EFFICIENT LEARNING OF SPARSE DIFFERENTIAL NETWORKS IN MULTI-SOURCE NON-PARANORMAL MODELS

Mojtaba Nikahd and Seyed Abolfazl Motahari

Department of Computer Engineering, Sharif University of Technology

Supplementary Material

S1 Proposed algorithm details

S1.1 Termination conditions

The algorithm progressively decreases the regularization parameter from infinity and identifies intervals over which the optimal solutions exhibit a fixed sparsity pattern. For each such interval, the method determines the unique sparsity structure and derives a closed-form expression for computing the global minimizer at any regularization value within the interval.

The proposed algorithm incorporates two termination criteria: one to ensure the uniqueness of the solution for a given regularization parameter and another to restrict the explored sparsity level along the solution path

for computational efficiency.

The uniqueness criterion is verified by checking the necessary optimality conditions for Problem (2.1) in the main manuscript. Specifically, when the sparsity pattern of the global minimizer is known in a segment, the subgradient optimality conditions reduce to a linear system in λ, as shown in Equation (2.3) of the main manuscript. For this linear system to be identifiable, the corresponding design matrix $\Gamma_{\mathcal{A}_t,\mathcal{A}_t}$ must be invertible. Accordingly, the algorithm includes an invertibility check as a termination criterion. Satisfying this condition ensures the uniqueness of the global minimizer along the solution path.

The second termination criterion restricts the explored sparsity level along the solution path. The parameter c, referred to as the active-set bound, specifies the maximum cardinality of the active set along the solution path. Importantly, c controls the maximum explored sparsity level but does not modify the underlying lasso-penalized D-trace optimization problem. In practice, c serves two purposes: (i) it limits the number of nonzero entries in the estimated differential networks, providing explicit sparsity control and preventing the construction of overly dense and impractical networks, and (ii) it enhances computational efficiency by bounding runtime and memory usage in high-dimensional settings.

S1.2 Active-set bound

Owing to the continuous piecewise linearity of the solution path established in Theorem 1 of the main manuscript, variables enter and leave the model sequentially as the regularization parameter decreases. This allows the algorithm to monitor the size of the active set along the path.

To control sparsity along the solution path, we introduce a parameter $c \in \mathbb{N}$, referred to as the active-set bound. This parameter specifies the maximum allowable cardinality of the active set and therefore determines the largest sparsity level explored by the algorithm. Equivalently, c limits the maximum number of nonzero entries in the estimated differential networks along the computed solution path. Importantly, c does not alter the underlying lasso-penalized D-trace optimization problem. It only limits how far along the exact solution path the algorithm proceeds, acting as a computational safeguard against excessively dense solutions. Specifically, the active-set bound enhances computational efficiency in high-dimensional settings, since both runtime and memory usage increase with the active-set size. Choosing an appropriate value for c enables efficient use of computational resources and prevents unnecessary computations. To examine the impact of varying c, we conduct a dedicated sensitivity analysis, the results of which are presented in Section S1.3.

S1.3 Sensitivity analysis of the active-set bound parameter c

To assess the effect of the active-set bound parameter c, we evaluated the proposed method across a range of values of c while keeping all other settings fixed. The experimental design follows that of the first simulation study in the original manuscript, except that the sample size was held constant. Specifically, the reported results are averaged over 100 independently generated datasets with dimensionality parameters $d \in \{50, 100\}$ and $m = m' = 500$. The differential network was constructed by randomly perturbing 20 edges of the underlying graph. Figure 1 reports precision–recall performance, computational time, memory consumption, and the number of detected knots as functions of c. The results show that increasing c yields estimators with higher recall. However, the marginal improvement diminishes as c grows. In contrast, runtime and memory consumption increase monotonically with c. Furthermore, the number of detected knots grows approximately linearly with the active-set bound. These findings demonstrate that the proposed method is statistically robust to moderate variation in c, particularly when c is chosen sufficiently large. At the same time, computational cost can be effectively controlled by avoiding excessive overestimation of this parameter.

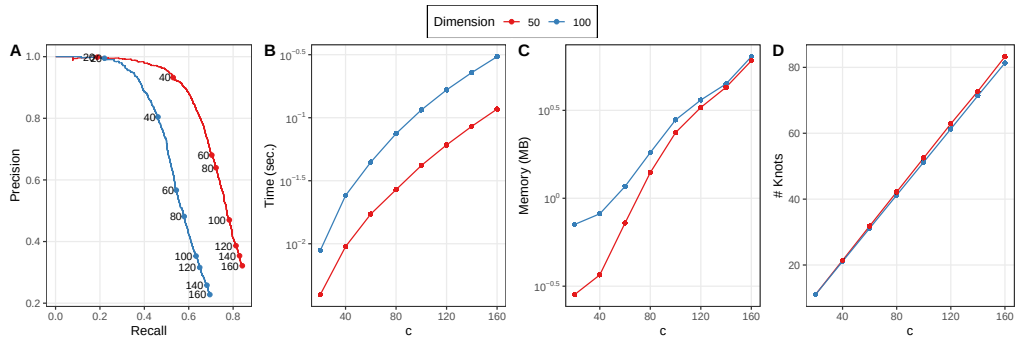

Figure 1: Impact of the active-set bound c on differential network inference performance and computational resources. (**A**) Precision–recall curves for dimensions $d \in \{50, 100\}$, with annotated points indicating selected values of c. (**B**) Runtime in seconds (log scale) as a function of c. (**C**) Memory consumption in megabytes (log scale) as a function of c. (**D**) Number of detected knots along the solution path. Experiments are averaged over 100 independently generated datasets, each with $m = m' = 500$ samples per group. Differential networks are generated by randomly perturbing 20 edges of the underlying graph.

S1.4 Computation complexity analysis

To analyze computational complexity, we define the matrices and sets as follows:

$$\Gamma = \frac{1}{2}\left\{\widehat{\Sigma} \otimes \widehat{\Sigma}' + \widehat{\Sigma}' \otimes \widehat{\Sigma}\right\}, \quad \mathbf{v} = \mathrm{vec}\left(\widehat{\Sigma} - \widehat{\Sigma}'\right), \quad \mathcal{A}^c = \left\{1, \ldots, d^2\right\} \setminus \mathcal{A}.$$

In each iteration, the computation of $\Gamma_{:,\mathcal{A}}$ and the matrix inverse $(\Gamma_{\mathcal{A},\mathcal{A}})^{-1}$ requires $\mathcal{O}(cd^2)$ and $\mathcal{O}(c^3)$ operations, respectively. Furthermore, Step 4 of Algorithm 1 in the main manuscript incurs an additional $\mathcal{O}(cd^2)$ computa-

tional cost. The remaining steps in each iteration contribute an additional $\mathcal{O}(c)$ operations. Accordingly, the overall computational complexity per iteration is $\mathcal{O}(cd^2 + c^3)$.

