Supplementary Materials
Neural Dynamics Discovery via Gaussian Process Recurrent Neural Networks, UAI 2019

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

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

1 Inference of GP-RNN with Poisson Response

Here we demonstrate more details with respect to the training procedure for the Poisson observation model. In details, $p(\mathbf{F}|\bar{\mathbf{x}}, \bar{\mathbf{z}})$ is first approximated using Laplace approximation with $q(\mathbf{F}|\bar{\mathbf{x}}, \bar{\mathbf{z}})$ over each $\mathbf{f}_i$ as $q(\mathbf{f}_i|\mathbf{x}_i, \bar{\mathbf{z}}) = \mathcal{N}(\hat{\mathbf{f}}_i, \Sigma^{-1})$ with mean and precision matrix computed as

$$\hat{\mathbf{f}}_i = \mathrm{argmax}_{\mathbf{f}_i}\, p(\mathbf{f}_i|\mathbf{x}_i, \bar{\mathbf{z}}), \tag{1}$$

$$\Sigma = -\nabla\nabla \log p(\mathbf{f}_i|\mathbf{x}_i, \bar{\mathbf{z}})|_{\mathbf{f}_i=\hat{\mathbf{f}}_i}, \tag{2}$$

where Σ is the Hessian of the negative log posterior. The optimization in eq. (1) can be realized by maximizing $p(\mathbf{x}_i, \mathbf{f}_i|\bar{\mathbf{z}}) = p(\mathbf{x}_i|\mathbf{f}_i)p(\mathbf{f}_i|\bar{\mathbf{z}})$ using Bayes' rule, and we denote $\Psi(\mathbf{f}_i) = \log p(\mathbf{x}_i, \mathbf{f}_i|\bar{\mathbf{z}})$, and can get

$$\Psi(\mathbf{f}_i) = \log p(\mathbf{x}_i|\mathbf{f}_i) - \frac{1}{2}\mathbf{f}_i^\top \mathbf{K}_z \mathbf{f}_i - \frac{1}{2}\log|\mathbf{K}_z|, \tag{3}$$

where the first and second derivative with respect to $\mathbf{f}_i$ is

$$\nabla\Psi(\mathbf{f}_i) = \nabla \log p(\mathbf{x}_i|\mathbf{f}_i) - \mathbf{K}_z^{-1}\mathbf{f}_i, \tag{4}$$

$$\nabla\nabla\Psi(\mathbf{f}_i) = \nabla\nabla \log p(\mathbf{x}_i|\mathbf{f}_i) - \mathbf{K}_z^{-1}. \tag{5}$$

Our goal is to obtain an Laplace approximation $q(\mathbf{x}_i|\bar{\mathbf{z}})$ for the marginal likelihood $p(\mathbf{x}_i|\bar{\mathbf{z}})$, here we have

$$p(\mathbf{x}_i|\bar{\mathbf{z}}) = \int p(\mathbf{x}_i, \mathbf{f}_i|\bar{\mathbf{z}}) d\mathbf{f}_i = \int \exp\big(\Psi(\mathbf{f}_i)\big) \mathrm{d}\mathbf{f}_i, \tag{6}$$

a Taylor expansion of $\Psi(\mathbf{f}_i)$ around $\hat{\mathbf{f}}_i$ is $\Psi(\mathbf{f}_i) \simeq \Psi(\hat{\mathbf{f}}_i) - \frac{1}{2}(\mathbf{f}_i - \hat{\mathbf{f}}_i)^\top\Sigma(\mathbf{f}_i - \hat{\mathbf{f}}_i)$, and thus an approximation $q(\mathbf{x}_i|\bar{\mathbf{z}})$ to the marginal likelihood can be obtained

$$q(\mathbf{x}_i|\bar{\mathbf{z}}) = \exp(\hat{\mathbf{f}}_i) \int \exp\big(-\frac{1}{2}(\mathbf{f}_i - \hat{\mathbf{f}}_i)^\top\Sigma(\mathbf{f}_i - \hat{\mathbf{f}}_i)\big)\mathrm{d}\mathbf{f}_i.$$

