Neural Dynamics Discovery via Gaussian Process Recurrent Neural Networks

Qi She
Intel Labs China
qi.she@intel.com

Anqi Wu
Princeton Neuroscience Institute
Princeton University
anqiw@princeton.edu

Abstract

Latent dynamics discovery is challenging in extracting complex dynamics from high-dimensional noisy neural data. Many dimensionality reduction methods have been widely adopted to extract low-dimensional, smooth and time-evolving latent trajectories. However, simple state transition structures, linear embedding assumptions, or inflexible inference networks impede the accurate recovery of dynamic portraits. In this paper, we propose a novel latent dynamic model that is capable of capturing nonlinear, non-Markovian, long short-term time-dependent dynamics via recurrent neural networks and tackling complex nonlinear embedding via non-parametric Gaussian process. Due to the complexity and intractability of the model and its inference, we also provide a powerful inference network with bi-directional long short-term memory networks that encode both past and future information into posterior distributions. In the experiment, we show that our model outperforms other state-of-the-art methods in reconstructing insightful latent dynamics from both simulated and experimental neural datasets with either Gaussian or Poisson observations, especially in the low-sample scenario. Our codes and additional materials are available at https://github.com/sheqi/GP-RNN_UAI2019.

1 INTRODUCTION

Deciphering interpretable latent *regularity* or *structure* from high-dimensional time series data is a challenging problem for neural data analysis. Many studies and theories in neuroscience posit that high-dimensional neural recordings are noisy observations of some underlying, low-dimensional, and time-varying signal of interest. Thus, robust and powerful statistical methods are needed to identify such latent dynamics, so as to provide insights into latent patterns which govern neural activity both spatially and temporally. A large body of literature has been proposed to learn concise, structured and insightful dynamical portraits from noisy high-dimensional neural recordings [1, 2, 3, 4, 5, 6, 7]. These methods can be categorized on the basis of four modeling strategies ("$\star$" indicates our contributions in these components):

Dynamical model ($\star$) Dynamical models describe the evolution of latent process: how future states depend on present and past states. One popular approach assumes that latent variables are governed by a linear dynamical system [8, 9], while a second choice models the evolution of latent states with a Gaussian process, relaxing linearity and imposing smoothness over latent states [10, 1]. However, linear dynamics cannot capture nonlinearities and non-Markov dynamical properties of complex systems; and Gaussian process only considers the pair-wise correlation of time points, instead of considering explicit temporal dynamics. We argue that the proposed dynamical model in this work is able to both capture the complex state transition structures and model the long short-term temporal dynamics efficiently and flexibly.

Mapping function ($\star$) Mapping functions reveal how latent states generate noise-free observations. A nonlinear transformation is often ignored when pursuing efficient and tractable algorithms. Most previous methods have assumed a fixed linear or log-linear relationship between latent variables and mean response levels [2, 3]. In many neuroscience problems, however, the relationship between noise-free observation space and the quantity it encodes can be highly nonlinear. Gao et al., [4] have explored a nonlinear embedding function using deep neural networks (DNNs), which requires a large amount of data to train a large set of model parameters and can not prop-

agate *uncertainty* from latent space to observation space. In this paper, we employ a non-parametric Bayesian approach, Gaussian process (GP), to model the nonlinear mapping function from latent space to observation space, which requires much less training data and propagates uncertainties with probabilistic distributions.

Observation model Neural responses can be mostly categorized into two types of signals, i.e., continuous voltage data and discrete spikes. For continuous neural responses, people usually use Gaussian distributions as generating distributions. For neural spike trains, a Poisson observation model is commonly considered to characterize stochastic, noisy neural spikes. In this work, we propose models and inference methods for both Gaussian and Poisson responses, but with a focus on the Poisson observation model. Directly modeling Poisson responses with a non-conjugate prior has an intractable solution, especially for complex generative models. In some previous methods, researchers have used a Gaussian approximation for Poisson spike counts through a variance stabilization transformation [12]. In our framework, we apply an effective optimization procedure for the Poisson model.

Inference method ($\star$) In our setting, due to the increased complexity of both the dynamical model and the mapping function, we should provide a more powerful inference method for recognizing latent states. Recent work has focused on utilizing variational inference for scalable computation, which takes advantage of both stochastic and distributed optimization [13]. Additionally, inference networks improve computational efficiency while still keeping rich approximated posterior distributions. One of the choices for inference networks for sequential data is multi-layer perceptrons (MLP) [14]. However, it is insufficient to capture the increasing temporal complexity as the dynamic evolves. Recurrent neural networks (RNNs), e.g., long short-term memory (LSTM) and gated recurrent unit (GRU) structures, are well known to capture dynamical structures for sequential data. We utilize RNNs as inference networks for encoding both past and future time information into the posterior distribution of latent states. Specifically, we use two LSTMs for mapping past and future time points jointly into the mean and diagonal covariance functions of the approximated Gaussian distribution. We show empirically that instead of considering only past time information as other recent works [15, 16], using both past and future time information can retrieve intrinsic latent structures more accurately.

Given current limitations in the dynamical model, mapping function, and inference method, we propose a novel method using recurrent neural networks (RNNs) as the dynamical model, Gaussian process (GP) for the nonlinear mapping function, and bi-directional LSTM structure as the inference network. This combination poses a richly distributed internal state representation and flexible nonlinear transition functions due to the representation power of RNNs (e.g., long short-term memory (LSTM) or gated recurrent unit (GRU) structures). Moreover, it shows expressive power for discovering structured latent space by nonlinear embeddings with Gaussian process thanks to its advantage in capturing uncertainty in a non-parametric Bayesian way. In addition, the bi-directional LSTM with increasing model complexity can further enhance inference capability because it summarizes either the past or the future or both at every time step, forming the most effective approximation to the variational posterior of the latent dynamic. Our framework is evaluated on both simulated and real-world neural data with detailed ablation analysis. The promising performance of our method demonstrates that our method is able to: (1) capture better and more insightful nonlinear, non-periodic dynamics from high-dimensional time series; (2) significantly improve prediction performance over baseline methods for noisy neuronal spiking activities; and (3) robustly and efficiently learn the turning curves of underlying complex neural systems from neuronal recording datasets.