S1.5 Resumable path property

Although the parameter c determines how far along the exact solution path the algorithm proceeds, it does not impose a practical limitation due to the algorithm's resumable path property. Specifically, after computing a segment characterized by the tuple $(\lambda_t, \mathcal{A}_t, \mathbf{s}_t)$, the algorithm can proceed from this state without recomputing any previously obtained segments. All quantities required for subsequent updates are fully determined by the tuple associated with the most recently identified knot. Thus, continuation only requires initializing Step 1 of Algorithm 1 in the main manuscript with these values.

This property implies that the active-set bound c does not induce an irreversible truncation of the solution path. If a higher sparsity level is subsequently desired, the algorithm may be resumed from the last computed segment with an enlarged value of c, thereby extending the path without repeating prior computations. Consequently, the method naturally supports incremental refinement and we can use an incremental refinement, allowing

the value of c to be adjusted progressively in practice. Specifically, one may initially compute a partial solution path using a moderate value of c, evaluate model adequacy, and extend the path further only if additional sparsity levels are required.

S2 Covariance estimator details

We focus on estimating the covariance matrix for the first heterogeneous collection of datasets; the second collection can be handled in a similar manner. Recall that the diagonal entries of each covariance matrix are assumed to be 1, implying that each covariance matrix is equivalent to its corresponding Pearson correlation matrix. Furthermore, the set of transformation functions $\mathbf{f}$ is assumed to be unknown, which prevents us from directly estimating the covariance matrix of the underlying GGM. However, it is well established that in multivariate Gaussian models, there exists a one-to-one correspondence between the Pearson correlation and Kendall's tau correlation (Kruskal, 1958; Kendall, 1948). This relationship also extends to non-paranormal distributions. Specifically, if $\mathbf{z} \sim \mathrm{NPN}(\mathbf{0}, \Sigma, \mathbf{f})$ represents a d-dimensional non-paranormal distribution, then for all $k, l \in [d]$,

$$\Sigma_{k,l} = \sin\left(\frac{\pi}{2}\tau_{k,l}\right), \tag{S2.1}$$

where $\tau_{k,l}$ denotes Kendall's tau correlation between z_k and z_l (Liu et al., 2012).

We leverage the relationship in equation (S2.1) to construct an estimator of the Pearson correlation matrix. Specifically, let $X^{(r)}$ denote the data matrix of size $m_r \times d$ for dataset $\mathcal{D}_r$. The empirical Kendall's tau correlation between the kth and lth variables in dataset $\mathcal{D}_r$ is given by

$$\widehat{\tau}_{k,l}^{(r)} = 2\frac{\sum_{1\le i<j\le m_r} \operatorname{sign}\left[\left(X_{i,k}^{(r)} - X_{j,k}^{(r)}\right)\left(X_{i,l}^{(r)} - X_{j,l}^{(r)}\right)\right]}{m_r\left(m_r-1\right)}. \tag{S2.2}$$

To aggregate information across the heterogeneous datasets, we compute a weighted average of the individual Kendall's tau estimates:

$$\widehat{\tau}_{k,l}^{(w)} = \sum_{r=1}^{N} \frac{m_r}{m}\widehat{\tau}_{k,l}^{(r)}, \tag{S2.3}$$

where $m = \sum_{r=1}^{N} m_r$ is the total number of samples across all N datasets. Using the transformation in equation (S2.1), we then define the estimator of the correlation matrix, denoted by $\tilde{\Sigma}$, as follows:

$$\tilde{\Sigma}_{k,l} = \begin{cases} \sin\left(\frac{\pi}{2}\widehat{\tau}_{k,l}^{(w)}\right), & k \neq l \\ 1, & k = l. \end{cases} \tag{S2.4}$$

While the non-paranormal framework elegantly handles unknown transformations, applying a naive rank-based aggregation like $\tilde{\Sigma}$ across heterogeneous datasets introduces a substantial methodological obstacle: the resulting matrix $\tilde{\Sigma}$ is not guaranteed to be positive semi-definite (PSD). Ensuring

a PSD covariance estimate is a strict mathematical requirement to guarantee the convexity of the optimization problem (2.1) in the main manuscript. This convexity is absolutely essential for the optimality conditions to hold, which in turn is required by Theorem 1 of the main manuscript to guarantee the identification of an exact global solution without relying on approximation. Therefore, our covariance estimator must be meticulously tailored to satisfy these requirements. To address this, we project $\tilde{\Sigma}$ onto the cone of PSD matrices using the method proposed by Zhao et al. (2014), which identifies the nearest PSD matrix to $\tilde{\Sigma}$ under the max norm. This algorithm approximates the solution up to an arbitrary precision μ, where smaller values of μ yield higher accuracy at the cost of slower convergence rate. We denote the resulting projected matrix by $\widehat{\Sigma}(\mu)$, representing the final PSD estimate obtained from $\tilde{\Sigma}$ via this projection method. In the following, we formally present a concentration bound for the estimator $\widehat{\Sigma}(\mu)$ in lemma 1, with the proof provided in Section S4. It is notable that this bound help us to establish the error bounds for our proposed differential network analysis with the same order of sample complexity provided in Yuan et al. (2017) with different statistical setting. Notably, this bound allows us to establish error guarantees for the proposed differential network estimator with the same order of sample complexity as in Yuan et al. (2017), despite operating

under a different statistical setting.

Lemma 1. *Let $\mu > 0$ and $\delta > 0$ be arbitrary positive constants. The estimator $\widehat{\Sigma}(\mu)$ satisfies the following concentration inequality:*

$$\mathbb{P}\left(\left\|\widehat{\Sigma}(\mu) - \Sigma\right\|_{\infty} \geq \delta\right) \leq d^2 \exp\left(\frac{-m\left(\max\left\{2\delta - \mu, 0\right\}\right)^2}{32\pi^2}\right).$$

In particular, when $\mu = \delta$, the bound simplifies to:

$$\mathbb{P}\left(\left\|\widehat{\Sigma}(\mu) - \Sigma\right\|_{\infty} \geq \mu\right) \leq d^2 \exp\left(\frac{-m\mu^2}{32\pi^2}\right).$$

S3 Proof of Theorem 1

In this section, we present the proof of Theorem 1 by outlining the main steps and providing detailed arguments in the subsequent paragraphs. We begin by establishing the continuity of the solution path $\widehat{\Delta}(\lambda)$ in Lemma 2. We then demonstrate that the solution path exhibits a piecewise linear structure, as shown in Lemma 3. Finally, we show that Algorithm 1 correctly identifies the sequence of segment tuples $(\lambda_t, \mathcal{A}_t, \mathbf{s}_t)_{t=0}^{T}$, which fully characterizes the solution path over the interval (λ_T, ∞). Part of the argument is conceptually related to the path-following framework of Rosset and Zhu (2007). We therefore first discuss the relationship between our setting and that framework, and then present the lemmas and proofs required for our analysis.