The Gaussian integral can be calculated analytically and we can obtain the approximated log marginal likelihood:

$$\log q(\mathbf{x}_i|\bar{\mathbf{z}}) = \log p(\mathbf{x}_i|\hat{\mathbf{f}}_i) - \frac{1}{2}\big(\hat{\mathbf{f}}_i^\top \mathbf{K}_z^{-1}\hat{\mathbf{f}}_i + \log|\mathrm{A}|\big), \tag{7}$$

where $\mathrm{A} = |\mathbf{K}_z||\mathbf{K}_z^{-1}\nabla\nabla \log p(\mathbf{x}_i|\hat{\mathbf{f}}_i)|$. Then it is straightforward to get MAP estimation of $\bar{\mathbf{z}}$ as

$$\bar{\mathbf{z}}_{\mathrm{MAP}} = \mathrm{argmax}_{\bar{\mathbf{z}}} \sum_{i=1}^{N} \log q(\mathbf{x}_i|\bar{\mathbf{z}})p(\bar{\mathbf{z}}). \tag{8}$$

We optimize eq. (1) and eq. (8) in a coordinate ascent manner iteratively. The results in the experiment section demonstrate that when having hybrid inference algorithms for $\mathbf{F}$ and $\bar{\mathbf{z}}$ update, we can obtain promising results.[1]

[1] The process is efficient, *e.g.* for 50 observational dimensions and 500 time points, recovery of 5-dimensional latent trajectories takes 5 mins until convergence. All experiments were run on a quad-core Intel i5 with 6GB RAM.

Algorithm 2 summarizes the inference method for the Poisson observation model based on MAP.

Algorithm 1 Inference of GP-RNN-Poisson

Input: dataset $\mathbf{x}_{1:T}$
Output: latent process $\mathbf{z}_{1:T}$, tuning curve $\mathbf{f}_{1:N}$, model parameters $\Theta = \{\rho, \sigma, \theta, \psi\}$
repeat
  for i = 1 : N **do**
    Compute the posterior mode $\hat{\mathbf{f}}_i$, and the precision matrix Σ via solving eq. (3), and get $q(\mathbf{f}_i|\mathbf{x}_i, \bar{\mathbf{z}}) = \mathcal{N}(\hat{\mathbf{f}}_i, \Sigma^{-1})$

    Compute the new approximated log marginal likelihood $\log q(\mathbf{x}_i|\bar{\mathbf{z}})$ as eq. (7).
  end for
  Solve $\bar{\mathbf{z}}_{\mathrm{MAP}} = \mathrm{argmax}_{\bar{\mathbf{z}}} \sum_{i=1}^{N} \log q(\mathbf{x}_i|\bar{\mathbf{z}})p(\bar{\mathbf{z}})$ (eq. (8))
  Compute Θ = $\mathrm{argmax}_{\Theta}\, p(\bar{\mathbf{x}}, \Theta; \bar{\mathbf{z}}_{\mathrm{MAP}}, \mathbf{f}_{1:N})$ using SGD
until convergence

2 Further Results of Latent Dynamics Recovery Compared to SOTAs

All the results below are the recovery of three-dimensional lorenz dynamics from gaussian or poisson noisy data (black line is the true dynamics, the red line is the estimated dynamics from high-dimensional neural responses). The results are compared with the SOTAs, and can be easily reproduced using our uploaded .ipynb files.

2.1 GP-RNN (ours) VS PGPLVM (NIPS2017)[1]

Please refer to Fig. 1 and Fig. 2 for comparisons of the two models with 500 data points.

