STRODE: Stochastic Boundary Ordinary Differential Equation

Hengguan Huang [1] **Hongfu Liu** [1] **Hao Wang** [2] **Chang Xiao** [1] **Ye Wang** [1]

Abstract

Perception of time from sequentially acquired sensory inputs is rooted in everyday behaviors of individual organisms. Yet, most algorithms for time-series modeling fail to learn dynamics of random event timings directly from visual or audio inputs, requiring timing annotations during training that are usually unavailable for real-world applications. For instance, neuroscience perspectives on postdiction imply that there exist variable temporal ranges within which the incoming sensory inputs can affect the earlier perception, but such temporal ranges are mostly unannotated for real applications such as automatic speech recognition (ASR). In this paper, we present a probabilistic ordinary differential equation (ODE), called STochastic boundaRy ODE (STRODE), that learns both the timings and the dynamics of time series data without requiring any timing annotations during training. STRODE allows the usage of differential equations to sample from the posterior point processes, efficiently and analytically. We further provide theoretical guarantees on the learning of STRODE. Our empirical results show that our approach successfully infers event timings of time series data. Our method achieves competitive or superior performances compared to existing state-of-the-art methods for both synthetic and real-world datasets.

1. Introduction

Perception of time from sequentially acquired sensory inputs is rooted in the everyday behavior of the individual organism. Numerous neuroscience studies have provided evidence for close connections between time perception and sensory input from multiple sensory modalities such as audition and vision (Bolognini et al., 2012; Murai & Yotsumoto, 2018; Ulrich et al., 1998). Decoupling the perception of time and content from sensory inputs is crucial for real-time perception in fast-changing environments (Staff, 2014; Toso et al., 2020). For example, to perform interceptive actions such as hitting a moving object, precise timing is required due to significant response latency of neurons; to identify ambiguous phoneme from the acoustic signal in real-time, again, the timing is required such that the human brain is able to strike a balance between postdiction (to ensure sufficient information is integrated from subsequent context) and prediction (to ensure decision outcomes are not delayed relative to the external world) (Gwilliams et al., 2018).

Despite the enormous practical successes of machine learning (in particular, deep learning), most of the algorithms for time-series modeling fail to learn dynamics of random event timings directly from visual or audio inputs, and still require their training data to have timing annotations, such as Latent ODE (Rubanova et al., 2019). However, for some real-world applications in which timing annotations are not available, it is necessary to introduce a "time perception" mechanism to handle such uncertainty. For instance, neuroscience perspectives on postdiction imply that there exist variable temporal ranges within which the incoming sensory inputs can affect the earlier perception, but such temporal ranges are mostly unannotated for real applications such as automatic speech recognition (ASR) (Yu & Li, 2018).

Can gaps between natural and artificial intelligence be bridged further through introducing the "time perception" mechanism? In this paper, we generalize neural ODE in handling a special type of boundary value problem with random boundary times which are described by a temporal point process (TPP). We present a probabilistic ordinary differential equation (ODE) called stochastic boundary ODE (STRODE) that infers both the timings and the dynamics of the time series data without requiring any timing annotations during training.

We adopt variational inference to optimize our model. The boundary time variables involved in STRODE are described as ODEs, consequently posing a major challenge for evaluating evidence lower bound (ELBO) with respect to sampling and inference of such distributions. To mitigate this challenge, we propose a method that allows for joint inference and differentiable sampling of such distributions through

[1]National University of Singapore [2]Rutgers University. Correspondence to: Hengguan Huang <hengguan@comp.nus.edu.sg>, Ye Wang <wangye@comp.nus.edu.sg>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

solving ODEs. Furthermore, the KL term between two differential equations in the ELBO is computationally intractable. We further provide an analytical upper bound for the KL term such that we have a closed-form solution for the ELBO.

Our empirical results show that our approach successfully infers event timings of the time series data. The experiments over Rotating MNIST Thumbnail, a synthetic video thumbnail dataset, show that our model is capable of inferring event timings of the irregularly sampled high-dimensional data without using timing annotations during training, whereby learning of the complex dynamics of irregularly sampled data is achieved. We further apply our model for postdictive modeling using CHiME-5 speech data. We demonstrate that our new model outperforms baseline neural ODEs.

2. Related Work

Ordinary differential equations (ODE) are powerful mathematical tools to describe continuous-time dynamics of an evolving system such as chemical transformation in chemistry (Verwer & Simpson, 1995), and laws of motion in physics and disease spreading (Komarova & Wodarz, 2010) in biology. They have been recently applied to model dynamics of hidden representations of neural networks (Chen et al., 2018). Such neural ODE models have increasingly gained attention in the machine learning community. For example, (Grathwohl et al., 2018) improves upon neural ODE in terms of efficiency by introducing an unbiased stochastic estimator of the likelihood. Since neural ODE assumes the data to be evenly distributed, ODE-recurrent neural network (ODE-RNN) (Rubanova et al., 2019) further extends this approach to handle irregularly sampled data by describing state transitions in recurrent neural network (RNN) as an initial value problem (IVP). Along a different line of research, jump stochastic differential equations (JSDE) (Jia & Benson, 2019) incorporate a temporal point process into neural ODE to model marked point processes. Though these variants of neural ODE are capable of learning dynamics of time series data with or without discontinuities, they require timing annotations for each data sample. Therefore, they are not applicable to our tasks where timing annotations are not available during training.

The closest work to ours is (Chen et al., 2020), in which neural event functions are introduced in neural ODE solvers to enable the learning of termination criteria. However, the design of such event function requires prior knowledge of the dynamical system. In contrast, our model adopts a regenerative point process as prior, whose parameters are learned from data itself. Furthermore, (Chen et al., 2020) focuses on solving IVPs, while our approach generalizes neural ODE in handling a special type of boundary value problem with random boundary conditions.

