
Spectral Differential Network Analysis for High-Dimensional Time Series

Michael Hellstern¹

Byol Kim²

¹University of Washington

Zaid Harchaoui¹

Ali Shojaie¹

²Sookmyung Women’s University

Abstract

Spectral networks derived from multivariate time series data arise in many domains, from brain science to Earth science. Often, it is of interest to study how these networks change under different conditions. For instance, to better understand epilepsy, it would be interesting to capture the changes in the brain connectivity network as a patient experiences a seizure, using electroencephalography data. A common approach relies on estimating the networks in each condition and calculating their difference. Such estimates may behave poorly in high dimensions as the networks themselves may not be sparse in structure while their difference may be. We build upon this observation to develop an estimator of the difference in inverse spectral densities across two conditions. Using an ℓ_1 penalty on the difference, consistency is established by only requiring the difference to be sparse. We illustrate the method on synthetic data experiments and on experiments with electroencephalography data.

1 INTRODUCTION

Spectral network analysis of multivariate time series plays a key role in fields ranging from oceanography and seismology to neuroscience (Laurindo et al., 2019; James et al., 2017; Bloch et al., 2022). We use the inverse spectral density matrix as our choice of network as the i, j entry more directly encodes the dependence between nodes i and j compared to other networks in the spectral domain such as coherence Dahlhaus (2000). Studying individual networks can provide in-

sights into how features interact; however, it is often of interest to study how networks change across conditions or in response to an external intervention (Shojaie, 2021). In neuroscience, for example, many neurodegenerative disorders are associated with abnormal brain connectivity networks (Bloch et al., 2022). Spectral features are regularly used to interpret many types of neuroscientific data, from electroencephalography to magnetoencephalography data (Gnassounou et al., 2023; Richard et al., 2020; Dupré la Tour et al., 2018).

Coherence, the frequency domain analog to correlation, is a common choice for investigating interactions in multivariate time series analysis. This notion is especially appealing in neuroscience applications, where activities captured at different frequencies better reveal brain oscillations in sensory-cognitive processes. Therefore, despite the availability of nonlinear association measures, such as mutual information (Belghazi et al., 2018) and transfer entropy (Ursino et al., 2020), coherence is commonly used by neuroscientists to define brain functional connectivity networks.

Similar to correlation, coherence includes the indirect effects of other nodes in the network. Thus, coherence may not be an informative measure of the dependence between nodes. The inverse spectral density is a more direct measure of dependence between nodes as the inverse spectral density between nodes i and j is a rescaling of the coherence after removing the linear effects of all other nodes $\{k \neq i, j\}$ (Dahlhaus, 2000). Therefore, the inverse spectral density better resembles the effective connectivity between brain regions (Friston, 2011), providing an initial understanding of how two regions may be causally related.

Modern data collection methods have facilitated the collection of data, where the dimensionality is much greater than the number of data points. This setting is often referred to as high-dimensional ($p \gg n$). When analyzing high-dimensional time series data, regularization techniques, such as the LASSO, are needed to make the problem computationally and statistically tractable (Banerjee et al., 2008). In the frequency domain, regularization can allow for better numerical sta-

bility and better performance (Böhm and von Sachs, 2009).

While several methods exist to directly estimate the difference in inverse covariance matrices (see Tsai et al. (2022) and references therein), no such methods exist for the difference in inverse spectral densities. Naïvely, to estimate the difference in inverse spectral densities, one could first estimate the inverse spectral density in each condition and then take their difference. In high-dimensions, consistency of the resulting estimate requires both networks to be sparse (Deb et al., 2024). This assumption may be unrealistic due to the presence of hub nodes (Wang et al., 2021) and thereby lead to degraded statistical performance.

In this paper, we propose Spectral D-trace Difference (SDD), a direct estimator of the difference in inverse spectral densities, which, to our knowledge, is the first method to directly target a differential network in the spectral domain. By leveraging advances in analysis of time series data, our direct estimator only requires sparsity of the difference, which is a more realistic assumption in many settings (Wang et al., 2021). For instance, despite identifying many nonzero coherence values in resting state and stimulation states, Bloch et al. (2022) found many coherence changes to be near zero. These small differences are likely due to noise in the estimation procedure, corresponding to no underlying change in coherence. Assuming sparsity of the difference, and only under a geometrically decaying time dependence condition, we show that our direct difference estimator consistently estimates the true difference. Performance of our estimator is studied in a variety of simulation settings, as well as a real data application.

Notation. We use $\|A\|_1$, $\|A\|_F$, $\|A\|_\infty$, $\|A\|_0$ to denote the ℓ_1 , Frobenius, infinity, and ℓ_0 norms of a matrix A , respectively. The element-wise absolute value of a matrix A is denoted as $|A|$. The conjugate transpose of a matrix or vector will be denoted as A^H . We denote the inner product between two matrices A and B as $\langle A, B \rangle = \text{Tr}(AB^T)$ and use $A \otimes B$ to represent their Kronecker product.

2 METHOD

Figure 1 provides a visual explanation of our method, SDD, and each step (A)–(E) is elaborated upon in this section.

(A) Inputs. Suppose we have mean-zero piecewise stationary data in two conditions, denoted as 1 and 2. In condition 1, there are n_1 observations of p covariates, while condition 2 has n_2 observations. It is worth

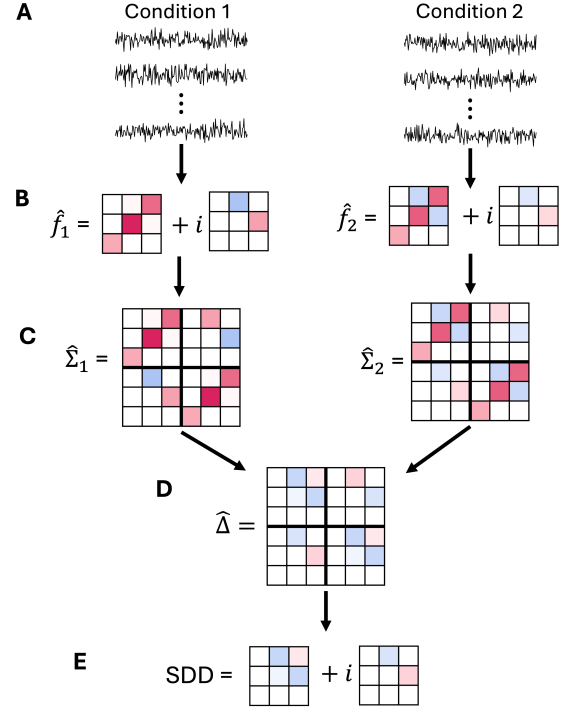


Figure 1: SDD Method. **(A) Inputs:** Time series datasets are observed for two conditions. **(B) Computing the Spectral Densities:** We then compute the spectral densities in each condition using the smoothed periodogram. **(C) Expanding to Real Space:** These spectral densities are expanded to the real space. **(D) Direct Estimation of the Difference in Inverse Spectral Densities:** Using the expanded spectral densities we directly compute the difference estimator. **(E) Output:** We form the difference in inverse spectral densities by taking the (1,1) and (2,1) blocks of $\hat{\Delta}$.

noting that the assumption of piecewise stationarity only requires stationarity within each condition and not stationarity across conditions. This may be satisfied in many neuroscience applications where experimental factors affecting the brain state are known from the experimental design. For example, in Bloch et al. (2022), conditions 1 and 2 might represent pre- and post-stimulation, respectively and stimulation times are known.

(B) Computing the Spectral Densities. The spectral density in condition $l \in \{1, 2\}$ at frequency λ , $f_l(\lambda)$, is defined as the Fourier transform of the autocovariance matrix. Intuitively, the spectral density represents the portion of the covariance of the signal that can be attributed to the specific frequency λ (Brockwell and Davis, 1991, pp. 331–332). Mathematically we can write our data as $\{\mathbf{x}_{l,t}\}$ for $t = 1, \dots, n_l$

where $\mathbf{x}_{l,t}$ is a p -dimensional vector in condition $l \in \{1, 2\}$. For each condition l , the autocovariance at lag $h \in \mathbb{Z}$ is denoted as $\Gamma_l(h) = \mathbb{E}(\mathbf{x}_{l,t}\mathbf{x}_{l,t+h}^T)$. If $\sum_{h=-\infty}^{\infty} |\Gamma_l(h)| < \infty$ then the spectral density in condition l at frequency λ exists and is defined as

$$f_l(\lambda) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} e^{-i\lambda h} \Gamma_l(h), \quad -\pi \leq \lambda \leq \pi.$$

To estimate the difference of *inverse* spectral densities, we will require estimates of the spectral densities themselves. We use the smoothed periodogram to estimate the spectral density matrix and refer to it as \hat{f}_l . To define the smoothed periodogram, we must first define the periodogram. For condition l , the periodogram at Fourier frequency $\{\lambda_j = 2\pi j/n_l, -\lfloor(n_l - 1)/2\rfloor \leq j \leq \lfloor n_l/2\rfloor\}$ is defined as

$$P_l(\lambda_j) = \frac{(\sum_{t=1}^{n_l} \mathbf{x}_{l,t} e^{-i\lambda_j t}) (\sum_{t=1}^{n_l} \mathbf{x}_{l,t} e^{-i\lambda_j t})^H}{2\pi n_l}.$$

Recall \mathbf{x}^H is the conjugate transpose of the vector \mathbf{x} . Using the Fast Fourier transform (FFT) algorithm, the periodogram can be computed quickly. The periodogram is known to be an inconsistent estimator of the spectral density so a smoothed version is used where the periodograms of $2M_{n_l}$ nearby frequencies are averaged (Brockwell and Davis, 1991, pp. 347–350). M_{n_l} is referred to as the bandwidth or smoothing span. That is, the smoothed periodogram at fourier frequency λ_j is generated as

$$\hat{f}_l(\lambda_j) = \frac{1}{2M_{n_l} + 1} \sum_{k=j-M_{n_l}}^{j+M_{n_l}} P_l(\lambda_k).$$

(C) Expanding to the Real Space. In practice, many different λ values are of interest, but for notational simplicity we assume λ is fixed throughout and suppress the dependence on λ . The spectral density, f_l , is complex-valued in general which can complicate the estimation. To transform this problem to the real space, we note that any complex matrix can be written as the sum of a real matrix and i times another real matrix (for $i^2 = -1$). Specifically we write the spectral density and its inverse as $f_l = A_l + iB_l$, and $f_l^{-1} = \tilde{A}_l + i\tilde{B}_l$, respectively, where A_l , B_l , \tilde{A}_l , and \tilde{B}_l are real matrices. With these representations, we can now work in the real space by studying $\Sigma_l = \begin{bmatrix} A_l & -B_l \\ B_l & A_l \end{bmatrix} \in \mathbb{R}^{2p \times 2p}$ instead of f_l . By Lemma A.1 of Fiecas et al. (2019),

$$\begin{bmatrix} A_l & -B_l \\ B_l & A_l \end{bmatrix} \begin{bmatrix} \tilde{A}_l & -\tilde{B}_l \\ \tilde{B}_l & \tilde{A}_l \end{bmatrix} = \Sigma_l \Sigma_l^{-1} = I_{2p},$$

where I_{2p} represents the $2p \times 2p$ identity matrix. In this representation, Σ_l^{-1} is the expansion of f_l^{-1} to the real space. Moreover, f_l and f_l^{-1} can easily be recovered by taking the (1,1) and (2,1) blocks of Σ_l and Σ_l^{-1} , respectively.

(D) Direct Estimation of the Difference in Inverse Spectral Densities. Having defined the problem in the real space, we use the ℓ_1 penalized D-trace loss function (Yuan et al., 2017) to directly estimate $\Delta = \Sigma_1^{-1} - \Sigma_2^{-1}$. To justify this estimation strategy, in Appendix A.1 we show that this loss is convex and the population loss is minimized at the true $\Delta = \Sigma_1^{-1} - \Sigma_2^{-1}$. The inputs to this estimator are $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$ which are formed by expanding \hat{f}_1 and \hat{f}_2 to the real space respectively. Our estimator is

$$\hat{\Delta} = \arg \min_{\Delta} \frac{1}{4} \left(\langle \hat{\Sigma}_2 \Delta, \Delta \hat{\Sigma}_1 \rangle + \langle \hat{\Sigma}_1 \Delta, \Delta, \hat{\Sigma}_2 \rangle \right) - \langle \Delta, \hat{\Sigma}_2 - \hat{\Sigma}_1 \rangle + \tau_{n_1, n_2} \|\Delta\|_1 \quad (1)$$

where τ_{n_1, n_2} is a penalty parameter depending on sample sizes n_1, n_2 and $\|\Delta\|_1$ is the ℓ_1 penalty which returns the sum of the absolute values of all entries of a matrix. We solve for $\hat{\Delta}$ using the alternating direction method of multipliers (ADMM) algorithm from Yuan et al. (2017). Since we are inputting matrices $\hat{\Sigma}_i \in \mathbb{R}^{2p \times 2p}$, the computational complexity of each iteration of this algorithm is $O(8p^3)$ while the memory requirements are $O(4p^2)$.

