

LEARNING NETWORK STRUCTURES FROM FIRING PATTERNS

Amin Karbasi[†] Amir Hesam Salavati* Martin Vetterli*

[†] School of Engineering and Applied Sciences, Yale, USA

*School of Computer and Communication Sciences, EPFL, Switzerland

September 30, 2015

Abstract

How can we decipher the hidden structure of a network based on limited observations? This question arises in many scenarios ranging from social to wireless and to neural networks. In such settings, we typically observe the nodes' behaviors (e.g., the time a node learns about a piece of information, or the time a node gets infected by a disease), and we are interested in inferring the true network over which the diffusion takes place. In this paper, we consider this problem over a neural network where our aim is to reconstruct the connectivity between neurons merely by observing their firing activity. We develop an iterative NEURal INFerence algorithm NEUINF to identify the type of effective neural connections (i.e. excitatory/inhibitory) based on the Perceptron learning rule. We provide theoretical bounds on the average performance of NEUINF as well as numerical analysis to compare the performance of the proposed approach to some previous art.

1 Introduction

Reconstruction of neuronal networks connectivity has been a major challenge for the past decades. Currently, invasive procedures are the only "reliable" approach to map the *connectome*. However, these approaches are prohibitively complex and time-consuming: it took more than 10 expert/year to map the whole connectome of *C. Elegans*, comprising only 302 neurons and a few thousands synaptic connections [1]. To map the whole brain of *fruit flies*, with around 10,000 neurons, we would have to spend around 4700 expert/year [2,3]. Following the same approach and using current technology, it is estimated that it will take around 14 billion man/year to completely map the human brain's connectome [2]. Although there is an increasing effort to make some parts of the invasive procedures automated, such approaches remain impractical for mid-sized/large networks. Furthermore, the current invasive techniques cannot be applied to live animals/humans, as it involves dissecting the brain.

As a result, *inverse* methods with the focus on mapping the (whole or partial) connectome from the activity of the neurons are compelling (or perhaps the only viable) alternatives. They can be applied to live specimen and potentially require much less time

and labor. Furthermore, rapid advancements in recording technologies has made it possible to simultaneously monitor the activities of tens [4] to hundreds of neurons [5], [6] (for a good review, see [7]). Upcoming technologies will increase this number significantly in near future [8]. There has also been progress in simultaneously recording and stimulating the neurons [9], [10], [11]. These advancements will soon alleviate the issue of data availability. Similarly, significant efforts have been made to improve the accuracy of inverse approaches in identifying neuronal connections (more on that in the next section). However, the current algorithms either fall short of capturing important details of realistic neural graphs (e.g., the synaptic delays) or become computationally infeasible as the size of the network grows or its characteristics change.

In this paper, we propose an inverse approach for identifying the *nature* of (effective) connections between neurons that reflects their causal relationship. More precisely, we consider a neural network where neurons follow the standard Leaky Integrate-and-Fire (LIF) model [12] and connections are associated with random delays. We propose an iterative NEURal INFerence algorithm NEUINF, that provably identifies both the effective connections and their types (i.e., excitatory/inhibitory). Furthermore, through an exhaustive set of numerical simulations (ranging over different network models and neuronal firing patterns), we study the performance of NEUINF compared to the previous state of the art.

2 Related Work

Identifying neural connections from a set of recorded neural activities is an instance of network tomography [13] and has been extensively studied in the past. *Cross Correlogram* is perhaps the most widely-used method to identify (functional) connection between a pair of neurons [7]. However, approaches based on Cross Correlogram usually fall short of identifying *causal* relation or *effective* connectivity of neurons. This is why some authors proposed *Granger causality measure* as an alternative to overcome the drawback of Cross Correlogram (see, [14]).

Another very popular approach to identify effective connectivities is Generalized Linear Models (GLMs) [15]. Methods based on GLM essentially take the effects of stimulus, self-history of the neurons and contribution of other neurons into account and calculate the *filters* through which all these factors affect the firing response of each neuron. GLM was recently used in reconstructing a real physiological circuit from recorded neural data [16]. The approaches based on GLMs are provably accurate (i.e. they identify the correct set of connections in the underlying graph) *if* the neural model used to generate spike data matches the one used in GLM. Otherwise, the final estimation will have some bias and variance from the correct results [17]. Furthermore, traditional GLM approaches suffer from high computational costs. Recently, however, several accurate approximations have been suggested to resolve this issue [17], [18]. Nevertheless, the convergence proofs still only applies when the model for neurons and that of GLM closely match each other.

Bayesian approaches are also widely used. In [19], a Maximum a Posteriori (MAP) approach is applied to solve the problem of connection identification, with excellent results in the regime of limited data, although at the expense of high computational costs. Bayesian approaches have also been used in identifying connections directly

from calcium-imaging data [20].

Recently, another line of work has focused on the connection mapping problem that are more tailored to LIF neurons. In particular, Bussel et al. [21] addresses the problem by converting the non-linear firing behavior of LIF neurons into a set of linear equations, which can be solved given a sufficient number of recorded samples. While being efficient, this algorithm is highly sensitive to the accuracy of spike time and heavily relies on the knowledge of model parameters, e.g. synaptic delays, which are very difficult to obtain. Additionally, Memmesheimer et al. [22] proposed an inference algorithm based on the Perceptron learning rule, similar to Baldassi et al. [23], for which they proved that under accurate spike times it identifies a simple n -to-1 feed forward network. They also proposed a heuristic extension that works with *finite precision* in recorded spike times. Nevertheless, their model does not take into account (random) synaptic delays. Moreover, as we will see later, having extra post-synaptic neurons even in a simple feed forward scenario can have a dramatic effect on the performance of the inference algorithm when the structure of the graph (i.e., here being feed-forward) is not known a priori.

Finally, we should mention that the *consistency problem* even for a n -to-1 feed forward network is NP-hard. In words, determining whether or not there exists a set of delays and weights such that we can fully match the set of input firing patterns to the output is very difficult [24]. Although this result does not necessarily mean finding such a configuration is impossible, it shows that finding provable "positive learning results" for the case of spiking neuron is quite difficult.

The proposed approach in this paper is similar to GLMs in that it does not rely on the knowledge of propagation delays, but differs in the final iterative algorithm and the provided theoretical analysis in that it only makes very general assumptions about the nature of connections. Furthermore, while the final algorithm update rule is inspired by the Perceptron learning rule, it differs from previous similar approaches in that the effect of propagation delays are considered in the algorithm.

