Journal of Computational Neuroscience (2018) 45:83–101
https://doi.org/10.1007/s10827-018-0692-x

Adjusted regularization of cortical covariance

Giuseppe Vinci[1] · **Valérie Ventura**[2,3,5] · **Matthew A. Smith**[4,5] · **Robert E. Kass**[2,3,5]

Received: 18 July 2017 / Revised: 13 July 2018 / Accepted: 30 July 2018 / Published online: 6 September 2018

Abstract

It is now common to record dozens to hundreds or more neurons simultaneously, and to ask how the network activity changes across experimental conditions. A natural framework for addressing questions of functional connectivity is to apply Gaussian graphical modeling to neural data, where each edge in the graph corresponds to a non-zero partial correlation between neurons. Because the number of possible edges is large, one strategy for estimating the graph has been to apply methods that aim to identify large sparse effects using an L_1 penalty. However, the partial correlations found in neural spike count data are neither large nor sparse, so techniques that perform well in sparse settings will typically perform poorly in the context of neural spike count data. Fortunately, the correlated firing for any pair of cortical neurons depends strongly on both their distance apart and the features for which they are tuned. We introduce a method that takes advantage of these known, strong effects by allowing the penalty to depend on them: thus, for example, the connection between pairs of neurons that are close together will be penalized less than pairs that are far apart. We show through simulations that this physiologically-motivated procedure performs substantially better than off-the-shelf generic tools, and we illustrate by applying the methodology to populations of neurons recorded with multielectrode arrays implanted in macaque visual cortex areas V1 and V4.

Keywords Bayesian inference · False discovery rate · Functional connectivity · Gaussian graphical model · Graphical lasso · High-dimensional estimation · Macaque visual cortex · Penalized maximum likelihood estimation

1 Introduction

The rapid growth in the number of neurons being recorded simultaneously (Ahrens et al. 2013; Alivisatos et al. 2013; Kerr and Denk 2008; Kipke et al. 2008) creates an urgent need for statistical procedures that can identify the structure of covariation in neural network activity (Shadlen and Newsome 1998; Brown et al. 2004; Cunningham and Yu 2014; Stevenson and Kording 2011; Song et al. 2013; Yatsenko et al. 2015; Cohen and Maunsell 2009; Cohen and Kohn 2011; Efron et al. 2001; Kelly and Kass 2012; Mitchell et al. 2009; Vinci et al. 2016). An appealing approach to network analysis begins by representing multivariate activity as a graph, that is, a set of nodes together with a specification of which nodes are connected by edges (Bassett and Sporns 2017). In the case of multi-neuron recordings, each node would correspond to a neuron. Because these recordings are typically noisy, capturing in full detail the interactions among neurons, which can occur at multiple timescales, is very difficult. An initial simplification is to consider the vector of spike counts, within a time interval of several hundred milliseconds, to be a Gaussian random vector X whose covariance matrix

Action Editor: Uri Eden

✉ Giuseppe Vinci
giuseppe.vinci@rice.edu

Valérie Ventura
vventura@andrew.cmu.edu

Matthew A. Smith
matt@smithlab.net

Robert E. Kass
kass@andrew.cmu.edu

[1] Department of Statistics, Rice University, 6100 Main St, Houston, TX 77005, USA

[2] Department of Statistics & Data Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

[3] Machine Learning Department, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

[4] Department of Ophthalmology, University of Pittsburgh, Eye and Ear Institute, 203 Lothrop St., Room 914, Pittsburgh, PA 15213, USA

[5] Center for the Neural Basis of Cognition, 4400 Fifth Avenue, Suite 115, Pittsburgh, PA 15213, USA

Σ defines a graph based on the inverse matrix $\Omega = \Sigma^{-1}$ (assuming Σ is invertible). Specifically, the edges in the graph correspond to the non-zero off-diagonal elements of $\Omega = [\omega_{ij}]$, that is, an edge between nodes i and j is absent if and only if $\omega_{ij} = 0$. Furthermore, $\omega_{ij} = 0$ if and only if the corresponding partial correlation satisfies $\rho_{ij} = 0$, and $\rho_{ij} = 0$ if and only if the i and j nodes are independent conditionally on all the other nodes, that is in our case, an edge exists between pairs of neurons that have a unique component of covariation that is not associated with all the other neurons. The time-scale, imposed by the scientific questions of interest or by objective choice of bin size, such as experimental task durations, affects the elements of Σ and hence the conclusions one might draw from the graph.

Such Gaussian graphical models are widely applied and studied (Murphy 2012). However, even this simple case becomes challenging as the number of neurons grows: although, for typical spike count data, estimation of any single correlation coefficient may incur a relatively small error, compounding thousands of such small errors produces an unstable estimate of the matrix Σ. Thus, some form of regularization in the estimation of Σ is usually applied. In recent years, the most commonly-applied form of covariance regularization has been the Graphical lasso (**Glasso**) (Yuan and Lin 2007; Friedman et al. 2008; Banerjee et al. 2008; Rothman et al. 2008; Mazumder and Hastie 2012). To define it, we write the Gaussian likelihood function as $L(\Omega; \mathbb{X}_n)$, where $\mathbb{X}_n = \{X^{(1)}, ..., X^{(n)}\}$ with $X^{(r)}$ representing the Gaussian random vector of spike counts of d neurons on trial r, for $r = 1, \ldots, n$, we assume the number of non-zero elements of Ω is comparatively small (so the matrix is sparse), and we maximize the penalized log likelihood function

$$\hat{\Omega}(\lambda) = \arg\max_{\Omega \succ 0} \log L(\Omega; \mathbb{X}_n) - \lambda \|\Omega\|_1, \tag{1}$$

where $\|\Omega\|_1 = \sum_{i,j=1}^{d} |\omega_{ij}|$ is the L_1 matrix norm of Ω (with or without the diagonal entries) (Yuan and Lin 2007; Friedman et al. 2008; d'Aspremont et al. 2008; Rothman et al. 2008; Mazumder and Hastie 2012) and Ω is assumed to be positive definite. The magnitude of the regularization parameter $\lambda > 0$ controls the degree of sparsity.

Glasso performs well in the presence of a small number of large effects, i.e., a small number of large non-zero off-diagonal elements of Ω, which corresponds to large signals relative to noise. In microelectrode array recordings, however, we expect instead to find a large number of small and noisy effects. Indeed, using realistic settings for a numerical simulation (spike counts on coarse time scales $300 \sim 1000$ ms) we found that Glasso and existing variants perform poorly (see Fig. 1 and the simulation section). We therefore sought to enhance off-the-shelf regularization by including information that is specific to the neural setting. In this paper we introduce a variant of Glasso that takes advantage of known neurophysiology: the covariation of pairs of neurons' spike counts depends on their distance apart and their tuning curve correlation (Smith and Kohn 2008; Smith and Sommer 2013; Goris et al. 2014; Vinci et al. 2016). We use a Bayesian formulation of the problem to allow the penalty to vary with each neuron pair, separately, so that edges can become less likely to be placed between neurons as their distance apart increases or their tuning curve correlation decreases – the relationship of the penalty to these two covariates is learned from the data. We call the method Graphical lasso with Adjusted Regularization (**GAR**). Figure 1 illustrates the typical benefit of applying GAR in comparison with Glasso. We provide an extensive simulation study to compare GAR with several variants of Glasso that have appeared in the literature. We also show how the Bayesian approach provides an elegant framework to construct the graph, in a manner similar to false discovery rate regression (Scott et al. 2015). Finally, we apply the method to populations of neurons recorded with multielectrode arrays implanted in macaque visual cortex areas V1 and V4 to illustrate aspects of network behavior that can be discovered with this approach.

2 Results

We first describe several penalized likelihood methods for estimating Ω, including GAR (Sections 2.1 and 2.2). We then explain how to infer a neuronal network connectivity graph from the estimate of Ω (Section 2.3). Finally, we illustrate the properties of these methods in an extensive simulation study (Section 2.4), and we estimate the connectivity graphs of populations of neurons recorded with multielectrode arrays implanted in macaque visual cortex areas V1 and V4 (Section 2.5).

2.1 Estimating the precision matrix Ω

The **Glasso** estimate of Ω is obtained in Eq. (1) where

$$\log L(\Omega; \mathbb{X}_n) = \frac{n}{2}\left(\log\det(\Omega) - \text{trace}(\hat{\Sigma}\Omega) - d\log(2\pi)\right), \tag{2}$$

$\hat{\Sigma}$ is the sample covariance matrix of $\mathbb{X}_n$, and $\lambda > 0$ is chosen according to one of several possible criteria (Yuan and Lin 2007; Liu et al. 2010; Foygel and Drton 2010). Here we use the criterion of Fan et al. (2009), taking λ to minimize the cross-validated risk of estimating Ω with $\hat{\Omega}(\lambda)$, calculated as the average loss, $-\log L(\hat{\Omega}_{\text{train}}(\lambda); \mathbb{X}_{\text{test}})$, across 500 random (90%, 10%) splits of the data $\mathbb{X}_n = (\mathbb{X}_{\text{train}}, \mathbb{X}_{\text{test}})$, where $\hat{\Omega}_{\text{train}}(\lambda)$ is the estimate in Eq. (1) based on $\mathbb{X}_{\text{train}}$.

A variant is the adaptive Glasso (**AGlasso**) (Fan et al. 2009) given by

$$\hat{\Omega}(\lambda) = \arg\max_{\Omega \succ 0} \log L(\Omega; \mathbb{X}_n) - \lambda \|Q \odot \Omega\|_1, \tag{3}$$

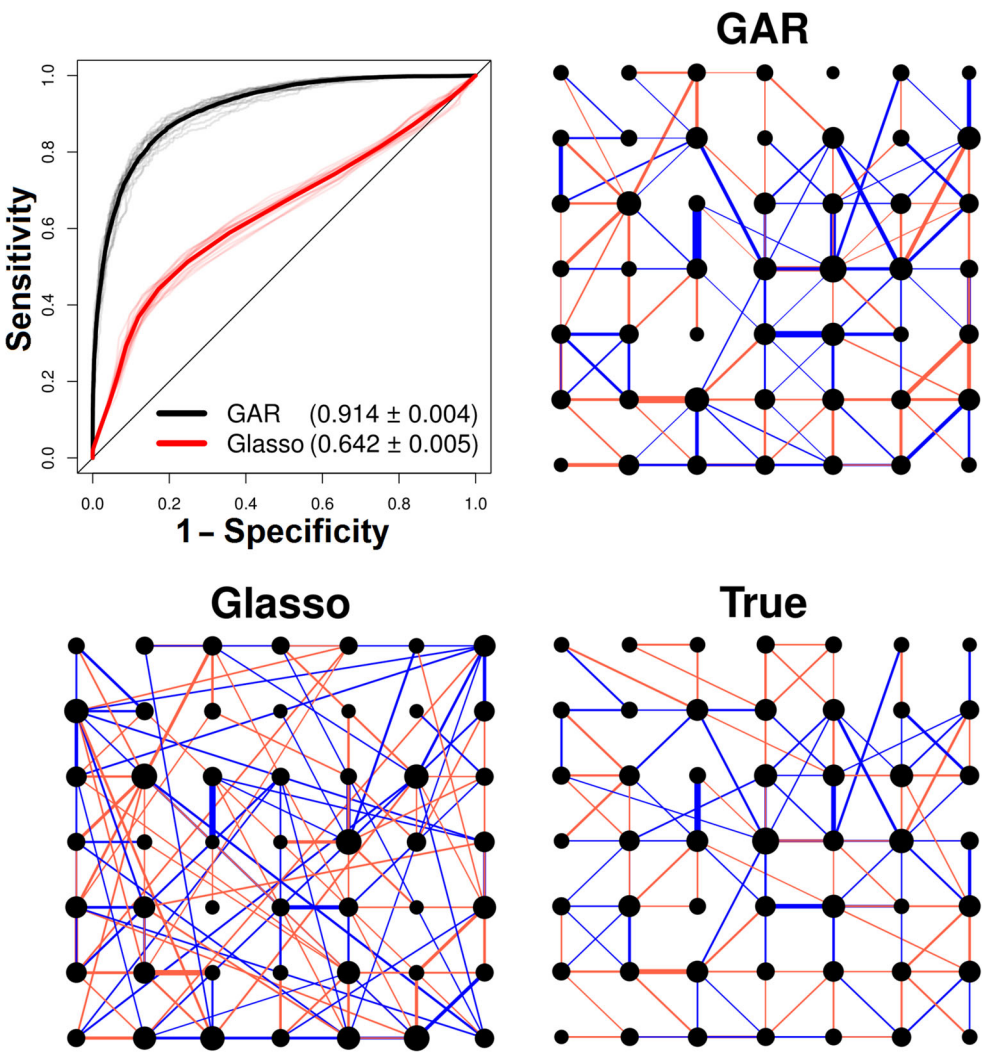

Fig. 1 Glasso and GAR graph estimation performances for a simulated network. The true graph contains $d = 49$ neurons and 118 edges, and is based on parameter settings derived from cortical data. Blue and red edges denote positive and negative partial correlations and the size of each node is proportional to the number of its connections. The GAR and Glasso estimates displayed are based on a sample size of $n = 200$ (see simulation Section 2.4 for details). For clarity, we show only the 118 strongest estimated connections (Glasso estimated 642 edges and GAR 204). ROC curves were obtained for 50 repeat simulated data; all 50 curves are plotted as thin lines and their averages as thick lines. The average area under the curves (AUC) is written in parentheses ± 2 simulation standard error: GAR is more accurate than Glasso

that is Eq. (1) but with the penalty $\|\Omega\|_1$ replaced by $\|Q \odot \Omega\|_1$, where $\odot$ denotes the entry-wise matrix multiplication and Q is a matrix containing values inversely related to the absolute values of an initial estimate $\hat{\Omega}$, for example $Q = [|\hat{\omega}_{ij}|^{-1/2}]$ with $\hat{\Omega} = [\hat{\omega}_{ij}]$ the inverse of the sample covariance matrix $\hat{\Sigma}$ (Fan et al. 2009). Hence, the AGlasso aims to penalize less/more the large/small entries of Ω. However, a reliable initial estimate of Ω is not always available; for instance, when the number of neurons d is greater than the number of trials n, $\hat{\Sigma}$ requires modifications to be inverted, such as adding a small constant to its diagonal to make it positive definite. Because the AGlasso estimate depends on the quality of the initial estimate of Ω, it does not necessarily outperform the Glasso estimate.