Table 1 summarizes the state-of-the-art methods for extracting latent state space from *high-dimensional spike trains*[1] by varying different model components discussed above. In a nutshell, our contributions are three-fold comparing to the listed methods:

- We propose to capture nonlinear, non-Markovian, long short-term time-dependent dynamics by incorporating recurrent neural networks in the latent variable model. Different from the vanilla RNN, we achieve a stochastic RNN structure by introducing latent variables;

- We incorporate Gaussian process for learning nonlinear embedding functions, which can achieve better reconstruction performance for the low-sample scenario and provide the posterior distribution with uncertainty instead of point estimation in neural networks. Together with RNN, we provide a GP-RNN model (Gaussian Process Recurrent Neural Network) that is capable of capturing better latent dynamics from complex high-dimensional neural population recordings;

[1] We focus on exploring intrinsic latent structures from spike trains, and the related works mentioned here are to our knowledge the most relevant with this research line. Although some excellent works take advantages of both RNN structures and Gaussian process for either modeling or inference [17, 18, 19, 20, 21], they are out of the scope in this work.

Model	Dynamics	Mapping function	Link function	Observation	Inference
PLDS [2]	LDS	Linear	exp	Poisson	LP
PfLDS [4]	LDS	NN	exp	Poisson	VI + inference network
GCLDS [3]	LDS	Linear	exp	Count	VI
LFADS [6]	RNN	Linear	exp	Poisson	VI + inference network
P-GPFA [11]	GP	Linear	Identity	Poisson	LP or VI
P-GPLVM [5]	GP	GP	exp	Poisson	LP
Ours : **GP-RNN**	**RNN**	**GP**	**exp**	**Poisson/Gaussian**	**VI + inference network**

Table 1: Comparison of different models. "PLDS": Poisson linear dynamical system [2]; "PfLDS": Poisson feed-forward neural network linear dynamical systems [4]; "GCLDS": generalized count linear dynamical systems [3]; "P-GPFA": Poisson Gaussian process factor analysis [11]; "P-GPLVM": Poisson Gaussian process latent variable model [5]; and our method GP-RNN: Gaussian process recurrent neural networks. "LDS" denotes Linear Dynamical Systems. "LP" and "VI" indicate Laplace approximation and variational inference, respectively.

- We evaluate the efficacy of different inference networks based on LSTM structures for inference and learning, and demonstrate that utilizing the bi-directional LSTM as the inference network can significantly improve model learning.

2 GAUSSIAN PROCESS RECURRENT NEURAL NETWORK (GP-RNN)

Suppose we have simultaneously recorded spike count data from N neurons. Let $x_{i,t}$ denote the spike count of neuron $i \in \{1, \dots, N\}$ at time $t \in \{1, \dots, T\}$. We aim to discover low-dimensional, time-evolving ($\mathbf{z}_t$ depends on $\mathbf{z}_{1:t-1}$) latent trajectory $\mathbf{z}_t \in \mathbb{R}^L$ ($L \ll N$, and L is the latent dimensionality), which governs the evolution of the high-dimensional neural population $\mathbf{x}_t = [x_{1,t}, x_{2,t}, ..., x_{N,t}] \in \mathbb{R}^N$ at time t.

Recurrent structure latent dynamics: Let $\mathbf{z}_t \in \mathbb{R}^L$ denote a (vector-valued) latent process, which evolves based on a recurrent structure (RNN) to capture the sequential dependence. At each time step t, the RNN reads the latent process $\mathbf{z}_{t-1}$ at the previous time step and updates its hidden state $\boldsymbol{\nu}_t \in \mathbb{R}^H$ by:

$$\boldsymbol{\nu}_t = \mathbf{RNN}_\theta(\mathbf{z}_{t-1}, \boldsymbol{\nu}_{t-1}), \tag{1}$$

where $\mathbf{RNN}_\theta$ is a deterministic nonlinear transition function with parameter θ. $\mathbf{RNN}_\theta$ can be implemented via long short-term memory (LSTM) or gated recurrent unit (GRU). It is denoted that the latent process $\mathbf{z}_t$ is modeled as random variables and $\boldsymbol{\nu}_t$ represents hidden states of the RNN model. We model the latent process $\mathbf{z}_t$ by parameterizing a factorization of the joint sequence probability distribution as a product of conditional probabilities such that:

$$p(\mathbf{z}_1, \cdots, \mathbf{z}_T) = \prod_{t=1}^{T} p(\mathbf{z}_t|\mathbf{z}_1, ..., \mathbf{z}_{t-1}) = \prod_{t=1}^{T} p(\mathbf{z}_t|\mathbf{z}_{<t})$$
$$p(\mathbf{z}_t|\mathbf{z}_{<t}) = p\big(\mathbf{z}_t; g_\psi\left(\boldsymbol{\nu}_t\right)\big), \tag{2}$$

where $g_\psi(\cdot)$ is an arbitrary differentiable function parametrized by ψ. The function $g_\psi(\cdot)$ maps the RNN state $\boldsymbol{\nu}_t$ to the parameter of the distribution of $\mathbf{z}_t$, which is modeled using a feed-forward neural network with 2 hidden layers as:

$$p\big(\mathbf{z}_t; g_\psi\left(\boldsymbol{\nu}_t\right)\big) = \mathcal{N}\big(\mu_{\mathbf{z}_t}, \mathrm{diag}(\sigma^2_{\mathbf{z}_t})\big), \tag{3}$$
$$[\mu_{\mathbf{z}_t}, \sigma^2_{\mathbf{z}_t}] = \mathrm{NN}_{2-\mathrm{layer}}(\boldsymbol{\nu}_t). \tag{4}$$

Nonlinear mapping function: Let $f_i : \mathbb{R}^L \to \mathbb{R}$ denote a nonlinear function mapping from the latent variable $\mathbf{z}_t \in \mathbb{R}^L$ to the i-th element of the observation vector $x_{i,t} \in \mathbb{R}$. f_i is usually referred as the neuronal tuning curve characterizing the firing rate of the neuron as a function of relevant stimulus in neural analysis. We provide a non-parametric Bayesian approach using Gaussian process (GP) as the prior for the mapping function f_i. Noticing that f_i is a time-invariant function, we can omit the notation for time step t and describe the GP prior as,

$$f_i(\mathbf{z}) \sim \mathcal{GP}(0, k_z), \tag{5}$$
$$k_z(\mathbf{z}, \mathbf{z}') = \rho \exp\left(\frac{-||\mathbf{z} - \mathbf{z}'||_2^2}{2\sigma^2}\right), \tag{6}$$

where k_z is a spatial covariance function over its L-dimensional input latent space. Note that the input $\mathbf{z}$ is a random variable with uncertainty (eq. (2)). Given that the neuronal tuning curve is usually assumed to be smooth, we use the common radial basis function (RBF) or smooth covariance function as eq. (6), where $\mathbf{z}'$ are arbitrary points in latent space, ρ is the marginal variance and σ is the length scale. We stack $f_i(\mathbf{z}_t)$ across T