3 Model and Problem Statement

As mentioned earlier, in this paper we consider a Leaky Integrate-and-Fire (LIF) model for the neurons [12]. In this model, each neuron accumulates the incoming (weighted) spikes from all of its neighbors and fires if the net sum exceeds a threshold θ . Otherwise, the membrane voltage decays exponentially fast. The differential equation for the LIF model is given by

$$\frac{dV(t)}{dt} = \frac{I(t) - V(t)}{\tau}, \quad \frac{dI(t)}{dt} = -\frac{I(t)}{\tau_e},$$

where V and I are the membrane voltage and input current, respectively, and τ and τ_e are the membrane time constants.

As for the graph, we assume that there are two types of connections: *excitatory* and *inhibitory*. In accordance with biological neuronal networks, we assume that the excitatory connections are more numerous than the inhibitory ones. Furthermore, we assume that the weight of connections is fixed and to keep the network balanced, we

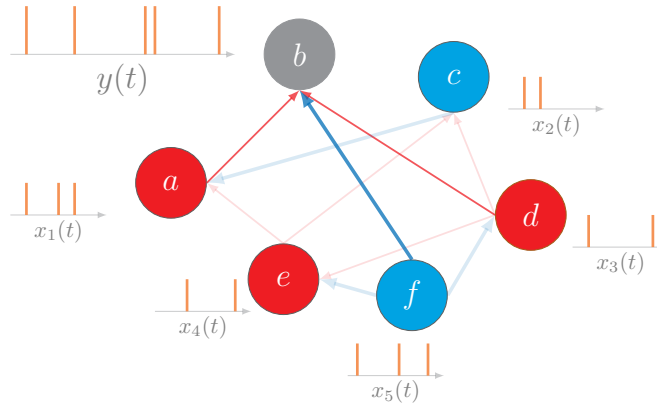


Figure 1: Network model: a recurrent neural network where we try to identify the incoming connections of node b by observing the spike trains $x_1(t), \dots, x_5(t)$ and $y(t)$. Here we assume that edges have random delays and neurons can be excitatory or inhibitory.

require for the inhibitory connections to have a larger weight (in magnitude) than excitatory ones. We typically set the weight of an excitatory connection to $+1\text{mV}$ and that of an inhibitory connection to $-\delta\text{mV}$, where $\delta = n_{exc}/n_{inh}$, and n_{exc} and n_{inh} are the number of excitatory and inhibitory neurons, respectively. Following Dale’s principle [25], we assume that all outgoing connections of a neuron have the same type, i.e., they are all either excitatory or inhibitory.

We also assume that neural connections have intrinsic delays which represent the time it takes for the information to propagate through the axons and synapses. The delay for each link is assumed to be a random number in the interval $[0, d_{\max}]$, where $d_{\max} > 0$ is the maximum delay. The delays do not change and once assigned, remains fixed. Figure 1 illustrates the model used to generate data in the simplest case considered in this paper.

Throughout this paper, we assume that the the connectivity matrix, as well as the intrinsic connection delays are *unknown* to the inference algorithm. The goal is to infer the connectivity pattern by only observing the spike trains. With NEUINF we propose an algorithm that iteratively identifies the connections by producing an analogue association matrix that reflects the accumulated belief of each connection. This matrix can then be transformed into a ternary adjacency matrix to infer the type of connections: void (no connection), excitatory, or inhibitory. The identified connections indicate the causal influence of neurons on each other. Note that in a setting where the activity of all nodes in the network are recorded, the resulting effective connectivity will be equivalent to capturing the synaptic connections.

4 Proposed Algorithm

In order to identify the connections, we can consider each post-synaptic neuron separately and find its incoming connections. Let $x_i(t)$ and $y(t)$ denote the firing state of

the i -th pre-synaptic and the post-synaptic neuron at time t . For the ease of presentation, let us assume momentarily that the firing state of the post-synaptic neuron at time t depends only on the states of its neighbors at time t . Thus, we effectively ignore the propagation delay and the integration procedure. Our goal is to find a set of weights W such that the average quadratic error E over a recording period T is minimized:

$$\min_W E = \min_W \frac{1}{T} \sum_{t=1}^T \left| f \left(\sum_{i=1}^n W_i x_i(t) - \theta \right) - y(t) \right|^2. \quad (1)$$

Here, W_i is our *belief* about the connection weight from the i -th pre-synaptic to the post-synaptic neuron and $f(\cdot)$ is the Heaviside step function (or its continuous approximations). By taking the derivative, we obtain

$$\nabla E = \frac{2}{T} \sum_{t=1}^T (f(\langle W, X_t \rangle - \theta) - y(t)) f'(\langle W, X_t \rangle - \theta) X_t^\top, \quad (2)$$

where $X_t = (x_1(t), \dots, x_n(t))$ is the state of pre-synaptic neurons at time t . Now, we can iteratively update our beliefs about W according to the following update rule:

$$W(\tau + 1) = W(\tau) - \alpha_\tau \nabla E(\tau), \quad (3)$$

where α_τ is a small number representing the learning rate.¹ The above update rule follows the standard gradient descent approach. However, a closer look at (2) reveals some interesting characteristics: the weight W_i , will be updated (for a given t) if both of the following conditions are satisfied: 1) $x_i(t) = 1$, and 2) $f(\langle W(\tau), X_t \rangle - \theta) \neq y(t)$. In words, we update the weight W_i in round τ if and only if the pre-synaptic neuron i fires and $W(\tau)$ does not correctly predict the state $y(t)$ of the post-synaptic neuron. Note that these two conditions are in fact reminiscent to the well-known Perceptron learning rule in neural networks [26].

Equipped with these observations, we propose NEUINF that does not use the derivative of function $f(\cdot)$ and takes into account both the unknown propagation delays and the neural integration procedure. Since we do not know the synaptic delays, we define an *integration window* Δ such that only the firings during this window are considered for the updates. In other words, if neuron i has fired between $t - \Delta$ and t we take such firings into account, otherwise we discard them. The details are shown below. Algorithm 1 is deterministic. In practice, we have found that a stochastic version of the above algorithm, which we call STOCHASTIC NEUINF, works much better. The only difference between the two algorithms is in the weight update rule. In each iteration τ , the deterministic version updates any weight i with $\Delta W_i \neq 0$, whereas in the stochastic version weights with $\Delta W_i \neq 0$ are updated with some probability β independently at random. Setting β inversely proportional to the recording duration improves the results even further.

In order to analyze the performance of NEUINF we make the following mild assumptions:

¹We usually choose α_τ in such a way that $\sum_\tau \alpha_\tau \rightarrow \infty$ and $\sum_\tau \alpha_\tau^2 < C$, for some positive constant C .

Algorithm 1 NEUINF

Input: The observations $\{x_i(t)\}$ and $\{y(t)\}$ over period T , a maximum number of iterations τ_{\max} .