The Bayesian Adaptive Glasso (**BAGlasso**; Wang 2012) supplements the Gaussian likelihood with a prior distribution for Ω

$$\pi(\Omega \mid \Lambda) \propto \prod_{i,j=1}^{d} e^{-\lambda_{ij}|\omega_{ij}|} \times I(\Omega \succ 0) \tag{4}$$

where $\Lambda = [\lambda_{ij}]$ is a symmetric matrix that contains a different penalty for each ω_{ij}, to make it possible to penalize less/more the large/small ω_{ij}. The data automatically tunes the penalties if we assume a sufficiently flexible hyperprior for Λ, for example independent Gamma distributions for each λ_{ij} (Wang 2012). The BAGlasso estimate of Ω is taken to be the mean (posterior expectation) or the mode (maximum a posteriori, a.k.a. MAP) of $\pi(\Omega \mid \mathbb{X}_n)$, the posterior distribution of Ω. The BAGlasso estimator is not necessarily better than the Glasso or the AGlasso estimator because the added model flexibility also induces more variability. Note that the simple case where $\lambda_{ij} = \lambda$ for all (i, j) is known as the Bayesian Glasso (**BGlasso**; Wang 2012); furthermore the BGlasso MAP estimate of Ω for a fixed λ is the Glasso estimate in Eq. (1).

The Glasso framework in Eq. (1) can also be extended into the Sparse-Low rank model (**SPL**)

$$(\hat{S}, \hat{L}) = \underset{\substack{S-L \succ 0,\\ \text{rank}(L) \le q}}{\arg\max} \log L(S - L; \mathbb{X}_n) - \lambda \|S\|_1. \tag{5}$$

where $\Omega = S - L \succ 0$ is assumed to be the combination of a sparse component S representing the dependence structure of the recorded neurons conditionally on all other recorded and latent neurons in the network, and a low-rank component $L \succeq 0$ aimed at capturing the network effect of latent neurons on the recorded ones (Chandrasekaran et al. 2012; Giraud and Tsybakov 2012; Yuan 2012; Yatsenko et al. 2015). The parameters λ and q in Eq. (5) may be selected via cross-validation, analogously to Glasso. If L is set to zero, then Eq. (5) is equivalent to Glasso in Eq. (1). Partial correlations based on the component S alone would represent the conditional dependence structure of the observed neurons in an unknown larger network containing a set of unobserved units of intrisic dimensionality assumed to be smaller than q. However, for small sample sizes n and large numbers of neurons d, the sparse and low-rank components may become too expensive to estimate accurately so that SPL might perform no better than other methods (Giraud and Tsybakov 2012; Yatsenko et al. 2015).

Proposed methods The AGlasso and BAGlasso penalize less the Ω matrix entries that are anticipated to be larger, where the "anticipation" is explicitly garnered from an initial estimate of Ω for AGlasso, or implicitly data driven in the BAGlasso. Sometimes we have available additional variables that carry information about the strength of the dependence between neurons, for example inter-neuron distance and tuning curve correlation, which have been observed to regulate the shared activity of neuron pairs (Smith and Kohn 2008; Smith and Sommer 2013; Goris et al. 2014; Yatsenko et al. 2015; Vinci et al. 2016). Our proposed methods take advantage of these known, strong effects by allowing the penalties to depend on them: for example, pairs of neurons that are close together will be penalized less than pairs that are far apart (see data analysis, Section 2.5).

Let $W_{ij} \in \mathbb{R}^m$ be a vector of m auxiliary quantities that carry information about the strength of the dependence between neurons i and j. The Smooth Adaptive Glasso (**SAGlasso**) is a "smooth" variant of the AGlasso, where the entries q_{ij} of the weight matrix Q in Eq. (3) are taken to be smooth functions of W_{ij}. This smoothing will both reduce the noise in Q and introduce the information carried by W_{ij} into the regularized estimation of Ω. To proceed, we regress $q_{ij}/\sqrt{q_{ii}q_{jj}}$ on W_{ij} using a local smoother such as smoothing splines or local polynomials, where we divide the q_{ij} by $\sqrt{q_{ii}q_{jj}}$ so that they are on the scale of the partial correlations

$$\rho_{ij} = -\frac{\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}}. \tag{6}$$

To ensure that the resulting weights q_{ij} are positive, we either perform a Gamma regression (Algorithm 1, Appendix), or log transform $q_{ij}/\sqrt{q_{ii}q_{jj}}$ first if a Gaussian regression is used.

The Graphical lasso with Adjusted Regularization (**GAR**) is a variant of the BAGlasso, where we impose additional structure on λ_{ij} in Eq. (4). Specifically, we assume that the penalties are functions of the auxiliary variables W_{ij} according to:

$$\lambda_{ij} = \alpha_i \alpha_j g(W_{ij}), \quad i < j, \tag{7}$$

where the function $g(W_{ij})$ is fitted to the data rather than pre-specified, and the α_i's are positive parameters that mimic the scaling components $\omega_{ii}^{-1/2}$ in Eq. (6), so that $g(W_{ij})$ is on the scale of the partial correlation ρ_{ij}. We render g identifiable by setting $\lambda_{ii} = \alpha_i^2$, which is reasonable since α_i mimics $\omega_{ii}^{-1/2}$, and the mean parameter of ω_{ii} in Eq. (4) is λ_{ii}^{-1}. Then, while the BAGlasso postulates a hyperprior for Λ in Eq. (4), here we assume a hyperprior on the parameters of Λ in Eq. (7), that is on $\Theta = \{\alpha, g\}$, where $\alpha = (\alpha_1, \ldots, \alpha_d)$, and g is the m-variate function of the auxiliary variables W. The GAR full Bayes estimate of Ω is taken to be the mean or the mode of $\pi(\Omega \mid \mathbb{X}_n, W)$, the posterior distribution of Ω given the data $\mathbb{X}_n$ and auxiliary covariates $W = \{W_{ij}\}$. Ideally, g should be as general a function as possible. However, the model is complex enough to make the posterior difficult to calculate or simulate from so we opted to use the simplest of non-parametric functions: a step function. In Section 2.2.2 we describe a Gibbs sampler to simulate from $\pi(\Omega, \Theta \mid \mathbb{X}_n, W)$, and thus obtain sample mean or mode estimates of Ω. An alternative that allows a general form of g is to take an empirical Bayes approach to obtain a point estimate of Θ as the maximizer of the likelihood of Θ given $\mathbb{X}_n$ and W, and calculate the posterior distribution of Ω conditional on that estimate, that is $\pi(\Omega \mid \mathbb{X}_n, W, \hat{\Theta})$, instead of calculating the full Bayes posterior. We implement this approach in Section 2.2.3 and apply it to our data in Section 2.5, taking g to be a regression spline with knots at the quartiles of the auxiliary variables. In the end, we have the full Bayes and the empirical Bayes GAR variants, and we can take the estimate of Ω to be either the mean or the mode of the corresponding posterior distribution. In our simulations (Section 2.4) we show that the improvement provided by GAR can be substantial compared to the competing methods, especially for values of d and n typically found in experimental neural data.

The proposed and existing methods considered here are similarly scalable. All methods aim to estimate the $d(d+1)/2$ parameters of the precision matrix Ω, but their regularizations have different complexities: Glasso uses a single regularization parameter, λ in Eq. (1), and SPL uses two, λ and q in Eq. (5), while AGlasso and BAGlasso allow $d(d+1)/2$ different penalties across the entries of

Ω. SAGlasso and GAR also allow a different penalty for each entry of Ω, but their effective number is smaller than $d(d+1)/2$. Indeed, the SAGlasso penalties all depend on W through a regression function that depends only on a limited number of regression coefficients. For GAR, all penalties are functions only of g and the d parameters α_i (Eq. (7)), where g is a regression function with a few degrees of freedom, e.g. splines with 3 or 4 knots. Morever, SAGlasso and GAR gain statistical efficiency when the auxiliary variables W are informative.

Finally, we note that a simple way to combine GAR with SPL (**GAR-SPL**) consists of replacing the penalty $\lambda\|S\|_1$ in Eq. (5) by $\xi\|\hat{\Lambda} \odot S\|_1$, where $\hat{\Lambda}$ is a GAR estimate of $\Lambda \mid (\mathbb{X}_n, W)$, and ξ and q are selected via cross-validation. In our simulations, GAR-SPL outperformed SPL but not GAR. A full Bayesian treatment of GAR-SPL, where the penalty matrix is estimated in direct combination with S rather than Ω, might provide a better performance; this is a topic of future research.

2.2 GAR estimation

In Sections 2.2.2 and 2.2.3 we describe Full and Empirical Bayes implementations of GAR, which both involve a data augmentation sampler that we present in Section 2.2.1. Algorithms and details are in Appendix.

2.2.1 Data augmentation

Both Full and Empirical Bayes implementations of GAR involve drawing samples from the posterior distribution

$$\pi(\Omega \mid \mathbb{X}_n, \Lambda) \propto \pi(\Omega \mid \Lambda) \times L(\Omega; \mathbb{X}_n), \tag{8}$$

where the likelihood $L(\Omega; \mathbb{X}_n)$ and the prior $\pi(\Omega \mid \Lambda)$ are defined in Eqs. (2) and (4). We proceed using a data augmentation strategy (Wang 2012) where we introduce the nuisance random quantity $\mathcal{T} = \{\tau_{ij}\}_{i<j}$, and jointly sample $(\Omega, \mathcal{T})$ from

$$\pi(\Omega, \mathcal{T} \mid \mathbb{X}_n, \Lambda) \propto \pi(\Omega, \mathcal{T} \mid \Lambda) \times L(\Omega; \mathbb{X}_n) \tag{9}$$

using the block Gibbs Sampler in Appendix, Algorithm 2, where

$$\pi(\Omega, \mathcal{T} \mid \Lambda) \propto \pi(\Omega \mid \mathcal{T}, \Lambda) \times \pi(\mathcal{T} \mid \Lambda), \tag{10}$$

$$\pi(\Omega \mid \mathcal{T}, \Lambda) = \prod_{i<j} \varphi\left(\omega_{ij} \mid 0, \tau_{ij}\lambda_{ij}^{-2}/2\right) \times \prod_{i=1}^{d} \gamma(\omega_{ii} \mid 1, \lambda_{ii}) \times I(\Omega \succ 0) \times C(\mathcal{T}, \Lambda), \tag{11}$$

$$\pi(\mathcal{T} \mid \Lambda) \propto C(\mathcal{T}, \Lambda)^{-1} \prod_{i<j} \gamma(\tau_{ij} \mid 1, 1), \tag{12}$$

$\varphi(u \mid \mu, \sigma^2)$ is the Gaussian p.d.f. with mean μ and variance σ^2, $\gamma(z \mid a, b)$ is the Gamma p.d.f. with shape and rate parameters a and b, and $C(\mathcal{T}, \Lambda)$ is the finite normalizing constant of Eq. (11). Then, because the Laplace distribution is a Gaussian scale mixture (Andrews and Mallows 1974; West 1987), we can write Eq. (8) as the integral of Eq. (9):

$$\pi(\Omega \mid \mathbb{X}_n, \Lambda) = \int \pi(\Omega, \mathcal{T} \mid \mathbb{X}_n, \Lambda)\, d\mathcal{T},$$

so that the matrix Ω in a sample $(\Omega, \mathcal{T})$ drawn from Eq. (9) is a sample from the posterior distribution in Eq. (8).

