Scalable covariance-based connectivity inference for synchronous neuronal networks

Taehoon Kim[1,2,✉], Dexiong Chen[2], Philipp Hornauer[1], Sreedhar Saseendran Kumar[1], Manuel Schröter[1], Karsten Borgwardt[2], and Andreas Hierlemann[1]

[1]Bio Engineering Laboratory, Department of Biosystems Science and Engineering, ETH Zurich, Basel, Switzerland
[2]Machine Learning and Computational Biology Laboratory, Department of Biosystems Science and Engineering, ETH Zurich, Basel, Switzerland

We present a novel method for inferring connectivity from large-scale neuronal networks with synchronous activity. Our approach leverages Dynamic Differential Covariance to address the associated computational challenges. First, we analyze spike trains generated from Leaky Integrate-and-Fire network simulations and evaluate the performance of several off-the-shelf multivariate connectivity inference methods. Next, we introduce a new approach, Fractional Dynamic Differential Covariance (FDDC), and demonstrate that it consistently outperforms the other methods. Finally, we apply FDDC to experimental data to assess the topological organization of inferred graphs of in vitro neural network recordings obtained using high-density microelectrode arrays (HD-MEAs). Our results indicate that FDDC-derived graphs exhibit a significant negative correlation between small-worldness and measures of network synchrony. In contrast, graphs inferred through the well-established pairwise correlation method do not show such a correlation. This finding implies that the graphs obtained through FDDC provide stronger evidence in support of the theoretical notion that networks with clustered connections tend to exhibit higher levels of synchronizability. We hypothesize that our findings have implications for the development of scalable connectivity inference methods for large-scale neural network data.

connectivity inference | graph theory | functional connectivity | neural network simulation | microelectrode array | in-vitro neural networks

Correspondence: ***taehoon.kim@bsse.ethz.ch***

Introduction

Large-scale electrophysiological methodology, such as high-density microelectrode arrays (HD-MEAs) play an important role in studying neuronal activity and for gaining insights into the functioning of neural networks (1). Recent years have seen significant progress in this field, particularly an increase in the number of recording electrodes, resulting in improved spatio-temporal recording resolution (2–5). However, such gains in information content of neuronal recordings come with a significant increase in data volume - which can pose several computational challenges. The sheer volume of data generated by these arrays can quickly become overwhelming, making it difficult to process the data in a timely manner. These challenges become even more pronounced when inferring functional connectivity from the recordings, because an increase in the number of recording electrodes often corresponds to an increase in the number of recorded neurons. As connectivity inference methods often rely on pairwise calculations (1), the inference becomes computationally intensive as the number of recorded neurons grows. Moreover, many of today's inference methods rely on the assumption that neurons in the network exhibit asynchronous firing activity, making them less suitable for inferring connections during synchronous network activity (6–9). Although approaches to filter out time windows of synchronized activity exist, such as detecting population bursts (10) or factoring out common activity through multivariate modeling (11, 12), they introduce additional computational load and conceptual complications. Therefore, previous studies have investigated multivariate methods to uncover the true synaptic connectivity between neurons by mitigating the influence of signals from other neurons in the network. Baker et al. (13) examined the theoretical feasibility of recovering neural connectivity and found that utilizing the precision matrix, derived from the inverse of the covariance matrix of spike trains, seems to be an effective approach for inferring underlying neuronal connections across diverse experimental scenarios. Furthermore, there is a considerable body of literature on connectivity inference from the perspective of maximum likelihood estimation with sparsity assumptions, often emphasizing the sparse estimation of precision matrices. While the majority of such approaches has focused on inferring functional connectivity between brain regions (14–20), relatively few studies have applied the latter to the inference of connectivity between individual neurons (21, 22). Although these methods hold considerable promise for scaling up to large networks (23), further exploration is required to evaluate their efficacy for large-scale neuronal networks (e.g., n>1000 neurons) and to assess their potential limitations.

The main goal of this study was to tackle the computational challenges associated with inferring connectivity in large-scale, synchronous neural networks. We, therefore, applied and evaluated scalable methods on simulated data and proposed an improved connectivity inference method. Specifically, we compared the performance of off-the-shelf multivariate connectivity inference methods that rely on the covariance computed from the spike trains of observed neurons in a network. We focused on these methods as more sophisticated methods may not scale for networks with large numbers of neurons (i.e. n>1000 neurons).

As a testing ground, we used spike trains generated from random networks with 2000 Leaky Integrate-and-Fire (LIF) neurons that mimicked synchronous network firing activity of hippocampal networks through adaptation of synaptic currents(24). We compared the performance of these inference methods based on their AUROC (Area Under the Re-

ceiver Operating Characteristic Curve) and AUPRC (Area Under the Precision-Recall Curve) values. Then we made two modifications to the best method, Dynamic differential covariance (DDC)(25), and introduced an improved method, Fractional Dynamic Differential Covariance (FDDC), which consistently showed superior performance. Finally, we applied FDDC to infer connections from in vitro neural recordings obtained using high-density microelectrode arrays (HD-MEAs) and compared the resulting graphs to those obtained from a well-studied pairwise inference method (cross correlogram (CCG)). Analysis of the topology of the inferred networks revealed that CCG-derived graphs showed higher small-worldness compared to the FDDC-derived graphs. Moreover, FDDC-derived graphs showed a clear negative correlation between small-worldness and a measure of network synchrony (Participation ratio) across different graph densities, while CCG-derived graphs did not. These results highlight the utility of fractional differentiation and inversion of the covariance matrix for extracting pairwise relations between neurons in synchronous networks. We argue that our findings have important implications for developing connectivity inference methods that can handle networks with large numbers of neurons.

Materials and Methods

In this section, we outline the LIF simulation that was used to simulate the activity of neural networks. Then we provide an overview of the different connectivity inference methods and describe the in vitro neural recordings used in this study. Lastly, we address how synchrony in the experimental neural recordings was quantified, as well as how we evaluated the topologically clustered organization of the inferred neuronal connectivity.

Simulating network activity with an adaptive exponential integrate-and-fire model. We employed an adaptive exponential integrate-and-fire model, as described in (26), for simulating network activity. Our implementation was based on the methodologies presented in (24, 27). In this context, we denote C the membrane capacitance and V_i the membrane potential of neuron i. Additionally, E_L refers to the leaky membrane potential and g_L is the leakage conductance.

$$C\frac{dV_i(t)}{dt} = -g_L(V_i(t)-E_L)+g_L\Delta\exp\left[\frac{(V_i(t)-V_T)}{\Delta}\right], \\ -I_{\text{syn}_i}(t)-w_i(t)+\sigma\sqrt{\frac{2}{\tau_m}}\xi(t),$$

$$I_{\text{syn}_i}(t) = g_E(t)(V_i(t)-E_E)+g_I(t)(V_i(t)-E_I),$$

where V_T represents the effective threshold for generating spikes. When the membrane potential V_i reaches V_T, it is reset to V_{reset} and remains constant for the duration of T_{refrac} (see Table S1). The threshold slope factor is denoted by Δ, and $I_{\text{syn}_i}(t)$ is the total synaptic current received by neuron i from its presynaptic neurons. The reversal potential for excitatory and inhibitory synapses is expressed as E_E and E_I, respectively. $\xi(t)$ is a Gaussian white noise with σ being the noise strength. Lastly, τ_m refers to the time constant of the noise, which can be calculated as $\tau_m = C/g_L$.

The following equation represents the change in excitatory and inhibitory conductance, denoted g_E and g_I respectively:

$$\tau_{E,I}\frac{dg_{E,I}(t)}{dt} = -g_{E,I}(t)+Q_{E,I}\sum_k \delta(t-t_k),$$

Excitatory/inhibitory conductance (g_E/g_I) increases by a quantity of Q_E/Q_I each time the presynaptic excitatory/inhibitory neuron fires at a specific time t_k. Furthermore, the synaptic conductance g_E/g_I undergoes an exponential decay, following a corresponding time constant τ_E/τ_I.

The following equation describes the adaptation current of neuron i, represented by w_i:

$$\tau_{w_i}\frac{dw_i(t)}{dt} = a(V_i(t)-E_L)-w_i(t)+b\sum_j \delta(t-t_j),$$

where the parameter a models the sub-threshold adaptation. When neuron i fires at time j, w_i increases by a value of b. Additionally, w_i decays exponentially with the time constant τ_{w_i}.

All LIF network simulations consisted of n = 2000 neurons with 80% excitatory and 20% inhibitory neurons. All inhibitory neurons were set to show no adaptation, e.g., (a,b) = (0,0), while excitatory neurons were set to specific (a,b) parameters (i.e., obtained through parameter search) to implement adaptation. All neurons were randomly connected with a connection probability of 2%. For each connected edge, the synaptic delay was implemented by randomly selecting a value in the range of [0 5] ms. At the beginning of the simulation, a shared Poissonian spike train with the same excitatory strength Q_E and a mean firing rate of 500 Hz was applied to drive all neurons in the network for 100 ms. Afterward, the network was purely driven by its own recurrent activity. For the downstream analysis, the first second (s) of activity was removed to consider only spontaneous recurrent activity. All described equations were numerically integrated with a step size of 0.1 ms using the 'Brian2' python library[1]. Network activities were simulated for 20 min as longer simulations could not be handled with the computing resources utilized in this study (e.g., computing instances with 64 GB RAM). All parameters used for the simulation are provided in the Supplementary material (Table S1).

Parameter search. A parameter search was performed to select the set of parameters (a,b,g_E,g_I) for generating network activities ranging from asynchronous irregular firing activity (Async) to synchronous firing activity (Sync). The search was performed in two steps. First, by fixing the adaptation parameters (a,b) = $(1\mu\text{S}, 5\text{nA})$ to implement "weakly adapting cells" as stated in (24), excitatory/inhibitory conductances (g_E,g_I) were searched in the range of [0 100] with a step size of 5 nS (400 combinations)(Fig. S1). For each parameter set, the network activity was simulated for

[1] https://brian2.readthedocs.io/en/stable/

5 s. We selected $(g_E, g_I) = (10, 65)$ as the default excitatory/inhibitory conductances, as this set of parameters showed spontaneous firing activity with the highest excitatory conductance (g_E) without saturation in the network firing rate (e.g., 200 Hz for $T_{\text{refrac}} = 5$ ms)(Fig. S1). As the second step, (a, b) was searched in the range of [0 80] with a step size of 1 μS, nA) (6400 combinations). 5 s of activity was simulated for each parameter set, and the simulated activities were quantified in how synchronous the spike trains were in the network.

Characterization of network activity. To characterize the network firing synchrony of LIF simulations, we computed the mean coefficient of the variation of interspike intervals (CV_{ISI}) and mean pairwise correlation (MPC) as implemented in Destexhe (2009) (24). We only considered neurons showing firing rates > 0.01 Hz for the following quantities. CV_{ISI} measures the temporal regularity of the network firing activity and is calculated as follows.

$$\text{CV}_{ISI} = \left\langle \frac{\sigma_i^{ISI}}{\mu_i^{ISI}} \right\rangle,$$

where σ_i^{ISI} and μ_i^{ISI} denote standard deviation and mean of the ISIs of neuron i. The bracket $\langle\rangle$ indicates an average over all neurons in the network.
The MPC, on the other hand, measures the degree of synchrony between all neurons in a network.

$$\text{MPC} = \left\langle \frac{\text{Cov}(\mathbf{x}_i, \mathbf{x}_j)}{\sigma(\mathbf{x}_i)\sigma(\mathbf{x}_j)} \right\rangle,$$

where $\mathbf{x}_i$, $\mathbf{x}_j$ denote binned spike trains of neurons i and j computed using a bin size of 5 ms. We considered the simulated network activity to show asynchronous activity only if $\text{CV}_{ISI} > 1$ and MPC < 0.1. If a simulated network did not satisfy one of these criteria, the network was considered to exhibit synchronous firing activity (24).

Parameter selection for LIF network simulations. The parameter search for the LIF networks showed that with $b > 60$ and $a < b$, simulations were more likely to show synchronous network activity (Fig. S1). This observation was in agreement with previously reported values (24, 28). From all 6400 combinations of (a, b), the parameter sets that generated simulations without lasting spontaneous activity or with very high network firing rates (30 Hz) were discarded. For both asynchronous and synchronous networks, we chose two parameter sets that exhibited the lowest or highest MPC, respectively. We then generated five 20 min simulations for each parameter set. From these ten network simulations of each network type, we selected the five most synchronous or asynchronous networks and used them for the inference tasks.

Perturbation of LIF network simulations using white noise. The inference was performed on noise-free LIF networks unless specified otherwise. For noise-perturbed simulations, the same connections and synaptic delays as in the non-perturbed counterpart were used. A wide range of noise currents was applied with values both smaller and larger than that of the leaky current (e.g., $g_L(V_i(t) - E_L)$), as we observed qualitative changes in network behavior in terms of MPC and firing irregularity (CV_{ISI}) in cases where the noise was larger than the leaky current (Fig. S2).

Functional connectivity inference methods. In this study, we focused on connectivity inference methods that could scale to large-scale networks (n $>$ 1000 neurons). When binning of spike trains was necessary, we used 5 ms as a bin size to reflect the synaptic delay of the simulation unless specified otherwise. We first implemented a well-established pairwise inference method that is widely used to study functional connectivity of neural circuits, the results of which served as a baseline in this study (1, 29). We anticipated that the method would perform exceptionally well for asynchronous networks. However, we expected its performance to be suboptimal for synchronous networks, where the spike time correlations between neurons would be masked by the synchronous firing activity of the entire network. We then assessed how the selected set of scalable multivariate inference methods compared to the pairwise method and drew conclusions on their usefulness.

Pairwise cross-correlograms. To assess pairwise relations between neurons, we adapted the cross-correlogram method (CCG) implemented in English et al. (30). For all neuron pairs in the network, cross-correlograms were computed using a bin size of 1 ms for a time window [-50 +50] ms. Then the count in each bin was compared to the baseline value (λ_{base}), which was generated as follows. The observed cross-correlograms were convolved with the partially hollow Gaussian Kernel (31) with a standard deviation of 10 ms, with a hollow fraction of 60% to model baseline (30). We estimated the probability of observing a spike count $x \geq n$ in the postsynaptic bin t using the Poisson distribution with a continuity correction (31).

$$P(x \geq n|\lambda_{\text{base}}(t)) = 1 - \sum_{a=0}^{n-1} \frac{e^{-\lambda_{\text{base}}(t)}\lambda_{\text{base}}(t)^a}{a!} - 0.5 * \frac{e^{-\lambda_{\text{base}}(t)}\lambda_{\text{base}}(t)^n}{n!}$$

We also computed $1 - P(x \geq n|\lambda_{\text{base}}(t))$ to estimate the likelihood that the connection was inhibitory. For each neuron pair, we took the maximum (peak, excitatory connection) and the minimum (trough, inhibitory connection) in the postsynaptic bins [0 5] ms, estimated the probability of respective values and compared the negative log-likelihoods. If the negative log-likelihood was larger for the maximum/minimum, we considered the edge to be an excitatory/inhibitory connection. As a result, we obtained a directed, weighted connectivity with each edge representing the negative log-likelihood.