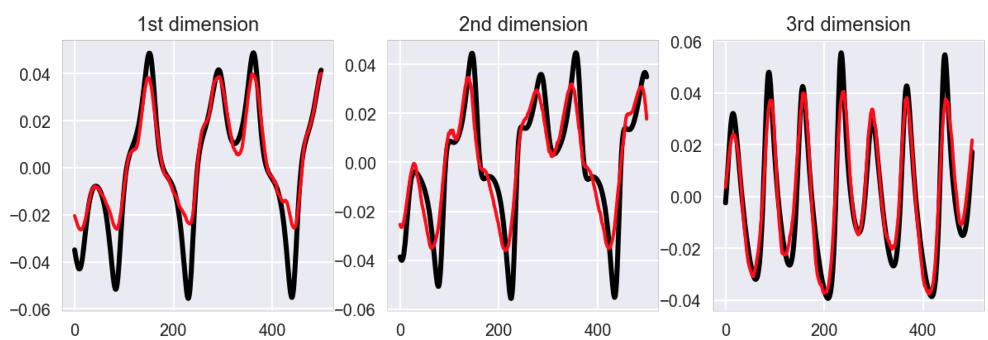

Figure 1: The model is "rnn dyn" (rnn dynamic model) with "gp map" (Gaussian process mapping function), the simulated process is "Lorenze dynamics" + "sin"(sinusoid mapping) + "poisson" response. The data points is 500, and the number of simulated neurons is 50.

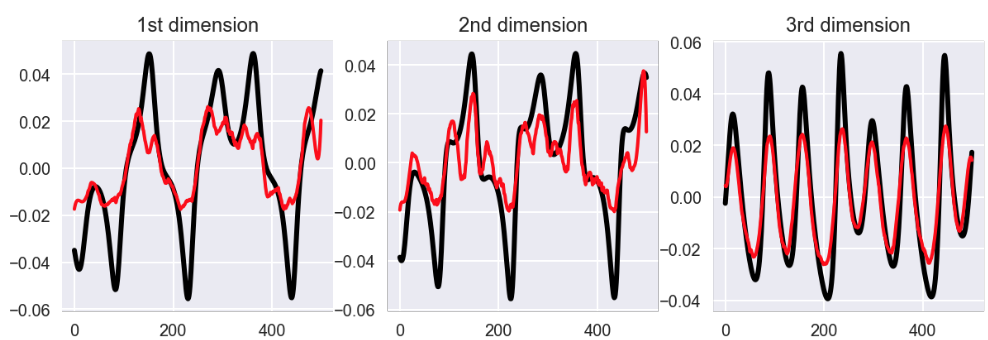

Figure 2: The model is "ar1 dyn" (1st order autoregressive dynamic model) with "gp map" (Gaussian process mapping function), the simulated process is "Lorenze dynamics" + "sin"(sinusoid mapping) + "poisson" response. The data points is 500, and the number of simulated neurons is 50.

2.2 GP-RNN (ours) VS PfLDS (NIPS2016) [2]

Please refer to Fig. 3 and Fig. 4 for comparisons of the two models with 200 data points.

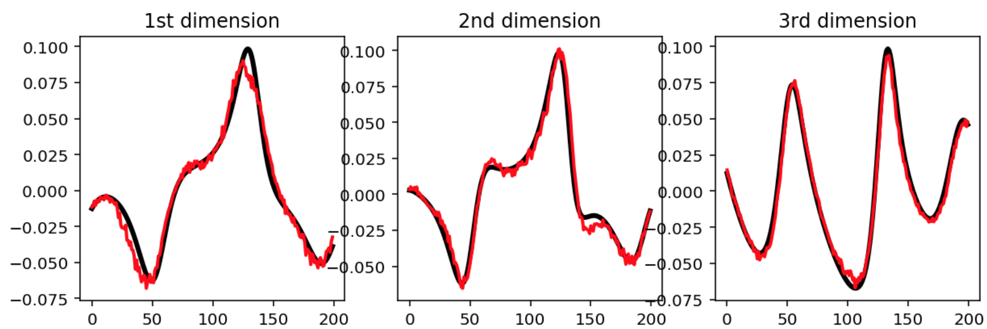

Figure 3: The model is “rnn dyn” (rnn dynamic model) with “gp map” (Gaussian process mapping function), the simulated process is “Lorenze dynamics” + “sin”(sinusoid mapping) + “poisson” response. The data points is 200, and the number of simulated neurons is 50.