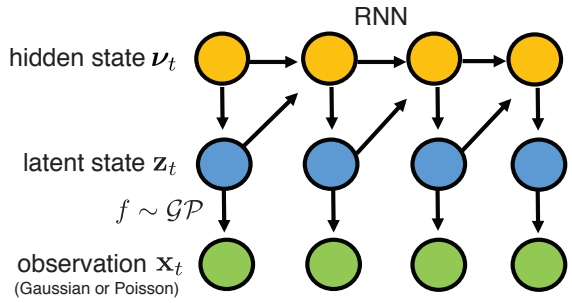

Figure 1: The proposed GP-RNN models the dynamics of hidden states $\boldsymbol{\nu}_t$ (yellow circle) with an RNN structure, and generates latent dynamics z_t (blue circle) given hidden states. Both hidden states $\boldsymbol{\nu}_t$ and latent dynamics z_t contribute to $\boldsymbol{\nu}_{t+1}$. The latent states z_t are mapped to observations x_t (green circle) via a Gaussian process mapping function f.

time steps to obtain $\mathbf{f}_i \in \mathbb{R}^T$. According to the definition of Gaussian process, $\mathbf{f}_i$ forms a multivariate normal distribution given latent vectors at all time steps, as

$$\mathbf{f}_i | \mathbf{z}_{1:T} \sim \mathcal{N}(0, \mathbf{K}_z), \tag{7}$$

with a $T \times T$ covariance matrix $\mathbf{K}_z$ generated by evaluating the covariance function k_z at all pairs of latent vectors in $\mathbf{z}_{1:T}$. Finally, by stacking $\mathbf{f}_i$ for N neurons, we form a matrix $\mathbf{F} \in \mathbb{R}^{N \times T}$ with $\mathbf{f}_i^\top$ on the i-th row.

Observation model: Real-world time series data is often categorized into real-valued data and count-valued data. For real-valued data, the observation model is usually a Gaussian distribution given the firing rate $f_i(\mathbf{z}_t)$ and some additive noise $\epsilon \sim \mathcal{N}(0, l)$. Marginalizing out ϵ, we obtain the observation model as

$$x_{i,t} | f_i, \mathbf{z}_t \sim \mathcal{N}\big(f_i(\mathbf{z}_t), l\big). \tag{8}$$

However the observation following Gaussian distribution is infeasible under count-valued setting. Considering neural spike trains, we assume that the spike rate $\lambda_{i,t} = \exp\big(f_i(\mathbf{z}_t)\big)$ (non-negative value), and the spike count of neuron i at time t is generated as

$$x_{i,t} | f_i, \mathbf{z}_t \sim \mathbf{Poisson}\Big(\exp\big(f_i(\mathbf{z}_t)\big)\Big). \tag{9}$$

In summary, our model uses an RNN structure to capture nonlinearity and long short-term temporal dependence of latent dynamics, while keeping the flexibility of non-parametric Bayesian (GP) in learning nonlinear mapping functions. Finally, we generate Gaussian observations with Gaussian additive noise given spike rates or propagate spike rates via an exponential link function to generate Poisson observations. The graphical model is shown in Fig. 1. Denote that RNN structure is not directly applied for latent process z_t, it is over z_t's prior via a neural network mapping (shown in eq. (1) and (2)), completely different from a simple RNN for latent states z_t as existing works, e.g., LFADS. This modeling strategy, similar to [15], establishes stochastic RNN dynamics, which gives a strong and flexible prior over the latent process. z_t is propagated with well-calibrated uncertainty via Gaussian process to the firing rate function f. The observation x_t is generated from f with Gaussian or Poisson noise based on the applications.

3 INFERENCE FOR GP-RNN

Gaussian response: When the observation is Gaussian, the tuning curve f_i in eq. (8) can be marginalized out due to the conjugacy. Variational Bayes Expectation-Maximization (VBEM) algorithm is adopted for estimating latent states $\mathbf{z}_{1:T}$ (E-step) and parameters $\Theta = \{\theta, \psi, \rho, \sigma\}$ (M-step). In E-step, we need to characterize the full posterior distribution $p(\mathbf{z}_{1:T} | \mathbf{x}_{1:T}, \Theta)$, which is intractable. We employ a Gaussian distribution as the variational approximate distribution. Denoting $\bar{\mathbf{z}} = \mathrm{vec}(\mathbf{z}_{1:T})$ and $\bar{\mathbf{x}} = \mathrm{vec}(\mathbf{x}_{1:T})$, we approximate $p(\bar{\mathbf{z}} | \bar{\mathbf{x}})$ with $q_\phi(\bar{\mathbf{z}}) = \mathcal{N}\big(\mu_\phi(\bar{\mathbf{x}}), \sigma_\phi^2(\bar{\mathbf{x}})\big)$, whose mean and variance are the outputs of a highly nonlinear function of observation $\bar{\mathbf{x}}$, and ϕ encodes the function parameters. We identify the optimal $\bar{\mathbf{z}}, \Theta$ and ϕ by maximizing a variational Bayesian lower bound (also called "$\mathbf{ELBO}''$) as

$$\mathcal{L}(\bar{\mathbf{z}}, \Theta, \phi) = \mathbb{E}_{q_\phi(\bar{\mathbf{z}})}\big[\log p_\Theta(\bar{\mathbf{z}}, \bar{\mathbf{x}})\big] - \mathbb{E}_{q_\phi(\bar{\mathbf{z}})}\big[\log q_\phi(\bar{\mathbf{z}})\big]. \tag{10}$$

The first term in eq. (10) represents an energy, encouraging $q_\phi(\bar{\mathbf{z}})$ to focus on the probability mass, $p_\Theta(\bar{\mathbf{z}}, \bar{\mathbf{x}})$. The second term (including the minus sign) represents the entropy of $q_\phi(\bar{\mathbf{z}})$, encouraging it to spread the probability mass thus avoiding concentrating on one point estimate. The entropy term in eq. (10) has a closed-form expression:

$$\mathbb{E}_{q_\phi(\bar{\mathbf{z}})}\big[\log q_\phi(\bar{\mathbf{z}})\big] = -\frac{LT}{2}\big(1 + \log(2\pi)\big) - \frac{1}{2}\log|\Sigma|. \tag{11}$$