(E) Output. Recall that the inverse spectral density, f_l^{-1} , represents the coherence between two nodes after removing the linear effects of all other nodes. Our goal is to estimate how this effective connectivity differs between the two conditions. That is, we wish to estimate $f_1^{-1} - f_2^{-1}$. Using our representation in the real space, the true difference $f_1^{-1} - f_2^{-1}$ is recovered by taking the (1,1) and (2,1) blocks of

$$\Sigma_1^{-1} - \Sigma_2^{-1} = \begin{bmatrix} \tilde{A}_1 & -\tilde{B}_1 \\ \tilde{B}_1 & \tilde{A}_1 \end{bmatrix} - \begin{bmatrix} \tilde{A}_2 & -\tilde{B}_2 \\ \tilde{B}_2 & \tilde{A}_2 \end{bmatrix} := \Delta.$$

Similarly, using our estimate $\hat{\Delta}$ of Δ we can estimate $f_1^{-1} - f_2^{-1}$ using the (1,1) and (2,1) blocks of $\hat{\Delta}$.

In the following section, we study the consistency of our estimator. Intuitively, if \hat{f}_l estimates the spectral density well, we should be able to estimate Δ well. Our theoretical results show that this is indeed the case and the rate at which the smoothed periodogram \hat{f}_l converges to the true periodogram is also the rate at which $\hat{\Delta}$ converges to Δ . Due to the dependence in the data, the smoothed periodogram and $\hat{\Delta}$ are not \sqrt{n} consistent as one might expect in the i.i.d case. Rather, we show that for an optimal choice of smoothing span, \hat{f}_l and $\hat{\Delta}$ converge to f_l and Δ respectively at a rate of $n^{-1/3}$.

3 THEORY

In this section, we study the theoretical properties of our SDD estimator, $\hat{\Delta}$. Specifically, we establish the convergence rate of $\hat{\Delta}$ to the population quantity Δ . From Equation 1, our estimator $\hat{\Delta}$ relies on expanded estimates of the spectral density $\hat{\Sigma}_1, \hat{\Sigma}_2$. Thus, it is natural that the rate of convergence of $\hat{\Delta}$ to Δ relies on the rates of convergence of \hat{f}_1 and \hat{f}_2 to f_1 and f_2 .

Theorem 1. *Suppose Assumption 1 in Appendix B holds and that $\tau_{n_1, n_2} \geq O(\min(n_1, n_2)^{-1/3})$ and $M_{n_l} = O(n_l^{2/3})$. For n_1 and n_2 large enough and for any $H > 0$, there exists some $C > 0$ such that with probability greater than $1 - Cp^2 \min(n_1, n_2)^{-H}$,*

$$\|\hat{\Delta} - \Delta\|_F \leq O\left(s_{\Delta} \min(n_1, n_2)^{-1/3}\right),$$

where s_{Δ} is the number of non-zero elements in the true difference matrix Δ .

The full proof of Theorem 1 is given in Appendix C. To establish convergence rates of \hat{f}_1 and \hat{f}_2 in high-dimensions, we rely in part on Theorem 3.1 of Fiecas et al. (2019), which uses the functional dependency framework of Wu (2005) and Wu and Zaffaroni (2018). Specifically, our convergence rates hinge on the assumption of a geometrically decaying dependence in the data. An overview of the functional dependence framework is given in Appendix B and a formal statement of the geometric decay assumption is given in Assumption 1 in Appendix B. The geometric decay assumption is mild and is satisfied by many processes including ARMA, ARMA-ARCH, and ARMA-GARCH processes, as well as other nonlinear autoregressive processes (Shao and Wu, 2007; Liu and Wu, 2010).

While our method uses a similar setup to Yuan et al. (2017), the analysis of our estimator is fundamentally different in two ways. First, we establish Theorem 1 only assuming sparsity of the difference and we explicitly avoid the irrepresentability assumption of Yuan et al. (2017) which has been shown not to hold in practice (Zhao and Yu, 2006). For example, Wang et al. (2021) show through simulations that for graphs of moderate or larger dimensions, this condition almost never holds. This is true even for very sparse differences and weak data dependence. Second, in contrast to the proof of Yuan et al. (2017), which assumes i.i.d. observations, our proofs also leverage advances in theoretical analysis of high-dimensional time series to handle the temporal dependence among observations over time. It is also worth noting that Theorem 1 also applies to differences in inverse spectral densities across frequencies.

We now discuss interesting intermediate results from the proof. In Lemma 1, we show that, with high

probability and a smoothing span of $M_{n_l} = O(n_l^{2/3})$, the deviations of the smoothed periodogram from the true spectral density in condition l are $O(n_l^{-1/3})$. Lemma 2 shows that with high probability, the estimated Hessian of the D-trace loss converges to the true Hessian at a rate of $O(\min(n_1, n_2)^{1/3})$. That is, $\|\hat{\Gamma} - \Gamma\|_{\infty} = O(\min(n_1, n_2)^{1/3})$ where $\hat{\Gamma} = 0.5(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1)$ and Γ is the same but without the hats. Lastly, Equation 6 establishes that, with high probability, the maximum norm of deviations of the derivative of the D-trace loss, $\left\|0.5(\hat{\Sigma}_1 \Delta \hat{\Sigma}_2 + \hat{\Sigma}_2 \Delta \hat{\Sigma}_1) - (\hat{\Sigma}_1 - \hat{\Sigma}_2)\right\|_{\infty}$, is $O(\min(n_1, n_2)^{-1/3})$.

Due to the dependence in the data, the convergence rate in Theorem 1 is slower than the $n^{-1/2}$ rates for i.i.d data. This is because, compared with parametric models that amount to fast decaying dependence among observations, our dependence assumption is very mild and general. In fact, as we choose M_n to minimize the deviations in Theorem 3.1 of Fiecas et al. (2019), our rate is the best we can expect to achieve using this general dependence framework when only assuming a geometrically decaying time dependence.

4 SIMULATIONS

In this section, we evaluate the performance of the proposed estimator $\hat{\Delta}$ using simulation studies. The code to reproduce these results as well as those in Section 5 are available at <https://github.com/mikehellstern/spectral-differential-networks>. All simulations use VAR(1) processes, with $p = 54$, as this allows us to compute the true spectral density using results from Sun et al. (2018). We control the sparsity of the difference in inverse spectral densities by using a block diagonal structure. In all settings, the transition matrix in condition 1 is the same as that in condition 2 except the last 3×3 block is multiplied by -1 . The first setting uses similar coefficients as in Sun et al. (2018). In the second and third simulation settings, the transition matrix consists of one 51×51 block and one 3×3 block. The larger block was generated with 60% and 95% sparsity in the second and third settings, respectively. All simulations were performed using $n = 100, 200, 500, 1000, 2000$ observations for both conditions. Smoothed periodograms were computed from data for each condition using a smoothing window of $M_n = \lceil n^{2/3} \rceil$ and then converted to the real space to generate $\hat{\Sigma}_l$. All computations were performed on a MacBook with a 2.3 GHz Quad-Core Intel Core i7 processor and 16 GB of memory. More information on the simulation setup, optimization techniques, and tuning parameter selection can be

found in Appendix D.

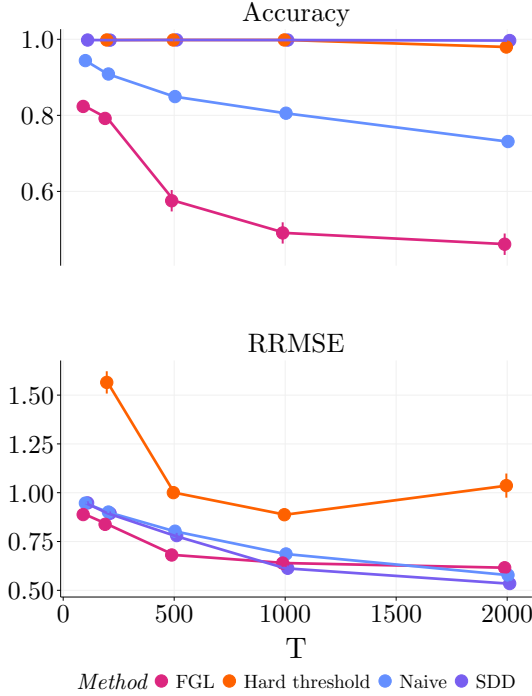


Figure 2: Simulation 1. Results are reported as mean (dots) and SE (vertical lines) where the mean and SE are taken across all frequencies for a given sample size n .

As previously noted, our method is the first to directly estimate the difference in inverse spectral densities and no direct competitors or state-of-the-art methods exist. However, in this section we will discuss and compare several alternative methods. While not focused on directly estimating the difference, a related method is the joint graphical lasso with a fusion penalty (FGL, Danaher et al., 2014). This method aims to simultaneously estimate inverses in each condition that are believed to share structural similarities. To encourage similarities in the inverse estimates, FGL uses an ℓ_1 penalty on their difference. In simulations, FGL was tuned using AIC from Danaher et al. (2014). We have also conceptualized two additional methods to compare to SDD. The first is a Naïve method, which estimates Δ by taking the difference after estimating individual inverse spectral densities separately, $\hat{\Delta}_N = \hat{\Sigma}_1^{-1} - \hat{\Sigma}_2^{-1}$. For the Naïve method, the graphical LASSO (GLASSO, Friedman et al., 2008) was used to estimate a sparse inverse spectral density in each condition. Specifically we used the fast implementation from Sustik and Calderhead (2012) available in the `glassoFast` package in R (GPL (≥ 3.0)).

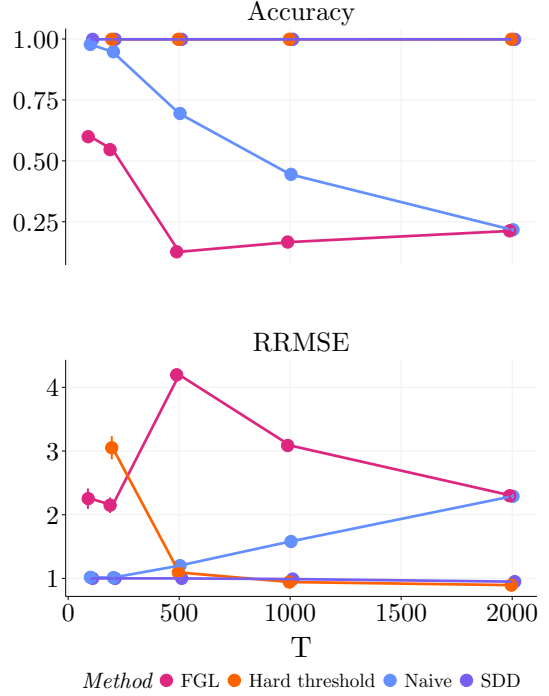


Figure 3: Simulation 2. Results are reported as mean (dots) and SE (vertical lines) where the mean and SE are taken across all frequencies for a given sample size n .

To induce sparsity in individual estimates and (potentially) their difference, GLASSO was tuned for each condition separately using eBIC with $\gamma = 0.5$ (Foygel and Drton, 2010). The next method directly thresholds the difference in inverse spectral densities to induce sparse estimates. Specifically, we estimate the inverse spectral density in each condition and take the difference and then use Hard thresholding. The thresholds were tuned using the same eBIC as SDD given in (7). Similar to Deb et al. (2024), all methods are compared based on the following metrics: Accuracy, Precision, Recall, and Relative Root Mean Square Error (RRMSE). The full definitions can be found in Appendix D.

Simulation results for Accuracy and RRMSE in simulations 1 and 2 are reported in Figures 2 and 3. Results for Precision and Recall for simulations 1 and 2 can be found in Appendix D Figures 8 and 9. Results from simulation 3 are in Appendix D, Figures 10 and 11. The full set of results for each of the simulations are in Appendix D Tables 1, 2, 3. In general, SDD has better accuracy than both FGL and the Naïve method with the difference in accuracy increasing as the sam-

ple size increases. This is likely due to the fact that, in general, SDD estimates much fewer edges than either method. This also results in fewer false positive edges but more false negatives, resulting in better precision for SDD at the expense of reduced recall; see Table 1 in Appendix D. Across all settings and sample sizes we also see that SDD generally has either lower or similar RRMSE compared to other methods. Overall, compared to FGL and the Naïve method, SDD identifies a similar number of true edges and achieves similar or lower RRMSE using a much smaller edge set. Compared to Hard thresholding, SDD achieves much better RRMSE and recall across all settings and sample sizes except setting 2 for large n where Hard thresholding slightly outperforms SDD in terms of RRMSE. To further support our analysis, we have also summarized the difference between SDD and each of the competing methods for each metric in Appendix D, Tables 4, 5, 6, 7, 8, 9, 10, 11, 12.

