Supplementary Materials for
Active Learning of Functional Networks from Spike Trains

1 Proof of Theorem 3.1

Here we give the extended version of the proof.

First, we have

$$H(\mathcal{N}(\boldsymbol{u}_{t+1}, \boldsymbol{C}_{t+1})) = \frac{1}{2}\log|\boldsymbol{C}_{t+1}| + const,$$

where $|\boldsymbol{C}_{t+1}|$ represents the determinant of $\boldsymbol{C}_{t+1}$.

In order to calculate $\boldsymbol{C}_{t+1}$, we have

$$\boldsymbol{C}_{t+1}^{-1} = -\frac{\partial^2 \log p(\boldsymbol{w}|\boldsymbol{u}_{t+1}, \boldsymbol{C}_{t+1})}{\partial \boldsymbol{w}^2},$$

because the inverse covariance matrix equals to the second partial derivative of the log-Gaussian density function w.r.t. $\boldsymbol{w}$. Furthermore, we can expand $\log p(\boldsymbol{w}|\boldsymbol{u}_{t+1}, \boldsymbol{C}_{t+1})$ as follows,

$$\begin{aligned}
&\log p(\boldsymbol{w}|\boldsymbol{u}_{t+1}, \boldsymbol{C}_{t+1})\\
&\approx \log p(\boldsymbol{w}|\boldsymbol{s}_{1:t+1}, \boldsymbol{r}_{1:t+1})\\
&= \log p(\boldsymbol{w}) + \log p(\boldsymbol{r}_{1:t}|\boldsymbol{w}, \boldsymbol{s}_{1:t}) + \log p(\boldsymbol{r}_{t+1}|\boldsymbol{w}, \boldsymbol{s}_{t+1}) +\\
&const\\
&\approx \log p(\boldsymbol{w}|\boldsymbol{u}_t, \boldsymbol{C}_t) + \log p(\boldsymbol{r}_{t+1}|\boldsymbol{w}, \boldsymbol{s}_{t+1}) + const,
\end{aligned}$$

where $\boldsymbol{u}_t$ and $\boldsymbol{C}_t$ are derived from previous observations $(\boldsymbol{s}_{1:t}, \boldsymbol{r}_{1:t})$ and treated as known parameters.

Now, we can get

$$\boldsymbol{C}_{t+1}^{-1} = \boldsymbol{C}_t^{-1} - \frac{\partial^2 \log p(\boldsymbol{r}_{t+1}|\boldsymbol{w}, \boldsymbol{s}_{t+1})}{\partial \boldsymbol{w}^2}. \tag{1.1}$$

The second term of Eq. (1.1) is the Fisher information (the negative of the second derivative of the log likelihood with respect to $\boldsymbol{w}$) and can be calculated as

$$\begin{aligned}
&F(\boldsymbol{w}, \boldsymbol{s}_{t+1}, \boldsymbol{r}_{t+1})\\
&= -\frac{\partial^2 \log p(\boldsymbol{r}_{t+1}|\boldsymbol{w}, \boldsymbol{s}_{t+1})}{\partial \boldsymbol{w}^2}\\
&= -\frac{\partial^2 \log p(\boldsymbol{r}_{t+1}|\boldsymbol{W}, \boldsymbol{s}_{t+1})}{\partial \boldsymbol{W}^2}\\
&= -\frac{\partial D(\boldsymbol{W}, \boldsymbol{s}_{t+1}, \boldsymbol{r}_{t+1})}{\partial \boldsymbol{W}}\\
&= -\frac{\partial D(\boldsymbol{W}, \boldsymbol{s}_{t+1}, \boldsymbol{r}_{t+1})}{\partial e^{\boldsymbol{W}\cdot \boldsymbol{s}_{t+1}}} \cdot \frac{\partial e^{\boldsymbol{W}\cdot \boldsymbol{s}_{t+1}}}{\partial \boldsymbol{W}\cdot \boldsymbol{s}_{t+1}} \cdot \frac{\partial \boldsymbol{W}\cdot \boldsymbol{s}_{t+1}}{\partial \boldsymbol{W}}
\end{aligned}$$

Given $\frac{\partial \boldsymbol{AXB}}{\partial \boldsymbol{X}} = \boldsymbol{B}^T \otimes \boldsymbol{A}$, we have

$$\begin{aligned}
&F(\boldsymbol{w}, \boldsymbol{s}_{t+1}, \boldsymbol{r}_{t+1})\\
&= (\boldsymbol{s}_{t+1} \otimes \boldsymbol{I}) \cdot diag(e^{\boldsymbol{W}\cdot \boldsymbol{s}_{t+1}}) \cdot ({\boldsymbol{s}_{t+1}}^T \otimes \boldsymbol{I}).
\end{aligned}$$

where $\otimes$ represents Kronecker product, *diag* is a function that takes all the elements of a matrix and reconstruct them into a diagonal matrix and $\boldsymbol{I}$ is a $m \times m$ identity matrix. It's interesting to see that the Fisher information does not depend on the response vector $\boldsymbol{r}_{t+1}$.