The gradients of eq. (10) with respect to ϕ, Θ can be evaluated by sampling directly from $q_\phi(\bar{\mathbf{z}})$, for example, using Monte Carlo integration to obtain noisy estimates of both the ELBO and its gradient [22, 23]. Score function estimator achieves it by leveraging a property of logarithms to write the gradient as

$$\nabla\mathcal{L}(\Theta, \phi) = \frac{1}{S}\sum_{s=1}^{S}\Big[\nabla \log q_\phi(\bar{\mathbf{z}}_s)\big(\log p_\Theta(\bar{\mathbf{z}}_s, \bar{\mathbf{x}}) - \log q_\phi(\bar{\mathbf{z}}_s)\big)\Big], \tag{12}$$

which first draws S samples $\{\bar{\mathbf{z}}_s\}_1^S$ from $q_\phi(\bar{\mathbf{z}})$, and then evaluates the empirical expectation using $\{\bar{\mathbf{z}}_s\}_1^S$. In general, the approximate gradient using score function estimator exhibits high variance [22], and practically we compute the integral with the "reparameterization trick"

Inference Network	Vanilla MF	VAE	r-LSTM	l-LSTM	bi-LSTM
Variational Approximation	$q(\mathbf{z}_t)$	$q(\mathbf{z}_t\|\mathbf{x}_t)$	$q(\mathbf{z}_t\|\mathbf{x}_{t:T})$	$q(\mathbf{z}_t\|\mathbf{x}_{1:t})$	$q(\mathbf{z}_t\|\mathbf{x}_{1:T})$

Table 2: Inference networks applied in variational approximation

proposed by [24]. We can parameterize the multivariate normal $\bar{\mathbf{z}} \sim q(\bar{\mathbf{z}}|\bar{\mathbf{x}})$ as

$$\bar{\mathbf{z}} = \mu_\phi(\bar{\mathbf{x}}) + R_\phi(\bar{\mathbf{x}})\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{13}$$

therefore $\mathbf{z}$ is distributed as a multivariate normal with mean $\mu_\phi(\bar{\mathbf{x}})$ and covariance $R_\phi(\bar{\mathbf{x}})R_\phi(\bar{\mathbf{x}})^\top$. We finally separate the gradient estimation as

$$\begin{aligned}\nabla_\Theta \mathcal{L}(\Theta, \phi) &= \mathbb{E}_{q_\phi(\bar{\mathbf{z}})}\left[\nabla_\Theta \log p_\Theta(\bar{\mathbf{z}}, \bar{\mathbf{x}})\right], \\ \nabla_\phi \mathcal{L}(\Theta, \phi) &= \mathbb{E}_\epsilon\Big[\nabla_\phi \log p_\Theta\big(\mu_\phi(\bar{\mathbf{x}}) + R_\phi(\bar{\mathbf{x}})\epsilon, \bar{\mathbf{x}}\big)\Big] \\ &\quad + \nabla_\phi H_\phi, \end{aligned} \tag{14}$$

where $H_\phi = \mathbb{E}_{q_\phi(\bar{\mathbf{z}})}\left[\log q_\phi(\bar{\mathbf{z}})\right]$ is the entropy of the variational distribution. Now both gradients can be approximated with Monte-Carlo estimates.

On the choice of the optimal variational distribution: In eq. (10), we consider the approximated posterior $q_\phi(\bar{\mathbf{z}})$ as a Gaussian, $\mathcal{N}\big(\mu_\phi(\bar{\mathbf{x}}), \sigma^2_\phi(\bar{\mathbf{x}})\big)$, whose mean and variance are the outputs of a highly nonlinear function of observation $\bar{\mathbf{x}}$. Here, we consider five structured q distributions by encoding $\bar{\mathbf{x}}$ in different sequential patterns shown in Table 2: (1) vanilla mean field (MF); (2) variational autoencoder (VAE); (3) LSTM conditioned on past observations (l-LSTM); (4) LSTM conditioned on future observations (r-LSTM) and (5) bi-directional LSTM (bi-LSTM) conditioned on both past and future observations.

For l-LSTM and r-LSTM, "l" or "r" is an abbreviation of "left" or "right", which considers past or future information. We parametrize mean μ_t and variance σ^2_t for the variational approximated posterior at time step t as a function of the hidden state h_t, e.g., for l-LSTM, $h_{t,l} = \mathbf{LSTM}(\mathbf{x}_{1:t})$. We illustrate the l/r/bi-LSTM structure of inference networks in Fig 2. Inference network maps observation $\bar{\mathbf{x}}$ to varational parameters μ_t, σ^2_t of approximate posterior $p(\bar{\mathbf{z}}|\bar{\mathbf{x}})$ via LSTM-based structures. The inference network maps observations to the mean and covariance functions of approximated Gaussian latent states. The parameterization (r-LSTM) can be written as

$$\begin{aligned}\mu_{t,r} &= W_{\mu_r} h_{t,r} + b_{\mu_r}, \\ \sigma^2_{t,r} &= \text{softplus}(W_{\sigma^2_r} h_{t,r} + b_{\sigma^2_r}). \end{aligned} \tag{15}$$

Similar with the l-LSTM. Here W and b are weights and bias mapping h_t to variational parameters. In bi-LSTM, we use a weighted mean and variance to parameterize the variational posterior as

$$\begin{aligned}\mu_{t,bi} &= (\mu_{t,r}\sigma^2_{t,l} + \mu_{t,l}\sigma^2_{t,r})/(\sigma^2_{t,l} + \sigma^2_{t,r}), \\ \sigma^2_{t,bi} &= (\sigma^2_{t,l}\sigma^2_{t,r})/(\sigma^2_{t,l} + \sigma^2_{t,r}). \end{aligned} \tag{16}$$

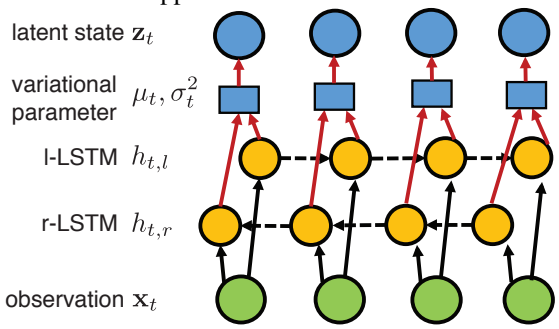

Figure 2: Inference network for l-LSTM, r-LSTM and bi-LSTM. Briefly, bi-LSTM is the joint effect of l/r-LSTM. The blue circle denotes latent states z_t, the blue square shows the variational parameters $(\mu_t,\ \sigma_t^2)$, the yellow circle denotes hidden states of two LSTMs $h_{t,l}$ and $h_{t,r}$, and the green circle represents observation data x_t.

