Efficient Modeling of Irregular Time-Series with Stochastic Optimal Control

Byoungwoo Park[*], **Hyungi Lee**[*], **Juho Lee**
KAIST
{bw.park, lhk2708, juholee}@kaist.ac.kr

Abstract

Many real-world datasets, such as healthcare, climate, and economic data, are often collected as irregular time series, which pose significant challenges for modeling. Previous research has approached this problem in two main directions: 1) Transformer-based models and 2) dynamics-based models. Transformer-based models efficiently handle irregular time series with simple architectures and time encoding but struggle with long sequences and require many parameters due to the lack of inductive biases. Continuous dynamics-based models offer accurate Bayesian inference of dynamic states but suffer from the complexity of sequential computation, leading to increased computational costs scaling with the length of time intervals. To address these limitations, we propose Parallel Bayesian Diffusion Filtering (PBDF), a variational inference algorithm based on parallelizable stochastic differential equations and stochastic optimal control theory. PBDF combines the parallel inference capabilities of Transformer-based models with the Bayesian inference of continuous-discrete state space models. Through empirical evaluations on the USHCN and Physionet datasets for both interpolation and extrapolation tasks, we demonstrate PBDF's superior performance and computational efficiency.

1 Efficient Modeling with Stochastic Optimal Control

In this section, we explain our proposed model for irregular time series data, called PBDF. We start by introducing the Continuous-Discrete State Space Model (CD-SSM) [10] and formulate the variational inference problem for the state variables in Sec 1.1. Next, we discuss amortized inference for the auxiliary variables and detail the efficient learning and inference algorithm for PBDF in Sec 1.2.

1.1 Controlled Continuous-Discrete State Space Model.

Let us consider for a set of time steps (regular or irregular) $\{t_i\}_{i=0}^K$ over an interval $\mathcal{T} = [0, T]$, *i.e.*, $0 \leq t_1 \leq \cdots \leq t_K \leq T$. The CD-SSM assumes a continuous-time Markov state trajectory $\mathbf{X}_{0:T}$ in a latent space $\mathcal{X}$ as a solution of the stochastic differential equation (SDE):

$$d\mathbf{X}_t = b(t, \mathbf{X}_t)dt + d\mathbf{W}_t, \quad \text{where} \quad \mathbf{X}_0 \sim \mu_0 \tag{1}$$

and $\{\mathbf{W}_t\}_{t\in[0,T]}$ is a $\mathcal{X}$-valued Wiener process that is independent of the μ_0. Since $\mathbf{X}_t$ is a Markov process, we can describe the time-evolution of $\mathbf{X}_t$ by a transition kernel $\mathbf{P}_t$, and for any measurable event $A \subset \mathcal{X}$ and $\mathbf{x}_{t_{i-1}}$, transition kernel $\mathbf{P}_{t_i}$ for the time t_i can be computed as $\mathbf{P}_{t_i}(\mathbf{x}_{t_{i-1}}, A) = \int_A \mathbf{p}_{t_i}(\mathbf{x}_{t_{i-1}}, \mathbf{x}_{t_i})d\mathbf{x}_{t_i}$, where the transition density $\mathbf{p}_t$ is obtained by a solution of the Fokker-Planck equation associated with $\mathbf{X}_t$. By utilizing transition kernel $\mathbf{P}_{t_i}$s for all $i \in [1 : K]$, we can define a

[*]Equal contribution

Workshop on Bayesian Decision-making and Uncertainty, 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

product law of $\{\mathbf{X}_{t_i}\}_{i\in[0:K]}$ as follows:

$$d\mathbb{P}(\mathbf{x}_{0:t_k}) = d\mu_0(\mathbf{x}_0)\prod_{i=1}^{K}\mathbf{P}_{t_i}(\mathbf{x}_{t_{i-1}}, d\mathbf{x}_{t_i}). \tag{2}$$

In practice, we can only access the observations $\{\mathbf{y}_{t_i}\}_{i=1}^K$ recorded for each discrete time steps $\{t_i\}_{i=1}^K$. Each y_{t_i} is assumed to be generated from a measurement model $\mathbf{g}_{t_i}(\mathbf{y}_{t_i}|\mathbf{X}_{t_i})$. Our goal is to infer the *filtering* distribution, the conditional distribution of $\mathbf{X}_{0:t_k}$ given a set of observations up to time t_k, $\mathcal{H}_{t_k} = \{\mathbf{y}_{t_i}|i \le k\}$,

$$d\mathbb{P}^\star(\mathbf{x}_{0:t_k}) = \frac{1}{\mathbf{Z}(\mathcal{H}_{t_k})}\prod_{i=0}^{K} g_{t_i}(\mathbf{y}_{t_i}|\mathbf{x}_{t_i})d\mathbb{P}(\mathbf{x}_{0:t_k}), \tag{3}$$

where $\mathbf{Z}(\mathcal{H}_{t_k}) = \mathbb{E}_{\mathbb{P}}\left[\prod_{i=1}^K g_{t_i}(\mathbf{y}_{t_i}|\mathbf{X}_{t_i})\right]$ is a marginal likelihood of $\mathcal{H}_{t_k}$. Sampling trajectories from $\mathbb{P}^\star$ is generally infeasible except for highly restrictive cases, such as when the drift function b in (1) is linear and the measurement model g in (3) is Gaussian.

SOC and VI. In this study, we propose a variational inference (VI) algorithm for filtering, based on the theory of stochastic optimal control (SOC) [3]. To do so, we introduce a path measure $\mathbb{P}^\alpha$ which is induced by the solutions of the following *affine-control* SDE:

$$\text{(Controlled State)} \quad d\mathbf{X}_t = [b(t,\mathbf{X}_t) + \alpha(t,\mathbf{X}_t;\mathcal{H}_t)]\,dt + d\tilde{\mathbf{W}}_t, \quad \text{where} \quad \mathbf{X}_0 \sim \mu_0. \tag{4}$$

Here, $\alpha \in \mathbb{R}^d$ denotes a *control function*, which is chosen by a user to achieve a specific objective. In our case, we design α to approximate the posterior path measure $\mathbb{P}^\star$. We propose an objective function $\mathcal{J}(\alpha)$ which is related the SOC problem with VI for the posterior path measure in (3).

