Continuous Latent Process Flows

Ruizhi Deng [*1 2] **Marcus A. Brubaker** [2 3] **Greg Mori** [1 2] **Andreas M. Lehrmann** [2]

Abstract

Partial observations of continuous time-series dynamics at arbitrary time stamps exist in many disciplines. Fitting this type of data using statistical models with continuous dynamics is not only promising at an intuitive level but also has practical benefits, including the ability to generate continuous trajectories and to perform inference on previously unseen time stamps. Despite exciting progress in this area, the existing models still face challenges in terms of their representational power and the quality of their variational approximations. We tackle these challenges with continuous latent process flows (CLPF), a principled architecture decoding continuous latent processes into continuous observable processes using a time-dependent normalizing flow driven by a stochastic differential equation. To optimize our model using maximum likelihood, we propose a novel piecewise construction of a variational posterior process and derive the corresponding variational lower bound using trajectory re-weighting. Our model shows favourable performance on synthetic data simulated from stochastic processes.

1. Introduction

Sparse and irregular observations of continuous dynamics are common in many areas of science, including finance (Zumbach & Müller, 2001; Gençay et al., 2001), healthcare (Goldberger et al., 2000), and physics (Rehfeld et al., 2011). Time-series models driven by stochastic differential equations (SDEs) provide an elegant framework for this challenging scenario and have recently gained popularity in the machine learning community (Deng et al., 2020; Hasan et al., 2020; Li et al., 2020). The SDEs are typically implemented by neural networks with trainable parameters and the latent processes defined by SDEs are decoded into an observable space with complex structure. As observations on irregular time grids can take place at arbitrary time stamps, models based on SDEs are a natural fit for this type of data. Due to the lack of closed-form transition densities for most SDEs, dedicated variational approximations have been developed to maximize the observational log-likelihoods (Archambeau et al., 2007; Hasan et al., 2020; Li et al., 2020).

In this work, we propose a model that is governed by latent dynamics defined by an expressive generic stochastic differential equation. Inspired by (Deng et al., 2020), we then use dynamic normalizing flows to decode each latent trajectory into a continuous observable process. Driven by different trajectories of the latent stochastic process continuously evolving with time, the dynamic normalizing flows can map a simple base process to a diverse class of observable processes. We illustrate this process in Fig. 1. This decoding is critical for the model to generate continuous trajectories and be trained to fit observations on irregular time grids using a variational approximation. Good variational approximation results rely on a variational posterior distribution close to the true posterior conditioned on the observations. Therefore, we also propose a method of defining and sampling from a flexible variational posterior process that is not constrained to be a Markov process based on a piecewise evaluation of SDEs. The proposed model excels at fitting observations on irregular time grids, generalizing to observations on more dense time grids, and generating trajectories continuous in time.

Contributions. In summary, we make the following contributions: (1) We propose a flow-based decoding of a generic SDE as a principled framework for continuous dynamics modeling of irregular time-series data. (2) We improve the variational approximation of the observational likelihood through a non-Markovian posterior process based on a piecewise evaluation of the underlying SDE; (3) We validate the effectiveness of our contributions in a series of studies and comparisons to state-of-the-art time-series models on synthetic datasets generated from popular stochastic processes.

2. Preliminaries

2.1. Stochastic Differential Equations

SDEs can be viewed as a stochastic analogue of ordinary differential equations in the sense that $\frac{\mathrm{d}\boldsymbol{Z}_t}{\mathrm{d}t} = \boldsymbol{\mu}(\boldsymbol{Z}_t, t) +$

[*]This work is done during internship at Borealis AI. [1]Simon Fraser University [2]Borealis AI [3]York University. Correspondence to: Ruizhi Deng <wsdmdeng@gmail.com>.

Accepted to Time Series Workshop at 38^{th} International Conference on Machine Learning.

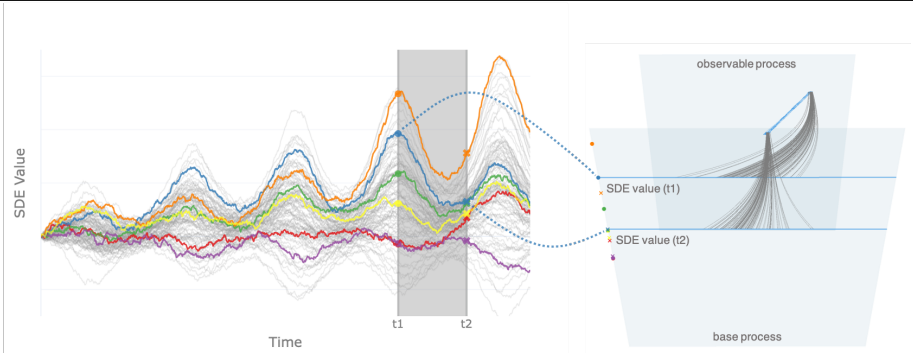

Figure 1: **Overview.** Our architecture uses a stochastic differential equation (SDE; left) to drive a time-dependent normalizing flow (NF; right). At time $t1$, $t2$ (grey bars), the values of the SDE trajectories (colored markers) serve as conditioning information for the decoding (grey lines; for clarity only shown for one latent trajectory) of a simple base process into a complex observable process. Since all stochastic processes and mappings are time-continuous, we can model observed data as partial realizations of a continuous process, enabling modelling of continuous dynamics and inference on irregular time grids.