Network deconvolution. We implemented Network deconvolution (ND) (32) to test if synaptic connectivity can be recovered from observed covariance matrices containing direct

and indirect interactions between neurons. We binned the spike trains of the neurons in the network with the predefined bin sizes (5 ms) to match the LIF simulations. Then, these binned spikes counts were used to compute the covariance matrix ($\mathbf{S}$), which represents the variance between the firing activity of neurons given the timescale (bin size). Under the strong assumption that the covariance matrix is a result of the graph signaling process of the latent graph $\mathbf{A}$, and the graph signal propagation is linear, time-invariant, and flow-preserving, the relation between the latent graph $\mathbf{A}$ and the covariance matrix $\mathbf{S}$ can be written as follows (32).

$$\mathbf{S} = \mathbf{A}(\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \cdots) = \mathbf{A}(\mathbf{I} - \mathbf{A})^{-\mathbf{1}},$$

which equivalently states $\mathbf{A} = \mathbf{S}(\mathbf{I} + \mathbf{S})^{-\mathbf{1}}$. From this relation, we computed ND in three steps. First, $\mathbf{S}$ was rescaled to ensure that the largest absolute eigenvalue of the $\mathbf{A}$ was strictly smaller than one. This was done by choosing a linear scaling factor α ($\mathbf{S}_{scaled} = \alpha\mathbf{S}$):

$$\alpha \leq min(\frac{\beta}{(1-\beta)\lambda_+}, \frac{-\beta}{(1+\beta)\lambda_-}),$$

where β is the target value for the largest absolute eigenvalue of $\mathbf{A}$ ($\beta < 1$), and λ_+/λ_- is the largest/smallest eigenvalue of $\mathbf{S}$. As a second step, the $\mathbf{S}_{scaled}$ was decomposed to get eigenvalues and eigenvectors such that $\mathbf{S}_{scaled} = \mathbf{U}\mathbf{\Sigma}_\mathbf{S}\mathbf{U}^{-\mathbf{1}}$. As the final step, we computed the diagonal eigenvalue matrix of $\mathbf{A}$, $\mathbf{\Sigma}_\mathbf{A}$, using the relation $\lambda_i^A = \frac{\lambda_i^S}{\lambda_i^S + 1}$. Here, λ_i^A was the i-th component of $\mathbf{\Sigma}_\mathbf{A}$ and λ_i^S was the i-th component of $\mathbf{\Sigma}_\mathbf{S}$. The latent graph $\mathbf{A}$ was then computed as $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}_\mathbf{A}\mathbf{U}^{-\mathbf{1}}$, and we considered $\mathbf{A}$ as the estimate of the connectivity. In this study, we chose $\beta = 0.99$ to get a sparse estimate of connectivity. For more details on the ND approach, we refer the readers to Feizi et al. (32).

Graphical lasso. The covariance matrix, derived from binned spike trains (bin size = 5 ms), can be inverted to obtain a precision matrix. Each entry in the precision matrix, $\mathbf{P}_{ij}$, quantifies the correlation between neuron i and j while factoring out all the activity of other neurons in the network. This is particularly valuable for identifying true pairwise correlations between neurons if the network exhibits synchronous firing activity. Instead of directly inverting the covariance matrix, the precision matrix can be estimated using sparse regularization methods, resulting in a sparser estimate of the precision matrix. This approach is beneficial when the covariance matrix is rank-deficient or too large for calculating the inverse. A widely used method among them is the graphical least absolute shrinkage and selection operator (graphical LASSO (Glasso))(33), which is defined as follows.

$$\min_{\mathbf{\Theta} \in \mathbb{S}^{\mathbf{n}}_{++}} -\text{logdet}\mathbf{\Theta} + \text{Tr}(\mathbf{S}\mathbf{\Theta}) + \gamma\|\mathbf{\Theta}\|_1,$$

where $\mathbb{S}^{\mathbf{n}}_{++}$ denotes a positive-semidefinite (PSD) matrix of dimensions $n \times n$. $\mathbf{S}$ is the observed covariance matrix, and $\mathbf{\Theta}$ is the estimated precision matrix. $\|.\|_1$ is an entrywise l_1-norm, and γ is regularization penalty. This objective can be optimized using the alternating direction method of multipliers (ADMM) (34) with an augmented Lagrangian, given by

$$\begin{aligned} L_\rho(\mathbf{\Theta}, \mathbf{Z}, \mathbf{\Lambda}) = &-\text{logdet}\mathbf{\Theta} + \text{Tr}(\mathbf{S}\mathbf{\Theta}) + \gamma\|\mathbf{Z}\|_1 + \text{Tr}[\Lambda(\mathbf{\Theta} - \mathbf{Z})] \\ &+ \frac{\rho}{2}\|\mathbf{\Theta} - \mathbf{Z}\|_F^2 \\ \text{s.t.} \quad &\mathbf{\Theta} = \mathbf{Z} \end{aligned}$$

where $\mathbf{\Lambda}$ is a dual variable and ρ is a penalty parameter ($\rho > 0$). $\|.\|_F$ is a Frobenius norm, e.g., a square root of the sum of squares of entries in a matrix. By additionally setting a scaled dual variable $\mathbf{U} = \mathbf{\Lambda}/\rho$, we get simplified update rules.

$$\begin{aligned} \mathbf{\Theta}^{k+1} &:= \underset{\mathbf{\Theta}}{\text{argmin}}(\text{Tr}(\mathbf{S}\mathbf{\Theta}) - \text{logdet}\mathbf{\Theta} + \frac{\rho}{2}\|\mathbf{\Theta} - \mathbf{Z}^k + \mathbf{U}^k\|_F^2) \\ \mathbf{Z}^{k+1} &:= \underset{\mathbf{Z}}{\text{argmin}}(\gamma\|\mathbf{Z}\|_1 + \frac{\rho}{2}\|\mathbf{\Theta}^{k+1} - \mathbf{Z} + \mathbf{U}^k\|_F^2) \\ \mathbf{U}^{k+1} &:= \mathbf{U}^k + \mathbf{\Theta}^{k+1} - \mathbf{Z}^{k+1} \end{aligned}$$

where $\mathbf{\Theta}$ is updated by setting the gradient to zero.

$$\mathbf{S} - \mathbf{\Theta}^{-1} + \rho(\mathbf{\Theta} - \mathbf{Z}^k + \mathbf{U}^k) = 0$$

By rewriting the equation,

$$\rho\mathbf{\Theta} - \mathbf{\Theta}^{-1} = \rho(\mathbf{Z}^k - \mathbf{U}^k) - \mathbf{S}$$

we can separate Θ on the left-hand side of the equation. The right-hand side of the equation can be decomposed as

$$\rho(\mathbf{Z}^k - \mathbf{U}^k) - \mathbf{S} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T$$

where $\mathbf{D}$ is the diagonal matrix of eigenvalues. If we consider $\tilde{\mathbf{\Theta}} = \mathbf{Q}\mathbf{\Theta}\mathbf{Q}^T$, we find the relation $\rho\tilde{\mathbf{\Theta}} - \tilde{\mathbf{\Theta}}^{-1} = \mathbf{D}$ and arrive at the following solution for each diagonal element of $\tilde{\mathbf{\Theta}}$.

$$\tilde{\mathbf{\Theta}}_{ii} = \frac{\lambda_i + \sqrt{\lambda_i^2 + 4\rho}}{2\rho}$$

We update $\mathbf{\Theta}^{k+1}$ by setting $\mathbf{\Theta}^{k+1} = \mathbf{Q}\tilde{\mathbf{\Theta}}\mathbf{Q}^T$. From (35), $\mathbf{Z}$ update is simplified to the soft thresholding operator F as follows.

$$\begin{aligned} \mathbf{Z}_{ij}^{k+1} &:= F_{\lambda/\rho}(\mathbf{\Theta}_{ij}^{k+1} + \mathbf{U}_{ij}^{k+1}) \\ \text{where} \quad F_\kappa(a) &= (a - \kappa)_+ - (-a - \kappa)_+ \end{aligned}$$

The ADMM was implemented using 'gglasso' Python library[2] (36). We computed eight logarithmically spaced γ values in the range $[10^{-1.5}, 10^{0.5}]$ (i.e., 'numpy.logspace(0.5, -1.5, 8)'). The best regularization parameter γ was selected using the extended Bayesian information criterion (eBIC) as suggested in (37). Other than γ, we used default parameters for both ADMM and eBIC computation unless specified otherwise.

[2] https://github.com/fabian-sp/GGLasso

Differential covariance. We implemented Differential covariance (Dcov) (38) to investigate the utility of directed temporal covariance in inferring synaptic connectivity. We represent the column vector $\mathbf{x}(t)$ that has the i-th entry equal to the z-transformed number of spikes or firing rate of the neuron i at time t. Dcov is defined as the covariance between the $\mathbf{x}(t)$ and the derivative.

$$\text{Dcov} = \langle \frac{d\mathbf{x}(t)}{dt}, \mathbf{x}(t) \rangle,$$

where $\langle , \rangle$ denotes time averaged vector outer product and differentiation is computed as $\frac{d\mathbf{x}(t)}{dt} = \frac{\mathbf{x}(t+1)-\mathbf{x}(t-1)}{2\Delta t}$ (38). The differentiation term in Dcov makes it a directed measure of connectivity.

Dynamic differential covariance. Dynamic differential covariance (DDC)(25) can be estimated as a matrix product between the Dcov and the precision matrix (i.e., the inverse of the covariance matrix).

$$\text{DDC} = \text{Dcov} \cdot \langle \mathbf{x}(t), \mathbf{x}(t) \rangle^{-1}$$

DDC improves upon Dcov by factoring out the influence of common input sources. For further proofs and characteristics of DDC, we refer readers to Chen et al. (25).

Fractional dynamic differential covariance. If a time series exhibits long decay in autocorrelation, taking fractional (non-integer) differentiation can preserve the lasting trend in the time series which would otherwise be lost for full integer differentiation (39). Fractional differentiation with a fractional order β, is defined as:

$$D^{\beta}\mathbf{x}(t) = \sum_{k=0}^{\infty} \omega_k \mathbf{x}(t-k)$$

where $\omega_0 = 1$, $\omega_1 = -\beta$, and $\omega_k = \frac{(-1)^k \prod_{i=0}^{k-1} \beta - i}{k!}$, for $k \geq 2$. If $\beta = 1$, we retrieve full-integer differentiation. In this study, we implemented up to 10-th order ($k = 10$) to compute fractional differentiation using Python library 'fdiff' [3]. By substituting the differentiation in DDC with fractional differentiation, we could compute the Fractional dynamic differential covariance (FDDC).

$$\text{FDDC} = \langle D^{\beta}\mathbf{x}(t), \mathbf{x}(t) \rangle \cdot \langle \mathbf{x}(t), \mathbf{x}(t) \rangle^{-1}$$

Spike train convolution. One strategy to increase the number of sample points from spike trains is to compute instantaneous rates (40). Instantaneous rates can be computed by convolving the spike trains using predefined kernels. We used two commonly used functions to generate instantaneous rates: alpha-function kernels (Alpha kernels) and Gaussian kernels. The Gaussian kernel has the form:

$$k_{\text{Gaussian}}(t) = \frac{1}{w\sqrt{2\pi}} \exp\left(-\frac{t^2}{2w^2}\right)$$

[3] https://fracdiff.github.io/fracdiff/

We denote a filter size (duration in ms) as w. As an asymmetric counterpart, we applied Alpha kernels as a complementary approach. The Alpha kernel is a decaying exponential function with a characteristic time constant (τ), which facilitates the modeling of postsynaptic currents (41). The kernel was defined as:

$$k_{\text{Alpha}}(t) = \begin{cases} (1/\tau^2)t \cdot \exp(-t/\tau), & \text{if} \quad t > 0 \\ 0, & \text{if} \quad t \leq 0 \end{cases}$$

We considered $\tau = w/\sqrt{2}$ for the consistency in notation with the Gaussian kernel. For both kernels, filter sizes (w) of 1, 3 ms were used to generate instantaneous rates. The filter size of 1 ms corresponded to the typical duration of an action potential (i.e., < 5 ms), and 3 ms corresponded to the timescale of longer synaptic delays (i.e., < 20 ms) in experimental studies (42, 43). We sampled points at a frequency of 1 kHz from these instantaneous rates.

Evaluation of inference performance. We assessed the performance of each connectivity inference method against the ground truth by calculating the Area Under the Receiver Operating Characteristic Curve (AUROC) and the Area Under the Precision-Recall Curve (AUPRC). The absolute values of computed inference scores (weights) were used to compute AUROC and AUPRC. Additionally, we investigated the impact of simulation duration on inference performance by testing 5, 10, 15, and 20 min of simulated activity. To determine whether multivariate methods benefited from an increase in the number of observed neurons, we sampled 500, 1000, and 1500 neurons for the analysis. We randomly sampled three times from each simulated network, so for one condition (e.g., 5 min simulation length, 500 sampled neurons), we reported the mean and standard deviation of AUROC/AUPRC values from 15 instances (i.e., five network simulations $\times$ three samples). We also compared wall-clock time to assess the scalability of the inference methods. While all multivariate methods were implemented in Python, CCG was implemented using C and Matlab adapted from English et al. (30)(see Section "Data and code availability"). Python implementation of CCG using a standard multiprocessing library ('map') was roughly five times slower than the values reported in this study. The computation time was measured excluding the computation for preprocessing (e.g., binning, kernel convolution of spike trains). For all inferences, we used computing instances with the same specification with 8 CPU cores (Intel(R) Xeon(R) Gold 6154 CPU @ 3.00GHz) and 64 GB of RAM.

Recording neural network activity in vitro.

High-density microelectrode array recordings. HD-MEA recordings used in this study were part of the dataset reported in a prior study (44). We performed recordings with commercially available 6-well high-density microelectrode array (HD-MEA) plates (MaxTwo system by Maxwell Biosystems, Zurich, Switzerland). Each well included a CMOS-based HD-MEA(2) featuring 26,400 electrodes arranged in a 120

x 220 electrode grid with a microelectrode center-to-center spacing (pitch) of 17.5 μm. The overall sensing area of this HD-MEA was 3.85 × 2.10 mm^2. The HD-MEA enabled the simultaneous recording of up to 1020 electrodes at a sampling rate of 10 kHz. Recordings were performed in an incubator chamber of the MaxTwo system at 37°C and 5% CO_2/ 95% O_2 and were conducted at days in vitro (DIVs) 22–25. Each recording started with a whole-array 'activity scan' to determine the active electrodes on the HD-MEA. The activity scan consisted of 29 dense electrode configurations to scan through the entire sensing area of the electrode array; each configuration was sequentially recorded for 60 s. From the activity scan, up to 1020 electrodes were selected by prioritizing electrodes with high firing rates. Using the selected electrodes, we recorded two hours of spontaneous activity from 22 HD-MEA wells or neuronal cultures.

Hippocampal dissociated neuronal culture. Rat primary neurons were acquired from the dissociated hippocampus of Wistar rats on embryonic day (E) 18, using the protocol described in (45). All animal experiments were conducted in accordance with the approved guidelines and Swiss federal laws on animal welfare, with approval from the Basel-Stadt veterinary office. HD-MEA wells were sterilized in 70% ethanol for 30 min prior to cell plating. Afterwards, the ethanol was removed, and the wells were rinsed three times with sterile distilled water before being left to dry. The HD-MEA wells were then coated with a layer of 0.05% polyethylenimine (Sigma-Aldrich, Buchs, Switzerland) in a borate buffer (Thermo Fisher Scientific, Waltham, MA, United States) to render the surface more hydrophilic. Then, a thin layer of laminin (Sigma-Aldrich, 0.02 mg/mL) in Neurobasal medium (Gibco, Thermo Fisher Scientific) was applied to the array and incubated for 30 min at 37°C to promote cell adhesion. We dissociated hippocampi of E18 Wistar rat in trypsin with 0.25% EDTA (Gibco), followed by trituration. Cell suspensions of $15,000$ cells in 7 μL were then plated on top of the electrode arrays. The plated wells were incubated at 37°C for 30 min before adding 2 mL of the plating medium. The plating medium consisted of Neurobasal, supplemented with 10% horse serum (HyClone, Thermo Fisher Scientific), 0.5 mM Glutamax (Invitrogen, Thermo Fisher Scientific), and 2% B-27 (Invitrogen). After 5 days, 50% of the plating medium was replaced with a growth medium, containing Brainphys medium supplemented with SM1 and N2-A (Stemcell technologies, Cologne, Germany). For the rest of the experiments, medium changes were performed twice a week using the same Brainphys-based medium. The wells were kept inside a humidified incubator at 37°C and 5% CO_2/ 95% O_2.