3. Background

3.1. Temporal Point process

A temporal point process (TPP) is a stochastic process that describes the temporal dependence among events. It provides an effective solution to solve the next-event-prediction problem. A TPP can be equivalently represented as multiple sets of random variables such as arrival times and waiting times. Let $\{t_i\}_{i=0}^N$ be a sequence of arrival times and let $\{\Delta t_i\}_{i=1}^N$ be a sequence of waiting times sampled from a TPP $T(\{p_i(t)\}_{i=0}^N)$, we have:

$$\{t_i\}_{i=0}^N \sim T(\{p_i(t)\}_{i=0}^N) \tag{1}$$

$$t_i \sim p_i(t) \tag{2}$$

$$\Delta t_i = t_i - t_{i-1} \tag{3}$$

where $p_i(t)$ denotes the probability density function of the arrival time t_i. All these random variables involved in TPP are characterized by a conditional intensity function $\lambda(t)$, which is defined to be the rate at which events are expected to occur at time t given histories. For example, the probability density $p_i(t)$ is written as:

$$p_i(t) = \lambda(t) e^{\int_{t_{i-1}}^{t} -\lambda(t)dt} \tag{4}$$

Given the last arrival time t_{i-1}, the expected next arrival time can be generated by:

$$\bar{t}_i = \int_{t_{i-1}}^{+\infty} t p_i(t) dt \tag{5}$$

Let the first arrival time be placed at $t_0 = 0$. If the process follows a Poisson process or more generally a regenerative process, it probabilistically restarts at any arrival time, i.e., for any i, we have

$$p_i(t) = p_0(t) = \lambda(t) e^{\int_0^t -\lambda(t)dt} \tag{6}$$

Parameter learning for TPP can be conducted by maximum likelihood estimation (MLE). Note that in this work, the arrival times of data samples are *not given* during training. Therefore it is impossible to directly cast it as a supervised learning problem that could be solved via MLE.

3.2. From Initial Value Problem to Boundary Value Problem

Ordinary differential equations (ODE), initially exploited to describe the phenomena in physical domain, have been applied to model dynamics of hidden representations of neural networks (Chen et al., 2018; Rubanova et al., 2019). For

simplicity, consider an autoregressive task which predicts future value based on histories. Suppose we are given a sequence of training samples $\mathbf{X} = \{x_i\}_{i=0}^N$ at times $\{t_i\}_{i=0}^N$ and let $t_0 = 0$. For $i < N$ and $t \in [t_i, t_{i+1}]$, the hidden state $h(t)$ satisfies an initial value problem (IVP):

$$h'(t) = f_{\theta_1}(h(t), t),\ h(t_i) = x_i \tag{7}$$

where f_{θ_1} is a neural network. The general solution of $h(t_{i+1})$:

$$h(t_{i+1}) = h(t_i) + \int_{t_i}^{t_{i+1}} f_{\theta_1}(h(t), t)dt \tag{8}$$

where $h(t_{i+1})$ can be approximated using the ODE numerical solvers such as the Euler method and the Runge-Kutta methods (Chen et al., 2018), i.e.:

$$\tilde{h}(t_{i+1}) = \text{ODESolve}(f_{\theta_1}, h(t_i), t_i, t_{i+1}) \tag{9}$$

Combining above N IVPs at all time steps, for $t \in [t_0, t_N]$, we have a boundary value problem (BVP) as follows:

$$h'(t) = f_{\theta_1}(h(t), t) \tag{10}$$

whose boundary conditions are:

$$\{h(t_0) = x_0, h(t_1) = x_1, ..., h(t_N) = x_N\} \tag{11}$$

To solve the above neural ODE that satisfies boundary conditions, we firstly follow Eq.(9) to construct a trial form of the solution of $h(t)$ at each boundary time except t_0. Then the model can be optimized by minimizing the mean squared error (MSE):

$$\frac{1}{N}\sum_{i=1}^{N}(\tilde{h}(t_i) - x_i)^2 \tag{12}$$

It can also be conducted by MLE. More specifically, by assuming that each boundary value x_i follows a Gaussian distribution $p(x_i) = \mathcal{N}(\mu_i, \sigma_i^2)$ whose mean μ_i and variance σ_i^2 are the output of a neural network, we have:

$$x_i \sim \mathcal{N}(\mu_i, \sigma_i^2),\ where\ [\mu_i, \sigma_i^2] = f_{\theta_2}(\tilde{h}(t_i)) \tag{13}$$

It follows that the log-likelihood of all boundary values is:

$$\log P(\mathbf{X}) = \sum_{i=1}^{N} \log p(x_i) \tag{14}$$

Both IVPs and BVPs assume timings of data samples are available during training and thus restricting its capability in handling real-world problems such as postdiction-based ASR, where postdictive temporal ranges are unannotated.

4. Stochastic Boundary Ordinary Differential Equation (STRODE)

4.1. Stochastic Boundary Value Problem

Stochastic boundary value problem (SBVP) is a special type of boundary value problem we propose in which the unobservable boundary conditions/times are described by a stochastic process, e.g. temporal point process. Let us consider a more challenging autoregressive task: we are only given a sequence of training samples $\mathbf{X} = \{x_i\}_{i=0}^N$, in which the first data sample x_0 occurs at $t_0 = 0$, our goal is not just one step ahead prediction, but to infer both boundary times $\{t_i\}_{i=1}^N$ and latent dynamics of the time-series data.

Let $\mathcal{T}(\{p_i(t|\mathbf{x}_i)\}_{i=1}^N)$ be a temporal point process whose realization consists of a sequence of boundary times $\{\tilde{t}_i\}_{i=1}^N$, each follows a density functions $p_i(t|\mathbf{x}_i)$ conditioned on the data sample x_i. For $t \in [0, \tilde{t}_N]$, the hidden states $h(t)$ satisfies an SBVP:

$$h'(t) = f_{\theta_o}(h(t), t) \tag{15}$$

whose boundary conditions are:

$$\{\tilde{t}_i\}_{i=1}^N \sim \mathcal{T}(\{p_i(t|\mathbf{x}_i)\}_{i=1}^N) \tag{16}$$

$$\{h(0) = x_0, h(\tilde{t}_1) = x_1, ..., h(\tilde{t}_N) = x_N\} \tag{17}$$

where f_{θ_o} is a neural network.