Proposition 1.1 (Variational Bound)**.** *If we choose the prior path measure as $\mathbb{P}$ and variational path measure as $\mathbb{P}^\alpha$, then the ELBO coincides with negative cost function $\mathcal{J}(\alpha)$:*

$$\underbrace{-\log p(\mathcal{H}_T)}_{\textit{Log-likelihood}} \le \mathcal{J}(\alpha|\mathcal{H}_T) := \mathbb{E}_{\mathbb{P}^\alpha}\left[\int_0^T \frac{1}{2}\left\|\alpha_s\right\|^2 ds - \sum_{i=1}^{K}\log g_i(\mathbf{y}_{t_i}|\mathbf{X}^\alpha_{t_i})\right]. \tag{5}$$

Proposition 1.1 states that the minimization problem involving $\mathcal{J}(\alpha)$ in equation (5) can be interpreted as VI for the path measure. In other words, it is possible to construct an approximate posterior, by parameterizing the control function with a suitably expressive family of functions such as DNN (*i.e.*, $\alpha := \alpha(\cdot;\theta)$). This enables a close approximation of the true posterior for the VI [22].

1.2 Efficient Modeling of the Latent System

Utilizing gradient-descent based optimization [12, 23] for (5) requires computing gradients through the simulated diffusion process over an interval $[0,T]$, which can be slow, unstable, and memory-intensive as time sequence length T increases. It contrasts with the core philosophy of many recent generative models, which focus on splitting the generative problem and solving them jointly.

Locally Linear Dynamics. Motivated by the simulation-free property of linear dynamical models [9], we explore linear drift functions that allow for the parallel computation of the approximate posterior distribution. This approach enables us to estimate a closed-form expression for the state at any time $[t,T]$ given the information $\mathcal{H}_t$ up to time $t \ge 0$. To enable the simulation-free state estimation for $\mathbf{X}_t$, we study a linear drift $b(t,\mathbf{X}_t) = \mathbf{A}\mathbf{X}_t$ and non-markov control $\alpha(t,\mathbf{X}_t,\mathcal{H}_t) \approx \alpha(\mathcal{H}_t)$. This linear formulation enable us to estimate the conditional distribution for any $t \in [0,T]$ for a given initial Gaussian distribution.

Recognizing the limitations of linear dynamics in real-world scenarios, we propose using locally linear dynamics as a more flexible approximation of the nonlinear drift function. This method leverages neural networks to fully utilize available information while maintaining a linear structure. To achieve this, we introduce a parameterization strategy inspired by [1, 11], where the transition matrix is defined as $\mathbf{A}_i = \sum_{l=1}^L w^{(l)}\mathbf{A}^{(l)}$. Here, the weights $\mathbf{w} = \{w^{(l)}\}_{l=1}^L = \mathbf{f}_\theta(\mathcal{H}_{t_i})$ are obtained

Algorithm 1 Training of PBDF

Input: $\{\mathbf{o}_{t_i}\}_{i=1}^K$, p_θ, q_ϕ, α, time-stamps $\mathcal{T}$
for $n = 1, \cdots, N$ **do**
  Estimate $q_\phi(\mathbf{y}_{t\in\mathcal{T}}|\mathbf{o}_{t\in\mathcal{T}})$
  $\mathcal{H}_T \sim q_\phi(\mathbf{y}_{t\in\mathcal{T}}|\mathbf{o}_{t\in\mathcal{T}})$
  Compute $\mathbf{b}_t = (\mathbf{A}(\mathcal{H}_t), \alpha(\mathcal{H}_t)), \forall t \in \mathcal{T}$
  $\{\mu_t^\alpha\}_{t\in\mathcal{T}} = \mathit{Scan}(\mathbf{m}_0, \mathbf{\Sigma}_0, \{\mathbf{b}_t\}_{t\in\mathcal{T}})$
  $\mathbf{X}^\alpha_{0:T} \sim \prod_{t\in\mathcal{T}} \mu_t^\alpha$
  Estimate $\mathcal{J}(\alpha|\mathcal{H}_T)$ in equation (5)
  Estimate $p_\theta(\mathbf{o}_{t\in\mathcal{T}}|\mathbf{y}_{t\in\mathcal{T}})$
  Optimize ELBO(θ, ϕ, α) in equation (9)
end for

Algorithm 2 Inference of PBDF

Input: $\{\mathbf{o}_{t_i}\}_{i=1}^K$, observed time-stamps $\mathcal{T}$, target time-stamps $\mathcal{T}'$
$\mathcal{H}_T \sim q_\phi(\mathbf{y}_{t\in\mathcal{T}}|\mathbf{o}_{t\in\mathcal{T}})$
Compute $\mathbf{b}_t = (\mathbf{A}(\mathcal{H}_t), \alpha(\mathcal{H}_t)), \forall t \in \mathcal{T}'$
$\{\mu_t^\alpha\}_{t\in\mathcal{T}'} = \mathit{Scan}(\mathbf{m}_0, \mathbf{\Sigma}_0, \{\mathbf{b}_t\}_{t\in\mathcal{T}'})$
$\mathbf{X}^\alpha_{t\in\mathcal{T}'} \sim \prod_{t\in\mathcal{T}'} \mu_t^\alpha$
$\hat{\mathbf{y}}_{t\in\mathcal{T}'} \sim g(\mathbf{y}_{t\in\mathcal{T}'}|\mathbf{X}^\alpha_{t\in\mathcal{T}'})$
$\hat{\mathbf{o}}_{t\in\mathcal{T}'} \sim p_\theta(\mathbf{o}_{t\in\mathcal{T}'}|\hat{\mathbf{y}}_{t\in\mathcal{T}'})$
Output: $\hat{\mathbf{o}}_{t\in\mathcal{T}'}$

from a neural network $\mathbf{f}_\theta$ with a softmax output and $\{\mathbf{A}^{(l)}\}_{l=1}^L$ represents a set of L parameterized diagonal matrices. Additionally, the control is parameterized by $\alpha_i = \mathbf{g}_\theta(\mathcal{H}_{t_i})$. Since $\mathcal{H}_t$ changes discretely at each observation time $\{t_i\}_{i=1}^K$, the drift function remains piecewise constant within each interval. This structure enables us to derive a closed-form solution for the intermediate latent states.