Spike-sorting and quality control of sorted units. For each HD-MEA/well, the recordings were filtered, and spike-sorted using 'Kilosort2' (46); the applied parameters are stated in Table S2. To be included in subsequent analyses, all inferred spike-sorted units had to pass a quality control: First, we removed units with a firing rate below 0.05 Hz and higher than 30 Hz. Then we computed the refractory period violation ratio, which was calculated as the fraction of interspike intervals (ISIs) less than 2 ms (47). In practice, we quantified the number of spikes within the [$\pm$2 ms] bins of the spike train autocorrelogram (ACG) and then computed the fraction between this count and the total number of spikes in a larger range of the ACG [$\pm$50 ms]. Any template exceeding a refractory period violation ratio of 0.3 was removed. Based on these preprocessing steps, the obtained units were considered to originate from single neurons.

Characterizing the spontaneous activity of in vitro hippocampal networks.

Participation ratio. We used the Participation Ratio (PR) to measure the correlated firing activity in each network recording, normalized by the number of neurons. The calculation and interpretation of the PR were based on the implementation of Recanatesi et al.(48). First, the spike trains of the baseline recordings were binned using a window size of 5 ms. The resulting binned spike trains were z-transformed and used to compute inner products, generating a correlation matrix. From the eigendecomposition of the correlation matrix, we collected the eigenvalues to assess the level of correlated activity between neurons. The participation ratio was defined as

$$PR = \frac{(\sum_i \lambda_i)^2}{\sum_i \lambda_i^2},$$

where λ_i is i-th eigenvalue of the correlation matrix. The resulting PR value indicates the number of principal components that were necessary to explain 80% to 90% of the total variance for typical Principal component analysis (PCA) eigenspectra (49). We then normalized the PR, by dividing it by the number of neurons (N) in the network.

$$\frac{1}{N} \leq \text{PR normalized} \leq 1$$

A normalized PR value of < 0.8, suggests that the majority of the variance in the network activity could be explained by less than 80% of principal components (48–50).

Small-worldness of inferred networks. The small-worldness of a network measures the extent to which a network exhibits both high local clustering and short average path lengths between nodes (51). To conclude whether a network shows small-worldness, Downes et al. (52) defined Small-World Index (SWI) as a ratio between clustering coefficient (CC) and characteristic path length (L_{path}). CC was defined as the fraction of possible triangles in the network that are present, while the L_{path} was the average number of steps along the shortest paths for all possible pairs of network nodes (51). In practice, CC and L_{path} need to be normalized against the expected values computed from random networks with the same number of nodes and edges as the original graph (29). If SWI > 1, the network is considered to show small-worldness. To investigate the differences in the inferred network structure from in vitro recordings, we inferred connectivity from the recordings using both FDDC and CCG. For each case, we applied thresholds to generate multiple binary, directed

graphs with density (ρ) of 5, 10, 15, and 20%. CC and the L_{path} were then computed from each thresholded graph, and lattice surrogates ($n = 30$) were generated by rewiring the edges in the thresholded graphs while preserving the distribution of in-degree and out-degree across nodes (53). From lattice surrogates, the average CC and L_{path} were calculated. Then SWI was computed as the ratio between the normalized CC and L_{path}.

$$\mathrm{SWI} = \frac{\mathrm{CC}/\mathrm{CC}_{\mathrm{rand}}}{\mathrm{L}_{\mathrm{path}}/\mathrm{L}_{\mathrm{rand}}},$$

where CC_{rand} and L_{rand} were the average clustering coefficient and characteristic path length of the lattice surrogates. The generation of lattice surrogate networks and subsequent computation of SWI was implemented using the Python library 'bctpy' [4].

Measurement of autocorrelation in convolved spike trains. To assess the temporal dependency present in a sequence of convolved spike trains, we computed the normalized autocorrelation.

$$\rho_k = \frac{\sum_{t=k+1}^{T}(y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{T}(y_t - \bar{y})^2},$$

where y_t is the value of the convolved spike train at time t, and $\bar{y}$ is the mean of the convolved spike train. T is the total number of observations and k is the lag. In this study, T was set to 20 min with the bin size of 1 ms, and k was computed up to 20 (20 ms). After computing ρ_k, we normalized the resulting values by dividing them by the maximum value ρ_0 ($k = 0$). This ensured that the normalized autocorrelation values ranged from -1 to 1. In the case of simulated networks, we randomly sampled 500 neurons and computed the autocorrelation only for those neurons that had a firing rate higher than 0.1 Hz. For the experimental data, we also computed the autocorrelation only for neurons that had a firing rate higher than 0.1 Hz. We characterized the degree of structure in the spike trains by computing the average area under the curve (AUC) of the autocorrelation curves. These AUC values were compared with the AUC values of randomized surrogates, which shuffled spike times randomly while keeping the total number of spikes.

Results

In this section, we first characterize the simulated networks and pinpoint the condition under which the pairwise inference approach falls short. After identifying the failure mode, we assess the performance of the multivariate inference methods. Then, we build up on the most promising method and apply a new inference approach, FDDC. Lastly, we investigate the FDDC further by applying it to HD-MEA network recordings from in vitro neuronal cultures and compare the inferred FDDC-derived connectivity graphs with those derived with the CCG method.

[4] https://github.com/aestrivex/bctpy

Simulated networks with asynchronous and synchronous network activity. The simulated asynchronous and synchronous networks showed clear differences in their average population firing rates and in both measures of network synchrony (MPC, CV_{ISI}). The asynchronous networks showed average population firing rates of 31 Hz (standard deviation 0.7 Hz), and the synchronous networks exhibited average population firing rates of 20 Hz (1.3 Hz). The average CV_{ISI} for the asynchronous networks was 1.7 (0.04), and 1.1 (0.11) for the synchronous networks. Moreover, the asynchronous networks showed on average lower MPC values (0.02 (0.002)) compared to the MPC values obtained for the synchronous networks (0.24 (0.06); Fig. 1).

Pairwise inference using cross-correlograms fails to recover connectivity of simulated synchronous network activity. Upon examining the inference methods on both asynchronous and synchronous simulated networks, we discovered that the CCG performed exceptionally well for the asynchronous networks while only attaining performance close to random success for the synchronous networks (i.e., AUROC $\approx$ 0.5, AUPRC $\approx$ 0.02). An overview of the performance across all methods is provided in Table 1 (For more details, see also Table S3 and Fig. 2A). Although none of the multivariate methods outperformed CCG for the asynchronous networks, DDC showed the best performance for the synchronous networks (Table 1, S4, S5). Moreover, as the number of observed neurons (n) increased, the average AUROC and AUPRC scores for DDC also showed improvement (Fig. 2B). In contrast, Glasso demonstrated inference performance that was close to random estimations, and ND, Dcov showed marginally better performance compared to Glasso (Table S4, S5, Fig. S3, S4). We additionally evaluated the AUROC and AUPRC values for undirected methods, such as ND and Glasso, by transforming the directed ground truth connectivity into undirected connections. The resulting scores remained consistently low for both methods (Table S6).

Improving the inference performance of Dynamic differential covariance using spike train convolution and fractional differentiation. We further explored the potential of using DDC to infer connections in synchronous networks by increasing the number of data points through computing instantaneous rates. Our findings showed that DDC's ability to infer connections was enhanced when instantaneous rates were used, with all cases reporting higher average AUPRC values (Fig. 3A, Table 2, S7, S8). Among the four kernels that were used to convolve the spike trains, the Gaussian kernel (w =3 ms) consistently showed the highest average AUROC and AUPRC scores.

Our focus then turned to a limitation of the differentiation used in DDC. We noted that the spike trains from simulated network activities and in vitro recordings were non-random processes (e.g., refractory periods). Therefore, we considered fractional differentiation as a means to capture lasting trends in the time series that would be overlooked with full-integer differentiation. Our analysis confirmed that both the simu-

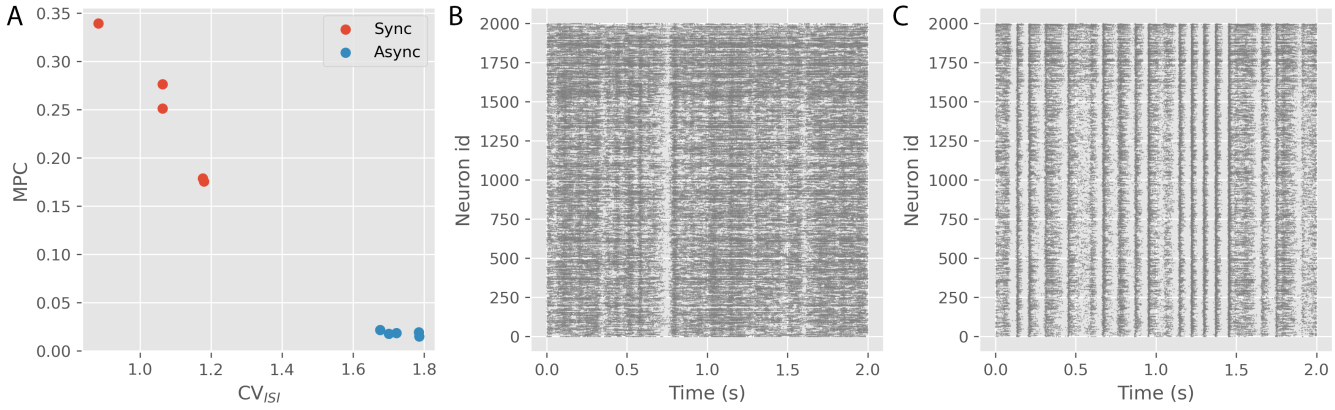

Fig. 1. LIF network simulations. (A): There was a clear difference in the measures of network synchrony (mean pairwise correlation (MPC), CV_{ISI}) between the two network simulation types. Asynchronous networks showed average population firing rates of 31 Hz (standard deviation 0.7 Hz), while synchronous networks exhibited average population firing rates of 20 Hz (1.3 Hz). (B): A spike raster plot showing 2 seconds of the network firing activity from an asynchronous network. (C): A raster plot showing 2 seconds of network firing activity from a synchronous network. In contrast to the asynchronous network (depicted in Panel B), a more prominent synchronized firing pattern is observed.

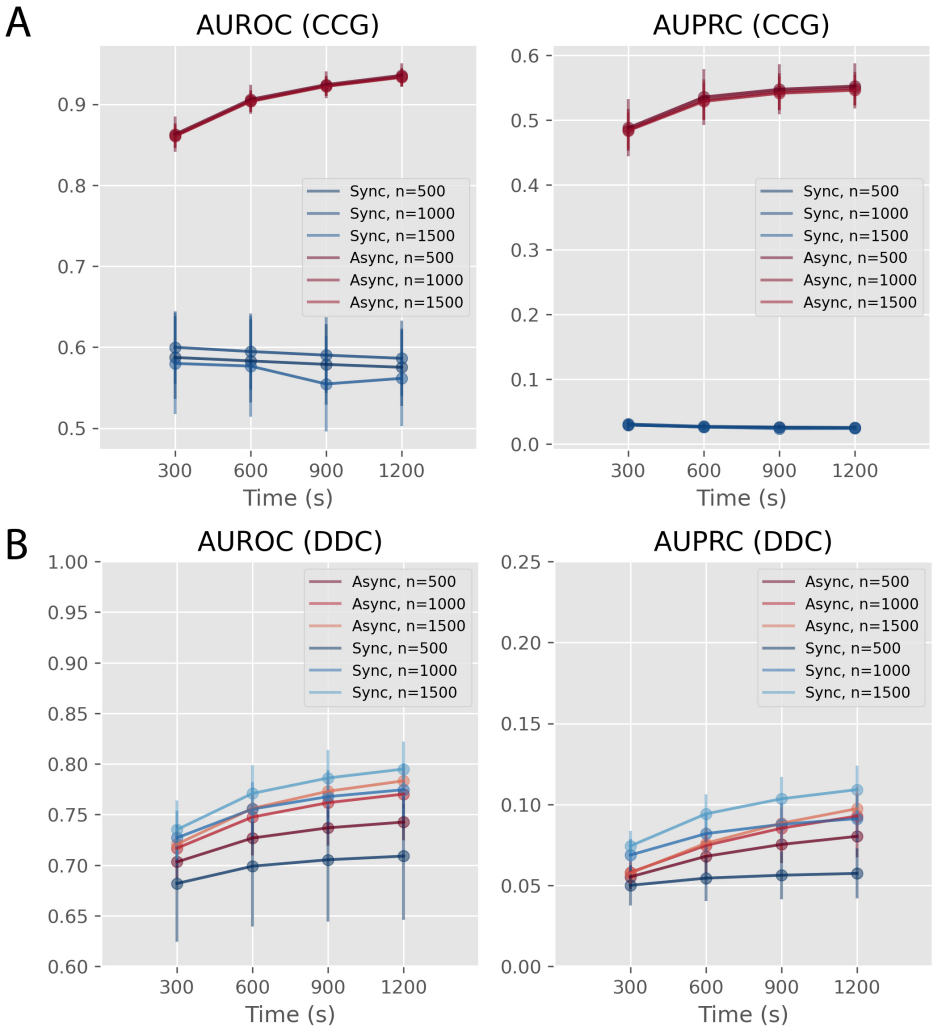

Fig. 2. Inference performances of CCG and DDC. The graphs in panels (A) and (B) display average AUROC (left) and AUPRC (right) values (mean $\pm$ standard deviation) across different sampling conditions (number of sampled neurons (n), simulation length used). (A): CCG demonstrated high AUROC and AUPRC values for the asynchronous networks but lower AUROC and AUPRC values in the case of synchronous networks. (B): DDC showed lower average AUROC and AUPRC for the asynchronous networks but showed higher average AUROC and AUPRC values for the synchronous networks compared to CCG.

Network act.	Method	AUROC	AUPRC	Comp. time (s) (averaged)
Async	**CCG**	**0.93 ± 1.1e-2**	**0.55 ± 2.4e-2**	**3300**
	ND	0.63 ± 5.4e-3	0.06 ± 2.4e-3	9
	Glasso	0.51 ± 1.0e-2	0.02 ± 2.7e-4	12
	Dcov	0.61 ± 1.3e-2	0.06 ± 6.2e-3	39
	DDC	0.78 ± 1.4e-2	0.10 ± 1.4e-2	40
Sync	CCG	0.56 ± 5.9e-2	0.02 ± 4.2e-3	2000
	ND	0.58 ± 2.1e-2	0.04 ± 2.4e-2	6
	Glasso	0.51 ± 6.0e-4	0.02 ± 1.7e-4	82
	Dcov	0.51 ± 2.4e-3	0.02 ± 2.5e-4	35
	DDC	**0.79 ± 2.7e-2**	**0.11 ± 1.5e-2**	**35**

Table 1. Comparison of inference performance. The performance of each inference method for the simulated network activities, based on 1500 sampled neurons and 20 min of simulation, is presented in the table. The average values rounded for two significant digits are reported for the computation time. For the asynchronous networks, CCG showed the best performance with the highest AUROC and AUPRC, but failed to recover the ground truth connections for the synchronous networks. DDC showed the best performance for the synchronous networks in addition to faster computation.