Relation to the piecewise linear path framework of Rosset and Zhu (2007). The piecewise linearity of regularized solution paths has been studied in the general framework of Rosset and Zhu (2007). In particular, Theorem 2 of Rosset and Zhu (2007) shows that optimization problems of the form

$$\hat{\beta}(\lambda) = \arg\min_{\beta \in \mathbb{R}^p} L(y, X\beta) + \lambda J(\beta),$$

with the ℓ_1 penalty

$$J(\beta) = \|\beta\|_1,$$

generate piecewise linear coefficient paths when the loss function can be written as

$$L(y, X\beta) = \sum_i l(y_i, \beta^\top x_i),$$

where $l(\cdot)$ is a differentiable piecewise quadratic function of the residual or margin. Based on this result, Rosset and Zhu (2007) propose a general algorithm for computing the solution path.

Our setting differs from this framework in two important aspects.

First, Theorem 1 establishes not only the piecewise linearity of the solution path but also its continuity with respect to the regularization parameter. This property is essential for the correctness of Algorithm 1, which traces the path by sequentially identifying its breakpoints.

Second, the loss function in the D-trace formulation does not belong to the class considered in Rosset and Zhu (2007). Specifically, our objective

function is

$$L_D(\Delta; \widehat{\Sigma}, \widehat{\Sigma}') = \frac{1}{4}\left(\langle \widehat{\Sigma}\Delta, \Delta\widehat{\Sigma}'\rangle + \langle \widehat{\Sigma}'\Delta, \Delta\widehat{\Sigma}\rangle\right) - \langle \Delta, \widehat{\Sigma} - \widehat{\Sigma}'\rangle,$$

which is a quadratic form in the matrix parameter Δ rather than a sum of functions of the form $l(y_i, \beta^\top x_i)$.

In principle, one could attempt to transform this quadratic objective into a least-squares form in order to apply the results of Rosset and Zhu (2007). By vectorizing the parameter matrix Δ, the objective can be written as

$$2\,\mathrm{vec}(\Delta)^\top \Gamma\,\mathrm{vec}(\Delta) + \mathrm{vec}(\Delta)^\top \mathbf{v},$$

where

$$\Gamma = \frac{1}{2}\left(\widehat{\Sigma} \otimes \widehat{\Sigma}' + \widehat{\Sigma}' \otimes \widehat{\Sigma}\right), \quad \mathbf{v} = \mathrm{vec}(\widehat{\Sigma}' - \widehat{\Sigma}).$$

Since Γ is positive semidefinite, it admits a factorization $\Gamma = X^\top X$, allowing the objective to be written as an equivalent least-squares problem. However, constructing such a representation requires factorizing the matrix $\Gamma \in \mathbb{R}^{d^2 \times d^2}$. The cost of this step alone is at least $\Omega(d^4)$ simply to access all entries of Γ, and standard decompositions such as the Cholesky factorization require $O(d^6)$ operations. In addition, constructing the corresponding response vector requires solving a linear system of dimension d^2.

Therefore, converting the D-trace objective to the form required by

Rosset and Zhu (2007) would introduce a computational overhead substantially larger than the complexity of the proposed algorithm itself. Instead, we directly exploit the structure of the D-trace loss to establish the piecewise linearity and continuity of the solution path and derive Algorithm 1 without performing this transformation.

It is also worth noting that existing methods for differential network estimation are typically formulated and implemented directly in terms of the matrix-valued D-trace objective and its gradient, without explicitly constructing or factorizing the full Kronecker-product Hessian; see, for example, Zhao et al. (2014); Yuan et al. (2017); Zhao et al. (2022). The main distinction of the proposed method is therefore not the avoidance of Kronecker factorization itself, but rather the development of a D-trace-specific active-set path-following algorithm that exploits the piecewise-linear structure of the regularization path. This enables efficient computation of exact solutions across a range of regularization parameters.

Lemma 2. *Suppose that optimization problem (2.1) admits a unique minimizer for all $\lambda \in (\lambda_T, \infty)$, and that $\widehat{\Sigma}$ and $\widehat{\Sigma}'$ are positive semi-definite matrices. Then the solution path $\widehat{\Delta}(\lambda)$ is continuous with respect to λ on the interval (λ_T, ∞).*

Proof. The Hessian of the function $L_D(\Delta; \widehat{\Sigma}, \widehat{\Sigma}')$ (defined in Equation (2.2)

of the main manuscript) with respect to Δ is $\left(\widehat{\Sigma} \otimes \widehat{\Sigma}' + \widehat{\Sigma}' \otimes \widehat{\Sigma}\right)/2$, where $\otimes$ denotes the Kronecker product. Since both $\widehat{\Sigma}$ and $\widehat{\Sigma}'$ are positive semi-definite matrices, the function $L_D(\Delta; \widehat{\Sigma}, \widehat{\Sigma}')$ is convex with respect to the parameter Δ. The ℓ_1 norm $\|\Delta\|_1$ is also convex. Consequently, the objective function in optimization problem (2.1), namely $L_D(\Delta; \widehat{\Sigma}, \widehat{\Sigma}') + \lambda\|\Delta\|_1$, is jointly convex in (Δ, λ) over the domain $\mathbb{R}^{p\times p} \times \mathbb{R}^+$.

Furthermore, the objective function is continuous in both Δ and λ parameters. The feasible set for Δ is $\mathbb{R}^{p\times p}$, which is nonempty and by assumption, optimization problem (2.1) has a unique minimizer for all $\lambda \in (\lambda_T, \infty)$. Hence, the conditions required by Theorem 5.2 in Still (2018) are satisfied. This theorem states that, for a parametric convex optimization problem, if the feasible set is nonempty and the objective function is continuous in both the optimization variable and the parameter, then the optimal solution is continuous with respect to the parameter whenever the minimizer is unique.