Theorem 1.2 (Simulation-free estimation)**.** *Let us consider the confrol-affine SDEs:*

$$d\mathbf{X}_t = [\mathbf{A}_i\mathbf{X}_t + \alpha_i]\, dt + \sigma d\mathbf{W}_t, \quad t \in [t_{i-1}, t_i). \tag{6}$$

Then, for an interval $[t_{i-1}, t_i)$ for all $i \in [K]$, the solution to (6) condition to initial distribution $\mu_0^\alpha = \mathcal{N}(\mathbf{m}_0, \mathbf{\Sigma}_0)$ is a Gaussian process $\mathcal{N}(\mathbf{m}_t, \mathbf{\Sigma}_t)$ with the first two moments is given by

$$\mathbf{m}_t = e^{(t-t_0)\mathbf{A}_i}\mathbf{m}_0 + \sum_{j=1}^{i-1}\left(e^{(t-t_k)\mathbf{A}_j}\mathbf{A}_j^{-1}(e^{(t_j-t_{k-1})\mathbf{A}_j} - \mathbf{I})\alpha_j\right) + \mathbf{A}_i^{-1}(e^{(t-t_{i-1})\mathbf{A}_i} - \mathbf{I})\alpha_i, \tag{7}$$

$$\mathbf{\Sigma}_t = e^{2(t-t_0)\mathbf{A}_i}\mathbf{\Sigma}_0 + \sum_{j=1}^{i-1}\left(e^{2(t-t_j)\mathbf{A}_j}\mathbf{A}_j^{-1}\frac{(e^{2(t_j-t_{j-1})\mathbf{A}_j} - \mathbf{I})}{2}\right) + \mathbf{A}_i^{-1}\frac{(e^{2(t-t_{i-1})\mathbf{A}_i} - \mathbf{I})}{2}. \tag{8}$$

Moreover, we use parallel scans to efficiently compute the the marginal distributions $\mu_{t_i}^\alpha = \mathcal{N}(\mathbf{m}_{t_i}, \mathbf{\Sigma}_{t_i})$ at each time stamp $\{t_i\}_{i=1}^K$. Given an associative operator $\otimes$ and a sequence of elements $[s_{t_1}, \cdots s_{t_K}]$, the parallel scan algorithm (*Scan*) computes the all-prefix-sum which returns the sequence $[s_{t_1}, (s_{t_1} \otimes s_{t_2}), \cdots, (s_{t_1} \otimes s_{t_2} \otimes \cdots \otimes s_{t_K})]$ in $\mathcal{O}(\log K)$ time. Since the Gaussian distribution can be chareacterized by the first two moments, applying the *Scan* to the $\mathbf{m}$ and $\mathbf{\Sigma}$ yields the desired computation. See Appendix B for more details.

Amortization. To enhance flexibility and efficiency, we treat $\{\mathbf{y}_{t_i}\}_{i=1}^K$ as an auxiliary variable in the latent space which is produced by a encoder $q_\phi(\mathbf{y}_{0:T}|\mathbf{o}_{0:T}) = \prod_{i=1}^k q_\phi(\mathbf{y}_{t_i}|\mathbf{o}_{t_i}) = \prod_{i=1}^k \mathcal{N}(\mathbf{y}_{t_i}|\mathbf{f}_\phi(\mathbf{o}_{t_i}), \sigma_\mathrm{l}\mathbf{I})$ with neural network $\mathbf{f}_\phi$ applied to the time series $\{\mathbf{o}_{t_i}\}_{i=1}^K$. This approach allows us to decouple the representation learning of $\mathbf{y}_t$ from the dynamics of $\mathbf{X}_t$, resulting in a more efficient parameterization. Additionally, it enables the modeling of nonlinear conditional distributions through a neural network decoder $p_\theta(\mathbf{o}_t \mid \mathbf{y}_t)$.

Training and Inference. We jointly train the encoder-decoder $\{\theta, \phi\}$ and the control $\{\alpha\}$ by maximizing the evidence lower bound (ELBO) of the observation log-likelihood for time series $\mathbf{o}_{0:T}$:

$$\log p_\theta(\mathbf{o}_{0:T}) \geq \mathbb{E}_{\mathcal{H}_T\sim q_\phi(\mathbf{y}_{0:T}|\mathbf{o}_{0:T})}\left[\log \frac{\prod_{i=0}^K p_\theta(\mathbf{o}_{t_i}|\mathbf{y}_{t_i})p_\theta(\mathcal{H}_T)}{\prod_{i=0}^K q_\phi(\mathbf{y}_{t_i}|\mathbf{o}_{t_i})}\right] \tag{9}$$

$$\gtrapprox \mathbb{E}_{\mathcal{H}_T\sim q_\phi(\mathbf{y}_{0:T}|\mathbf{o}_{0:T})}\sum_{i=1}^K [\log p_\theta(\mathbf{o}_{t_i}|\mathbf{y}_{t_i}) - \mathcal{J}(\alpha(\theta)|\mathcal{H}_T)] = \mathrm{ELBO}(\theta, \phi, \alpha) \tag{10}$$

The variational parameters $\{\theta, \phi\}$ can be optimized separately for each sequence, and the prior over the auxiliary variable $p_\theta(\mathcal{H}_T)$ can be computed using the ELBO proposed in (5) as part of our variational inference procedure for the latent posterior $\mathbb{P}^\star$ in proposed Sec 1.1. The overall training and inference processes are summarized in the Algorithm 1 and the Algorithm 2, respectively.

