# Fast and Scalable Learning of Sparse Changes in High-Dimensional Gaussian Graphical Model Structure

**Beilun Wang** **Arshdeep Sekhon** **Yanjun Qi**
Department of Computer Science, University of Virginia
www.jointggm.org

## Abstract

We focus on the problem of estimating the change in the dependency structures of two $p$-dimensional Gaussian Graphical models (GGMs). Previous studies for sparse change estimation in GGMs involve expensive and difficult non-smooth optimization. We propose a novel method, DIFFEE for estimating DIFFerential networks via an Elementary Estimator under a high-dimensional situation. DIFFEE is solved through a faster and closed form solution that enables it to work in large-scale settings. We conduct a rigorous statistical analysis showing that surprisingly DIFFEE achieves the same asymptotic convergence rates as the state-of-the-art estimators that are much more difficult to compute. Our experimental results on multiple synthetic datasets and one real-world data about brain connectivity show strong performance improvements over baselines, as well as significant computational benefits.

## 1 Introduction

Learning the change of interactions between random variables is an essential task in many real-world applications. For instance, identifying the difference in brain connectivity networks of subjects from different groups can shed light on understanding psychiatric diseases [10]. As another example in gene expression analysis, interests may not center on a particular graph representing interactions among genes, but instead on how gene interactions change when external stimuli change [14]. Such change detection can significantly simplify network-driven studies about diseases, drugs or system understanding.

In this paper we consider Gaussian graphical models (GGMs) and focus on estimating changes in the dependency structure of two $p$-dimensional GGMs, based on $n_c$ and $n_d$ samples drawn from the models, respectively. Recent literature has made significant advances on estimating the statistical dependency structure of GGMs based on samples drawn from the model [1][13] (reviewed in 2.1). Detecting structural changes naturally involves two sets of data samples. Given two sets of data (in the form of two matrices) $\mathbf{X}_c \in \mathbb{R}^{n_c \times p}$ and $\mathbf{X}_d \in \mathbb{R}^{n_d \times p}$ identically and independently drawn from normal distributions $N_p(\mu_c, \Sigma_c)$ and $N_p(\mu_d, \Sigma_d)$ respectively, our goal is to estimate the structural change $\Delta$ (defined by [35]) [1]:

$$\Delta = \Omega_d - \Omega_c \tag{1.1}$$

Here $\mu_c, \mu_d \in \mathbb{R}^p$ describes the mean and $\Sigma_c, \Sigma_d \in \mathbb{R}^{p \times p}$ represents covariance matrices. In Eq. (1.1), the precision matrix $\Omega_c := (\Sigma_c)^{-1}$ and $\Omega_d := (\Sigma_d)^{-1}$. The conditional dependency graph structure of a GGM is encoded by the sparsity pattern of its precision matrix. The entries of $\Delta$ describe if the magnitude of conditional dependency of a pair of random variables changes between two conditions. They can also be interpreted as the differences in the partial covariance of each pair of random variables between the two conditions.

In particular, we focus on estimating the change $\Delta$ under a high-dimensional situation, where the number of variables $p$ may exceed the number of observations: $p > \max(n_c, n_d)$. In such high-dimensional settings, it is still possible to conduct consistent estimation by leveraging low-dimensional structure such as sparsity constraints. A sparse $\Delta$ indicates few of its entries are non-zero. In the context of estimating structural changes of two GGMs, this translates into a differential network with few edges. However, we do not assume the individual structures $\Omega_c$ and $\Omega_d$ to be sparse, and they may both correspond to dense matrices. Our main

Proceedings of the $21^{st}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84. 

[1] Using which of the two sample sets as 'c' set (or 'd' set) does not affect the computational cost and the statistical convergence rates of our model. For instance, on samples from a controlled disease study 'c' may represent the 'control' group and 'd' may represent the 'disease' group.

objective is to get an estimated $\widehat{\Delta}$ of the true change $\Delta^*$ such that the estimation error $(\widehat{\Delta} - \Delta^*)$ is bounded.

A naive approach to detecting structural changes in GGMs is a two-step procedure in which we estimate $\widehat{\Omega}_d$ and $\widehat{\Omega}_c$ from two sets of samples separately and obtain $\widehat{\Delta} = \widehat{\Omega}_d - \widehat{\Omega}_c$. However, in a high-dimensional setting, this strategy needs to assume that both $\Omega_d$ and $\Omega_c$ are sparse (in order to achieve consistent estimation). This is not necessarily true even if the change $\Delta$ is sparse. A motivating example s from identifying the difference in connectivity networks among brain regions (functional networks) of subjects from different groups. Recent literature in neuroscience has suggested functional networks are not sparse. On the other hand, differences in functional connections across subjects should be sparse [2]. In the application of estimating genetic networks of two conditions, each individual network might contain hub nodes and therefore not entirely sparse.

This has motivated a few recent studies to directly estimate the changes of structures from two sets of samples. Zhang et al. used the fused norm for regularizing maximum likelihood estimation (MLE) to simultaneously learn two GGMs with a sparsity-inducing penalty on the difference [33]. The resulting penalized MLE framework is a log-determinant program, which can be solved by block coordinate descent algorithms [33] or the alternating direction method of multipliers (ADMM) by the JGL-fused package [9]. Later Liu et al. proposed to use density ratio estimation (DRE) to directly learn structural changes without having to identify the structures of each individual graphical model. The authors focused on exponential family-based pairwise Markov networks [17] and solve the resulting optimization using proximal gradient descent [16]. A follow-up study showed that under certain conditions the DRE method recovers the correct parameter sparsity with high probability [16]. More recently, Fazayeli et al. introduced a regularized density ratio estimator for direct structured change estimation in Ising model structure. Theoretically, the authors showed that the estimation error converges to zero under milder conditions than DRE [12]. Another related regularized convex program to directly learn structural changes without going through the learning of the individual GGMs is the Diff-CLIME method [34]. Diff-CLIME uses an $\ell_1$ minimization formulation constrained by the covariance-precision matching. Diff-CLIME reduces the estimation problem to solving linear programs (LP) and can be solved by using any standard LP solvers. Another recent work relaxes the Gaussian assumption in Diff-CLIME model to a semiparametric distribution [30]. All previous studies have used $\ell_1$ regularized convex formulation for estimating structural changes. While state-of-the-art optimization methods have been developed to solve the resulting non-smooth programs, their iterative algorithms are very expensive for large-scale problems.