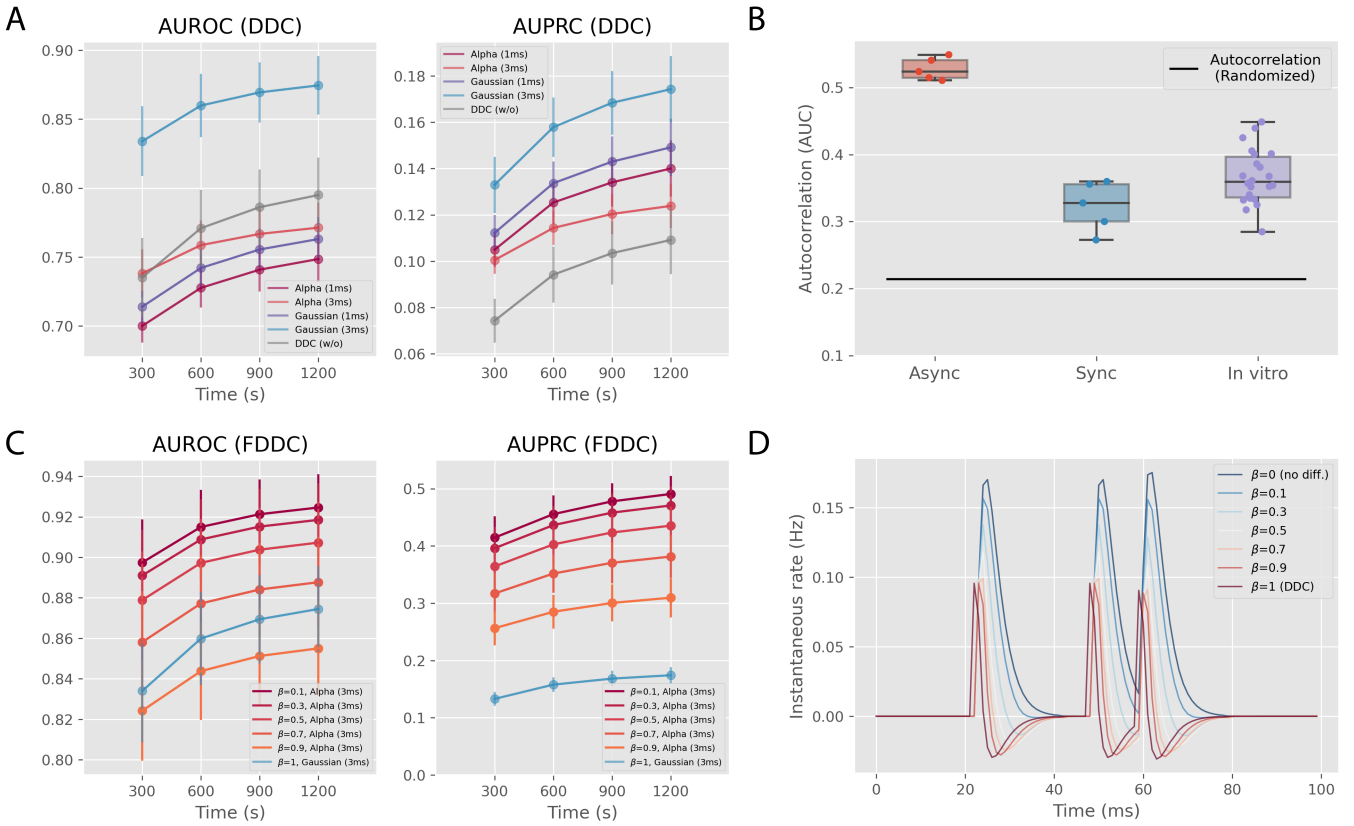

Fig. 3. Enhancing inference performance of DDC for synchronous networks (n=1500) (A): AUROC (left) and AUPRC (right) values for the four kernel variants are shown. Gaussian kernel ($w = 3$ ms) consistently showed the best performance. (B): The plot compares the average AUC values of autocorrelation curves for 10 simulated networks (5 asynchronous and 5 synchronous) and 22 in vitro recordings against the average value of random surrogates (standard deviation (σ) < 10^{-5}). For all networks, the average AUC values of autocorrelation curves were larger than those of random surrogates. (C): The performances of FDDC on the synchronous networks for different fractional orders (β) are presented (AUROC (left), AUPRC (right)). The results showed that as smaller fractional derivative orders were applied, the average AUROC and AUPRC values increased. The best-performing method from panel A (Gaussian kernel ($w = 3$ ms), colored in cyan) is also plotted for comparison. (D): The plot illustrates how fractional differentiation can preserve the original time series ($\beta = 0$, no diff.) compared to the full integer differentiation ($\beta = 1$, the same as DDC). As the β increases, the differentiated time series diverges from the original time series.

lated and experimental spike trains displayed stronger autocorrelation than the randomized surrogate spike trains (Fig. 3B), indicating the presence of non-random temporal structures in these time series. Compared to DDC using the Gaussian kernel (w =3 ms), a consistent improvement in the average AUPRC values was observed for all kernels when fractional differentiation was used (see Table 2, S9, S10). Among these kernels, the Alpha kernel (w =3 ms)(see method section "Spike train convolution") resulted in the best performance (Fig. 3C, Table 2). We also observed that smaller fractional orders (β) resulted in enhanced inference performances, owing to better preservation of the original time series with smaller β (Fig. 3C, D).

Throughout the study, FDDC (β =0.1) combined with Alpha

Network act.	Method	Kernel	Filter size (w, ms)	AUROC	AUPRC	Comp. time (s) (averaged)
Async	DDC	Alpha	1	0.74 ± 0.01	0.13 ± 0.02	290
			3	0.79 ± 0.01	0.11 ± 0.02	300
		Gaussian	1	0.75 ± 0.01	0.13 ± 0.02	300
			3	0.85 ± 0.02	0.14 ± 0.02	310
	FDDC (β=0.1)	Alpha	1	0.79 ± 0.01	0.17 ± 0.02	380
			3	$\mathbf{0.87 \pm 0.01}$	$\mathbf{0.31 \pm 0.04}$	**380**
		Gaussian	1	0.80 ± 0.01	0.18 ± 0.03	380
			3	0.86 ± 0.01	0.24 ± 0.03	380
Sync	DDC	Alpha	1	0.75 ± 0.02	0.14 ± 0.01	320
			3	0.77 ± 0.02	0.12 ± 0.01	290
		Gaussian	1	0.76 ± 0.02	0.15 ± 0.01	290
			3	0.87 ± 0.02	0.17 ± 0.01	290
	FDDC (β=0.1)	Alpha	1	0.81 ± 0.02	0.28 ± 0.02	380
			3	$\mathbf{0.92 \pm 0.02}$	$\mathbf{0.49 \pm 0.03}$	**380**
		Gaussian	1	0.83 ± 0.02	0.30 ± 0.02	380
			3	0.91 ± 0.01	0.39 ± 0.02	380

Table 2. Performances of modified DDC methods on convolved spike trains. The table shows the performance (AUROC, AUPRC) of modified DDC methods on simulated network activities (1500 sampled neurons and 20 min of simulation). FDDC ($\beta = 0.1$) with Alpha kernel ($w = 3$ ms) showed the best performance among all combinations (β, w) tested in this study. Among the DDC methods utilizing full-integer differentiation, the Gaussian kernel ($w = 3$ ms) emerged as the most effective yet showed lower average AUROC, AUPRC values than the best FDDC method.

kernel (w =3 ms) showed the best performance among all methods that were assessed for the synchronous networks. To test the robustness of this FDDC approach, we probed its performance on the noise-perturbed simulations. We subjected the networks to a broad spectrum of noise levels. At the higher end of the noise range, where the noise level was larger than the leaky current (≈0.2 nA), we observed a recovery of the ground truth connectivity induced by the noise (Table 3). In the lower noise range, we observed consistent performances for both asynchronous and synchronous networks. Regarding average AUROC values, the noise-perturbed asynchronous and synchronous networks showed a decline of approximately 7% and 20%. Similarly, the average AUPRC values of the noise-perturbed asynchronous and synchronous networks displayed a reduction of approximately 32% and 73%, respectively.

Network connectivity and topology of in vitro neuronal networks differ between FDDC and CCG-derived connectivity graphs. To probe our ability to infer ground truth connectivity using FDDC from synchronous network activity, we inferred connectivity from in vitro hippocampal neuronal cultures using CCG and the best FDDC method. We focused on comparing the resulting graph topology between the two inferred graphs by quantifying the small-world index (SWI) and intersecting edges between them. When comparing the inferred graphs with varying graph densities (e.g., $\rho = 5$, 10, 15, 20 %), CCG-derived graphs generally showed higher SWIs than FDDC-derived graphs (Fig. 4A). For all graph densities, the graphs obtained through CCG consistently exhibited small-worldness with SWI > 1, whereas a few of the FDDC-derived graphs displayed SWI < 1. Next, we investigated whether there was a relationship between synchronized network firing activities and graph topology by analyzing the correlation between the normalized participation ratio (PR) and SWI. For the graphs derived from FDDC, we found a significant negative linear correlation between the PR and SWI (ordinary least squares, two-sided t-test, p-value < 0.01) across all tested graph densities. This finding shows that the more the connections clustered between the nodes, the more synchronous was the population firing activity of the networks. However, we did not observe any linear correlation for CCG-derived graphs for all tested graph densities (Fig. 4B,C). Finally, we calculated the number of intersecting edges across all graph densities and found that there were always less than 40% of edges that intersected (Fig. 4D-H).

Discussion

In the first part of our study, we compared the performance of selected multivariate inference methods for inferring neuronal connectivity in networks with asynchronous and synchronous activity obtained from LIF simulations. Our findings indicate that when computation time is not a primary concern, the multivariate methods were consistently less effective than CCG for asynchronous networks. Furthermore, we observed subpar performance by ND, Dcov, and Glasso in both asynchronous and synchronous networks. The poor performance of ND may stem from its inability to explicitly consider negative edge weights for the underlying connectivity graph (32). Similar approaches based on graph signal propagation assume undirected and non-negative edges for connectivity graphs (54–57). This assumption may be appropriate for analyzing brain connectivity in data acquired from techniques such as functional magnetic resonance imaging (fMRI), electroencephalography (EEG) and magnetoen-

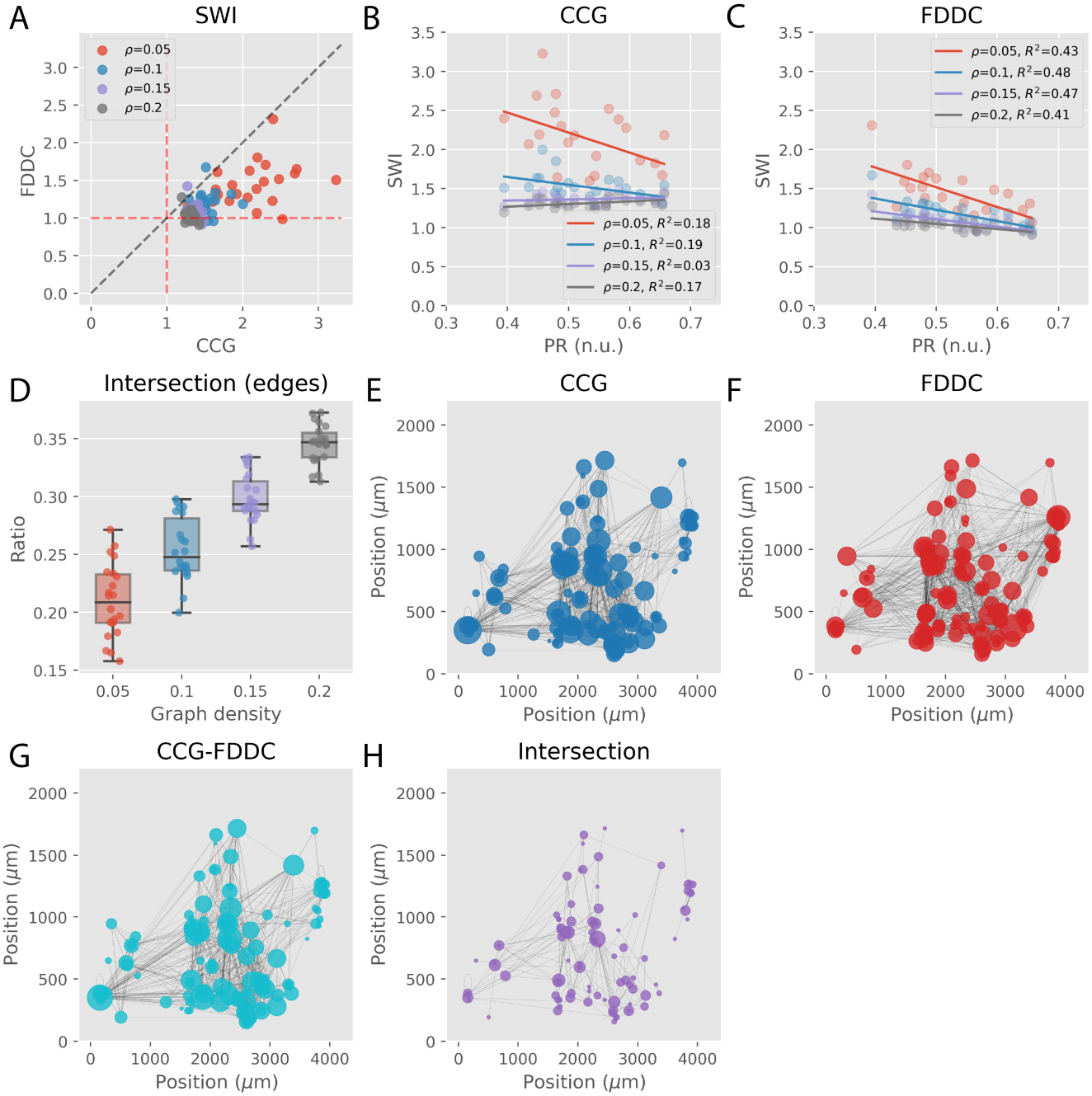

Fig. 4. Comparing connectivity graphs of in vitro neuronal networks inferred using CCG and FDDC. (A): The CCG-derived graphs exhibited a small-worldness index (SWI) with values greater than 1 for all the tested graph densities. On the other hand, some of the FDDC-derived graphs had SWI values smaller than 1. In general, the SWI values were higher for CCG-derived graphs compared to FDDC-derived graphs. (B): There was no significant linear correlation between PR and SWI for CCG-derived graphs at any graph density. (C): There was a negative linear correlation between PR and SWI for FDDC-derived graphs across all the graph densities. (D): The ratios of intersecting edges to all existing edges are presented. None of the in vitro neural networks shared more than 40% of edges for all tested graph densities. (E): The panel depicts an example CCG-derived graph, where the size of each node represents the total degree (sum of in-degree and out-degree) of the respective nodes. (F): Panel F depicts the corresponding FDDC-derived graph for the graph shown in (E). (G): Panel G depicts the difference between the CCG-derived graph (panel E) and the FDDC-derived graph (panel F), i.e., only the edges that are exclusive to the CCG-derived graph. (H): Intersection between the CCG-derived graph (panel E) and FDDC-derived graph (panel F) is presented.

Network act.	Noise (σ, nA)	AUROC	AUPRC	Comp. time (s) (averaged)
Async	0	0.87 ± 0.01	0.31 ± 0.04	380
	0.001	0.81 ± 0.01	0.21 ± 0.03	380
	0.005	0.81 ± 0.01	0.21 ± 0.03	390
	0.01	0.81 ± 0.01	0.21 ± 0.03	390
	0.05	0.81 ± 0.01	0.21 ± 0.03	390
	0.1	0.81 ± 0.01	0.21 ± 0.03	390
	0.5	0.82 ± 0.01	0.21 ± 0.03	420
	1	0.82 ± 0.01	0.21 ± 0.04	410
Sync	0	0.92 ± 0.02	0.49 ± 0.03	380
	0.001	0.74 ± 0.04	0.13 ± 0.03	410
	0.005	0.74 ± 0.04	0.13 ± 0.03	420
	0.01	0.74 ± 0.04	0.13 ± 0.03	420
	0.05	0.75 ± 0.04	0.13 ± 0.03	410
	0.1	0.75 ± 0.04	0.13 ± 0.03	390
	0.5	0.83 ± 0.05	0.22 ± 0.06	390
	1	0.91 ± 0.01	0.36 ± 0.03	380

Table 3. Performances of FDDC ($\beta = 0.1$, Alpha Kernel ($w = 3$ ms)) on the noise-perturbed simulations. The table shows the performance of FDDC (β =0.1) on noise-perturbed simulated networks (1500 sampled neurons and 20 min of simulation). When the noise level was smaller than the leaky current (≈0.2nA), the average AUROC and AUPRC values remained stable. However, in the higher noise range, both values increased, indicating a noise-induced recovery of the ground truth connectivity.

cephalography (MEG) (58–60). However, the assumption may not accurately reflect inhibitory connections in neuronal cultures. Moreover, Dcov and Glasso, which are both conceptual components of DDC, performed poorly, while DDC emerged as the most effective method on synchronous networks. The findings suggest that accurate recovery of the ground-truth connectivity in synchronous networks requires a combination of directed information obtained through differentiation of the original time series (Dcov) and consideration of other neurons' impact (Glasso).

