Continuous Latent Process Flows

Anonymous Author(s)
Affiliation
Address
email

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 ablation studies demonstrate the effectiveness of our contributions in various inference tasks on irregular time grids. Comparisons to state-of-the-art baselines show our model's favourable performance on both synthetic and real-world time-series data.

1 Introduction

Sparse and irregular observations of continuous dynamics are common in many areas of science, including finance [34, 14], healthcare [15], and physics [30]. 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 [10, 17, 24]. 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 [1, 17, 24].

Despite the progress of existing works, challenges still remain for SDE-based models to be applied to irregular time series data. One major challenge is the model's representation power. The continuous-time flow process (CTFP; [10]) uses a series of invertible mappings continuously indexed by time to transform a simple Wiener process to a more complex stochastic process. The use of a simple latent process and invertible transformations permits CTFP models to evaluate the exact likelihood of observations on any time grid efficiently, but they also limit the set of stochastic processes that CTFP can express to some specific form which can be obtained using Ito's Lemma and excludes many commonly seen stochastic processes. Another constraint of representation power in practice is the Lipschitz property of transformations in the models. The latent SDE model proposed by Hasan et al. and CTFP both transform a latent stochastic process with constant variance to an observable one using injective mappings. Due to the Lipschitz property existing in many invertible neural network architectures, some processes that can be written as a non-Lipschitz transformation of a

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

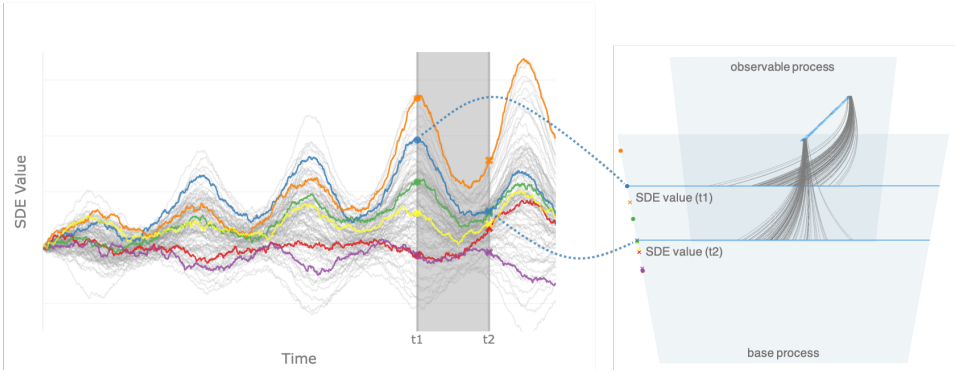


Figure 1: **Overview.** Our architecture uses a stochastic differential equation (SDE; left) to drive a time-dependent normalizing flow (NF; right). At time $t1$, $t2$ (grey bars), the values of the SDE trajectories (colored markers) serve as conditioning information for the decoding (grey lines; for 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.

simple process, like geometric Brownian motion, cannot be expressed by these models unless specific choices of non-Lipschitz decoders are used. Apart from the model's representation power, variational inference is another challenge in training SDE-based models. The latent SDE model in the work of Li et al. [24] uses a principled method of variational approximation based on the re-weighting of trajectories between a variational posterior and a prior process. The variational posterior process is constructed using a single SDE conditioned on the observations. As a result, it is restricted to be a Markov process, which could limit its capability to approximate the true posterior well enough.

In this work, we propose a model that is governed by latent dynamics defined by an expressive generic stochastic differential equation. Inspired by [10], 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 ablation studies and comparisons to state-of-the-art time-series models, both on synthetic and real-world datasets.

2 Preliminaries

2.1 Stochastic Differential Equations

SDEs can be viewed as a stochastic analogue of ordinary differential equations in the sense that $\frac{\mathrm{d}\boldsymbol{Z}_t}{\mathrm{d}t} = \boldsymbol{\mu}(\boldsymbol{Z}_t, t) + \text{random noise} \cdot \boldsymbol{\sigma}(\boldsymbol{Z}_t, t)$. 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 [27, 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 [24, 17, 1]. 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 [27, 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 [27, Chapter 8.6], we obtain the variational lower bound

$$\log p(\boldsymbol{x}) = \log \mathbb{E}[p(\boldsymbol{x}|\boldsymbol{Z}_t)] = \log \mathbb{E}[p(\boldsymbol{x}|\hat{\boldsymbol{Z}}_t)\boldsymbol{M}_T] \geq \mathbb{E}[\log p(\boldsymbol{x}|\hat{\boldsymbol{Z}}_t) + \log \boldsymbol{M}_T], \tag{2}$$

where $\boldsymbol{M}_T = \exp(-\int_0^T \frac{1}{2} \left|\boldsymbol{u}(\hat{\boldsymbol{Z}}_t, t)\right|^2\ \mathrm{d}t - \int_0^T \boldsymbol{u}(\hat{\boldsymbol{Z}}_t, t)^T\ \mathrm{d}\boldsymbol{W}_t)$. See [24] for a formal proof.