All operations should be performed element-wisely on the corresponding vectors. The Gaussian approximated posterior $q(\mathbf{z}_t|\mathbf{x}_{1:T}) \sim \mathcal{N}(\mu_{t,bi}, \sigma^2_{t,bi})$ thus summarizes both the past and future information from observations.

Algorithm 1 summarizes the inference method for the Gaussian observation model based on variational inference.

Algorithm 1 Inference of GP-RNN-Gaussian

Input: dataset $\mathbf{x}_{1:T}$
Output: latent process $\mathbf{z}_{1:T}$, model parameters $\Theta = \{\rho, \sigma, \theta, \psi\}$, variational parameter ϕ
repeat
  Evaluate $\mu_\phi(\mathbf{x}_{1:T})$ and $\sigma^2_\phi(\mathbf{x}_{1:T})$ based on eq. (16)
  Sample $\mathbf{z}_{1:T} \sim \mathcal{N}\big(\mu_\phi(\mathbf{x}_{1:T}), \sigma^2_\phi(\mathbf{x}_{1:T})\big)$
  Evaluate $\mathcal{L}(\phi, \Theta)$ based on eq. (10)
  Compute $\nabla_\Theta \mathcal{L}$ and $\nabla_\phi \mathcal{L}$ based on eq. (14)
  Update Θ and ϕ using ADAM
until convergence

Poisson response: When the observation model is Poisson, the integration over the mapping function f in eq. (9) is now intractable due to the non-conjugacy between its GP prior and the Poisson data distribution. The interplay between $\mathbf{z}$ and f involves a highly-nonlinear transformation, which makes inference difficult. Inducing points [25] and decoupled Laplace approximation [5] have been recently introduced to release this dependence and make inference tractable. In this paper, we adapt a straightforward maximum a posteriori (MAP) estimation

for training both $\mathbf{F}$ and $\bar{\mathbf{z}}$, as

$$\begin{aligned}\mathbf{F}, \bar{\mathbf{z}} &= \text{argmax}_{\mathbf{F},\bar{\mathbf{z}}}\, p(\bar{\mathbf{x}}, \mathbf{F}, \bar{\mathbf{z}}) \\ &= \text{argmax}_{\mathbf{F},\bar{\mathbf{z}}}\, p(\bar{\mathbf{x}}|\mathbf{F}) p(\mathbf{F}|\bar{\mathbf{z}}, \rho, \sigma) p(\bar{\mathbf{z}}|\theta, \psi),\end{aligned} \tag{17}$$

where the joint distribution $p(\bar{\mathbf{x}}, \mathbf{F}, \bar{\mathbf{z}})$ of latent variables $\bar{\mathbf{z}}, \mathbf{F}$ and observations $\bar{\mathbf{x}}$ of the RHS for eq. (18) is

$$\begin{aligned}p(\bar{\mathbf{x}}, \mathbf{F}, \bar{\mathbf{z}}) &= p(\bar{\mathbf{x}}|\mathbf{F}) p(\mathbf{F}|\bar{\mathbf{z}}, \rho, \sigma) p(\bar{\mathbf{z}}|\theta, \psi) \\ &= \prod_{i=1}^{N}\prod_{t=1}^{T} \underbrace{p(x_{i,t}|f_{i,t})}_{\textbf{Poisson}} \prod_{i=1}^{N} \underbrace{p(\mathbf{f}_i|\bar{\mathbf{z}}, \rho, \sigma)}_{\mathcal{GP}} \prod_{t=1}^{T} \underbrace{p_\theta(\mathbf{z}_t|\mathbf{z}_{<t})}_{\textbf{RNN}}.\end{aligned} \tag{18}$$

Eq. (18) is a joint probability with three main components: (1) Poisson spiking (observation model); (2) Gaussian process ($\mathcal{GP}$, nonlinear embedding); and (3) recurrent neural networks (**RNN**, dynamical model). During the training procedure, we adapt composing inference [26], fixing $\mathbf{F}$ or $\bar{\mathbf{z}}$ while optimizing the other in a coordinate ascent manner. More details and the pseudo-algorithm for inference of GP-RNN-Poisson can be found in the supplementary.

4 EXPERIMENTS

To demonstrate the superiority of GP-RNN in latent dynamics recovery, we compare it against other state-of-the-art methods on both extensive simulated data and a real visual cortex neural dataset.

4.1 Recovery of Lorenz Dynamics

First we recover the well-known Lorenz dynamics in a nonlinear system. The Lorenz system describes a two dimensional flow of fluids with $z_{1,2,3}$ as latent states:

$$\frac{dz_1}{dt} = \sigma(z_2 - z_1), \frac{dz_2}{dt} = z_1(\rho - z_3) - z_2, \frac{dz_3}{dt} = z_1 y - \beta z_3. \tag{19}$$

This system has chaotic solutions (for certain parameter values) that revolve around the so-called Lorenz attractor. Lorenz uses the values $\sigma = 10$, $\beta = 8/3$ and $\rho = 28$, exhibiting a chaotic behavior, which generates a nonlinear, non-periodic, and three-dimensional complex system. It has been utilized for testing latent structure discovery in recent works [27, 28, 6].

We simulated a three-dimensional latent dynamic using Lorenz system as in eq. (19), and then apply three different mapping functions for simulations: $\mathbf{x}_t = \mathbf{w}^\top \mathbf{z}_t + \Phi + \eta$; $\mathbf{x}_t = \tanh(\mathbf{w}^\top \mathbf{z}_t + \Phi) + \eta$; and $\mathbf{x}_t = \sin(\mathbf{w}^\top \mathbf{z}_t + \Phi) + \eta$. Note that the oscillatory response of sine wave is well-known as the properties of grid cells [4]). Thus, we generate simulated data with nonlinear dynamics and linear/nonlinear mapping functions. Gaussian response is the Gaussian noise corrupted version of $\mathbf{x}_t$; Poisson spike trains are generated from a Poisson distribution with $\exp(\mathbf{x}_t)$ as the spike rate.

In our simulation, the latent dimension is 3 and the number of neurons is 50, thus $\mathbf{z}_t \in \mathbb{R}^3$ and $\mathbf{w} \in \mathbb{R}^{3\times 50}$. We randomly generate weights $\mathbf{w}$ and bias Φ uniformly from region [0, 1.0], and the noise η is drawn from $\mathcal{N}(0, I)$. We test the ability of each method to infer the latent dynamics of the Lorenz system (i.e., the values of the three dynamic variables) from Gaussian and Poisson responses, respectively. Models are compared in three aspects: inference network, dynamical model and mapping function.

