUCTION OF EP AND CEP

The Expectation Propagation (EP) (Minka, 2001b) and Conditional EP (CEP) (Wang & Zhe, 2019) frameworks approximate complex probabilistic models with distributions in the exponential family.

Consider a model with latent variables $\boldsymbol{\theta}$ and observed data $\mathcal{D} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$. The joint probability is:

$$p(\boldsymbol{\theta}, \mathcal{D}) = p(\boldsymbol{\theta}) \prod_{n=1}^{N} p(\mathbf{y}_n|\boldsymbol{\theta}) \qquad (28)$$

The posterior $p(\boldsymbol{\theta}|\mathcal{D})$ is usually intractable. EP approximates each term with an exponential family distribution:

$$p(y_n|\boldsymbol{\theta}) \approx c_n f_n(\boldsymbol{\theta}) \qquad (29)$$
$$p(\boldsymbol{\theta}) \approx c_0 f_0(\boldsymbol{\theta}) \qquad (30)$$

where $f_n(\boldsymbol{\theta}) \propto \exp(\boldsymbol{\lambda}_n^\top \boldsymbol{\phi}(\boldsymbol{\theta}))$ are in the exponential family with natural parameters $\boldsymbol{\lambda}_n$ and sufficient statistics $\boldsymbol{\phi}(\boldsymbol{\theta})$.

The joint probability is approximated by:

$$p(\boldsymbol{\theta}, \mathcal{D}) \approx f_0(\boldsymbol{\theta}) \prod_{n=1}^{N} f_n(\boldsymbol{\theta}) \cdot \text{const} \qquad (31)$$

giving a tractable approximate posterior $q(\boldsymbol{\theta}) \approx p(\boldsymbol{\theta}|\mathcal{D})$.

EP optimizes the approximations f_n by repeatedly:

1) Computing the calibrated distribution $q^{\backslash n}$ excluding f_n.

2) Constructing the tilted distribution $\widetilde{p}$ incorporating the true likelihood.

3) Projecting $\widetilde{p}$ back to the exponential family by moment matching.

4) Updating $f_n \approx \frac{q^*}{q^{\backslash n}}$ where q^* is the projection.

The intractable moment matching in Step 3 is key. CEP exploits factorized $f_n = \prod_m f_{nm}(\boldsymbol{\theta}_m)$ with disjoint $\boldsymbol{\theta}_m$. It uses nested expectations:

$$\mathbb{E}_{\widetilde{p}}[\phi(\boldsymbol{\theta}_m)] = \mathbb{E}_{\widetilde{p}(\boldsymbol{\theta}_{\backslash m})}\mathbb{E}_{\widetilde{p}(\boldsymbol{\theta}_m|\boldsymbol{\theta}_{\backslash m})}[\phi(\boldsymbol{\theta}_m)] \qquad (32)$$

The inner expectation is tractable. For the outer expectation, CEP approximates the marginal tilted distribution with the current posterior:

$$\mathbb{E}_{\widetilde{p}(\boldsymbol{\theta}_{\backslash m})}[\mathbf{g}(\boldsymbol{\theta}_{\backslash m})] \approx \mathbb{E}_{q(\boldsymbol{\theta}_{\backslash m})}[\mathbf{g}(\boldsymbol{\theta}_{\backslash m})] \qquad (33)$$

If still intractable, the delta method is used to approximate the expectation with a Taylor expansion.

Once the conditional moment $\mathbf{g}(\boldsymbol{\theta}_{\backslash m})$ is obtained, CEP substitutes the expectation $\mathbb{E}_{q(\boldsymbol{\theta}_{\backslash m})}[\boldsymbol{\theta}_{\backslash m}]$ to compute the matched moment for constructing q^*.

Online inference Update

We then applied the EP and CEP to approximate the running posterior $p(\Theta \mid \mathcal{D}_{t_n} \cup \mathbf{y}_{n+1})$. With the incremental version of Bayes'rule (10), the key is to work out the close-form factors in the likelihood approximation (12). In other words, we adopt conditional moment match techniques to handle:

$$\mathcal{N}(\mathbf{y}_{n+1}^d \mid \mathbf{u}^d \mathbf{V}(t_{n+1}), \tau^{-1}) \approx$$
$$\mathcal{Z} f_{n+1}^d(\mathbf{Z}(t_{n+1})) f_{n+1}^d(\mathbf{u}_d) f_{n+1}^d(\tau) \qquad (34)$$

Then we follow the standard CEP procedure to compute the conditional moment of $\{\mathbf{Z}(t_{n+1}), \mathbf{u}_d, \tau\}$ and update $f_{n+1}^d(\mathbf{u}_d) = \mathcal{N}(\mathbf{u}_d \mid \hat{\mathbf{m}}_{n+1}^d, \hat{\mathbf{V}}_{n+1}^d)$ and $f_{n+1}^d(\tau) = \text{Gamma}(\tau \mid \hat{a}_{n+1}, \hat{b}_{n+1})$, $f_{n+1}^d(\mathbf{Z}(t_{n+1})) = \text{concat}[\mathcal{N}(\hat{\mu}_i, \hat{\mathbf{S}}_i)]$.