2.2 Normalizing Flows

Normalizing flows [31, 11, 22, 12, 28, 21, 2, 7, 23, 29] employ a bijective mapping $f : \mathbb{R}^d \to \mathbb{R}^d$ to 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 $\boldsymbol{x} = f(\boldsymbol{y})$. As a result of invertibility, 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.

Continuous Indexing. More recently, normalizing flows have been augmented with a continuous index [5, 10, 9]. For instance, the continuous-time flow process (CTFP; [10]) models irregular observations of a continuous-time stochastic process. 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. Despite its benefits of exact log-likelihood computation of arbitrary finite-dimensional distributions, the expressive power of CTFP to model stochastic processes is limited in the follow two aspects: (1) An application of Itô's lemma [27, Chapter 4.2] shows that CTFP can only represent stochastic process of the form

$$\mathrm{d}f(\boldsymbol{W}_t, t) = \{\frac{\mathrm{d}f}{\mathrm{d}t}(\boldsymbol{W}_t, t) + \frac{1}{2}\operatorname{Tr}(\mathbf{H}_{\boldsymbol{w}} f(\boldsymbol{W}_t, t))\}\ \mathrm{d}t + (\nabla_{\boldsymbol{w}} f^T(\boldsymbol{W}_t, t))^T\ \mathrm{d}\boldsymbol{W}_t, \tag{3}$$

where $\mathbf{H}_{\boldsymbol{w}} f$ is the Hessian matrix of f with respect to $\boldsymbol{w}$ and $\nabla_{\boldsymbol{w}} f$ is the derivative. A variety of stochastic processes, from simple processes like the Ornstein-Uhlenbeck (OU) process to more complex non-Markov processes, fall outside of this limited class and cannot be learned using CTFP (see supplementary material for formal proofs); (2) Many normalizing flow architectures are compositions of Lipschitz-continuous transformations [7, 6, 16]. Certain stochastic processes that are non-Lipschitz transformations of simple processes cannot be modeled by CTFP without prior knowledge about the functional form of the observable processes and custom-tailored normalizing flows with non-Lipschitz transformations [18]. For example, geometric Brownian motion (GBM) can be written as an exponential transformation of Brownian motion, but it is impossible for CTFP models to represent geometric Brownian motion unless an exponential activation function is added to the output.

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 decode each sample z of $\boldsymbol{Z}$ into a stochastic processes with continuous trajectories. Latent CTFP can be used to estimate finite-dimensional distributions with variational approximation. However, it is unclear how $\boldsymbol{Z}$ with finite dimensions can improve the representation power of modeling continuous stochastic processes.

3 Model

Equipped with these tools, we can now describe our problem setting and model architecture. 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,

$$\mathcal{L} = \log p_{\boldsymbol{X}_{t_1},\ldots,\boldsymbol{X}_{t_n}}(\boldsymbol{x}_{t_1},\ldots,\boldsymbol{x}_{t_n}), \tag{4}$$

for an inhomogeneous collection of sequences with varying lengths and time stamps. At test-time, in addition to the maximization of log-likelihoods, we are also interested in sampling sparse, dense, or irregular trajectories with finite-dimensional distributions that conform with these log-likelihoods. 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 architecture of the normalizing flow itself can be specified in multiple ways, e.g., as an augmented neural ODE [13] or as a series of affine transformations [9]. 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 $\boldsymbol{W}_t$,

$$\mathrm{d}\boldsymbol{Z}_t = \boldsymbol{\mu}_\gamma(\boldsymbol{Z}_t, t)\,\mathrm{d}t + \boldsymbol{\sigma}_\gamma(\boldsymbol{Z}_t, t)\,\mathrm{d}\boldsymbol{W}_t, \tag{5}$$

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 (see Appendix for details). 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 [10, 9, 5], we model $\boldsymbol{X}_t$ as

$$\boldsymbol{X}_t = F_\theta(\boldsymbol{O}_t; \boldsymbol{Z}_t, t), \tag{6}$$

where $\boldsymbol{O}_t$ is a d-dimensional stochastic 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 the base process $\boldsymbol{O}_t$ are continuous with respect to time t. Unlike [10], who use a Wiener process as base process, we use the Ornstein–Uhlenbeck (OU) process, which 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 and is primarily determined by the latent process $\boldsymbol{Z}_t$ and flow transformation F_θ.