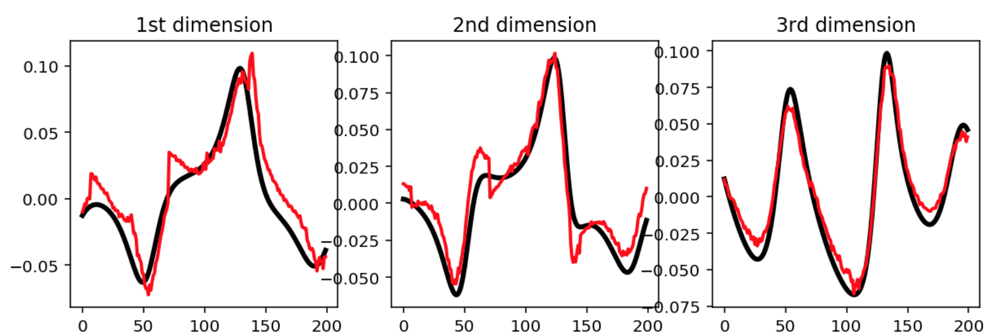

Figure 4: The model is “ar1 dyn” (1st order autoregressive dynamic model) with “nn map” (neural network mapping function), the simulated process is “Lorenze dynamics” + “sin”(sinusoid mapping) + “poisson” response. The data points is 200, and the number of simulated neurons is 50.

2.3 GP-RNN (ours) VS LFADS (Nature methods 2018) [3]

Please refer to Fig. 5 and Fig. 6 for comparisons of the two models with 200 data points. Note that for fair comparison, we implement LFADS ourselves so that we can control the variants effectively.

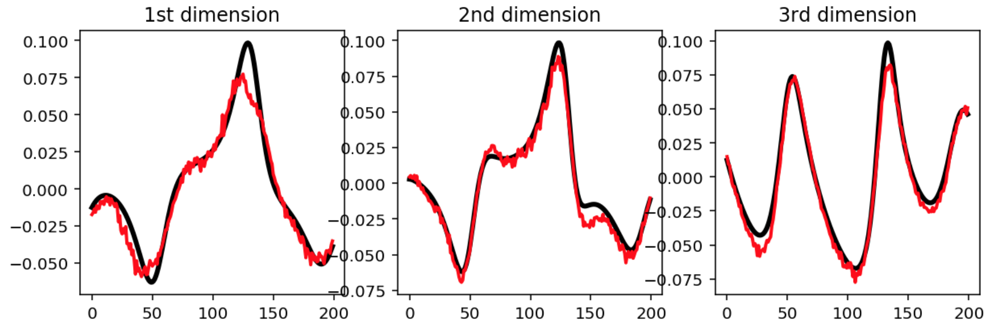

Figure 5: The model is “rnn dyn” (rnn dynamic model) with “gp map” (Gaussian process mapping function), the simulated process is “Lorenze dynamics” + “tanh”(sinusoid mapping) + “poisson” response. The data points is 200, and the number of simulated neurons is 50.