Output: Connections Belief Matrix W

Initialize $W(0) = (0, \dots, 0)_{1 \times n}$

for $\tau = 1 \rightarrow \tau_{\max}$ **do**

for $t = 1, \dots, T$ **do**

 set $v_i(t) = \sum_{t_i=t-\Delta}^t x_i(t), \forall i$

 set $\hat{y}(t) = f(\sum_i W_i(t)v_i(t) - \theta)$

if $\hat{y}(t) \neq y(t)$ **then**

$\Delta W = (\hat{y}(t) - y(t)) V_t^T$

 Break

end if

end for

 Update: $W(\tau + 1) = W(\tau) - \alpha_\tau \Delta W$

end for

(A1) The probability of firing for a (pre-synaptic) neuron does not depend on its being excitatory or inhibitory.

(A2) Excitatory (inhibitory) connections increase (decrease) the probability of the post-synaptic neuron to fire.

With the above assumptions we can prove that NEUINF provides the desired ordering for the *average* values of different connection types, i.e., it returns higher values for excitatory connections than inhibitory or non-existent (void) connections. While this may seem like a weak guarantee, due to the generality (and almost triviality) of assumptions A1 and A2, it can be applied to a wide range of scenarios. We can have tighter guarantees if we make more strict assumptions about the type of the neurons and statistical properties for the firing patterns (as in the case of GLM) or structure of the graph (e.g., random/small-world graphs).

Theorem 1. *Under the assumptions A1 and A2, and for a sufficiently small learning step α_τ , NEUINF outputs*

$$\bar{w}_{exc} \geq \bar{w}_{void} \geq \bar{w}_{inh},$$

where \bar{w}_{exc} , \bar{w}_{void} and \bar{w}_{inh} denote the expected values returned by NEUINF for excitatory, void, and inhibitory connections, respectively. Here, the expectation is taken over the randomness of pre-synaptic spike patterns.

The proof of the theorem is provided in the supplementary materials. To give the reader an intuition, we use induction to show that the correct ordering for the *average value* of our beliefs about the type of synaptic connections is preserved in each iteration of the inference algorithm NEUINF.

Remark 1. *The two assumptions A1 and A2 are quite broad and are generally held in practice. As we do not make any assumptions on the network type (e.g., random/small-world graphs), its structure (e.g., feed forward/recurrent) and the statistical properties for the firing patterns, Theorem 1 applies to a wide range of scenarios.*

In the next section, we compare the performance of NEUINF with previous art.

5 Numerical Experiments

To assess the effect of different network parameters on the performance of the algorithm, we generate several Erdos-Renyi *directed random graphs* where two nodes are connected with probability p and independently of every other node. In accordance with real biological networks we assume that the probability of a connection being excitatory p_+ is five times that of being inhibitory p_- , namely, $p_+ = 5p_-$. For each graph, we set the firing threshold to $\theta = 5\text{mV}$ and the maximum delay to $d_{\max} = 10\text{ms}$. We also assume that the weight of an excitatory connection is $+1\text{mV}$ whereas that of an inhibitory connection is $-\delta\text{mV}$, with $\delta = \frac{p_+}{p_-}$, to keep the network balanced. For all the simulations, we assume that neurons have a refractory period of 1ms and the membrane potential is reset after a neuron fires. The time constants of artificial neurons is set to $\tau = 10\text{ms}$ and $\tau_e = 2\text{ms}$. All the data corresponding to neural activity is generated by the Brian neural simulator [27] to make sure that the numerical experiments are as close as possible to the real neural networks.

For our experiments, we consider both feed-forward and recurrent networks. In the case of feed-forward networks, the pre-synaptic neurons, and in the case of recurrent networks, all neurons received traffic from "outside" which account for spike trains from neurons whose activities is not recorded. This traffic is modeled as a random process that results in each neuron being stimulated with some probability q independent of the activity of neurons "inside" the network ².

Once the data is generated, we compare the performance of NEUINF with Cross Correlogram and GLM and study the effect of different parameters such as recording duration T , knowledge of the network topology type (feed-forward or recurrent), and importance of estimating synaptic delays for increasing the spike prediction accuracy. Note that there are several ways of implementing the GLM algorithm and to interpret the final result in terms of the neural connections. In this paper, we use the code provided in [15]³ and modified it to serve our setup. Given the variety methods to identify the type of pairwise neural connections (e.g., peak value of or the area underneath the post-synaptic coupling filters), we have selected the criteria that gives best results in each considered case.

We evaluate the performance of the considered algorithms according to the following three criteria:

1. The average values of the *association matrix* for the excitatory/void/inhibitory connections.
2. The *precision* and *recall* of the algorithm over the *ternary adjacency matrix*.
3. The spike prediction capability.

We use the K-Means algorithm (for $K = 3$) to transform the analog association matrix into the digital adjacency matrix. As for the spike prediction quality, \bar{Q} , the performance is measured in terms of the number of mismatches between the predicted and

²This incoming traffic will also get the network up and running from the initial resting state.

³Available at http://pillowlab.cps.utexas.edu/code_GLM.html, thanks to *Pillow Lab*.

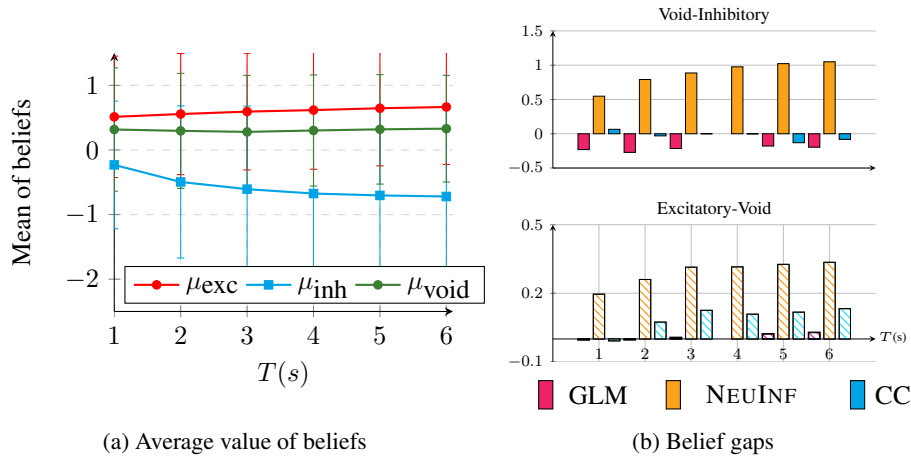


Figure 2: The left figure shows the effect of recording duration T on the belief returned for *incoming* connections by different inference algorithms for a recurrent network of $n = 60$ neurons.

actual output spike train, for all post-synaptic neurons, divided by the total number of spikes in the actual spike train. More precisely, let $y_j(t)$ and $\hat{y}_j(t)$ denote the actual and predicted state of the post-synaptic neuron j at time t . Then, we have $\bar{Q} = (\sum_t \sum_{j=1}^n |y_j(t) - \hat{y}_j(t)|) / (\sum_t \sum_{j=1}^n y_j(t))$.