2.2.2 Full Bayes estimation

Let g in Eq. (7) be a step function with K steps and value $\beta_k > 0$ in step k, that is:

$$g(W_{ij}) = \sum_{k=1}^{K} \beta_k I_{A_k}(W_{ij}), \tag{13}$$

where I is the indicator function, $\forall h \neq l$, $A_h \cap A_l = \emptyset$, and $\cup_{k=1}^{K} A_k = \mathbb{R}^m$. Let $\alpha = (\alpha_1, ..., \alpha_d)$, $\beta = (\beta_1, ..., \beta_K)$, and $\Theta = \{\alpha, \beta\}$, where $\lambda_{ii} = \alpha_{ii}^2$. Following the augmentation strategy in Section 2.2.1, Eq. (10) reduces to

$$\begin{aligned} &\pi(\Omega, \mathcal{T} \mid \Theta, W) \\ &= \prod_{i<j} \varphi\left(\omega_{ij} \mid 0, \frac{\tau_{ij}}{2\alpha_i^2\alpha_j^2 \sum_{k=1}^{K} \beta_k^2 I_{A_k}(W_{ij})}\right) \gamma(\tau_{ij} \mid 1, 1) \\ &\quad \times \prod_{i=1}^{d} \gamma(\omega_{ii} \mid 1, \alpha_i^2) \times I(\Omega \succ 0) \times G(\Theta) \end{aligned} \tag{14}$$

where $G(\Theta)$ is the finite normalizing constant. We further assume the hyperprior density on Θ

$$\pi(\Theta) \propto \prod_{i=1}^{d} \alpha_i^{r-1} e^{-s\alpha_i^2} \times \prod_{k=1}^{K} \beta_k^{r'-1} e^{-s'\beta_k^2} \times G(\Theta)^{-1} \tag{15}$$

with $r = s = 1$, $r' = 0.01$, and $s' = 0.00001$; results were not sensitive to the choice of these parameters. We use the Gibbs sampler in Appendix, Algorithm 3, to sample from the full joint posterior distribution

$$\pi(\Omega, \Theta, \mathcal{T} \mid \mathbb{X}_n, W) \propto \pi(\Omega, \Theta, \mathcal{T} \mid W) \times L(\Omega; \mathbb{X}_n), \tag{16}$$

where the likelihood $L(\Omega; \mathbb{X}_n)$ is defined in Eq. (2), and the prior joint distribution $\pi(\Omega, \Theta, \mathcal{T} \mid W)$ is the product of Eqs. (14) and (15).

In practice, we take the number of steps K in Eq. (13) to be relatively small, e.g. 4 or 5, if we expect the penalties to change relatively slowly with W. Otherwise, to increase the flexibility of g while ensuring that enough data points contribute to estimating each β_k, we can afford of take $K \approx \sqrt{d}$ since $d(d+1)/2$ values of W are available. We further locate the steps at evenly spaced empirical quantiles

of the W's, using a hierarchical quantile splitting when W is multi-dimensional, so that each step contains approximately the same number of W's. If no auxiliary quantity W is available, full Bayes GAR can still be applied by setting $K = 1$. If we wanted to constrain g to be monotonic increasing, we could enrich the prior in Eq. (15) with the factor $\prod_{k=2}^{K} I(\beta_{k-1} < \beta_k)$, which requires β_k to be sampled from the same distribution at step 2 of Algorithm 3, but truncated to be within the interval $(\beta_{k-1}, \beta_{k+1})$, where $\beta_0 = 0$ and $\beta_{K+1} = \infty$. Different features of g may be imposed in similar ways.

2.2.3 Empirical Bayes estimation

By assuming Eq. (7) and $\lambda_{ii} = \alpha_i^2$, Eq. (4) reduces to

$$\pi(\Omega \mid \Theta, W) = \prod_{i<j} \alpha_i \alpha_j g(W_{ij}) e^{-2\alpha_i \alpha_j g(W_{ij})|\omega_{ij}|} \times \prod_{i=1}^{d} \alpha_i^2 e^{-\alpha_i^2 \omega_{ii}} \times I(\Omega \succ 0) \times G(\Theta) \tag{17}$$

where $\Theta = \{\alpha, g\}$, g is a positive function of any form estimable in a Gamma regression, and $G(\Theta)$ is the normalizing constant. We further assume the prior density

$$\pi(\Theta) \propto p(\Theta) \times G(\Theta)^{-1} \tag{18}$$

where $p(\Theta)$ is a density on Θ. We estimate Θ by maximizing the posterior density

$$\pi(\Theta \mid \mathbb{X}_n, W) = \int_{\Omega \succ 0} \pi(\Omega, \Theta \mid \mathbb{X}_n, W)\, d\Omega \tag{19}$$

using an Expectation-Maximization algorithm (Dempster et al. 1977; Gelman et al. 2004):

– E-STEP: Given the current estimate Θ^{old}, we compute the expectation

$$\mathbb{E}[\log \pi(\Omega, \Theta \mid \mathbb{X}_n, W) \mid \mathbb{X}_n, \Theta^{\text{old}}, W], \tag{20}$$

with respect to $\Omega \sim \pi(\Omega \mid \mathbb{X}_n, \Theta^{\text{old}}, W)$, which reduces to $c + Q(\Theta \mid \Theta^{\text{old}})$, where c is a constant of Θ and

$$Q(\Theta \mid \Theta^{\text{old}}) = \sum_{i<j} \left\{ \log[\alpha_i \alpha_j g(W_{ij})] - 2\alpha_i \alpha_j g(W_{ij}) \bar{\omega}_{ij} \right\} + \sum_{i=1}^{d} \{2 \log \alpha_i - \alpha_i^2 \bar{\omega}_{ii}\} + \log p(\Theta),$$

where $\bar{\omega}_{ij} = \mathbb{E}[|\omega_{ij}| \mid \mathbb{X}_n, \Theta^{\text{old}}, W]$ can be approximated using the Gibbs sampler (Section 2.2.1).

– M-STEP: $Q(\Theta \mid \Theta^{\text{old}})$ is concave with respect to g and $\alpha_i, i = 1, ..., n$, so we can maximize it with respect to Θ by circularly optimizing with respect to α and g until convergence, as follows: assuming $p(\Theta) \propto 1$ in Eq. (18), $\frac{\partial Q}{\partial \alpha_i} = 0$ subject to $\alpha_i > 0$ yields the maximizer

$$\alpha_i = \left(\sqrt{\eta^2 + 8\bar{\omega}_{ii}(d+1)} - \eta \right) / (4\bar{\omega}_{ii}), \tag{21}$$

where $\eta = 2 \sum_{j \neq i} \alpha_j g(W_{ij}) \bar{\omega}_{ij}$, and the maximizing function g is obtained by regressing $y_{ij} = 2\alpha_i \alpha_j \bar{\omega}_{ij}$ on $W_{ij}, i < j$, in a Gamma regression model.

We summarize the procedure for the case $p(\Theta) \propto 1$ in Appendix, Algorithm 4.

2.3 Estimating a connectivity graph

Here, we explain how to estimate graphs based on partial correlation estimates, so as to control the edge false discovery and false nondiscovery rates.

Let $E = \{e_{ij}\}$ be a true graph, where $e_{ij} = 1$ when nodes i and j are connected by an edge, and $e_{ij} = 0$ otherwise. We build graph estimators of two kinds:

1. A δ**-graph** has edges $\hat{E}_\delta = \{\hat{e}_{ij}(\delta)\}$ such that, for a threshold $\delta \in [0, 1)$,

$$\hat{e}_{ij}(\delta) = \begin{cases} 1, & \text{if } |\hat{\rho}_{ij}| > \delta \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

where $\hat{\rho}_{ij} = -\hat{\omega}_{ij} / \sqrt{\hat{\omega}_{ii} \hat{\omega}_{jj}}$ is a point estimate of the partial correlation in Eq. (6). That is, an edge is present in the graph estimate if the corresponding estimated absolute partial correlation $|\hat{\rho}_{ij}|$ exceeds some threshold δ.

2. A (p, δ)**-graph** has edges $\hat{E}_{p,\delta} = \{\hat{e}_{ij}(p, \delta)\}$ such that, for thresholds $p, \delta \in [0, 1)$,

$$\hat{e}_{ij}(p, \delta) = \begin{cases} 1, & \text{if } \pi_{ij}(\delta) > p \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

where

$$\pi_{ij}(\delta) = \mathbb{P}(|\rho_{ij}| > \delta \mid \mathbb{X}_n, W) \tag{24}$$

is the edge posterior probability, that is the posterior probability that the magnitude of the partial correlation ρ_{ij} exceeds δ. The (p, δ)-graph uses the full posterior probability of the partial correlations rather than the point estimates $\hat{\rho}_{ij}$, setting an edge to zero if the edge posterior probability (Eq. (24)) is smaller than p. In our simulations, (p, δ)-graphs were often more precise than δ-graphs.

One must choose values for δ and p. Using $\delta = 0$ in Eq. (22) could produce a completely dense graph, while using a large value would only identify strong edges, which is not sensible for neural networks since connections can be small and numerous. Ideally, δ should be as close as possible to the minimum magnitude of the true non-zero partial

correlations; we take a robust estimate of that minimum to be the 5-th quantile of the magnitudes of the non-zero MAP partial correlation estimates. We use the same δ in Eq. (23), and choose p to control the false discovery and non-discovery rates of the graph edges (FDR and FNR, respectively), that is the rate of false detected edges (number of (i, j) such that $\hat{e}_{ij} = 1$ but $e_{ij} = 0$) out of all detections (number of $\hat{e}_{ij} = 1$) and the rate of true missed connections (number of (i, j) such that $\hat{e}_{ij} = 0$ but $e_{ij} = 1$) out of all non-detections (number of $\hat{e}_{ij} = 0$):

$$\text{FDR} = \mathbb{E}\left[\frac{\sum_{i<j}\hat{e}_{ij}(1-e_{ij})}{\sum_{i<j}\hat{e}_{ij}}\right] \tag{25}$$

and

$$\text{FNR} = \mathbb{E}\left[\frac{\sum_{i<j}(1-\hat{e}_{ij})e_{ij}}{\sum_{i<j}(1-\hat{e}_{ij})}\right], \tag{26}$$

where the expectations are taken with respect to the data $\mathbb{X}_n = \{X^{(1)}, ..., X^{(n)}\}$. Then for a fixed δ, a (p, δ)-graph can be selected by choosing either

$$p^* = \min\{p : \text{FDR} \le C\} \quad \text{or} \quad p^{**} = \max\{p : \text{FNR} \le C\}$$

for some desired upper bound C, depending on whether we want to control the FDR or the FNR, or choose p to balance the FDR and FNR, that is FDR $\approx$ FNR (the intersection of the green and red curves in Fig. 2). However, the FDR and FNR are easy to approximate in simulations when the true graph E is known, but not otherwise. An alternative is to calculate their Bayesian counterparts, obtained by Eqs. (25) and (26) but with expectation taken conditionally on the data $\mathbb{X}_n$, which yields

$$\text{FDR}_\Pi = \frac{\sum_{i<j}\hat{e}_{ij}(1-\pi_{ij}(\delta))}{\sum_{i<j}\hat{e}_{ij}} \tag{27}$$

and

$$\text{FNR}_\Pi = \frac{\sum_{i<j}(1-\hat{e}_{ij})\pi_{ij}(\delta)}{\sum_{i<j}(1-\hat{e}_{ij})}, \tag{28}$$

where $\pi_{ij}(\delta)$ is the edge posterior probability defined in Eq. (24). Equation (27) has the same form as the Bayesian FDR in multiple hypothesis testing (Efron et al. 2001; Efron 2007) and FDR-regression (Scott et al. 2015), obtained as the average of the local FDRs, i.e. posterior probabilities that the hypotheses are null, across the rejections. In our framework $(1 - \pi_{ij}(\delta))$ is the local FDR for the pair (i, j). In Fig. 2 we show by simulation that bounding FDR_Π or FNR_Π also appears to bound their frequentist counterparts. This result is not surprising because empirical estimates of the Bayesian FDR are typically upward biased estimates of the frequentist FDR (Efron and Tibshirani 2002).