Therefore, it follows that the solution path $\widehat{\Delta}(\lambda)$ is continuous with respect to λ on the interval (λ_T, ∞). □

Lemma 3. *Assume that $\widehat{\Sigma}$ and $\widehat{\Sigma}'$ are positive semi-definite. Let $\vec{\Delta}(\lambda) = \mathrm{vec}\left(\widehat{\Delta}(\lambda)\right)$ denote the vectorized solution of problem (2.1). There exists a sequence of segment tuples $(\lambda_t, \mathcal{A}_t, \mathbf{s}_t)_{t=0}^T$, with $\infty = \lambda_0 > \lambda_1 > \lambda_2 >$*

$\cdots > \lambda_T$, *where* $\mathcal{A}_t \neq \mathcal{A}_{t+1}$ *for all* $t \in \{0, 1, \ldots, T-1\}$, *such that for every* $t \in \{0, 1, \ldots, T-1\}$ *and for all* $\lambda \in [\lambda_{t+1}, \lambda_t]$, *the solution* $\vec{\Delta}(\lambda)$ *satisfies:*

$$\vec{\Delta}_{\mathcal{A}_t}(\lambda) = -\left(\Gamma_{\mathcal{A}_t,\mathcal{A}_t}\right)^{-1}\left(\mathbf{v}_{\mathcal{A}_t} + \lambda \mathbf{s}_t\right), \tag{S3.1}$$

$$\vec{\Delta}_{\mathcal{A}_t^c}(\lambda) = 0, \tag{S3.2}$$

provided that the matrix $\Gamma_{\mathcal{A}_t,\mathcal{A}_t}$ *is invertible. Here,* $\mathcal{A}_t$ *denotes the active set in the* t*-th linear segment, and* $\mathbf{s}_t$ *is the corresponding sign vector.*

Proof. Our analysis is conceptually related to the path-following arguments used in Rosset and Zhu (2007), but is adapted to the specific structure of the D-trace loss. In particular, we analyze the subgradient optimality conditions of problem (2.1). The proof exploits the convexity of the Lasso-penalized D-trace loss function to characterize the solution path via subgradient conditions that capture the relationship among the differential network estimator, the covariance matrix estimators, and the regularization parameter λ.

We begin the proof by assuming that the active set (the indices of the nonzero coefficients) and its corresponding sign vector are known for a given value of λ. Under this assumption, we show that solving the optimization problem (2.1) reduces to solving a linear system restricted to the active components. This analysis is then extended to show that the same rela-

tionship holds for all values of λ within an interval around the given point, thereby establishing that the solution path is piecewise linear with respect to λ.

As shown in the proof of Lemma 2, under the assumption that the covariance matrix estimators $\widehat{\Sigma}$ and $\widehat{\Sigma}'$ are positive semi-definite, the Lasso-penalized D-trace loss function defined in Equation (2.1) is convex. This convexity property permits the use of subgradient-based optimality conditions to characterize the solution (Beck, 2017). Specifically, the optimal solution $\widehat{\Delta}(\lambda)$ satisfies the following condition:

$$\begin{aligned} 0_{d,d} &\in \partial\left(L_D\left(\widehat{\Delta}\left(\lambda\right);\widehat{\Sigma},\widehat{\Sigma}'\right)+\lambda\left\|\widehat{\Delta}\left(\lambda\right)\right\|_1\right) \\ &= \frac{1}{2}\left(\widehat{\Sigma}\widehat{\Delta}\left(\lambda\right)\widehat{\Sigma}'+\widehat{\Sigma}'\widehat{\Delta}\left(\lambda\right)\widehat{\Sigma}\right)-\widehat{\Sigma}+\widehat{\Sigma}'+\lambda\partial\left\|\widehat{\Delta}\left(\lambda\right)\right\|_1, \end{aligned}$$

where $\widehat{\Delta}(\lambda)$ denotes the minimizer of the objective in Equation (2.1) corresponding to the regularization parameter λ.

We reformulate the subgradient optimality condition in vectorized form as:

$$\vec{0}\in\left\{\Gamma\vec{\Delta}\left(\lambda\right)+\mathbf{v}+\lambda\partial\left\|\vec{\Delta}\left(\lambda\right)\right\|_1\right\},$$

where $\Gamma=\frac{1}{2}\left\{\widehat{\Sigma}\otimes\widehat{\Sigma}'+\widehat{\Sigma}'\otimes\widehat{\Sigma}\right\}$, $\mathbf{v}=\operatorname{vec}\left(\widehat{\Sigma}'-\widehat{\Sigma}\right)$, $\vec{\mathbf{0}}=\operatorname{vec}\left(0_{d,d}\right)$ and $\vec{\Delta}\left(\lambda\right)=\operatorname{vec}\left(\widehat{\Delta}\left(\lambda\right)\right)$. Hence, we have:

$$\Gamma\vec{\Delta}\left(\lambda\right)+\mathbf{v}=-\lambda\nu\left(\lambda\right), \tag{S3.3}$$

where each component $\nu_i(\lambda)$ of $\nu(\lambda)$ satisfies:

$$\nu_i(\lambda) \in \begin{cases} \{1\}, & \text{if } \vec{\Delta}_i(\lambda) > 0 \\ \{-1\}, & \text{if } \vec{\Delta}_i(\lambda) < 0 \\ [-1,1], & \text{if } \vec{\Delta}_i(\lambda) = 0 \end{cases}, i = 1, \ldots, d^2.$$

Therefore, for each $i \in \{1, \ldots, d^2\}$, the subgradient optimality condition given in Equation (S3.3) can be equivalently expressed as follows:

$$\begin{cases} \Gamma_{i,:}\vec{\Delta}(\lambda) + \mathbf{v}_i = -\lambda \text{sign}\left(\vec{\Delta}_i(\lambda)\right), & \text{if } \vec{\Delta}_i(\lambda) \neq 0 \\ |\Gamma_{i,:}\vec{\Delta}(\lambda) + \mathbf{v}_i| \leq \lambda, & \text{if } \vec{\Delta}_i(\lambda) = 0. \end{cases} \tag{S3.4}$$

Let us define the set of active variables $\mathcal{A}_\lambda$ as the indices corresponding to the nonzero entries of $\vec{\Delta}(\lambda)$, i.e.,

$$\mathcal{A}_\lambda = \left\{ i \in \left\{1, \ldots, d^2\right\} : \vec{\Delta}_i(\lambda) \neq 0 \right\}.$$

Correspondingly, we define the active sign vector $\mathbf{s}_{\mathcal{A}_\lambda}$ as:

$$\mathbf{s}_{\mathcal{A}_\lambda} = \text{sign}\left(\vec{\Delta}_{\mathcal{A}_\lambda}(\lambda)\right).$$

It is obvious that given an active set $\mathcal{A}_\lambda$ and its associated sign vector $\mathbf{s}_{\mathcal{A}_\lambda}$, the subgradient optimality conditions specified in Equation (S9.1) reduce to a system of linear equations. Moreover, if the submatrix $\Gamma_{\mathcal{A}_\lambda,\mathcal{A}_\lambda}$ is invertible,

this system admits a unique solution, given by:

$$\vec{\Delta}_{\mathcal{A}_\lambda}(\lambda) = -\left(\Gamma_{\mathcal{A}_\lambda,\mathcal{A}_\lambda}\right)^{-1}\left(\mathbf{v}_{\mathcal{A}_\lambda} + \lambda \mathbf{s}_{\mathcal{A}_\lambda}\right), \tag{S3.5}$$

$$\vec{\Delta}_{\mathcal{A}_\lambda^c}(\lambda) = 0. \tag{S3.6}$$

Accordingly, the exact solution to the optimization problem (2.1) can be explicitly computed for any value of λ, provided that the active set $\mathcal{A}_\lambda$ and its corresponding sign vector $\mathbf{s}_{\mathcal{A}_\lambda}$ are known, and the submatrix $\Gamma_{\mathcal{A}_\lambda,\mathcal{A}_\lambda}$ is invertible.