Flow Architecture. The continuously indexed normalizing flow $F_\theta(\,\cdot\,; \boldsymbol{z}_t, t)$ can be implemented in multiple ways. Deng et al. use an ANODE [13], defined as the solution to the initial value problem

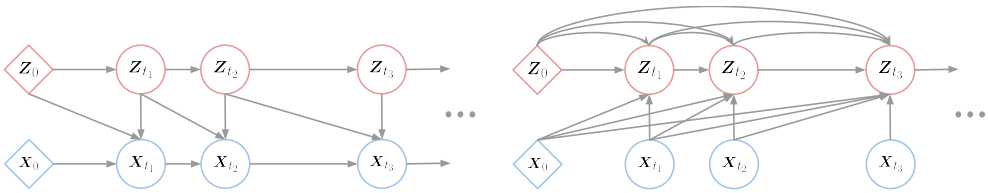
$$\frac{\mathrm{d}}{\mathrm{d}\tau}\begin{pmatrix}\boldsymbol{h}(\tau)\\ \boldsymbol{a}(\tau)\end{pmatrix} = \begin{pmatrix} f_\theta(\boldsymbol{h}(\tau), \boldsymbol{a}(\tau), \tau)\\ g_\theta(\boldsymbol{a}(\tau), \tau)\end{pmatrix}, \quad \begin{pmatrix}\boldsymbol{h}(\tau_0)\\ \boldsymbol{a}(\tau_0)\end{pmatrix} = \begin{pmatrix}\boldsymbol{o}_t\\ (\boldsymbol{z}_t, t)^T\end{pmatrix}, \tag{7}$$

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. Alternatively, Cornish et al. propose a variant of continuously indexed normalizing flows based on a series of N affine transformations f_i,

$$\boldsymbol{h}_t^{(i+1)} = f_i(\boldsymbol{h}_t^{(i)}; \boldsymbol{z}_t, t) = k(\boldsymbol{h}_t^{(i)} \cdot \exp(-u^{(i)}(\boldsymbol{z}_t, t)) - v^{(i)}(\boldsymbol{z}_t, t)), \tag{8}$$

where $\boldsymbol{h}_t^{(0)} = \boldsymbol{o}_t$, $\boldsymbol{h}_t^{(N)} = \boldsymbol{x}_t$, $u^{(i)}$ and $v^{(i)}$ are unconstrained transformations, and k is an invertible mapping like a residual flow [7]. The temporal relationship between $\boldsymbol{X}_t$ and $\boldsymbol{Z}_t$ from a graphical model point of view is shown in Fig. 2.

[1] More precisely, the conditional distribution $p_{\mathbf{O}_{t_i}|\mathbf{O}_{t_j}}(\mathbf{o}_{t_i}|\mathbf{o}_{t_j})$ must exist in closed form for any $t_j < t_i$.



(a) Generation (b) Inference

Figure 2: **Graphical Model of Generation and Inference.** Red circles represent latent variable and blue circles represent observed variables. $\boldsymbol{Z}_0$ and $\boldsymbol{X}_0$ are assumed to be given.

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.(4)) induced by a time-dependent decoding of an SDE (Eq.(6)) 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.(5). 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 distribution of the latent process. In summary, we first express the observational log-likelihood as a marginalization over piecewise factors conditioned on a latent trajectory, then approximate this intractable computation with a piecewise variational posterior process, and finally derive a variational lower bound for it.

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,

$$\begin{aligned}\mathcal{L} &= \log p_{\boldsymbol{X}_{t_1},\ldots,\boldsymbol{X}_{t_n}}(\boldsymbol{x}_{t_1},\ldots,\boldsymbol{x}_{t_n}) \\ &= \log \mathbb{E}_{\omega\sim\boldsymbol{W}_t}\left[p_{\boldsymbol{X}_{t_1},\ldots,\boldsymbol{X}_{t_n}|\boldsymbol{Z}_t}(\boldsymbol{x}_{t_1},\ldots,\boldsymbol{x}_{t_n}|\boldsymbol{Z}_t(\omega))\right] \\ &= \log \mathbb{E}_{\omega\sim\boldsymbol{W}_t}\left[\prod_{i=1}^{n} p_{\boldsymbol{X}_{t_i}|\boldsymbol{X}_{t_{i-1}},\boldsymbol{Z}_{t_i},\boldsymbol{Z}_{t_{i-1}}}(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}},\boldsymbol{Z}_{t_i}(\omega),\boldsymbol{Z}_{t_{i-1}}(\omega))\right],\end{aligned} \tag{9}$$

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,

$$\begin{aligned}&\log p_{\boldsymbol{X}_{t_i}|\boldsymbol{X}_{t_{i-1}},\boldsymbol{Z}_{t_i},\boldsymbol{Z}_{t_{i-1}}}(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}},\boldsymbol{Z}_{t_i}(\omega),\boldsymbol{Z}_{t_{i-1}}(\omega)) \\ &= \log p_{\boldsymbol{O}_{t_i}|\boldsymbol{O}_{t_{i-1}}}(\boldsymbol{o}_{t_i}|\boldsymbol{o}_{t_{i-1}}) - \log\left|\det\frac{\partial F_\theta(\boldsymbol{o}_{t_i};\boldsymbol{Z}_{t_i}(\omega),t_i)}{\partial \boldsymbol{o}_{t_i}}\right|,\end{aligned} \tag{10}$$