In this paper, we propose a simple estimator, namely DIFFerential networks via an Elementary Estimator (DIFFEE) for fast and scalable learning of sparse structural change in high-dimensional GGMs. Briefly speaking, DIFFEE provides the following benefits:

- **Novel approach:** DIFFEE presents a novel way of structural change estimation by extending the elementary estimator for sparse GGM[31]. (Section 2.3)
- **Closed-Form optimization:** We optimize DIFFEE through a closed-form manner that can dramatically improve its entire time complexity to $O(p^3)$. The closed-form solution makes DIFFEE scalable to much larger values of $p$, compared to the aforementioned state-of-the-art. (Section 2.4)
- **Convergence rate:** We theoretically prove that DIFFEE achieves the same sharp convergence rate as the aforementioned regularized convex programs. (Section 2.5)
- **Evaluation:** DIFFEE is evaluated using several simulated datasets and one real-world neuroscience dataset. It improves the state-of-the-art baselines with better estimation F-1 scores as well as significant computational advantages. (Section 3)

## 2 Method

Our main goal is to design a simple estimator with closed-form solutions that can achieve the same sharp convergence rates as the state-of-the-art regularized convex formulations under high-dimensional settings. This has been achieved by the so-called elementary estimators [31] in the context of learning sparse GGM from one set of samples. Inspired by [31], naturally, we ask the following question: *is there an elementary estimator that can estimate structural changes in GGMs with a closed-form solution and achieves the near-optimal convergence rate?* This section provides an affirmative response of "YES" by proposing the DIFFEE algorithm. In the rest of this section, we first review the background of elementary estimator and then propose to estimate the differential network through an elementary estimator. Finally, we provide a rigorous theoretical analysis of DIFFEE's convergence rates.

*Notations:* Given a $p$-dimensional vector $x = (x_1, x_2, \ldots, x_p)^T \in \mathbb{R}^p$, $\| x \|_1 = \sum_i |x_i|$ represents the $\ell_1$-norm of $x$. $\| x \|_\infty = \max_i |x_i|$ is the $\ell_\infty$-norm of $x$. $\| x \|_2 = \sqrt{\sum_i x_i^2}$ describes the $\ell_2$-norm of $x$.

### 2.1 Background: Elementary Estimator for Estimating Sparse GGM in Closed Form

Sparse Gaussian Graphical Model(sGGM)[15, 18, 32] assumes data samples are independently and identically drawn from $N_p(\mu, \Sigma)$, a multivariate normal distribution with mean $\mu$ and covariance matrix $\Sigma$. The conditional dependency graph structure among its $p$ random variables is encoded by the sparsity pattern of the inverse covariance matrix (precision matrix) $\Omega$. $\Omega := (\Sigma)^{-1}$. An edge does not connect $j$-th node (variable) and $k$-th node (variable) if and only if $\Omega_{jk} = 0$ (i.e., conditionally independent). sGGM imposes an $\ell_1$ penalty on the parameter $\Omega$.

**Regularized MLE:** Over the past decade, significant progress has been made on estimating sGGMs based on samples drawn from the model. Most sGGM estimation [1, 32] are based on minimizing the $\ell_1$-regularized Gaussian negative log likelihood:

$$\underset{\Omega}{\text{argmin}} - \log\det(\Omega) + < \Omega, \Sigma > + \lambda_n ||\Omega||_1 \qquad (2.1)$$

Friedman et al. [13] used a blockwise coordinate descent algorithm called the graphical lasso to efficiently solve the regularized MLE formulation. Alternatively, Meinshausen et al. [19] introduced a neighborhood selection approach that applies a lasso linear regression on each variable separately and combines the result to learn the conditional dependency structure.

**CLIME:** Later Cai et al. [5] proposed a constrained $\ell_1$ minimization method for inverse matrix estimation (abbreviated as CLIME) formulated as follows:

$$\underset{\Omega}{\text{argmin}} ||\Omega||_1 \quad \text{subject to: } ||\Sigma\Omega - I||_\infty \leq \lambda_n \qquad (2.2)$$

The above formulation can be decomposed into column-wise linear programming. However the computational cost of this LP formulation gets significantly demanding as $p$ increases.

**EE-sGGM:** Recently, Yang et al. [31] proposed a closed-form estimator for learning Gaussian graphical models through the following form:

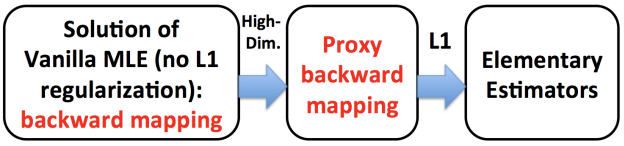
$$\underset{\Omega}{\text{argmin}} ||\Omega||_{1,,\text{off}} \quad \text{subject to:} ||\Omega - [T_v(\widehat{\Sigma})]^{-1}||_{\infty,\text{off}} \leq \lambda_n \qquad (2.3)$$

Eq. (2.3) is a special case of the elementary estimator for graphical models (GM) of exponential families proposed in [31], namely Elementary Estimators-GM. It has the following generic formulation:

$$\underset{\theta}{\text{argmin}} ||\theta||_1 \quad \text{Subject to: } ||\theta - \mathcal{B}^*(\widehat{\phi})||_\infty \leq \lambda_n \qquad (2.4)$$

Here $\mathcal{B}^*(\widehat{\phi})$ is the so-called proxy of backward mapping for the target GM (more details in Section A.1). $\lambda_n$ is a regularization parameter. $\widehat{\phi}$ is the empirical mean of the sufficient statistics. For example, in the case of Gaussian GM, $\widehat{\phi}$ is the sample covariance matrix.

Figure 1: Basic idea of elementary estimators for graphical model.




The key idea in Eq. (2.4) (summarized in Figure 1) is to investigate the vanilla MLE and where it "breaks down" for estimating a graphical model of exponential families in the case of high-dimensions [31]. Essentially the vanilla graphical model MLE can be expressed as a backward mapping that computes the model parameters from some given moments in an exponential family distribution. For instance, in the case of learning GGM with vanilla MLE, the backward mapping is $\widehat{\Sigma}^{-1}$ that estimates $\Omega$ from the sample covariance matrix (moment) $\widehat{\Sigma}$.