5.1 Numerical Results

Here we only report the results for the more general case of recurrent networks. Related results for feed-forward topology is reported in the longer version of this article [28]. For this part, we consider a network of $n = 60$ neurons, 50 excitatory and 10 inhibitory, with a connection probability of 0.2. Figure 2a illustrates the effect of T on the average value of beliefs returned by the considered algorithms. As shown in the figure, STOCHASTIC NEUINF outputs an association matrix with the desired ordering, i.e. higher values for excitatory connections and lower values for inhibitory ones.

To make the comparison between different algorithms more consistent, the association matrix returned by each algorithm is normalized in such a way that the ℓ_∞ -norm of each column is 1 and then the matrix is whitened. Figure 2b shows the gap between the beliefs for each pair of connection types (i.e., between excitatory and "void" as well as "void" and inhibitory). Again, STOCHASTIC NEUINF outperforms the other considered algorithms.

The *precision* and *recall* of different approaches, after "ternarification" using the K-Means algorithm, is shown in Figure 3. In identifying excitatory and "void" connections, STOCHASTIC NEUINF outperforms both CC and GLM. Furthermore, although it seems the CC algorithm is better in identifying inhibitory connections, it does that at the expense of very low *precision*.

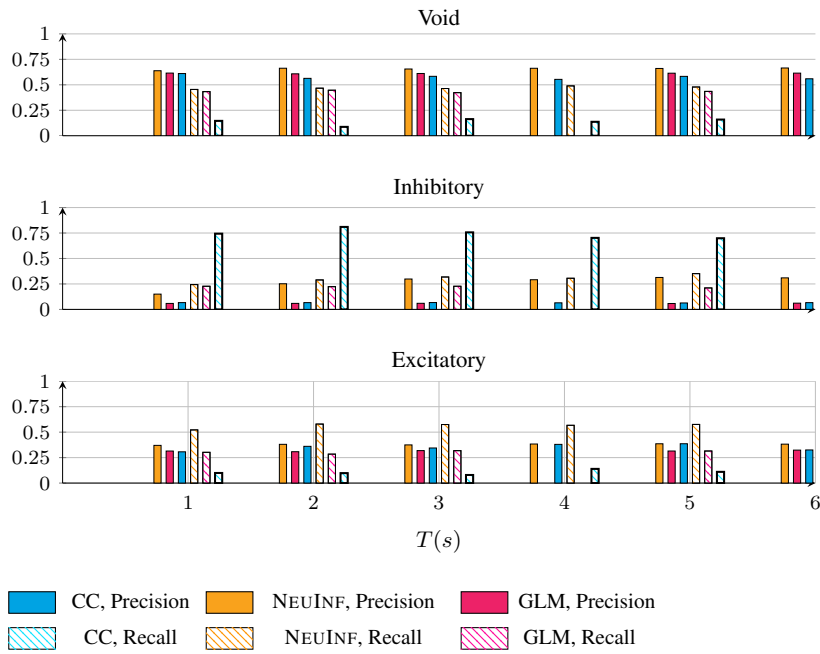


Figure 3: Performance of the algorithms after "ternarification" of the association matrix.

5.2 Sparsity helps

We also observe another trend in our simulations: Sparsity, both in the firing patterns and network topology, improves the performance. Figure 11 illustrates the performance of STOCHASTIC NEUINF in differentiating connection types in feed-forward/recurrent networks for different values of connection probability p , and probability of being triggered by outside traffic, q .

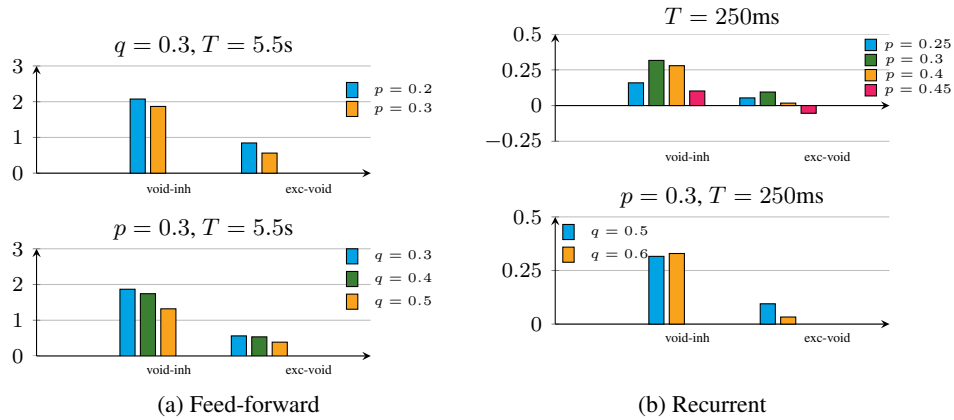


Figure 4: The gap between excitatory-void and void-inhibitory connections in the association matrix returned by STOCHASTIC NEUINF for feed-forward and recurrent network in different scenarios.

6 Conclusions

In this paper, we introduced an iterative online algorithm we call NEUINF that identifies the structure of a neural network from neuron’s activity alone. In contrast to previous algorithms that need very precise and often impractical information about the neural network (e.g., topology, synaptic delays, precise timings of spike trains, etc), NEUINF performs well even in the face of imprecise spike train recordings with unknown topology and synaptic delays. We also theoretically showed that NEUINF is capable of differentiating between connection types as it receives enough samples. By using the numerical simulations that precisely mimic the activity of real neural networks, we also evaluated the performance of NEUINF in comparison with previous state-of-the-art and observed that in all considered cases NEUINF outperforms them.