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

Piecewise Construction of Variational Posterior. Computing the expectation over latent trajectories $\boldsymbol{Z}_t(\omega)$ is typically intractable. Therefore, 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. Previous methods [24] use a single SDE to propose the variational posterior, which implies that the posterior process is still restricted to be a Markov process. Instead, we develop a 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 stationary and independent increments of Wiener processes, i.e., $\boldsymbol{W}_{s+t} - \boldsymbol{W}_s$ has the same distribution as $\boldsymbol{W}_t - \boldsymbol{W}_0$. 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 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

$$\mathrm{d}\widehat{\boldsymbol{Z}}_t = \boldsymbol{\mu}_\gamma(\widehat{\boldsymbol{Z}}_t, t + t_{i-1})\,\mathrm{d}t + \boldsymbol{\sigma}_\gamma(\widehat{\boldsymbol{Z}}_t, t + t_{i-1})\,\mathrm{d}\boldsymbol{W}_t^{(i)}, \tag{11}$$

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

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

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

$$\mathrm{d}\tilde{\boldsymbol{Z}}_t = \boldsymbol{\mu}_{\phi_i}(\tilde{\boldsymbol{Z}}_t, t + t_{i-1})\,\mathrm{d}t + \boldsymbol{\sigma}_\gamma(\tilde{\boldsymbol{Z}}_t, t + t_{i-1})\,\mathrm{d}\boldsymbol{W}_t^{(i)} \tag{13}$$

for each expectation $\mathbb{E}_{\omega^{(i)}\sim \boldsymbol{W}_t^{(i)}}\Big[p(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \boldsymbol{z}_{t_i}, \boldsymbol{z}_{t_{i-1}}, \omega^{(i)})\mathbb{E}_{\omega^{(i+1)}\sim \boldsymbol{W}_t^{(i+1)}}[\ldots]\ldots\Big]$ in Eq.(12).

Variational Lower Bound with Piecewise Reweighting. The posterior SDEs in Eq.(13) form the basis for a variational approximation of the expectations in Eq.(12). Specifically, sampling $\tilde{\boldsymbol{z}}$ from the posterior SDE, the expectation can be rewritten as

$$\mathbb{E}_{\omega^{(i)}\sim \boldsymbol{W}_t^{(i)}}\Big[p(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \tilde{\boldsymbol{z}}_{t_i}, \boldsymbol{z}_{t_{i-1}}, \omega^{(i)})\boldsymbol{M}^{(i)}(\omega^{(i)})\mathbb{E}_{\omega^{(i+1)}\sim \boldsymbol{W}_t^{(i+1)}}[\ldots]\ldots\Big]\,, \tag{14}$$

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 reweighting term for the sampled trajectory between the prior latent SDE (Eq.(11)) and posterior latent SDE (Eq.(13)), 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:

$$\begin{aligned}\mathcal{L} &= \log \mathbb{E}_{\omega^{(1)}\sim \boldsymbol{W}_t^{(1)}}\Big[p(\boldsymbol{x}_{t_1}|\boldsymbol{x}_{t_0}, \tilde{\boldsymbol{z}}_{t_1}, \tilde{\boldsymbol{z}}_{t_0}, \omega^{(1)})\boldsymbol{M}^{(1)}\ldots\mathbb{E}_{\omega^{(i)}\sim \boldsymbol{W}_t^{(i)}}\Big[p(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \tilde{\boldsymbol{z}}_{t_i}, \tilde{\boldsymbol{z}}_{t_{i-1}}, \omega^{(i)})\boldsymbol{M}^{(i)}\ldots\Big]\Big] \\ &= \log \mathbb{E}_{\omega^{(1)},\ldots,\omega^{(n)}\sim \boldsymbol{W}_t^{(1)}\times\cdots\times\boldsymbol{W}_t^{(n)}}\left[\prod_{i=1}^{n} p(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \tilde{\boldsymbol{z}}_{t_i}, \tilde{\boldsymbol{z}}_{t_{i-1}}, \omega^{(i)})\boldsymbol{M}^{(i)}(\omega^{(i)})\right] \\ &\geqslant \mathbb{E}_{\omega^{(1)},\ldots,\omega^{(n)}\sim \boldsymbol{W}_t^{(1)}\times\cdots\times\boldsymbol{W}_t^{(n)}}\left[\sum_{i=1}^{n}\log p(\boldsymbol{x}_{t_i}|\boldsymbol{x}_{t_{i-1}}, \tilde{\boldsymbol{z}}_{t_i}, \tilde{\boldsymbol{z}}_{t_{i-1}}, \omega^{(i)}) + \sum_{i=1}^{n}\log \boldsymbol{M}^{(i)}(\omega^{(i)})\right].\end{aligned} \tag{15}$$

The bound above can be further extended into a tighter bound in IWAE [4] 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, $\{\boldsymbol{X}_{t_1}, \ldots, \boldsymbol{X}_{t_i}\}$, and the sequence of previously sampled latent states, $\{\boldsymbol{Z}_{t_1}, \ldots, \boldsymbol{Z}_{t_{i-1}}\}$, as inputs. As a result, the variational posterior distributions of latent states $\boldsymbol{Z}_{t_i}$ are no longer constrained to be Markov and the parameterization of the variational posterior can flexibly adapt to different time grids.