random noise $\cdot\, \boldsymbol{\sigma}(\boldsymbol{Z}_t, t)$. Let $\boldsymbol{Z}$ be a variable which continuously evolves with time. An m-dimensional SDE describing the stochastic dynamics of $\boldsymbol{Z}$ usually takes the form

$$\mathrm{d}\boldsymbol{Z}_t = \boldsymbol{\mu}(\boldsymbol{Z}_t, t)\,\mathrm{d}t + \boldsymbol{\sigma}(\boldsymbol{Z}_t, t)\,\mathrm{d}\boldsymbol{W}_t, \tag{1}$$

where $\boldsymbol{\mu}$ maps to an m-dimensional vector, $\boldsymbol{\sigma}$ is an $m \times k$ matrix, and $\boldsymbol{W}_t$ is a k-dimensional Wiener process. The solution of an SDE is a continuous-time stochastic process $\boldsymbol{Z}_t$ that satisfies the integral equation $\boldsymbol{Z}_t = \boldsymbol{Z}_0 + \int_0^t \boldsymbol{\mu}(\boldsymbol{Z}_s, s)\,\mathrm{d}s + \int_0^t \boldsymbol{\sigma}(\boldsymbol{Z}_s, s)\,\mathrm{d}\boldsymbol{W}_s$ with initial condition $\boldsymbol{Z}_0$, where the stochastic integral should be interpreted as a traditional Itô integral (Oksendal, 2013, Chapter 3.1). For each sample trajectory $\boldsymbol{\omega} \sim \boldsymbol{W}_t$, the stochastic process $\boldsymbol{Z}_t$ maps $\boldsymbol{\omega}$ to a different trajectory $\boldsymbol{Z}_t(\boldsymbol{\omega})$.

Latent Dynamics and Variational Bound. SDEs have been used as models of latent dynamics in a variety of contexts (Li et al., 2020; Hasan et al., 2020; Archambeau et al., 2007). As closed-form finite-dimensional solutions to SDEs are rare, variational approximations are often used in practice. Li et al. propose a principled way of re-weighting the trajectories of latent SDEs for variational approximations using Girsanov's theorem (Oksendal, 2013, Chapter 8.6). Specifically, consider a prior process and a variational posterior process in the interval $[0, T]$ defined by two stochastic differential equations $\mathrm{d}\boldsymbol{Z}_t = \boldsymbol{\mu_1}(\boldsymbol{Z}_t, t)\,\mathrm{d}t + \boldsymbol{\sigma}(\boldsymbol{Z}_t, t)\,\mathrm{d}\boldsymbol{W}_t$ and $\mathrm{d}\hat{\boldsymbol{Z}}_t = \boldsymbol{\mu_2}(\hat{\boldsymbol{Z}}_t, t)\,\mathrm{d}t + \boldsymbol{\sigma}(\hat{\boldsymbol{Z}}_t, t)\,\mathrm{d}\boldsymbol{W}_t$, respectively. Furthermore, let $p(\boldsymbol{x}|\boldsymbol{Z}_t)$ denote the probability of observing $\boldsymbol{x}$ conditioned on the trajectory of the latent process $\boldsymbol{Z}_t$ in the interval $[0, T]$. If there exists a mapping $\boldsymbol{u} : \mathbb{R}^m \times [0, T] \to \mathbb{R}^k$ such that $\boldsymbol{\sigma}(\boldsymbol{z}, t)\boldsymbol{u}(\boldsymbol{z}, t) = \boldsymbol{\mu_2}(\boldsymbol{z}, t) - \boldsymbol{\mu_1}(\boldsymbol{z}, t)$ and $\boldsymbol{u}$ satisfies Novikov's condition (Oksendal, 2013, Chapter 8.6), we obtain the variational lower bound $\log p(\boldsymbol{x}) = \log \mathbb{E}[p(\boldsymbol{x}|\boldsymbol{Z}_t)] \geq \mathbb{E}[\log p(\boldsymbol{x}|\hat{\boldsymbol{Z}}_t) + \log \boldsymbol{M}_T]$, with $\boldsymbol{M}_T = \exp(-\int_0^T \frac{1}{2}\left|\boldsymbol{u}(\hat{\boldsymbol{Z}}_t, t)\right|^2 \mathrm{d}t - \int_0^T \boldsymbol{u}(\hat{\boldsymbol{Z}}_t, t)^T\,\mathrm{d}\boldsymbol{W}_t)$. See (Li et al., 2020) for a formal proof.