4.2. Learning

The neural ODE that involves in SBVP is dubbed as stochastic boundary ordinary differential equation (STRODE). Learning the parameters of neural networks and TPP of the STRODE is, equivalently, solving the SBVP. It is challenging due to the difficulty in estimating parameters of $\mathcal{T}(\{p_i(t|x_i)\}_{i=1}^N)$ without having any observable boundary times. In this work, we adopt variational inference to optimize our model. Let $q_i(t|x_i)$ be the approximate posterior of the boundary time conditioned on the data sample x_i and let $p_i(t)$ be the corresponding prior. The evidence lower bound (ELBO) can be written as:

$$\begin{aligned}\log P(\mathbf{X}) \geqslant \sum_{i=1}^{N}\{&\mathbb{E}_{\tilde{t}_i \sim q_i(t|x_i)} \log p(x_i|\tilde{t}_i) \\ &-\text{KL}(q_i(t|x_i)||p_i(t))\}\end{aligned} \tag{18}$$

Most of the existing TPPs make strong assumptions about the generative processes of the event data and use a fixed parametric form of the intensity function, restricting the expressive power of the respective processes. Instead, our model adopts a general form of TPP in which the arrival time variable could be an arbitrary distribution, consequently posing a major challenge for evaluating ELBO with respect to sampling and inference of such distributions. In the following subsections, we will show how to obtain a closed-form solution of ELBO.

4.3. ODE-based Sampling and Inference of TPP

Sampling from a TPP is usually performed via thinning algorithm (Ogata, 1981). However, such an operation is not differentiable as there is no guarantee that the samples drawn from the algorithm may converge to the exact gradient estimate of the ELBO. To tackle this problem, we model TPP as ODEs such that sampling is achieved by solving such ODEs. More importantly, all operations are differentiable.

Given the last boundary time t_{i-1}, the expected next boundary time is written as:

$$\bar{t}_i = \int_{t_{i-1}}^{+\infty} t q_i(t|x_i) dt \tag{19}$$

Inspired by this, we derive an IVP whose solution allows us to obtain a sequence of boundary time samples $\{\tilde{t}_i\}_{i=1}^N$. To be more specific, suppose the initial boundary time is placed at $\tilde{t}_0 = 0$. Given the last boundary time sample $\tilde{t}_{i-1}$, let $t \in [0, \tilde{t}_{i-1}]$, the next boundary time $\Phi_i(t)$ satisfies an IVP:

$$\Phi_i{}'(t) = -t q_i(t|x_i),\ \Phi_i(0) = \int_0^{+\infty} t q_i(t|x_i) dt \tag{20}$$

Then the general solution of $\tilde{t}_i$ is written as:

$$\tilde{t}_i = \Phi_i(t_{i-1}) = \Phi_i(0) + \int_0^{\tilde{t}_{i-1}} -t q_i(t|x_i) dt$$

The ODE numerical solver, e.g. Euler method fails to generate the approximate solution as $\Phi_i(0)$ is computationally intractable. We, therefore, adopt the neural network f_{θ_Φ} to approximate the solution of $\tilde{t}_i$, such that:

$$\tilde{t}_i = \Phi_i(\tilde{t}_{i-1}) = f_{\theta_\Phi}(\tilde{t}_{i-1}, x_i) \tag{21}$$

Notably, we use $\tilde{t}_{i-1}$ as the bias of the last layer of f_{θ_Φ} to guarantee that $\tilde{t}_i$ is greater than $\tilde{t}_{i-1}$ (implementation details in the Supplement).

Then $\Phi_i{}'(t)$ can be obtained by differentiating $\Phi_i(t)$ with respect to t. It follows that the approximate posterior $q_i(t|x_i)$ can be written as:

$$q_i(t|x_i) = \frac{-\Phi_i{}'(t)}{t} \tag{22}$$

4.4. Prior Point Process

We impose a regenerative process prior on our STRODE. Inspired by (Chen et al., 2018; Rubanova et al., 2019), we model the cumulative conditional intensity function of the process as an initial value ODE. Let $t \in [0, +\infty)$, the cumulative conditional intensity function $\phi_i(t)$ satisfies an initial value problem (IVP):

$$\phi_i'(t) = \lambda_i(t),\ \phi_i(0) = 0 \tag{23}$$

Since our regenerative process prior probabilistically restarts itself for any arrivals, the general solution for conditional intensity function at t can be written as

$$\phi_i(t) = \phi_0(t) = \int_0^t \lambda_i(s) ds \tag{24}$$

such that the prior $p_i(t)$ is reformulated as follows:

$$p_i(t) = \phi_i'(t) e^{-\phi_i(t)} \tag{25}$$

Solving the above IVP requires designing of a specific functional form of the intensity function $\lambda_i(t)$, which results in a TPP with limited expressive power. To alleviate this problem, instead of parameterizing $\lambda_i(t)$, we adopt a neural network f_{θ_ϕ} to approximate $\phi_i(t)$. To ensure that both $\lambda_i(t)$ and $\phi_i(t)$ are constrained to be positive, we adopt similar neural network architecture of (Omi et al., 2019).

4.5. Upper Bound of ODE-based Kullback–Leibler (KL) Divergence

With both $q_i(t|x_i)$ and $p_i(t)$ being defined as differential equations, the KL term of the ELBO defined in Eq.(18) can be written as :

$$\mathrm{KL}(q_i(t|x_i)||p_i(t)) = \int_0^{+\infty} \frac{-\Phi_i{}'(t)}{t} \log \frac{-\Phi_i{}'(t)}{t\phi_i'(t)e^{-\phi_i(t)}} \tag{26}$$

The above KL term is computationally intractable as the upper limit of integration approaches infinity, and it remains unknown whether such an improper integral converges. The following theorem provides an analytical upper bound for the KL term such that we have a closed-form solution for the ELBO.