Table 1: Test MSE ($\times 10^{-2}$) for inter/extra-polation on USHCN and Physionet. The best results are highlighted in **bold**, while the second-best results are shown in blue. † indicates result from [17].

Model	Interpolation		Extrapolation		Runtime (sec./epoch)	
	USHCN	Physionet	USHCN	Physionet	USHCN	Physionet
mTAND†	1.766 ± 0.009	0.208 ± 0.025	2.360 ± 0.038	**0.340 ± 0.020**	7	10
RKN-$\Delta_t^\dagger$	0.009 ± 0.002	0.186 ± 0.030	1.491 ± 0.272	0.703 ± 0.050	94	39
GRU-$\Delta_t^\dagger$	0.090 ± 0.059	0.271 ± 0.057	2.081 ± 0.054	0.870 ± 0.077	3	5
GRU-D†	0.944 ± 0.011	0.338 ± 0.027	1.718 ± 0.015	0.873 ± 0.071	292	5736
Latent ODE†	1.798 ± 0.009	0.212 ± 0.027	2.034 ± 0.005	0.725 ± 0.072	110	791
ODE-RNN†	0.831 ± 0.008	0.236 ± 0.009	1.955 ± 0.466	0.467 ± 0.006	81	299
GRU-ODE-B†	0.841 ± 0.142	0.521 ± 0.038	5.437 ± 1.020	0.798 ± 0.071	389	90
CRU†	0.016 ± 0.006	0.182 ± 0.091	1.273 ± 0.066	0.629 ± 0.093	122 (57.8)*	114 (63.5)*
PBDF (Ours)	**0.006 ± 0.001**	**0.116 ± 0.011**	**0.941± 0.014**	0.627 ± 0.019	2.3	3.8

2 Experiment

In this section, we present empirical results that validate the effectiveness of PBDF on irregular time-series modeling. Here, we conduct experiments on two tasks: interpolation and extrapolation, using two different real-world datasets, USHCN [13] and Physionet [20]. We compare our approach against various baselines including RNN architecture (RKN-Δ_t [1], GRU-Δ_t [6], GRU-D [4]) as well as dynamics-based models (Latent ODE [5, 15], ODE-RNN [15], GRU-ODE-B [7], CRU [17]) and attention-based models (mTAND [19]), which have been developed for modeling irregular time series data. In addition to reporting the test MSE loss for each task and dataset, we also provide the actual training time per epoch for each model to validate PBDF's computational efficiency. For all experiments, we followed the same experimental setup as in [18]. Additional experimental details can be found in Appendix E.

Interpolation We begin by evaluating the effectiveness of PBDF on the interpolation task. Following the approach of [17] and [15], each model is required to infer all time points $t \in \mathcal{T}$ based on the complete set of observations $\mathbf{x}_{\mathcal{T}}$. The interpolation results presented in Table 1 clearly indicate that PBDF outperforms other baselines in terms of test MSE loss for the Physionet dataset and comparable performance for the USHCN dataset.

Extrapolation We evaluated PBDF's performance on the extrapolation task following the experimental setup of [17]. Each model infer values for all time stamps $t \in \mathcal{T}'$, where $\mathcal{T}'$ denotes the union of observed time stamps $\mathcal{T} = \{t_i\}_{i=1}^{k}$ and unseen time stamps $\mathcal{T}_u = \{t_i\}_{i=k+1}^{N}$, *i.e.*, $\mathcal{T}' = \mathcal{T} \cup \mathcal{T}_u$. For the Physionet dataset, input time stamps $\mathcal{T}$ covered the first 24 hours, while target time stamps $\mathcal{T}'$ spanned the first 48 hours. In the USHCN dataset, the timeline was split evenly, with $t_k = \frac{N}{2}$. We report the mean squared error (MSE) for unseen time stamps $\mathcal{T}_u = \mathcal{T}' - \mathcal{T}$. As shown in Table 1, PBDF consistently outperformed all baselines in terms of MSE on both datasets. Particularly for USHCN, PBDF achieved a substantial performance margin over the second-best model.

Computational Efficiency To assess the training costs relative to dynamics-based models that rely on numerical simulation, we re-ran CRU on the same hardware used for training our model (highlited by * in Table 1. Specifically, we utilized a single NVIDIA RTX A6000 GPU. As shown in Table 1, PBDF substantially reduces training costs when compared to dynamics-based models. Notably, PBDF exhibited a faster runtime than CRU while achieving superior performance. These findings demonstrate PBDF's capability to accurately approximate the posterior distribution for unseen time points, all while maintaining computational efficiency.

3 Conclusion

In this work, we introduce a novel variational inference method for approximating the filtering distribution of CD-SSM by leveraging SOC. Linear approximation of controlled drift function enable us parallel computation of the latent dynamics, thereby improving the efficiency of inference algorithm of filtering distribution. Furthermore, our approach incorporates amortization, which successfully models complex real-world time-series data such as USHCN and Physionet.

References

[1] Philipp Becker, Harit Pandya, Gregor Gebhardt, Cheng Zhao, C James Taylor, and Gerhard Neumann. Recurrent kalman networks: Factorized inference in high-dimensional deep feature spaces. In *International conference on machine learning*, pages 544–552. PMLR, 2019.

[2] Guy E Blelloch. Prefix sums and their applications. 1990.

[3] René Carmona. *Lectures on BSDEs, stochastic control, and stochastic differential games with financial applications*. SIAM, 2016.

[4] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

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

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[7] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. *Advances in neural information processing systems*, 32, 2019.

[8] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. In *NeurIPS*, 2019.

[9] Wei Deng, Weijian Luo, Yixin Tan, Marin Biloš, Yu Chen, Yuriy Nevmyvaka, and Ricky TQ Chen. Variational schr\" odinger diffusion models. *arXiv preprint arXiv:2405.04795*, 2024.

[10] Andrew H Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.

[11] Alexej Klushyn, Richard Kurle, Maximilian Soelch, Botond Cseke, and Patrick van der Smagt. Latent matters: Learning deep state-space models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 10234–10245. Curran Associates, Inc., 2021.