2.2. Continuous Time Flow Process

Normalizing flows (Rezende & Mohamed, 2015; Dinh et al., 2014; Kingma et al., 2016; Dinh et al., 2017; Papamakarios et al., 2017; Kingma & Dhariwal, 2018; Behrmann et al., 2019; Chen et al., 2019; Kobyzev et al., 2019; Papamakarios et al., 2021) are bijective mappings $f : \mathbb{R}^d \to \mathbb{R}^d$ that transform a random variable $\boldsymbol{Y}$ with a simple base distribution $p_{\boldsymbol{Y}}$ to a random variable $\boldsymbol{X}$ with a complex target distribution $p_{\boldsymbol{X}}$. We can sample from a normalizing flow by first sampling $\boldsymbol{y} \sim p_{\boldsymbol{Y}}$ and then transforming it to $\boldsymbol{x} = f(\boldsymbol{y})$. Normalizing flows can also be used for density estimation. Using the change-of-variables formula, we have $\log p_{\boldsymbol{X}}(\boldsymbol{x}) = \log p_{\boldsymbol{Y}}(g(\boldsymbol{x})) + \log \left|\det\left(\frac{\partial g}{\partial \boldsymbol{x}}\right)\right|$, where g is the inverse of f.

Recently, the continuous-time flow process (CTFP; (Deng et al., 2020)) was proposed to model irregular observations of a continuous-time stochastic process by augmenting normalizing flows with a continuous time index. Specifically, CTFP transforms a simple d-dimensional Wiener process $\boldsymbol{W}_t$ to another continuous stochastic process $\boldsymbol{X}_t$ using the transformation $\boldsymbol{X}_t = f(\boldsymbol{W}_t, t)$, where $f(\boldsymbol{w}, t)$ is an invertible mapping for each t. It has the benefits of exact log-likelihood computation of arbitrary finite-dimensional distributions and generating continuous trajectories.

A latent variant of CTFP is further augmented with a static latent variable to introduce non-Markovian behavior. It models continuous stochastic processes as $\boldsymbol{X}_t = f(\boldsymbol{W}_t, t; \boldsymbol{Z})$, where $\boldsymbol{Z}$ is a latent variable with standard Gaussian distribution and $f(\cdot, \cdot; z)$ is a CTFP model that decodes each sample z of $\boldsymbol{Z}$ into a stochastic processes with continuous trajectories. Latent CTFP can be used to estimate finite-dimensional distributions with a variational approximation.

3. Model

Let $\{(\boldsymbol{x}_{t_i}, t_i)\}_{i=1}^n$ denote a sequence of d-dimensional observations sampled at arbitrary points in time, where $\boldsymbol{x}_{t_i}$ and t_i denote the value and time stamp of the observation, respectively. The observations are assumed to be partial realizations of a continuous-time stochastic process $\boldsymbol{X}_t$. Our

training objective is the maximization of the observational log-likelihood induced by $\boldsymbol{X}_t$ on a given time grid,

$$\mathcal{L} = \log p_{\boldsymbol{X}_{t_1},\dots,\boldsymbol{X}_{t_n}}(\boldsymbol{x}_{t_1},\dots,\boldsymbol{x}_{t_n}), \tag{2}$$

for an inhomogeneous collection of sequences with varying lengths and time stamps. We model this challenging scenario with Continuous Latent Process Flows (CLPF). In Section 3.1, we present our model in more detail. Training and inference methods will be discussed in Section 3.2.

3.1. Continuous Latent Process Flows

A Continuous Latent Process Flow consists of two major components: an SDE describing the continuous latent dynamics of an observable stochastic process and a continuously indexed normalizing flow serving as a time-dependent decoder. The following paragraphs discuss the relationship between these components in more detail.

Continuous Latent Dynamics. Analogous to our overview in Section 2.1, we model the evolution of an m-dimensional time-continuous latent state $\boldsymbol{Z}_t$ in the time interval $[0, T]$ using a flexible stochastic differential equation driven by an m-dimensional Wiener Process $\boldsymbol{W}_t$,

$$\mathrm{d}\boldsymbol{Z}_t = \boldsymbol{\mu}_\gamma(\boldsymbol{Z}_t, t)\,\mathrm{d}t + \boldsymbol{\sigma}_\gamma(\boldsymbol{Z}_t, t)\,\mathrm{d}\boldsymbol{W}_t, \tag{3}$$

where γ denotes the (shared) learnable parameters of the drift function $\boldsymbol{\mu}$ and variance function $\boldsymbol{\sigma}$. In our experiments, we implement $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ using deep neural networks. Importantly, the latent state $\boldsymbol{Z}_t$ exists for each $t \in [0, T]$ and can be sampled on any given time grid, which can be irregular and different for each sequence.