Specifically, for $f_{n+1}^d(\tau) = \text{Gamma}(\tau \mid \hat{a}_{n+1}, \hat{b}_{n+1})$ we have:

$$\hat{a}_{n+1} = \frac{1}{2} \qquad (35)$$
$$\hat{b}_{n+1} = \frac{1}{2}\mathbb{E}_q[(\mathbf{y}_{n+1}^d - \mathbf{u}^d \mathbf{V}(t_{n+1}))^2] \qquad (36)$$

For $f_{n+1}^d(\mathbf{u}_d) = \mathcal{N}(\mathbf{u}_d \mid \hat{\mathbf{m}}_{n+1}^d, \hat{\mathbf{V}}_{n+1}^d)$, we have:

$$\hat{\mathbf{V}}_{n+1}^d = [\mathbb{E}_q[\tau \cdot \mathbf{Z}(t_{n+1})\mathbf{Z}^T(t_{n+1})]]^{-1} \qquad (37)$$
$$\hat{\mathbf{m}}_{n+1}^d = \hat{\mathbf{V}}_{n+1}^d \cdot \mathbb{E}_q[\tau \mathbf{y}_{n+1}^d \mathbf{Z}(t_{n+1})] \qquad (38)$$

For $f_{n+1}^d(\mathbf{Z}(t_{n+1})) = \text{concat}[\mathcal{N}(\hat{\mu}_i, \hat{S}_i)] = \mathcal{N}(\hat{\mu}_\mathbf{i}, \hat{\mathbf{S}}_i)$, we have:

$$\hat{\mathbf{S}}_i = [\mathbb{E}_q[\tau \cdot \mathbf{u}_d \mathbf{u}_d^T]]^{-1} \qquad (39)$$
$$\hat{\mu}_\mathbf{i} = \hat{\mathbf{S}}_i \cdot \mathbb{E}_q[\tau \mathbf{y}_{n+1}^d \mathbf{u}_d] \qquad (40)$$

All the expectation is taken over the current approximated posterior $q(\Theta \mid \mathcal{D}_{t_n})$.

With these message factors from the new-arriving likelihood, the online update is easy. We follow the (13), (14) and (15) to merge the factors and obtain the closed-form online update for the global posterior.

.3. Derivation of the PROBABILISTIC IMPUTATION AT ARBITRARY TIME STAMPS

Consider a general state space model, which includes a sequence of states $\mathbf{x}_1, \ldots, \mathbf{x}_M$ and the observed data $\mathcal{D}$. The states are at time $t_1, \ldots, t_M$ respectively. The key of the state space model is that the prior of the states is a Markov chain. The joint probability has the following form,

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_M, \mathcal{D})$$
$$= p(\mathbf{x}_1) \prod_{j=1}^{M-1} p(\mathbf{x}_{j+1}|\mathbf{x}_j) \cdot p(\mathcal{D}|\mathbf{x}_1, \ldots, \mathbf{x}_M). \qquad (41)$$

Note that here we do not assume the data likelihood is factorized over each state, like those typically used in Kalman filtering. In our point process model, the likelihood often couples multiple states together.

Suppose we have run some posterior inference to obtain the posterior of these states $q(\mathbf{x}_1, \ldots, \mathbf{x}_M)$, and we can easily pick up the marginal posterior of each state and each pair of the states. Now we want to calculate the posterior distribution of the state at time t^* such that $t_m < t^* < t_{m+1}$. Denote the corresponding state by $\mathbf{x}^*$, our goal is to compute $p(\mathbf{x}^*|\mathcal{D})$. To do so, we consider incorporating $\mathbf{x}^*$ in the joint

probability (41),

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_m, \mathbf{x}^*, \mathbf{x}_{m+1}, \ldots, \mathbf{x}_M, \mathcal{D})$$

$$= p(\mathbf{x}_1) \prod_{j=1}^{m-1} p(\mathbf{x}_{j+1}|\mathbf{x}_j) \cdot p(\mathbf{x}^*|\mathbf{x}_m) p(\mathbf{x}_{m+1}|\mathbf{x}^*)$$

$$\cdot \prod_{j=m+1}^{M} p(\mathbf{x}_{j+1}|\mathbf{x}_j) \cdot p(\mathcal{D}|\mathbf{x}_1, \ldots, \mathbf{x}_M). \tag{42}$$