Given the above equations, the objective function can be solved as

$$\begin{aligned}
&\arg\min_{\boldsymbol{s}_{t+1}} E_{\boldsymbol{r}_{t+1}} H(p(\boldsymbol{w}|\boldsymbol{s}_{1:t+1}, \boldsymbol{r}_{1:t+1}))\\
&= \arg\min_{\boldsymbol{s}_{t+1}} E_{\boldsymbol{r}_{t+1}} \log|\boldsymbol{C}_{t+1}|\\
&= \arg\max_{\boldsymbol{s}_{t+1}} E_{\boldsymbol{r}_{t+1}} \log|{\boldsymbol{C}_{t+1}}^{-1}|\\
&= \arg\max_{\boldsymbol{s}_{t+1}} \log|\boldsymbol{C}_t^{-1} + F(\boldsymbol{w}, \boldsymbol{s}_{t+1}, \boldsymbol{r}_{t+1})|\\
&= \arg\max_{\boldsymbol{s}_{t+1}} (\log|{\boldsymbol{C}_t}^{-1}| + \log|\boldsymbol{I} + \boldsymbol{C}_t \cdot F(\boldsymbol{w}, \boldsymbol{s}_{t+1}, \boldsymbol{r}_{t+1})|)\\
&= \arg\max_{\boldsymbol{s}_{t+1}} \log|\boldsymbol{I} + \boldsymbol{C}_t \cdot F(\boldsymbol{w}, \boldsymbol{s}_{t+1}, \boldsymbol{r}_{t+1})|\\
&= \arg\max_{\boldsymbol{s}_{t+1}} tr(\log(\boldsymbol{I} + \boldsymbol{C}_t \cdot F(\boldsymbol{w}, \boldsymbol{s}_{t+1}, \boldsymbol{r}_{t+1})))\\
&\approx \arg\max_{\boldsymbol{s}_{t+1}} tr(\boldsymbol{C}_t \cdot F(\boldsymbol{w}, \boldsymbol{s}_{t+1}, \boldsymbol{r}_{t+1}))\\
&= \arg\max_{\boldsymbol{s}_{t+1}} (e^{\boldsymbol{W}\cdot \boldsymbol{s}_{t+1}})^T \cdot ({\boldsymbol{s}_{t+1}}^T \otimes \boldsymbol{I}) \cdot \boldsymbol{C}_t \cdot (\boldsymbol{s}_{t+1} \otimes \boldsymbol{J}),
\end{aligned} \tag{1.2}$$

where *tr* is the function to calculate the trace of a matrix, $\boldsymbol{I}$ is an $m \times m$ identity matrix and $\boldsymbol{J}$ is a $m \times 1$ vector with ones in all its entries.

2 Synthetic Data

Here we give the detailed procedure to generate synthetic data.

We use three steps to generate simulated spike train data. First, the structure of the functional network is proposed. Then, a GLM parameter matrix is created according to the functional network. Finally, simulated spike train data is generated by running the GLM. All the parameters in the simulation process are chosen to mimic real neurons.

Propose structure of the functional network. In this step, we use two methods to build the functional network: 1) manually creating some small networks to show the effectiveness and intuition of our models; 2) generating random networks to evaluate the generalizability of our models. The random networks are generated as follows. First the number of nodes m is chosen. Then for each pair of nodes in the network, they have a probability of 0.3 of getting connected and for each connection, it has a probability of 0.2 of being an inhibitory connection. We use $[1, 20]$ as time lags for an inhibitory connection and a time lag drawn from $[1, 2]$ for an excitatory connection. Each node also has an inhibitory connection directed to itself to mimic the refractory period of neurons.

Create GLM parameter matrix from the functional network. First, for any neuron i, the value of b_i is randomly drawn from $[-9, -3]$, which means a spontaneous firing rate ranging from 0.0001 to 0.05. Then other values in the parameter matrix are picked according to the connections in this network. For example, if there is an edge from neuron i to neuron j with time lag l, the corresponding value in the parameter matrix $\boldsymbol{W}$ will be set to a none-zero value to represent the strength of this connection. If the connection is excitatory, the value will be randomly drawn from $[0, -b_i]$. If the connection is inhibitory, the value will be set to b_i to ensure the inhibition of all excitatory inputs.