[12] Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *AISTATS*, 2020.

[13] MJ Menne, CN Williams Jr, RS Vose, and Data Files. Long-term daily climate records from stations across the contiguous united states, 2015.

[14] Bernt Oksendal. *Stochastic Differential Equations : An Introduction with Applications*. Springer-Verlag, Berlin, Heidelberg, 1992.

[15] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

[16] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *NeurIPS*, 2019.

[17] Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling irregular time series with continuous recurrent units. In *International conference on machine learning*, pages 19388–19405. PMLR, 2022.

[18] Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling irregular time series with continuous recurrent units. In *International Conference on Machine Learning (ICML)*, 2022.

[19] Satya Narayan Shukla and Benjamin M Marlin. Multi-time attention networks for irregularly sampled time series. *arXiv preprint arXiv:2101.10318*, 2021.

[20] Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 computing in cardiology*, pages 245–248. IEEE, 2012.

[21] Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[22] Winnie Xu, Ricky TQ Chen, Xuechen Li, and David Duvenaud. Infinitely deep bayesian neural networks with stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 721–738. PMLR, 2022.

[23] Qinsheng Zhang and Yongxin Chen. Path integral sampler: A stochastic control approach for sampling. In *International Conference on Learning Representations*, 2022.

[24] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR, 2020.

A Proof of Theorem 1.2

Proof. For any $t \in [t_i, t_{i+1})$, the solution to (6) at time t is given as

$$\mathbf{X}_t = e^{\Delta_i(t)\mathbf{A}_i}\left(\mathbf{X}_{t_i} + \int_{t_i}^{t} e^{-\Delta_i(s)\mathbf{A}_i}\alpha_i ds + \int_{t_i}^{t} e^{-\Delta_i(s)\mathbf{A}_i} d\mathbf{W}_s\right), \Delta_i(t) = \begin{cases} t - t_i, & \text{for } t > t_i \\ 0, & \text{for } t \leq t_i. \end{cases} \tag{11}$$

Given that we have defined $\mathbf{X}_0 \sim \mathcal{N}(\mathbf{m}_0, \mathbf{\Sigma}_0)$, $\mathbf{X}_{t_i}$ is a Gaussian process for any $i \in [1:K]$. The first two moments of Gaussian process can be computed from (11). First, given $\mathbf{A}_i$ is diagonal, the integral can be computed as $\int_{t_i}^{t} e^{-\Delta_i(s)\mathbf{A}_i} ds = \mathbf{A}_i^{-1}(\mathbf{I} - e^{-\Delta_i(t)\mathbf{A}_i})$ and $\mathbf{M}_i(t) := \int_{t_i}^{t} e^{-\Delta_i(s)\mathbf{A}_i} d\mathbf{W}_s$ is a martingale process with respect to $\mathbb{P}^\alpha$ *i.e.*, $\mathbb{E}_{\mathbb{P}^\alpha}[\mathbf{M}_i(t)] = 0$. Hence, since α_i is time-invariant vector, the mean $\mathbb{E}_{\mathbb{P}}[\mathbf{X}_t] = \mathbf{m}_t$ can be computed as

$$\mathbf{m}_t = e^{\Delta_i(t)\mathbf{A}_i}\mathbf{m}_{t_i} + \mathbf{A_i}^{-1}(e^{\Delta_i(t)\mathbf{A}_i} - \mathbf{I})\alpha_i. \tag{12}$$

Secondly, for a covariance $\mathbb{E}_{\mathbb{P}}[(\mathbf{X}_t - \mathbf{m}_t)(\mathbf{X}_t - \mathbf{m}_t)^T] = \mathbf{\Sigma}_t$, we can compute

$$\mathbf{\Sigma}_t = \mathbb{E}_{\mathbb{P}^\alpha}\left[e^{2\Delta_i(t)\mathbf{A}_i}\left(\mathbf{X}_{t_i} - \mathbf{m}_{t_i} + \mathbf{M}_i(t)\right)\left(\mathbf{X}_{t_i} - \mathbf{m}_{t_i} + \mathbf{M}_i(t)\right)^T\right] \tag{13}$$

$$\overset{(i)}{=} e^{2\Delta_i(t)\mathbf{A}_i}\mathbb{E}_{\mathbb{P}^\alpha}\left[(\mathbf{X}_{t_i} - \mathbf{m}_{t_i})(\mathbf{X}_{t_i} - \mathbf{m}_{t_i})^T + \|\mathbf{M}_i(t)\|_2^2\right] \tag{14}$$

$$\overset{(ii)}{=} e^{2\Delta_i(t)\mathbf{A}_i}\mathbf{\Sigma}_{t_i} + \frac{1}{2}\mathbf{A}_i^{-1}(e^{2\Delta_i(t)\mathbf{A}_i} - \mathbf{I}), \tag{15}$$

where (i) follows from the fact that $\mathbf{M}_i(t)$ is a martingale and we use Itô isometry in (ii):

$$\mathbb{E}_{\mathbb{P}^\alpha}\left[\|\mathbf{M}_i(t)\|_2^2\right] = \mathbb{E}_{\mathbb{P}^\alpha}\left[\int_{t_i}^{t}\left\|e^{-\Delta_i(s)\mathbf{A}_i}\right\|_2^2 ds\right] = \frac{1}{2}\mathbf{A}_i^{-1}(\mathbf{I} - e^{-2\Delta_i(t)\mathbf{A}_i}). \tag{16}$$

Hence, we get the Gaussian law of $\mathbf{X}_t$ at time $t \in [t_i, t_{i+1})$, which is given by

$$\mathcal{N}\left(e^{\Delta_i(t)\mathbf{A}_i}\mathbf{m}_{t_i} + \mathbf{A}^{-1}(e^{\Delta_i(t)\mathbf{A}_i} - \mathbf{I})\alpha_i, e^{2\Delta_i(t)\mathbf{A}_i}\mathbf{\Sigma}_{t_i} + \frac{1}{2}\mathbf{A}_i^{-1}(e^{2\Delta_i(t)\mathbf{A}_i} - \mathbf{I})\right). \tag{17}$$