We also found that using instantaneous rates to infer connectivity improved the inference performance of DDC. It is worth noting that the idea of convolving spike trains with kernels has been previously proposed by Chen et al. (25), and our study provides quantitative support for this approach. In our experiments, we found that when using full-integer differentiation, the best inference performance was obtained with the Gaussian kernel ($w = 3$ ms). However, when fractional differentiation was employed, the Alpha kernel ($w = 3$ ms) yielded the best performance. This observation suggests that fractional differentiation can preserve useful temporal information that was uniquely captured by an asymmetric kernel. The use of fractional differentiation with a lower fractional order (β) improved the inference performance. It should be noted that β can be even smaller than the values examined in this study but must be strictly greater than 0; otherwise, we retrieve an identity matrix. In this study, our focus was to demonstrate that FDDC with any β outperformed DDC with full-integer differentiation when using instantaneous rates generated with the Alpha kernel ($w = 3$ ms). In practice, we suggest determining the smallest β by conducting statistical tests such as the Dickey–Fuller test (61) to ensure stationarity in the differentiated time series. On the other hand, exploring the best-performing FDDC method ($\beta = 0.1$, Alpha kernel ($w = 3$ ms)) on noise-perturbed networks revealed that the inference using FDDC on synchronous networks was more vulnerable compared to the case of asynchronous networks. This finding suggests FDDC is most effective for networks with robust and consistent synchronized firing patterns.

We compared the inferred connectivity graphs obtained from in vitro neural networks by applying both CCG and FDDC to assess the relevance of FDDC to experimental data. As small-worldness has been previously reported as a characteristic feature of inferred connectivity based on cross-covariance for neuronal cultures (62, 63), we compared the SWI resulting from CCG- and FDDC-derived graphs. In this study, cross-covariance reduces to CCG, as we assumed the original time series were stationary (29), given that FDDC showed improved inference performance over DDC for the tested fractional orders (64). The connectivity graphs obtained using CCG indeed exhibited small-worldness across all tested graph densities. However, some of the connectivity graphs derived using FDDC did not show small-worldness with SWIs smaller than 1, and most of the FDDC-derived graphs showed smaller SWIs compared to CCG-derived graphs. Previous theoretical studies on network dynamics have indicated that networks with clustered connections are more prone to show synchronization (51, 65–67). Our results showed that this relation was robust for FDDC-derived graphs, while no clear correlation was observed for the CCG-derived graphs. We speculate that FDDC-derived graphs generated more conservative connectivity that reflected the synaptic connectivity responsible for the synchronized firing activity.

In conclusion, we propose that the combination of fractional differentiation and DDC presents a promising solution to address the scalability challenges associated with synchronous networks containing thousands of neurons. However, we ac-

knowledge that even the proposed FDDC method would not scale for networks with a much larger number of neurons. The most computationally expensive step in FDDC is the inversion of a large covariance matrix. Therefore, we recommend further exploring methods such as Cholesky decomposition (68) to compute the inverse efficiently or to solve equations directly using proximal gradient methods (69, 70). By implementing these modifications, we believe that derivative inference methods could achieve scalability beyond thousands of neurons.

Conflict of Interest Statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author Contributions

TK, DC, PH, SK contributed to the conception and design of the study. AH, DC, MS, and KB provided scientific supervision to TK. AH and KB provided the necessary resources and materials for the current study. TK performed the experiment and analysis. TK wrote the first draft of the manuscript. All authors contributed to the manuscript revision and approved the submitted version.

Funding

This work was supported by the European Research Council Advanced Grant 694829 'neuroXscales' and a Swiss Data Science Center project grant (C18-10).

ACKNOWLEDGEMENTS

We thank Maxwell Biosystems, Zurich, Switzerland, for providing the recording hardware and software support. Dr. Julian Bartram and Xiaohan Xue, all at ETH Zurich, are acknowledged for engaging discussion and feedback.

Data and code availability

The scripts used in this study are available from the following GitHub repository: https://github.com/arahangua/FDDC.

Bibliography

1. Ildefons Magrans de Abril, Junichiro Yoshimoto, and Kenji Doya. Connectivity inference from neural recording data: Challenges, mathematical bases and research directions, 2018.
2. Jan Müller, Marco Ballini, Paolo Livi, Yihui Chen, Milos Radivojevic, Amir Shadmani, Vijay Viswam, Ian L Jones, Michele Fiscella, Roland Diggelmann, Alexander Stettler, Urs Frey, Douglas J Bakkum, and Andreas Hierlemann. High-resolution CMOS MEA platform to study neurons at subcellular, cellular, and network levels. *Lab Chip*, 15(13):2767–2780, July 2015.
3. Jelena Dragas, Vijay Viswam, Amir Shadmani, Yihui Chen, Raziyeh Bounik, Alexander Stettler, Milos Radivojevic, Sydney Geissler, Marie Obien, Jan Müller, and Andreas Hierlemann. A Multi-Functional microelectrode array featuring 59760 electrodes, 2048 electrophysiology channels, stimulation, impedance measurement and neurotransmitter detection channels. *IEEE J. Solid-State Circuits*, 52(6):1576–1590, June 2017.
4. Xinyue Yuan, Andreas Hierlemann, and Urs Frey. Extracellular recording of entire neural networks using a Dual-Mode microelectrode array with 19584 electrodes and high SNR. *IEEE J. Solid-State Circuits*, 56(8):2466–2475, August 2021.
5. Alessandro Maccione, Matteo Garofalo, Thierry Nieus, Mariateresa Tedesco, Luca Berdondini, and Sergio Martinoia. Multiscale functional connectivity estimation on low-density neuronal cultures recorded by high-density CMOS micro electrode arrays. *J. Neurosci. Methods*, 207(2):161–171, June 2012.
6. Jonathon Shlens, Greg D Field, Jeffrey L Gauthier, Matthew I Grivich, Dumitru Petrusca, Alexander Sher, Alan M Litke, and E J Chichilnisky. The structure of multi-neuron firing patterns in primate retina. *J. Neurosci.*, 26(32):8254–8266, August 2006.
7. Abhranil Das and Ila R Fiete. Systematic errors in connectivity inferred from activity in strongly recurrent networks. *Nat. Neurosci.*, 23(10):1286–1296, October 2020.
8. Elad Ganmor, Ronen Segev, and Elad Schneidman. Sparse low-order interaction network underlies a highly correlated and learnable neural population code. *Proc. Natl. Acad. Sci. U. S. A.*, 108(23):9679–9684, 2011.
9. Murat Okatan, Matthew A Wilson, and Emery N Brown. Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. *Neural Comput.*, 17 (9):1927–1961, September 2005.
10. Ellese Cotterill, Paul Charlesworth, Christopher W Thomas, Ole Paulsen, and Stephen J Eglen. A comparison of computational methods for detecting bursts in neuronal spike trains and their application to human stem cell-derived neuronal networks. *J. Neurophysiol.*, 116 (2):306–321, August 2016.
11. Nicholas Timme, Wesley Alford, Benjamin Flecker, and John M Beggs. Synergy, redundancy, and multivariate information measures: An experimentalist's perspective. *J. Comput. Neurosci.*, 36(2):119–140, 2014.

12. Patricia Wollstadt, Joseph Lizier, Raul Vicente, Conor Finn, Mario Martinez-Zarzuela, Pedro Mediano, Leonardo Novelli, and Michael Wibral. IDTxl: The information dynamics toolkit xl: a python package for the efficient analysis of multivariate information dynamics in networks. *J. Open Source Softw.*, 4(34):1081, February 2019.
13. Cody Baker, Emmanouil Froudarakis, Dimitri Yatsenko, Andreas S Tolias, and Robert Rosenbaum. Inference of synaptic connectivity and external variability in neural microcircuits. *J. Comput. Neurosci.*, 48(2):123–147, May 2020.
14. Minjie Wang and Genevera I Allen. Thresholded graphical lasso adjusts for latent variables. *Biometrika*, page asac060, November 2022.
15. Hang Yin, Xinyue Liu, and Xiangnan Kong. Gaussian mixture graphical lasso with application to edge detection in brain networks. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1430–1435, December 2020.
16. Hang Yin, Xiangnan Kong, and Xinyue Liu. Coherent graphical lasso for brain network discovery. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1392–1397, November 2018.
17. Ricardo Pio Monti, Peter Hellyer, David Sharp, Robert Leech, Christoforos Anagnostopoulos, and Giovanni Montana. Estimating time-varying brain connectivity networks from functional MRI time series. *Neuroimage*, 103:427–443, December 2014.
18. Harsh Shrivastava, Xinshi Chen, Binghong Chen, Guanghui Lan, Srinivas Aluru, Han Liu, and Le Song. GLAD: Learning sparse graph recovery. In *Eighth International Conference on Learning Representations*, April 2020.
19. Yunan Zhu and Ivor Cribben. Sparse graphical models for functional connectivity networks: Best methods and the autocorrelation issue. *Brain Connect.*, 8(3):139–165, April 2018.
20. Saverio Ranciati, Alberto Roverato, and Alessandra Luati. Fused graphical lasso for brain networks with symmetries. *J. R. Stat. Soc. Ser. C Appl. Stat.*, 70(5):1299–1322, November 2021.
21. Dimitri Yatsenko, Krešimir Josić, Alexander S Ecker, Emmanouil Froudarakis, R James Cotton, and Andreas S Tolias. Improved estimation and interpretation of correlations in neural circuits. *PLoS Comput. Biol.*, 11(3):1–28, 2015.
22. Giuseppe Vinci, Valérie Ventura, Matthew A Smith, and Robert E Kass. APPLICATION TO MULTIPLE-NEURON SPIKE COUNT DATA. *Ann. Appl. Stat.*, 12(2):1068–1095, 2018.
23. Daniela M Witten, Jerome H Friedman, and Noah Simon. New insights and faster computations for the graphical lasso. *J. Comput. Graph. Stat.*, 20(4):892–900, 2011.
24. Alain Destexhe. Self-sustained asynchronous irregular states and Up-Down states in thalamic, cortical and thalamocortical networks of nonlinear integrate-and-fire neurons. *J. Comput. Neurosci.*, 27(3):493–506, December 2009.
25. Yusi Chen, Burke Q Rosen, and Terrence J Sejnowski. Dynamical differential covariance recovers directional network structure in multiscale neural systems. *Proc. Natl. Acad. Sci. U. S. A.*, 119(24):e2117234119, June 2022.
26. Romain Brette and Wulfram Gerstner. Adaptive exponential Integrate-and-Fire model as an effective description of neuronal activity. *J. Neurophysiol.*, 94(5):3637–3642, November 2005.
27. Eduarda Susin and Alain Destexhe. Integration, coincidence detection and resonance in networks of spiking neurons expressing gamma oscillations and asynchronous states. *PLoS Comput. Biol.*, 17(9):e1009416, September 2021.
28. Jennifer S Goldman, Lionel Kusch, Bahar Hazal Yalçinkaya, Damien Depannemaecker, Trang-Anh E Nghiem, Viktor Jirsa, and Alain Destexhe. A comprehensive neural simulation of slow-wave sleep and highly responsive wakefulness dynamics. *bioRxiv*, page 2021.08.31.458365, September 2021.
29. Daniele Poli, Vito P Pastore, and Paolo Massobrio. Functional connectivity in in vitro neuronal assemblies. *Front. Neural Circuits*, 9:57, October 2015.
30. Daniel Fine English, Sam McKenzie, Talfan Evans, Kanghwan Kim, Euisik Yoon, and György Buzsáki. Pyramidal Cell-Interneuron circuit architecture and dynamics in hippocampal networks. *Neuron*, 96(2):505–520.e7, October 2017.
31. Eran Stark and Moshe Abeles. Unbiased estimation of precise temporal correlations between spike trains. *J. Neurosci. Methods*, 179(1):90–100, April 2009.
32. Soheil Feizi, Daniel Marbach, Muriel Médard, and Manolis Kellis. Network deconvolution as a general method to distinguish direct dependencies in networks. *Nat. Biotechnol.*, 31(8):726–733, August 2013.
33. Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, July 2008.
34. Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.
35. R Tyrrell Rockafellar. Convex analysis princeton university press. *Princeton, NJ*, 1970.
36. Fabian Schaipp, Oleg Vlasovets, and Christian L. Müller. Gglasso - a python package for general graphical lasso computation. *Journal of Open Source Software*, 6(68):3865, 2021. doi: 10.21105/joss.03865.
37. Rina Foygel and Mathias Drton. Extended bayesian information criteria for gaussian graphical models. In J Lafferty, C Williams, J Shawe-Taylor, R Zemel, and A Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.
38. Tiger W Lin, Anup Das, Giri P Krishnan, Maxim Bazhenov, and Terrence J Sejnowski. Differential covariance: A new class of methods to estimate sparse connectivity from neural recordings. *Neural Comput.*, 29(10):2581–2632, October 2017.
39. J R M Hosking. Fractional differencing. *Biometrika*, 68(1):165–176, April 1981.
40. Hideaki Shimazaki and Shigeru Shinomoto. Kernel bandwidth optimization in spike rate estimation. *J. Comput. Neurosci.*, 29(1-2):171–182, August 2010.
41. Eilen Nordlie, Tom Tetzlaff, and Gaute T Einevoll. Rate dynamics of leaky Integrate-and-Fire neurons with strong synapses. *Front. Comput. Neurosci.*, 4:149, December 2010.
42. Eric R Kandel, James H Schwartz, Thomas M Jessell, Department of Biochemistry, Molecular Biophysics Thomas Jessell, Steven Siegelbaum, and A J Hudspeth. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.
43. Jyothsna Suresh, Mihailo Radojicic, Lorenzo L Pesce, Anita Bhansali, Janice Wang, Andrew K Tryba, Jeremy D Marks, and Wim van Drongelen. Network burst activity in hippocampal neuronal cultures: the role of synaptic and intrinsic currents. *J. Neurophysiol.*, 115(6):3073–3089, June 2016.
44. Taehoon Kim, Dexiong Chen, Philipp Hornauer, Vishalini Emmenegger, Julian Bartram, Silvia Ronchi, Andreas Hierlemann, Manuel Schröter, and Damian Roqueiro. Predicting in vitro single-neuron firing rates upon pharmacological perturbation using graph neural networks. *Front. Neuroinform.*, 16:1032538, 2022.
45. Silvia Ronchi, Michele Fiscella, Camilla Marchetti, Vijay Viswam, Jan Müller, Urs Frey, and Andreas Hierlemann. Single-Cell electrical stimulation using CMOS-Based High-Density microelectrode arrays. *Front. Neurosci.*, 13(March), 2019.
46. Marius Pachitariu, Nicholas Steinmetz, Shabnam Kadir, Matteo Carandini, and Kenneth D Harris. Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels. *bioRxiv*, page 061481, 2016.
47. Daniel N Hill, Samar B Mehta, and David Kleinfeld. Quality metrics to accompany spike sorting of extracellular signals. *Journal of Neuroscience*, 31(24):8699–8705, 2011.
48. Stefano Recanatesi, Serena Bradde, Vijay Balasubramanian, Nicholas A Steinmetz, and Eric Shea-Brown. A scale-dependent measure of system dimensionality. *Patterns (N Y)*, 3 (8):100555, August 2022.
49. Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv*, page 214262, November 2017.
50. George J Stigler, Chairman, and Universities-National Bureau Committee for Economic Research. Business concentration and price policy. Technical Report univ55-1, National Bureau of Economic Research, January 1955.
51. D J Watts and S H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393 (6684):440–442, June 1998.
52. M D Humphries, K Gurney, and T J Prescott. The brainstem reticular formation is a small-world, not scale-free, network. *Proc. Biol. Sci.*, 273(1585):503–511, February 2006.
53. Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, 52(3):1059–1069, September 2010.
54. Gonzalo Mateos, Santiago Segarra, Antonio G Marques, and Alejandro Ribeiro. Connecting the dots: Identifying network structure via graph signal processing. *IEEE Signal Process. Mag.*, 36(3):16–43, May 2019.
55. Fernando Gama, Elvin Isufi, Geert Leus, and Alejandro Ribeiro. Graphs, convolutions, and neural networks: From graph filters to graph neural networks. *IEEE Signal Process. Mag.*, 37(6):128–138, November 2020.
56. Santiago Segarra, Antonio G Marques, Gonzalo Mateos, and Alejandro Ribeiro. Network topology inference from spectral templates. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):467–483, September 2017.
57. Rasoul Shafipour, Santiago Segarra, Antonio G Marques, and Gonzalo Mateos. Network topology inference from non-stationary graph signals. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5870–5874, March 2017.
58. Weiyu Huang, Leah Goldsberry, Nicholas F Wymbs, Scott T Grafton, Danielle S Bassett, and Alejandro Ribeiro. Graph frequency analysis of brain signals. *IEEE J. Sel. Top. Signal Process.*, 10(7):1189–1203, October 2016.
59. Weiyu Huang, Thomas A W Bolton, John D Medaglia, Danielle S Bassett, Alejandro Ribeiro, and Dimitri Van De Ville. A graph signal processing perspective on functional brain imaging. *Proc. IEEE*, 106(5):868–885, May 2018.
60. Liu Rui, Hossein Nejati, Seyed Hamid Safavi, and Ngai-Man Cheung. Simultaneous low-rank component and graph estimation for high-dimensional graph signals: Application to brain imaging. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4134–4138, March 2017.
61. David A Dickey and Wayne A Fuller. Distribution of the estimators for autoregressive time series with a unit root. *J. Am. Stat. Assoc.*, 74(366a):427–431, June 1979.
62. Julia H Downes, Mark W Hammond, Dimitris Xydas, Matthew C Spencer, Victor M Becerra, Kevin Warwick, Ben J Whalley, and Slawomir J Nasuto. Emergence of a small-world functional network in cultured neurons. *PLoS Comput. Biol.*, 8(5):e1002522, May 2012.
63. Manuel S Schroeter, Paul Charlesworth, Manfred G Kitzbichler, Ole Paulsen, and Edward T Bullmore. Emergence of rich-club topology and coordinated dynamics in development of hippocampal functional networks in vitro. *J. Neurosci.*, 35(14):5459–5470, April 2015.
64. Lennart Ljung. System identification. In Ales Procházka, Jan Uhlíř, P W J Rayner, and N G Kingsbury, editors, *Signal Analysis and Prediction*, pages 163–173. Birkhäuser Boston, Boston, MA, 1998.
65. Danielle S Bassett and Edward T Bullmore. Small-World brain networks revisited. *Neuroscientist*, 23(5):499–516, 2017.
66. Ye Wu, Yun Shang, Maoyin Chen, Changsong Zhou, and Jürgen Kurths. Synchronization in small-world networks. *Chaos*, 18(3):037111, September 2008.
67. Vivek Kohar, Peng Ji, Anshul Choudhary, Sudeshna Sinha, and Jüergen Kurths. Synchronization in time-varying networks. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 90 (2):022812, August 2014.
68. C Benoit. Note sur une méthode de résolution des équations normales provenant de l'application de la méthode des moindres carrésa un systeme d'équations . . . . *Bull. géod.*, 1924.
69. A Beck and M Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imag. Sci.*, 2009.
70. Amir Beck and Marc Teboulle. Gradient-based algorithms with applications to signal-recovery problems. In *Convex Optimization in Signal Processing and Communications*, pages 42–88. Cambridge University Press, December 2009.