However, this backward mapping is not available in a closed form for many classes of graphical models, such as Ising model. Even if it has a simple closed form, the backward mapping is normally not well-defined in high-dimensional settings. In the case of GGM, when given the sample covariance $\widehat{\Sigma}$, we cannot just compute the vanilla MLE solution as $[\widehat{\Sigma}]^{-1}$ since $\widehat{\Sigma}$ is rank-deficient when $p > n$. Therefore Yang et al. [31] used carefully constructed proxy backward maps for Eq. (2.4) that are both available in closed-form, and well-defined in high-dimensional settings for GGM and Ising models. $[T_v(\widehat{\Sigma})]^{-1}$ in Eq. (2.3) is the proxy backward mapping Yang et al. used for GGM (more details in Section 2.3 and in Appendix Section A.1).

When given the term $\mathcal{B}^*(\widehat{\phi})$ in Eq. (2.4), the solution of Eq. (2.4) is closed-form and involves only simple thresholding operations. This solution is

$$\widehat{\theta} = S_{\lambda_n}(\mathcal{B}^*(\widehat{\phi}))$$

where the function $S(\cdot)$ is an element-wise soft-thresholding with parameter $\lambda$:

$$[S_\lambda(A)]_{ij} = \text{sign}(A_{ij}) \max(|A_{ij}| - \lambda, 0) \qquad (2.5)$$

The optimization in Eq. (2.4) is decomposable into independent element-wise subproblems. Each subproblem corresponds to soft-thresholding. Essentially the final estimators are obtained by performing simple thresholding operations on the proxy backward maps. This class of estimators is thus both computationally practical and highly scalable. Using the theoretical framework proposed by [20] for regularized M-estimators, Yang et al. further proved that the resulting algorithms achieve strong statistical guarantees with sharp convergence rates.

### 2.2 Previous Estimators for Change Estimation in GGM Structure

Multiple estimators have been proposed to estimate sparse differential network from two sets of samples.

**FusedGLasso (Regularized MLE):** The most straightforward estimator for differential network was to extend the classic Graphical lasso estimator [32] for sparse GGM with an added sparsity penalty on the differential network (i.e., fused norm).

$$\begin{aligned}\underset{\Omega_c,\Omega_d\succ 0,\Delta}{\operatorname{argmin}}\ & n_c(-\log\det(\Omega_c)+<\Omega_c,\widehat{\Sigma}_c>)\\ &+n_d(-\log\det(\Omega_d)+<\Omega_d,\widehat{\Sigma}_d>)\\ &+\lambda_2(||\Omega_c||_1+||\Omega_d||_1)+\lambda_n||\Delta||_1\end{aligned} \tag{2.6}$$

This was solved by block coordinate descent algorithms in [33]. Later the alternating direction method of multipliers (ADMM) was used to solve Eq. (2.6) that needs to run SVD in one sub-procedure [9].

**Diff-CLIME:** Another recent study [35] extended the CLIME estimator to directly learn the $\Delta$ through a constrained optimization formulation.

$$\begin{aligned}&\underset{\Delta}{\operatorname{argmin}}\ ||\Delta||_1\\ &\text{Subject to: } ||\widehat{\Sigma}_c\Delta\widehat{\Sigma}_d-(\widehat{\Sigma}_c-\widehat{\Sigma}_d)||_\infty\le\lambda_n\end{aligned} \tag{2.7}$$

This reduces the estimation to solving multiple linear programming problems.

**DensityRatio:** The third category of estimators optimizes the following loss:

$$\underset{\Delta}{\operatorname{argmax}}\,\mathcal{L}_{\text{KLIEP}}(\Delta)-\lambda_n\parallel\Delta\parallel_1-\lambda_2\parallel\Delta\parallel_2 \tag{2.8}$$

Here KLIEP minimizes the KL divergence between the true probability density $p_d(x)$ and the estimated $\widehat{p}_d(x)=r(x;\Delta)p_c(x)$ without explicitly modeling the true $p_c(x)$ and $p_d(x)$. Its key idea is the formulation of density ratio term $r(x;\Delta)$ for directly estimating sparse differential network of graphical models in exponential families. This DensityRatio estimator uses the elastic-net penalty for enforcing $\Delta$ to be sparse. The resulting optimization was solved using proximal gradient descent methods in [16].

### 2.3 Proposed Method: DIFFEE

The aforementioned studies cannot avoid certain steps involving expensive computation in their iterative optimization, such as SVD operations in the FusedGLasso, linear programming in the Diff-CLIME, and calculating the normalization term in the Density-Ratio estimator. We aim to propose a scalable and theoretically-guaranteed estimator for estimating sparse differential network under large-scale settings.

**Differential Network by Elementary Estimators (DIFFEE):** Computationally elementary estimators are much faster than their regularized convex program peers for graphical model estimation. Therefore we extend it to the following general estimator for estimating sparse change in GGM structure:

$$\begin{aligned}&\underset{\Delta}{\operatorname{argmin}}\ ||\Delta||_1\\ &\text{Subject to: } ||\Delta-\mathcal{B}^*(\widehat{\Sigma}_d,\widehat{\Sigma}_c)||_\infty\le\lambda_n\end{aligned} \tag{2.9}$$

The basic idea in Eq. (2.9) is to use a well-defined proxy function $\mathcal{B}^*(\widehat{\Sigma}_d,\widehat{\Sigma}_c)$ to approximate the backward mapping (the vanilla graphical model MLE solution), so that $\mathcal{B}^*(\widehat{\Sigma}_d,\widehat{\Sigma}_c)$ is both well-defined under high-dimensional situations and also has a simple closed-form.

As shown by Figure 1, there are three components in the estimation pipeline of elementary estimator for GM: (1) Backward mapping that is the vanilla MLE solution for estimating an exponential graphical model; (2) Proxy backward mapping $\mathcal{B}^*(\widehat{\Sigma}_d,\widehat{\Sigma}_c$ for dimensional settings; and (3) The closed-form solution of Eq. (2.9) as the final estimator.