4 Related Work

The introduction of neural ODEs [6] unified modeling approaches based on differential equations and modern machine learning. Explorations of time series modeling leveraging this class of approach have been conducted, which we review here.

The ODE-RNN model [32] propagates a latent state across time using ordinary differential equations. As a result, the entire latent trajectory is solely determined by its initial value. Even though latent ODE models have continuous latent trajectories, the latent state is decoded into observations at each time step independently. Neural controlled differential equations (CDEs) [20] and rough differential equations (RDEs) [25] propagate a hidden state across time continuously using controlled differential equations driven by functions of time interpolated from observations on irregular time grids. Neural

ODE Processes (NDPs) [26] construct a distribution over neural ODEs in order to effectively manage uncertainty over the underlying process that generates the data.

Among existing time series works with continuous dynamics, the latent SDE model [24] is most similar to ours. The SDE model includes an adjoint sensitivity method for training SDEs; the derivation of the variational lower bound in our proposed model is based on the same principle of trajectory re-weighting between two stochastic differential equations. The posterior process there is defined as a global stochastic differential equation. Our model further exploits the given observation time grid of each sequence to induce a piecewise posterior process with richer structure.

The continuous-time flow process [10] (CTFP) models irregular time series data as an incomplete realization of continuous-time stochastic processes. We have discussed the limits of CTFP in Sec. 2.2. Because CTFP is a generative model that is guaranteed to generate continuous trajectories, we use it as the decoder of a latent process for better inductive bias in modeling continuous dynamics.

Alternative training frameworks have also been explored for neural SDEs [17, 19]. Stochastic differential equations can be learned as latent dynamics with a variational approximation [17]. Connections between generative adversarial network (GAN) objectives and neural SDEs have been drawn [19]. Brownian motion inputs can be mapped to time-series outputs. Alongside a CDE-based discriminator GAN-based training can be conducted to obtain continuous-time generative models.

5 Experiments

We compare our proposed architecture against several baseline models with continuous dynamics that can be used to fit irregular time-series data, including CTFP, latent CTFP, latent SDE, and latent ODE. We also run experiments with Variational RNN (VRNN) [8], a model that can be used to fit sequential data. However, VRNN does not define a continuous dynamical model and cannot generate trajectories with finite-dimensional distributions consistent with its log-likelihood estimation. We defer all the implementation and training details to the supplementary material.

5.1 Synthetic Data Study

We evaluated our model on sets of synthetic data sampled from known stochastic processes, verifying the ability of our model to capture a variety of continuous dynamics. We use the following processes:

Geometric Brownian Motion $\mathrm{d}\boldsymbol{X}_t = \mu \boldsymbol{X}_t \,\mathrm{d}t + \sigma \boldsymbol{X}_t \,\mathrm{d}\boldsymbol{W}_t$. Even though geometric Brownian motion can theoretically be captured by the CTFP model, it would require the normalizing flow to be non-Lipschitz. In contrast, there is no such constraint for the proposed model.

Linear SDE $\mathrm{d}\boldsymbol{X}_t = (a(t)\boldsymbol{X}_t + b(t)) \,\mathrm{d}t + \sigma(t) \,\mathrm{d}\boldsymbol{W}_t$. The drift term of the SDE is a linear transformation of $\boldsymbol{X}_t$ and the variance term is a deterministic function of time t. An application of Itô's lemma shows that the solution is a stochastic process that cannot be captured by CTFP.

Continuous AR(4) Process. This process tests our model's ability to capture non-Markov processes:

$$\begin{aligned} \boldsymbol{X}_t &= [d, 0, 0, 0]\boldsymbol{Y}_t, \\ \mathrm{d}\boldsymbol{Y}_t &= A\boldsymbol{Y}_t \,\mathrm{d}t + e \,\mathrm{d}\boldsymbol{W}_t, \end{aligned}, \quad \text{where } A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_1 & a_2 & a_3 & a_4 \end{bmatrix}, \; d = 1, \; e = [0, 0, 0, 1]^T. \tag{16}$$

Stochastic Lorenz Curve. Here, we test our model's ability to capture multi-dimensional (3D) data:

$$\begin{aligned} \mathrm{d}\boldsymbol{X}_t &= \sigma(\boldsymbol{Y}_t - \boldsymbol{X}_t) \,\mathrm{d}t + \alpha_x \,\mathrm{d}\boldsymbol{W}_t, \\ \mathrm{d}\boldsymbol{Y}_t &= (\boldsymbol{X}_t(\rho - \boldsymbol{Z}_t) - \boldsymbol{Y}_t) \,\mathrm{d}t + \alpha_y \,\mathrm{d}\boldsymbol{W}_t, \\ \mathrm{d}\boldsymbol{Z}_t &= (\boldsymbol{X}_t\boldsymbol{Y}_t - \beta\boldsymbol{Z}_t) \,\mathrm{d}t + \alpha_z \,\mathrm{d}\boldsymbol{W}_t. \end{aligned} \tag{17}$$

In all cases, we sample the observation time stamps from a homogeneous Poisson process with rate λ. To demonstrate our model's ability to generalize to different time grids, we evaluate it using different rates λ. An approximate numerical solution to SDEs is obtained using the Euler-Maruyama scheme for the Itô integral.