ROC curves summarize graph estimation performance Other useful measures of detection error are sensitivity and specificity

$$\text{SENS} = \frac{\sum_{i<j}\hat{e}_{ij}e_{ij}}{\sum_{i<j}e_{ij}} \tag{29}$$

and

$$\text{SPEC} = \frac{\sum_{i<j}(1-\hat{e}_{ij})(1-e_{ij})}{\sum_{i<j}(1-e_{ij})}, \tag{30}$$

which give the proportions of true edges correctly identified and of missing edges correctly omitted, respectively. By tuning the parameters defining the estimated edges $\hat{e}_{ij}$'s, for example λ in Eq. (1) and p in Eq. (23), we can obtain the curve of SENS versus 1- SPEC, known as Receiver Operating Characteristic (ROC) curve (see Figs. 1 and 3). A point above the ROC curve denotes an edge detection performance that cannot be achieved by the estimator, i.e. no values of the tuning parameters could make the estimator produce that outcome. Therefore, a larger area under a ROC curve (AUC) indicates a better edge detection performance;

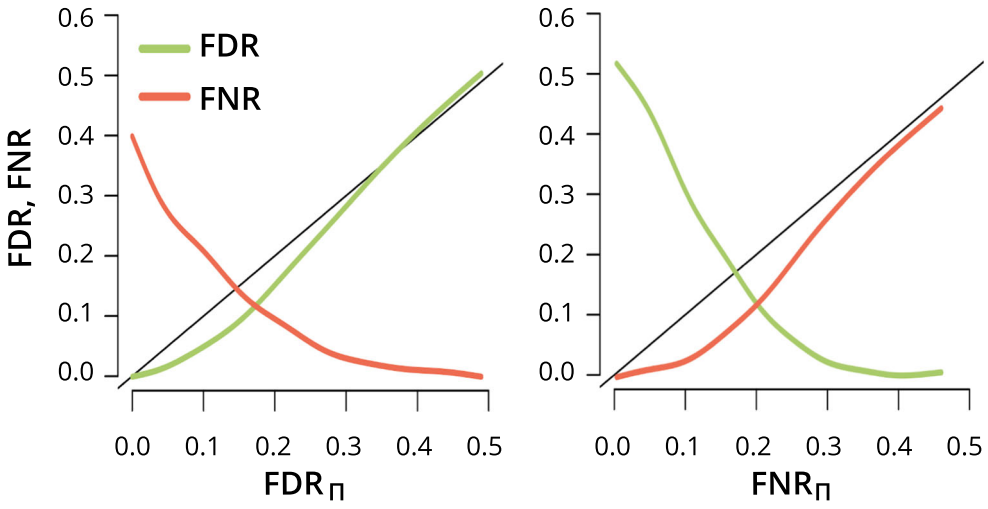

Fig. 2 Bayesian FDR and FNR, FDR_Π and FNR_Π, plotted on the x-axes, control their frequentist counterparts FDR and FNR, plotted on the y-axes, in a simulation based on $d = 100$ neurons and sample size $n = 500$ (Section 2.4). The black lines are the first bisectors $x = y$

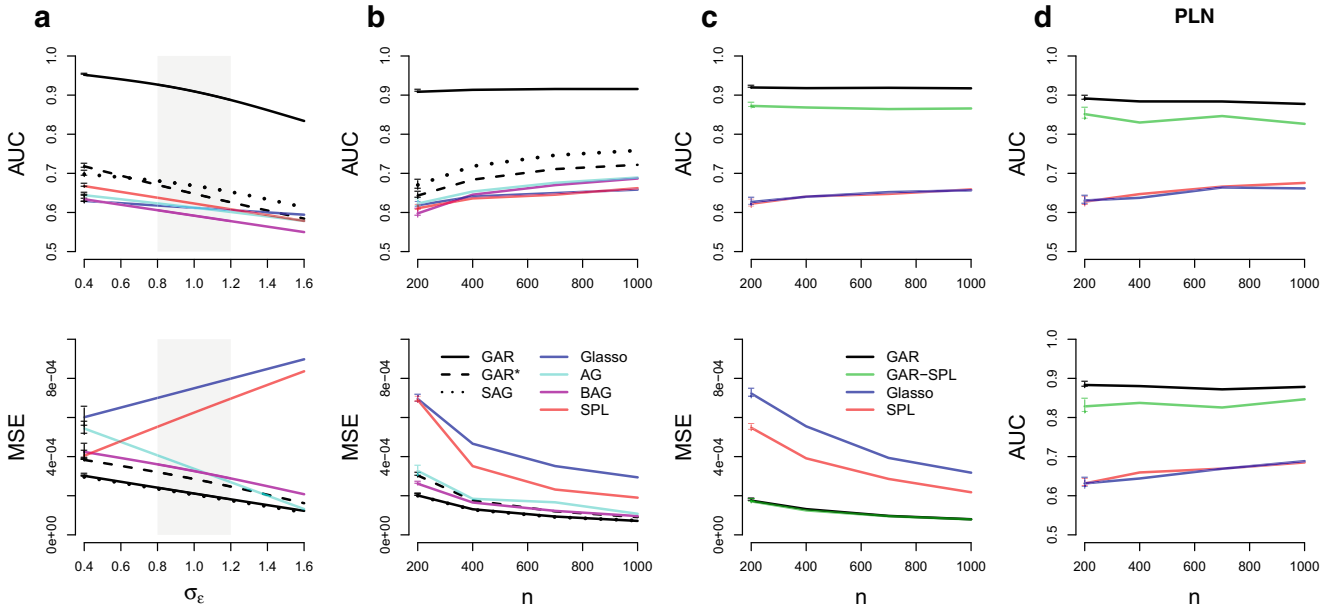

Fig. 3 Performance of graph estimates, measured by areas under ROC curves (AUC), and of partial correlation estimates, measured by MSEs, for $d = 50$ neurons simulated according to Eq. (31) as functions of σ_ϵ in **a**, where $\sigma_\epsilon \approx 1$ (shaded grey area) represents a real data scenario, and as functions of the sample size n in **b**. **c** AUC and MSE for $d = 50$ observed neurons conditionally on $q = 20$ latent ones as functions of n. **d** AUC when methods are applied to non-Gaussian Poisson-lognormal (PLN) data without (top) and with (bottom) latent neurons, as functions of n. The vertical intervals on the left of all curves are 95% simulation intervals

we use this metric to compare graph estimators in our simulations (Figs. 1 and 3).

2.4 Statistical properties of the estimates of Ω in simulated data

We compared the performance of the various estimators of Ω in an extensive simulation study. We simulated data sets of n d-dimensional Gaussian vectors $X^{(1)}, ..., X^{(n)} \overset{\text{iid}}{\sim} N(\mu, \Omega^{-1})$, with μ chosen to match typical values found in experimental data, and with $\Omega = [\omega_{ij}]$ generated as follows: for $i < j$,

$$\begin{cases} \omega_{ij} = Z_{ij} \exp(bW_{ij} + \epsilon_{ij}) \\ Z_{ij} \sim I(z = -1)\eta + I(z = 1)(1 - \eta) \\ \epsilon_{ij} \sim N(0, \sigma_\epsilon^2) \end{cases} \tag{31}$$

where $I(A) = 1$ if A is true and $I(A) = 0$ otherwise, so that $Z_{ij} \in \{-1, 1\}$ makes ω_{ij} negative with probability η and positive with probability $1 - \eta$. We use a simple auxiliary variable W_{ij}, which we take to be the physical distance between neurons i and j on a 4 mm × 4 mm Utah electrode array, and used $b = -2$, $\eta = 0.75$, and $\sigma_\epsilon = 1$ to generate values of ω_{ij} that are consistent with the experimental data analyzed in Section 2.5. We then symmetrized Ω by setting $\omega_{ji} = \omega_{ij}$, and set the diagonal entries ω_{ii} to the smallest positive ω^* that rendered Ω positive definite. We achieved sparsity by resetting to zero the smallest half of the partial correlations so that half of the graph edges were null, and rescaled $\Sigma = \Omega^{-1}$ so its diagonal elements would have magnitude similar to the experimental data analyzed in Section 2.5.

Figure 1 shows a simulated network of $d = 49$ neurons with dependence structure in Eq. (31), the Glasso graph estimate, and the empirical Bayes GAR estimate (Algorithm 4) with g (Eq. (7)) taken to be a regression spline: GAR uses the spatial information of the inter-neuron distances and yields more accurate connectivity graph estimates. Here and in the rest of the paper, all GAR parameters, including the penalty parameters, are assigned a prior distribution and are thus selected implicitly. Similarly for the other Bayesian methods. For the non-Bayesian methods, Glasso, AGlasso, SAGlasso, and SPL, the penalty parameters are selected by cross validation (Fan et al. 2009).

Figure 3a shows the AUC and mean squared error (MSE) of the partial correlation matrix estimate for a simulated network of $d = 50$ neurons and sample size $n = 200$, for moderate deviations from $\sigma_\epsilon = 1$ in Eq. (31), where the MSE is the average of the squared error, $\sum_{i<j}(\hat{\rho}_{ij} - \rho_{ij})^2/(2d(d-1))$, across repeat simulations. We show only the performance of the full Bayes GAR estimate; it is comparable to that of the empirical Bayes GAR estimate but faster to compute. Figure 3b shows how AUC and MSE vary with the sample size n for fixed $\sigma_\epsilon = 1$ and $d = 50$. GAR outperforms all other methods in Fig. 3a and b. The SAGlasso is the next best method. It is easier to implement

than GAR but provides only a modest improvement over other methods.

Figures 3a and b also show the performance of GAR*, the full Bayes estimate that uses a constant $g(w)$ in Eq. (7). Comparing GAR and GAR* shows the added benefit of including auxiliary information within our novel Bayesian framework. Note that GAR* appears to outperform the Glasso and its variants even without adapting the regularization to the covariate. This may be due to the parameters α_i (Eq. (7)) forcing the penalty to be on the standardized scale of the partial correlations rather than on the scale of the precision matrix entries, which are known to be highly sensitive to the variance of the Gaussian vector X (Yuan and Lin 2007). It is also possible that, while the Glasso variants AGlasso and BAGlasso also attempt to attenuate the data scaling effect on the regularization, they involve $d(d+1)/2$ regularization parameters (the entries of Q in Eq. (3) and the λ_{ij}'s in Eq. (4)), whereas GAR* achieves the same goal with only $d+1$ parameters (the α_i's and the constant g), which might reduce the variance of the partial correlation estimates.

The problem of estimating pairwise dependences conditionally on both observed and latent units (here neurons) has been dealt with previously by applying SPL in Eq. (5). In Fig. 3c we compare the performances of GAR, Glasso, SPL, and GAR-SPL to estimate the dependence structure of $d = 50$ recorded neurons when an additional $q = 20$ neurons belong to the network but their activity is not observed. GAR outperformed SPL, likely because the information extracted from the inter-neuron distance overcompensated the missed information about the activity of the latent variables. SPL outperformed Glasso only in terms of MSE. The variant GAR-SPL outperformed SPL but not GAR, likely because of the additional variance due to the estimation of the low-rank component. However, a full Bayesian treatment of GAR-SPL might provide a better performance; this is a topic of future research.

Figure 3d repeats the AUC curves of Fig. 3b and c but with non-Gaussian data generated from the multivariate Poisson-lognormal distribution (Vinci et al. 2016)

$$\begin{aligned} X_i^{(r)} \mid Z_i^{(r)} &\sim \text{Poisson}\left(e^{Z_i^{(r)}}\right),\ i = 1, ..., d \\ (Z_1^{(r)}, ..., Z_d^{(r)})' &\sim N(\mu, \Sigma), \end{aligned} \tag{32}$$

where the spike counts $X^{(r)} = (X_1^{(r)}, ..., X_d^{(r)})$ on trial r are independent given their latent log-rates $Z_1^{(r)}, ..., Z_d^{(r)}$, and μ and Σ were set to match typical values found in experimental spike count data ($\mu_i \approx 2$, $\Sigma_{ii} \approx 0.25$, implying about 8.37 spikes/s on average). Under this model, dependences between spike counts are weaker than those of their latent rates due to Poisson-noise corruption (Vinci et al. 2016; Behseta et al. 2009). Since the dependences between the log-rates likely provide a better representation of input correlation (Vinci et al. 2016), we estimate the neuronal graph based on the partial correlations $\Omega = \Sigma^{-1}$ of the latent rates, applying first the square root transformation to the spike counts to improve their fit to a Gaussian distribution (Kass et al. 2014; Georgopoulos and Ashe 2000; Yu et al. 2009). Figure 3d shows that GAR outperformed all other methods, presumably because it successfully extracted the connectivity information carried by inter-neuron distance despite the Poisson noise. Also as in Fig. 3c, GAR-SPL outperformed SPL but not GAR. We repeated the simulation of Fig. 3d with values of μ_i small enough to induce neurons' firing rates of about 0.1 spikes/s and, compared to Fig. 3d, noted a loss of AUC of about 5% for GAR and GAR-SPL, and about 20% for all other methods that do not use auxiliary information.

2.5 Estimating neural connectivity in macaque visual cortex

Spike data were recorded from the V1 and V4 visual cortex of two *rhesus macaque* monkeys using 100-electrode Utah arrays. For the V1 data (Kelly et al. 2010; Scott et al. 2015; Cowley et al. 2016), visual stimuli were presented to an anesthetized animal. The stimuli were either a 30s sequence of drifting sinusoidal gratings (98 different orientations and two blanks, 300ms each), or blank gray screen (Fig. 4a). The 30s stimuli sequence was randomly ordered, and then repeated in that same order 120 times. For the V4 data, the visual stimuli were either vertical drifting sinusoidal gratings or a blank gray screen (Fig. 4b). Each trial began with the animal fixating a small dot for 150ms before the grating or blank screen was presented for 1000ms. Then the stimulus and fixation point were extinguished and the animal received a liquid reward for making an eye movement to a target 8 degrees from fixation in a random direction. Each stimulus (the vertical grating or the blank) was repeated 126 times. All procedures were approved by the Institutional Animal Care and Use Committee of the University of Pittsburgh and Albert Einstein College of Medicine, and were in compliance with the guidelines set forth in the National Institutes of Health's Guide for the Care and Use of Laboratory Animals.