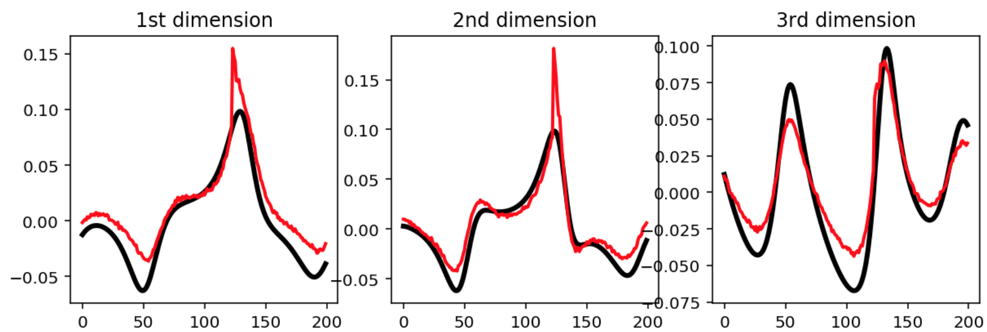

Figure 6: The model is “rnn dyn” (rnn dynamic model) with “nn map” (neural network mapping function), the simulated process is “Lorenze dynamics” + “tanh”(sinusoid mapping) + “poisson” response. The data points is 200, and the number of simulated neurons is 50.

3 Further Results w.r.t. Sample Perturbation

The tables in the main paper are shown averaged RMSE without standard errors due to page limit, please find more details as shown below.

Gaussian	AR1-GPLVM				
	MF	VAE	r-LSTM	l-LSTM	bi-LSTM
linear	4.12 ± 0.16	4.10 ± 0.16	4.01 ± 0.16	3.27 ± 0.13	$\underline{\mathbf{1.64 \pm 0.06}}$
tanh	3.20 ± 0.11	3.22 ± 0.13	3.01 ± 0.13	2.46 ± 0.13	$\underline{\mathbf{1.17 \pm 0.05}}$
sine	3.12 ± 0.13	3.12 ± 0.13	2.74 ± 0.12	2.33 ± 0.12	$\underline{\mathbf{1.02 \pm 0.04}}$

Table 1: (More details for original paper Table 3) Inference network and dynamical model analysis. Root mean square error (**RMSE**, 10^{-2}) and standard error (**ste**, 10^{-2}) of latent trajectories reconstructed from various simulated models are presented. First-order autoregressive (AR1) is shown with three mapping functions: linear, tanh and sine, and five variational approximations. The observations are Gaussian responses with 50 observational dimensions and 200 time points. Underlined and bold fonts indicate best performance.

Gaussian	GP-RNN				
	MF	VAE	r-LSTM	l-LSTM	bi-LSTM
linear	2.17 ± 0.08	2.17 ± 0.08	1.98 ± 0.08	1.54 ± 0.06	**0.96± 0.06**
tanh	2.01 ± 0.08	2.01 ± 0.08	1.83 ± 0.07	1.41 ± 0.06	**0.78± 0.05**
sine	1.81 ± 0.08	1.78 ± 0.08	1.34 ± 0.06	1.12 ± 0.06	**0.56± 0.03**

Table 2: (More details for original paper Table 3) Inference network and dynamical model analysis. Root mean square error (**RMSE**, 10^{-2}) and standard error (**ste**, 10^{-2}) of latent trajectories reconstructed from various simulated models are presented. Recurrent neural network (e.g., LSTM) is shown with three mapping functions: linear, tanh and sine, and five variational approximations. The observations are Gaussian responses with 50 observational dimensions and 200 time points. Underlined and bold fonts indicate best performance.

Poisson	AR1-GPLVM				
	MF	VAE	r-LSTM	l-LSTM	bi-LSTM
linear	6.34 ± 0.25	6.34 ± 0.25	6.02 ± 0.22	5.71 ± 0.20	**3.67 ± 0.15**
tanh	3.22 ± 0.13	3.21 ± 0.13	3.01 ± 0.13	2.84 ± 0.12	**1.57 ± 0.07**
sine	2.80 ± 0.12	2.79 ± 0.12	2.77 ± 0.12	2.51 ± 0.11	**1.49 ± 0.06**

Table 3: (More details for original paper Table 4) Root mean square error (**RMSE**, 10^{-2}) and standard error (**ste**, 10^{-2}) of latent trajectories reconstructed from Poisson responses in test datasets. Underlined and bold fonts highlight best performance.