Time-Dependent Decoding. Latent variable models decode a latent state into an observable variable with complex distribution. As an observed sequence $\{(\boldsymbol{x}_{t_i}, t_i)\}_{i=1}^n$ is assumed to be a partial realization of a continuous-time stochastic process, continuous trajectories of the latent process $\boldsymbol{Z}_t$ should be decoded into continuous trajectories of the observable process $\boldsymbol{X}_t$, and not discrete distributions. Following recent advances in dynamic normalizing flows (Deng et al., 2020; Cornish et al., 2020; Caterini et al., 2020), we model $\boldsymbol{X}_t$ as

$$\boldsymbol{X}_t = F_\theta(\boldsymbol{O}_t; \boldsymbol{Z}_t, t), \tag{4}$$

where $\boldsymbol{O}_t$ is a d-dimensional Ornstein–Uhlenbeck (OU) process with closed-form transition density[1] and $F_\theta(\cdot\,; \boldsymbol{z}_t, t)$ is a normalizing flow parameterized by θ for any $\boldsymbol{z}_t, t$. The transformation F_θ decodes each sample path of $\boldsymbol{Z}_t$ into a complex distribution over continuous trajectories $\boldsymbol{X}_t$ if F_θ is a continuous mapping and the sampled trajectories of $\boldsymbol{O}_t$ are continuous with respect to time t. The OU process has a stationary marginal distribution and bounded variance. As a result, the variance of the observation process does not increase due to the increase of variance in the base process.

Flow Architecture. The continuously indexed normalizing flow $F_\theta(\,\cdot\,; \boldsymbol{z}_t, t)$ can be implemented in multiple ways. We use an ANODE (Dupont et al., 2019), similar to Deng et al.. The mapping F_θ is defined as the solution to the initial value problem

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\begin{pmatrix}\boldsymbol{h}(\tau)\\ \boldsymbol{a}(\tau)\end{pmatrix} = \begin{pmatrix}f_\theta(\boldsymbol{h}(\tau), \boldsymbol{a}(\tau), \tau)\\ g_\theta(\boldsymbol{a}(\tau), \tau)\end{pmatrix}, \quad \begin{pmatrix}\boldsymbol{h}(\tau_0)\\ \boldsymbol{a}(\tau_0)\end{pmatrix} = \begin{pmatrix}\boldsymbol{o}_t\\ (\boldsymbol{z}_t, t)^T\end{pmatrix}, \tag{5}$$

where $\tau \in [\tau_0, \tau_1]$, $\boldsymbol{h}(\tau) \in \mathbb{R}^d$, $\boldsymbol{a}(\tau) \in \mathbb{R}^{m+1}$, $f_\theta : \mathbb{R}^d \times \mathbb{R}^{m+1} \times [\tau_0, \tau_1] \to \mathbb{R}^d$, $g_\theta : \mathbb{R}^{m+1} \times [t_0, t_1] \to \mathbb{R}$, and F_θ is defined as the solution of $\boldsymbol{h}(\tau)$ at $\tau = \tau_1$. Note the difference between t and τ: while $t \in [0, T]$ describes the continuous process dynamics, $\tau \in [\tau_0, \tau_1]$ describes the continuous time-dependent decoding at each time step t.

3.2. Training and Inference

With the model fully specified, we can now focus our attention on training and inference. Computing the observational log-likelihood (Eq.(2)) induced by a time-dependent decoding of an SDE (Eq.(4)) on an arbitrary time grid is challenging, because only few SDEs have closed-form transition densities. Consequently, variational approximations are needed for flexible SDEs such as Eq.(3). We propose a principled way of approximating the observational log-likelihood with a variational lower bound based on a novel piecewise construction of the posterior latent process.

Observational Log-Likelihood. The observational log-likelihood can be written as an expectation over latent trajectories of the conditional likelihood, which can be evaluated in closed form,

$$\begin{aligned}\mathcal{L} &= \log \mathbb{E}_{\boldsymbol{W}_t}\left[p_{\boldsymbol{X}_{t_1},\dots,\boldsymbol{X}_{t_n}|\boldsymbol{Z}_t}(\boldsymbol{x}_{t_1},\dots,\boldsymbol{x}_{t_n}|\boldsymbol{Z}_t(\omega))\right]\\ &= \log \mathbb{E}_{\boldsymbol{W}_t}\left[\prod_{i=1}^n p_{\boldsymbol{X}_{t_i}|\boldsymbol{X}_{t_{i-1}},\boldsymbol{Z}_{t_i},\boldsymbol{Z}_{t_{i-1}}}(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \boldsymbol{Z}_{t_i}(\omega), \boldsymbol{Z}_{t_{i-1}}(\omega))\right],\end{aligned} \tag{6}$$

where $\boldsymbol{Z}_t(\omega)$ denotes a sample trajectory of $\boldsymbol{Z}_t$ driven by $\omega \sim \boldsymbol{W}_t$. For simplicity, we assume w.l.o.g. and in this section only $\boldsymbol{Z}_0, \boldsymbol{X}_0$ to be given. As a result of invertibility, the conditional likelihood terms $p_{\boldsymbol{X}_{t_i}|\boldsymbol{X}_{t_{i-1}},\boldsymbol{Z}_{t_i},\boldsymbol{Z}_{t_{i-1}}}$ can be computed using the change-of-variables formula,