We applied GAR to the neurons' spike counts in the repeated trials of duration 300ms and 1000ms in the V1 and V4 experiments, respectively, to estimate the connectivity at the trial time scale. We square-root transformed the spike counts to mitigate the dependence between variance and mean and thus improve their fit to Gaussian distribution we assume in this paper.

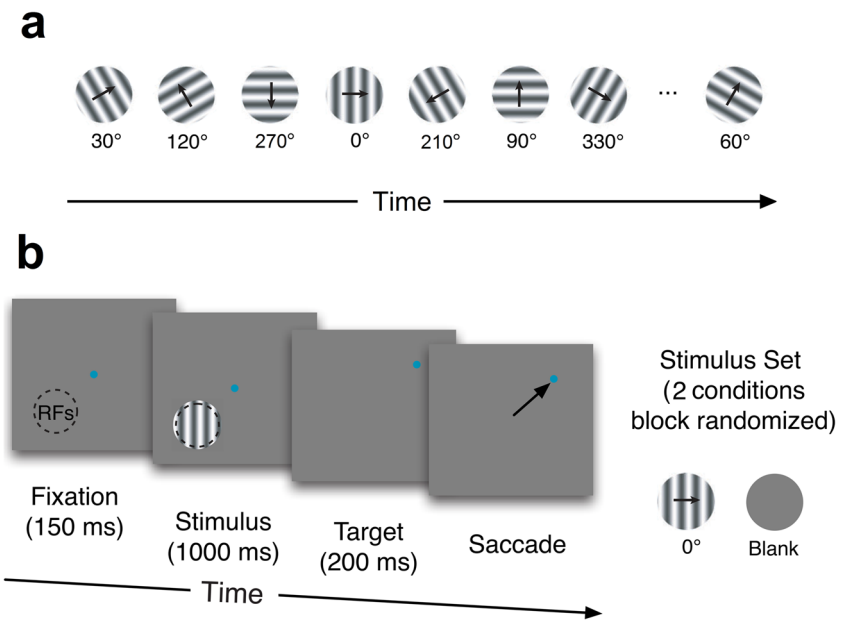

Fig. 4 Experiments. **a** The V1 stimuli consist of 30 second sequences of 98 randomly ordered oriented drifting gratings, or a blank gray screen. **b** The V4 stimuli consist of a blank gray screen or a vertical drifting sinusoidal grating appearing in the aggregate receptive field of the V4 neurons (RFs, indicated by the dashed circle)

V1 data We obtained recordings for 128 candidate neuronal units by sorting the voltage signals of the 76 electrodes with the best signal to noise ratio (Kelly et al. 2007), in response to a sequence of 98 drifting sinusoidal gratings and blank screen (Fig. 4a). Previous analyses of these data have been published (Kelly et al. 2010; Scott et al. 2015; Cowley et al. 2016). To produce easily readable graphs, we analysed only 100 neurons, selected as follows: we first retained the highest spiking neuron on each of the 76 electrodes that had identifiable action potentials, and then added the 24 highest spiking remaining neurons. The firing rates of these 100 neurons ranged from 0.61 to 31.97 spikes/s, with mean 6.27, and 2.5th and 97.5th percentiles 1.12 and 16.20.

Neurons in V1 are driven by drifting gratings of orientation $\theta \in (0, 360]$ (Scott et al. 2015), and their average firing rates are usually described by sinusoidal tuning functions of θ, with similar tuning in diametrically opposite orientations θ and $\theta + 180$ degrees (Butts and Goldman 2006; Smith and Kohn 2008; Scott et al. 2015; Vinci et al. 2016). Hence maximal firing rates occur at θ^* and $\theta^* + 180$, for some θ^*, and minimal firing rates occur in the orthogonal orientations, $\theta^* \pm 90$. The tuning similarity of two neurons can be quantified by their tuning curve correlation (TCC), computed as the Pearson's correlation of the two neurons' tuning curves across stimuli (Smith and Kohn 2008; Smith and Sommer 2013; Ecker et al. 2014; Kass et al. 2014; Vinci et al. 2016). The strength of the dependence between two neurons' activities has been observed to increase with TCC and decrease with inter-neuron distance (DIST) (Smith and Kohn 2008; Smith and Sommer 2013; Goris et al. 2014; Vinci et al. 2016), so we considered these two auxiliary quantities for our GAR connectivity graph estimation.

Figure 5 shows the results of GAR applied to 300 ms square root transformed spike counts for the vertical grating ($\theta = 0$) and blank screen conditions. The estimated penalty functions $g(W_1, W_2)$ in Eq. (7) (fitted using Algorithm 4, with g a bivariate regression splines) are plotted in Fig. 5a: they increase with $W_1 = \text{DIST}$ and decrease with $W_2 = \text{TCC}$, which is consistent with previous analyses of V1 macaque data (Goris et al. 2014; Smith and Kohn 2008; Scott et al. 2015) where neurons dependences in macaque V1 were observed to decrease with inter-neuron distance and increase with tuning curve similarity. Fig. 5b shows that the corresponding edge posterior probabilities (Eq. (24)) decrease with DIST and increase with TCC, on average; the horizontal lines denote probability thresholds that lead to different graph FDR_Π controls (Eqs. (23), (25), and (27)). We obtained similar results for all grating orientations. Finally, Fig. 6a displays the estimated (p, δ)-graphs (Eq. (23)): the graph for the vertical grating contains 1160 edges (859 positive and 301 negative connections) with 10% FDR_Π, and 350 edges (307 positive, 43 negative) with 5% FDR_Π; the graph under blank screen contains 1246 edges (937 positive, 309 negative) with 10% FDR_Π, and 405 edges (362 positive, 43 negative) with 5% FDR_Π.

To investigate the extent to which covariation is tuned to orientation we computed the average absolute correlation

$$\text{AC}(\theta) = \frac{1}{d(d-1)} \sum_{i \neq j} |c_{ij}(\theta)|, \tag{33}$$

and the multivariate Gaussian Mutual Information (Shannon 1964; Guerrero 1994; Cover and Thomas 2006)

$$\text{MI}(\theta) = -\frac{1}{2} \log \det C(\theta), \tag{34}$$

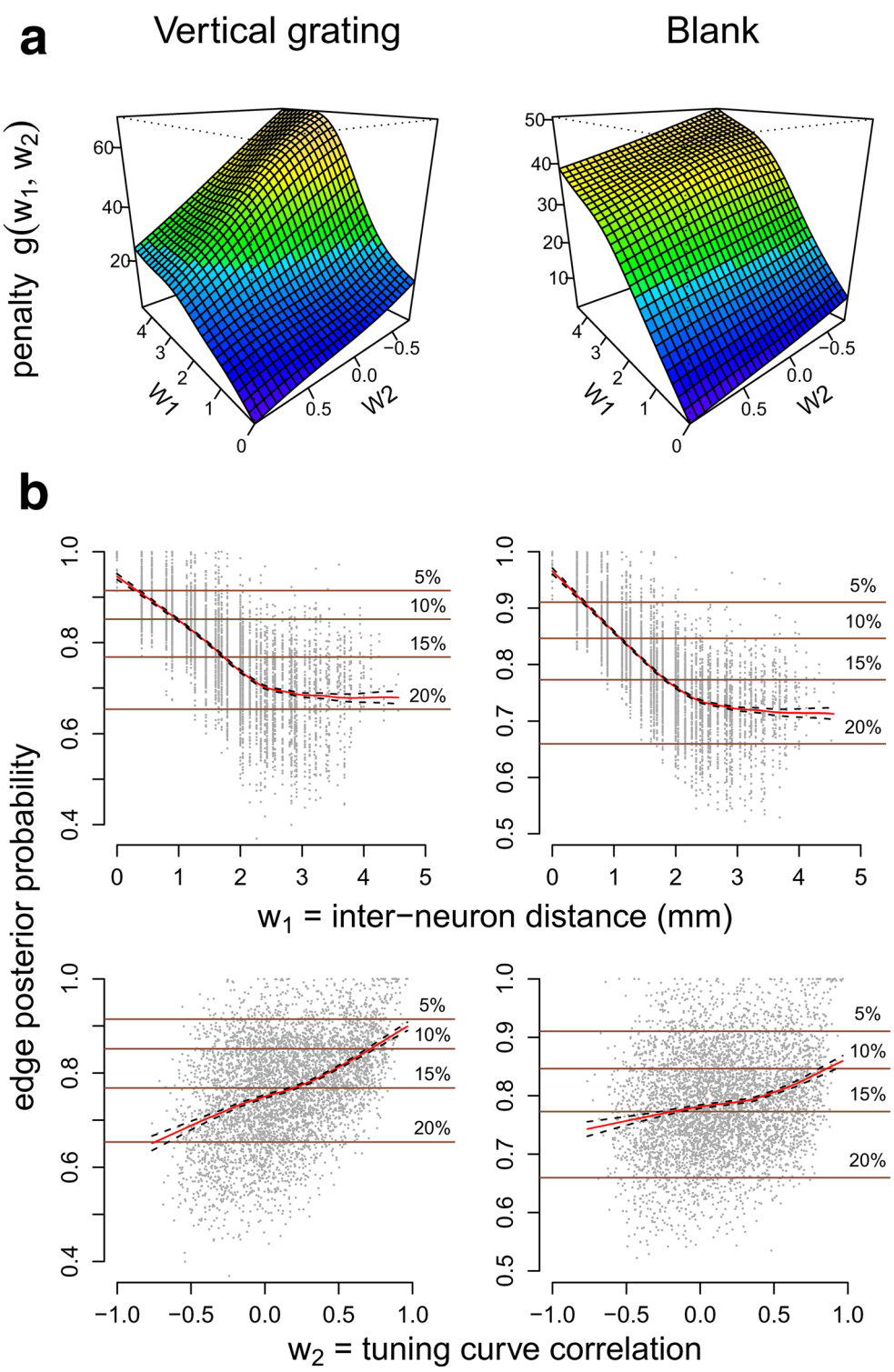

Fig. 5 **a** Estimated penalty functions (Eq. (7)) and **b** edge posterior probabilities (Eq. (24)) for all pairs of V1 neurons, with average plotted in red, under vertical grating and blank conditions. The penalty $g(W_1, W_2)$ increases with $W_1 =$ inter-neuron distance and decreases with $W_2 =$ tuning curve correlation. The average edge posterior probability decreases with W_1 and increases with W_2. The horizontal lines are the probability thresholds that lead to the corresponding 5, 10, 15, or 20% FDR_Π controls in (p, δ)-graphs (Eqs. (23), (25), and (27))

where $C(\theta) = \text{diag}(\Sigma(\theta))^{-1/2} \times \Sigma(\theta) \times \text{diag}(\Sigma(\theta))^{-1/2} = [c_{ij}(\theta)]$ is the correlation matrix of the square-rooted spike counts at orientation θ, and $\Sigma(\theta) = \Omega(\theta)^{-1}$. Larger $\text{AC}(\theta)$ and $\text{MI}(\theta)$ imply stronger connectivity. Figure 6b shows the posterior means of $\text{AC}(\theta)$ and $\text{MI}(\theta)$ as functions of θ, with 95% posterior probability bands; both measures display similar variations in connectivity and it appears that grating orientations θ and $\theta + 180$ yield graphs with similar connectivity. To confirm this, we considered the network connectivity at orientations $\Delta\theta$ apart: given some