Analysis of inference network and dynamical model: Table 3 and 4 show performance of variational approximation techniques applied to both P-GPLVM with AR1 kernel (AR1-GPLVM) and GP-RNN models on Gaussian and Poisson response data respectively. P-GPLVM with AR1 kernel is mathematically equal to the GPLVM model with LDS when the linear mapping matrix in LDS is full-rank. Therefore we are essentially comparing between LDS and RNN for dynamic modeling. In general, GP-RNN outperforms AR1-GPLVM via capturing complex dynamics of nonlinear systems with powerful RNN representations.

bi-LSTM inference networks render best results due to its consideration of both past and future information. Meanwhile, l-LSTM demonstrates the importance of past dependence with better results than r-LSTM. Overall, LSTM-style inference networks have more promising results than models considering current observations only (e.g., MF and VAE).

Moreover, the inference network of VAE is not a much more expressive variational model for approximating posterior distribution compared with vanilla mean field methods. With only current time points, both of them have similar inference power (as shown in Table 3 and 4 columns of "MF" and "VAE"). VAE only has global parameters of the neural network for mapping data points to posteriors, while vanilla MF has local parameters for each data point. VAE can be scaled to large-scale datasets, but the performance is upper-bounded by vanilla MF [26].

Analysis of mapping function:

Table 5 shows the comparison between a neural network and a Gaussian process as the nonlinear mapping functions. The dynamical model is RNN, and the true mapping functions include linear, tanh, and sine functions. The number of data points for training (N) are 50, 100, 200 and 500. The subsequent 50 time points following the training time points are used for testing the accuracy of reconstructions of latent trajectories. In Table 5, We can tell that a Gaussian process provides a superior mapping function for smaller datasets for training (columns of "GP"

Gaussian	AR1-GPLVM					GP-RNN				
	MF	VAE	r-LSTM	l-LSTM	bi-LSTM	MF	VAE	r-LSTM	l-LSTM	bi-LSTM
linear	4.12	4.10	4.01	3.27	**1.64**	2.17	2.17	1.98	1.54	**0.96**
tanh	3.20	3.22	3.01	2.46	**1.17**	2.01	2.01	1.83	1.41	**0.78**
sine	3.12	3.12	2.74	2.33	**1.02**	1.81	1.78	1.34	1.12	**0.56**

Table 3: Inference network and dynamical model analysis. Root mean square error (**RMSE**, 10^{-2}) of latent trajectories reconstructed from various simulated models are presented. We compare two latent dynamical models: first-order autoregressive (AR1) and recurrent neural network (e.g., LSTM), three mapping functions: linear, tanh and sine, and five variational approximations listed in Table 2. The observations are Gaussian responses with 50 observational dimensions and 200 time points. Underlined and bold fonts indicate best performance. Results with standard errors (ste) can be found in the supplementary.

Poisson	AR1-GPLVM					GP-RNN				
	MF	VAE	r-LSTM	l-LSTM	bi-LSTM	MF	VAE	r-LSTM	l-LSTM	bi-LSTM
linear	6.34	6.34	6.02	5.71	**3.67**	6.01	6.01	5.94	5.71	**3.10**
tanh	3.22	3.21	3.01	2.84	**1.57**	3.09	3.11	2.98	2.54	**1.21**
sine	2.80	2.79	2.77	2.51	**1.49**	2.67	2.67	2.43	2.33	**1.14**

Table 4: Root mean square error (**RMSE**, 10^{-2}) of latent trajectories reconstructed from Poisson responses in test datasets. Underlined and bold fonts highlight best performance. Results with standard errors (ste) can be found in the supplementary.

# Data	linear		tanh		sine	
	GP	NN	GP	NN	GP	NN
N = 50	**2.51**	3.88	**1.45**	2.75	**1.97**	3.43
N = 100	**1.27**	1.65	**1.15**	1.45	**1.03**	1.31
N = 200	**0.96**	1.29	**0.78**	1.22	**0.56**	0.70
N = 500	**0.34**	0.35	**0.26**	**0.26**	**0.12**	**0.12**

Table 5: Mapping function analysis. **RMSE** (10^{-2}) of latent trajectory reconstruction using Gaussian process (GP-RNN) and neural network (NN-RNN) mapping functions are shown. Both of them are combined with an RNN dynamical model component. We simulate 50 trials and present averaged **RMSE** results across all trials. Linear, tanh and sine mapping functions are used to generate the data. "N" indicates the number of data points for training in each trial, and **RMSE** is the result of subsequent 50 time points for testing. Results with standard errors (ste) can be found in the supplementary.

and "NN"). When we have more time points, the prediction performance of a neural network mapping is comparable with a Gaussian process (rows of $N = 200$ and 500). Bigger datasets can help to learn complex Lorenz dynamics, and meanwhile, prevent the overfitting problem in neural network models. Smaller datasets may affect latent dynamics recovery but a Gaussian process mapping enhances nonlinear embedding recovery via keeping the local constraints.

Comparison with state-of-the-art methods:

Consistent with results reported in state-of-the-art methods, we compare R^2 values for latent trajectory reconstruction of our GP-RNN method against others as shown in Table 6. The inference network of our model is bi-LSTM since the simulated results shown above demonstrate its stronger power in model fitting. Note that we use the Poisson model and compare it with recently developed models for analyzing spike trains. For each dimension of Lorenz dynamics, GP-RNN significantly outperforms baseline methods, e.g., 10.8% (z_1), 11.2% (z_2) and 0.5% (z_3) increment of R^2 values compared with the second best model P-GPLVM. We have also found several excellent works combining RNN structures with Gaussian process for either modeling or inference [17, 19, 20, 21], but note that they are not in the research line of exploring latent intrinsic structures of high-dimensional real or count-valued data as stated in our work. The methods we compared in our paper (e.g., PLDS [2], GCLDS [3], PfLDS [4], P-GPFA [11], and P-GPLVM [5]) are to our knowledge recently proposed methods analyzing the same problems and can be more worthwhile being compared.