Furthermore, given recurrence forms of mean (12) and covariance (15), the first two moments of Gaussian distribution for each time steps t_i can be computed sequentially. For simplicity, assume that $\mathbf{A} = \mathbf{A}_i = \mathbf{A}_j$ for all $i, j \in [1, \cdots, k]$, as this can be easily extended to the case where $\mathbf{A}_i \neq \mathbf{A}_j$. Then, for a mean $\mathbf{m}_{t_i}$ we have,

$$\mathbf{m}_{t_1} = e^{\Delta_0(t_1)\mathbf{A}}\mathbf{m}_{t_0} + \mathbf{A}^{-1}(e^{\Delta_0(t_1)\mathbf{A}} - \mathbf{I})\alpha^1 \tag{18}$$

$$\mathbf{m}_{t_2} = e^{\Delta_0(t_2)\mathbf{A}}\mathbf{m}_{t_0} + e^{\Delta_1(t_2)\mathbf{A}}\mathbf{A}^{-1}(e^{\Delta_0(t_1)\mathbf{A}} - \mathbf{I})\alpha^1 + \mathbf{A}^{-1}(e^{\Delta_1(t_2)\mathbf{A}} - \mathbf{I})\alpha^2 \tag{19}$$

$$\vdots \quad = \quad \vdots \tag{20}$$

$$\mathbf{m}_{t_i} = e^{\Delta_0(t_i)\mathbf{A}}\mathbf{m}_{t_0} + \sum_{k=1}^{i}\left(e^{\Delta_k(t_i)\mathbf{A}}\mathbf{A}^{-1}(e^{\Delta_{k-1}(t_k)\mathbf{A}} - \mathbf{I})\alpha^k\right) \tag{21}$$

Moreover, for a covariance $\mathbf{\Sigma}_{t_i}$, similar calculation yields

$$\mathbf{\Sigma}_{t_1} = e^{2\Delta_0(t_1)\mathbf{A}}\mathbf{\Sigma}_{t_0} + \frac{1}{2}\mathbf{A}^{-1}(e^{2\Delta_0(t_1)\mathbf{A}} - \mathbf{I}) \tag{22}$$

$$\mathbf{\Sigma}_{t_2} = e^{2\Delta_0(t_2)\mathbf{A}}\mathbf{\Sigma}_{t_0} + \frac{1}{2}e^{2\Delta_1(t_2)\mathbf{A}}\mathbf{A}^{-1}(e^{2\Delta_0(t_1)\mathbf{A}} - \mathbf{I}) + \frac{1}{2}\mathbf{A}^{-1}(e^{2\Delta_1(t_2)\mathbf{A}} - \mathbf{I}) \tag{23}$$

$$\vdots = \vdots \tag{24}$$

$$\mathbf{\Sigma}_{t_i} = e^{2\Delta_0(t_i)\mathbf{A}}\mathbf{\Sigma}_{t_0} + \frac{1}{2}\sum_{k=1}^{i}\left(e^{2\Delta_k(t_i)\mathbf{A}}\mathbf{A}^{-1}(e^{2\Delta_{k-1}(t_k)\mathbf{A}} - \mathbf{I})\right) \tag{25}$$

Therefore, given a quadruplet $(\mathbf{m}_{t_0}, \mathbf{\Sigma}_{t_0}, \mathbf{A}, \{\alpha^i\}_{i=1}^K)$, the mean and covariance for any time $t \in [0, T]$ can be computed. This concludes proof. □

B Parallel Scan

The parallelization of scan operation so called all-prefix-sums algorithms [2] have been well studied. Given an associative operator $\otimes$ and a sequence of elements $[s_{t_1}, \cdots s_{t_K}]$, the parallel scan algorithm computes the all-prefix-sum which returns the sequence $[s_{t_1}, (s_{t_1} \otimes s_{t_2}), \cdots, (s_{t_1} \otimes s_{t_2} \otimes \cdots \otimes s_{t_K})]$ in $\mathcal{O}(\log K)$ time. We already have verified that the first two moments $\{\mathbf{m}_{t_i}, \mathbf{\Sigma}_{t_i}\}_{i=1}^K$ of the controlled distributions $\{\mu_{t_i}^\alpha\}_{i=1}^K$ can be estimated by the linear recurrences in (12, 15),

$$\mathbf{m}_{t_i} = \hat{\mathbf{A}}_{i-1}\mathbf{m}_{t_{i-1}} + \hat{\mathbf{B}}_{i-1}\alpha_{i-1} \tag{26}$$

$$\mathbf{\Sigma}_{t_i} = \hat{\mathbf{A}}_{i-1}^2\mathbf{\Sigma}_{t_{-1}} + \bar{\mathbf{B}}_{i-1}\mathbf{1}_d, \tag{27}$$

where, for brevity, we define $\hat{\mathbf{A}}_{i-1} = e^{\Delta_{i-1}(t_i)\mathbf{A}_{i-1}}$, $\hat{\mathbf{B}}_{i-1} = \mathbf{A}_{i-1}^{-1}(e^{\Delta_{i-1}(t_i)\mathbf{A}_{i-1}} - \mathbf{I})$, $\bar{\mathbf{B}}_{i-1} = \frac{1}{2}\mathbf{A}_{i-1}^{-1}(e^{2\Delta_{i-1}(t_i)\mathbf{A}_{i-1}} - \mathbf{I})$ and $\mathbf{1}_d = (1, \cdots, 1) \in \mathbb{R}^d$. For a parallel scan, we will define the sequence of tuple $\{\mathbf{M}_i\}_{i=1}^k$, such that each element is $\mathbf{M}_i = (\hat{\mathbf{A}}_{i-1}, \hat{\mathbf{B}}_{i-1}\alpha_{i-1})$ and $\{\mathbf{S}_i\}_{i=1}^k$, such that each element is $\mathbf{S}_i = (\hat{\mathbf{A}}_{i-1}^2, \bar{\mathbf{B}}_{i-1}\mathbf{1}_d)$ for $\{\mathbf{m}_{t_i}\}_{i=1}^k$ and $\{\mathbf{\Sigma}_{t_i}\}_{i=1}^k$, respectively.