Results on synthetic datasets are displayed in Table 1. We train and evaluate all the models on observations of sample trajectories in the interval $[0, 30]$ with observation time stamps sampled from a

Table 1: **Quantitative Evaluation (Synthetic Data)**. We show test negative log-likelihoods (NLLs) of four synthetic stochastic processes across different models. Below each process, we indicate the intensity of the Poisson point process from which the timestamps for the test sequences were sampled for testing. [GBM: geometric Brownian motion (ground truth NLLs: $[\lambda = 2, \lambda = 20] = [0.388, -0.788]$); LSDE: linear SDE; CAR: continuous auto-regressive process; SLC: stochastic Lorenz curve]

Model	GBM		LSDE		CAR		SLC	
	$\lambda = 2$	$\lambda = 20$	$\lambda = 2$	$\lambda = 20$	$\lambda = 2$	$\lambda = 20$	$\lambda = 20$	$\lambda = 40$
VRNN	0.435	-0.595	-0.626	-1.647	2.351	2.159	6.617	6.616
Latent ODE	2.139	2.028	0.900	0.847	5.030	4.942	9.088	9.066
CTFP	3.023	0.708	-0.474	-1.783	372.557	48.514	0.342	-0.873
Latent CTFP	1.502	-0.131	-0.460	-1.775	415.480	69.050	2.831	0.909
Latent SDE	1.233	1.340	-0.001	0.1524	4.9342	4.9079	8.353	8.271
CLPF (**ours**)	**0.435**	**-0.707**	**-0.826**	**-1.923**	**1.325**	**-0.054**	**-2.899**	**-4.255**

Poisson process with $\lambda = 2$ except for Stochastic Lorenz Curve which is sampled in the interval $[0, 2]$. The results on synthetic datasets show the proposed CLPF's competitive performance across the board. On geometric Brownian motion and LSDE data, our models and latent SDE outperform CTFP models. We attribute the competitive performance to the expressive power of a generic SDE over static latent variables. CTFP models perform relatively poorly on the Continuous AR(4) process which is non-Markov. As the Continuous AR(4) process has an underlying 4-dimensional stochastic process $\boldsymbol{Y}_t$ and is generated by projecting $\boldsymbol{Y}_t$ to the one-dimensional observation space $\boldsymbol{X}_t$, models driven by a high-diemsional latent process like latent SDE and CLPF show relatively better performance.

To evaluate the model's performance in capturing the continuous dynamics, we increase the (average) density of observation and generate observation time stamps from a Poisson process with $\lambda = 20$ ($\lambda = 40$ for SLC). The results show models that can generate continuous trajectories, including CTFP, latent CTFP and CLPF, generalize better to dense observations than other models. We visualize qualitative sample sequences from CLPF and VRNN models trained with data from LSDE on dense time grids to approximate trajectories in a continuous time interval and compare them with samples from LSDE in the first row of Fig. 3. We run both CLPF and VRNN on a time grid between 0 and 30 with a gap of 0.01 between steps. Samples from CLPF models share important visual properties with samples from the ground truth process. Despite its competitive density estimation results on sparse time grids, VRNN fails to generate sequences visually similar to the ground truth.

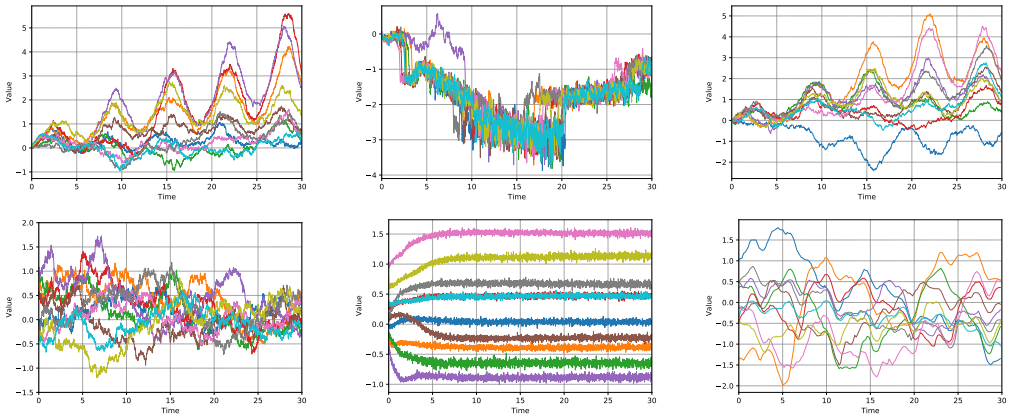
5.2 Real-world Data

Many real-world systems, despite having continuous dynamics, are recorded via observations at a fixed sampling rate. Mujoco-Hopper [32], Beijing Air-Quality Dataset (BAQD) [3] and PTB Diagnostic Database (PTBDB) [33] are examples of such datasets. We create training/testing data at irregular times by drawing times from a Poisson process and mapping to their nearest observed sample points (see details in Supplementary).