5 APPLICATION TO EEG DATA

We apply our direct difference estimator (SSD) and competing estimators to electroencephalograms (EEG) data from Hatlestad-Hall et al. (2022). Data were recorded with a 64 channel EEG array for 111 healthy subjects at a sampling frequency of 1024 Hz. Four minutes of brain activity was recorded while subjects were resting with their eyes closed. For 42 subjects, a second session was recorded 2–3 months after the initial session. For our analysis, we used the pre-cleaned data provided in OpenNeuro Dataset ds003775 (Markiewicz et al., 2021). Specific cleaning steps can be found in Hatlestad-Hall et al. (2022). We also downsampled the data to 512 Hz.

To validate our method, we analyzed the subjects with follow-up and without follow-up separately. For those with follow-up, we estimated the difference in networks from the first to the second session (*across session analysis*). For those without, we estimated the difference in networks from 0-60s to 120-180s (*within session analysis*). The 60-120s block was used as a rest. It has been shown that brain networks are temporally dynamic (Zalesky et al., 2014; Nobukawa et al., 2019). Therefore, *a priori*, we expect the differential networks to be sparser in the within session analysis compared to the across session analysis and finding such results would provide validation of our method.

We compare the sparsity of the estimated differential network for SDD, FGL, the Naïve, and Hard threshold methods across the commonly-used Theta, Beta, Gamma and High-Gamma bands. Sparsity was defined as the proportion of entries in the estimated difference that were non-zero. Five or six evenly spaced

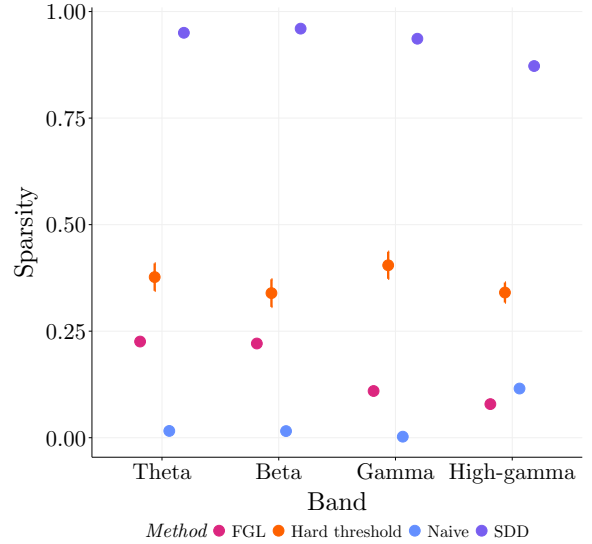


Figure 4: EEG Within Session Analysis. Vertical lines indicate SE.

frequencies were considered for each band. They were (in Hz): Theta: (4,5,6,7,8), Beta: (12, 16, 20, 24, 28), Gamma: (30, 40, 50, 60, 70), High-gamma: (80, 95, 110, 125, 140, 150). Results averaged over subjects and frequencies within each band are presented in Figures 4 and 5.

Figures 4 and 5 shows that SDD is the only method that estimates a sparser difference in the within session analysis compared to the across session analysis across all frequency bands. On the other hand, competing methods have similar or less sparsity in the within session analysis compared to the across-session analysis. These results clearly indicate that SDD estimates are more consistent with the experimental setting in this application.

6 APPLICATION TO SIMULATED μ ECoG DATA

We next apply our SDD estimator to simulated micro-electrocorticography (μ ECoG) data. The simulation design is based on the optogenetic stimulation μ ECoG experiments from Yazdan-Shahmorad et al. (2016, 2018); Bloch et al. (2022). Studying how stimulation changes the brain network using such experiments can help design protocols to change the brain network in a targeted way. This can lead to treatments for conditions, such as schizophrenia or epilepsy, which are known to stem from abnormal brain connectivity networks (Bloch et al., 2022). Our simulations used a sim-

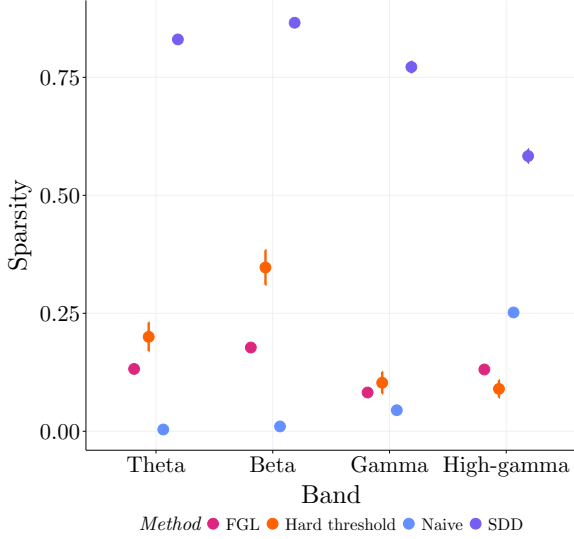


Figure 5: EEG Across Session Analysis. Vertical lines indicate SE.

ilar setup to Yazdan-Shahmorad et al. (2016). Specifically, a total of 32 experimental and 4 control sessions were simulated. Using the same session types as Bloch et al. (2022), of the 32 experimental sessions, 23 were generated using a simulated 10ms stimulation delay and 9 were generated with a 100ms delay. Each experimental session consisted of five 10-minute stimulation blocks (stimulation state) with a 5-minute recording block (resting state) before and after. Thus, each experimental session alternated between resting state and stimulation state blocks consisting of 6 total resting state blocks and 5 total stimulation state blocks. We simulated data using a sampling frequency of 500Hz.

Data in all sessions was simulated using a VAR(1) process with i.i.d standard normal errors. It is worth noting that due to the autoregressive nature of the process, autocovariance matrices between variables will not be diagonal in general despite using a diagonal covariance matrix for the errors (Lütkepohl, 2005). Since the experiments in Yazdan-Shahmorad et al. (2016) included a varying number of good electrodes out of the 96 total electrodes, we randomly selected between $p = 56$ and $p = 94$ good electrodes for each session. Coefficients in a base transition matrix ($A_1 \in \mathbb{R}^{p \times p}$) were generated by randomly sampling from a standard normal distribution. To obtain a stable time series, the matrix A_1 was modified to have maximum eigenvalue < 1 . Similar to Yazdan-Shahmorad et al. (2016), two electrodes in each session were randomly chosen to be stimulated. The effect of stimulation was modeled by

changing the coefficient in A_1 between stimulation electrodes which is denoted as A_{stim} . For control sessions, there was no stimulation so $A_1 = A_{\text{stim}}$. The effect of a 10ms and 100ms stimulation delay was simulated by changing the coefficient between stim sites to be $2 \max(|A_1|)$ and $0.5 \max(|A_1|)$, respectively. In this way, we simulate a weaker effect of the 100ms stimulation delay (Bloch et al., 2022). It is worth noting that, in contrast to the simulations in Section 4, the transition matrices are dense with no block structure. Thus, it is not immediately clear how the true spectral density changes as the transition matrices change.

During stimulation blocks, data were simulated using the corresponding A_{stim} matrix. To simulate a lasting, but weakened effect of stimulation on the following resting state, resting state blocks were simulated using a progressively increasing effect of stimulation. Specifically, resting state block 1 uses A_1 since no stimulation has occurred. The effect of stimulation for resting state blocks 2 through 6 linearly increases from ≈ 0.05 to ≈ 0.6 times the stimulation effect. For example, for a 10ms delay session, resting state block 2 was simulated using a VAR(1) process with transition matrix equal to A_1 where the coefficient between stimulation sites was replaced with $\approx 0.05 * 2 \max(|A_1|)$.

Using our simulated data, we study how stimulation changes the connectivity between stimulation sites. However, instead of using a naïve estimator of the connectivity changes as in Bloch et al. (2022), we use SDD. Using SDD, we estimate the change in inverse spectral densities from resting state to the following resting state (RS to RS) and from resting state to the following stimulation state (RS to SS). For each session there are 5 RS to RS differences (RS1 to RS2, RS2 to RS3, ..., RS5 to RS6) and 5 RS to SS difference (RS1 to SS1, ..., RS5 to SS5). Note that an estimated SDD coefficient of zero indicates no change while non-zero coefficients indicate a change. This could be either a change in magnitude of the connection, a new connection, or a lost connection. The frequencies we study in each band are the same as in Section 5. Results are averaged over all sessions, frequencies within each band, delay (control, 10ms, or 100ms) and RS to RS or SS to SS.

To study how stimulation changes the edges between stimulation sites, we compute the probability of edge change between the stimulation sites for each band, delay, and RS to RS or RS to SS. The RS to RS and RS to SS results are displayed in Figures 6 and 7 respectively. In the RS to SS analysis, stimulation with a 10ms or 100ms delay is more likely to induce a change in connectivity between the stimulation sites across all frequency bands compared to control sessions. After stimulation ends (RS to RS analysis), this connectiv-

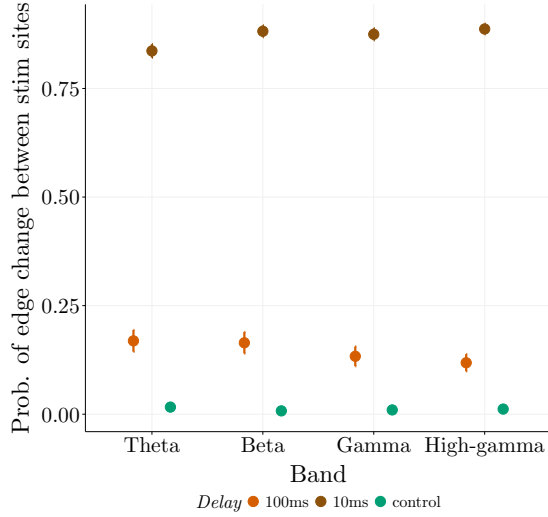


Figure 6: RS to RS analysis. Probability of Change in Edges Between Stimulation Sites by Frequency Band and Delay. Vertical bars indicate standard errors.

ity change is attenuated, but still persists. This is in line with our simulation framework and prior work which showed a change in functional connectivity between stimulation sites in the stimulation state which persisted into the following resting state (Bloch et al., 2022).

7 DISCUSSION

The SDD estimator was compared to the joint graphical lasso with a fusion penalty (FGL, Danaher et al., 2014), the Naïve method of estimating sparse inverse spectral densities and taking their difference, and directly hard thresholding the difference. The four methods were compared across three different simulation settings, where the true difference in inverse spectral densities was sparse. Comparison metrics included precision, recall, accuracy, and relative root mean square error (RRMSE). When compared to the other methods, SDD performed better in accurately identifying edges and non-edges and the SDD estimates had a lower relative root mean square error. Using EEG data we further validated SDD by showing it estimated sparser differences between networks that were closer in time, in line with expected behavior.

While consistency of SDD is established in Theorem 1, this result does not quantify the degree of uncertainty around these estimates. A particularly interesting future direction is developing methods for uncertainty quantification, such as confidence intervals and hy-

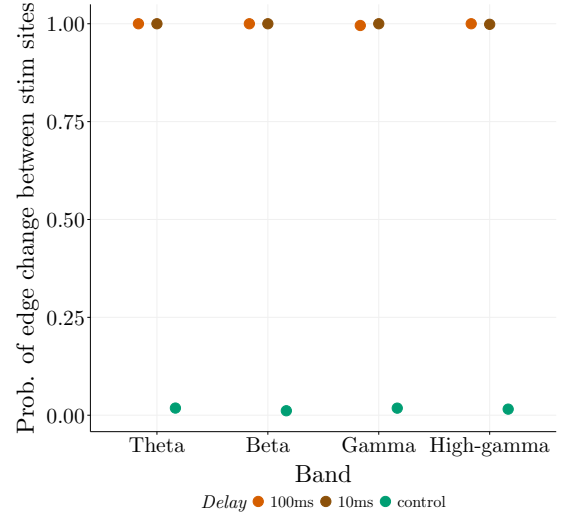


Figure 7: RS to SS analysis. Probability of Change in Edges Between Stimulation Sites by Frequency Band and Delay. Vertical bars indicate standard errors.

pothesis testing. These may be important when the relationship between two brain regions or nodes in the graph is of primary interest or when the sample size is small and noise may be large relative to the signal.