Now, let us define a binary operator $\otimes$:

$$\mathbf{M}_i \otimes \mathbf{M}_{i+1} = (\hat{\mathbf{A}}_i \circ \hat{\mathbf{A}}_{i-1}, \hat{\mathbf{A}}_i \circ \hat{\mathbf{B}}_{i-1}\alpha_{i-1} + \hat{\mathbf{B}}_i\alpha_i) \tag{28}$$

$$\mathbf{S}_i \otimes \mathbf{S}_{i+1} = (\hat{\mathbf{A}}_i^2 \circ \hat{\mathbf{A}}_{i-1}^2, \hat{\mathbf{A}}_i^2 \circ \bar{\mathbf{B}}_{i-1}\mathbf{1}_d + \bar{\mathbf{B}}_i\mathbf{1}_d) \tag{29}$$

We can verifying that $\otimes$ is associative operator since it satisfying:

$$(\mathbf{M}_{s+1} \otimes \mathbf{M}_{t+1}) \otimes \mathbf{M}_{u-1} = (\hat{\mathbf{A}}_t \circ \hat{\mathbf{A}}_s, \hat{\mathbf{A}}_t \circ \hat{\mathbf{B}}_s\alpha_s + \hat{\mathbf{B}}_t\alpha_t) \otimes (\hat{\mathbf{A}}_u, \hat{\mathbf{B}}_u\alpha_u) \tag{30}$$

$$= (\hat{\mathbf{A}}_u \circ \hat{\mathbf{A}}_t \circ \hat{\mathbf{A}}_s, \hat{\mathbf{A}}_u \circ \hat{\mathbf{A}}_t \circ \hat{\mathbf{B}}_s\alpha_s + \hat{\mathbf{A}}_u \circ \hat{\mathbf{B}}_t\alpha_t + \hat{\mathbf{B}}_u\alpha_u) \tag{31}$$

$$\mathbf{M}_{s+1} \otimes (\mathbf{M}_{t+1} \otimes \mathbf{M}_{u+1}) = (\hat{\mathbf{A}}_s, \hat{\mathbf{B}}_s\alpha_s) \otimes (\hat{\mathbf{A}}_u \circ \hat{\mathbf{A}}_t, \hat{\mathbf{A}}_u \circ \hat{\mathbf{B}}_t\alpha_t + \hat{\mathbf{B}}_u\alpha_u) \tag{32}$$

$$= (\hat{\mathbf{A}}_u \circ \hat{\mathbf{A}}_t \circ \hat{\mathbf{A}}_s, \hat{\mathbf{A}}_u \circ \hat{\mathbf{A}}_t \circ \hat{\mathbf{B}}_s\alpha_s + \hat{\mathbf{A}}_u \circ \hat{\mathbf{B}}_t\alpha_t + \hat{\mathbf{B}}_u\alpha_u). \tag{33}$$

Now, the operator $\otimes$ yields similar results for $\mathbf{S}_i$, both $\mathbf{M}_i$ and $\mathbf{S}_i$ can be computed using a parallel scan algorithm. In other words, we can access the marginal distributions for $\mathbb{P}^\alpha$ at each observed time stamp in a parallel way. See [21] for a comprehensive example.

C Proof of Proposition 1.1

Proof. For a latent dynamics $\mathbf{X}_{0:T}$, the ELBO is given as

$$\log p(\mathbf{y}_{0:T}) = \log \mathbb{E}_{\mathbb{P}}\left[\prod_{i=1}^K g_{t_i}(\mathbf{y}_{t_i}|\mathbf{X}_{t_i})\right] \tag{34}$$

$$\overset{(i)}{=} \log \mathbb{E}_{\mathbb{P}^\alpha}\left[\prod_{i=1}^K g_{t_i}(\mathbf{y}_{t_i}|\mathbf{X}_{t_i}^\alpha)\frac{d\mathbb{P}}{d\mathbb{P}^\alpha}(\mathbf{X}_{0:T}^\alpha)\right] \tag{35}$$

$$\overset{(ii)}{\geq} \mathbb{E}_{\mathbb{P}^\alpha}\left[\sum_{i=1}^K \log g_{t_i}(\mathbf{y}_{t_i}|\mathbf{X}_{t_i}^\alpha) + \log\frac{d\mathbb{P}}{d\mathbb{P}^\alpha}(\mathbf{X}_{0:T}^\alpha)\right] \tag{36}$$

$$\overset{(iii)}{=} \mathbb{E}_{\mathbb{P}^\alpha}\left[\sum_{i=1}^K \log g_{t_i}(\mathbf{y}_{t_i}|\mathbf{X}_{t_i}^\alpha) - \int_0^T \frac{1}{2}\left\|\alpha_s\right\|^2 ds\right], \tag{37}$$

where (i) follows from the change of measure, (ii) follows by applying the Jensen's inequality. For a (iii), we utilize the Girsanov theorem [14], $\frac{d\mathbb{P}}{d\mathbb{P}^\alpha} = \exp\left(-\int_0^T \frac{1}{2}\left\|\alpha_s\right\|^2 + \int_0^T \alpha_s d\tilde{\mathbf{W}}_s\right)$ with a $\mathbb{P}^\alpha$ adapted Wiener process $\tilde{\mathbf{W}}_s$ and by the definition of α in Sec 1.2, we get

$$\mathbb{E}_{\mathbb{P}^\alpha}\left[\log\frac{d\mathbb{P}}{d\mathbb{P}^\alpha}\right] = \mathbb{E}_{\mathbb{P}^\alpha}\left[-\int_0^T \frac{1}{2}\left\|\alpha_s\right\|^2 ds + \int_0^T \alpha_s d\tilde{\mathbf{W}}_s\right] = \mathbb{E}_{\mathbb{P}^\alpha}\left[-\int_0^T \frac{1}{2}\left\|\alpha_s\right\|^2 ds\right]. \tag{38}$$

This concludes the proof. □

D Derivation of Amortized ELBO in (9).