**(1) Backward Mapping:** The density ratio of two Gaussian distributions is naturally an exponential-family distribution (see Section A.1.1). Based on [29], learning an exponential family distribution from data means to estimate its canonical parameter. For an exponential family distribution, computing the canonical parameter through vanilla graphical model MLE can be expressed as a backward mapping (the first step in Figure 1). Through simple derivations in Eq. (A.8), we can easily conclude that the differential network $\Delta$ is one entry of the canonical parameter for this distribution. When using vanilla MLE to learn this exponential distribution (i.e., estimating canonical parameter), the backward mapping of $\Delta$ can be easily inferred from the two sample covariance matrices using $\left(\widehat{\Sigma}_d^{-1}-\widehat{\Sigma}_c^{-1}\right)(\textit{Section A.1})$.

**(2) Proxy Backward Mapping:** Now the key is to find a closed-form and statistical guaranteed estimator as proxy backward mapping of $\Delta$ under high-dimensional cases. Inspired by the elementary estimator for sGGM, we choose $[T_v(\widehat{\Sigma}_d)]^{-1}-[T_v(\widehat{\Sigma}_c)]^{-1})$ as the proxy backward mapping for $\Delta$. Here

$$[T_v(A)]_{ij}:=\rho_v(A_{ij}) \tag{2.10}$$

where $\rho_v(\cdot)$ is chosen to be a soft-thresholding function. We therefore obtain the following DIFFEE objective function for estimating sparse changes in GGM structure:

$$\begin{aligned}&\underset{\Delta}{\operatorname{argmin}}\ ||\Delta||_1\\ &\text{Subject to: } ||\Delta-\left([T_v(\widehat{\Sigma}_d)]^{-1}-[T_v(\widehat{\Sigma}_c)]^{-1}\right)||_\infty\le\lambda_n\end{aligned} \tag{2.11}$$

Here $\lambda_n>0$ is the tuning parameter.

The optimization in Eq. (2.11) seeks an estimator with minimum complexity with regard to the $\ell_1$ regularization, at the same time being close enough to the 'initial estimator' $[T_v(\widehat{\Sigma}_d)]^{-1}-[T_v(\widehat{\Sigma}_c)]^{-1}$ according to the element-wise $\ell_\infty$ norm. This formulation ensures

that the final estimator (solution of Eq. (2.11)) has the desired sparse structure.

Theoretically, the choice of $\ell_1$ and $\ell_\infty$ in Eq. (2.9) connects to the asymptotic error bounds of the final estimators. In Section 2.5, we theoretically prove that the statistical convergence rate of DIFFEE achieves the same sharp convergence rate as the state-of-the-art estimators for differential network. Our proofs are inspired by the unified framework of the high-dimensional statistics[20] and EE for sGGM[31].

[31] proved that when (p>n), the proxy backward mapping $[T_v(\widehat{\Sigma})]^{-1}$ in their EE-sGGM achieves the sharp convergence rate to its truth (i.e., by proving $||T_v(\widehat{\Sigma}))^{-1} - \Sigma^{*-1}||_\infty = O(\sqrt{\frac{\log p}{n}})$). The proof was extended from the previous study [26] who devised $T_v(\widehat{\Sigma})$ for estimating covariance matrix consistently under high-dimensional cases. We use the convergence results from [26] and [31] in Section 2.5 for deriving the statistical convergence rates of DIFFEE (details in Section A.2).

**(3) Closed Form Solution:** To solve Eq. (2.11), we get the following closed form solution:

$$\widehat{\Delta} = S_{\lambda_n}([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}) \tag{2.12}$$

Where $[T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}$ is the pre-computed proxy backward mapping. Here $[S_\lambda(A)]_{ij} = \text{sign}(A_{ij})\max(|A_{ij}|-\lambda, 0)$ is the same soft-thresholding function in Eq. (2.5). Algorithm 1 shows the detailed steps of the DIFFEE estimator. Being non-iterative, the closed form solution helps DIFFEE achieve significant computational advantages over other estimators.

**Algorithm 1** DIFFEE

**input** Two data matrices $\mathbf{X}_c$ and $\mathbf{X}_d$.
**input** Hyper-parameter: $\lambda_n$ and $v$
**output** $\Delta$
1: Compute $[T_v(\widehat{\Sigma}_c)]^{-1}$ and $[T_v(\widehat{\Sigma}_d)]^{-1}$ from $\widehat{\Sigma}_c$ and $\widehat{\Sigma}_d$.
2: Compute $\Delta = S_{\lambda_n}([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1})$
**output** $\Delta$

### 2.4 Analysis of Computational Complexity

The closed form solution (Eq. (2.12)) brings significant advantages in hyper-parameter tuning. This is because we only need to compute the proxy backward mapping $[T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}$ once. Then the model selection just executes a fast and simple element-wise soft-thresholding operator using different values of hyper-parameter $\lambda_n$ ( Eq. (2.12)).

In details, DIFFEE includes four non-iterative operations in its computation:

1. Estimating two covariance matrices. The computational complexity is $O(\max(n_c, n_d)p^2)$.
2. The element-wise soft-thresholding operations $[T_v(\cdot)]$, that cost $O(p^2)$.
3. The matrix inversions [2] $[T_v(\cdot)]^{-1}$ to get the proxy backward mapping, that cost $O(p^3)$.
4. The element-wise soft-thresholding operation $S_{\lambda_n}$ that costs $O(p^2)$.

Table 1: Compare the asymptotic time complexity. DIFFEE is the best among all the estimators. Here $T$ is the number of iterations.

| DIFFEE | FusedGLasso | Density Ratio | Diff-CLIME |
|---|---|---|---|
| $O(p^3)$ | $O(T*p^3)$ | $O((n_c+p^2)^3)$ | $O(p^8)$ |

Therefore, the total asymptotic computational complexity of DIFFEE estimator is $O(p^3)$ .

In Table 1, we compare the asymptotic computational complexity of our method to the baselines. DIFFEE achieves the best computational complexity compared to the state-of-the-art baselines. This is because:

- All existing estimators for differential network estimation have used an iterative optimization procedure to find the solution. In each iteration, their estimations require at least $O(p^3)$ computational cost.
- For tuning the sparsity hyperparameter $\lambda_n$, DIFFEE only needs to re-run its element-wise soft-thresholding operation $S_{\lambda_n}$ that cost $O(p^2)$. In contrast, all the baselines have to re-run the whole algorithm for each value of the hyper-parameter $\lambda_n$.
- Most estimators have two hyperparameters for tuning. FusedGlasso (Eq. (2.6)) and DensityRatio (Eq. (2.8)) both need to tune the hyperparameter $\lambda_2$ [3]. Both tuning are much more expensive than DIFFEE in computation. DIFFEE needs to tune the hyperparamter $v$, but it costs only $O(p^2)$.
- Diff-CLIME has one hyperparameter $\lambda_n$ for tuning, however, its asymptotic time cost ($O(p^8)$) is significantly more demanding than DIFFEE [4]. In summary, Diff-CLIME can not handle large-scale cases, like $p > 100$. For example, in our experiments Diff-CLIME can not even finish on a case of $p = 200$ after two days of running.