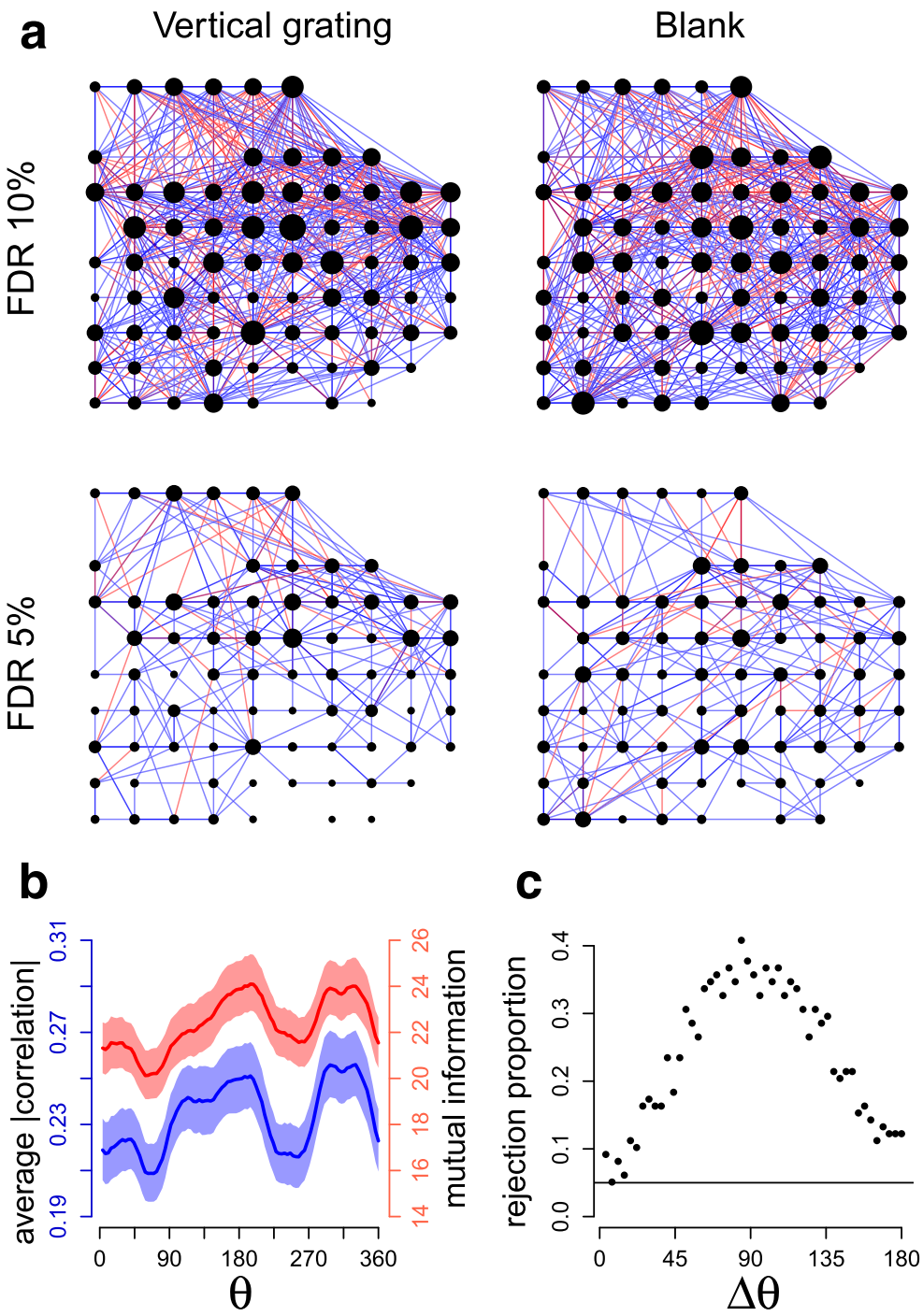

Fig. 6 Connectivity of V1 neurons. **a** Estimated connectivity $(p, \delta)-$graphs under vertical grating and blank conditions, with respective number of edges 1160 ± 87 and 1246 ± 87 (95% bootstrap intervals) at 10% FDR_{Π}, and 350 ± 29 and 405 ± 30 at 5% FDR_{Π}. The node positions represent the individual active channels on the 4×4 mm electrode array, blue and red edges denote positive and negative partial correlations, and a node size is proportional to the number of its connections. **b** Average across neuron pairs of absolute correlation (blue, Eq. (33)) and mutual information (red, Eq. (34)) with 95% posterior probability bands, as function of grating orientation θ. Larger values imply stronger connectivity. Both measures show maximal values about 20% above their minimum values. **c** Proportions of rejections across θ values of the 5%-level permutation tests that compare connectivities between orientations $\Delta\theta$ apart. Connectivity changes smoothly as a function of $\Delta\theta$, and the connectivity at an arbitrary orientation θ differs maximally from the connectivity at the orthogonal orientations $\theta \pm \Delta\theta$ with $\Delta\theta = 90$

orientation θ and some $\Delta\theta$, we (i) calculated

$$D(\Delta\theta) = \sum_{i<j} |c_{ij}(\theta) - c_{ij}(\theta + \Delta\theta)|, \tag{35}$$

(ii) obtained a permutation test p-value (Kass et al. 2014) of the null hypothesis that the connectivity is the same at θ and $\theta + \Delta\theta$, i.e. $D(\Delta\theta) = 0$, and repeated (i) and (ii) for all 98 values of $\theta \in [0, 360]$. Figure 6c shows how the proportion of 5%–level test rejections varies with $\Delta\theta$: the connectivity changes smoothly as a function of $\Delta\theta$, and the connectivity at an arbitrary orientation θ differs maximally from the connectivity at the orthogonal orientations $\theta \pm 90$, and minimally at orientation $\theta + 180$.

We applied GAR to estimate the V1 neuron network because our simulation study suggests it is the most accurate method. For comparison's sake, we also applied AGlasso (Eq. (3)) and drew qualitatively similar conclusions: 681 edges were identified for vertical grating and

more, specifically 945, for blank screen (although the resulting graphs look quite different since only about 50% of these edges were also identified by GAR at 10% FDR_Π within each condition) and the mutual information and average absolute correlation showed variations with orientation that were similar to, but more volatile than in Fig. 6b.

V4 data V4 data were recorded in response to a vertical grating ($\theta = 0$) and a blank screen (Fig. 4b). We selected 100 out of the 152 available candidate neuronal units, according to the same criteria used for the V1 data. The firing rates of these 100 neurons ranged from 0.04 to 38.54 spikes/s, with mean 6.64, and 2.5th and 97.5th percentiles 0.09 and 27.33. We applied GAR (Algorithm 4 with a univariate regression spline) to the 1000 ms square root transformed spike counts with inter-neuron distance (DIST) as an auxiliary quantity, and estimated the connectivity in the two conditions. The estimated penalty function $g(W)$ (Eq. (7), Fig. 7a) increases with $W = \text{DIST}$ and the edge posterior probability (Eq. (24), Fig. 7b) decreases with DIST, which is consistent with the previous analyses in Smith and Sommer (2013) and Vinci et al. (2016). Note that the posterior probability of non-zero partial correlations ($|\rho| > \delta = .005$) remains substantial at 70–75% between neurons that are over 2mm apart, but because these correlations have magnitude (absolute value of posterior mean) close to the threshold δ (not shown), they do not appear in the (p, δ)-graphs displayed in Fig. 7c, at 10% and 5% FDR_Π. The graph under vertical grating contains 333 edges (229 positive and 104 negative) with 10% FDR_Π, and 59 edges (53 positive and 6 negative) with 5% FDR_Π the graph under blank screen contains 573 edges (400 positive and 173 negative) with 10% FDR_Π, and 143 edges (115 positive and 28 negative) with 5% FDR_Π. These results suggest that neural connectivity is denser in the spontaneous activity induced by the blank screen. This was confirmed by a permutation test (Kass et al. 2014) based on the statistic $D(a, b) = \sum_{i<j} |c_{ij}(a) - c_{ij}(b)|$, analogous to Eq. (35), with null hypothesis $D(a, b) = 0$: the p-value was smaller than 10^{-8}.

For comparison sake, we also estimated the V4 neuron network using the AGlasso (Eq. (3)) and found 297 and 492 edges for vertical grating and blank screen, respectively; about 42% of AGlasso's edges were also discovered by GAR at 10% FDR_Π in the vertical grating condition, and about 44% in the blank screen condition.

Remarks In the two datasets we analysed here, the estimated function g (Eq. (7)) increased with inter-neuron distance and decreased with tuning curve correlation; conversely, the edge posterior probabilities (Eq. (24) decreased and increased with inter-neuron distance and tuning curve correlation, respectively (Figs. 5 and 7). Because g was fitted using splines, the dependence of the regularization on the auxiliary variables was estimated from the data rather than prespecified. Hence, GAR automatically extracted the neurons functional connectivity information carried by the auxiliary variables and incorporated it into the estimation of the partial correlations and dependence graphs. These are the main results of our data analyses. Note that the penalty function g is not constrained to be monotonic. While lateral connectivity in a region or within a single cortical area decreases with distance, which was encoded in our two data examples by g increasing with distance, there are many long neural pathways in the brain (see Van Den Heuvel and Sporns 2011). If these also induce functional connectivity, GAR will extract that information and fit a penalty function g that varies accordingly.

We further note that in both analyses reported here, the number of positive connections were approximately two to four times the number of negative ones. This result is consistent with previous analyses in macaque visual cortex where the majority of pairwise correlations were positive (Smith and Kohn 2008; Smith and Sommer 2013), which in turn suggests that finding a majority of positive partial correlations is reasonable (if a positive-definite covariance matrix has mostly positive entries, then its inverse has mostly negative entries, and consequently partial correlations are mostly positive according to Eq. (6)); a majority of positive partial correlations has also been observed in mouse visual cortex (Yatsenko et al. 2015). Moreover, it is also known that in macaque visual cortex the ratio of excitatory to inhibitory neurons is about 80/20 (Markram et al. 2004), which, depending on the relative proportion of inhibitory to excitatory neurons recorded and their connection strengths and probabilities, might favors positive functional connections. We also found that the partial correlations in areas V1 and V4 had similar magnitudes (Fig. 8a), but that the correlations were larger and the connectivity denser in V1 (Fig. 8b and c), which is consistent with previous findings in V1 (Smith and Kohn 2008) and V4 (Smith and Sommer 2013; Vinci et al. 2016). Denser connectivity and higher correlations in V1 may be due to differences in time-scales and correlation between cortical layers (Smith et al. 2013), since the V1 data were targeted at more superficial layers than V4, as well as other differences in connectivity structure between the two cortical regions. In addition, slow fluctuations in activity due to anesthesia may have played a role in the higher correlation values in V1 (Ecker et al. 2014), although the values present in the data analyzed here are similar to other V1 reports in awake animals (Gutnisky and Dragoi 2008; Poort and Roelfsema 2009; Samonds et al. 2009; Rasch et al. 2011).

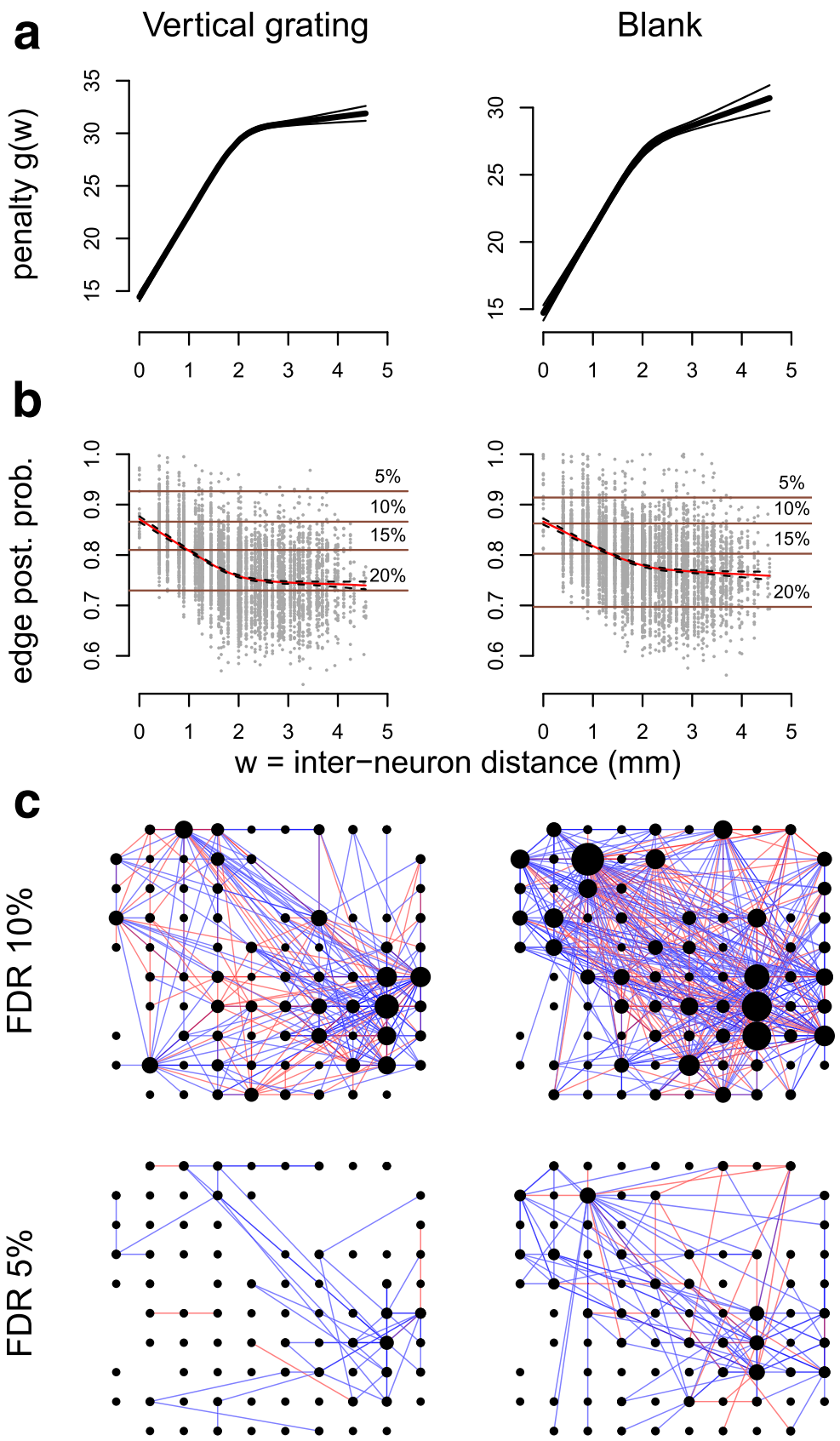

Fig. 7 **a** Estimated penalty functions (Eq. (7)) and **b** edge posterior probabilities (Eq. (24)) for all pairs of V4 neurons, with average plotted in red, under vertical grating and blank conditions. The penalty $g(W)$ increases with $W =$ inter-neuron distance. The average edge posterior probability decreases with W. The horizontal lines are the probability thresholds that lead to the corresponding 5, 10, 15, or 20% FDR_Π controls in (p, δ)-graphs (Eqs. (23), (25), and (27)). **c** Estimated connectivity $(p, \delta)-$graphs under the two conditions, with respective number of edges 333 ± 79 and 573 ± 89 (95% bootstrap intervals) at 10% FDR_Π, and 59 ± 19 and 143 ± 21 at 5% FDR_Π. The node positions represent the individual active channels on the 4×4 mm electrode array, blue and red edges denote positive and negative partial correlations, and a node size is proportional to the number of its connections