Theorem 1. *Suppose we are given two arbitrary distributions, $q(t)$ and $p(t)$ with $t \in [0, +\infty)$. Let $m = -e^{-t}$. Let ϵ be a positive real constant and let $g : [-1, 0) \to R$ be a continuous function. There exists a $G : [-1, 0) \to [0, +\infty)$ that satisfies an initial value problem:*

$$G'(m) = g(m),\ \ G(-1) = 0$$
$$where\ g(m) = \frac{-q(-\log(-m))}{m} \log \frac{q(-\log(-m))}{p(-\log(-m))}$$

Such that as $\epsilon \to 0$, we have:

$$\lim_{\epsilon \to 0}(\mathrm{KL}(q(t)||p(t))) \leqslant \lim_{\epsilon \to 0}(G(-\epsilon) + \|G(-2\epsilon) - G(-\epsilon)\|)$$

In the proof of Theorem 1 (Appendix 2), an IVP is introduced to assist derivation of the upper bound of the KL divergence between two arbitrary distributions. Considering specific functional forms of the KL , let $m = -e^{-t}$, let $M = -\log(-m)$ and let $m \in [-1, 0)$, such an IVP can be written as:

$$G'(m) = g(m),\ \ G(-1) = 0 \tag{27}$$

where $g(m)$ is written as:

$$g(m) = \frac{-\Phi_i{}'(M)}{-mM} \log \frac{\Phi_i{}'(M)}{M\phi_i'(M)e^{-\phi_i(M)}} \tag{28}$$

Then Theorem 1 separates the general solution of the KL, $\lim_{l\to 0} G(l)$ into two: $G(-\epsilon)$ and an improper integral:

$$G(-\epsilon) + \lim_{l\to 0} \int_{-\epsilon}^{l} g(m)dm \tag{29}$$

where ϵ in this work is set as the step size of the Euler method that we apply to calculate $G(-\epsilon)$. Obviously, the second term of Eq. (29) can be written as an IVP. Theorem 1 then introduces another IVP with delay ϵ, consequently applying the following lemma to derive the upper bound of the second term.

Lemma 1. *Let ϵ be a positive real constant. Let $U \subset R^n$ be an open set. Let $f_1, f_2 : [a-2\epsilon, a) \to R^n$ be a continuously differentiable function and $\|f_1{}'\| \leqslant M$ where M is a positive constant. Let $y_1, y_2 : [a-\epsilon, a) \to U$ satisfy the initial value problem:*

$$y_1' = f_1(t), \;\; y_1(a-\epsilon) = x_1$$
$$y_2' = f_2(t) = f_1(t-\epsilon), \;\; y_2(a-\epsilon) = x_2$$

Suppose both x_1 and x_2 depend on ϵ. As $\epsilon \to 0$, we have:

$$\lim_{\epsilon\to 0}\left(\lim_{t\to a} \|y_1(t) - y_2(t)\|\right) \leqslant \lim_{\epsilon\to 0} \|x_1 - x_2\|$$

The Lemma (proof given in Appendix 1) states that given two IVPs, one of which is imposed with a delay, the difference between two terminal states is bounded by the difference between initial states. In contrast, Gronwall's Inequality (Theorem 2.1) (Howard, 1998) provides a bound involving an unbounded Lipschitz constant of the ODE, which is not applicable for our problem setting. Therefore, with Lemma 1, the second term in Eq. (29) is upper bounded by:

$$\lim_{l\to 0} \int_{-\epsilon}^{l} g(m)dm \leqslant \|G(-2\epsilon) - G(-\epsilon)\| \tag{30}$$

Combining Eq. (27-30), the upper bound of the KL term of the ELBO is written as:

$$\mathrm{KL}(q_i(t|x_i)||p_i(t)) \leqslant G(-\epsilon) + \|G(-2\epsilon) - G(-\epsilon)\| \tag{31}$$

where both $G(-\epsilon)$ and $G(-2\epsilon)$ are calculated by solving IVP defined in Eq. (27).

4.6. Model Architecture and Implementation of STRODE

In the previous sections, we have discussed using STRODE to describe latent dynamics and infer boundary times of irregularly sampled time series data. Conceptually, both the ODE and the boundary conditions of SBVP are defined over time series data itself. This can be an inappropriate way to model highly structured irregularly sampled high-dimensional data, such as video thumbnails or previews (Liu et al., 2015), which is characterized by complex dependencies.

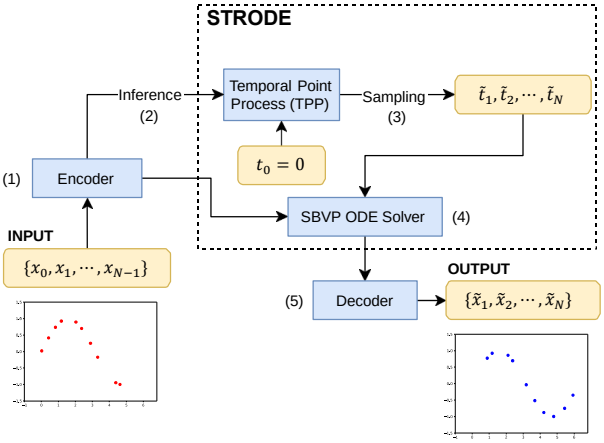

Figure 1: Architecture of STRODE for the toy dataset

STRODE can be conveniently extended to handle such data by introducing an encoder f_{θ_e} to describe its boundary conditions, which can be written as:

$$\{h(0) = f_{\theta_e}(x_0), h(\tilde{t}_1) = f_{\theta_e}(x_1), ..., h(\tilde{t}_N) = f_{\theta_e}(x_N)\} \tag{32}$$

such that the ODE in SBVP is capable of modeling complex dynamics of the hidden representations.

In our experiment on the toy dataset (see Section 5.1), we use STRODE to handle irregularly sampled data from sine waves for an autoregressive task. Here, we provide details for its implementation. (see Supplement for STRODE implementations on other tasks).