$$\begin{aligned}&\log p_{\boldsymbol{X}_{t_i}|\boldsymbol{X}_{t_{i-1}},\boldsymbol{Z}_{t_i},\boldsymbol{Z}_{t_{i-1}}}(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \boldsymbol{Z}_{t_i}(\omega), \boldsymbol{Z}_{t_{i-1}}(\omega))\\ &= \log p_{\boldsymbol{O}_{t_i}|\boldsymbol{O}_{t_{i-1}}}(\boldsymbol{o}_{t_i}|\boldsymbol{o}_{t_{i-1}}) - \log\left|\det \frac{\partial F_\theta(\boldsymbol{o}_{t_i}; \boldsymbol{Z}_{t_i}(\omega), t_i)}{\partial \boldsymbol{o}_{t_i}}\right|,\end{aligned} \tag{7}$$

where $\boldsymbol{o}_{t_i} = F_\theta^{-1}(\boldsymbol{x}_{t_i}; \boldsymbol{Z}_{t_i}(\omega), t_i)$.

Piecewise Construction of Variational Posterior. We use a variational approximation of the observational log-likelihood for both training and inference. Good variational approximations rely on variational posteriors that are close enough to the true posterior of the latent trajectory conditioned on observations. We develop a

[1] $p_{\mathbf{O}_{t_i}|\mathbf{O}_{t_j}}(\mathbf{o}_{t_i}|\mathbf{o}_{t_j})$ exists in closed form for any $t_j < t_i$.

method that naturally adapts to different time grids and is not constrained by the Markov property of SDE solutions.

Our construction makes use of a further decomposition of the observational log-likelihood based on the following observations: $\{\boldsymbol{W}_{s+t} - \boldsymbol{W}_s\}_{t\geqslant 0}$ is also a Wiener process $\forall s \geqslant 0$ and the solution to Eq. 3 is a Markov process. Specifically, let $\{(\boldsymbol{\Omega}^{(i)}, \boldsymbol{\mathcal{F}}^{(i)}_{t_i - t_{i-1}}, P^{(i)})\}_{i=1}^n$ be a series of probability spaces on which n independent m-dimensional Wiener processes $\boldsymbol{W}_t^{(i)}$ are defined. We can sample a complete trajectory of the Wiener process $\boldsymbol{W}_t$ in the interval $[0, T]$ by sampling independent trajectories $\omega^{(i)}$ of length $t_i - t_{i-1}$ from $\boldsymbol{\Omega}^{(i)}$ and adding them, i.e., $\omega_t = \sum_{\{i:t_i<t\}} \omega^{(i)}_{t_i - t_{i-1}} + \omega^{(i^*)}_{t-t_{i^*-1}}$, where $i^* = \max\{i : t_i < t\} + 1$. As a result, we can solve the latent stochastic differential equations in a piecewise manner. $\boldsymbol{Z}_{t_i}$ is obtained by solving the following stochastic differential equation

$$\mathrm{d}\widehat{\boldsymbol{Z}}_t = \boldsymbol{\mu}_\gamma(\widehat{\boldsymbol{Z}}_t, t+t_{i-1})\,\mathrm{d}t + \boldsymbol{\sigma}_\gamma(\widehat{\boldsymbol{Z}}_t, t+t_{i-1})\,\mathrm{d}\boldsymbol{W}_t^{(i)}, \quad (8)$$

with $\boldsymbol{Z}_{t_{i-1}}$ being the initial value. The log-likelihood of the observations can thus be rewritten as

$$\begin{aligned}\mathcal{L} &= \log \mathbb{E}_{\boldsymbol{W}_t^{(1)}\times\cdots\times\boldsymbol{W}_t^{(n)}}\left[\prod_{i=1}^n p(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \boldsymbol{z}_{t_i}, \boldsymbol{z}_{t_{i-1}})\right] \\ &= \log \mathbb{E}_{\boldsymbol{W}_t^{(1)}}\Big[p(\boldsymbol{x}_{t_1}|\boldsymbol{x}_{t_0}, \boldsymbol{z}_{t_1}, \boldsymbol{z}_{t_0})\dots \\ &\qquad \mathbb{E}_{\boldsymbol{W}_t^{(i)}}\Big[p(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \boldsymbol{z}_{t_i}, \boldsymbol{z}_{t_{i-1}})\mathbb{E}_{\boldsymbol{W}_t^{(i+1)}}\dots\Big]\Big].\end{aligned} \quad (9)$$

In preparation of our variational approximation, we can now introduce one posterior SDE

$$\mathrm{d}\tilde{\boldsymbol{Z}}_t = \boldsymbol{\mu}_{\phi_i}(\tilde{\boldsymbol{Z}}_t, t+t_{i-1})\,\mathrm{d}t + \boldsymbol{\sigma}_\gamma(\tilde{\boldsymbol{Z}}_t, t+t_{i-1})\,\mathrm{d}\boldsymbol{W}_t^{(i)} \quad (10)$$

for each expectation $\mathbb{E}_{\boldsymbol{W}_t^{(i)}}\left[p(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \boldsymbol{z}_{t_i}, \boldsymbol{z}_{t_{i-1}})\dots\right]$ in Eq.(9).