8 CONCLUSION

We proposed a direct estimate of the differences in inverse spectral densities between two conditions, termed the Spectral D-trace Difference (SDD) estimator. By using the direct difference estimator, we only need to assume sparsity in the difference of inverse spectral densities. Compared to the usual assumptions of sparsity of each inverse spectral density, the sparsity of the difference is more realistic if, for example, the inverse spectral densities do not change much between conditions. This is indeed what we expect in many biological settings, especially neurodegenerative disorders that are associated with changes in brain connectivity. Convergence rates of our estimator to the true difference were derived using only an assumption on decaying dependence within the time series.

Acknowledgements

This work was partially supported by U.S. NIH grants R01-MH125429 and R01-GM133848.

References

Onureena Banerjee, Laurent El Ghaoui, and Alexandre 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.
- Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR, 2018.
- Julien Bloch, Alexander Greaves-Tunnell, Eric Shear-Brown, Zaid Harchaoui, Ali Shojaie, and Azadeh Yazdan-Shahmorad. Network structure mediates functional reorganization induced by optogenetic stimulation of non-human primate sensorimotor cortex. *Isience*, 25(5), 2022.
- Hilmar Böhm and Rainer von Sachs. Shrinkage estimation in the frequency domain of multivariate time series. *Journal of Multivariate Analysis*, 100(5):913–935, 2009.
- Peter J Brockwell and Richard A Davis. *Time series: theory and methods*. Springer science & business media, 1991.
- Rainer Dahlhaus. Graphical interaction models for multivariate time series. *Metrika*, 51:157–172, 2000.
- Patrick Danaher, Pei Wang, and Daniela M Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(2):373–397, 2014.
- Navonil Deb, Amy Kuceyeski, and Sumanta Basu. Regularized estimation of sparse spectral precision matrices. *arXiv preprint arXiv:2401.11128*, 2024.
- Tom Dupré la Tour, Thomas Moreau, Mainak Jas, and Alexandre Gramfort. Multivariate convolutional sparse coding for electromagnetic brain signals. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- Mark Fiecas, Chenlei Leng, Weidong Liu, and Yi Yu. Spectral analysis of high-dimensional time series. 2019.
- Rina Foygel and Mathias Drton. Extended bayesian information criteria for gaussian graphical models. *Advances in neural information processing systems*, 23, 2010.
- Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.
- Karl J Friston. Functional and effective connectivity: a review. *Brain connectivity*, 1(1):13–36, 2011.
- Théo Gnassounou, Rémi Flamary, and Alexandre Gramfort. Convolution monge mapping normalization for learning on sleep data. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 10457–10476. Curran Associates, Inc., 2023.
- Christoffer Hatlestad-Hall, Trine Waage Rygvold, and Stein Andersson. "srm resting-state eeg", 2022.
- SR James, HA Knox, RE Abbott, and EJ Screaton. Improved moving window cross-spectral analysis for resolving large temporal seismic velocity changes in permafrost. *Geophysical Research Letters*, 44(9):4018–4026, 2017.
- Lucas C Laurindo, Leo Siqueira, Arthur J Mariano, and Ben P Kirtman. Cross-spectral analysis of the sst/10-m wind speed coupling resolved by satellite products and climate model simulations. *Climate dynamics*, 52(9):5071–5098, 2019.
- Weidong Liu and Wei Biao Wu. Asymptotics of spectral density estimates. *Econometric Theory*, 26(4):1218–1245, 2010.
- Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.
- Christopher J Markiewicz, Krzysztof J Gorgolewski, Franklin Feingold, Ross Blair, Yaroslav O Halchenko, Eric Miller, Nell Hardcastle, Joe Wexler, Oscar Esteban, Mathias Goncalves, et al. The openneuro resource for sharing of neuroscience data. *Elife*, 10:e71774, 2021.
- Sahand N Negahban, Pradeep Ravikumar, Martin J Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. 2012.
- Sou Nobukawa, Mitsuru Kikuchi, and Tetsuya Takahashi. Changes in functional connectivity dynamics with aging: a dynamical phase synchronization approach. *Neuroimage*, 188:357–368, 2019.
- Hugo Richard, Luigi Gresele, Aapo Hyvarinen, Bertrand Thirion, Alexandre Gramfort, and Pierre Ablin. Modeling shared responses in neuroimaging studies through multiview ica. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19149–19162. Curran Associates, Inc., 2020.
- Xiaofeng Shao and Wei Biao Wu. Asymptotic spectral theory for nonlinear time series. 2007.
- Ali Shojaie. Differential network analysis: A statistical perspective. *Wiley Interdisciplinary Reviews: Computational Statistics*, 13(2):e1508, 2021.

- Yiming Sun, Yige Li, Amy Kuceyeski, and Sumanta Basu. Large spectral density matrix estimation by thresholding. *arXiv preprint arXiv:1812.00532*, 2018.
- Mátyás A Sustik and Ben Calderhead. Glassofast: an efficient glasso implementation. *UTCS Technical Report TR-12-29 2012*, 2012.
- Katherine Tsai, Oluwasanmi Koyejo, and Mladen Kolar. Joint gaussian graphical model estimation: A survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, 14(6):e1582, 2022.
- Mauro Ursino, Giulia Ricci, and Elisa Magosso. Transfer entropy as a measure of brain connectivity: A critical analysis with the help of neural mass models. *Frontiers in computational neuroscience*, 14:45, 2020.
- Yue Wang, Jing Ma, and Ali Shojaie. Direct estimation of differential granger causality between two high-dimensional time series. *arXiv preprint arXiv:2109.07609*, 2021.
- Wei Biao Wu. Nonlinear system theory: Another look at dependence. *Proceedings of the National Academy of Sciences*, 102(40):14150–14154, 2005.
- Wei Biao Wu and Paolo Zaffaroni. Asymptotic theory for spectral density estimates of general multivariate time series. *Econometric Theory*, 34(1):1–22, 2018.
- Azadeh Yazdan-Shahmorad, Camilo Diaz-Botia, Timothy L Hanson, Viktor Kharazia, Peter Ledochowitsch, Michel M Maharbiz, and Philip N Sabes. A large-scale interface for optogenetic stimulation and recording in nonhuman primates. *Neuron*, 89(5):927–939, 2016.
- Azadeh Yazdan-Shahmorad, Daniel B Silversmith, Viktor Kharazia, and Philip N Sabes. Targeted cortical reorganization using optogenetics in non-human primates. *Elife*, 7:e31034, 2018.
- Huili Yuan, Ruibin Xi, Chong Chen, and Minghua Deng. Differential network analysis via lasso penalized d-trace loss. *Biometrika*, 104(4):755–770, 2017.
- Andrew Zalesky, Alex Fornito, Luca Cocchi, Leonardo L Gollo, and Michael Breakspear. Time-resolved resting-state brain networks. *Proceedings of the National Academy of Sciences*, 111(28):10341–10346, 2014.
- Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine learning research*, 7 (Nov):2541–2563, 2006.

Checklist

1. For all models and algorithms presented, check if you include:

- (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes/No/Not Applicable] Not Applicable. No new models released.
- (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes/No/Not Applicable] Not Applicable. No new models released.
- (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable] Not Applicable. No new models released.

2. For any theoretical claim, check if you include:

- (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable] Yes, see Section 3 as well as Appendix C and Appendix B.
- (b) Complete proofs of all theoretical results. [Yes/No/Not Applicable] Yes, see Appendix C.
- (c) Clear explanations of any assumptions. [Yes/No/Not Applicable] Yes, see Section 3 and Appendix B.

3. For all figures and tables that present empirical results, check if you include:

- (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable] Yes, all code to reproduce experimental and application results are provided in the supplemental material.
- (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/Not Applicable] Yes, training details are contained in the paper and in the supplemental material.
- (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes/No/Not Applicable] Yes, these are included in the main paper as well as the supplemental material.
- (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/Not Applicable] Yes computing infrastructure is discussed in Section 4.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

- (a) Citations of the creator If your work uses existing assets. [Yes/No/Not Applicable] Yes, existing assets are cited in the paper as well as in Appendix D.
 - (b) The license information of the assets, if applicable. [Yes/No/Not Applicable] Yes, licenses are included in the paper as well as in Appendix D.
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/Not Applicable] Yes, new assets are available in the supplemental material.
 - (d) Information about consent from data providers/curators. [Yes/No/Not Applicable] Not Applicable.
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/Not Applicable] Not Applicable.
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
- (a) The full text of instructions given to participants and screenshots. [Yes/No/Not Applicable] Not Applicable.
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/Not Applicable] Not Applicable.
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/Not Applicable] Not Applicable.

Supplementary Materials

Contents

A ADDITIONAL DETAILS FOR METHODS	12
A.1 D-trace Loss	12
B FUNCTIONAL DEPENDENCE FRAMEWORK	13
C THEORETICAL RESULTS	13
D SIMULATION SPECIFICS AND RESULTS	17

A ADDITIONAL DETAILS FOR METHODS

A.1 D-trace Loss

To estimate Δ , we use the ℓ_1 penalized D-trace loss function from Yuan et al. (2017):

$$L_D(\Delta, \Sigma_2, \Sigma_1) = \frac{1}{4} (\langle \Sigma_2 \Delta, \Delta \Sigma_1 \rangle + \langle \Sigma_1 \Delta, \Delta \Sigma_2 \rangle) - \langle \Delta, \Sigma_2 - \Sigma_1 \rangle$$

where $\langle A, B \rangle = \text{Tr}(AB^T)$. Similar to Yuan et al. (2017), we can take the derivative with respect to Δ to see that $\Sigma_1^{-1} - \Sigma_2^{-1}$ minimizes the D-trace loss. More specifically, let

$$\frac{\partial L_D}{\partial \Delta} = (\Sigma_2 \Delta \Sigma_1 + \Sigma_1 \Delta \Sigma_2) / 2 - (\Sigma_2 - \Sigma_1) . \quad (2)$$

Evaluating $\frac{\partial L_D}{\partial \Delta}$ at $\Delta = \Sigma_1^{-1} - \Sigma_2^{-1}$ yields 0, establishing that $\Delta = \Sigma_1^{-1} - \Sigma_2^{-1}$ minimizes L_D . Furthermore, the Hessian of L_D with respect to Δ is $\frac{\partial^2 L_D}{\partial \Delta^2} = (\Sigma_1 \otimes \Sigma_2 + \Sigma_2 \otimes \Sigma_1) / 2$ where \otimes is the kronecker product. Assuming both Σ_1 and Σ_2 are positive definite, L_D is convex in Δ and $\Delta = \Sigma_1^{-1} - \Sigma_2^{-1}$ is the unique minimizer.

In practice, the population quantites Σ_1 and Σ_2 are not available. Instead, we use the estimates $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$. We can additionally incorporate an ℓ_1 penalty to estimate a sparse Δ which gives us our estimator in (1).

B FUNCTIONAL DEPENDENCE FRAMEWORK

Let ϵ_0^* and $\{\epsilon_t\}_{t \in \mathbb{Z}}$ be i.i.d. vectors in \mathbb{R}^b and $\mathcal{F}_t = (\dots, \epsilon_{t-1}, \epsilon_t)$. We further let $\mathbf{x}_{l,t} = (x_{l,1t}, \dots, x_{l,pt})^T$ be a p -dimensional process in condition l where

$$x_{l,jt} = R_{l,j}(\mathcal{F}_t).$$

Note that l indexes the condition, in this case 1 or 2, and j indexes the variable, $1, \dots, p$. To measure the dependence of $\mathbf{x}_{l,t}$ on ϵ_0 we can replace ϵ_0 in \mathcal{F}_t with an i.i.d. copy ϵ_0^* . This gives $\mathcal{F}'_t = (\dots, \epsilon_{-1}, \epsilon_0^*, \epsilon_1, \dots, \epsilon_t)$ and

$$x'_{l,jt} = R_{l,j}(\mathcal{F}'_t).$$

The dependence on ϵ_0 can then be measured by

$$\theta_{l,jt} = \left(\mathbb{E} |x_{l,jt} - x'_{l,jt}|^2 \right)^{1/2}$$

where $\mathbb{E} |x|^2$ is the expected value of $|x|^2$. Controlling this dependence measure is essential for establishing the convergence rates of \hat{f}_l . The needed conditions on $\theta_{l,jt}$ are laid out in Assumption 1.