### 2.5 Strong Statistical Guarantees of DIFFEE

In this section, we provide a statistical convergence analysis of DIFFEE Eq. (2.9) under the following struc-

[2] Many faster algorithms exist for speeding up matrix inversion and matrix multiplication. The best known asymptotic cost of matrix inversion is $O(p^{2.373})$ (Wikipedia). Besides both operations can be further improved up by paralelization

[3] The optimization problem of DensityRatio is a quadratic programming problem with $n_c + p^2$ variables. Based on the result from [4], the computational complexity of quadratic problem with $b$ variables is $O(b^3)$. Therefore, the time complexity of DensityRatio is $O((n_c + p^2)^3)$.

[4] The optimization problem of Diff-CLIME is a linear programming problem with $p^2$ variables. Based on the result from [6], the computational complexity of linear problem with $b$ variables is $O(b^4)$. Therefore, the time complexity of Diff-CLIME is $O((p^2)^4)$.

tural assumption:

**(C-Sparsity):** The 'true' canonical exponential family parameter for $\Delta^*$ (sparse change between two GGM structures) is exactly sparse with $k$ non-zero entries indexed by a supported set $S$. All other elements equal to 0 (in $S^c$).

**Theorem 2.1.** *Consider any differential network in Eq. (1.1) whose sparse canonical parameter $\Delta^*$ satisfies the **(C-Sparsity)** assumption. Suppose we compute the solution of Eq. (2.9) with a bounded $\lambda_n$ such that $\lambda_n \geq ||\Delta^* - \mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)||_\infty$, then the optimal solution $\widehat{\Delta}$ satisfies the following error bounds:*

$$\begin{aligned} ||\widehat{\Delta} - \Delta^*||_\infty &\leq 2\lambda_n \\ ||\widehat{\Delta} - \Delta^*||_F &\leq 4\sqrt{k}\lambda_n \\ ||\widehat{\Delta} - \Delta^*||_1 &\leq 8k\lambda_n \end{aligned} \tag{2.13}$$

*Proof.* See detailed proof in Section A.2.2 □

Theorem (2.1) provides a general bound for any selection of $\lambda_n$ and $\mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)$. We then use Theorem (2.1) to derive the statistical convergence rate of DIFFEE whose choice of the proxy backward mapping is $\mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c) = [T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}$. This gives us the following corollary:

**Corollary 2.2.** *Suppose the high-dimensional setting, i.e., $p > \max(n_c, n_d)$. Let $v := a\sqrt{\frac{\log p}{\min(n_c, n_d)}}$. Then for $\lambda_n := \frac{8\kappa_1 a}{\kappa_2}\sqrt{\frac{\log p}{\min(n_c, n_d)}}$ and $\min(n_c, n_d) > c \log p$, with a probability of at least $1 - 2C_1 \exp(-C_2 Kp \log(Kp))$, the estimated optimal solution $\widehat{\Delta}$ has the following error bound:*

$$\begin{aligned} ||\widehat{\Delta} - \Delta^*||_\infty &\leq \frac{16\kappa_1 a}{\kappa_2}\sqrt{\frac{\log p}{\min(n_c, n_d)}} \\ ||\widehat{\Delta} - \Delta^*||_F &\leq \frac{32\kappa_1 a}{\kappa_2}\sqrt{\frac{k \log p}{\min(n_c, n_d)}} \\ ||\widehat{\Delta} - \Delta^*||_1 &\leq \frac{64\kappa_1 a}{\kappa_2} k\sqrt{\frac{\log p}{\min(n_c, n_d)}} \end{aligned} \tag{2.14}$$

*where $a$, $c$, $\kappa_1$ and $\kappa_2$ are constants.*

*Proof.* See detailed proof in Section A.2.4 (especially from Eq. (A.31) to Eq. (A.36)). □

DIFFEE has achieved the same convergence rates as the Diff-CLIME[35] and the DensityRatio estimator [16]. The FusedGLasso estimator has not provided such convergence rate analysis.

To derive the statistical error bound of DIFFEE, we need to assume that $[T_v(\widehat{\Sigma}_c)]^{-1}$ and $[T_v(\widehat{\Sigma}_d)]^{-1}$ are well-defined. This is ensured by assuming that the true $\Omega_c^*$ and $\Omega_d^*$ satisfy the following conditions [31]:

**(C-MinInf$-\Sigma$):** The true $\Omega_c^*$ and $\Omega_d^*$ of Eq. (1.1) have bounded induced operator norm, i.e., $|||{\Omega_c}^*|||_\infty := \sup_{w \neq 0 \in \mathbb{R}^p} \frac{||{\Sigma_c}^* w||_\infty}{||w||_\infty} \leq \kappa_1$ and $|||{\Omega_d}^*|||_\infty := \sup_{w \neq 0 \in \mathbb{R}^p} \frac{||{\Sigma_d}^* w||_\infty}{||w||_\infty} \leq \kappa_1$.

**(C-Sparse-$\Sigma$):** The two true covariance matrices $\Sigma_c^*$ and $\Sigma_d^*$ are "approximately sparse" (following [3]). For some constant $0 \leq q < 1$ and $c_0(p)$, $\max_i \sum_{j=1}^p |[\Sigma_c^*]_{ij}|^q \leq c_0(p)$ and $\max_i \sum_{j=1}^p |[\Sigma_d^*]_{ij}|^q \leq c_0(p)$. [5]

We additionally require $\inf_{w \neq 0 \in \mathbb{R}^p} \frac{||\Omega_c^* w||_\infty}{||w||_\infty} \geq \kappa_2$ and $\inf_{w \neq 0 \in \mathbb{R}^p} \frac{||\Omega_d^* w||_\infty}{||w||_\infty} \geq \kappa_2$.

## 3 Experiments

We use two models of simulated datasets as well as a real world dataset for empirical comparisons.