Variational Lower Bound with Piecewise Reweighting. The posterior SDEs in Eq.(10) form the basis for a variational approximation of the expectations in Eq.(9). Specifically, sampling $\tilde{\boldsymbol{z}}$ from the posterior SDE, the expectation can be rewritten as

$$\mathbb{E}_{\boldsymbol{W}_t^{(i)}}\left[p(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \tilde{\boldsymbol{z}}_{t_i}, \boldsymbol{z}_{t_{i-1}}, \omega^{(i)})\boldsymbol{M}^{(i)}(\omega^{(i)})\mathbb{E}_{\boldsymbol{W}_t^{(i+1)}}\dots\right], \quad (11)$$

where $\boldsymbol{M}^{(i)} = \exp(-\int_0^{t_i-t_{i-1}} \frac{1}{2}|\boldsymbol{u}(\tilde{\boldsymbol{Z}}_s, s)|^2\,\mathrm{d}s - \int_0^{t_i-t_{i-1}} \boldsymbol{u}(\tilde{\boldsymbol{Z}}_s, s)^T\,\mathrm{d}\boldsymbol{W}_s^{(i)})$ serves as a re-weighting term for the sampled trajectory between the prior latent SDE (Eq.(8)) and posterior latent SDE (Eq.(10)), with $\boldsymbol{u}$ satisfying $\boldsymbol{\sigma}_\gamma(\boldsymbol{z}, s+t_{i-1})\boldsymbol{u}(\boldsymbol{z}, s) = \boldsymbol{\mu}_{\phi_i}(\boldsymbol{z}, s+t_{i-1}) - \boldsymbol{\mu}_\gamma(\boldsymbol{z}, s+t_{i-1})$. By defining and sampling a latent state from the posterior latent SDEs for each time interval, we obtain the following evidence lower bound (ELBO) of the log-likelihood:

Table 1: **Quantitative Evaluation (Synthetic Data)**. We show test negative log-likelihoods (NLLs) of four synthetic stochastic processes across different models. [GBM: geometric Brownian motion (ground truth NLLs: 0.388); LSDE: linear stochastic differential equation; CAR: continuous autoregressive process]

Model	GBM	LSDE	CAR
Latent ODE	2.139	0.900	5.030
CTFP	3.023	-0.474	372.557
Latent CTFP	1.502	-0.460	415.480
Latent SDE	1.233	-0.001	4.9342
CLPF (**ours**)	**0.435**	**-0.826**	**1.325**

$$\begin{aligned}\mathcal{L} &= \log \mathbb{E}_{\boldsymbol{W}_t^{(1)}}\Big[p(\boldsymbol{x}_{t_1}|\boldsymbol{x}_{t_0}, \tilde{\boldsymbol{z}}_{t_1}, \tilde{\boldsymbol{z}}_{t_0})\boldsymbol{M}^{(1)}(\omega^{(1)})\dots \\ &\qquad \mathbb{E}_{\boldsymbol{W}_t^{(i)}}\Big[p(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \tilde{\boldsymbol{z}}_{t_i}, \tilde{\boldsymbol{z}}_{t_{i-1}})\boldsymbol{M}^{(i)(\omega^{(i)})}\dots\Big]\dots\Big] \\ &= \log \mathbb{E}_{\boldsymbol{W}_t^{(1)}\times\cdots\times\boldsymbol{W}_t^{(n)}}\left[\prod_{i=1}^n p(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \tilde{\boldsymbol{z}}_{t_i}, \tilde{\boldsymbol{z}}_{t_{i-1}})\boldsymbol{M}^{(i)}(\omega^{(i)})\right] \\ &\geqslant \mathbb{E}_{\boldsymbol{W}_t^{(1)}\times\cdots\times\boldsymbol{W}_t^{(n)}}\left[\sum_{i=1}^n \log p(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \tilde{\boldsymbol{z}}_{t_i}, \tilde{\boldsymbol{z}}_{t_{i-1}}) + \right. \\ &\qquad \left. \sum_{i=1}^n \log \boldsymbol{M}^{(i)}(\omega^{(i)})\right].\end{aligned} \quad (12)$$

The bound above can be further extended into a tighter bound in IWAE (Burda et al., 2016) form by drawing multiple independent samples from each $\boldsymbol{W}_t^{(i)}$. The variational parameter ϕ_i is the output of an encoder RNN that takes the sequence of observations up to t_i, $\{\boldsymbol{X}_{t_1}, \dots, \boldsymbol{X}_{t_i}\}$, and the sequence of previously sampled latent states, $\{\boldsymbol{Z}_{t_1}, \dots, \boldsymbol{Z}_{t_{i-1}}\}$, as inputs.