A key observation is that as the regularization parameter λ decreases, the system of linear equations given in Equations (S3.5) and (S3.6) remains unchanged across different values of λ, provided that the corresponding active set and its associated sign vector remain fixed. This observation implies the existence of a sequence $(\lambda_t)_{t=0}^T$, where $\infty = \lambda_0 > \lambda_1 > \lambda_2 > \cdots > \lambda_T$, such that for each $t \in \{0, 1, \ldots, T-1\}$, the active set and its sign pattern remain constant throughout the interval $(\lambda_{t+1}, \lambda_t)$. This invariance implies that the solution $\vec{\Delta}(\lambda)$ evolves linearly with respect to λ over each such interval. More concretely, let $\mathcal{A}_t$ and $\mathbf{s}_t$ denote the active set and its associated sign vector for all $\lambda \in (\lambda_{t+1}, \lambda_t)$. Then, for each $t \in \{0, 1, \ldots, T-1\}$

and all $\lambda \in (\lambda_{t+1}, \lambda_t)$, the solution is given by:

$$\vec{\Delta}_{\mathcal{A}_t}(\lambda) = -\left(\Gamma_{\mathcal{A}_t,\mathcal{A}_t}\right)^{-1}\left(\mathbf{v}_{\mathcal{A}_t} + \lambda \mathbf{s}_t\right),$$

$$\vec{\Delta}_{\mathcal{A}_t^c}(\lambda) = 0,$$

provided that the matrix $\Gamma_{\mathcal{A}_t,\mathcal{A}_t}$ is invertible. Moreover, since Lemma 2 guarantees continuity of $\widehat{\Delta}(\lambda)$, these expressions in fact hold on the closed interval $\lambda \in [\lambda_{t+1}, \lambda_t]$, thereby establishing the continuous piecewise-linear structure of the solution path. □

Lemma 3 establishes that the solution path $\widehat{\Delta}(\lambda)$ over the interval (λ_T, ∞) is continuous and piecewise linear, with its structure fully characterized by the sequence of segment tuples $(\lambda_t, \mathcal{A}_t, \mathbf{s}_t)_{t=0}^T$. Moreover, it shows that each pair of successive linear segments is associated with a distinct active set. In what follows, we focus on proving that Algorithm 1 correctly recovers this sequence of segment tuples, thereby completing the proof of the theorem.

To establish the correctness of Algorithm 1 in recovering the segment tuples $(\lambda_t, \mathcal{A}_t, \mathbf{s}_t)_{t=0}^T$, we employ a mathematical induction argument consisting of two parts: the base case and the induction step. Algorithm 1 initializes the first segment tuple as $(\infty, \emptyset, \emptyset)$. For all $\lambda \in (\lambda_1, \infty)$, it is obvious that $\vec{\Delta}(\lambda) = \vec{\mathbf{0}}$. Hence the corresponding active set is empty and

the initial tuple $(\lambda_0, \mathcal{A}_0, \mathbf{s}_0) = (\infty, \emptyset, \emptyset)$ is correctly initialized. Thus, the base case holds.

We now turn to the induction step. More concretely, assuming that the t-th tuple $(\lambda_t, \mathcal{A}_t, \mathbf{s}_t)$ has been identified correctly, we aim to show that the algorithm's update rules produce the correct $(t+1)$st tuple $(\lambda_{t+1}, \mathcal{A}_{t+1}, \mathbf{s}_{t+1})$. This is achieved by enforcing the subgradient optimality conditions. At each iteration, the algorithm evaluates all possible events—namely, the entry or exit of variables from the active set—and selects the event that would first violate the optimality conditions as λ decreases. The active set and the corresponding sign vector are then update at the identified λ_{t+1}, thereby maintaining feasibility. By iteratively applying this procedure, the algorithm guarantees that the subgradient optimality conditions are satisfied throughout the entire piecewise-linear solution path. Consequently, it suffices to verify that the evaluation phase (Steps 4–6 of Algorithm 1) and the update phase (Steps 7–12 of Algorithm 1) correctly identify the first violating event and maintain the validity of the subgradient conditions at each iteration. To this end, we first establish the correctness of the evaluation phase, followed by a verification of the correctness of the update phase.

In the evaluation phase (Steps 4–6 of Algorithm 1), all potential events that may violate the optimality conditions are examined and categorized

into two types: exit events and entry events. This classification is motivated by Lemma 3, which establishes that consecutive linear segments of the solution path are characterized by distinct active sets. Both categories are assessed to identify the nearest possible event to the current knot λ_t that violates the optimality conditions. After determining the candidate knots for each category, the one closest to λ_t is selected as the next knot, denoted by λ_{t+1}. We first analyze the scenario in which variables exit the active set, followed by the scenario in which variables enter it.

When a variable exits the active set, its corresponding component in the differential network estimator becomes zero. Since the current segment tuple $(\lambda_t, \mathcal{A}_t, \mathbf{s}_t)$ is given, Equation (S3.1) characterizes the direction of the solution path along the current linear segment. Based on this characterization, one can determine the value of the regularization parameter λ at which each variable becomes zero along this line. Specifically, for each variable $u \in \mathcal{A}$, the following condition holds by applying Equation (S3.1):

$$-\left[\left(\Gamma_{\mathcal{A}_t,\mathcal{A}_t}\right)^{-1}\right]_{u,:}\left(\mathbf{v}_{\mathcal{A}_t}+\lambda_t^{\text{exit}}(u)\,\mathbf{s}_t\right)=0,$$

where $\lambda_t^{\text{exit}}(u)$ denotes the value of λ at which the variable u becomes zero during the t-th identified line. Solving for $\lambda_t^{\text{exit}}(u)$ yields:

$$\lambda_t^{\text{exit}}(u)=\frac{-\left[\left(\Gamma_{\mathcal{A}_t,\mathcal{A}_t}\right)^{-1}\right]_{u,:}\mathbf{v}_{\mathcal{A}_t}}{\left[\left(\Gamma_{\mathcal{A}_t,\mathcal{A}_t}\right)^{-1}\right]_{u,:}\mathbf{s}_t}.$$