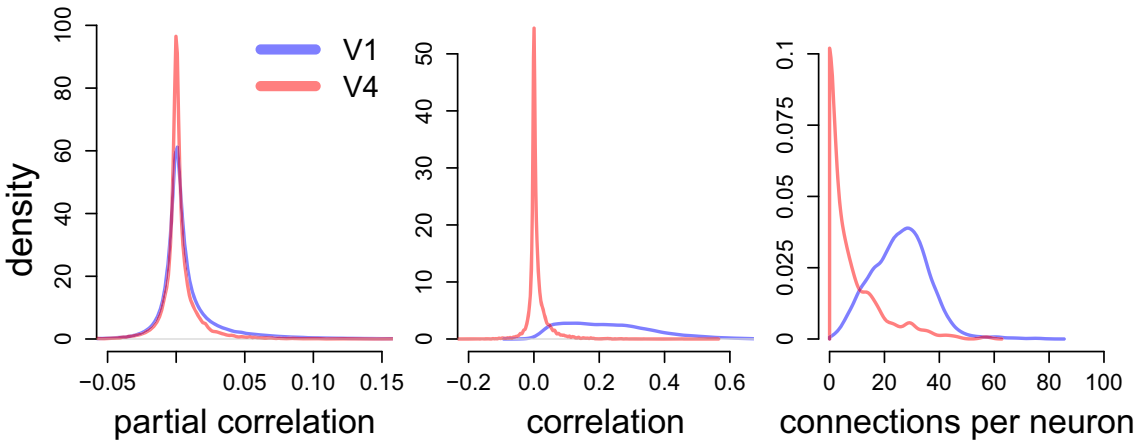

Fig. 8 Distributions of partial correlations, correlations, and number of connections per neuron in areas V1 and V4. Partial correlations in areas V1 and V4 have similar magnitudes. Correlations and number of connections per neuron are larger in area V1

3 Discussion

We have derived and implemented a method for pairwise covariate adjustment of the regularization penalty in the Graphical lasso, and have shown that it can greatly improve identification of the functional connectivity graph in high - dimensional neural spike count data. We have also illustrated the use of this technique in studying network activity by analyzing data from cortical areas V1 and V4, where the covariates were distance between neurons and tuning curve correlation. We expect this approach to be applicable to neural activity throughout cortex, and in subcortical areas as well. At the very least, partial correlation may be expected to change with inter-neuron distance, regardless of where these neurons are located, although cell-type shape specific information may also be useful to include, for example through a group-lasso type regularization (Yuan and Lin 2006), to adjust for anisotropies of neurons' axonal projections (Sincich and Blasdel 2001). In addition, even in the absence of a well-defined tuning curve, it is reasonable to expect partial correlation to depend on other characterizations of trial-averaged responses across experimental conditions (based, for example, on the PSTH), or anatomical connectivity and genetic information about neurons. Thus, we suggest the incorporation of this kind of covariate information is likely to be helpful in a wide range of problems involving neural functional connectivity.

While we are inclined, based on the research reported here, to think that the general idea of incorporating covariate information into regularization is a good one, there are many different ways to carry it out. These could involve alternative forms of regularization (e.g. an elastic net may be better suited than an L_1 penalty to regularize a large number of small effects), within both Bayesian and non-Bayesian frameworks, as well as regularization combined with dimensionality reduction (Chandrasekaran et al. 2012; Yuan 2012; Yatsenko et al. 2015). These are topics for future research. Also, in previous work we noted that when spike count correlation is viewed as resulting from underlying firing-rate correlation after corruption by Poisson-like noise (Vinci et al. 2016; Behseta et al. 2009), the firing-rate correlation can be much larger than spike count correlation, and may be expected to be more sensitive to experimental manipulation because it likely provides a better representation of input correlation (Vinci et al. 2016). A natural next step, therefore, is to nest GAR within the hierarchical model of Vinci et al. (2016). We are currently working on that extension to the method developed here. In addition, covariate-adapted regularization may be applied to high-dimensional point process models of neural spike trains (Kass et al. 2014) and time series models for local field potentials and other continuous-time neural signals. We hope to investigate the utility of the general idea in these diverse neural contexts.

Acknowledgments Data from visual area V1 were collected by Matthew A. Smith, Adam Kohn, and Ryan Kelly in the Kohn laboratory at Albert Einstein College of Medicine, and are available from CRCNS at http://crcns.org/data-sets/vc/pvc-11. We are grateful to Adam Kohn and Tai Sing Lee for research support. Data from visual area V4 were collected in the Smith laboratory at the University of Pittsburgh. We are grateful to Samantha Schmitt for assistance with data collection. Giuseppe Vinci was supported by the National Institute of Health (NIH R90DA023426) and by the Rice Academy Postdoctoral Fellowship. Robert E. Kass and Valérie Ventura were supported by the National Institute of Mental Health (NIH R01MH064537). Matthew A. Smith was supported by the National Institute of Health (NIH R01EY022928 and P30EY008098), Research to Prevent Blindness, and the Eye and Ear Foundation of Pittsburgh.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

Appendix

SAGlasso algorithm

There are several ways to build the weight matrix Q of SAGlasso. We used Gamma regression, as described in Algorithm 1, which can be implemented efficiently with standard statistical software, e.g. R packages glm (Dobson and Barnett 2008; Hastie and Pregibon 1992; McCullagh and Nelder 1989; Venables and Ripley 2002), mgcv (Wood 2011), and gam (Hastie and Tibshirani 1990). Note that in Eq. (3), Q is typically estimated by the square rooted absolute entries of the inverse sample covariance matrix. In SAGlasso, we observed a slightly better performance without applying any transformation.

Algorithm 1 *SAGlasso*

Input: $\mathbb{X}_n$, W, $\lambda > 0$, and $\hat{\Omega}$, preliminary estimate of Ω.
1. Obtain $\{Y_{ij}\}_{i<j}$, where $y_{ij} = |\hat{\omega}_{ij}|/(\hat{\omega}_{ii}\hat{\omega}_{jj})^{1/2}$.
2. Fit Gamma regression of Y_{ij} on W_{ij} with rate $g(W_{ij})$.
3. Obtain $\hat{Q} = [\hat{q}_{ij}]$ where $\hat{q}_{ij} = \hat{g}(W_{ij})/(\hat{\omega}_{ii}\hat{\omega}_{jj})^{1/2}$, for $i \neq j$, and $\hat{q}_{ii} = |\hat{\omega}_{ii}|^{-1}$, for $i = 1, ..., d$.
4. Solve Eq. (3) with $Q = \hat{Q}$ and λ.

Output: Estimate $\hat{\Omega}(\lambda)$.

GAR algorithms

GAR Algorithms 2–4 are derived in Section 2.2, and implemented in our R package "GARggm" available in ModelDB.

In Algorithm 2, $U \sim \text{InvGaussian}(a, b)$ has p.d.f. $p(u) = \left(\frac{b}{2\pi u^3}\right)^{1/2} \exp\left\{-b(u-a)^2/(2a^2u)\right\}$. Moreover, given a matrix M, M_{ij} is the i-th row and j-th column entry of M; M_{-ij} is the j-th column of M without the i-th entry; M_{i-j} is the i-th row of M without the j-th entry; and M_{-i-j} is the submatrix obtained by removing the i-th row and the j-th column from M. Algorithms 3 and 4 both produce posterior samples of Ω whose average approximates the posterior mean of Ω. The posterior mode of Ω can be obtained by solving Eq. (1) with $\lambda\|\Omega\|$ replaced by $\|\hat{\Lambda} \odot \Omega\|_1$, where $\hat{\Lambda}$ is the estimated penalty matrix from either Algorithm 3 or 4, and $\odot$ denotes the entry-wise matrix multiplication. This optimization can be performed using R functions such as `glasso` (package glasso, Friedman et al. (2008)) with argument `rho` set equal to $2\hat{\Lambda}/n$; see also the R package QUIC, Hsieh et al. (2011). We solve the SPL problem in Eq. (5) by the EM algorithm of Yuan (2012) involving Glasso, and we impose the GAR penalty matrix $\hat{\Lambda}$ on S in the Maximization step to obtain the GAR-SPL estimate. For $d \sim 100$, we suggest to run the Gibbs samplers for at least $B = 2000$ iterations, including a burn-in period of 300 iterations. The Gamma regression in step 2b of Algorithm 4 can be implemented either parametrically or nonparametrically by using standard statistical software e.g. R packages glm (Dobson and Barnett 2008; Hastie and Pregibon 1992; McCullagh and Nelder 1989; Venables and Ripley 2002) and mgcv (Wood 2011); in the data analyses we used splines (Kass et al. 2014).

Algorithm 2 *Block Gibbs sampler for* $\Omega \sim \pi(\Omega \mid \mathbb{X}_n, \Lambda)$.

Input: $S = \sum_{r=1}^{n}(X^{(r)} - \bar{X})'(X^{(r)} - \bar{X})$ and Λ; start value of Ω; number of iterations B.
For $b = 1, ..., B$:
1. For $i < j$: sample $\tau_{ij}^{-1} \sim \text{InvGaussian}((\lambda_{ij}|\omega_{ij}|)^{-1}, 2)$.
2. For $i = 1, ..., d$: compute $\Omega_{-ii} = \Omega'_{i-i} := \eta$ and $\omega_{ii} := \xi + \eta'\Omega_{-i-i}^{-1}\eta$, where $\xi \sim \Gamma(n/2+1, S_{ii}/2+\lambda_{ii})$, and $\eta \sim N(-AS_{-ii}, A)$, with $A = [(S_{ii} + 2\lambda_{ii})\Omega_{-i-i}^{-1} + D]^{-1}$ and $D = 2\text{diag}\left(\frac{\lambda_{1i}^2}{\tau_{1i}}, ..., \frac{\lambda_{(i-1)i}^2}{\tau_{(i-1)i}}, \frac{\lambda_{(i+1)i}^2}{\tau_{(i+1)i}}, ..., \frac{\lambda_{di}^2}{\tau_{di}}\right)$.
3. Set $\Omega^{(b)} = \Omega$.

Output: Sequence $\Omega^{(1)}, ..., \Omega^{(B)}$.

Algorithm 3 *GAR - Full Bayes*

Input: $\mathbb{X}_n$ and W; parameters r, r', s, s', K; sets $\{A_k\}_{k=1}^{K}$; start values of Ω, α, and β; number of iterations B.
For $b = 1, ..., B$:
1. For $i = 1, ..., d$: $\alpha_i^2 \mid \text{rest} \sim \Gamma\left((r+d+1)/2,\ s + C_i\right)$, where $C_i = \omega_{ii} + \sum_{j\neq i}\alpha_j^2 \sum_{k=1}^{K}\beta_k^2 I_{A_k}(W_{ij})\omega_{ij}^2\tau_{ij}^{-1}$.
2. For $k = 1, ..., K$: $\beta_k^2 \mid \text{rest} \sim \Gamma\left((r' + D_k)/2,\ s' + E_k\right)$, where $D_k = \sum_{i<j} I_{A_k}(W_{ij})$, and $E_k = \sum_{i<j} I_{A_k}(W_{ij})\omega_{ij}^2\alpha_i^2\alpha_j^2\tau_{ij}^{-1}$.
3. Do steps 1-2 of Algorithm 2 with current Λ.
4. Set $\Omega^{(b)} = \Omega$ and $\Theta^{(b)} = \{\alpha, \beta\}$.

Output: Sequences $\{\Omega^{(1)}, \Theta^{(1)}\}, ..., \{\Omega^{(B)}, \Theta^{(B)}\}$.

Algorithm 4 *GAR* - Empirical Bayes

Input: $\mathbb{X}_n$ and W; start values of Ω and Θ.
1. E-STEP: For $i \leq j$, approximate $\bar{\omega}_{ij} = \mathbb{E}[|\omega_{ij}| \mid \mathbb{X}_n, \Theta, W]$ by Algorithm 2.
2. M-STEP: Iterate a)-b) until convergence:
   a) For $i = 1, ..., d$, update α_i according to Eq. (21).
   b) Obtain g as the rate function of the Gamma regression of $y_{ij} = 2\alpha_i\alpha_j\bar{\omega}_{ii}$ on $W_{ij}, i < j$.
3. Iterate 1-2 until convergence.