- The first model mimics real world networks with a sparse differential network containing only hub nodes. This model can evaluate whether the method can efficiently infer the hub nodes in the differential network or not. In [35], the authors claim that if the change estimator also assumes the sparsity structure in $\Omega_c$ and $\Omega_d$, then the estimator cannot achieve a good result on datasets generated by this data model.
- The second data simulation model, in contrast, generates random graphs that differ by a sparse random differential network. It evaluates the estimation performance of a certain estimator for inferring the randomly-generated differential networks.
- The real world dataset is a human brain fMRI dataset with two groups of subjects: autism and control. Our choice of this dataset is motivated by the recent literature in neuroscience that has suggested functional networks are not sparse. On the other hand, differences in functional connections across subjects should be sparse [2].

The two simulation models allow for a thorough evaluation of DIFFEE vs the baseline methods. The real-world data allows us to compare DIFFEE versus the baselines through classification using the estimated differential graph.

### 3.1 Experimental Setup

**Baselines:** We compare DIFFEE with (1) FusedGLasso [9], (2) DensityRatio [17], and (3) Diff-CLIME [35].

**Evaluation metrics:** We evaluate DIFFEE and the baseline methods on F1-score and running time cost.

[5] This indicates for some positive constant $d$, $[\Sigma_c^*]_{jj} \leq d$ and $[\Sigma_d^*]_{jj} \leq d$ for all diagonal entries. Moreover, if $q = 0$, then this condition reduces to $\Sigma_d^*$ and $\Sigma_c^*$ being sparse.

More details in Section B.

**Hyper-parameters:** We need to tune the value of three hyper-parameters in these experiments: $v$, $\lambda_n$ and $\lambda_2$. In detail:

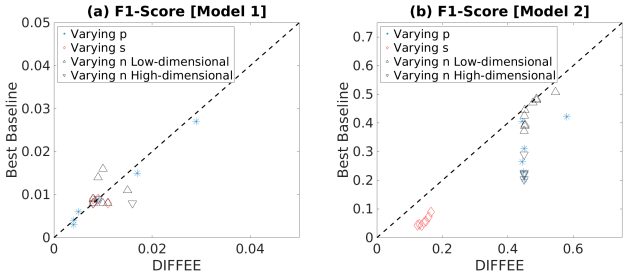
- $v$ is used for soft-thresholding in DIFFEE. We choose $v$ from the set $\{0.001i|i = 1, 2, \ldots, 1000\}$ and pick a value that makes $T_v(\Sigma_c)$ and $T_v(\Sigma_d)$ invertible.
- $\lambda_n$ is the main hyper-parameter that control the sparsity of the estimated differential network. Based on our convergence rate analysis in Section 2.5, $\lambda_n \geq C\sqrt{\frac{\log p}{\min(n_c,n_d)}}$. Accordingly, we choose $\lambda_n$ from a range of $\{0.01 \times \sqrt{\frac{\log p}{\min(n_c,n_d)}} \times i|i \in \{1, 2, 3, \ldots, 30\}\}$. The $\lambda_n$ in the DensityRatio is tuned by their package.
- $\lambda_2$ controls individual graph's sparsity in Fused-GLasso. We choose $\lambda_1 = 0.0001$ (a very small value) for all experiments to ensure only the differential network is sparse. $\lambda_2$ in the DensityRatio is set to 0.2 according to their package.

**Two models to generate simulated datasets:** Using the following two graph models, we generate multiple sets of synthetic multivariate-Gaussian datasets.

- **Model 1 – mimic real-world networks with hub nodes:** Inspired by [35], this model assumes that the graphs mimic real-world networks [21]. We first generate $\Omega_d$ as a network with $s \cdot \frac{p(p-1)}{2}$ edges following a power-law degree distribution with an expected power parameter of 2. Here $s$ is a parameter that controls the sparsity of the two graphs. A larger value of $s$ corresponds to denser graphs. Next, the value of each nonzero entry of $\Omega_d$ is generated from a uniform distribution with $[-10/p, -4/p] \cup [4/p, 10/p]$, where division by $p$ ensures the positive definiteness of $\Omega_c$ and $\Omega_d$. The diagonals are then set to 1 and $\Omega_d$ is symmetrized by averaging it with its transpose $(\frac{1}{2}(\Omega_d + \Omega_d^T))$. The differential network $\Delta$ is generated by the top 20% edges of the top 2 hub nodes in $\Omega_d$. $\Omega_c = \Omega_d - \Delta$.
- **Model 2 – random graph model:** Following [25], this model assumes $\Omega_c = \mathbf{B}_c + \mathbf{B}_S + \delta_c I$ and $\Omega_d = \mathbf{B}_d + \mathbf{B}_S + \delta_d I$, where each off-diagonal entry in $\mathbf{B}_c$ and $\mathbf{B}_d$ are generated independently and equals 0.5 with probability 0.1 and 0 with probability 0.9. The shared part $\mathbf{B}_S$ is generated independently and equal to 0.5 with probability $0.1s$ and 0 with probability $1 - 0.1s$. Similar to Model 1, $s$ controls the sparsity of the two graphs. $\delta_c$ and $\delta_d$ are selected large enough to guarantee the positive definiteness. A clear differential network structure $\Delta = \mathbf{B}_d - \mathbf{B}_c$ exists between these two graphs.

Following Model 1 or Model 2, for each case of simulated data generation, we generate two blocks of data samples following the distribution $N(0, (\Omega_c)^{-1})$ and $N(0, (\Omega_d)^{-1})$. Details see Section B.

Figure 2: F1-Score of DIFFEE vs the F1-Score of the best performing baseline. The more points below the diagonal line, the better. (a) On simulated datasets from Model 1 (b) On simulated datasets from Model 2. (Black up-triangles describe 'varying $(n_c, n_d)$ in low dimensions'; Black down-triangles describe 'varying $(n_c, n_d)$ in high-dimensions; Red diamonds represent 'varying $s$'; and Blue stars represent 'varying $p$'.)