To identify a candidate for the next successive knot along the solution path, corresponding to a potential exit event, we must determine the closest value of λ (as it decreases from λ_t) at which a variable in the active set becomes zero. This is achieved by computing $\lambda_t^{\text{exit}}(u)$ for all $u \in \mathcal{A}_t$, and selecting the largest among those that are strictly less than λ_t. Formally, the candidate knot associated with an exit event, denoted by λ^{exit} is obtained by solving the following optimization problem:

$$\begin{aligned} \lambda^{\text{exit}} = \max_{u} \quad & \lambda_t^{\text{exit}}(u), \\ \text{s.t.} \quad & \lambda_t^{\text{exit}}(u) < \lambda_t, u \in \mathcal{A}_t. \end{aligned}$$

This optimization procedure identifies the earliest potential exit event occurring after λ_t as the regularization parameter λ decreases. Consequently, no exit events occur for any $\lambda \in (\lambda^{\text{exit}}, \lambda_t)$. Moreover, if the variable(s) attaining the maximum—i.e., those for which $\lambda_t^{\text{exit}}(u) = \lambda^{\text{exit}}$—are not removed from the active set at $\lambda = \lambda^{\text{exit}}$, and no earlier knot is detected as λ decreases from λ_t, then the linearity of the current segment implies that the sign of the corresponding coefficient would change, thereby violating the assumed sign pattern. Therefore, a knot must necessarily lie within the interval $[\lambda^{\text{exit}}, \lambda_t)$. This scenario is explicitly addressed in Step 5 of Algorithm 1, where a notational simplification is employed: specifically, $g(u)$ is used to denote $\lambda_t^{\text{exit}}(u)$ for the sake of clarity.

We now consider the case in which one or more variables enter the active set. Specifically, when a variable $u \in \mathcal{A}_t^c$ becomes active at a particular value of the regularization parameter λ, with associated sign s_u, the following optimality condition must be satisfied:

$$\Gamma_{u,:}\vec{\Delta}(\lambda) + \mathbf{v}_u = -\lambda s_u, \tag{S3.7}$$

which is derived from the subgradient optimality conditions presented in Equation (S9.1) of Lemma 3. Since the current segment tuple $(\lambda_t, \mathcal{A}_t, \mathbf{s}_t)$ is given, Equation (S3.1) characterizes the direction of the solution path along the current linear segment. By substituting this expression into Equation (S3.7), we can compute the value of the regularization parameter λ at which each variable satisfies the entry condition. Specifically, for each $u \in \mathcal{A}_t^c$ and $s_u \in \{-1, 1\}$, the following equation holds:

$$\Gamma_{u,\mathcal{A}_t}\left(\Gamma_{\mathcal{A}_t,\mathcal{A}_t}\right)^{-1}\left(\mathbf{v}_{\mathcal{A}_t} + \lambda_t^{\text{entry}}(u, s_u)\,\mathbf{s}_t\right) - v_u = s_u\lambda_t^{\text{entry}}(u, s_u), \tag{S3.8}$$

where $\lambda_t^{\text{entry}}(u, s_u)$ denotes the value of λ at which the variable u with sign s_u meets the entry condition in Equation (S3.7). Solving for $\lambda_t^{\text{entry}}(u, s_u)$ yields:

$$\lambda_t^{\text{entry}}(u, s_u) = \frac{\Gamma_{u,\mathcal{A}_t}\left(\Gamma_{\mathcal{A}_t,\mathcal{A}_t}\right)^{-1}\mathbf{v}_{\mathcal{A}_t} - v_u}{s_u - \Gamma_{u,\mathcal{A}_t}\left(\Gamma_{\mathcal{A}_t,\mathcal{A}_t}\right)^{-1}\mathbf{s}_t}.$$

As in the case of exit events, to identify a candidate for the next successive knot along the solution path, corresponding to a potential entry

event, we must determine the closest value of λ (as it decreases from λ_t) at which one or some variables in the complement set of active set satisfies the entrance conditions provided in Equation (S3.7). This is achieved by computing $\lambda_t^{\text{entry}}(u, s_u)$ for all $u \in \mathcal{A}_t$, and selecting the largest among those that are strictly less than λ_t. Formally, the candidate knot associated with an entry event, denoted by λ^{entry} is obtained by solving the following optimization problem:

$$\begin{aligned} \lambda^{\text{entry}} = \max_{u, s_u} \quad & \lambda_t^{\text{entry}}(u, s_u), \\ \text{s.t.} \quad & \lambda_t^{\text{entry}}(u, s_u) < \lambda_t, u \in \mathcal{A}_t^c, s_u \in \{-1, 1\}. \end{aligned}$$

This optimization procedure identifies the earliest potential entry event occurring after λ_t as the regularization parameter λ decreases. Consequently, no entry events occur for any $\lambda \in (\lambda^{\text{entry}}, \lambda_t)$. Moreover, if the variable(s) attaining the maximum—i.e., those for which $\lambda_t^{\text{entry}}(u) = \lambda^{\text{entry}}$—are not added to the active set at $\lambda = \lambda^{\text{entry}}$, and no earlier knot is detected as λ decreases from λ_t, then the linear dependence of the term $\Gamma_{u,:}\vec{\Delta}(\lambda) + \mathbf{v}_u$ on λ implies that, for those variable(s), this term exceeds λ^{entry} in magnitude for all $\lambda < \lambda^{\text{entry}}$. This violates the optimality conditions given in Equation (S9.1) of Lemma 3. Therefore, a knot must necessarily lie within the interval $[\lambda^{\text{exit}}, \lambda_t)$. This argument is analogous to the argument previously established for the case of exit events. Notably, this scenario is explicitly

addressed in Step 4 of Algorithm 1, where a notational simplification is employed: specifically, $h\left(u, s_u\right)$ is used to denote $\lambda_t^{\text{entry}}\left(u, s_u\right)$ for the sake of clarity.

By analyzing the two possible cases—variable entry and exit—the value of the tuning parameter at the next knot must correspond to the closest impending event in order to preserve the optimality conditions in both cases. That is, $\lambda_{t+1} = \max\left\{\lambda^{\text{exit}}, \lambda^{\text{entry}}\right\}$, which corresponds to Step 6 of Algorithm 1. By consolidating the arguments from both scenarios, it follows that the optimality conditions are satisfied for all $\lambda \in (\lambda_{t+1}, \lambda_t)$. Conversely, the optimality conditions are violated for any $\lambda < \lambda_{t+1}$ if the active set remains unchanged. This implies that λ_{t+1} indeed corresponds to a valid knot on the solution path. Consequently, Algorithm 1 correctly identifies the knots at each iteration, as Steps 4–6 implement this procedure.