References

- [1] Duncan J Watts and Steven H Strogatz, “Collective dynamics of small-world networks,” *nature*, 1998.
- [2] Stephen M Plaza, Louis K Scheffer, and Dmitri B Chklovskii, “Toward large-scale connectome reconstructions,” *Curr. Opin. Neurobiology*, vol. 25, pp. 201–210, 2014.
- [3] Ann-Shyn Chiang, Chih-Yung Lin, Chao-Chun Chuang, Hsiu-Ming Chang, Chang-Huain Hsieh, Chang-Wei Yeh, Chi-Tin Shih, Jian-Jheng Wu, Guo-Tzau Wang, Yung-Chang Chen, et al., “Three-dimensional reconstruction of brain-wide wiring networks in drosophila at single-cell resolution,” *Current Biology*, vol. 21, no. 1, pp. 1–11, 2011.
- [4] Rodrigo Perin, Thomas K Berger, and Henry Markram, “A synaptic organizing principle for cortical neuronal groups,” *Proceedings of the National Academy of Sciences*, vol. 108, no. 13, pp. 5419–5424, 2011.
- [5] György Buzsáki, “Large-scale recording of neuronal ensembles,” *Nature neuroscience*, vol. 7, no. 5, pp. 446–451, 2004.
- [6] Benjamin F Grewe, Dominik Langer, Hansjörg Kasper, Björn M Kampa, and Fritjof Helmchen, “High-speed in vivo calcium imaging reveals neuronal network activity with near-millisecond precision,” *Nature methods*, vol. 7, no. 5, pp. 399–405, 2010.
- [7] Emery N Brown, Robert E Kass, and Partha P Mitra, “Multiple neural spike train data analysis: state-of-the-art and future challenges,” *Nature neuroscience*, vol. 7, no. 5, pp. 456–461, 2004.
- [8] Gaute T Einevoll, Felix Franke, Espen Hagen, Christophe Pouzat, and Kenneth D Harris, “Towards reliable spike-train recordings from thousands of neurons with multielectrodes,” *Current opinion in neurobiology*, vol. 22, no. 1, pp. 11–17, 2012.
- [9] Dion Khodagholy, Jennifer N Gelin, Thomas Thesen, Werner Doyle, Orrin Devinsky, George G Malliaras, and György Buzsáki, “Neurogrid: recording action potentials from the surface of the brain,” *Nature neuroscience*, 2014.
- [10] Carolina Gutierrez Herrera and Antoine R Adamantidis, “An integrated microprobe for the brain,” *Nature Biotechnology*, vol. 33, no. 3, pp. 259–260, 2015.
- [11] Gabriel Bertotti, Dmytro Velychko, Norman Dodel, Stefan Keil, Dirk Wolansky, et al., “A cmos-based sensor array for in-vitro neural tissue interfacing with 4225 recording sites and 1024 stimulation sites,” in *Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE*. IEEE, 2014, pp. 304–307.
- [12] Wulfram Gerstner and Werner M Kistler, *Spiking neuron models: Single neurons, populations, plasticity*, Cambridge university press, 2002.
- [13] Rui Castro, Mark Coates, Gang Liang, Robert Nowak, and Bin Yu, “Network tomography: recent developments,” *Statistical science*, pp. 499–517, 2004.
- [14] Sanggyun Kim, David Putrino, Soumya Ghosh, and Emery N Brown, “A granger causality measure for point process models of ensemble neural spiking activity,” *PLoS computational biology*, vol. 7, no. 3, pp. e1001110, 2011.
- [15] Jonathan W Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M Litke, EJ Chichilnisky, and Eero P Simoncelli, “Spatio-temporal correlations and visual signalling in a complete neuronal population,” *Nature*, vol. 454, no. 7207, pp. 995–999, 2008.

- [16] Felipe Gerhard, Tilman Kispersky, Gabrielle J Gutierrez, Eve Marder, Mark Kramer, and Uri Eden, “Successful reconstruction of a physiological circuit with known connectivity from spiking activity alone,” *PLoS computational biology*, vol. 9, no. 7, pp. e1003138, 2013.
- [17] Alexandro D Ramirez and Liam Paninski, “Fast inference in generalized linear models via expected log-likelihoods,” *Journal of computational neuroscience*, vol. 36, no. 2, pp. 215–234, 2014.
- [18] Yury V. Zaytsev, Abigail Morrison, and Moritz Deger, “Reconstruction of recurrent synaptic connectivity of thousands of neurons from simulated spiking activity,” *Journal of Computational Neuroscience*, vol. 39, no. 1, pp. 77–103, 2015.
- [19] Ian H Stevenson, James M Rebesco, Nicholas G Hatsopoulos, Zach Haga, Lee E Miller, and Konrad P Körding, “Bayesian inference of functional connectivity and network structure from spikes,” *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 17, no. 3, pp. 203–213, 2009.
- [20] Yuriy Mishchenko, Joshua T Vogelstein, Liam Paninski, et al., “A bayesian approach for inferring neuronal connectivity from calcium fluorescent imaging data,” *The Annals of Applied Statistics*, vol. 5, no. 2B, pp. 1229–1261, 2011.
- [21] Frank Van Bussel, Birgit Kriener, and Marc Timme, “Inferring synaptic connectivity from spatio-temporal spike patterns,” *Frontiers in computational neuroscience*, vol. 5, 2011.
- [22] Raoul-Martin Memmesheimer, Ran Rubin, Bence P Ölveczky, and Haim Sompolinsky, “Learning precisely timed spikes,” *Neuron*, vol. 82, no. 4, pp. 925–938, 2014.
- [23] Carlo Baldassi, Alfredo Braunstein, Nicolas Brunel, and Riccardo Zecchina, “Efficient supervised learning in networks with binary synapses,” *BMC Neuroscience*, vol. 8, no. Suppl 2, pp. S13, 2007.
- [24] Wolfgang Maass and Michael Schmitt, “On the complexity of learning from spiking neurons with temporal coding,” *Information and Computation*, vol. 153, no. 1, pp. 26–46, 1999.
- [25] Henry Dale, “Pharmacology and nerve-endings,” *J. Royal Society of Medicine*, vol. 28, no. 3, pp. 319–332, 1935.
- [26] Frank Rosenblatt, *The perceptron, a perceiving and recognizing automaton Project Para*, Cornell Aeronautical Laboratory, 1957.
- [27] Dan Goodman and Romain Brette, “Brian: a simulator for spiking neural networks in python,” *Frontiers in neuroinformatics*, vol. 2, 2008.
- [28] Amin Krabsi, Amir Hesam Salavati, and Martin Vetterli, “Learning network structures from firing patterns,” 2015.

A Proof of Theorem 1

In the following, we prove the theorem for the deterministic version of the algorithm, i.e., NEUNF. The extension to the stochastic version, STOCHASTIC NEUNF, is straightforward.