Now, we marginalize out $\mathbf{x}_{1:M\setminus\{m,m+1\}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{m-1}, \mathbf{x}_{m+2}, \ldots, \mathbf{x}_M\}$. Note that since $\mathbf{x}^*$ does not appear in the likelihood, we can take it out from the integral,

$$p(\mathbf{x}_m, \mathbf{x}_{m+1}, \mathbf{x}^*, \mathcal{D})$$

$$= \int p(\mathbf{x}_1) \prod_{j=1}^{m-1} p(\mathbf{x}_{j+1}|\mathbf{x}_j)$$

$$\prod_{j=m+1}^{M} p(\mathbf{x}_{j+1}|\mathbf{x}_j) \cdot p(\mathcal{D}|\mathbf{x}_1, \ldots, \mathbf{x}_M) \mathrm{d}\mathbf{x}_{1:M\setminus\{m,m+1\}}$$

$$\cdot p(\mathbf{x}^*|\mathbf{x}_m) p(\mathbf{x}_{m+1}|\mathbf{x}^*)$$

$$= \frac{p(\mathbf{x}_m, \mathbf{x}_{m+1}, \mathcal{D}) p(\mathbf{x}^*|\mathbf{x}_m) p(\mathbf{x}_{m+1}|\mathbf{x}^*)}{p(\mathbf{x}_{m+1}|\mathbf{x}_m)}. \tag{43}$$

Therefore, we have

$$p(\mathbf{x}_m, \mathbf{x}_{m+1}, \mathbf{x}^*|\mathcal{D}) \propto$$
$$p(\mathbf{x}_m, \mathbf{x}_{m+1}|\mathcal{D}) p(\mathbf{x}^*|\mathbf{x}_m) p(\mathbf{x}_{m+1}|\mathbf{x}^*). \tag{44}$$

Suppose we are able to obtain $p(\mathbf{x}_m, \mathbf{x}_{m+1}|\mathcal{D}) \approx q(\mathbf{x}_m, \mathbf{x}_{m+1})$. We now need to obtain the posterior of $\mathbf{x}^*$. In the LTI SDE model, we know that the state transition is a Gaussian jump. Let us denote

$$p(\mathbf{x}^*|\mathbf{x}_m) = \mathcal{N}(\mathbf{x}^*|\mathbf{A}_1\mathbf{x}_m, \mathbf{Q}_1)$$
$$p(\mathbf{x}_{m+1}|\mathbf{x}^*) = \mathcal{N}(\mathbf{x}_{m+1}|\mathbf{A}_2\mathbf{x}^*, \mathbf{Q}_2). \tag{45}$$

Then, we can simply merge the natural parameters of the two Gaussian and obtain

$$p(\mathbf{x}_m, \mathbf{x}_{m+1}, \mathbf{x}^*|\mathcal{D}) = p(\mathbf{x}_m, \mathbf{x}_{m+1}|\mathcal{D}) \mathcal{N}(\mathbf{x}^*|\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*), \tag{46}$$

where

$$(\boldsymbol{\Sigma}^*)^{-1} = \mathbf{Q}_1^{-1} + \mathbf{A}_2^{\top} \mathbf{Q}_2^{-1} \mathbf{A}_2,$$
$$(\boldsymbol{\Sigma}^*)^{-1} \boldsymbol{\mu}^* = \mathbf{Q}_1^{-1} \mathbf{A}_1 \mathbf{x}_m + \mathbf{A}_2^{\top} \mathbf{Q}_2^{-1} \mathbf{x}_{m+1}. \tag{47}$$

Those are the formulas for probabilistic imputation as arbitrary time stamps.

.4. More experimental results

The NLLK scores of probabilistic imputation approaches across all datasets with different observed ratios are shown in table 6. We can see that *BayOTIDE* , an online method that only processes data once, beats the offline baselines and achieves the best performance in all cases.

To further show the sensitivity of *BayOTIDE* over the different latent space dimensions, we evaluate our method on the Guangzhou-traffic dataset (observed ratio = 70%) with different settings of latent space dimension and evaluate the test CRPC score. The results are shown in table 7. We found that increasing the latent space dimension, especially the trend factor dimension (D_r), can improve the model performance. However, the improvement is not linear, and the model may suffer from overfitting when the latent space dimension is too large. A seasonal factor dimension (D_s) that is too high may also lead to overfitting and degrading the model performance.