4. Experiments

We compare our proposed architecture against several baseline models with continuous dynamics including CTFP, latent CTFP, latent SDE, and latent ODE to fit irregular time series simulated from common continuous stochastic processes including geometric Brownian motion (GBM), linear stochastic differential equations (LSDE), and continuous autoregressive process (CAR). We report negative log likelihood and the results are displayed in Table 1. We defer more details to the supplementary material

5. Conclusion

We have presented Continuous Latent Process Flows (CLPF), a generative model of continuous dynamics that enables inference on arbitrary real time grids, a complex operation for which we have also introduced a powerful piecewise variational approximation. Our architecture is built around the representation power of a flexible stochastic differential equation driving a continuously indexed normalizing flow. A set of qualitative results on synthetic datasets demonstrates the effectiveness of our model.

References

Archambeau, C., Cornford, D., Opper, M., and Shawe-Taylor, J. Gaussian process approximations of stochastic differential equations. In *Gaussian Processes in Practice*, pp. 1–16. PMLR, 2007.

Bayram, M., Partal, T., and Buyukoz, G. O. Numerical methods for simulation of stochastic differential equations. *Advances in Difference Equations*, 2018(1):1–10, 2018.

Behrmann, J., Grathwohl, W., Chen, R. T., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks. In *ICML*, pp. 573–582. PMLR, 2019.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. In *ICLR*, 2016.

Caterini, A., Cornish, R., Sejdinovic, D., and Doucet, A. Variational inference with continuously-indexed normalizing flows. *arXiv preprint arXiv:2007.05426*, 2020.

Chen, T. Q., Behrmann, J., Duvenaud, D. K., and Jacobsen, J.-H. Residual flows for invertible generative modeling. In *Advances in NeurIPS*, pp. 9913–9923, 2019.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pp. 2980–2988, 2015.

Cornish, R., Caterini, A., Deligiannidis, G., and Doucet, A. Relaxing bijectivity constraints with continuously indexed normalising flows. In *ICML*, pp. 2133–2143. PMLR, 2020.

Deng, R., Chang, B., Brubaker, M. A., Mori, G., and Lehrmann, A. Modeling continuous stochastic processes with dynamic normalizing flows. In *Advances in NeurIPS*, 2020.

Dinh, L., Krueger, D., and Bengio, Y. NICE: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using Real NVP. In *ICLR*, 2017.

Dupont, E., Doucet, A., and Teh, Y. W. Augmented Neural ODEs. In *NeurIPS*, 2019.

Gençay, R., Dacorogna, M., Muller, U. A., Pictet, O., and Olsen, R. *An introduction to high-frequency finance*. Elsevier, 2001.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23): e215–e220, 2000.

Hasan, A., Pereira, J. M., Farsiu, S., and Tarokh, V. Identifying latent stochastic differential equations with variational auto-encoders. *arXiv preprint arXiv:2007.06075*, 2020.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in NeurIPS*, pp. 10215–10224, 2018.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. In *Advances in NeurIPS*, pp. 4743–4751, 2016.

Kobyzev, I., Prince, S., and Brubaker, M. A. Normalizing flows: Introduction and ideas. *arXiv preprint arXiv:1908.09257*, 2019.

Li, X., Wong, T.-K. L., Chen, R. T., and Duvenaud, D. Scalable gradients for stochastic differential equations. In *AISTATS*, 2020.

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

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In *Advances in NeurIPS*, pp. 2338–2347, 2017.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Rehfeld, K., Marwan, N., Heitzig, J., and Kurths, J. Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics*, 18(3):389–404, 2011.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538, 2015.

Rubanova, Y., Chen, T. Q., and Duvenaud, D. K. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in NeurIPS*, pp. 5321–5331, 2019.

Zumbach, G. and Müller, U. Operators on inhomogeneous time series. *International Journal of Theoretical and Applied Finance*, 4(01):147–177, 2001.

A. Synthetic Dataset Specifications

We compare our model against the baseline models using data simulated from three continuous stochastic processes: geometric Brownian motion (GBM), linear SDE (LSDE), and continuous auto-regressive process (CAR). We simulate the observations of GBM, LSDE, and CAR in the time interval $[0, 30]$. For each trajectory, we sample the observation time stamps from an independent homogeneous Poisson process with intensity 2 (i.e., the average interarrival time of observations is 0.5). The observation values for geometric Brownian motion are sampled according to the exact transition density. The observation values of the LSDE and CAR processes are simulated using the Euler-Maruyama method (Bayram et al., 2018) with a step size of $1e-5$. For each process, we simulate 10000 sequences, of which 7000 are used for training, 1000 are used for validation and 2000 are used for evaluation.

In the remainder of this section we provide details about the parameters of the stochastic processes:

Geometric Brownian Motion. The stochastic process can be represented by the stochastic differential equation $\mathrm{d}\boldsymbol{X}_t = 0.2\boldsymbol{X}_t\,\mathrm{d}t + 0.1\boldsymbol{X}_t\,\mathrm{d}\boldsymbol{W}_t$, with an initial value $\boldsymbol{X}_0 = 1$.

Linear SDE. The linear SDE we simulated has the form $\mathrm{d}\boldsymbol{X}_t = (0.5\sin(t)\boldsymbol{X}_t + 0.5\cos(t))\,\mathrm{d}t + \frac{0.2}{1+\exp(-t)}\,\mathrm{d}\boldsymbol{W}_t$. The initial value was set to 0.