We use induction to prove the theorem. With some abuse of notation, let $w_j(\tau)$ denote the output value of the algorithm at round τ of optimization for the incoming connection from neuron j . We will show that

$$\bar{w}_{exc}(\tau) > \bar{w}_{void}(\tau) > \bar{w}_{inh}(\tau).$$

To start, note that since we start from the all-zero vector for the inferred connectivity matrix, initially we will have $\hat{y}(t) = 0$, i.e., no predicted output spikes. This means that given a sufficiently small step size, α_τ , the initial updates will be only when there is an output spike, i.e., $y(t^*) = 1$ for some t^* , and some of pre-synaptic neurons have fired, i.e., $v_i(t^*) > 0$, for some i 's. This means that, for some t^* , we will have

$$\bar{w}_j(1) = \bar{w}_j(0) - \alpha_\tau \mathbb{E}\{\Delta w_j(0)\} = \alpha_\tau \Pr\{y(t^*) > 0 | v_j(t^*) > 0\} \Pr\{v_j(t^*) > 0\}$$

Now, if neuron j is connected to the output neuron through an excitatory, inhibitory or void connection, we will have

$$\bar{w}_{exc}(1) = \alpha_\tau \Pr\{y(t^*) > 0 | G_j > 0, v_j(t^*) > 0\} \Pr\{v_j(t^*) > 0\},$$

$$\bar{w}_{void}(1) = \alpha_\tau \Pr\{y(t^*) > 0 | G_j = 0, v_j(t^*) > 0\} \Pr\{v_j(t^*) > 0\},$$

$$\bar{w}_{inh}(1) = \alpha_\tau \Pr\{y(t^*) > 0 | G_j < 0, v_j(t^*) > 0\} \Pr\{v_j(t^*) > 0\}$$

By definition, we have $\Pr\{y(t^*) > 0 | G_j > 0, v_j(t^*) > 0\} > \Pr\{y(t^*) > 0 | G_j = 0, v_j(t^*) > 0\} > \Pr\{y(t^*) > 0 | G_j < 0, v_j(t^*) > 0\}$. Thus, we will have

$$\bar{w}_{exc}(1) > \bar{w}_{void}(1) > \bar{w}_{inh}(1)$$

Now assuming that induction's criteria hold up to iteration τ , we show that $\bar{w}_{exc}(\tau+1) > \bar{w}_{void}(\tau+1) > \bar{w}_{inh}(\tau+1)$. To this end, we distinguish two events

1. Predicted firing pattern matched the actual output firing pattern, for all $t = 1, \dots, T$.
2. There is a t^* for which $y(t^*) \neq \hat{y}(t^*)$, i.e., there is a mismatch between prediction and reality.

The first event means that the weights will not get updated and the algorithm will terminate. Thus, the ordering up to this iteration is preserved and the desired result is proven. In the case of second event above, we will have

$$\begin{aligned} \bar{w}_j(\tau+1) &= \bar{w}_j(\tau) - \alpha_\tau \mathbb{E}\{\Delta w_j(\tau)\} \\ &= \bar{w}_j(\tau) + \alpha_\tau \Pr\{v_j(t^*) > 0\} [\Pr\{y(t^*) = 1 | v_j(t^*) > 0\} - \Pr\{y(t^*) = 0 | v_j(t^*) > 0\}] \\ &= \bar{w}_j(\tau) + \alpha_\tau \Pr\{v_j(t^*) > 0\} [2\Pr\{y(t^*) = 1 | v_j(t^*) > 0\} - 1]. \end{aligned} \quad (4)$$

Now we know that the induction criteria holds up to iteration τ . Thus, to show $\bar{w}_{exc}(\tau+1) > \bar{w}_{void}(\tau+1) > \bar{w}_{inh}(\tau+1)$, it is sufficient to show that

$$\Pr\{y(t^*) = 1 | G_j > 0, v_j(t^*) > 0\} > \Pr\{y(t^*) = 1 | G_j = 0, v_j(t^*) > 0\} > \Pr\{y(t^*) = 1 | G_j < 0, v_j(t^*) > 0\},$$

which holds by the definition of excitatory, inhibitory and void connections. This proves the theorem.

Remark 2. *Note that the above theorem is only based on the assumptions that excitatory (inhibitory) neurons have positive (negative) effect in triggering the other neighboring neurons and that the probability of firing at any given moment does not depend on the type of the neuron (i.e., its being excitatory or inhibitory). In other words, we did not make assumptions on the network type (e.g., ErdsRnyi random graph or small-world networks), its structure (e.g., being feed-forward or recurrent) and the statistical properties for the firing patterns of neurons (e.g., they being independent of/correlated with each other), which makes the theorem rather general.*

However, we can show that the speed of the algorithm and how fast it reaches certain level of accuracy depends on several properties of the network. In particular, it is not difficult to show that for a feed-forward random graph, i.i.d. input firing patterns result in faster convergence and improved accuracies whereas correlated input reduces the speed of convergence. Due to lack of space, we have not included the corresponding numerical results for these scenarios though.

B More Numerical Results

B.1 Feed-forward networks

In the main body of the paper, we reported the results for a scenario with recurrent networks. Here, we focus on a more special case with feed-forward topology (which is, for instance, more relevant to transmitting sensory signals).

We start with a single layer feed-forward network that has $n = 75$ pre-synaptic and $m = 15$ post-synaptic neurons, the connection probability being set to $p = 0.2$. Figure 5 illustrates the effect of the recording duration T on the accuracy (in terms of the average value of "beliefs" about the three connection types). We have considered three approaches: Cross Correlogram (CC), GLM, and STOCHASTIC NEUINF with $\beta \simeq 0.1$. To make the comparison between different algorithms more consistent, the association matrix returned by each algorithm is normalized in such a way that the ℓ_∞ -norm of each column is 1 and then the matrix is whitened. As shown in the figure, STOCHASTIC NEUINF outputs an association matrix with higher values for excitatory connections and lower values for inhibitory ones. Furthermore, longer durations improve the quality of the results. In the figure, μ_{recurr} illustrates the average of the values returned by the considered algorithms for the non-existent *recurrent* connections in the post-synaptic layer.