Our STRODE consists of five major components: encoder, decoder, inference, sampling and SBVP ODE solver, which are shown in Fig. 1. The detailed implementation of such components are described as following:

(1) **Encoder**: our encoder contains 2 fully connected layers and ReLU, each with 8 hidden nodes.

(2) **Inference**: the inference of the approximate posterior of the boundary time $q_i(t|x_i)$ requires calculating the derivative of $\Phi_i(t)$ with respect to t. In this experiment, $\Phi_i(t)$ is implemented as the neural network f_{θ_Φ} (Eq. (21)), whose architecture contains 2 fully connected layers, each with 16 hidden nodes and Tanh, whose outputs are further transformed into a scalar by another

fully connected layer with Softplus. Then ${\Phi_i}'(t)$ is obtained by computing the derivative of the neural network f_{θ_Φ} with respect to its input of t, using automatic differentiation (Paszke et al., 2017). To ensure that the approximate posterior $q_i(t|x_i)$ is positive-valued, the neural network weights are constrained to be negative. In doing so, if a weight is updated to be a positive value during training, we replace it with zero. Similarly, the corresponding prior $p_i(t)$ requires $\phi_i(t)$, which is implemented by the neural network f_{θ_ϕ}. The architecture of such a neural network is similar to that of f_{θ_Φ}, except that its neural network weights are constrained to be positive during training. To do this, if a weight is updated to be a negative value during training, we replace it with zero.

(3) **Sampling**: the boundary time samples are sequentially generated through adopting Eq. 21.

(4) **SBVP ODE Solver**: with boundary time samples, we follow Eq. 9 to obtain the prediction at $\tilde{t}_i$:

$$\tilde{x}_i = \tilde{h}(\tilde{t}_i) = \text{ODESolve}(f_{\theta_o}, h(\tilde{t}_{i-1}), \tilde{t}_{i-1}, \tilde{t}_i) \tag{33}$$

where the neural network f_{θ_o} includes 2 fully connected layers, each with 8 hidden nodes and Tanh.

(5) **Decoder**: the implementation is similar to what we adopt for the encoder. The difference is that the last layer of the decoder transforms data from a high-dimensional space to a one-dimensional one.

4.7. Latency-free Postdictive Modeling with Evenly Sampled Data

Neuroscience investigations suggest that our visual and auditory system can process information retroactively, such that the incoming sensory inputs can affect the earlier perception (Stiles et al., 2018). There are advantages to this process: the accuracy of "prediction" is reassured with sufficient future information to be integrated. For example, in human speech processing, understanding a word aids in distinguishing its constituent phonemes from another (Gwilliams et al., 2018). However, such a process is difficult to be incorporated into existing algorithms for acoustic modeling, as the temporal range of subsequent context is mostly unannotated. Furthermore, such processes could results in higher latency than other acoustic models.

Here we propose another STRODE variant, dubbed regenerative STRODE, where both the dynamics of ODE and the posterior point process are capable of restarting themselves, allowing the latent temporal range to be captured for every feature frame.

To be specific, suppose we are given a sequence of acoustic features $\mathbf{X} = \{\mathbf{x}_0, ..., \mathbf{x}_N\}$ and training labels $\mathbf{Y} = \{\mathbf{y}_0, ..., \mathbf{y}_N\}$. Given that our posterior point process probabilistically restarts when each feature frame occurs, Eq.(21) for generating the i-th boundary time sample is rewritten as:

$$\tilde{t}_i = \Phi_i(\tilde{t}_0 + \epsilon_0) = f_{\theta_\Phi}(\tilde{t}_0 + \epsilon_0, x_i) \tag{34}$$

where ϵ_0 is a small positive value. We set it as 1×10^{-6}. Then the corresponding postdictive temporal ranges for i-th frame is $[i, i + \tilde{t}_i]$. Let $t \in [i, i + \tilde{t}_i]$, we assume the hidden state $h(t)$ satisfies the following ODE in which the dynamics restarts at each feature frame:

$$h'(t) = f_{\theta_o}(h(t-i), t-i),\ h(i) = f_{\theta_e}(\mathbf{x}_i) \tag{35}$$

Then the ODE solution at time $i + \tilde{t}_i$ can be written as:

$$\tilde{h}(i + \tilde{t}_i) = \text{ODESolve}(f_{\theta_o}, h(i + \tilde{t}_i), i, i + \tilde{t}_i) \tag{36}$$

Next, we use the ODE solution $\tilde{h}(i + \tilde{t}_i)$ as an additional input of the acoustic model. Such design allows our model to look-ahead without compromising input latency.

We further impose a prior distribution for the ODE solution $\tilde{h}(i + \tilde{t}_i)$, which can be treated as a way of injecting prior knowledge for solutions of ODE. For instance, we can use the Gaussian distribution $\mathcal{N}(\mu_0, \sigma_0^2)$, whose mean μ_0 and variance σ_0^2 are the output of a non-linear function f_{θ_p} of x_{i+1}, to constrain the future ODE solution:

$$\tilde{h}(i + \tilde{t}_i) \sim \mathcal{N}(\mu_0, \sigma_0^2),\ where\ [\mu_0, \sigma_0^2] = f_{\theta_p}(\mathbf{x}_{i+1}) \tag{37}$$

For optimization, we simply add the log-likelihood of all ODE solutions $p(\tilde{h}(i + \tilde{t}_i))$ to the original ELBO:

$$\begin{aligned} &\sum_{i=0}^{N} \{\mathbb{E}_{t_i \sim q_i(t|\mathbf{x}_i)} \log p(\mathbf{y}_i|\mathbf{x}_i, t_i) \\ &-\text{KL}(q_i(t|\mathbf{x}_i) \| p_i(t)) + \log p(\tilde{h}(i + \tilde{t}_i))\} \end{aligned} \tag{38}$$

5. Experiments

We evaluate our STRODE on both synthetic and real-world time-series data. More specifically, we conduct the preliminary experiments on irregularly sampled 1D synthetic time-series data and synthetic video thumbnail data based on MNIST to demonstrate its effectiveness in learning both the latent dynamics and timings of the time-series data. We further evaluate our model on CHiME-5, a realistic conversational speech recognition dataset to explore STRODE's potential in posdictive modeling.