We demonstrate the ability of our model to fit this type of observation as a surrogate for its ability to capture real-world continuous dynamics. Results are reported in Table 3. Our model achieves competitive performance with respect to other methods that generate continuous dynamics. We do observe low NLL by VRNN on these data, though note that VRNN does not generate true continuous dynamics. We also visualize one dimension of sample sequences generated from CLPF and VRNN models trained with BAQD data on a dense time grid and compare with ground truth samples from the dataset in Fig. 3 (2nd row). Despite the favourable NLL of density estimation on sparse irregular time grids, we can see VRNN fails to generate continuous time trajectories. Moreover, individual sequences generated by the VRNN model show a lack of variation in common with ground truth sequences from the dataset and sequences generated by CLPF. We believe VRNN is capturing local piecewise constant aspects of the data, hence obtaining low NLL as measured on individual sample points though not capturing the distribution over sequences.

5.3 Ablation Study

We compared the proposed CLPF models against three variants that are obtained by making changes to key components of CLPF. The results with these models are shown in Table 2. **CLPF-Markov** is a variant created by replacing the piecewise construction with a single Markov posterior process in latent SDE [24] style. **CLPF-Independent** is a variant where the continuously indexed normalizing



(a) CLPF (b) VRNN (c) Ground Truth

Figure 3: **Qualitative Samples on Dense Time Grid of CLPF and VRNN Models.** We generate the sample sequences from CLPF and VRNN trained with LSDE (first row) and BAQD (second row) data and compare the samples generated from the model to samples from the ground truth process (LSDE) or training set (BAQD).

Table 2: **Ablation Study (Synthetic Data).** We show test negative log-likelihood across different variants of the proposed models. We report IWAE bounds using $K = 125$ samples with observation time stamps sampled from a Poisson point process with $\lambda = 2$. [CLPF-Markov: single posterior SDE instead of piecewise construction; CLPF-Independent: independent decoder instead of CTFP-decoder; CLPF-Wiener: Wiener base process instead of OU-process]

Model	GBM	LSDE	CAR	SLC
CLPF-Markov	0.456	-0.823	1.358	-2.532
CLPF-Independent	0.891	-0.774	5.109	7.924
CLPF-Wiener	0.391	-0.282	1.438	-1.748
Latent SDE	1.233	-0.001	4.934	8.353
CLPF	0.435	-0.826	1.325	-2.899

Table 3: **Quantitative Evaluation (Real-World Data).** Test negative log-likelihood is shown (lower values better). For CTFP, the reported values are exact; for the other three models, we report IWAE bounds using $K = 125$ samples. CLPF-ANODE stands for CLPF model implemented with augmented neural ODE. CLPF-iRes stands for CLPF implemented with indexed residual flows.

Model	Mujoco [32]	BAQD [3]	PTBDB [33]
VRNN [8]	-12.153	-6.985	-2.012
Latent ODE [32]	23.222	-0.107	-1.012
Latent SDE [24]	3.210	1.224	-1.158
CTFP [10]	-6.318	-0.093	-1.281
Latent CTFP [10]	-6.761	-0.331	-1.692
CLPF-ANODE (**ours**)	-0.903	-0.371	-1.521
CLPF-iRes (**ours**)	-6.939	-0.195	-1.434

flows and CTFP are replaced with static distributions and independent decoders. **CLPF-Wiener** is a variant that uses Wiener process as the base process for the CTFP decoder.

CLPF-Markov differs from CLPF in a manner similar to the difference between latent SDE and CLPF-Independent. In both comparisons, we can see modest improvements of performance when we replace the posterior process defined as a single SDE with a piecewise constructed posterior. Both the comparisons between CLPF and CLPF-Independent and between CLPF-Markov and latent SDE show that CTFP-style decoding into continuous trajectories is indeed a better inductive bias when modeling data with continuous dynamics. The better performance of CLPF over CLPF-Wiener justifies our choice of an OU process with stationary and bounded variance over a Wiener process given the model's expressiveness.

6 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. An ablation study as well as a comparison to state-of-the-art baselines for continuous dynamics have demonstrated the effectiveness of our contributions on both synthetic and real-world datasets. A set of qualitative results support our findings. In the future, we plan to explore the theoretical properties of our model in more detail, including an analysis of its universal approximation properties and its ability to capture non-stationary dynamics.