Let $\mathbf{o}_{0:T}$ is given time-series and $\mathbf{y}_{0:T} \sim q_\phi(\mathbf{y}_{0:T}|\mathbf{o}_{0:T})$. For a auxiliary variable $\mathbf{y}_{0:T}$, the ELBO is given as

$$\log p_\theta(\mathbf{o}_{0:T}) \geq \mathbb{E}_{q_\phi(\mathbf{y}_{0:T}|\mathbf{o}_{0:T})}\left[\log \frac{\prod_{i=1}^K p_\theta(\mathbf{o}_{t_i}|\mathbf{y}_{t_i})p(\mathbf{y}_{0:T})}{\prod_{i=1}^K q_\phi(\mathbf{y}_{t_i}|\mathbf{o}_{t_i})}\right] \tag{39}$$

$$= \mathbb{E}_{q_\phi(\mathbf{y}_{0:T}|\mathbf{o}_{0:T})}\left[\sum_{i=1}^K \log p_\theta(\mathbf{o}_{t_i}|\mathbf{y}_{t_i}) + \log p(\mathbf{y}_{0:T}) - \log \prod_{i=1}^K q_\phi(\mathbf{y}_{t_i}|\mathbf{o}_{t_i})\right] \tag{40}$$

$$\geq \mathbb{E}_{q_\phi(\mathbf{y}_{0:T}|\mathbf{o}_{0:T})}\left[\sum_{i=1}^K \log p_\theta(\mathbf{o}_{t_i}|\mathbf{y}_{t_i}) - \mathcal{J}(\alpha|\mathcal{H}_T) - \log \prod_{i=1}^K q_\phi(\mathbf{y}_{t_i}|\mathbf{o}_{t_i})\right] \tag{41}$$

$$= \mathbb{E}_{q_\phi(\mathbf{y}_{0:T}|\mathbf{o}_{0:T})}\left[\sum_{i=1}^K \log p_\theta(\mathbf{o}_{t_i}|\mathbf{y}_{t_i}) - \mathcal{J}(\alpha|\mathcal{H}_T) - \sum_{i=1}^K \log q_\phi(\mathbf{y}_{t_i}|\mathbf{o}_{t_i})\right] \tag{42}$$

$$\overset{(i)}{=} \mathbb{E}_{q_\phi(\mathbf{y}_{0:T}|\mathbf{o}_{0:T})}\left[\sum_{i=1}^K \log p_\theta(\mathbf{o}_{t_i}|\mathbf{y}_{t_i}) - \mathcal{J}(\alpha|\mathcal{H}_T)\right], \tag{43}$$

where (i) follows from $\mathbb{E}_{q_\phi(\mathbf{y}_{0:T}|\mathbf{o}_{0:T})}\left[-\sum_{i=1}^K \log q_\phi(\mathbf{y}_{t_i}|\mathbf{o}_{t_i})\right] = C$ since q_ϕ is Gaussian distribution with constant covariance matrix.

E Implementation Details

Training. For all experiments, we employed the same experimental setup as [18]. We train our model for 100 epochs using Adam optimizer with learning rate $1e-3$ and batch size of 50. Moreover, we scale by $1e-6$ for the regularization term in $\mathcal{J}(\alpha)$ in (5). We report the mean and std over 5 runs with different seeds. For a pair comparison, we keep the number of parameters similar with CRU which approximately $18K$ and $28K$, for USHCN and Physionet respectively.

Network. For $\mathbf{f}_\theta$ and $\mathbf{g}_\theta$, we adopt the transformer architecture $\mathbf{T}_\theta$ from [24], which utilizes self-attention to capture long-term dependencies in the encoded latent observations while maintaining computational efficiency. Specifically, the transformer encodes the history of observations up to the current time, *i.e.*, $\mathbf{z}_t = \mathbf{T}_\theta(\mathcal{H}_t)$, using masked attention. This history-dependent variable $\mathbf{z}_t$ is then fed into $\mathbf{f}_\theta$ and $\mathbf{g}_\theta$, *i.e.*, $\mathbf{w} = \mathbf{f}_\theta(\mathbf{z}_t), \alpha_i = \mathbf{g}_\theta(\mathbf{z}_t)$. We define $\mathbf{f}_\theta$ as a single linear layer with a softmax output and $\mathbf{g}_\theta$ as a trainable matrix $\mathbf{B}_\theta \in \mathbb{R}^{d\times d}$. The neural network encoder $\mathbf{f}_\phi$ is a 3-layer MLP, and the decoder p_θ is a 1-layer MLP. We set $L = 20$ for the matrix $\mathbf{A}$ and set the latent dimension as 20 for USHCN and 25 for Physionet. For transformer $\mathbf{T}_\theta$, we set the depth as 4 for all datasets without dropout.

E.1 Datasets

USHCN. The USHCN dataset [13] includes daily measurements from $1,218$ weather stations across the US, covering five variables: precipitation, snowfall, snow depth, and minimum and maximum temperature. We follow the pre-processing steps outlined in [8], but select a subset of $1,168$ stations over a four-year period starting from 1990, consistent with [18]. Moreover, we make the time series irregular by subsampling 50% of the time points and randomly removing 20% of the measurements.

Physionet. The Physionet dataset [20] contains 8000 multivariate clinical time-series obtained from the intensive care unit (ICU). Each time-series includes various clinical features recorded during the first 48 hours after the patient's admission to the ICU. We preprocess the data as in [16]. Although the dataset contains a total of 41 measurements, we eliminate 4 static features, *i.e.*, age, gender, height, and ICU-type, leaving 37 time-varying features. We round the time-steps to 6-minute intervals, following [18].

For each dataset, we normalize the features to lie within the range $[0, 1]$. Additionally, we scale the time stamps by a factor of 0.3 for USHCN and 0.2 for Physionet. The data is split into 60% for training, 20% for validation, and 20% for testing.