Supplementary Material

Table S1. LIF simulation parameters. In this study, Q_E and Q_I were fixed as (10 nS, 65 nS) after the parameter search.

Parameters	Value	Unit
C (membrane capacitance)	200	pF
E_L (leaky membrane potential)	-60	mV
g_L (leakage conductance)	10	nS
V_T (effective threshold)	-50	mV
V_{reset} (reset potential)	-60	mV
T_{refrac} (refractory period)	5	5ms
Δ (threshold slope factor)	2.5	ms
E_E (reversal potential (exc.))	0	mV
E_I (reversal potential (inh.))	-80	mV
Q_E (synaptic strength change (exc.))	10	nS
Q_I (synaptic strength change (inh.))	65	nS
τ_m (time constant for white noise)	20	ms

Table S2. Parameters for spike-sorting (Kilosort2)

Parameters	value
ops.fs	20000
ops.fshigh	150
ops.minfr_goodchannels	0
ops.Th	[10 4]
ops.lam	10
ops.AUCsplit	0.9
ops.minFR	1/50
ops.momentum	[20 400]
ops.sigmaMask	30
ops.ThPre	8
ops.spkTh	-6
ops.reorder	1
ops.nskip	25
ops.nfilt_factor	4
ops.ntbuff	64
ops.NT	4*64*1024+ ops.ntbuff
ops.whiteningRange	32
ops.nSkipCov	25
ops.scaleproc	200
ops.nPCs	3

Table S3. Performance of CCG method. The table shows the performance (AUROC, AUPRC) of CCG method across different inference conditions.

				AUROC		AUPRC		Comp. time (s)	
Method	Network	Neurons	Duration (s)	mean	std.	mean	std.	mean	std.
CCG	Async	500	300	0.86	0.022	0.49	0.044	66	16
			600	0.91	0.018	0.54	0.043	110	20
			900	0.92	0.017	0.55	0.039	160	36
			1200	0.94	0.015	0.55	0.035	200	42
		1000	300	0.86	0.015	0.48	0.032	350	50
			600	0.9	0.013	0.53	0.031	630	79
			900	0.92	0.012	0.54	0.028	930	140
			1200	0.93	0.011	0.55	0.025	1200	210
		1500	300	0.86	0.014	0.48	0.028	880	46
			600	0.9	0.013	0.53	0.028	1600	150
			900	0.92	0.012	0.54	0.026	2500	420
			1200	0.93	0.011	0.55	0.024	3300	290
	Sync	500	300	0.59	0.051	0.029	0.0082	51	5.3
			600	0.58	0.051	0.026	0.005	88	12
			900	0.58	0.05	0.025	0.0041	120	17
			1200	0.58	0.048	0.025	0.0036	150	15
		1000	300	0.6	0.045	0.031	0.0083	250	30
			600	0.59	0.047	0.027	0.0051	470	69
			900	0.59	0.047	0.026	0.0043	680	92
			1200	0.59	0.047	0.025	0.0038	880	140
		1500	300	0.58	0.063	0.029	0.0093	570	57
			600	0.58	0.062	0.026	0.0058	1100	200
			900	0.55	0.059	0.024	0.0045	1600	260
			1200	0.56	0.059	0.024	0.0042	2000	250

Table S4. Performance of multivariate methods (asynchronous networks). The table shows the performance (AUROC, AUPRC) of multivariate methods across different inference conditions.

				AUROC		AUPRC		Comp. time (s)	
Network	Method	Neurons	Duration (s)	mean	std	mean	std	mean	std
Async	Dcov	500	300	0.61	0.017	0.063	0.0071	2	0.18
			600	0.61	0.018	0.064	0.0076	4.2	0.27
			900	0.61	0.018	0.064	0.0078	6.4	0.67
			1200	0.61	0.018	0.065	0.0078	8.9	1.8
		1000	300	0.6	0.013	0.063	0.0059	5.6	1.3
			600	0.61	0.014	0.063	0.0063	11	3.2
			900	0.61	0.014	0.064	0.0065	18	7.7
			1200	0.61	0.014	0.064	0.0065	23	7.3
		1500	300	0.6	0.012	0.062	0.0056	10	2.8
			600	0.61	0.013	0.063	0.006	20	5.8
			900	0.61	0.013	0.064	0.0061	31	11
			1200	0.61	0.013	0.064	0.0062	39	9.3
	DDC	500	300	0.7	0.016	0.055	0.0074	2	0.12
			600	0.73	0.017	0.068	0.01	4.1	0.09
			900	0.74	0.018	0.075	0.012	6.2	0.091
			1200	0.74	0.018	0.08	0.013	8.1	0.14
		1000	300	0.72	0.014	0.058	0.0074	5.1	0.13
			600	0.75	0.014	0.075	0.011	10	0.22
			900	0.76	0.015	0.085	0.013	15	0.22
			1200	0.77	0.014	0.093	0.014	20	0.39
		1500	300	0.72	0.013	0.058	0.0066	11	4.8
			600	0.76	0.014	0.076	0.01	21	7.6
			900	0.77	0.015	0.088	0.012	31	12
			1200	0.78	0.014	0.097	0.014	40	13
	Glasso	500	300	0.51	0.0015	0.022	0.00053	1.9	0.97
			600	0.51	0.0015	0.022	0.00053	1.5	0.076
			900	0.51	0.0014	0.022	0.00054	1.8	0.28
			1200	0.51	0.0014	0.022	0.00053	2	0.21
		1000	300	0.51	0.0008	0.022	0.00028	4.9	0.98
			600	0.51	0.00082	0.022	0.00029	4.7	0.16
			900	0.51	0.00079	0.022	0.00029	5.3	0.15
			1200	0.51	0.00078	0.022	0.00029	5.9	0.19
		1500	300	0.51	0.00068	0.022	0.00026	10	1.3
			600	0.51	0.00068	0.022	0.00026	10	0.68
			900	0.51	0.00067	0.022	0.00027	11	0.69
			1200	0.51	0.00065	0.022	0.00027	12	0.74
	ND	500	300	0.6	0.0059	0.042	0.0017	0.41	0.04
			600	0.6	0.0063	0.043	0.0018	0.63	0.024
			900	0.6	0.007	0.043	0.0018	0.86	0.026
			1200	0.6	0.0065	0.043	0.0018	1.1	0.021
		1000	300	0.61	0.0055	0.049	0.0021	1.5	0.044
			600	0.62	0.0061	0.051	0.0023	1.9	0.045
			900	0.62	0.0063	0.052	0.0024	2.5	0.08
			1200	0.62	0.0062	0.052	0.0024	3	0.097
		1500	300	0.62	0.0046	0.055	0.0021	5.3	4.4
			600	0.63	0.0055	0.058	0.0023	6.4	4.9
			900	0.63	0.0054	0.059	0.0024	7.5	5.2
			1200	0.63	0.0054	0.059	0.0024	8.5	5.1

Table S5. Performance of multivariate methods (synchronous networks). The table shows the performance (AUROC, AUPRC) of multivariate methods across different inference conditions.

				AUROC		AUPRC		Comp. time (s)	
Network	Method	Neurons	Duration (s)	mean	std	mean	std	mean	std
Sync	Dcov	500	300	0.51	0.0056	0.021	0.00053	2	0.13
			600	0.51	0.0054	0.021	0.00053	4.1	0.13
			900	0.51	0.0054	0.021	0.00053	6.1	0.23
			1200	0.51	0.0054	0.021	0.00052	8.2	0.27
		1000	300	0.51	0.0032	0.021	0.00036	5	0.075
			600	0.51	0.003	0.021	0.00035	10	0.23
			900	0.51	0.003	0.021	0.00035	15	0.42
			1200	0.51	0.003	0.021	0.00035	20	0.58
		1500	300	0.51	0.0027	0.021	0.00026	8.8	0.12
			600	0.51	0.0024	0.021	0.00025	18	0.32
			900	0.51	0.0024	0.021	0.00025	26	0.56
			1200	0.51	0.0024	0.021	0.00025	35	0.69
	DDC	500	300	0.68	0.058	0.05	0.012	2.8	1.5
			600	0.7	0.06	0.054	0.014	5	1.9
			900	0.71	0.061	0.056	0.015	7.1	2.3
			1200	0.71	0.063	0.057	0.015	9.1	2.2
		1000	300	0.73	0.027	0.069	0.0098	6.1	2.1
			600	0.76	0.026	0.082	0.012	13	5.4
			900	0.77	0.026	0.088	0.013	17	3.4
			1200	0.77	0.026	0.091	0.014	23	6.1
		1500	300	0.74	0.029	0.074	0.0095	9	0.13
			600	0.77	0.028	0.094	0.012	18	0.3
			900	0.79	0.027	0.1	0.014	27	0.61
			1200	0.79	0.027	0.11	0.015	35	0.48
	Glasso	500	300	0.51	0.0026	0.021	0.0005	31	54
			600	0.51	0.0028	0.021	0.0005	28	48
			900	0.51	0.0027	0.021	0.00049	23	40
			1200	0.51	0.0028	0.021	0.00051	23	38
		1000	300	0.51	0.00083	0.021	0.00031	45	55
			600	0.51	0.00087	0.021	0.00031	35	42
			900	0.51	0.00086	0.021	0.00032	53	71
			1200	0.51	0.00087	0.021	0.00032	53	73
		1500	300	0.51	0.00056	0.021	0.00017	70	71
			600	0.51	0.00055	0.021	0.00017	80	91
			900	0.51	0.00055	0.021	0.00017	78	86
			1200	0.51	0.00056	0.021	0.00017	82	93
	ND	500	300	0.55	0.021	0.028	0.0033	1.2	1.6
			600	0.55	0.021	0.028	0.0034	1.4	1.7
			900	0.55	0.021	0.028	0.0034	1.6	1.5
			1200	0.55	0.021	0.028	0.0034	1.8	1.6
		1000	300	0.57	0.017	0.033	0.0037	2.7	2.4
			600	0.57	0.017	0.034	0.0037	3.5	3
			900	0.57	0.017	0.034	0.0038	4	3
			1200	0.57	0.016	0.034	0.0037	4	1.9
		1500	300	0.57	0.021	0.035	0.0049	3.3	0.083
			600	0.58	0.021	0.036	0.0052	4.2	0.046
			900	0.58	0.021	0.036	0.0053	5.3	0.23
			1200	0.58	0.021	0.037	0.0053	6.2	0.17

Table S6. Performance of undirected multivariate methods. The table shows the performance (AUROC, AUPRC) of the undirected multivariate methods (Glasso, ND) evaluated using the undirected ground truth connections.

				AUROC		AUPRC		Comp. time (s)	
Network	Method	Neurons	Duration (s)	mean	std	mean	std	mean	std
Async	Glasso	500	300	0.51	0.0015	0.043	0.00089	1.9	0.97
			600	0.51	0.0015	0.043	0.00088	1.5	0.076
			900	0.51	0.0014	0.043	0.00089	1.8	0.28
			1200	0.51	0.0014	0.043	0.00088	2	0.21
		1000	300	0.51	0.00081	0.043	0.00054	4.9	0.98
			600	0.51	0.00083	0.043	0.00055	4.7	0.16
			900	0.51	0.00079	0.043	0.00055	5.3	0.15
			1200	0.51	0.00079	0.043	0.00054	5.9	0.19
		1500	300	0.51	0.00066	0.044	0.00044	10	1.3
			600	0.51	0.00066	0.044	0.00045	10	0.68
			900	0.51	0.00065	0.044	0.00046	11	0.69
			1200	0.51	0.00064	0.044	0.00046	12	0.74
	ND	500	300	0.6	0.006	0.082	0.0032	0.41	0.04
			600	0.6	0.0064	0.084	0.0034	0.63	0.024
			900	0.61	0.007	0.084	0.0034	0.86	0.026
			1200	0.61	0.0066	0.085	0.0035	1.1	0.021
		1000	300	0.61	0.0057	0.097	0.0042	1.5	0.044
			600	0.62	0.0063	0.1	0.0045	1.9	0.045
			900	0.62	0.0065	0.1	0.0046	2.5	0.08
			1200	0.62	0.0063	0.1	0.0047	3	0.097
		1500	300	0.62	0.0047	0.11	0.0041	5.3	4.4
			600	0.63	0.0057	0.11	0.0047	6.4	4.9
			900	0.63	0.0056	0.11	0.0047	7.5	5.2
			1200	0.63	0.0056	0.12	0.0048	8.5	5.1
Sync	Glasso	500	300	0.51	0.0026	0.042	0.00089	31	54
			600	0.51	0.0028	0.042	0.00093	28	48
			900	0.51	0.0028	0.042	0.00092	23	40
			1200	0.51	0.0028	0.042	0.00094	23	38
		1000	300	0.51	0.00083	0.042	0.00058	45	55
			600	0.51	0.00086	0.042	0.00058	35	42
			900	0.51	0.00085	0.042	0.00058	53	71
			1200	0.51	0.00086	0.042	0.00058	53	73
		1500	300	0.51	0.00059	0.042	0.00033	70	71
			600	0.51	0.00056	0.042	0.00032	80	91
			900	0.51	0.00057	0.042	0.00033	78	86
			1200	0.51	0.00058	0.042	0.00032	82	93
	ND	500	300	0.55	0.021	0.055	0.0064	1.2	1.6
			600	0.55	0.021	0.056	0.0065	1.4	1.7
			900	0.55	0.021	0.056	0.0065	1.6	1.5
			1200	0.55	0.021	0.056	0.0065	1.8	1.6
		1000	300	0.57	0.017	0.065	0.0071	2.7	2.4
			600	0.57	0.017	0.066	0.0072	3.5	3
			900	0.57	0.017	0.067	0.0073	4	3
			1200	0.57	0.017	0.067	0.0073	4	1.9
		1500	300	0.58	0.021	0.069	0.0096	3.3	0.083
			600	0.58	0.022	0.071	0.01	4.2	0.046
			900	0.58	0.022	0.072	0.01	5.3	0.23
			1200	0.58	0.022	0.072	0.01	6.2	0.17

Table S7. Performance of DDC on convolved spike trains (asynchronous networks). The performance (AUROC, AUPRC) of DDC on convolved spike trains is presented in the table for four different kernels.