5.1. Toy Dataset

We start by conducting an autoregressive task using STRODE on a toy dataset with 1D sequential data. We aim to demonstrate the capability of our STRODE in inferring

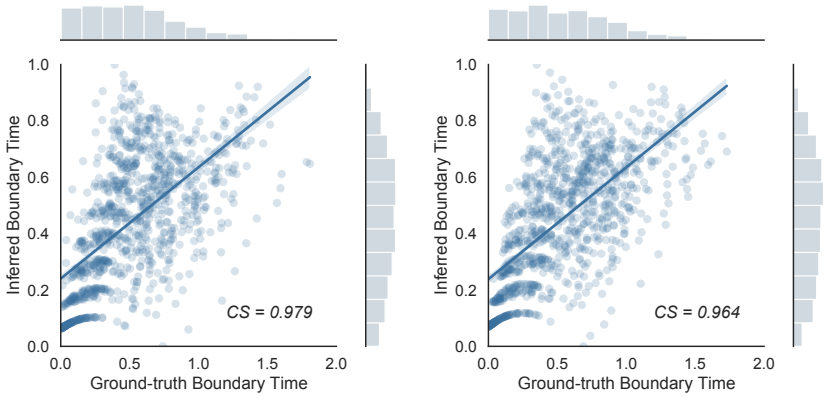

Figure 2: Left: boundary time samples produced by STRODE for sine waves irregularly sampled with Hawkes process data; Right: boundary time samples produced by STRODE for sine waves irregularly sampled with Poisson process data.

informative boundary times given the time series data alone. Our datasets are generated by the sine function $\mathrm{A}\sin(wt)+\eta$, where the amplitude $\mathrm{A} = 1$; the frequency $w = 1.0$; η represents Gaussian random noises; t is the time point drawn from point processes. In this experiment, we consider two general point processes, namely a Poisson process with the conditional intensity $\lambda = 10.0$ and a Hawkes process with the conditional intensity $\lambda = 10.0$, $\alpha = 0.5$, $\beta = 1.0$. For each process, we create a dataset containing 5200 sequences with 10 different time points in each sequence. We use 5000 sequences for training 100 sequences for validation and 100 sequences for testing. Training procedure of STRODE for this task are provided in the Supplement.

5.1.1. Results and Analysis

We evaluate our model by calculating the cosine similarity (CS) between the inferred boundary time and the ground truth. We display the boundary time samples produced by STRODE for both Poisson process data and Hawkes process data shown in Fig. 2. Note that we use min-max normalization to scale the boundary time samples between 0 and 1. We can see that STRODE achieves CS of 0.979 and 0.964 for the Hawkes process data and the Poisson process data respectively. This suggests that our model is capable of inferring timings of the irregularly sampled sine waves.

5.2. Rotating MNIST Thumbnail: Evaluation on Irregularly Sampled High-dimensional Data without Timings

We further study the generalization of STRODE for irregularly sampled high-dimensional data. To do this, we construct a synthetic video thumbnail dataset based on MNIST, the Rotating MNIST Thumbnail. We first generate 12000 video sequences of the rotating MNIST handwritten digits with constant angular velocity. We then adopt a point process and an exponential function to generate time points for selecting frames from video sequences. We generate the corresponding video thumbnails for each video sequence using a Hawkes process with $\lambda = 40$, $\alpha = 0.5$, and $\beta = 1.0$ and an exponential function $t = e^a + \epsilon$ where a is evenly distributed and ϵ is normally distributed, resulting in two subsets of Rotating MNIST Thumbnail: Hawkes and Exponential. We designate video thumbnails that are rotating by 0° to 180° as the training and validation data, and assign video thumbnails rotating by 180° to 360° as the test data. For each subset, we generate 10000 video thumbnails for training, 1000 video thumbnails for validation and 1000 video thumbnails for testing.

5.2.1. Models

We follow implementation in (Rubanova et al., 2019) to incorporate STRODE to RNN. We then obtain a new variant of STRODE, dubbed as *STRODE-RNN*. We compare STRODE and STRODE-RNN with two baselines including: (i) NODE (Chen et al., 2018), which in this task is an extension of neural ODE through adopting convolution layers in the encoder and deconvolution layers in the decoder. (ii) ODE-RNN (Rubanova et al., 2019), which is an extension of ODE-RNN through adopting convolution layers in the encoder and deconvolution layers in the decoder as well. All baselines adopt the same architecture of encoder and decoder and take the same input as our STRODEs while having the similar number of parameters.

5.2.2. Training Procedure

Similar to experiments on toy dataset, our STRODEs on Rotating MNIST Thumbnail is trained by the ELBO, in which the likelihood term is simplified by an MSE term,

Table 1: Cosine similarity (CS) (mean± std) and MSE results on two subsets of Rotating MNIST Thumbnail

DATASET	Hawkes		Exponential	
	CS	MSE ($\times 10^{-3}$)	CS	MSE ($\times 10^{-3}$)
NODE (Chen et al., 2018)	0.907	6.66±0.03	0.923	7.69±0.02
ODE-RNN (Rubanova et al., 2019)	0.907	7.08±0.26	0.923	6.77±0.10
STRODE (Ours)	0.952±0.011	6.42±0.14	0.972±0.006	7.60±0.03
STRODE-RNN (Ours)	**0.978±0.016**	**6.14±0.27**	**0.974±0.003**	**6.74±0.17**

using the Adam optimizer with a learning rate in the range $[2 \times 10^{-4}, 6 \times 10^{-4}]$. We follow the training strategies of β-VAE (Higgins et al., 2017) to reweight the importance of the MSE term and the KL term. We repeat this training procedure across 3 different random seeds.