Assumption 1. Assume for some constant $0 < \rho < 1$,

$$\max_{l=\{1,2\}} \max_{j=1,\dots,p} \theta_{l,jt} = O(\rho^t),$$

and for some constant $\kappa > 0$, $C_0 > 0$,

$$\max_{l=\{1,2\}} \max_{j=1,\dots,p} \mathbb{E}(\exp(\kappa |\mathbf{x}_{l,j0}|)) \leq C_0.$$

Assumption 1 is essentially a restatement of Assumption 1 in Fiecas et al. (2019), except that we slightly generalize it to account for the fact that we have data from two separate conditions. This condition requires that the dependence on ϵ_0 decreases geometrically as time increases. A similar condition is used in Wu and Zaffaroni (2018) and Liu and Wu (2010) and essentially requires that the dependence in the data on prior time points cannot be too strong. As noted in Shao and Wu (2007) and examples 1 and 2 of Liu and Wu (2010), many processes satisfy this constraint, including ARMA, ARMA-ARCH, and ARMA-GARCH processes as well as other nonlinear autoregressive processes.

Assumption 1 is satisfied if for example Assumption 1 from Fiecas et al. (2019) is satisfied for both conditions. For example if $\{\rho_1, \kappa_1, C_{0,1}\}$ and $\{\rho_2, \kappa_2, C_{0,2}\}$ are the parameters that satisfy Assumption 1 of Fiecas et al. (2019) for the data in conditions 1 and 2 respectively, then $\{\max(\rho_1, \rho_2), \min(\kappa_1, \kappa_2), \max(C_{0,1}, C_{0,2})\}$ satisfy our Assumption 1. We are now ready to state the consistency of our estimator, $\hat{\Delta}$.

C THEORETICAL RESULTS

To prove Theorem 1 we will use the framework of Negahban et al. (2012), specifically Corollary 1. This requires establishing that the D-trace loss satisfies the Restricted Strong Convexity (RSC) condition and that the regularizing penalty is decomposable. An essential ingredient to all these results is a concentration bound on \hat{f}_l . We begin by establishing this bound.

Lemma 1 (Concentration inequality on \hat{f}_l). *Under Assumption 1, for any $H > 0$, and for an optimal bandwidth choice of $M_{n_l}^* = O(n_l^{2/3})$, there exists a constant $C_1 > 0$ that depends only on κ and C_0 such that*

$$\mathbb{P} \left(\sup_{\{\lambda_j = 2\pi j/n_l, \lfloor (n_l-1)/2 \rfloor \leq j \leq \lfloor n_l/2 \rfloor\}} \|\hat{f}_l(\lambda_j) - f_l(\lambda_j)\|_\infty > C_1 8^{2/3} 2n_l^{-\frac{1}{3}} \right) \leq C_2 p^2 n_l^{-H}, \quad (3)$$

where $l \in \{1, 2\}$ indicates condition and the constant $C_2 > 0$ is constant in n_l but depends on C_1 and H .

Proof. Under Assumption 1 we have that Theorem 3.1 from (Fiecas et al., 2019) holds so that for any $\delta > 0$ and $H > 0$ and for $C_1 > 0$ depending only on κ and C_0

$$\mathbb{P} \left(\sup_{\{\lambda_j = 2\pi j/n_l, \lfloor (n_l-1)/2 \rfloor \leq j \leq \lfloor n_l/2 \rfloor\}} \|\hat{f}_l(\lambda_j) - f_l(\lambda_j)\|_\infty > \frac{C_1 M_{n_l}}{n_l} + \frac{8n_l^\delta}{M_{n_l}^{1/2+\delta}} \right) \leq C_2 p^2 n_l^{-H},$$

where l indexes condition 1 or 2 and C_2 is constant in n_l but depends on C_1 and H . Note that this same inequality holds for both conditions $l = 1, 2$ because Assumption 1 holds for both conditions. We allow n_l to differ between conditions for full generality. We can then select M_{n_l} to minimize the deviations. That is we select M_{n_l} to minimize $\frac{C_1 M_{n_l}}{n_l} + \frac{8n_l^\delta}{M_{n_l}^{1/2+\delta}}$. The algebra is omitted but setting $\frac{\partial}{\partial M_{n_l}} \left[\frac{C_1 M_{n_l}}{n_l} + \frac{8n_l^\delta}{M_{n_l}^{1/2+\delta}} \right] = 0$ and solving we get $M_{n_l}^* = \left(\frac{8}{C_1} \left(\frac{1}{2} + \delta \right) \right)^{2/(3+2\delta)} n_l^{\frac{2+2\delta}{3+2\delta}}$. It can be shown that the second derivative is positive on $\delta > 0$ so this is indeed a global minimum (on $\delta > 0$). Plugging this back into the equation yields

$$\left. \frac{C_1 M_{n_l}}{n_l} + \frac{8n_l^\delta}{M_{n_l}^{1/2+\delta}} \right|_{M_{n_l}=M_{n_l}^*} = C_1^{\frac{1+2\delta}{3+2\delta}} 8^{\frac{2}{3+2\delta}} \left(\left(\frac{1}{2} + \delta \right)^{\frac{2}{3+2\delta}} + \left(\frac{1}{2} + \delta \right)^{-\frac{1+2\delta}{3+2\delta}} \right) n_l^{-\frac{1}{3+2\delta}}.$$

One can further show that $\left(\left(\frac{1}{2} + \delta \right)^{\frac{2}{3+2\delta}} + \left(\frac{1}{2} + \delta \right)^{-\frac{1+2\delta}{3+2\delta}} \right)$ is maximized at $\delta = 1/2$ with a maximum value of 2 (algebra omitted). Furthermore $C_1^{\frac{1+2\delta}{3+2\delta}} \leq C_1$ and $8^{\frac{2}{3+2\delta}} \leq 8^{2/3}$ for all $\delta > 0$. Therefore we have that for all $\delta > 0$,

$$\frac{C_1 M_{n_l}^*}{n_l} + \frac{8n_l^\delta}{(M_{n_l}^*)^{1/2+\delta}} \leq C_1 8^{2/3} 2 n_l^{-\frac{1}{3+2\delta}}.$$

Letting $\delta \rightarrow 0$ the RHS of our concentration bound becomes $C_1 8^{2/3} 2 n_l^{-\frac{1}{3}}$ establishing the lemma. \square

Remark 1. Since $\|\hat{\Sigma}_l(\lambda_j) - \Sigma_l(\lambda_j)\|_\infty \leq \|\hat{f}_l(\lambda_j) - f_l(\lambda_j)\|_\infty$, it follows that Lemma 1 also holds for $\|\hat{\Sigma}_l(\lambda_j) - \Sigma_l(\lambda_j)\|_\infty$.

To establish RSC of the D-trace loss function we will also need a concentration inequality on the second derivative of the D-trace loss which is stated in Lemma 2.

Lemma 2 (Concentration inequality on second derivative of D-trace loss). *Under the conditions from Lemma 1 we have that for $\min(n_1, n_2) \geq 8^{9/2} C_1^3 / C_f^3$, with probability $> 1 - 2C_2 p^2 \min(n_1, n_2)^{-H}$,*

$$\left\| 0.5 \left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1 \right) - 0.5 \left(\Sigma_1 \otimes \Sigma_2 + \Sigma_2 \otimes \Sigma_1 \right) \right\|_\infty \leq \frac{C_f C_1 8^{5/2}}{\min(n_1, n_2)^{1/3}},$$

where $C_f = \max(\|f_1\|_\infty, \|f_2\|_\infty)$ and C_1 is the same as in Lemma 1.

Proof. Begin by noting that both

$$\|\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 - \Sigma_2 \otimes \Sigma_1\|_\infty, \|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\|_\infty \leq \|\Sigma_1\|_\infty \|\hat{\Sigma}_2 - \Sigma_2\|_\infty + \|\Sigma_2\|_\infty \|\hat{\Sigma}_1 - \Sigma_1\|_\infty + \|\hat{\Sigma}_1 - \Sigma_1\|_\infty \|\hat{\Sigma}_2 - \Sigma_2\|_\infty.$$

Also we have that $\|\Sigma_l\|_\infty \leq \|f_l\|_\infty$. Using Lemma 1, we get that with probability $> 1 - 2C_2 p^2 \min(n_1, n_2)^{-H}$,

$$\begin{aligned} \|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\|_\infty &\leq C_f \left(C_1 8^{3/2} 2 n_1^{-1/3} + C_1 8^{3/2} 2 n_2^{-1/3} \right) + C_1^2 8^{6/2} 4 n_1^{-1/3} n_2^{-1/3} \\ &\leq C_f C_1 8^{3/2} 4 \min(n_1, n_2)^{-1/3} + C_1^2 8^{6/2} 4 \min(n_1, n_2)^{-2/3}. \end{aligned}$$

We can see that for $\min(n_1, n_2) \geq 8^{9/2} C_1^3 / C_f^3$ the 2nd term on the RHS is smaller than the 1st term on the RHS so we can combine them to simplify the bound to be 2 times the first term on the RHS or

$$\|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\|_\infty \leq \frac{C_f C_1 8^{5/2}}{\min(n_1, n_2)^{1/3}}.$$

This also holds for $\|\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 - \Sigma_2 \otimes \Sigma_1\|_\infty$. Defining

$$\text{Event } \mathcal{A} := \left\{ \|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\|_\infty \leq \frac{C_f C_1 8^{5/2}}{\min(n_1, n_2)^{1/3}} \right\} \cap \left\{ \|\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 - \Sigma_2 \otimes \Sigma_1\|_\infty \leq \frac{C_f C_1 8^{5/2}}{\min(n_1, n_2)^{1/3}} \right\}$$

then from the above results, for $\min(n_1, n_2) \geq 8^{9/2} C_1^3 / C_f^3$, Event \mathcal{A} occurs with probability $> 1 - 2C_2 p^2 \min(n_1, n_2)^{-H}$ and we have

$$\left\| 0.5 \left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1 \right) - 0.5 \left(\Sigma_1 \otimes \Sigma_2 + \Sigma_2 \otimes \Sigma_1 \right) \right\|_\infty \leq \frac{C_f C_1 8^{5/2}}{\min(n_1, n_2)^{1/3}}. \quad (4)$$

□

Using Lemma 1 and Lemma 2 we can establish that the D-trace loss satisfies the restricted strong convexity condition. This is formalized in Lemma 3

Lemma 3 (RSC of D-trace loss). *Under the conditions of Lemma 1, for $\min(n_1, n_2) \geq \max \left(8^{9/2} C_1^3 / C_f^3, \left(\frac{32 s_\Delta C_f C_1 8^{5/2}}{\lambda_{\min}(f_1) \lambda_{\min}(f_2)} \right)^3 \right)$ with probability $> 1 - 2C_2 p^2 \min(n_1, n_2)^{-H}$ RSC holds for the D-trace loss with $\kappa_L = \lambda_{\min}(f_1) \lambda_{\min}(f_2) / 2$. We use $\lambda_{\min}(A)$ to denote the minimum eigenvalue of a matrix A and s_Δ to denote the number of non-zero entries of Δ . The constant C_f is defined in Lemma 2 while C_1, C_2 are defined in Lemma 1.*

Proof. To show that the D-trace loss function satisfies the RSC condition, we must show that $\mathbf{m}^T \left(\nabla^2 L(\Delta, \hat{\Sigma}_1, \hat{\Sigma}_2) \right) \mathbf{m} > \kappa_L \|\mathbf{m}\|_2^2$ for all $\mathbf{m} \in \mathcal{C}(S_\Delta)$ where $\mathcal{C}(S_\Delta) = \left\{ \mathbf{m} \in \mathbb{R}^{4p^2} \mid \|\mathbf{m}_{S_\Delta^c}\|_1 \leq 3 \|\mathbf{m}_{S_\Delta}\|_1 \right\}$ for some κ_L . For our loss, we let S_Δ be the support of Δ , the true difference. The notation \mathbf{m}_{S_Δ} selects the entries of \mathbf{m} that correspond to the non-zero indices of Δ while $\mathbf{m}_{S_\Delta^c}$ selects the entries corresponding to the zero indices of Δ .