References

[1] Cedric Archambeau, Dan Cornford, Manfred Opper, and John Shawe-Taylor. Gaussian process approximations of stochastic differential equations. In *Gaussian Processes in Practice*, pages 1–16. PMLR, 2007.

[2] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Joern-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning (ICML)*, pages 573–582. PMLR, 2019.

[3] R Bousseljot, D Kreiseler, and A Schnabel. Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet. *Biomedizinische Technik/Biomedical Engineering*, 40(s1):317–318, 1995.

[4] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

[5] Anthony Caterini, Rob Cornish, Dino Sejdinovic, and Arnaud Doucet. Variational inference with continuously-indexed normalizing flows. *arXiv preprint arXiv:2007.05426*, 2020.

[6] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6571–6583, 2018.

[7] Tian Qi Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9913–9923, 2019.

[8] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988, 2015.

[9] Rob Cornish, Anthony Caterini, George Deligiannidis, and Arnaud Doucet. Relaxing bijectivity constraints with continuously indexed normalising flows. In *International Conference on Machine Learning (ICML)*, pages 2133–2143. PMLR, 2020.

[10] Ruizhi Deng, Bo Chang, Marcus A Brubaker, Greg Mori, and Andreas Lehrmann. Modeling continuous stochastic processes with dynamic normalizing flows. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[11] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[12] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations (ICLR)*, 2017.

[13] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented Neural ODEs. In *NeurIPS*, 2019.

[14] Ramazan Gençay, Michel Dacorogna, Ulrich A Muller, Olivier Pictet, and Richard Olsen. *An introduction to high-frequency finance*. Elsevier, 2001.

[15] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101 (23):e215–e220, 2000.

[16] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, and David Duvenaud. Scalable reversible generative models with free-form continuous dynamics. In *International Conference on Learning Representations (ICLR)*, 2019.

[17] Ali Hasan, João M Pereira, Sina Farsiu, and Vahid Tarokh. Identifying latent stochastic differential equations with variational auto-encoders. *arXiv preprint arXiv:2007.06075*, 2020.

[18] Priyank Jaini, Ivan Kobyzev, Yaoliang Yu, and Marcus Brubaker. Tails of lipschitz triangular flows. In *International Conference on Machine Learning (ICML)*, pages 4673–4681. PMLR, 2020.

[19] Patrick Kidger, James Foster, Xuechen Li, Harald Oberhauser, and Terry Lyons. Neural sdes as infinite-dimensional gans. In *International Conference on Machine Learning (ICML)*. PMLR, 2021.

[20] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[21] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10215–10224, 2018.

[22] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4743–4751, 2016.

[23] Ivan Kobyzev, Simon Prince, and Marcus A Brubaker. Normalizing flows: Introduction and ideas. *arXiv preprint arXiv:1908.09257*, 2019.

[24] Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[25] James Morrill, Cristopher Salvi, Patrick Kidger, James Foster, and Terry Lyons. Neural rough differential equations for long time series. In *International Conference on Machine Learning (ICML)*. PMLR, 2021.

[26] Alexander Norcliffe, Cristian Bodnar, Ben Day, Jacob Moss, and Pietro Lio. Neural ode processes. In *International Conference on Learning Representations (ICLR)*, 2021.

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

[28] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2338–2347, 2017.

[29] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[30] Kira Rehfeld, Norbert Marwan, Jobst Heitzig, and Jürgen Kurths. Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics*, 18(3):389–404, 2011.

[31] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

[32] Yulia Rubanova, Tian Qi Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5321–5331, 2019.

[33] Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. Cautionary tales on air-quality improvement in beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2205):20170457, 2017.

[34] Gilles Zumbach and Ulrich Müller. Operators on inhomogeneous time series. *International Journal of Theoretical and Applied Finance*, 4(01):147–177, 2001.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The list of contributions can be found at the end of Section 1. We have supported our claims via the model description in Section 3 and experimental studies in Section 5.
   (b) Did you describe the limitations of your work? [Yes] In Section 3, the challenges around numerical approximation are described. In our experiments we note challenges regarding discrete fixed samples from continuous signals and analyze performance compared to VRNN.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We proposed an approach for modeling general continuous time series, and we do not use it in specific applications. Time series can contain sensitive information, though our algorithms/experiments are generic in nature and not tuned to any specific task.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We included our code as part of the supplemental material. The data are publicly available.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The details can be found in Sec. 5 and the appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We followed the same routine as in prior work and created the same tables for comparison.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Full experimental details are in the supplemental material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] PTBDB uses an Open Data Commons Attribution License (ODC-By)
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] PTBDB detailed information on consent was not found in our searches.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] PTBDB does not contain personally identifiable information, others are not related to human subjects; none of the datasets in experiments contain offensive content; no new data is being curated.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not create/collect any data.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We did not create/collect any data.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We did not create/collect any data.