5.2.3. Results and Analysis

We report test-set CS and MSE on both Hawkes and Exponential subsets of Rotating MNIST Thumbnail, as shown in Table 1. As timings of each video frame are unknown, both NODE and ODE-RNN set the difference between the initial time and terminate time of the ODE solvers as a constant (i.e.$= 1$). As such, a sequence of natural numbers are adopted when calculating CS with ground truth boundary time for both NODE and ODE-RNN. For the Hawkes task, we can see that STRODE outperforms both NODE and ODE-RNN in terms of CS and MSE score and our STRODE-RNN archives the lowest MSE and the highest CS score. This suggests that our STRODEs are capable of inferring both timings and complex dynamics of high-dimensional time-series data. We further apply our STRODEs to handle the video thumbnails drawn from exponential functions. Again, our STRODEs perform much better than baselines in terms of CS score, which demonstrates the expressive power of STRODEs in handling data whose generation doesn't follow a TPP.

5.3. CHiME-5: Latency-free Postdictive Modeling with Evenly Sampled Data

CHiME-5 was originally designed for the CHiME 2018 challenge (Barker et al., 2018). It contains a very challenging problem of conversational speech recognition in everyday home environments. For a fair comparison with (Huang et al., 2020), only the audio data recorded by binaural microphones is employed for training and evaluation in this experiment. The Train, Dev and Eval include about 40 hours, 4 hours, and 5 hours of real conversational speech respectively. The evaluation was performed with a trigram language model trained from the transcription of CHiME-5.

5.3.1. Models

We adopted SRU (Lei et al., 2017) as the building block to construct all RNNs involved in both our STRODE and baselines. Note that Latent ODE (Rubanova et al., 2019) is not adopted as one of our baselines, as it employs a seq2seq architecture, which is not applicable for HMM-based acoustic modeling. For simplicity, we denote regenerative STRODE as STRODE in this task. We compare our STRODE with the following baseline models: (i) ODE-RNN with 9 stacked SRU layers in the encoder and 9 stacked SRU layers in the decoder (ii) RTN with a 6-layer SRU encoder and a 9-layer SRU decoder.

All baselines except RTN take the same input as our STRODE. RTN takes multiple utterances as input, which contains significantly more contextual information than our STRODE and other baselines. To ensure similar numbers of model parameters for different models, we set the number of RNN hidden states per layer to 1,100 for ODE-RNN and 900 for STRODE. The size of ODE solutions for both ODE-RNN and STRODE is set as 128. More implementation details of STRODE are provided in the Supplement.

5.3.2. Training Procedure

For a fair comparison, we adopted the same configuration with (Huang et al., 2020) to train all GMM-HMM. They were then used to derive the state targets for subsequent DNN acoustic model training through forced alignment for Train of CHiME-5. The state targets of CHiME-5 for subsequent STRODE-based and baseline acoustic model training were obtained by aligning the training data with the DNN acoustic model through the iterative procedure outlined in (Dahl et al., 2012). Our STRODE and baselines on CHiME-5 are trained by maximizing the ELBO, using BPTT (Werbos, 1990) and SGD with learning rates ranging from 0.13 to 0.19. We apply a dropout rate of 0.1 to the connections between neural network layers except that of ODE solvers. We reweight the importance of both KL terms and the prior log-likelihood. Both weights are set as 1×10^{-3}.

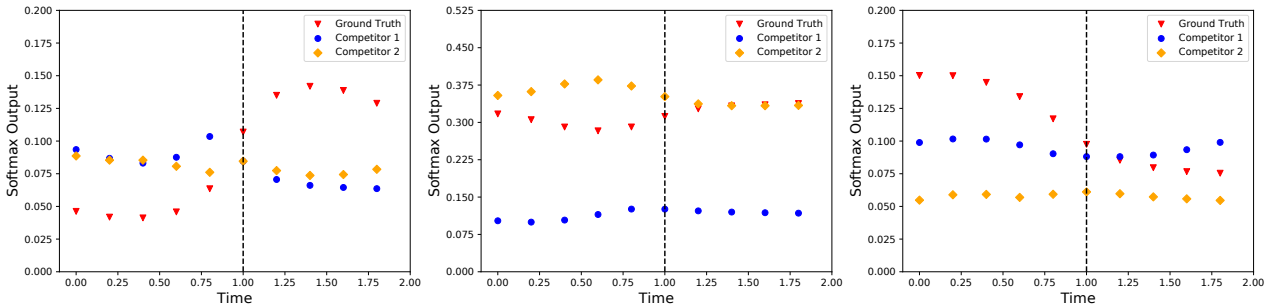

Figure 3: The Softmax outputs by taking the ODE solutions $\{h(i+t)\}$ at future time points as an extra input of the acoustic model. The dotted line corresponds to the original Softmax output of STRODE. Both Fig. 3(left) and Fig. 3(middle) provide examples that match the assumption of postdictive modeling, while Fig. 3(right) provides some counter examples.

Table 2: Model configurations and the training time for CHiME-5. N: number of hidden states per layer; P: number of model parameters; T: Training time per epoch (hr).

Model	N	P	T
ODE-RNN (Rubanova et al., 2019)	1100	77M	0.6
RTN (Huang et al., 2020)	1024	70M	0.3
STRODE (Ours)	900	76M	0.7

Table 3: WER (%) on eval of CHiME-5.

Model	WER
Kaldi DNN (Povey et al., 2011)	64.5
ODE-RNN (Rubanova et al., 2019)	59.0
RTN (Huang et al., 2020)	57.4
STRODE (Ours)	**56.3**

5.3.3. Results and Analysis

Training Speed and Model Complexity of STRODE Table 2 shows the configurations of baseline models and the new STRODE model for CHiME-5. The training time per epoch for CHiME-5 is also reported. In our experiments, the timing experiments use PyTorch package and are performed on Ubuntu 16.04 with a single Intel Xeon Silver 4214 CPU and a GTX 2080Ti GPU. Each model takes around 20 iterations, with their average running time reported. We can see that STRODE runs as fast as ODE-RNN with a similar number of parameters. The running time of STRODE is comparable to RTN, though it runs 2.5 times faster than ours.