### 3.2 Experiments on Simulated Datasets

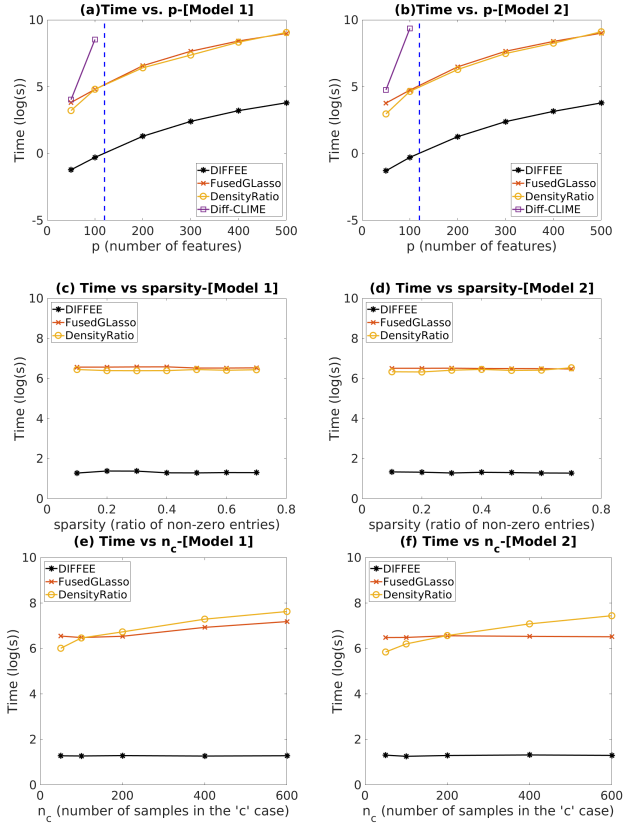
**Experimental Design:** By varying the number of features $p$, amount of sparsity $s$, and the number of samples ($n_c$,$n_d$), we can generate many cases of simulated datasets. This allows us to comprehensively evaluate DIFFEE across a wide range of data situations. To this end, we design the following three sets of synthetic experiments by varying $p$, $s$, $n_c$, and $n_d$:

- $p$ (the number of features): The first set of experiments varies $p$ in the set of $\{50, 100, 200, 300, 400, 500\}$ while setting $n_c$ and $n_d$ as $p/2$ and the sparsity parameter $s = 0.2$.
- $s$ (the sparsity): In the second set of experiments, we vary the value of the sparsity parameter $s$ in the set of $\{0.1, 0.2, \ldots, 0.7\}$, while using $p = 200$ and $n_c = n_d = p/2$.
- $n_c$ and $n_d$ (the number of samples): In the third set, we vary the number of samples in both groups and set $p = 200$ and $s = 0.2$. We group this set of experiments into two categories: low-dimensional cases, and high-dimensional cases. For the high dimensional case, we vary $n_c$ and $n_d$ from the value set of $\{p/4, p/2\}$. Similarly, for the low dimensional case, we vary $n_c$ and $n_d$ from the value set of $\{p, 2p, 3p\}$.

**Experiment Results:** We compare DIFFEE with the baselines regarding two aspects– (a) Effectiveness, and (b) Scalability.

**(a) Effectiveness**: We evaluate the prediction effectiveness of using F1-Score. Figure 2 presents the summarized results of our DIFFEE versus baselines on all 50 cases of simulated datasets. As explained above, the simulated datasets are generated by varying the parameters $p$, $s$, $n_c$, and $n_d$ by data Model 1 and Model 2. In Figure 2 (a) and (b), we plot the F1-Score of DIFFEE vs the F1-Score of the best performing baseline on each simulated case from Model 1 and Model 2, respectively. Each point in the two figures is obtained by comparing DIFFEE vs. the best baseline among all baselines on

Figure 3: Time Cost(log(seconds)) of DIFFEE versus the baseline methods (a) Time vs. number of features(p) for Model 1. (b)Time vs. number of features(p) for Model 2. (c) Time vs. sparsity(s) for Model 1. (d) Time vs. sparsity(s) for Model 2. (e)Time vs. number of samples in 'c' case ($n_c$) for Model 1. (f) Time vs. number of samples in the 'c' case ($n_c$) for Model 2.




one simulated case. Each point below the line $y = x$ indicates that DIFFEE achieves better performance over baselines. Overall Figure 2 shows that DIFFEE outperforms the corresponding best baseline in almost all cases. The only two points for which DIFFEE doesn't do as well as the corresponding best baseline DensityRatio are two low dimensional cases. This is as expected because the design of DIFFEE is for high-dimensional cases (i.e., the choices of proxy backward mapping). Details of F1-Scores from all simulation cases and discussions of low F1 values on Model 1 are in Appendix.

**(b) Scalability**: To evaluate DIFFEE and the baselines on scalability, Figure 3 presents the time cost vs. varying $p$, varying sparsity ($s$) and varying number of samples in the 'c' group ($n_c$). Figure 3 (a),(c) and (e) show time results from data Model 1. Figure 3 (b),(d) and (f) correspond to datasets from Model 2. We interpolate the points of computation time from each estimator into curves. For each simulation case, the computation time for each estimator is the summation of a method's execution time over all values of $\lambda_n$. Figure 3 shows that in general the time costs of FusedGLasso and DensityRatio are roughly comparable. DIFFEE is about 100 times better than both (detailed numbers are provided in Table 3 to Table 10). Diff-CLIME is extremely slow when $p$ increases. Because Figure 3 (c),(d),(e) and (f) are about data cases with $p = 200$, we can not run Diff-CLIME on these cases (it cannot finish any $p = 200$ case for a single value of $\lambda_n$ by one day). Interestingly, the empirical time results match the computational analysis in Table 1. Especially DensityRatio's time cost grows quickly when $n_c$ increases. In contrast the running time of DIFFEE and FusedGLasso are not connected strongly to the size of samples. Overall DIFFEE costs much less computation time than the baselines and can significantly scale up to larger $p$.

### 3.3 A Real-World Dataset about Functional Connectivity among Brain Regions

We then use DIFFEE for a classification task on a well-known human brain fMRI dataset: ABIDE[10].

**ABIDE Dataset:** This data is from the Autism Brain Imaging Data Exchange (ABIDE) [10], a publicly available resting-state fMRI dataset. The ABIDE data aims to understand human brain connectivity and how it reflects neural disorders [27]. The data is retrieved from the Preprocessed Connectomes Project [7], where preprocessing is performed using the Configurable Pipeline for the Analysis of Connectomes (CPAC) [8] without global signal correction or band-pass filtering. After preprocessing with this pipeline, 871 individuals remain (468 diagnosed with autism). Signals for the 160 (number of features $p = 160$) regions of interest (ROIs) in the often-used Dosenbach Atlas [11] are examined.