				AUROC		AUPRC		Comp. time (s)	
Network	Method	Neurons	Duration (s)	mean	std	mean	std	mean	std
Async	DDC (Alpha, 1 ms)	500	300	0.68	0.016	0.065	0.011	19	0.49
			600	0.71	0.018	0.092	0.018	39	1.4
			900	0.73	0.018	0.11	0.023	59	1.2
			1200	0.74	0.019	0.12	0.026	79	0.91
		1000	300	0.68	0.013	0.065	0.0097	44	0.62
			600	0.71	0.014	0.092	0.016	88	0.74
			900	0.73	0.015	0.11	0.02	130	0.93
			1200	0.74	0.015	0.13	0.023	180	3.1
		1500	300	0.68	0.011	0.064	0.0087	72	3.6
			600	0.71	0.012	0.092	0.015	150	3.7
			900	0.73	0.012	0.11	0.019	220	4.4
			1200	0.74	0.013	0.13	0.022	290	5.3
	DDC (Alpha, 3 ms)	500	300	0.72	0.017	0.066	0.011	20	0.76
			600	0.75	0.018	0.079	0.014	40	1.2
			900	0.76	0.018	0.087	0.016	61	1.1
			1200	0.76	0.017	0.092	0.018	82	1.2
		1000	300	0.73	0.013	0.069	0.01	45	1.2
			600	0.76	0.014	0.087	0.014	90	2.5
			900	0.77	0.014	0.097	0.016	130	3.7
			1200	0.78	0.014	0.1	0.018	180	5
		1500	300	0.73	0.012	0.068	0.0088	76	3
			600	0.77	0.013	0.088	0.012	150	4
			900	0.78	0.013	0.1	0.014	230	4.7
			1200	0.79	0.013	0.11	0.016	300	7.5
	DDC (Gaussian, 1 ms)	500	300	0.68	0.016	0.067	0.012	20	0.67
			600	0.72	0.018	0.095	0.019	40	1.2
			900	0.73	0.018	0.11	0.023	59	2
			1200	0.74	0.018	0.13	0.026	80	1.4
		1000	300	0.69	0.014	0.068	0.01	44	0.55
			600	0.72	0.014	0.096	0.017	89	1.4
			900	0.74	0.015	0.12	0.021	130	2
			1200	0.75	0.015	0.13	0.024	180	1.8
		1500	300	0.69	0.011	0.067	0.0092	74	2.5
			600	0.72	0.012	0.096	0.016	150	3.5
			900	0.74	0.012	0.12	0.02	220	5.6
			1200	0.75	0.013	0.13	0.023	300	6.7
	DDC (Gaussian, 3 ms)	500	300	0.76	0.02	0.073	0.012	21	0.41
			600	0.78	0.021	0.086	0.015	41	0.66
			900	0.78	0.021	0.093	0.016	62	1.2
			1200	0.79	0.021	0.097	0.017	83	1.7
		1000	300	0.79	0.016	0.084	0.013	46	1.3
			600	0.81	0.017	0.1	0.017	92	1.1
			900	0.82	0.016	0.11	0.019	140	1.9
			1200	0.83	0.016	0.12	0.02	180	2.2
		1500	300	0.8	0.015	0.088	0.012	77	2.7
			600	0.83	0.015	0.11	0.016	150	2.9
			900	0.84	0.015	0.13	0.018	230	4.1
			1200	0.85	0.015	0.14	0.019	310	6.6

Table S8. Performance of DDC on convolved spike trains (synchronous networks). The performance (AUROC, AUPRC) of DDC on convolved spike trains is presented in the table for four different kernels.

				AUROC		AUPRC		Comp. time (s)	
Network	Method	Neurons	Duration (s)	mean	std	mean	std	mean	std
Sync	DDC (Alpha, 1 ms)	500	300	0.66	0.045	0.074	0.018	20	3
			600	0.68	0.049	0.081	0.021	41	6.4
			900	0.68	0.05	0.083	0.021	62	9
			1200	0.69	0.052	0.085	0.022	81	8.7
		1000	300	0.69	0.012	0.099	0.0089	43	1.6
			600	0.72	0.014	0.11	0.011	84	2.6
			900	0.73	0.016	0.12	0.013	130	4.9
			1200	0.73	0.017	0.12	0.014	170	5.8
		1500	300	0.7	0.012	0.1	0.0074	71	2.6
			600	0.73	0.014	0.13	0.0092	150	13
			900	0.74	0.016	0.13	0.011	230	35
			1200	0.75	0.017	0.14	0.012	320	62
	DDC (Alpha, 3 ms)	500	300	0.68	0.052	0.062	0.013	19	0.68
			600	0.69	0.053	0.065	0.014	37	1.4
			900	0.7	0.054	0.066	0.015	56	2.1
			1200	0.7	0.055	0.067	0.015	75	3
		1000	300	0.73	0.017	0.09	0.0065	43	0.77
			600	0.74	0.017	0.099	0.0075	86	1.8
			900	0.75	0.017	0.1	0.0083	130	3.2
			1200	0.75	0.018	0.1	0.0088	170	3.2
		1500	300	0.74	0.017	0.1	0.006	72	3.2
			600	0.76	0.018	0.11	0.0074	140	7.6
			900	0.77	0.018	0.12	0.0087	220	9.8
			1200	0.77	0.018	0.12	0.0095	290	13
	DDC (Gaussian, 1 ms)	500	300	0.67	0.048	0.078	0.019	19	0.39
			600	0.69	0.051	0.085	0.022	39	0.41
			900	0.7	0.053	0.088	0.023	58	0.74
			1200	0.7	0.055	0.09	0.023	77	0.82
		1000	300	0.71	0.012	0.11	0.0091	42	0.88
			600	0.73	0.014	0.12	0.012	85	1.4
			900	0.74	0.016	0.13	0.013	130	2.4
			1200	0.75	0.016	0.13	0.014	170	3.7
		1500	300	0.71	0.012	0.11	0.0077	77	15
			600	0.74	0.014	0.13	0.0093	160	33
			900	0.76	0.015	0.14	0.011	240	55
			1200	0.76	0.016	0.15	0.012	290	4.1
	DDC (Gaussian, 3 ms)	500	300	0.75	0.069	0.077	0.019	20	0.61
			600	0.76	0.071	0.083	0.02	39	1.1
			900	0.77	0.072	0.085	0.021	59	1.8
			1200	0.77	0.073	0.086	0.022	78	2.5
		1000	300	0.82	0.023	0.12	0.011	48	9
			600	0.84	0.021	0.13	0.012	98	20
			900	0.85	0.02	0.14	0.012	150	27
			1200	0.85	0.02	0.14	0.013	190	34
		1500	300	0.83	0.025	0.13	0.012	72	2.4
			600	0.86	0.023	0.16	0.013	150	7.1
			900	0.87	0.022	0.17	0.014	220	13
			1200	0.87	0.021	0.17	0.014	290	15

Table S9. Performance of FDDC on convolved spike trains (asynchronous networks). The performance (AUROC, AUPRC) of FDDC ($\beta = 0.1$) on convolved spike trains is presented in the table for four different kernels.

				AUROC		AUPRC		Comp. time (s)	
Network	Method	Neurons	Duration (s)	mean	std	mean	std	mean	std
Async	FDDC (Alpha, 1 ms, $\beta = 0.1$)	500	300	0.72	0.017	0.099	0.017	28	0.45
			600	0.73	0.016	0.12	0.018	56	0.8
			900	0.73	0.016	0.12	0.017	83	0.89
			1200	0.73	0.016	0.13	0.016	110	1.2
		1000	300	0.75	0.014	0.1	0.017	58	1.3
			600	0.76	0.013	0.13	0.02	120	1.6
			900	0.77	0.013	0.14	0.02	180	2.4
			1200	0.77	0.013	0.15	0.019	240	3.4
		1500	300	0.76	0.013	0.11	0.017	93	3.1
			600	0.78	0.013	0.14	0.022	190	3.4
			900	0.79	0.013	0.16	0.023	280	4.9
			1200	0.79	0.012	0.17	0.022	380	4.5
	FDDC (Alpha, 3 ms, $\beta = 0.1$)	500	300	0.79	0.02	0.16	0.031	27	0.5
			600	0.81	0.02	0.19	0.033	55	1
			900	0.81	0.02	0.21	0.033	83	1.2
			1200	0.81	0.019	0.22	0.033	110	1.3
		1000	300	0.82	0.017	0.17	0.032	59	1.6
			600	0.84	0.016	0.23	0.039	120	2
			900	0.85	0.016	0.25	0.039	180	2.9
			1200	0.85	0.015	0.27	0.038	240	3.1
		1500	300	0.84	0.016	0.17	0.032	92	2.8
			600	0.86	0.015	0.24	0.042	190	4.2
			900	0.87	0.015	0.28	0.044	290	5.2
			1200	0.87	0.014	0.31	0.044	380	5.2
	FDDC (Gaussian, 1 ms, $\beta = 0.1$)	500	300	0.73	0.017	0.1	0.019	27	0.72
			600	0.74	0.017	0.12	0.02	55	1.3
			900	0.74	0.017	0.13	0.019	82	2
			1200	0.74	0.016	0.14	0.018	110	1.8
		1000	300	0.75	0.014	0.11	0.019	58	1.9
			600	0.77	0.014	0.14	0.022	120	1.9
			900	0.78	0.013	0.16	0.022	180	2.6
			1200	0.78	0.013	0.16	0.022	230	2.6
		1500	300	0.77	0.014	0.11	0.018	92	2.9
			600	0.79	0.013	0.15	0.024	190	2.7
			900	0.8	0.013	0.17	0.025	280	4.3
			1200	0.8	0.013	0.18	0.025	380	6.8
	FDDC (Gaussian, 3 ms, $\beta = 0.1$)	500	300	0.78	0.02	0.12	0.026	27	0.88
			600	0.79	0.019	0.15	0.031	56	1.1
			900	0.8	0.019	0.17	0.033	83	1.3
			1200	0.8	0.019	0.18	0.033	110	1.6
		1000	300	0.81	0.016	0.13	0.025	58	1.7
			600	0.83	0.015	0.18	0.031	120	1.8
			900	0.84	0.015	0.2	0.033	180	2.1
			1200	0.84	0.015	0.22	0.034	240	2.9
		1500	300	0.82	0.015	0.14	0.023	93	2.6
			600	0.84	0.014	0.19	0.03	190	2.5
			900	0.85	0.014	0.22	0.033	290	3.9
			1200	0.86	0.013	0.24	0.033	380	5

Table S10. Performance of FDDC on convolved spike trains (synchronous networks). The performance (AUROC, AUPRC) of FDDC ($\beta = 0.1$) on convolved spike trains is presented in the table for four different kernels.

				AUROC		AUPRC		Comp. time (s)	
Network	Method	Neurons	Duration (s)	mean	std	mean	std	mean	std
Sync	FDDC (Alpha, 1 ms, $\beta = 0.1$)	500	300	0.73	0.065	0.14	0.043	27	0.47
			600	0.74	0.066	0.15	0.045	55	0.81
			900	0.74	0.067	0.15	0.046	82	1.3
			1200	0.74	0.067	0.15	0.046	110	2.1
		1000	300	0.78	0.018	0.21	0.017	58	1.5
			600	0.79	0.018	0.23	0.017	120	1.6
			900	0.8	0.018	0.24	0.016	180	2.1
			1200	0.8	0.017	0.24	0.015	240	2.1
		1500	300	0.79	0.019	0.23	0.018	92	2.2
			600	0.8	0.018	0.26	0.02	190	1.9
			900	0.81	0.018	0.27	0.019	280	3.2
			1200	0.81	0.018	0.28	0.016	380	4.6
	FDDC (Alpha, 3 ms, $\beta = 0.1$)	500	300	0.81	0.086	0.24	0.078	27	0.37
			600	0.82	0.088	0.25	0.081	55	0.85
			900	0.83	0.088	0.26	0.083	82	0.75
			1200	0.83	0.089	0.26	0.084	110	0.7
		1000	300	0.88	0.02	0.37	0.032	58	1.5
			600	0.9	0.018	0.4	0.03	120	1.6
			900	0.9	0.017	0.42	0.03	180	2.5
			1200	0.91	0.017	0.42	0.029	240	3.5
		1500	300	0.9	0.021	0.41	0.037	93	2.8
			600	0.92	0.018	0.46	0.033	190	2.9
			900	0.92	0.017	0.48	0.032	280	3.7
			1200	0.92	0.016	0.49	0.031	380	4.9
	FDDC (Gaussian, 1 ms, $\beta = 0.1$)	500	300	0.74	0.068	0.16	0.047	27	0.71
			600	0.75	0.069	0.16	0.05	55	1.4
			900	0.75	0.07	0.17	0.051	82	1.9
			1200	0.76	0.07	0.17	0.051	110	2.6
		1000	300	0.79	0.018	0.23	0.018	58	1.9
			600	0.8	0.018	0.25	0.018	120	2.5
			900	0.81	0.017	0.26	0.018	180	3.6
			1200	0.81	0.017	0.26	0.016	240	4.4
		1500	300	0.8	0.019	0.25	0.02	93	2.7
			600	0.82	0.018	0.28	0.021	190	3.5
			900	0.82	0.018	0.29	0.02	280	4
			1200	0.83	0.018	0.3	0.018	380	3.7
	FDDC (Gaussian, 3 ms, $\beta = 0.1$)	500	300	0.79	0.081	0.2	0.062	27	0.53
			600	0.8	0.082	0.21	0.066	55	0.75
			900	0.81	0.083	0.22	0.067	83	0.65
			1200	0.81	0.084	0.22	0.068	110	0.8
		1000	300	0.86	0.018	0.29	0.022	58	1.4
			600	0.88	0.016	0.32	0.019	120	1.3
			900	0.89	0.015	0.34	0.019	180	2.1
			1200	0.89	0.014	0.34	0.018	240	2.7
		1500	300	0.88	0.02	0.32	0.025	94	2.5
			600	0.9	0.017	0.36	0.021	190	3.7
			900	0.91	0.015	0.38	0.02	290	4.6
			1200	0.91	0.014	0.39	0.02	380	6.6