We now turn our attention to the update phase (Steps 7–12 of Algorithm 1), during which the active set and its corresponding sign vector for the next segment (segment $t+1$) are determined. In the evaluation phase, we identified a set of candidate events that must occur at λ_{t+1}, and these events dictate the necessary updates to the active set. No other modifications are permissible. Specifically, removing a non-candidate variable from the active set would introduce a discontinuity in the solution path,

contradicting the continuity property established in Lemma 2. Similarly, adding a non-candidate variable would violate the optimality conditions, as previously discussed. Therefore, applying only the identified candidate events at λ_{t+1} is essential for preserving the validity of the solution. This update procedure is formalized in Steps 7–12 of Algorithm 1, which ensures that the algorithm correctly updates the segment tuples at each iteration. Consequently, the inductive step holds, and Algorithm 1 correctly identifies the sequence of segment tuples $(\lambda_t, \mathcal{A}_t, \mathbf{s}_t)_{t=0}^{T}$, thereby completing the proof.

S4 Proof of Lemma 1

The proof of the lemma proceeds in three steps. First, we derive a concentration bound for the weighted Kendall's tau estimator $\widehat{\tau}_{k,l}^{(w)}$ for all pairs $k \neq l$. Second, we use this bound to obtain a concentration result for the corresponding entry of the correlation matrix estimator $\tilde{\Sigma}_{k,l}$. Finally, we establish a Probably Approximately Correct (PAC) bound in the max norm for the proposed covariance matrix estimator $\widehat{\Sigma}(\mu)$, as stated in the lemma.

For the sake of convenience, let $z_{r,i} = \left\{X_{i,k}^{(r)}, X_{i,l}^{(r)}\right\}$, and define the function $h\left(z_{r,i}, z_{r,j}\right) = \operatorname{sign}\left(\left(X_{i,k}^{(r)} - X_{j,k}^{(r)}\right)\left(X_{i,l}^{(r)} - X_{j,l}^{(r)}\right)\right)$. Recall that the statistic $\widehat{\tau}_{k,l}^{(w)}$ is a function over the samples, i.e., $\widehat{\tau}_{k,l}^{(w)} : (\mathbb{R} \times \mathbb{R})^m \mapsto \mathbb{R}$, and

it can be rewritten as follows:

$$\widehat{\tau}_{k,l}^{(w)}\left(z_{1,1},\ldots,z_{N,m_N}\right)=\sum_{r=1}^{N}\frac{2\alpha_r}{m_r\left(m_r-1\right)}\sum_{1\leq i<j\leq m_r}h\left(z_{r,i},z_{r,j}\right). \tag{S4.1}$$

To provide a concentration bound for all pairs $k\neq l$ on $\widehat{\tau}_{k,l}^{(w)}$, we apply McDiarmid's Inequality (McDiarmid et al., 1989). Given the assumption of independence between datasets and samples within datasets, we focus on demonstrating the bounded difference property, which is a sufficient condition for applying McDiarmid's Inequality to the weighted average of the individual Kendall's tau estimators, $\widehat{\tau}_{k,l}^{(w)}$. For all indices $k,l\in[d]$ with $k\neq l$, for any $r\in[N]$, $u\in[m_r]$, and for any collection of points $z_{1,1},\ldots,z_{N,m_N},z_{r,u}^{*}\in\mathbb{R}\times\mathbb{R}$, we have

$$\begin{aligned}
&\left|\widehat{\tau}_{k,l}^{(w)}\left(z_{1,1},\ldots,z_{r,u},\ldots,z_{N,m_N}\right)-\widehat{\tau}_{k,l}^{(w)}\left(z_{1,1},\ldots,z_{r,u}^{*},\ldots,z_{N,m_N}\right)\right|\\
&\overset{(a)}{=}\frac{2\alpha_r}{m_r\left(m_r-1\right)}\left|\sum_{j\neq u}h\left(z_{r,u},z_{r,j}\right)-h\left(z_{r,u}^{*},z_{r,j}\right)\right|\\
&\leq\frac{2\alpha_r}{m_r\left(m_r-1\right)}\times 2\left(m_r-1\right)=\frac{4\alpha_r}{m_r},
\end{aligned}$$

where (a) follows by substituting equation (S4.1) and eliminating the similar terms. This provides the bounds required by the bounded differences property condition. Therefore, for all $k,l\in[d]$ with $k\neq l$, and for any $t>0$, McDiarmid's Inequality gives:

$$\mathbb{P}\left(\left|\widehat{\tau}_{k,l}^{(w)}-\tau_{k,l}^{(w)}\right|\geq t\right)\leq 2\exp\left(\frac{-2t^2}{\sum_{r=1}^{N}\sum_{i=1}^{m_r}\left(\frac{4\alpha_r}{m_r}\right)^2}\right)=2\exp\left(\frac{-t^2}{8\sum_{r=1}^{N}\frac{{\alpha_r}^2}{m_r}}\right).$$

We set $\alpha_r = \frac{m_r}{m}$ for all $r \in [N]$ to obtain the tightest bound. Thus, we obtain:

$$\mathbb{P}\left(\left|\widehat{\tau}_{k,l}^{(w)} - \tau_{k,l}^{(w)}\right| \geq t\right) \leq 2\exp\left(\frac{-mt^2}{8}\right). \tag{S4.2}$$

To derive an error bound for the corresponding entry of the correlation matrix estimator, we utilize the Lipschitz continuity of the sine function. Specifically, we obtain:

$$\begin{aligned}\mathbb{P}\left(\left|\tilde{\Sigma}_{k,l} - \Sigma_{k,l}\right| \geq t\right) &\overset{(a)}{=} \mathbb{P}\left(\left|\sin\left(\frac{\pi}{2}\left[\widehat{\tau}_w\right]_{k,l}\right) - \sin\left(\frac{\pi}{2}\left[\tau\right]_{k,l}\right)\right| \geq \frac{t}{4}\right) \\ &\overset{(b)}{\leq} \mathbb{P}\left(\left|\frac{\pi}{2}\left[\widehat{\tau}_w\right]_{k,l} - \frac{\pi}{2}\left[\tau_w\right]_{k,l}\right| \geq t\right) \overset{(c)}{\leq} 2\exp\left(\frac{-mt^2}{2\pi^2}\right),\end{aligned}$$

where step (a) follows from the equation in (2.6), step (b) uses the fact that the sine function is 1-Lipschitz, i.e., $|\sin(x) - \sin(y)| \leq |x - y|$, and step (c) follows from the concentration inequality established in (S4.2). Consequently, by applying a union bound over all off-diagonal entries, we obtain the following concentration inequality in the max norm for any $t > 0$:

$$\mathbb{P}\left(\left\|\tilde{\Sigma} - \Sigma\right\|_\infty \geq t\right) \leq d^2\exp\left(\frac{-mt^2}{2\pi^2}\right). \tag{S4.3}$$