However, the ordering between the *average* belief values is only one side of the story. The *variance* of the beliefs as well as the *gap* between them are the other important factors that will affect the results returned by the K-Means algorithm. Figure 6a

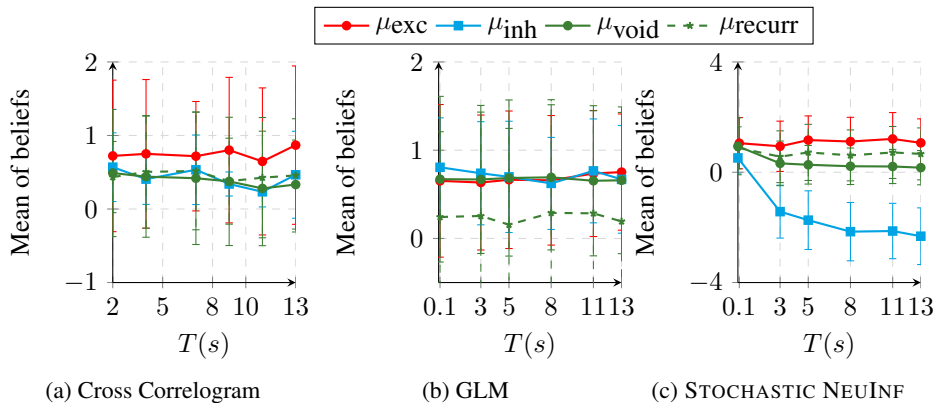


Figure 5: The mean value of beliefs for the *incoming* connections given by different inference algorithms for a feed-forward network of $n = 75$ and $m = 15$ pre- and post-synaptic neurons.

illustrates the gap between the beliefs (i.e., between excitatory and "void" as well as "void" and inhibitory) for the three inference algorithms. As can be seen, STOCHASTIC NEUINF significantly outperforms the other considered algorithms. Furthermore, in the same figure, we also show (by crossed-orange bars) the performance of STOCHASTIC NEUINF when it is given as side information that the graph to infer is a feed-forward network. As we observe, this side information considerably improves the performance.

Furthermore, once the returned analog graphs are turned into a ternary graph by the K-Means method, we can also evaluate the quality of spike prediction. Figure 6b illustrates the evolution of \bar{Q} as a function of T , the duration of recorded samples, when NEUINF is applied to a feed-forward network with $n = 60$ pre- and $m = 48$ post-synaptic neurons. As shown in the figure, the quality improves as T grows. Furthermore, note that NEUINF consistently performs better in reconstructing the output spikes. Figure 6b also emphasizes the importance of identifying connection delays in reproducing the output spikes: when we perform spike reproduction with the actual connectivity matrix, G , without considering the synaptic delays, the reproduction is not 100% accurate and a mismatch of around 30% will remain.

Figure 7 illustrates the *precision* and *recall* for feed-forward network with $n = 75$ pre- and $m = 15$ post-synaptic neurons, in a scenario where algorithms operated in the topology-aware situation, i.e. they knew that the graph is feed-forward.⁴ Here again STOCHASTIC NEUINF outperforms the other algorithms.

B.2 Examples of Inferred Graphs

Figure 8 illustrates examples of association matrices inferred by STOCHASTIC NEUINF and GLM, in comparison with the actual ternary matrix used in producing neural data (shown in Figure 8a), for a feed-forward network and in the topology-aware situation

⁴Similar results can be obtained for the topology-unaware situation. However, based on the inferred graphs shown in Figure 9, the K-Means algorithm might not be the best solution to transform the association matrix into the ternary adjacency matrix. More clever algorithms should perform better in that task.

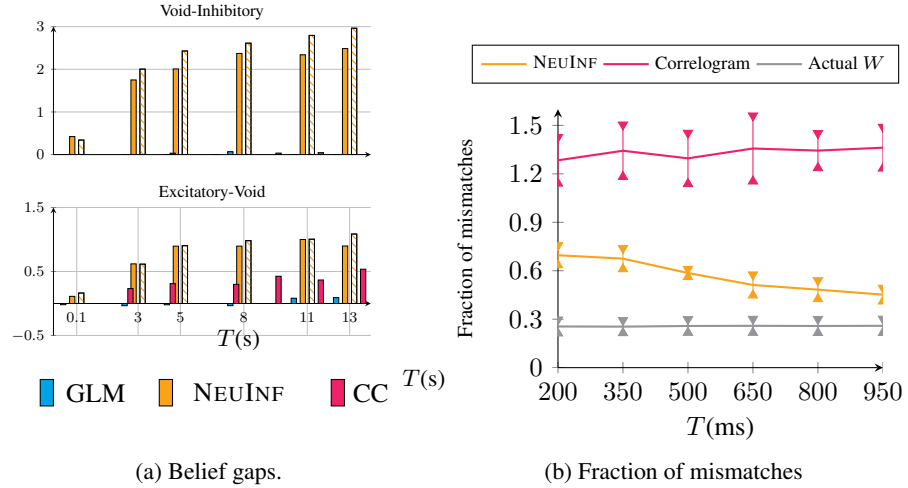


Figure 6: The left figure shows the effect of recording duration T on the belief gaps for different algorithms, where the crossed-bars indicate a scenario when the topology of the graph is known *a priori*. The right figure illustrates the effect of T on spike prediction quality.

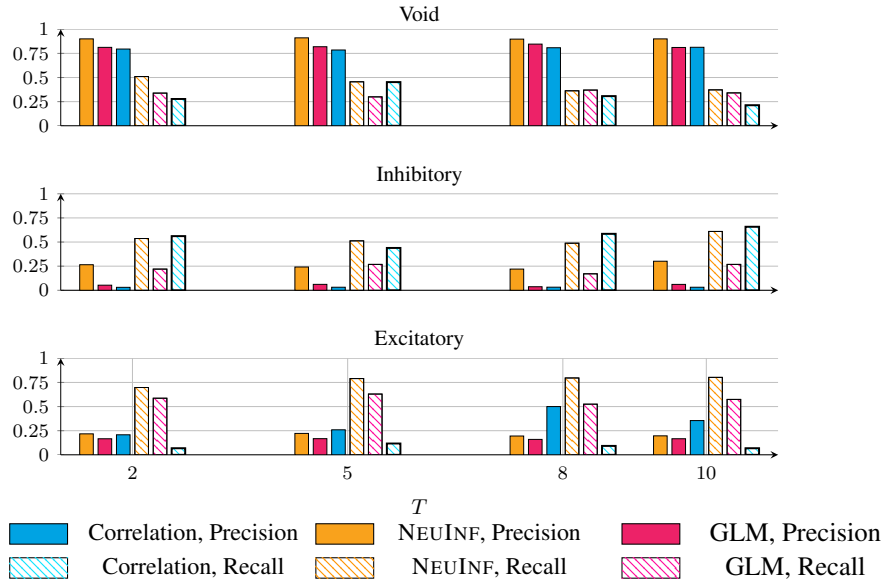


Figure 7: Performance of the algorithms after "ternarification" of the association matrix.