First note that for $\mathbf{m} \in \mathcal{C}(S_\Delta)$, $\|\mathbf{m}\|_1 \leq 4 \|\mathbf{m}_{S_\Delta}\|_1 \leq 4\sqrt{s_\Delta} \|\mathbf{m}_{S_\Delta}\|_2$. Recall $\nabla^2 L(\Delta, \hat{\Sigma}_1, \hat{\Sigma}_2) = 0.5(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1)$. Showing RSC proceeds the same as in Wang et al. (2021).

$$\begin{aligned} \mathbf{m}^T (\hat{\Sigma}_1 \otimes \hat{\Sigma}_2) \mathbf{m} &\geq \mathbf{m}^T (\Sigma_1 \otimes \Sigma_2) \mathbf{m} + \mathbf{m}^T \left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2 \right) \mathbf{m} \\ &\geq \lambda_{\min}(\Sigma_1) \lambda_{\min}(\Sigma_2) \|\mathbf{m}\|_2^2 - \left| \mathbf{m}^T \left(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2 \right) \mathbf{m} \right| \\ &\geq \lambda_{\min}(\Sigma_1) \lambda_{\min}(\Sigma_2) \|\mathbf{m}\|_2^2 - \|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\|_\infty \|\mathbf{m}\|_1^2 \\ &\geq \lambda_{\min}(\Sigma_1) \lambda_{\min}(\Sigma_2) \|\mathbf{m}\|_2^2 - 16 s_\Delta \|\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 - \Sigma_1 \otimes \Sigma_2\|_\infty \|\mathbf{m}\|_2^2. \end{aligned} \quad (5)$$

The last inequality follows from the fact that $\|\mathbf{m}\|_1 = \|\mathbf{m}_{S_\Delta^c}\|_1 + \|\mathbf{m}_{S_\Delta}\|_1 \leq 4 \|\mathbf{m}_{S_\Delta}\|_1 \leq 4\sqrt{s_\Delta} \|\mathbf{m}_{S_\Delta}\|_2 \leq 4\sqrt{s_\Delta} \|\mathbf{m}\|_2$. Similarly

$$\mathbf{m}^T (\hat{\Sigma}_2 \otimes \hat{\Sigma}_1) \mathbf{m} \geq \lambda_{\min}(\Sigma_1) \lambda_{\min}(\Sigma_2) \|\mathbf{m}\|_2^2 - 16 s_\Delta \|\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 - \Sigma_2 \otimes \Sigma_1\|_\infty \|\mathbf{m}\|_2^2.$$

Using Lemma 2 and the fact that $\lambda_{\min}(\Sigma_l) = \lambda_{\min}(f_l)$ we have that, for $\min(n_1, n_2) \geq 8^{9/2} C_1^3 / C_f^3$ and with probability $> 1 - 2C_2 p^2 \min(n_1, n_2)^{-H}$,

$$\mathbf{m}^T 0.5(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1) \mathbf{m} \geq \lambda_{\min}(f_1) \lambda_{\min}(f_2) \|\mathbf{m}\|_2^2 - 16 s_\Delta \frac{C_f C_1 8^{5/2}}{\min(n_1, n_2)^{1/3}} \|\mathbf{m}\|_2^2$$

So for $\min(n_1, n_2) \geq \left(\frac{32s_\Delta C_f C_1 8^{5/2}}{\lambda_{\min}(f_1)\lambda_{\min}(f_2)} \right)^3$ then the 2nd term on the RHS is $\leq 0.5^*(1\text{st term on RHS})$. Thus we get

$$\mathbf{m}^T 0.5(\hat{\Sigma}_1 \otimes \hat{\Sigma}_2 + \hat{\Sigma}_2 \otimes \hat{\Sigma}_1) \mathbf{m} \geq \frac{\lambda_{\min}(f_1)\lambda_{\min}(f_2)}{2} \|\mathbf{m}\|_2^2$$

Therefore we conclude that for $\min(n_1, n_2) \geq \max \left(8^{9/2} C_1^3 / C_f^3, \left(\frac{32s_\Delta C_f C_1 8^{5/2}}{\lambda_{\min}(f_1)\lambda_{\min}(f_2)} \right)^3 \right)$ with probability $> 1 - 2C_2 p^2 \min(n_1, n_2)^{-H}$ RSC holds with $\kappa_L = \lambda_{\min}(f_1)\lambda_{\min}(f_2)/2$. \square

We are now ready to prove Theorem 1.

Proof of Theorem 1. We first proceed by establishing the decomposability of our penalty term, $\|\Delta\|_1$. Decomposability is defined in Definition 1 of Negahban et al. (2012). Let $\overline{\mathcal{M}} = \{\theta : \mathbb{R}^{2p \times 2p} | \theta_{ij} \neq 0 \text{ for all } (i, j) \in S_\Delta\}$ and $\overline{\mathcal{M}}^\perp = \{\gamma : \mathbb{R}^{2p \times 2p} | \gamma_{ij} = 0 \text{ for all } (i, j) \in S_\Delta\}$. Then $\|\cdot\|_1$ is decomposable since $\|\theta + \gamma\|_1 = \|\theta\|_1 + \|\gamma\|_1$ for all $\theta \in \overline{\mathcal{M}}$ and $\gamma \in \overline{\mathcal{M}}^\perp$.

Next we discuss a suitable choice of penalty scale τ_{n_1, n_2} . From Theorem 1 in Negahban et al. (2012) we require $\tau_{n_1, n_2} \geq 2\|\nabla L_D(\Delta, \hat{\Sigma}_1, \hat{\Sigma}_2)\|_\infty$. We know that $2\|\nabla L_D(\Delta, \hat{\Sigma}_1, \hat{\Sigma}_2)\|_\infty = 2\|0.5(\hat{\Sigma}_1 \Delta \hat{\Sigma}_2 + \hat{\Sigma}_2 \Delta \hat{\Sigma}_1) - (\hat{\Sigma}_1 - \hat{\Sigma}_2)\|_\infty$.

Let $\Gamma = 0.5(\Sigma_2 \otimes \Sigma_1 + \Sigma_1 \otimes \Sigma_2)$ and $\hat{\Gamma} = 0.5(\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 + \hat{\Sigma}_1 \otimes \hat{\Sigma}_2)$. Then we write

$$\begin{aligned} \|0.5(\hat{\Sigma}_1 \Delta \hat{\Sigma}_2 + \hat{\Sigma}_2 \Delta \hat{\Sigma}_1) - (\hat{\Sigma}_1 - \hat{\Sigma}_2)\|_\infty &= \|\hat{\Gamma} \text{vec}(\Delta) - (\text{vec}(\hat{\Sigma}_1) - \text{vec}(\hat{\Sigma}_2))\|_\infty \\ &= \|(\hat{\Gamma} - \Gamma) \text{vec}(\Delta) - (\text{vec}(\hat{\Sigma}_1) - \text{vec}(\Sigma_1)) - (\text{vec}(\hat{\Sigma}_2) - \text{vec}(\Sigma_2))\|_\infty \\ &\leq \|(\hat{\Gamma} - \Gamma) \text{vec}(\Delta)\|_\infty + \|\hat{\Sigma}_1 - \Sigma_1\|_\infty + \|\hat{\Sigma}_2 - \Sigma_2\|_\infty \\ &\leq \|\hat{\Gamma} - \Gamma\|_\infty \|\Delta\|_1 + \|\hat{\Sigma}_1 - \Sigma_1\|_\infty + \|\hat{\Sigma}_2 - \Sigma_2\|_\infty. \end{aligned}$$

By Lemma 2 we have for $\min(n_1, n_2) \geq \frac{C_1^3}{C_f^3} 8^{9/2}$, with probability $> 1 - 2C_2 p^2 \min(n_1, n_2)^{-H}$,

$$\begin{aligned} \|0.5(\hat{\Sigma}_1 \Delta \hat{\Sigma}_2 + \hat{\Sigma}_2 \Delta \hat{\Sigma}_1) - (\hat{\Sigma}_1 - \hat{\Sigma}_2)\|_\infty &\leq \frac{C_1 C_f 8^{5/2}}{\min(n_1, n_2)^{1/3}} \|\Delta\|_1 + \frac{C_1 8^{3/2} 2}{n_1^{1/3}} + \frac{C_1 8^{3/2} 2}{n_2^{1/3}} \\ &\leq \frac{(C_1 C_f 8^{5/2} \|\Delta\|_1 + 4C_1 8^{3/2})}{\min(n_1, n_2)^{1/3}}. \end{aligned} \quad (6)$$

Therefore if we choose

$$\tau_{n_1, n_2} \geq \frac{2(C_1 C_f 8^{5/2} \|\Delta\|_1 + 4C_1 8^{3/2})}{\min(n_1, n_2)^{1/3}},$$

we will satisfy $\tau_{n_1, n_2} \geq 2\|\nabla L_D(\Delta, \hat{\Sigma}_1, \hat{\Sigma}_2)\|_\infty$ with probability $> 1 - 2C_2 p^2 \min(n_1, n_2)^{-H}$ when $\min(n_1, n_2) \geq \frac{C_1^3}{C_f^3} 8^{9/2}$.

Combining the above with Lemma 3 we get that for $\tau_{n_1, n_2} \geq \frac{2}{\min(n_1, n_2)^{1/3}} (C_1 C_f 8^{5/2} \|\Delta\|_1 + 4C_1 8^{3/2})$, $\min(n_1, n_2) \geq \max \left(8^{9/2} C_1^3 / C_f^3, \left(\frac{32s_\Delta C_f C_1 8^{5/2}}{\lambda_{\min}(f_1)\lambda_{\min}(f_2)} \right)^3 \right)$, and $M_{n, l} = O(n_l^{2/3})$ we have with probability greater than $1 - Cp^2 \min(n_1, n_2)^{-H}$ for any $H > 0$,

$$\|\hat{\Delta} - \Delta\|_F \leq \frac{\sqrt{18s_\Delta \tau_{n_1, n_2}}}{\lambda_{\min}(f_1)\lambda_{\min}(f_2)},$$

where we have defined $C = 2C_2$. Since τ_{n_1, n_2} is of order $\min(n_1, n_2)^{-1/3}$, convergence of our D-trace estimator is also of order $\min(n_1, n_2)^{-1/3}$. □

D SIMULATION SPECIFICS AND RESULTS

We generate data in condition l as

$$\mathbf{x}_{l,t} = A_l \mathbf{x}_{l,t-1} + \boldsymbol{\epsilon}_{l,t}$$

where $\boldsymbol{\epsilon}_{l,t} \sim N_p(\mathbf{0}, I_p)$. From Sun et al. (2018), the spectral density at frequency λ is known to be

$$f_l(\lambda) = \frac{1}{2\pi} (\mathcal{A}_l(e^{-i\lambda}))^{-1} I_p \left((\mathcal{A}_l(e^{-i\lambda}))^{-1} \right)^H,$$

where $\mathcal{A}_l(z) = I_p - A_l z$. When the transition matrix A_l is block diagonal, the spectral density $f_l(\lambda)$ is also block diagonal. Since the inverse of a block diagonal matrix can be computed block by block we can easily generate a sparse difference matrix by enforcing the transition matrix in conditions 1 and 2 to be the same except for a small block. For example, with $p = 54$ if we generate A_1 and A_2 as block diagonal with the same 51×51 block and only differing in the final 3×3 block, then their spectral densities will only differ in this last 3×3 block. When converting to the real space, the expanded $\Sigma_i \in (2p \times 2p)$ and thus the true difference $\Delta = \Sigma_1 - \Sigma_2$ will differ in four 3×3 blocks.

For simulation setting 1, the transition matrix is the same as in (Sun et al., 2018). Specifically, it consists of 18

blocks of dimension 3×3 , where each block is $\begin{bmatrix} 0.5 & 0.9 & 0 \\ 0 & 0.5 & 0.9 \\ 0 & 0 & 0.5 \end{bmatrix}$. In both simulation setting 2 and setting 3, 60% of

the coefficients in the 3×3 block were randomly drawn from either a $\text{Uniform}(-0.5, -0.2)$ or a $\text{Uniform}(0.2, 0.5)$ each with equal probability. In the second and third setting, 40% and 5% respectively of the entries of the larger block were randomly drawn from a $\text{Uniform}(-0.5, -0.2)$ and $\text{Uniform}(0.2, 0.5)$ each with equal probability. This corresponded to 60% and 95% sparsity respectively. The number of non-zero entries in the difference in expanded inverse spectral densities, which we will refer to as edges, varies by frequency but is almost always 22 for Simulation 1, 14 for Simulation 2, and 28 for Simulation 3.

For all settings 100 evenly spaced Fourier frequencies from 0 to $\pi - n^{-1}$ was used. These were used as the spectral density is conjugate symmetric around 0. In the case of $n = 100$ only 50 Fourier frequencies were used as there are only 50 Fourier frequencies from 0 to $\pi - n^{-1}$. Given the availability of results from multiple frequencies, each simulation was run once and the results are averaged across all frequencies. The standard error of each metric was also computed across frequencies and are reported in parentheses.