Poisson	GP-RNN				
	MF	VAE	r-LSTM	l-LSTM	bi-LSTM
linear	6.01±0.20	6.01±0.20	5.94±0.20	5.71±0.21	**3.10±0.13**
tanh	3.09±0.15	3.11±0.15	2.98±0.13	2.54±0.13	**1.21±0.04**
sine	2.67±0.13	2.67±0.13	2.43±0.11	2.33±0.10	**1.14±0.04**

Table 4: (More details for original paper Table 4) Root mean square error (**RMSE**, 10^{-2}) and standard error (**ste**, 10^{-2}) of latent trajectories reconstructed from Poisson responses in test datasets. Underlined and bold fonts highlight best performance.

# Data points	linear		tanh		sine	
	GP	NN	GP	NN	GP	NN
N = 50	**2.51±0.11**	3.88±0.24	**1.45±0.06**	2.75 ± 0.20	**1.97±0.10**	3.43 ± 0.22
N = 100	**1.27±0.06**	1.65±0.15	**1.15±0.04**	1.45± 0.14	**1.03±0.04**	1.31 ± 1.14
N = 200	**0.96±0.03**	1.29± 0.07	**0.78±0.03**	1.22± 0.06	**0.56±0.03**	0.70 ± 0.05
N = 500	**0.34±0.02**	0.35±0.02	**0.26±0.01**	**0.26±0.01**	**0.12±0.01**	**0.12±0.01**

Table 5: (More details for original paper Table 5) Mapping function analysis. **RMSE** (10^{-2}) and standard error (**ste**, 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.

Dimension	PLDS	GCLDS	PfLDS	P-GPFA	P-GPLVM	GP-RNN
z_1	0.641 ± 0.10	0.435 ± 0.14	0.698 ± 0.07	0.733 ± 0.05	0.784 ± 0.06	**0.869 ± 0.02**
z_2	0.547 ± 0.12	0.364 ± 0.17	0.659 ± 0.06	0.720 ± 0.05	0.785 ± 0.06	**0.873 ± 0.02**
z_3	0.903 ± 0.02	0.755 ± 0.07	0.797 ± 0.06	0.960 ± 0.01	0.966 ± 0.01	**0.971 ± 0.01**

Table 6: (More details for original paper Table 6) R^2 (best possible score is 1.0) values and standard error (**ste**) 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), generalized count linear dynamical system (GCLDS), Poisson feed-forward neural network linear dynamical system (PfLDS), and Poisson-Gaussian process latent variable model(P-GPLVM). 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.

Dim	PLDS	P-GPFA	LFADS	PfLDS	P-GPLVM	GP-RNN
2	0.68 ± 0.15	0.69 ± 0.10	0.73 ± 0.17	0.73 ± 0.14	0.74 ± 0.15	**0.77± 0.13**
4	0.69 ± 0.15	0.72 ± 0.12	0.74 ± 0.16	0.73 ± 0.13	0.75 ± 0.14	**0.78±0.14**
6	0.72 ± 0.17	0.73 ± 0.15	0.74 ± 0.20	0.74 ± 0.13	0.77 ± 0.10	**0.80±0.10**
8	0.74 ± 0.15	0.74 ± 0.10	0.75 ± 0.15	0.75 ± 0.14	0.77 ± 0.16	**0.80±0.10**
10	0.75 ± 0.15	0.74 ± 0.12	0.77 ± 0.17	0.76 ± 0.13	0.77 ± 0.15	**0.81±0.12**

Table 7: (More details for original paper Table 7) Predictive R^2 with and standard error (**ste**) 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.

References

[1] Anqi Wu, Nicholas A 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*, pages 3496–3505, 2017.

[2] 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*, pages 163–171, 2016.

[3] 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.