4.2 Application to Macaque V1 Neural Data

We apply GP-RNN to the neurons recorded from the primary visual cortex of a macaque [29]. Data was obtained when the monkey was watching sinusoidal grating driftings with 72 orientations ($0°$, $5°$, $10°$, $\cdots$, $355°$), and had 50 repeated trials for each orientation. Following [4], we consider 63 well-behaved neurons based on their tuning curves, and bin 900 ms spiking activity with window size

Dimension	PLDS	GCLDS	PfLDS	P-GPFA	P-GPLVM	GP-RNN
z_1	0.641	0.435	0.698	0.733	0.784	**0.869**
z_2	0.547	0.364	0.659	0.720	0.785	**0.873**
z_3	0.903	0.755	0.797	0.960	0.966	**0.971**

Table 6: R^2 (best possible score is 1.0) values of our method and other state-of-the-art methods for the prediction of Lorenz-based spike trains. The included methods are Poisson linear dynamical system (PLDS [2]), generalized count linear dynamical system (GCLDS [3]), Poisson feed-forward neural network linear dynamical system (PfLDS [4]), and Poisson-Gaussian process latent variable model(P-GPLVM [5]). GP-RNN recovers more variance of the latent Lorenz dynamics, as measured by R^2 between the linearly transformed estimation of each model and the true Lorenz dynamics. Results with standard errors (ste) can be found in the supplementary.

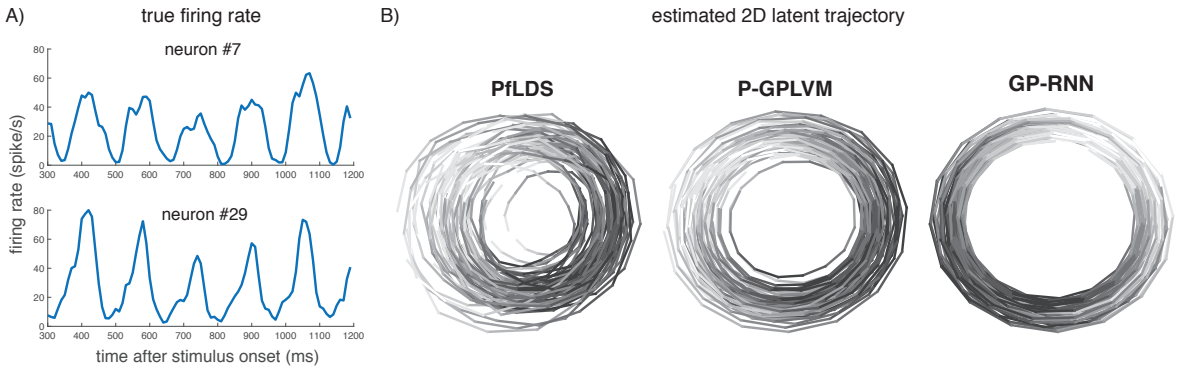

Figure 3: A) True firing rates for 2 example neurons for orientation 0° averaged across 30 trials. We can tell there exists clear periodicity in the firing rate time series given the sinusoidal grating stimulus. B) 2-dimensional latent trajectories of 10 out of 30 trials using PfLDS, P-GPLVM and GP-RNN. Color denotes the phase of the grating stimulus implied in (A). Each circle corresponds to a period of latent dynamics $\mathbf{z}_{1:T}$ (T = 90) inferred by the models. Each trial is estimated from 63-neuronal spike trains. The latent embedding is smoother and more structured when applying GP-RNN, which is interpretable since the stimulus is sinusoidal for each orientation across time. We can tell that the phase of latent dynamics inferred by GP-RNN is better locked to the phase of the stimulus.

$\Delta t = 10$ ms, resulting in 90 time points for each trial.

We take orientation 0° as an example for visualizing 2-dimensional (2D) latent trajectory estimation. The other orientations exhibit similar patterns. The true firing rates of two example neurons are presented in Fig. 3 (A), which exhibit clear periodic patterns locked to the phase of the sinusoidal grating stimulus. In order to get latent dynamics estimation, we fit our model with randomly selected 30 repeated trials, which are used to learn RNN dynamics parameters and GP hyperparameters shared across all trials, and trial-dependent latent dynamics. We also apply PfLDS and P-GPLVM to the same data. For better visualization purpose, Fig. 3(B) shows the results of 10 best trials, which are selected with 10 smallest variances from the mean trajectory within each model. PfLDS has a worse performance compared with the other two methods. Different from the result shown in [4], we report the result of 30 trials for training instead of 120. Benefiting from the non-parametric Bayes (Gaussian process), in such a small-data scenario, GP-RNN extracts much more clear, compact, and structured latent trajectories, which well capture oscillatory patterns in neural responses for the grating stimulus. Meanwhile, the proposed model is able to convey interpretable sinusoidal stimulus patterns in 2D rings without including the external stimulus as the model variable. Therefore, GP-RNN with nonlinear dynamics and nonlinear embedding function can help extract latent patterns more efficiently. Although P-GPLVM also achieves promising results compared with PfLDS (still worse than our GP-RNN), P-GPLVM needs much more effort than GP-RNN to fine-tune the optimization hyperparameters.

We next show the quantitative prediction performance of multiple methods. The evaluation procedure is well known as "co-smoothing" [27, 5], which is a standard leave-one-neuron-out test. We select all the trials with 0°, 90°, 180°, and 270° orientations of sinusoidal grating drifting. We split all the trials into training sets (40 trials) and test sets (10 trials). The model-specific parameters, e.g., RNN dynamics and GP mapping function for

GP-RNN, are estimated using training sets (all neurons). Then we fix the estimated model parameters and leave one neuron in test trials out and infer latent trajectories based on the remaining neurons. The left-out neuron spiking activity is then predicted given inferred latents of test trials and estimated parameters from training trials. Consistent with the results reported in the previous literature, the prediction is quantified by R^2. It shows the prediction performance of the firing rates compared with empirical firing rates of the left-out neurons. We iterate over all neurons as left-out ones and average the prediction R^2 values for each model shown in Table. 7. In this neural dataset, each recently proposed method can only increase the R^2 value by a small amount, which is still non-trivial to achieve. GP-RNN has already doubled the increment from PfLDS (13% increase of R^2 value) to P-GPLVM (7% increase). P-GPLVM and PfLDS have comparable

Dim	PLDS	P-GPFA	LFADS	PfLDS	P-GPLVM	GP-RNN
2	0.68	0.69	0.73	0.73	0.74	**0.77**
4	0.69	0.72	0.74	0.73	0.75	**0.78**
6	0.72	0.73	0.74	0.74	0.77	**0.80**
8	0.74	0.74	0.75	0.75	0.77	**0.80**
10	0.75	0.74	0.77	0.76	0.77	**0.81**

Table 7: Predictive R^2 on neural spiking activity of test dataset. The column "Dim" indicates the dimension of latent process **z**. GP-RNN has consistently the best performance when increasing predefined latent dimensions.

results and we think they benefit from nonlinear mapping functions, i.e., feed-forward neural network and Gaussian process. PLDS and P-GPFA use linear mapping but cannot capture nonlinear embeddings, and require more latent dimensionality to achieve similar results as P-GPLVM and PfLDS. Our GP-RNN with RNN dynamics and GP mapping provides the most competitive prediction accuracy, due to its nonlinear dynamical model encoding time dependence and complex nonlinear embedding function with uncertainty propagation.