**Cross-validation:** Classification is performed using the 3-fold cross-validation suggested by the literature [22][28]. The subjects are randomly partitioned into three equal sets: a training set, a validation set, and a test set. Each estimator produces $\widehat{\Delta}$ using the training set. Then, these differential networks are used as inputs to linear discriminant analysis (LDA), which is tuned via cross-validation on the validation set. Finally, accuracy is calculated by running LDA on the test set. This classification process aims to assess the ability of an estimator to learn the differential patterns of the connectome structures. Notably, the DensityRatio method cannot be compared on this data, because the method does not provide the precision matrices necessary for LDA.

**Classification Results:** Table 2 displays the maximum accuracy achieved by DIFFEE, FusedGLasso, and Diff-CLIME, after tuning over hyperparameters. DIFFEE yields a classification accuracy of 57.58% distinguishing the autism and control groups, outperforming the FusedGLasso and Diff-CLIME estimators.

Table 2: Classification accuracy obtained on the ABIDE dataset using DIFFEE, FusedGLasso, and Diff-CLIME. DIFFEE achieves the highest classification accuracy.

| Method | DIFFEE | FusedGLasso | Diff-CLIME |
|---|---|---|---|
| Accuracy (%) | **57.58%** | 56.90% | 53.79% |

## 4 Conclusion

This paper proposes a simple closed-form estimator, DIFFEE for learning sparse change between two GGM structures. DIFFEE can scale up to large-scale settings ($p > 100$) and achieves the same asymptotic convergence rate as previous estimators. Empirically DIFFEE improves the state-of-the-art with better F1-scores and cheaper time cost (about 100 times faster).

## References


[1] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

[2] E. Belilovsky, G. Varoquaux, and M. B. Blaschko. Testing for differences in gaussian graphical models: applications to brain connectivity. In *Advances in Neural Information Processing Systems*, pages 595–603, 2016.

[3] P. J. Bickel and E. Levina. Covariance regularization by thresholding. *The Annals of Statistics*, pages 2577–2604, 2008.

[4] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[5] T. Cai, W. Liu, and X. Luo. A constrained l1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

[6] T. H. Cormen. *Introduction to algorithms*. MIT press, 2009.

[7] C. Craddock. Preprocessed connectomes project: open sharing of preprocessed neuroimaging data and derivatives. In *61st Annual Meeting*. AACAP, 2014.

[8] C. Craddock, S. Sikka, B. Cheung, R. Khanuja, S. Ghosh, C. Yan, Q. Li, D. Lurie, J. Vogelstein, R. Burns, et al. Towards automated analysis of connectomes: The configurable pipeline for the analysis of connectomes (c-pac). *Front Neuroinform*, 42, 2013.

[9] P. Danaher, P. Wang, and D. M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2013.

[10] A. Di Martino, C.-G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, M. Assaf, S. Y. Bookheimer, M. Dapretto, et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19(6):659–667, 2014.

[11] N. U. Dosenbach, B. Nardos, A. L. Cohen, D. A. Fair, J. D. Power, J. A. Church, S. M. Nelson, G. S. Wig, A. C. Vogel, C. N. Lessov-Schlaggar, et al. Prediction of individual brain maturity using fmri. *Science*, 329(5997):1358–1361, 2010.

[12] F. Fazayeli and A. Banerjee. Generalized direct change estimation in ising model structure. In *International Conference on Machine Learning*, pages 2281–2290, 2016.

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

[14] T. Ideker and N. J. Krogan. Differential network biology. *Molecular systems biology*, 8(1):565, 2012.

[15] S. L. Lauritzen. *Graphical models*. Oxford University Press, 1996.

[16] S. Liu, K. Fukumizu, and T. Suzuki. Learning sparse structural changes in high-dimensional markov networks. *Behaviormetrika*, 44(1):265–286, 2017.

[17] S. Liu, J. A. Quinn, M. U. Gutmann, T. Suzuki, and M. Sugiyama. Direct learning of sparse changes in markov networks by density ratio estimation. *Neural computation*, 26(6):1169–1197, 2014.

[18] K. V. Mardia, J. T. Kent, and J. M. Bibby. Multivariate analysis. 1980.

[19] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, pages 1436–1462, 2006.

[20] S. Negahban, B. Yu, M. J. Wainwright, and P. K. Ravikumar. A unified framework for high-dimensional analysis of $m$-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009.

[21] M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[22] R. A. Poldrack, P. C. Fletcher, R. N. Henson, K. J. Worsley, M. Brett, and T. E. Nichols. Guidelines for reporting an fmri study. *Neuroimage*, 40(2):409–414, 2008.

[23] P. Ravikumar, M. J. Wainwright, G. Raskutti, B. Yu, et al. High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

[24] B. D. Ripley. *Stochastic simulation*, volume 316. John Wiley & Sons, 2009.

[25] A. J. Rothman, P. J. Bickel, E. Levina, J. Zhu, et al. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

[26] A. J. Rothman, E. Levina, and J. Zhu. Generalized thresholding of large covariance matrices. *Journal of the American Statistical Association*, 104(485):177–186, 2009.

[27] D. C. Van Essen, S. M. Smith, D. M. Barch, T. E. Behrens, E. Yacoub, K. Ugurbil, W.-M. H. Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

[28] G. Varoquaux, A. Gramfort, J.-B. Poline, and B. Thirion. Brain covariance selection: better individual functional connectivity models using population prior. In *Advances in neural information processing systems*, pages 2334–2342, 2010.

[29] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[30] P. Xu and Q. Gu. Semiparametric differential graph models. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1064–1072. Curran Associates, Inc., 2016.

[31] E. Yang, A. C. Lozano, and P. K. Ravikumar. Elementary estimators for graphical models. In *Advances in Neural Information Processing Systems*, pages 2159–2167, 2014.

[32] M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[33] B. Zhang and Y. Wang. Learning structural changes of gaussian graphical models in controlled experiments. *arXiv preprint arXiv:1203.3532*, 2012.

[34] S. D. Zhao, T. T. Cai, and H. Li. Direct estimation of differential networks. page asu009.

[35] S. D. Zhao, T. T. Cai, and H. Li. Direct estimation of differential networks. *Biometrika*, 101(2):253–268, 2014.