Generate simulated spike train data. Given a GLM parameter matrix, we run the GLM with initial states of neurons set to be 1. At each time step, the probabilities of getting a spike are computed before Gaussian noises with mean 0 and variance 0.0005 are added. Then Bernoulli distributions with these probabilities are used to draw values of 0 or 1. If a neuron is intervened in the simulated experiment, its value will be set to 0 no matter what the probability is. We continue this process until the desired length of recording is reached.

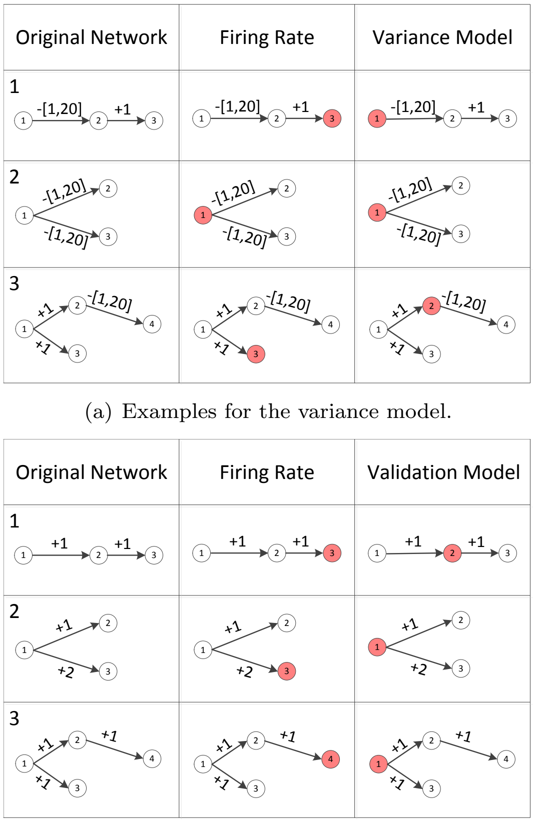
3 Small Networks

Here we report experiments that could validate the intuitions behind our active learning models.

Since the variance and validation model are using different strategies to choose interventions, we use some small networks to illustrate the effectiveness of their intuitions. In the following experiments, in addition to the aforementioned two baseline methods, we add another one, *First recording*, which infers the GLM only with the initial recording.

Variance model. Figure 1(a) shows three types of functional networks with inhibitory connections. All the neurons in these networks have a spontaneous firing rate of 0.05. The first column in Figure 1(a) shows the structures of the proposed networks. The second column shows the neuron that will be picked as the intervention target if it's selected by firing rate. The third column shows the neuron that the variance model will choose. For instance, in network 1, node 3 is chosen according to firing rate and node 1 is chosen by the variance model. It's clear that when network 1 is inferred by using only observational data, the connection from node 1 to node 2 will likely be missed. By silencing node 1, which is chosen by the variance model, we can discover this connection. One the other hand, choosing interventions by firing rate will not help.

To illustrate the effectiveness of the variance model, we show the error e of the inferred model when an additional recording is add under the guidance of different methods. As shown in Figure 2(a), if the simulated intervention experiments are conducted under the guidance of the variance model, we can always achieve the highest accuracy.

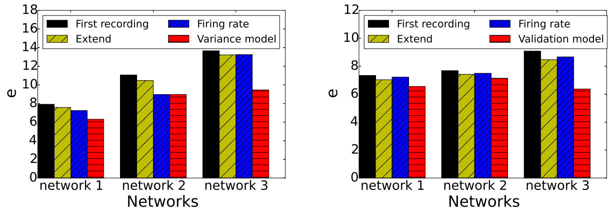


(a) Examples for the variance model.

(b) Examples for the validation model.

Figure 1: The neurons that will be chosen by different methods for small networks.

Validation model. Similarly for the validation model, we manually create three functional networks shown in Figure 1(b). The networks are constructed with only excitatory connections and each node has a firing rate of 0.0001 except node 1 has a firing rate of 0.05, which means the activities of the networks are mainly driven by node 1. By comparing the intervention targets chosen by the validataion model and according to the firing rate, we can see that the validation model could always choose the node that can maximally filter spurious connections. As shown in Figure 2(b), the validation model could achieve the highest accuracy gain in all three cases.



(a) When examples for the variance model are used.

(b) When examples for the validation model are used.

Figure 2: The errors of the inferred functional networks by different methods for small networks.