To solve the SDD estimation problem (1), the alternating direction method of multipliers (ADMM) algorithm from Yuan et al. (2017) was used. Specifically, we used the `L1_dts` function in the `Difdtl` package available on GitHub at SusanYuan/Difdtl (GPL (≥ 2)). To generate the sequence of penalties, $\{\tau_{n_1, n_2}\}$, we used 20 values on a log-linear scale from $0.001 * \tau_{n_1, n_2, \max}$ to $\tau_{n_1, n_2, \max}$ where $\tau_{n_1, n_2, \max}$ represents the minimum value where all entries of $\hat{\Delta}$ are 0. In this case, $\tau_{n_1, n_2, \max} = 2 \max(|\hat{\Sigma}_1 - \hat{\Sigma}_2|)$. The penalty τ_{n_1, n_2}^* was chosen as the τ_{n_1, n_2} that minimizes the eBIC which is computed as

$$\text{eBIC}(\hat{\Delta})_\gamma = \min(n_1, n_2) \left\| \frac{1}{2} \left(\hat{\Sigma}_1 \hat{\Delta} \hat{\Sigma}_2 + \hat{\Sigma}_2 \hat{\Delta} \hat{\Sigma}_1 - \hat{\Sigma}_2 + \hat{\Sigma}_1 \right) \right\|_\infty + \log(\min(n_1, n_2)) |E| + 4\gamma |E| \log(p), \quad (7)$$

where $|E|$ is the number of unique edges in $\hat{\Delta}$. Since $\hat{\Delta}$ represents the difference in expanded spectral densities $|E|$ is the number of non-zero entries in the upper triangular portions, including diagonals, of the submatrices $\hat{\Delta}_{1:p, 1:p}$ and $\hat{\Delta}_{1:p, (p+1):2p}$. For all applications we use $\gamma = 0.5$. For FGL, Naïve, and Hard Thresholding, 20 tuning parameter values were also used. FGL requires two tuning parameter values, λ_1, λ_2 , so a grid of 20 (λ_1, λ_2) pairs was considered. Since the parameter λ_1 controls the sparsity of each inverse matrix while λ_2 controls sparsity of the difference we considered only two values for λ_1 and 10 values for λ_2 . The values for λ_1 were $(0.01\lambda_{1, \max}, 0.1\lambda_{1, \max})$ where $\lambda_{1, \max} = \left\| \hat{\Sigma}_{1, O} + \hat{\Sigma}_{2, O} \right\|_\infty / 2$ and $\hat{\Sigma}_{i, O}$ is the off-diagonals of $\hat{\Sigma}_i$. The

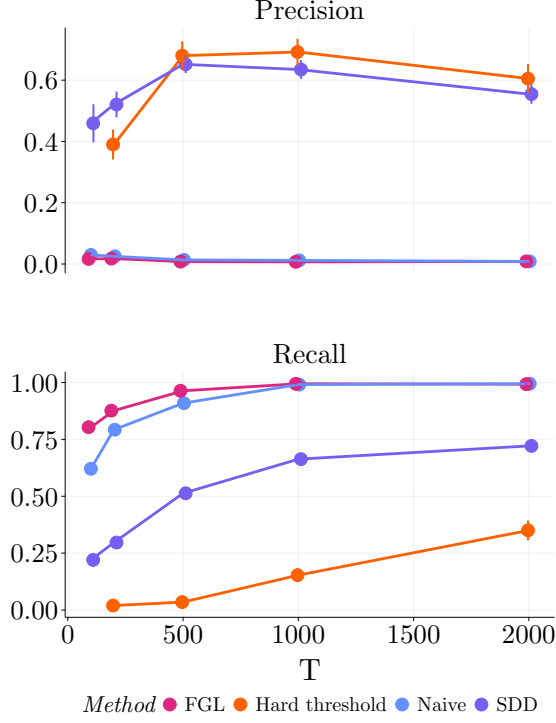


Figure 8: Simulation 1 Precision and Recall. Results are reported as mean (dots) and SE (vertical lines) where the mean and SE are taken across all frequencies for a given sample size n .

λ_2 sequence was generated using 10 values on a log-linear scale from $0.0001 * \lambda_{2,\max}$ to $\lambda_{2,\max}$ where $\lambda_{2,\max} = \max \left(\left\| \hat{\Sigma}_{1,O} \right\|_{\infty} - 0.01\lambda_{1,\max}, \left\| \hat{\Sigma}_{2,O} \right\|_{\infty} - 0.01\lambda_{1,\max} \right)$. For the Naïve method, we first computed $\lambda_{\max,i} = \left\| \hat{\Sigma}_i \right\|_{\infty}$ in each condition and then computed the tuning parameters as 20 evenly-spaced values on a log-linear scale from $0.0001\lambda_{\max,i}$ to $\lambda_{\max,i}$. Similarly, for the Hard Thresholding method, we used 20 evenly-spaced values on a log-linear scale from $\min(|\hat{\Delta}_H|)$ to $\max(|\hat{\Delta}_H|)$ where $\hat{\Delta}_H$ is generated by inverting $\hat{\Sigma}_i$ and forming the difference. Note that for $n = 100$, it is not possible to invert $\hat{\Sigma}_i$.

Let TP, FN, TN, FP denote the true positive, false negative, true negative, and false positive edges identified by either SDD or the Naïve method, respectively. A value greater than 1×10^{-6} in absolute value was considered an edge. The metrics are defined as follows

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} & \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{Accuracy} &= \frac{\text{TP} + \text{TN}}{4p^2} & \text{RRMSE} &= \sqrt{\frac{\sum_{i,j} (\hat{\Delta}_{i,j} - \Delta_{i,j})^2}{\sum_{i,j} (\Delta_{i,j})^2}}, \end{aligned}$$

where $\hat{\Delta}$ represents the difference estimator, either SDD or the Naïve difference, Δ is the true difference. The denominator of the accuracy measure is $4p^2$ as expanding a $p \times p$ spectral density to the real space gives a $2p \times 2p$ matrix which has $4p^2$ entries. We also report the number of average number of true edges and the average number of estimated edges across frequencies.

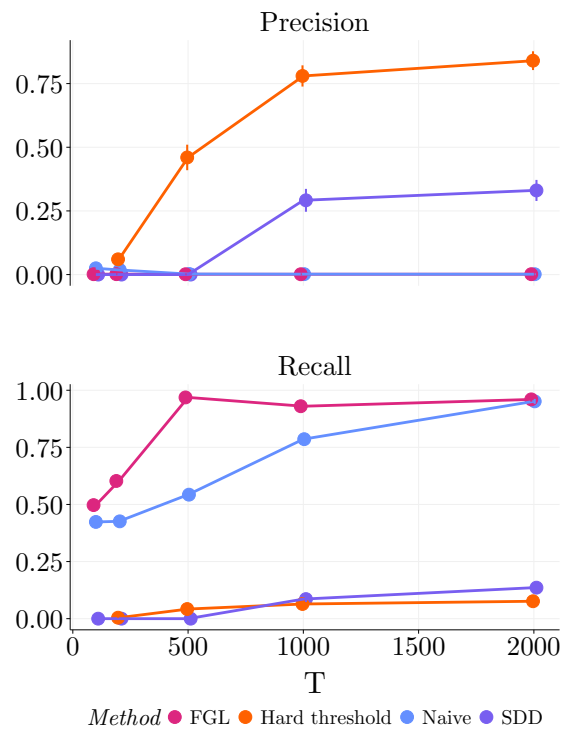


Figure 9: Simulation 2 Precision and Recall. Results are reported as mean (dots) and SE (vertical lines) where the mean and SE are taken across all frequencies for a given sample size n .

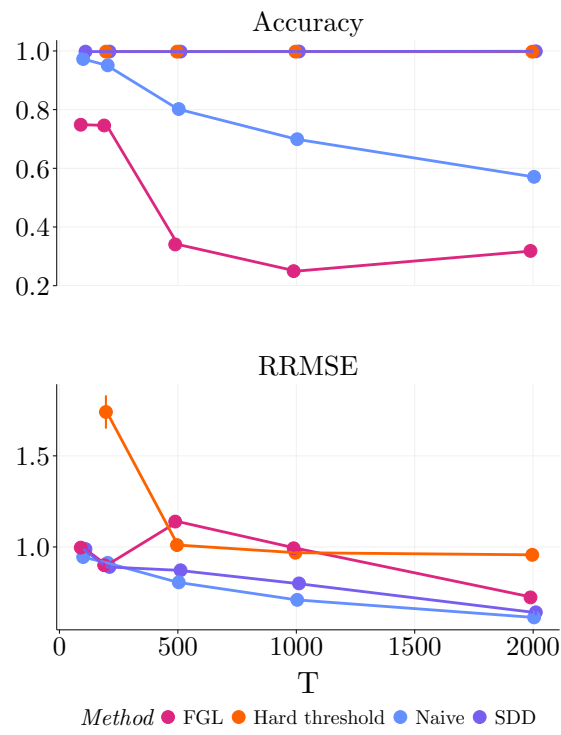


Figure 10: Simulation 3 Accuracy and RRMSE. Results are reported as mean (dots) and SE (vertical lines) where the mean and SE are taken across all frequencies for a given sample size n .

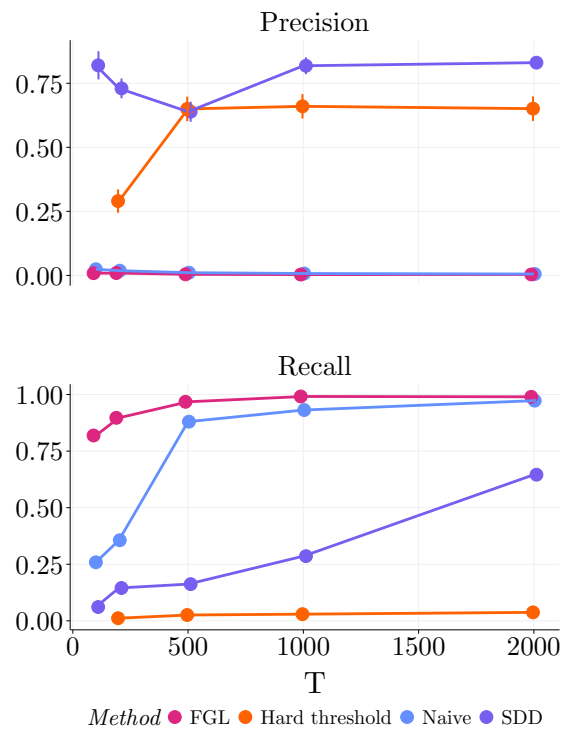


Figure 11: Simulation 3 Precision and Recall. Results are reported as mean (dots) and SE (vertical lines) where the mean and SE are taken across all frequencies for a given sample size n .

Table 1: Simulation 1. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies for a given sample size n .

n	# True edges	SDD				
		# Est edges	Precision	Recall	Accuracy	RRMSE
100	21.6 (0.32)	10.6 (1.04)	0.46 (0.06)	0.22 (0.03)	1.00 (0.00)	0.95 (0.01)
200	21.8 (0.16)	16.7 (1.00)	0.52 (0.04)	0.30 (0.02)	1.00 (0.00)	0.89 (0.01)
500	21.9 (0.08)	21.1 (1.15)	0.65 (0.03)	0.51 (0.02)	1.00 (0.00)	0.78 (0.01)
1000	21.9 (0.08)	31.1 (2.03)	0.63 (0.03)	0.66 (0.02)	1.00 (0.00)	0.61 (0.01)
2000	21.9 (0.08)	48.7 (4.60)	0.55 (0.03)	0.72 (0.02)	1.00 (0.00)	0.53 (0.01)
FGL						
		2073 (194)	0.02 (0.00)	0.80 (0.02)	0.82 (0.02)	0.89 (0.01)
		2444 (137)	0.02 (0.00)	0.88 (0.01)	0.79 (0.01)	0.84 (0.01)
		4973 (326)	0.01 (0.00)	0.96 (0.01)	0.58 (0.03)	0.68 (0.01)
		5962 (324)	0.01 (0.00)	0.99 (0.00)	0.49 (0.03)	0.64 (0.02)
		6308 (327)	0.01 (0.00)	0.99 (0.00)	0.46 (0.03)	0.62 (0.02)
Naïve						
		659 (46.1)	0.03 (0.00)	0.62 (0.02)	0.94 (0.00)	0.95 (0.01)
		1077 (56.9)	0.03 (0.00)	0.79 (0.02)	0.91 (0.00)	0.90 (0.00)
		1775 (72.1)	0.01 (0.00)	0.91 (0.01)	0.85 (0.01)	0.80 (0.01)
		2289 (104)	0.01 (0.00)	0.99 (0.00)	0.81 (0.01)	0.69 (0.01)
		3158 (151)	0.01 (0.00)	0.99 (0.00)	0.73 (0.01)	0.58 (0.01)
Hard threshold						
		-	-	-	-	-
		0.69 (0.05)	0.39 (0.05)	0.02 (0.00)	1.00 (0.00)	1.57 (0.06)
		0.79 (0.06)	0.68 (0.05)	0.03 (0.00)	1.00 (0.00)	1.00 (0.00)
		5.01 (1.11)	0.69 (0.04)	0.15 (0.03)	1.00 (0.00)	0.89 (0.02)
		230 (53.1)	0.60 (0.05)	0.35 (0.04)	0.98 (0.00)	1.04 (0.06)