Quantitative evaluation of STRODE on CHiME-5 Table 3 shows the word recognition performance of the baseline models and the new STRODE model for CHiME-5. It is observed that ODE-RNN outperforms the DNN baseline from Kaldi s5 (Povey et al., 2011). STRODE performs the best among all models in terms of WER, outperforming the ODE-RNN by 2.7%. Compared with the state-of-the-art acoustic RTN model, our STRODE achieves 0.9% absolute WER reduction even though the input of our STRODE contains much less contextual information than RTN.

Qualitative evaluation of Postdictive Temporal Range Produced by STRODE We generate postdictive temporal range $[i, i+\tilde{t}_i]$ for randomly selected feature frames from utterances in Dev of CHiME-5. For $t \in [0, 2\tilde{t}_i]$, we obtain a sequence of ODE solutions $\{h(i+t)\}$ at future time points, and produce the Softmax output by taking $\{h(i+t)\}$ as an extra input of the acoustic model. In Fig. 3, we only display the Softmax outputs for the ground truth label (red), and the others are from the class labels with the top 2 highest Softmax output values. It is evident from Fig. 3(left) that the Softmax outputs for ground truth label increases over time, and tops exactly after $i + \tilde{t}_i$. Such a pattern matches the assumption of postdictive modeling. For the Fig. 3(middle), STRODE fails to produce accurate results without integrating sufficient information. Interestingly, the STRODE hits the target after $i+\tilde{t}_i$, which suggests that our model requires a longer context for targeting the ground truth label. We also show some counter examples Fig. 3(right) in which STRODE fails to produce accurate results when more contextual information is integrated.

6. Conclusion

We propose a novel neural ODE model named STochastic boundaRy ODE (STRODE) for handling time series data without timing annotation during training. We provide theoretical guarantees on the learning of STRODE. We show that our model is capable of inferring timings and the dynamics of time series data without requiring any timing annotations during training. We demonstrate that our STRODE can be applied to address postdictive modeling. Our experiments on CHiME-5 show that our method outperforms ODE-RNN acoustic model in ASR.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions, Ling Qin and Kai Song for their assistance in proofreading the initial manuscript. This project was partially funded by research grant R-252-000-B78-114 from the Ministry of Education, Singapore.

References

Barker, J., Watanabe, S., Vincent, E., and Trmal, J. The fifth'chime'speech separation and recognition challenge: dataset, task and baselines. *arXiv preprint arXiv:1803.10609*, 2018.

Bolognini, N., Cecchetto, C., Geraci, C., Maravita, A., Pascual-Leone, A., and Papagno, C. Hearing shapes our perception of time: temporal discrimination of tactile stimuli in deaf people. *Journal of Cognitive Neuroscience*, 24(2):276–286, 2012.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366*, 2018.

Chen, R. T., Amos, B., and Nickel, M. Learning neural event functions for ordinary differential equations. *arXiv preprint arXiv:2011.03902*, 2020.

Dahl, G. E., Yu, D., Deng, L., and Acero, A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2012.

Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., and Duvenaud, D. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Gwilliams, L., Linzen, T., Poeppel, D., and Marantz, A. In spoken word recognition, the future predicts the past. *Journal of Neuroscience*, 38(35):7585–7599, 2018.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. *Iclr*, 2(5):6, 2017.

Howard, R. The gronwall inequality. *lecture notes*, 1998.

Huang, H., Xue, F., Wang, H., and Wang, Y. Deep graph random process for relational-thinking-based speech recognition. In *International Conference on Machine Learning*, pp. 4531–4541. PMLR, 2020.

Jia, J. and Benson, A. R. Neural jump stochastic differential equations. *arXiv preprint arXiv:1905.10403*, 2019.

Komarova, N. L. and Wodarz, D. Ode models for oncolytic virus dynamics. *Journal of theoretical biology*, 263(4): 530–543, 2010.

Lei, T., Zhang, Y., Wang, S. I., Dai, H., and Artzi, Y. Simple recurrent units for highly parallelizable recurrence. *arXiv preprint arXiv:1709.02755*, 2017.

Liu, W., Mei, T., Zhang, Y., Che, C., and Luo, J. Multi-task deep visual-semantic embedding for video thumbnail selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3707–3715, 2015.

Murai, Y. and Yotsumoto, Y. Optimal multisensory integration leads to optimal time estimation. *Scientific reports*, 8(1):1–11, 2018.

Ogata, Y. On lewis' simulation method for point processes. *IEEE transactions on information theory*, 27(1):23–31, 1981.

Omi, T., Ueda, N., and Aihara, K. Fully neural network based model for general temporal point processes. *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Povey, D. et al. The Kaldi speech recognition toolkit. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 2011.

Rubanova, Y., Chen, R. T., and Duvenaud, D. Latent ordinary differential equations for irregularly-sampled time series. 2019.

Staff, P. O. Correction: The duration of uncertain times: Audiovisual information about intervals is integrated in a statistically optimal fashion. *PloS one*, 9(4):e96134, 2014.

Stiles, N. R., Li, M., Levitan, C. A., Kamitani, Y., and Shimojo, S. What you saw is what you will hear: Two new illusions with audiovisual postdictive effects. *PloS one*, 13(10):e0204217, 2018.

Toso, A., Fassihi, A., Paz, L., Pulecchi, F., and Diamond, M. E. A sensory integration account for time perception. *bioRxiv*, 2020.

Ulrich, R., Rinkenauer, G., and Miller, J. Effects of stimulus duration and intensity on simple reaction time and response force. *Journal of Experimental Psychology: Human Perception and Performance*, 24(3):915, 1998.

Verwer, J. G. and Simpson, D. Explicit methods for stiff odes from atmospheric chemistry. *Applied Numerical Mathematics*, 18(1-3):413–430, 1995.

Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

Yu, D. and Li, J. Recent progresses in deep learning based acoustic models (updated). *arXiv preprint arXiv:1804.09298*, 2018.