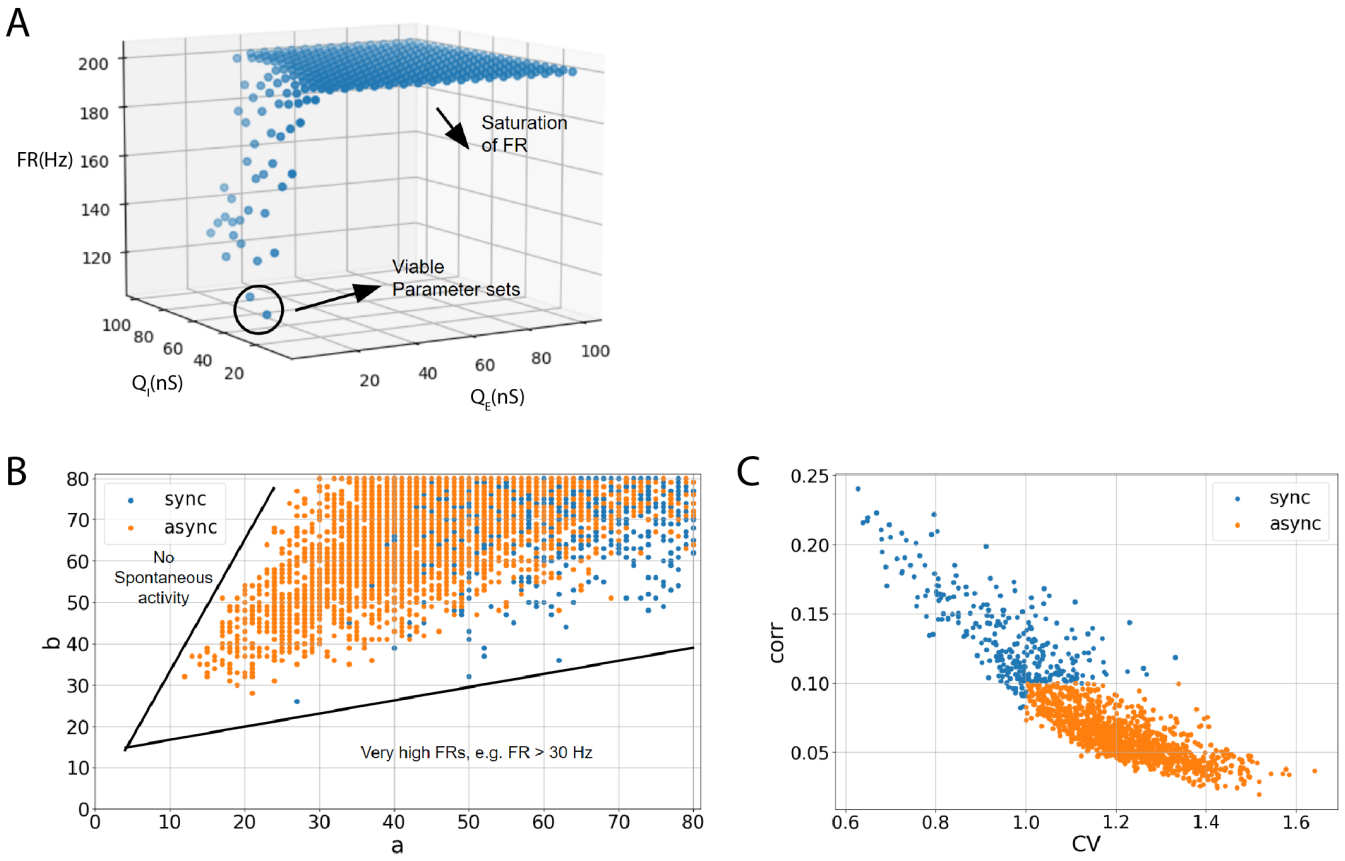

Fig. S1. Parameter search of LIF network simulations. (A): While fixing the adaptation parameters (a,b) = (1 μS,5nA) for excitatory neurons to implement 'weakly adapting cells' (24), the amounts for postsynaptic excitatory/inhibitory conductance change ($Q_{E,I}$) were searched in the range of [0 100] with a step size of 5. Each parameter set was simulated for 5 seconds, and the corresponding average population firing rates are shown. (B): Under the fixed parameter choice, (Q_E, Q_I) = (10, 65) nS, (a,b) parameter sets were searched in the range [0 80] with a step size of 1. Each parameter set was simulated for 5 seconds and network activities were color-coded to indicate either synchronous network activity ('sync') or asynchronous network activity ('async'). (C): Based on the criteria of (24), network simulations that showed a mean pairwise correlation ('corr') higher than 0.1 and a lower mean coefficient of variation ('CV') than 1 were labeled as synchronous ('sync'). All other network simulations that did not meet these two criteria were labeled as asynchronous networks ('async').

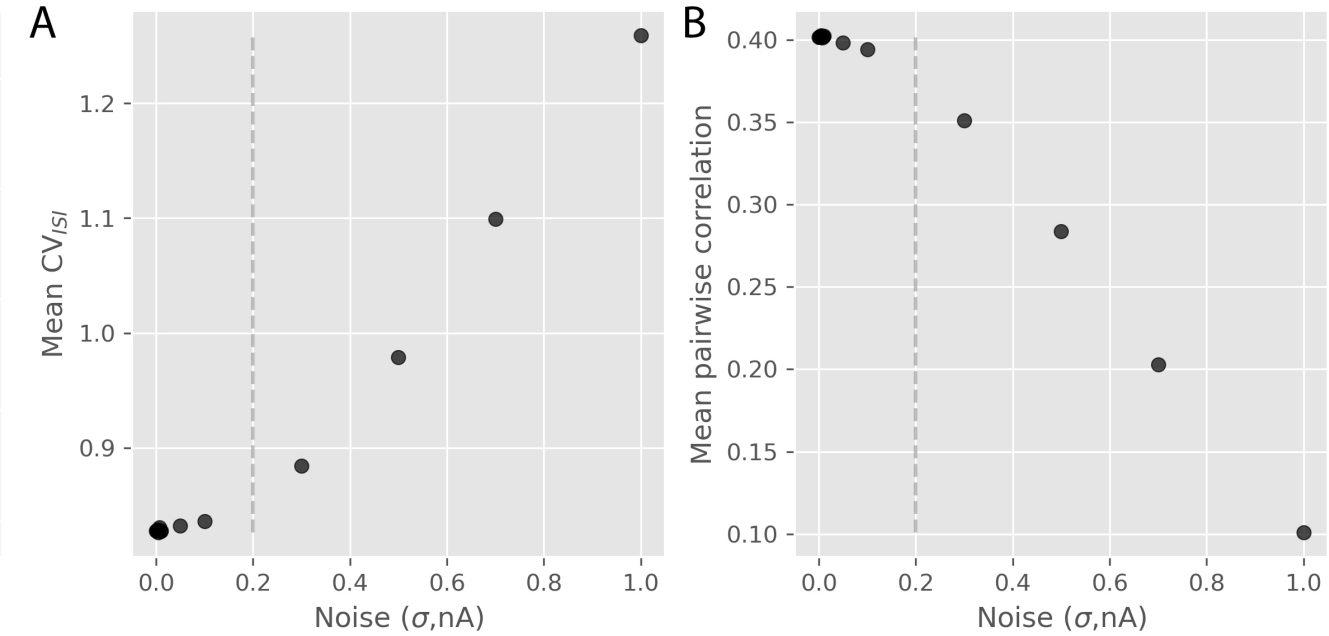

Fig. S2. Noise-induced network activity changes in synchronous LIF simulations. (A): For noises that were below 0.2 nA ($\approx$ size of the leaky current), the mean CV_{ISI} of the simulated synchronous networks remained relatively stable. However, with larger noises, there was a rapid increase in mean CV_{ISI}, and mean CV_{ISI} values larger than 1.0 were observed. (B): Similarly, with larger noise levels that exceed 0.2 nA, there was a steeper decrease in the mean pairwise correlation of the simulated synchronous networks compared to the case of lower noise levels.

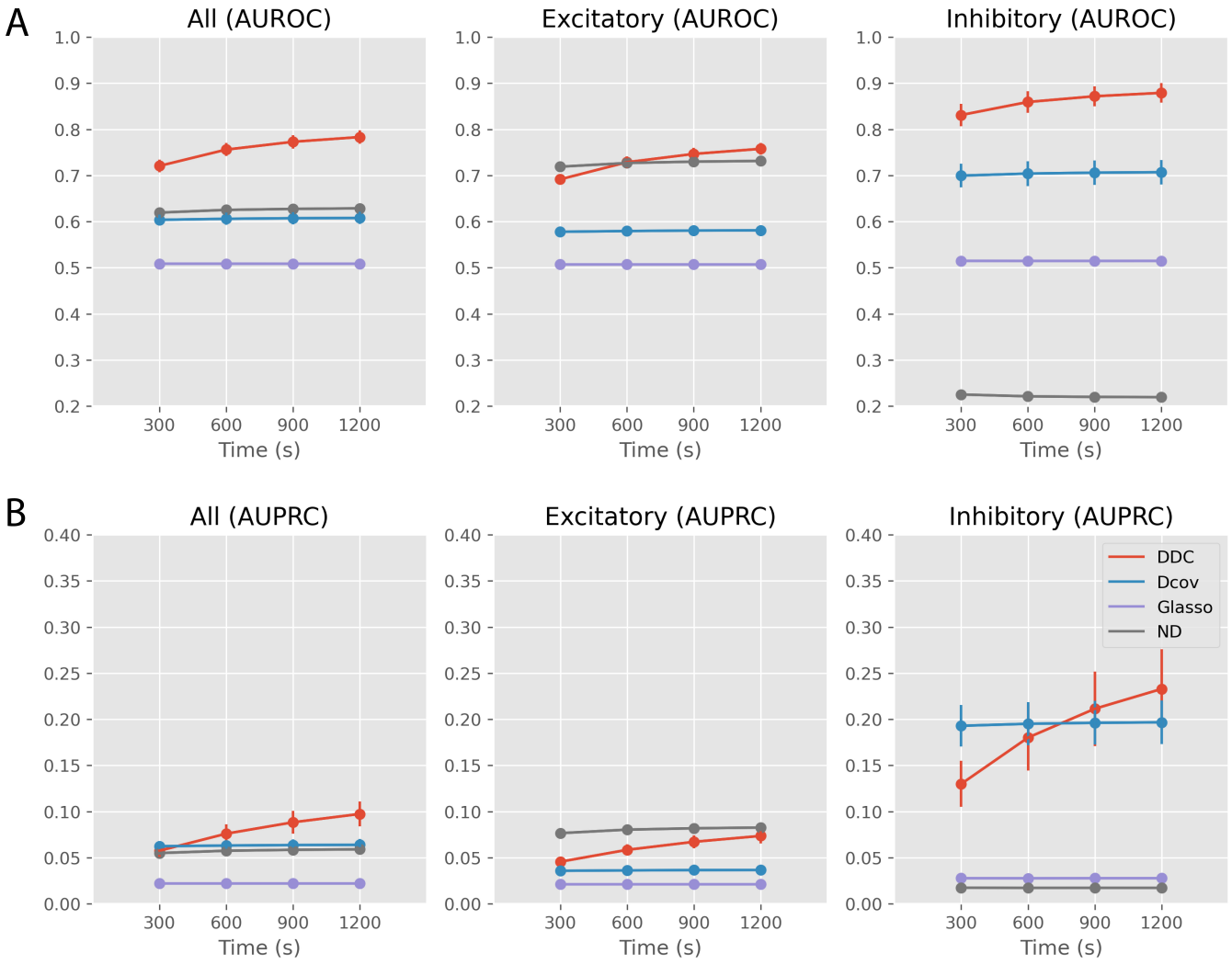

Fig. S3. Inference performances of the multivariate methods (asynchronous networks). (A): AUROC values (mean ± standard deviation) are presented for three evaluation conditions. (left): all ground truth connections, (middle): excitatory ground truth connections, (right): inhibitory ground truth connections. (B): AUPRC values are presented for three evaluation conditions. (left): all ground truth connections, (middle): excitatory ground truth connections, (right): inhibitory ground truth connections.

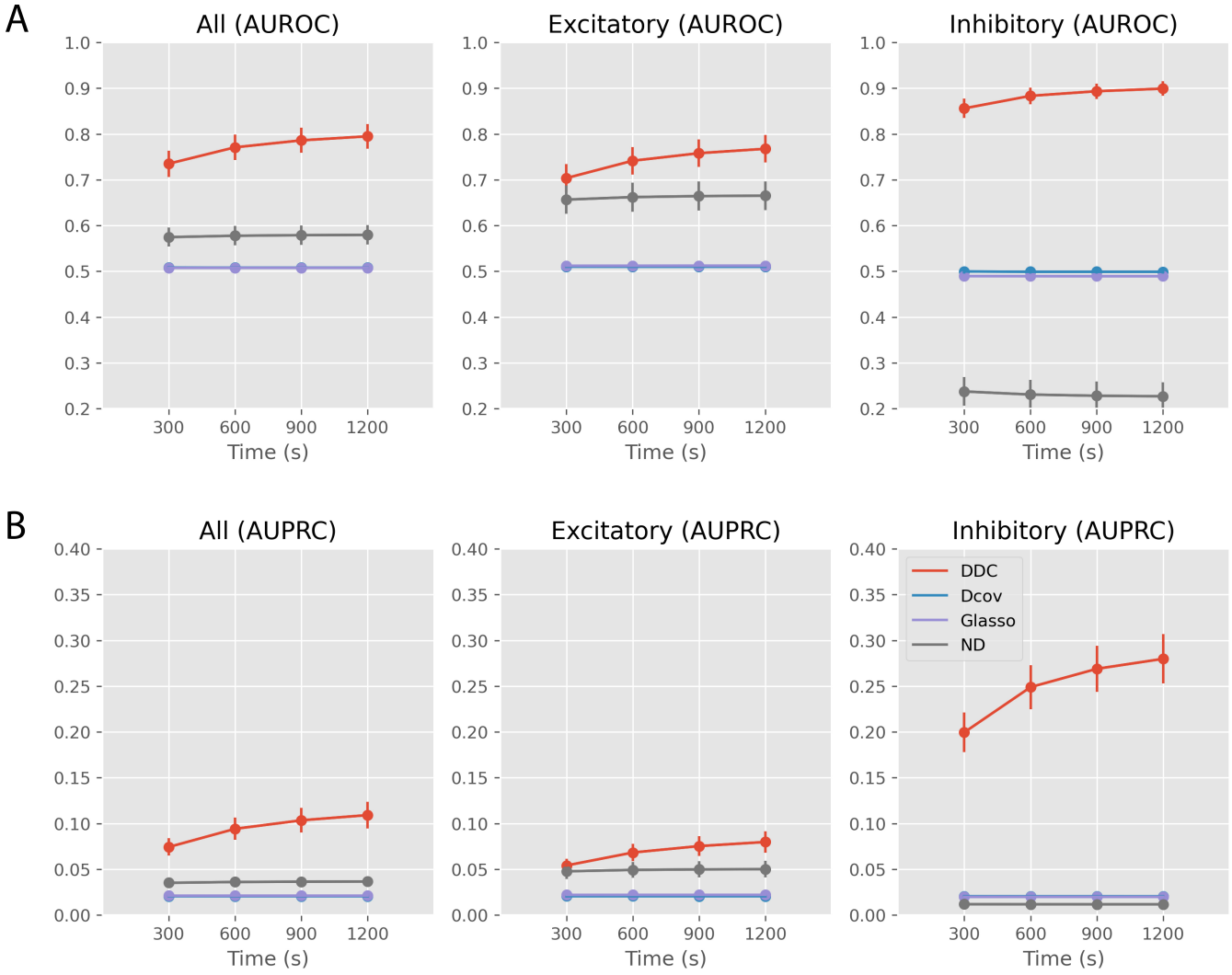

Fig. S4. Inference performances of the multivariate methods (synchronous networks). (A): AUROC values (mean ± standard deviation) are presented for three evaluation conditions. (left): all ground truth connections, (middle): excitatory ground truth connections, (right): inhibitory ground truth connections. (B): AUPRC values are presented for three evaluation conditions. (left): all ground truth connections, (middle): excitatory ground truth connections, (right): inhibitory ground truth connections.