Output: Estimate of Θ.

Computational efficiency of estimators

Table 1 contains the computation times of the graph estimators considered for $d = 50, 100$ and $n = 200, 500$, using the programming language R, CPU Quad-core 2.6 GHz Intel Core i7, and RAM 16 GB 2133 MHz DDR4. These times could be improved substantially by using a lower level language such as C++. Glasso, AGlasso, SPL, and SAGlasso are fitted with tuning parameter optimization based on ten-fold cross-validation involving 500 random splits over a fine grid of 20 values of the tuning parameter about its optimal value. GAR full Bayes (Algorithm 3; $K = \lfloor\sqrt{d}\rceil$) involved $B = 2000$ iterations, where the Gibbs sampler converged after about 300 iterations. GAR empirical Bayes (Algorithm 4; splines with 3 knots) involved 30 EM iterations, each including 500 iterations of Gibbs sampler for the E-step; the efficiency of this method may be improved by replacing the Gibbs sampler with some alternate faster approximation of Eq. (20).

Table 1 Computational time in seconds with 95% confidence intervals for $d = 50, 100$ and $n = 200, 500$

	$d = 50$		$d = 100$	
Method	$n = 200$	$n = 500$	$n = 200$	$n = 500$
Glasso	150 ± 1	160 ± 1	1198 ± 6	1057 ± 10
AGlasso	65 ± 1	87 ± 1	410 ± 3	501 ± 7
SPL	541 ± 19	641 ± 12	4626 ± 41	2709 ± 23
BAGlasso	96 ± 3	97 ± 2	825 ± 5	824 ± 5
SAGlasso	91 ± 1	116 ± 2	550 ± 18	626 ± 8
GAR-FB	92 ± 1	92 ± 1	827 ± 11	823 ± 3
GAR-EB	654 ± 22	724 ± 5	4163 ± 38	4337 ± 40

References

Ahrens, M.B., Orger, M.B., Robson, D.N., Li, J.M., Keller, P.J. (2013). Whole-brain functional imaging at cellular resolution using light-sheet microscopy. *Nature Methods*, *10*(5), 413–420.

Alivisatos, A.P., Andrews, A.M., Boyden, E.S., Chun, M., Church, G.M., Deisseroth, K., et al. (2013). Nanotools for neuroscience and brain activity mapping.

Andrews, D.F., & Mallows, C.L. (1974). Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B (Methodological)*, *1*, 99–102.

Banerjee, O., Ghaoui, L.E., d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine learning research*, *9*, 485–516.

Bassett, D.S., & Sporns, O. (2017). Network neuroscience. *Nature Neuroscience*, *20*(3), 353–364.

Behseta, S., Berdyyeva, T., Olson, C.R., Kass, R.E. (2009). Bayesian correction for attenuation of correlation in multi-trial spike count data. *Journal of Neurophysiology*, *101*(4), 2186–2193.

Brown, E.N., Kass, R.E., Mitra, P.P. (2004). Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience*, *7*(5), 456–461.

Butts, D.A., & Goldman, M.S. (2006). Tuning curves, neuronal variability, and sensory coding. *PLoS Biol*, *4*(4), e92.

Chandrasekaran, V., Parrilo, P.A., Willsky, A.S. (2012). Latent variable graphical model selection via convex optimization. *Annals of Statistics*, *40*(4), 1935–1967.

Cohen, M.R., & Kohn, A. (2011). Measuring and interpreting neuronal correlations. *Nature Neuroscience*, *14*(7), 811–819.

Cohen, M.R., & Maunsell, J.H. (2009). Attention improves performance primarily by reducing interneuronal correlations. *Nature Neuroscience*, *12*(12), 1594–1600.

Cover, T.M., & Thomas, J.A. (2006). *Elements of information theory*, 2nd edition. Wiley-Interscience: NJ.

Cowley, B.R., Smith, M.A., Kohn, A., Yu, B.M. (2016). Stimulus-driven population activity patterns in macaque primary visual cortex. *PLOS Computational Biology*, *12*(12), e1005185.

Cunningham, J.P., & Yu, B.M. (2014). Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, *17*(11), 1500–1509.

d'Aspremont, A., Banerjee, O., El Ghaoui, L. (2008). First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, *30*(1), 56–66.

Dempster, A.P., Laird, N.M., Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society. Series B (methodological), 1–38.

Dobson, A.J., & Barnett, A. (2008). *An introduction to generalized linear models*. Boca Raton: CRC Press, Chapman & Hall.

Ecker, A.S., Berens, P., Cotton, R.J., Subramaniyan, M., Denfield, G.H., Cadwell, C.R., et al. (2014). State dependence of noise correlations in macaque primary visual cortex. *Neuron*, *82*(1), 235–248.

Efron, B., Tibshirani, R., Storey, J.D., Tusher, V. (2001). Empirical Bayes analysis of a microarray experiment. *Journal of the American statistical association*, *96*(456), 1151–1160.

Efron, B., & Tibshirani, R. (2002). Empirical Bayes methods and false discovery rates for microarrays. *Genetic Epidemiology*, *23*(1), 70–86.

Efron, B. (2007). Size, power and false discovery rates. The Annals of Statistics 1351–1377.

Fan, J., Feng, Y., Wu, Y. (2009). Network exploration via the adaptive LASSO and SCAD penalties. *The Annals of Applied Statistics*, *3*(2), 521.

Foygel, R., & Drton, M. (2010). Extended Bayesian information criteria for Gaussian graphical models. In *Advances in neural information processing systems* (pp. 604–612).

Friedman, J., Hastie, T., Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*(3), 432–441.

Gelman, A., Carlin, J., Stern, H.S., Rubin, D.B. (2004). *Bayesian data analysis*. New York: CRC Press.

Georgopoulos, A.P., & Ashe, J. (2000). One motor cortex, two different views. *Nature Neuroscience*, *3*(10), 963.

Giraud, C., & Tsybakov, A. (2012). Discussion: Latent variable graphical model selection via convex optimization. *Annals of Statistics*, *40*(4), 1984–1988.

Goris, R.L., Movshon, J.A., Simoncelli, E.P. (2014). Partitioning neuronal variability. *Nature Neuroscience*, *17*(6), 858–865.

Guerrero, J.L. (1994). Multivariate mutual information: sampling distribution with applications. *Communications in Statistics-Theory and Methods*, *23*(5), 1319–1339.

Gutnisky, D.A., & Dragoi, V. (2008). Adaptive coding of visual information in neural populations. *Nature*, *452*(7184), 220–224.

Hastie, T.J., & Pregibon, D. (1992). Generalized linear models. In Chambers, J.M., & Hastie, T.J. (Eds.) *Wadsworth & Brooks/Cole*.

Hastie, T.J., & Tibshirani, R.J. (1990). *Generalized additive models* Vol. 43. Boca Raton: CRC press, Chapman & Hall.

Hsieh, C.J., Dhillon, I.S., Ravikumar, P.K., Sustik, M.A. (2011). Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in neural information processing systems* (pp. 2330–2338).

Kass, R.E., Eden, U.T., Brown, E.N. (2014). *Analysis of neural data*. New York: Springer.

Kelly, R.C., Smith, M.A., Samonds, J.M., Kohn, A., Bonds, A.B., Movshon, J.A., Lee, T.S. (2007). Comparison of recordings from microelectrode arrays and single electrodes in the visual cortex. *Journal of Neuroscience*, *27*(2), 261–264.

Kelly, R.C., Smith, M.A., Kass, R.E., Lee, T.S. (2010). Local field potentials indicate network state and account for neuronal response variability. *Journal of Computational Neuroscience*, *29*(3), 567–579.

Kelly, R.C., & Kass, R.E. (2012). A framework for evaluating pairwise and multiway synchrony among stimulus-driven neurons. *Neural Computation*, *24*(8), 2007–2032.

Kerr, J.N., & Denk, W. (2008). Imaging *in vivo*: watching the brain in action. *Nature Reviews Neuroscience*, *9*(3), 195–205.

Kipke, D.R., Shain, W., Buzsáki, G., Fetz, E., Henderson, J.M., Hetke, J.F., Schalk, G. (2008). Advanced neurotechnologies for chronic neural interfaces: new horizons and clinical opportunities. *Journal of Neuroscience*, *28*(46), 11830–11838.

Liu, H., Roeder, K., Wasserman, L. (2010). Stability approach to regularization selection (stars) for high dimensional graphical models. In *Advances in neural information processing systems* (pp. 1432–1440).

Markram, H., Toledo-Rodriguez, M., Wang, Y., Gupta, A., Silberberg, G., Wu, C. (2004). Interneurons of the neocortical inhibitory system. *Nature Reviews Neuroscience*, *5*(10), 793.

Mazumder, R., & Hastie, T. (2012). The graphical lasso: new insights and alternatives. Electronic journal of statistics 6.

McCullagh, P., & Nelder, J.A. (1989). Generalised linear models II.

Mitchell, J.F., Sundberg, K.A., Reynolds, J.H. (2009). Spatial attention decorrelates intrinsic activity fluctuations in macaque area V4. *Neuron*, *63*(6), 879–888.

Murphy, K.P. (2012). *Machine learning: a probabilistic perspective*. Cambridge: MIT press.

Poort, J., & Roelfsema, P.R. (2009). Noise correlations have little influence on the coding of selective attention in area V1. *Cerebral Cortex*, *19*(3), 543–553.

Rasch, M.J., Schuch, K., Logothetis, N.K., Maass, W. (2011). Statistical comparison of spike responses to natural stimuli in monkey area V1 with simulated responses of a detailed laminar network model for a patch of V1. *Journal of Neurophysiology*, *105*(2), 757–778.

Rothman, A.J., Bickel, P.J., Levina, E., Zhu, J. (2008). Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, *2*, 494–515.

Samonds, J.M., Potetz, B.R., Lee, T.S. (2009). Cooperative and competitive interactions facilitate stereo computations in macaque primary visual cortex. *Journal of Neuroscience*, *29*(50), 15780–15795.

Scott, J.G., Kelly, R.C., Smith, M.A., Zhou, P., Kass, R.E. (2015). False discovery rate regression: an application to neural synchrony detection in primary visual cortex. *Journal of the American Statistical Association*, *110*(510), 459–471.

Shadlen, M.N., & Newsome, W.T. (1998). The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *Journal of Neuroscience*, *18*(10), 3870–3896.

Shannon, C.E. (1964). *Mathematical theory of communications*. Urbana: University of Illinois Press.

Sincich, L.C., & Blasdel, G.G. (2001). Oriented axon projections in primary visual cortex of the monkey. *Journal of Neuroscience*, *21*(12), 4416–4426.

Smith, M.A., & Kohn, A. (2008). Spatial and temporal scales of neuronal correlation in primary visual cortex. *Journal of Neuroscience*, *28*(48), 12591–12603.

Smith, M.A., & Sommer, M.A. (2013). Spatial and temporal scales of neuronal correlation in visual area V4. *Journal of Neuroscience*, *33*(12), 5422–5432.

Smith, M.A., Jia, X., Zandvakili, A., Kohn, A. (2013). Laminar dependence of neuronal correlations in visual cortex. *Journal of neurophysiology*, *109*(4), 940–947.

Song, D., Wang, H., Tu, C.Y., Marmarelis, V.Z., Hampson, R.E., Deadwyler, S.A., Berger, T.W. (2013). Identification of sparse neural functional connectivity using penalized likelihood estimation and basis functions. *Journal of computational neuroscience*, *35*(3), 335–357.

Stevenson, I.H., & Kording, K.P. (2011). How advances in neural recording affect data analysis. *Nature Neuroscience*, *14*(2), 139–142.

Van Den Heuvel, M.P., & Sporns, O. (2011). Rich-club organization of the human connectome. *Journal of Neuroscience*, *31*(44), 15775–15786.

Venables, W.N., & Ripley, B.D. (2002). *Modern applied statistics with S*. New York: Springer.

Vinci, G., Ventura, V., Smith, M.A., Kass, R.E. (2016). Separating spike count correlation from firing rate correlation. *Neural Computation*, *28*(5), 849–881.

Wang, H. (2012). Bayesian graphical lasso models and efficient posterior computation. *Bayesian Analysis*, *7*(4), 867–886.

West, M. (1987). On scale mixtures of normal distributions. *Biometrika*, *1*, 646–8.

Wood, S.N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(1), 3–36.

Yatsenko, D., Josić, K., Ecker, A.S., Froudarakis, E., Cotton, R.J., Tolias, A.S. (2015). Improved estimation and interpretation of correlations in neural circuits. *PLoS Computational Biology*, *11*(3), e1004083.

Yu, B.M., Cunningham, J., Santhanam, G., Ryu, S.I., Shenoy, K.V., Sahani, M. (2009). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In *Advances in neural information processing systems* (pp. 1881–1888).

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(1), 49–67.

Yuan, M., & Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, *94*(1), 19–35.

Yuan, M. (2012). Discussion: latent variable graphical model selection via convex optimization. *Annals of Statistics*, *40*(4), 1968–1972.