For the online imputation, the results on the *Solar-Power* and *Uber-Move* is shown in Figure 3a and Figure 3b.

Observed-ratio=50%	*Traffic-GuangZhou*	*Solar-Power*	*Uber-Move*
Number of trend factor D_r	30	50	30
Number of seasonality factor D_s	10	5	5
Trend factor smoothness (ν)	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{3}{2}$
Trend factor lengthscale	0.1	0.001	0.1
Trend factor variance	1.0	1.0	1.0
Seasonal factor frequency ($2\pi p$)	15	10	15
Seasonal factor lengthscale	0.05	0.5	0.05
Damping epochs	5	2	5
Observed-ratio=70%	*Traffic-GuangZhou*	*Solar-Power*	*Uber-Move*
Number of trend factor D_r	30	50	30
Number of seasonality factor D_s	10	5	5
Trend factor smoothness (ν)	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
Trend factor lengthscale	0.1	0.0005	0.1
Trend factor variance	1.0	1.0	1.0
Seasonal factor frequency ($2\pi p$)	15	100	15
Seasonal factor lengthscale	0.05	0.5	0.05
Damping epochs	5	2	5

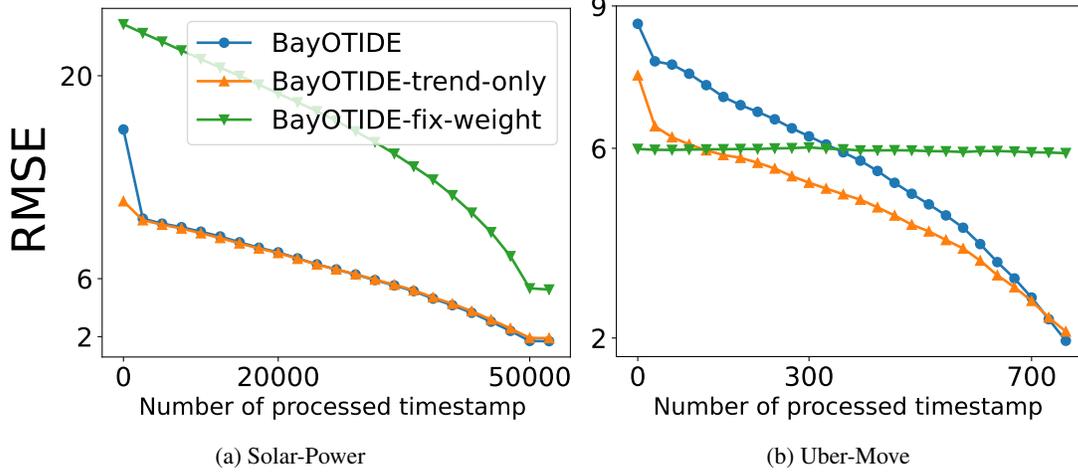Table 5: The hyperparameter setting of *BayOTIDE* for the imputation task.

(a) Solar-Power   (b) Uber-Move

Figure 3: Online imputation results on *Solar-Power* and *Uber-Move*.

Dataset	Traffic-GuangZhou		Solar-Power		Uber-Move	
Observed ratio	50%	70%	50%	70%	50%	70%
Probabilistic & Offline						
Multi-Task GP	7.339	6.921	4.921	4.292	4.426	4.027
GP-VAE	5.353	4.691	6.921	6.006	7.323	5.827
CSDI	3.942	3.518	3.433	2.921	2.415	2.322
CSBI	3.912	3.527	3.537	3.016	2.424	2.331
Probabilistic & Online						
BayOTIDE-fix weight	10.239	8.905	4.116	4.093	3.249	3.252
BayOTIDE-trend only	**2.897**	**2.852**	1.944	1.878	2.169	2.146
BayOTIDE	3.244	3.078	**1.885**	**1.852**	**2.167**	**2.100**

Table 6: The negative log-likelihood score (NLLK) of all probabilistic imputation methods on all datasets with observed ratio $= \{50\%, 70\%\}$

Test CRPC	$D_r = 5$	$D_r = 10$	$D_r = 20$	$D_r = 30$	$D_r = 40$
$D_s = 3$	0.068	0.064	0.061	0.057	0.056
$D_s = 5$	0.067	0.063	0.059	0.056	0.055
$D_s = 10$	0.066	0.063	0.058	0.053	0.053
$D_s = 20$	0.081	0.074	0.065	0.059	0.058

Table 7: The test CRPC on Guangzhou-traffic dataset(observed ratio = 70%) with different settings of latent space dimension.