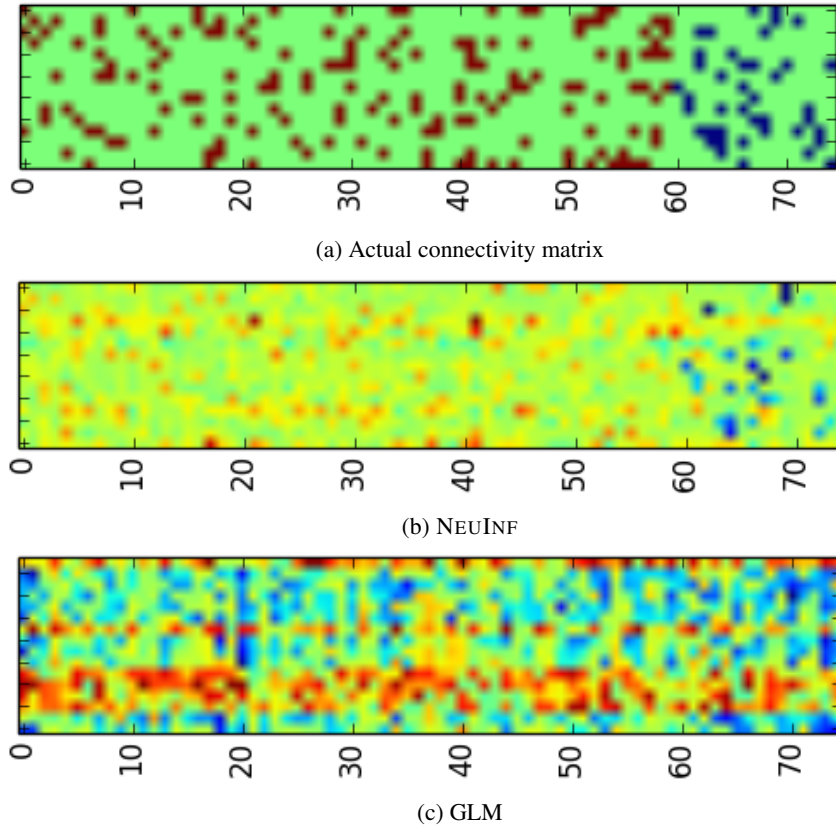


Figure 8: The inferred graphs for different algorithms. Red, blue and green pixels in the actual graph represent excitatory, inhibitory and void connections in the actual connectivity matrix.

(for all algorithms). As can be seen from the figure, the graph returned by STOCHASTIC NEUINF is much more similar to the actual version.

Figure 9 illustrates the same scenario, but for a topology-unaware situation.

Similar results for the recurrent scenario is shown in Figure 10 for an example graph returned graph by STOCHASTIC NEUINF in comparison with the ternary graph used to produce neural data (the "ground truth").

B.3 Sparsity helps

We also observe another trend in our simulations: Sparsity, both in the firing patterns and network topology, improves the performance. Figure 11 illustrates the performance of STOCHASTIC NEUINF in differentiating connection types in feed-forward/recurrent networks for different values of connection probability p , and probability of being triggered by outside traffic, q .

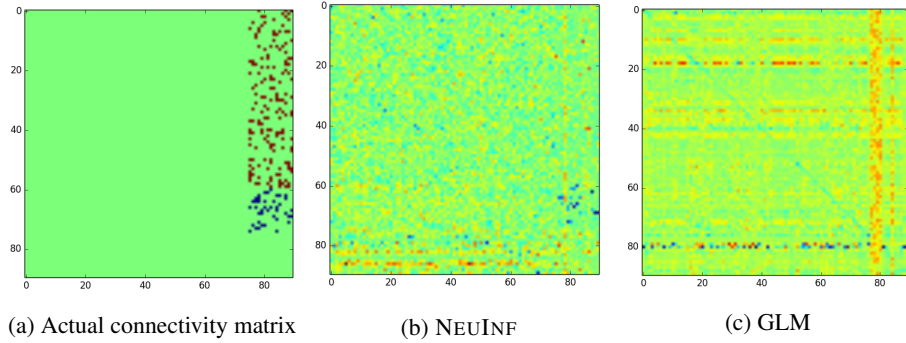


Figure 9: The inferred graphs for different algorithms. Red, blue and green pixels in the actual graph represent excitatory, inhibitory and void connections in the actual connectivity matrix.

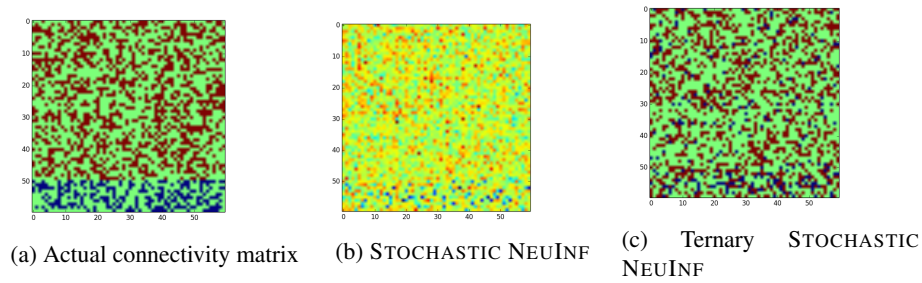


Figure 10: The inferred graphs for different algorithms. Red, blue and green pixels in the actual graph represent excitatory, inhibitory and void connections in the actual connectivity matrix.

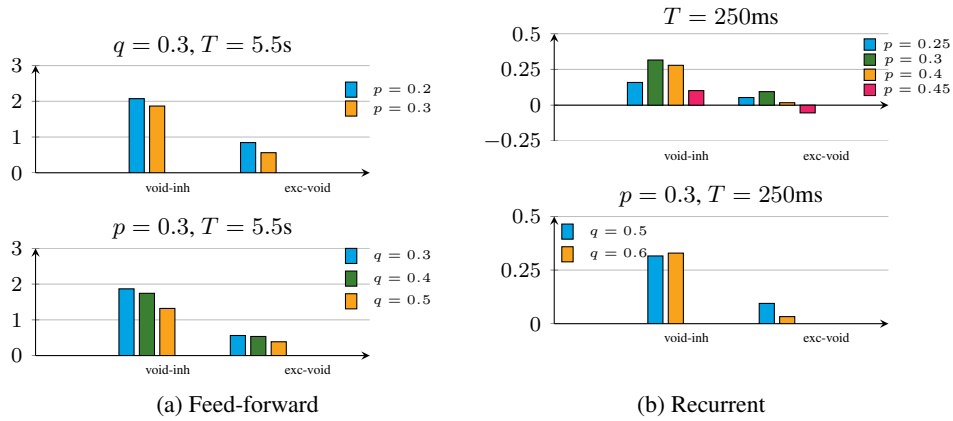


Figure 11: The gap between excitatory-void and void-inhibitory connections in the association matrix returned by STOCHASTIC NEUINF for feed-forward and recurrent network in different scenarios.