Table 2: Simulation 2. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies for a given sample size n .

n	# True edges	SDD				
		# Est edges	Precision	Recall	Accuracy	RRMSE
100	13.7 (0.21)	1.52 (0.27)	0.00 (0.00)	0.00 (0.00)	1.00 (0.00)	1.00 (0.00)
200	13.9 (0.11)	0.90 (0.13)	0.00 (0.00)	0.00 (0.00)	1.00 (0.00)	1.00 (0.00)
500	14.0 (0.04)	2.24 (0.35)	0.00 (0.00)	0.00 (0.00)	1.00 (0.00)	1.00 (0.00)
1000	14.0 (0.04)	3.80 (0.47)	0.29 (0.04)	0.09 (0.01)	1.00 (0.00)	0.99 (0.01)
2000	14.0 (0.04)	6.52 (0.83)	0.33 (0.04)	0.14 (0.01)	1.00 (0.00)	0.95 (0.01)
FGL						
		4675 (227)	0.00 (0.00)	0.50 (0.02)	0.60 (0.02)	2.25 (0.16)
		5297 (185)	0.00 (0.00)	0.60 (0.02)	0.55 (0.02)	2.15 (0.12)
		10201 (118)	0.00 (0.00)	0.97 (0.01)	0.13 (0.01)	4.20 (0.07)
		9731 (155)	0.00 (0.00)	0.93 (0.02)	0.17 (0.01)	3.09 (0.06)
		9196 (176)	0.00 (0.00)	0.96 (0.01)	0.21 (0.02)	2.30 (0.05)
Naïve						
		259 (10.9)	0.02 (0.00)	0.42 (0.01)	0.98 (0.00)	1.02 (0.01)
		609 (46.3)	0.02 (0.00)	0.43 (0.00)	0.95 (0.00)	1.01 (0.01)
		3566 (56.3)	0.00 (0.00)	0.54 (0.01)	0.69 (0.00)	1.20 (0.03)
		6487 (58.2)	0.00 (0.00)	0.79 (0.02)	0.44 (0.00)	1.58 (0.04)
		9145 (98.2)	0.00 (0.00)	0.95 (0.01)	0.22 (0.01)	2.29 (0.04)
Hard threshold						
		-	-	-	-	-
		0.72 (0.05)	0.06 (0.02)	0.00 (0.00)	1.00 (0.00)	3.05 (0.18)
		0.93 (0.08)	0.46 (0.05)	0.04 (0.01)	1.00 (0.00)	1.09 (0.02)
		0.93 (0.06)	0.78 (0.04)	0.06 (0.00)	1.00 (0.00)	0.94 (0.01)
		1.08 (0.06)	0.84 (0.04)	0.08 (0.00)	1.00 (0.00)	0.89 (0.01)

Table 3: Simulation 3. Results are reported as Mean (SE) where the mean and SE are taken across all frequencies for a given sample size n .

n	# True edges	SDD				
		# Est edges	Precision	Recall	Accuracy	RRMSE
100	27.4 (0.39)	1.76 (0.12)	0.82 (0.05)	0.06 (0.00)	1.00 (0.00)	0.99 (0.00)
200	27.7 (0.20)	4.80 (0.31)	0.73 (0.04)	0.14 (0.01)	1.00 (0.00)	0.89 (0.01)
500	27.9 (0.12)	6.68 (0.56)	0.64 (0.04)	0.16 (0.01)	1.00 (0.00)	0.87 (0.01)
1000	27.9 (0.12)	8.86 (0.71)	0.82 (0.03)	0.29 (0.02)	1.00 (0.00)	0.80 (0.02)
2000	27.9 (0.12)	22.1 (0.77)	0.83 (0.01)	0.65 (0.02)	1.00 (0.00)	0.64 (0.01)
FGL						
		2950 (157)	0.01 (0.00)	0.82 (0.03)	0.75 (0.01)	1.00 (0.02)
		2984 (97.0)	0.01 (0.00)	0.90 (0.02)	0.75 (0.01)	0.90 (0.01)
		7718 (254)	0.00 (0.00)	0.97 (0.01)	0.34 (0.02)	1.14 (0.03)
		8780 (137)	0.00 (0.00)	0.99 (0.00)	0.25 (0.01)	0.99 (0.02)
		7975 (156)	0.00 (0.00)	0.99 (0.01)	0.32 (0.01)	0.72 (0.01)
Naïve						
		308 (12.7)	0.02 (0.00)	0.26 (0.01)	0.97 (0.00)	0.94 (0.01)
		563 (20.2)	0.02 (0.00)	0.36 (0.01)	0.95 (0.00)	0.91 (0.01)
		2331 (34.6)	0.01 (0.00)	0.88 (0.02)	0.80 (0.00)	0.81 (0.01)
		3532 (49.8)	0.01 (0.00)	0.93 (0.01)	0.70 (0.00)	0.71 (0.01)
		5030 (61.8)	0.01 (0.00)	0.97 (0.01)	0.57 (0.01)	0.61 (0.01)
Hard threshold						
		-	-	-	-	-
		0.58 (0.05)	0.29 (0.05)	0.01 (0.00)	1.00 (0.00)	1.74 (0.09)
		0.72 (0.06)	0.65 (0.05)	0.03 (0.00)	1.00 (0.00)	1.01 (0.00)
		0.81 (0.07)	0.66 (0.05)	0.03 (0.00)	1.00 (0.00)	0.97 (0.00)
		2.67 (1.78)	0.65 (0.05)	0.04 (0.01)	1.00 (0.00)	0.96 (0.00)

Table 4: Simulation 1. Difference Between SDD and FGL for Each Metric.

n	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-2062 (194)	0.44 (0.06)	-0.58 (0.03)	0.17 (0.02)	0.06 (0.01)
200	-2428 (137)	0.50 (0.04)	-0.58 (0.02)	0.21 (0.01)	0.06 (0.02)
500	-4952 (326)	0.64 (0.03)	-0.45 (0.02)	0.42 (0.03)	0.10 (0.02)
1000	-5931 (325)	0.63 (0.03)	-0.33 (0.02)	0.51 (0.03)	-0.03 (0.03)
2000	-6259 (329)	0.55 (0.03)	-0.27 (0.02)	0.54 (0.03)	-0.08 (0.03)

Table 5: Simulation 1. Difference Between SDD and Naïve Method for Each Metric.

n	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-649 (46.0)	0.43 (0.06)	-0.40 (0.03)	0.05 (0.00)	0.00 (0.00)
200	-1060 (56.4)	0.50 (0.04)	-0.50 (0.01)	0.09 (0.00)	-0.01 (0.01)
500	-1754 (71.9)	0.64 (0.03)	-0.40 (0.02)	0.15 (0.01)	-0.02 (0.01)
1000	-2258 (103)	0.62 (0.03)	-0.33 (0.02)	0.19 (0.01)	-0.07 (0.01)
2000	-3109 (149)	0.55 (0.03)	-0.27 (0.02)	0.27 (0.01)	-0.04 (0.01)

Table 6: Simulation 1. Difference Between SDD and Hard Thresholding for Each Metric.

n	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-	-	-	-	-
200	16.0 (1.00)	0.13 (0.07)	0.28 (0.02)	-0.00 (0.00)	-0.67 (0.06)
500	20.3 (1.15)	-0.03 (0.06)	0.48 (0.02)	0.00 (0.00)	-0.22 (0.01)
1000	26.1 (1.86)	-0.06 (0.05)	0.51 (0.04)	-0.00 (0.00)	-0.27 (0.03)
2000	-182 (52.5)	-0.05 (0.05)	0.37 (0.05)	0.02 (0.00)	-0.50 (0.07)

Table 7: Simulation 2. Difference Between SDD and FGL for Each Metric.

n	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-4674 (227)	-0.00 (0.00)	-0.50 (0.02)	0.40 (0.02)	-1.25 (0.16)
200	-5296 (185)	-0.00 (0.00)	-0.60 (0.02)	0.45 (0.02)	-1.15 (0.12)
500	-10199 (119)	-0.00 (0.00)	-0.97 (0.01)	0.87 (0.01)	-3.20 (0.07)
1000	-9727 (155)	0.29 (0.04)	-0.84 (0.02)	0.83 (0.01)	-2.10 (0.06)
2000	-9189 (177)	0.33 (0.04)	-0.82 (0.02)	0.79 (0.02)	-1.35 (0.05)

Table 8: Simulation 2. Difference Between SDD and Naïve Method for Each Metric.

n	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-257 (10.7)	-0.02 (0.00)	-0.42 (0.01)	0.02 (0.00)	-0.02 (0.01)
200	-608 (46.2)	-0.02 (0.00)	-0.43 (0.00)	0.05 (0.00)	-0.01 (0.01)
500	-3564 (56.3)	-0.00 (0.00)	-0.54 (0.01)	0.30 (0.00)	-0.20 (0.03)
1000	-6483 (58.4)	0.29 (0.04)	-0.70 (0.02)	0.55 (0.00)	-0.59 (0.04)
2000	-9139 (98.6)	0.33 (0.04)	-0.82 (0.02)	0.78 (0.01)	-1.34 (0.04)

Table 9: Simulation 2. Difference Between SDD and Hard Thresholding for Each Metric.

n	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-	-	-	-	-
200	0.18 (0.14)	-0.06 (0.02)	-0.00 (0.00)	-0.00 (0.00)	-2.05 (0.18)
500	1.31 (0.36)	-0.46 (0.05)	-0.04 (0.01)	-0.00 (0.00)	-0.09 (0.02)
1000	2.87 (0.48)	-0.49 (0.06)	0.02 (0.01)	-0.00 (0.00)	0.05 (0.01)
2000	5.44 (0.84)	-0.51 (0.06)	0.06 (0.01)	-0.00 (0.00)	0.05 (0.01)

Table 10: Simulation 3. Difference Between SDD and FGL for Each Metric.

n	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-2948 (157)	0.81 (0.05)	-0.76 (0.03)	0.25 (0.01)	-0.01 (0.01)
200	-2979 (96.9)	0.72 (0.04)	-0.75 (0.02)	0.25 (0.01)	-0.01 (0.02)
500	-7711 (254)	0.63 (0.04)	-0.81 (0.02)	0.66 (0.02)	-0.27 (0.03)
1000	-8771 (137)	0.82 (0.03)	-0.71 (0.02)	0.75 (0.01)	-0.19 (0.02)
2000	-7953 (155)	0.83 (0.01)	-0.34 (0.02)	0.68 (0.01)	-0.08 (0.02)

Table 11: Simulation 3. Difference Between SDD and Naïve Method for Each Metric.

n	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-306 (12.7)	0.80 (0.05)	-0.20 (0.01)	0.03 (0.00)	0.04 (0.00)
200	-558 (20.0)	0.71 (0.04)	-0.21 (0.01)	0.05 (0.00)	-0.02 (0.00)
500	-2324 (34.7)	0.63 (0.04)	-0.72 (0.02)	0.20 (0.00)	0.07 (0.01)
1000	-3523 (49.6)	0.81 (0.03)	-0.65 (0.02)	0.30 (0.00)	0.09 (0.01)
2000	-5008 (61.8)	0.82 (0.01)	-0.33 (0.02)	0.43 (0.01)	0.03 (0.01)

Table 12: Simulation 3. Difference Between SDD and Hard Thresholding Method for Each Metric.

n	# Est edges	Precision	Recall	Accuracy	RRMSE
100	-	-	-	-	-
200	4.22 (0.31)	0.44 (0.06)	0.13 (0.01)	0.00 (0.00)	-0.85 (0.09)
500	5.96 (0.56)	-0.01 (0.06)	0.14 (0.01)	0.00 (0.00)	-0.14 (0.01)
1000	8.05 (0.72)	0.16 (0.06)	0.26 (0.02)	0.00 (0.00)	-0.17 (0.02)
2000	19.4 (1.87)	0.18 (0.05)	0.61 (0.02)	0.00 (0.00)	-0.32 (0.02)