Continuous AR(4) Process. A CAR process $\boldsymbol{X}_t$ can be obtained by projecting a high-dimensional process to a low dimension. This process tests our model's ability to capture non-Markov processes:

$$\begin{aligned}&\boldsymbol{X}_t = [1,0,0,0]\boldsymbol{Y}_t,\\ &\mathrm{d}\boldsymbol{Y}_t = A\boldsymbol{Y}_t\,\mathrm{d}t + e\,\mathrm{d}\boldsymbol{W}_t,\end{aligned}\quad \text{where } A = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ a_1 & a_2 & a_3 & a_4\end{bmatrix},$$
$$e = [0,0,0,1],\ [a_1,a_2,a_3,a_4] = [+0.002, +0.005, -0.003, -0.002] \tag{13}$$

B. Model Architectures

We keep the key hyperparameters of our model and the baseline models in a similar range, including the dimensions of the recurrent neural networks' hidden states, the dimensions of latent states, and the hidden dimensions of decoders. We set the latent dimension to 2 for geometric Brownian motion and linear SDE, and 4 for the continuous auto-regressive process. For all models that use a recurrent neural network, we use gated recurrent units (GRU) with a hidden state of size 16.

Continuous Latent Process Flows (CLPF) We use two fully-connected network with two hidden layers to implement the drift $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}$ networks, for both the prior and posterior SDE. The hidden layer dimensions for (prior SDE, posterior SDE) are $(32, 32)$. We use a GRU as the encoder of observations $\boldsymbol{X}_{t_i}$ and latent states $\boldsymbol{Z}_{t_i}$ to produce ϕ_i at each step i in Equation 10 in the main paper. The GRU takes the observation $\boldsymbol{X}_{t_i}$, the latent state $\boldsymbol{Z}_{t_i}$, the current and previous time stamps t_i and t_{i-1}, and the difference between the two time stamps as inputs. The updated hidden state is projected to a context vector of size 16. The projected vector is concatenated with $\boldsymbol{X}_{t_i}$ and t_i as part of the input to the drift function $\boldsymbol{\mu}_{\phi_i}$ of the posterior process in the interval $[t_{i-1}, t_i]$.

We use five blocks of the generative variant of augmented neural ODE (ANODE) (Deng et al., 2020) to implement the indexed normalizing flows in all experiments. In each ANODE block, the function $\boldsymbol{h}$ in Equation 5 in the main paper is implemented as a neural network with 4 hidden layers of dimension $[8, 32, 32, 8]$; the funtion g is implemented as a zero mapping.

Continuous Time Flow Process (CTFP) and Latent CTFP For CTFP (Deng et al., 2020) and the decoder of its latent variant, we also use 5 ANODE blocks with the same number of hidden dimensions as CLPF. The encoder of the latent CTFP model is an ODE-RNN (Rubanova et al., 2019). The ODE-RNN model consists of a recurrent neural network and a neural ODE module implemented by a network with one hidden layer of dimension 100. The default values in the official implementation[2] of latent CTFP are adopted for other hyperparameters of the model architecture.

Latent ODE For the latent ODE model (Rubanova et al., 2019), we use the same encoder as latent CTFP. The latent ODE decoder uses a neural ODE with one hidden layer of dimension 100 to propagate the latent state across a time interval deterministically. The latent state propagated to each observation time stamp is mapped to the mean and variance of a Gaussian observational distribution by a fully-connected network with 4 hidden layers of dimension $[16, 64, 64, 16]$. We use the default values in the official implementation[3] of latent ODE for other hyperparameters of the model architecture.

Variational RNN (VRNN) The backbone of VRNN (Chung et al., 2015) is a recurrent neural network. A one-layer GRU is used to implement the recurrent neural network. During inference, the hidden state is projected to the mean and variance of a Gaussian distribution of the latent state by a fully-connected network. During generation, the hidden state is directly mapped to the parameters of the latent distribution. The sample of the latent state is decoded to the parameters of an observational Gaussian distribution by a fully-connected

[2] https://github.com/BorealisAI/continuous-time-flow-process
[3] https://github.com/YuliaRubanova/latent_ode

network with 4 hidden layers of dimension $[16, 64, 64, 16]$. In the recurrence operation, the GRU takes the latest latent sample and observation as inputs. We also concatenate the time stamp of the current observation as well as the difference between the time stamps of the current and previous observation to the input.

Experiment Settings For each process, we use 7000 sequences for training, 1000 sequences for validation and 2000 sequences for test. We train all the models with flat learning rate of 1e-3 until convergence. We cap the number of training epochs to 100 for GBM and LSDE and 200 for CAR. For models optimized with IWAE bound, we use 3 samples of latent state (trajectory) for training, 25 samples for validation and 125 samples for evaluation. To solve the latent stochastic process in the continuous latent process flow model, we used Euler-Maruyama scheme with adaptive step size. The automatic differentiation engine of PyTorch is used for backpropagation.