4.3 Implementation Notes

We have encountered the risk of over-parameterization during our experiments. When the algorithm breaks down, increasing the number of hidden nodes of RNN structures cannot improve the results much. We successfully avoid it via (1) using cross-validation to choose the number of hidden states (the risk happened with more than 30 hidden nodes in this experimental dataset); (2) adopting Dropout(0.3)/L2 regularization for RNN gates. Too many hidden states of RNN dynamics will lead to learning both hidden states ν and cell states c failure, also too few hidden states report much lower prediction performance (we fix 30 hidden nodes ultimately in our experiments); (3) applying orthogonal initialization for RNN gates and clipping gradients tricks during training; and (4) instead of marginalizing out the latent function f in the Poisson model, adopting the composing inference strategy and using GPFA to initialize f. The experiments are benefited from the probabilistic modeling library "Edward" [26].

With respect to the stable learning process, it is robust when applying orthogonal initialization for RNN gates, Xavier Initialization for parameters of fully connected layers (mapping hidden states ν to latent states z), and clipping gradients tricks during training. This combination is a relatively effective way of eliminating exploding and vanishing gradients, and provides a robust learning process.

Concerning sample perturbations, in the simulation, we randomly (both Poisson and Gaussian noise) generated the observations and parameters of mapping functions (Gaussian noise) for 10 times; and with real neural data, we shuffled the training/testing datasets for 10 times. The learning was based on these sample perturbations (trial variants) and the above-mentioned initialization strategies. The analysis of the sample perturbations are listed in the supplementary materials with standard errors.

5 CONCLUSION

To discover the insightful latent structure from neural data, we propose an unsupervised Gaussian process recurrent neural network (GP-RNN), utilizing the representation power of recurrent neural networks and the flexible nonlinear mapping function with Gaussian process. We show that GP-RNN is superior at recovering more structured latent trajectories as well as having better quantitative performance compared with other state-of-the-art methods. Besides the visual cortex dataset tested in the paper, the proposed model can also be potentially applied to analyzing the neural dynamics of primary motor cortex, prefrontal cortex (PFC) or posterior parietal cortex (PPC) which plays a significant role in cognition (evidence integration, short term memory, spatial reasoning, etc.). The model can also be applied to other domains, e.g., finance, healthcare, for extracting low-dimensional, underlying latent states from complicated time series. Our codes and additional materials are available at `https://github.com/sheqi/GP-RNN_UAI2019`.

References

[1] M Yu Byron, John P Cunningham, Gopal Santhanam, Stephen I Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1881–1888, 2009.

[2] Jakob H Macke, Lars Buesing, John P Cunningham, M Yu Byron, Krishna V Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1350–1358, 2011.

[3] Yuanjun Gao, Lars Busing, Krishna V Shenoy, and John P Cunningham. High-dimensional neural spike train analysis with generalized count linear dynamical systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2044–2052, 2015.

[4] Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical neural population models through nonlinear embeddings. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 163–171, 2016.

[5] Anqi Wu, Nicholas G Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3499–3508, 2017.

[6] Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, page 1, 2018.

[7] Qi She, Yuan Gao, Kai Xu, and Rosa HM Chan. Reduced-rank linear dynamical systems. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[8] Rahul G Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *The Thirty-first AAAI Conference on Artificial Intelligence (AAAI)*, pages 2101–2109, 2017.

[9] Qi She and Rosa HM Chan. Stochastic dynamical systems based latent structure discovery in high-dimensional time series. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 886–890. IEEE, 2018.

[10] Neil D Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 329–336, 2004.

[11] Hooram Nam. Poisson extension of gaussian process factor analysis for modeling spiking neural populations. *Master's thesis, Department of Neural Computation and Behaviour, Max Planck Institute for Biological Cybernetics, Tübingen*, 2015.

[12] Guan Yu. Variance stabilizing transformations of poisson, binomial and negative binomial distributions. *Statistics & Probability Letters*, 79(14):1621–1629, 2009.

[13] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research (JMLR)*, 14(1):1303–1347, 2013.

[14] M Bishop Christopher. *Pattern recognition and machine learning*. Springer-Verlag New York, 2016.

[15] 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 (NeurIPS)*, pages 2980–2988, 2015.

[16] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

[17] Roger Frigola, Yutian Chen, and Carl Edward Rasmussen. Variational gaussian process state-space models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3680–3688, 2014.

[18] Trung V Nguyen, Edwin V Bonilla, et al. Collaborative multi-output gaussian processes. In *Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 643–652, 2014.

[19] César Lincoln C Mattos, Zhenwen Dai, Andreas Damianou, Jeremy Forth, Guilherme A Barreto, and Neil D Lawrence. Recurrent gaussian processes. *arXiv preprint arXiv:1511.06644*, 2015.

[20] Andreas Svensson, Arno Solin, Simo Särkkä, and Thomas Schön. Computationally efficient bayesian learning of gaussian process state space models. In *Artificial Intelligence and Statistics (AISTATS), 2016 International Conference on*, pages 213–221, 2016.

[21] Stefanos Eleftheriadis, Tom Nicholson, Marc Deisenroth, and James Hensman. Identification of gaussian process state space models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5309–5319, 2017.

[22] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics (AISTATS), 2014 International Conference on*, pages 814–822, 2014.

[23] Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*, 2015.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[25] Andreas C Damianou, Michalis K Titsias, and Neil D Lawrence. Variational inference for latent variables and uncertain inputs in gaussian processes. *The Journal of Machine Learning Research (JMLR)*, 17(1):1425–1486, 2016.

[26] Dustin Tran, Matthew D. Hoffman, Rif A. Saurous, Eugene Brevdo, Kevin Murphy, and David M. Blei. Deep probabilistic programming. In *International Conference on Learning Representations (ICLR)*, 2017.

[27] Yuan Zhao and Il Memming Park. Variational latent gaussian process for recovering single-trial dynamics from population spike trains. *Neural Computation*, 29(5):1293–1316, 2017.

[28] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial Intelligence and Statistics (AISTATS), 2017 International Conference on*, pages 914–922, 2017.

[29] Arnulf BA Graf, Adam Kohn, Mehrdad Jazayeri, and J Anthony Movshon. Decoding the activity of neuronal populations in macaque primary visual cortex. *Nature neuroscience*, 14(2):239, 2011.