Finally, we adopt the projection method proposed by Zhao et al. (2014) to map $\tilde{\Sigma}$ onto the positive semi-definite cone, resulting in the estimator $\widehat{\Sigma}(\mu)$. According to equation (A.6) in Zhao et al. (2014), the following inequality holds:

$$\left\|\widehat{\Sigma}(\mu) - \Sigma\right\|_\infty \leq 2\left\|\tilde{\Sigma} - \Sigma\right\|_\infty + \frac{\mu}{2}, \tag{S4.4}$$

where $\mu > 0$ is an arbitrary parameter that regulates the approximation error introduced by the projection. Consequently, for any $\delta > 0$, we have:

$$\begin{aligned}\mathbb{P}\left(\left\|\widehat{\Sigma}(\mu) - \Sigma\right\|_\infty \geq \delta\right) &\overset{(a)}{\leq} \mathbb{P}\left(2\left\|\tilde{\Sigma} - \Sigma\right\|_\infty + \frac{\mu}{2} \geq \delta\right) \\ &\overset{(b)}{\leq} d^2 \exp\left(\frac{-m\left(\max\left\{2\delta - \mu, 0\right\}\right)^2}{32\pi^2}\right), \qquad \text{(S4.5)}\end{aligned}$$

where step (a) follows from inequality (S4.4), and step (b) is obtained by applying the concentration bound given in (S4.3), using the substitution $t = (2\delta - \mu)/4$ for the case $\delta > \mu/2$. For the case $\delta < \mu/2$, we use the fact that $\max\{2\delta - \mu, 0\} = 0$, leading to a trivial upper bound on the probability. Therefore, by setting $\delta = \mu$, we obtain:

$$\mathbb{P}\left(\left\|\widehat{\Sigma}(\mu) - \Sigma\right\|_\infty \geq \mu\right) \leq d^2 \exp\left(\frac{-m\mu^2}{32\pi^2}\right),$$

which completes the argument.

S5 Proof of Theorem 2

To establish the results of our theorem, we adopt the proof strategy developed in Theorem 1 of Yuan et al. (2017). We begin by presenting the key definitions and lemmas that underpin this approach, and then proceed to apply them in the proof of our theorem. The original proof presented in Yuan et al. (2017) relies on the satisfaction of the tail condition, denoted by $\mathcal{T}(f, \nu_*)$, for the covariance matrix estimator. Specifically,

this condition requires that there exist a constant $\nu_* > 0$ and a function $f\colon \mathbb{N} \times (0,\infty) \to (0,\infty)$ such that

$$\mathbb{P}\left(\left\|\widehat{\Sigma}^m - \Sigma\right\|_\infty < \delta\right) > 1 - d^2/f\left(m,\delta\right)$$

for any $0 < \delta < 1/\nu_*$, where Σ denotes the true covariance matrix and $\widehat{\Sigma}^m$ is the corresponding estimator constructed from m independent samples. Moreover, assuming the tail condition holds, we define the following inverse functions for any fixed $\delta > 0$, $m \in \mathbb{N}$, and for all $r \geq 1$:

$$m_f(\delta, r) = \max\left\{m : f\left(\delta, r\right) \leq r\right\}, \tag{S5.1}$$

$$\delta_f(m, r) = \max\left\{\delta : f\left(m, \delta\right) \leq r\right\}. \tag{S5.2}$$

The original proof provided in Yuan et al. (2017) establishes a general result applicable to a broad class of covariance matrix estimators, which can be reformulated as the following lemma.

Lemma 4. *Let the threshold parameter $\bar{\delta}$ be defined as:*

$$\min\left\{\sqrt{M^2 + (6s\kappa_\Gamma)^{-1}} - M, \sqrt{2M^2 + \frac{\alpha}{24s\left(2sM^2\kappa_\Gamma^2 + \kappa_\Gamma\right)}} - M, \frac{\alpha M}{4-\alpha}, \frac{1}{\nu_*}\right\},$$

where $M = \max\left\{\|\Sigma\|_\infty, \|\Sigma'\|_\infty\right\}$, with Σ and Σ' denoting the true covariance matrices of the two groups, respectively. Assume the following conditions are satisfied for some $r \geq 1$:

(C1) the covariance matrix estimator $\widehat{\Sigma}^m$ meets the tail condition $\mathcal{T}\left(f, \nu_*\right)$,

where the function f is monotonically increasing in both m and δ, and continuous in δ.

(C2) the number of samples m satisfies $m > m_f\left(\bar{\delta}, r\right)$.

(C3) the irrepresentability condition holds, i.e.,

$$\max_{e\in\mathcal{A}^{*c}} \left\|\Gamma^*_{e,\mathbf{A}^*}\left(\Gamma^*_{\mathcal{A}^*,\mathcal{A}^*}\right)^{-1}\right\|_1 < 1.$$

Let the regularization parameter λ be chosen as:

$$\lambda = \max\left\{\frac{2(4-\alpha)(\delta_f+\delta'_f)}{\alpha}, \frac{24sM(2sM^2\kappa_\Gamma^2+\kappa_\Gamma)(\delta_f\delta'_f+M\delta_f+M\delta'_f)}{\alpha}\right\}, \tag{S5.3}$$

where $\alpha = 1 - \max_{e\in\mathcal{A}^{*c}} \left\|\Gamma^*_{e,\mathbf{A}^*}\left(\Gamma^*_{\mathcal{A}^*,\mathcal{A}^*}\right)^{-1}\right\|_1$, $\delta_f = \delta_f(m,r)$, and $\delta'_f = \delta_f(m',r)$. Then, with probability at least $1-d^2r^{-1}$, the support of $\widehat{\Delta}(\lambda)$ lies in the support of Δ^*, and the following error bound holds:

$$\left\|\widehat{\Delta}(\lambda) - \Delta^*\right\|_F \le \sqrt{s}\left\|\widehat{\Delta}(\lambda) - \Delta^*\right\|_\infty \le K\max\left\{\delta_f, \delta'_f\right\},$$

with

$$\begin{aligned}
K =& \sqrt{s}\left\{\kappa_\Gamma + 3s\kappa_\Gamma^2 AB\right\}\left(2 + 4\max\left\{\frac{M}{A}, \frac{6sMB(2sM^2\kappa_\Gamma^2+\kappa_\Gamma)}{\alpha}\right\}\right) \\
&+ 6s^{1.5}M\kappa_\Gamma^2 B, \\
A =& \frac{\alpha M}{4-\alpha}, \\
B =& A + 2M.
\end{aligned}$$

Proof. The result in Lemma 4 follows from Lemma A1 and Equation (A6) in Yuan et al. (2017) with minor modifications. For completeness, we restate the key elements needed to derive the present version of the lemma. Specifically, Equation (A6) in Yuan et al. (2017) states that

$$\left\|\widehat{\Delta}(\lambda) - \Delta^*\right\|_\infty \leq \left\{\kappa_\Gamma + 3s\kappa_\Gamma^2(\delta_f\delta_f' + M\delta_f + M\delta_f')\right\}(\delta_f + \delta_f' + \lambda) + 6sM\kappa_\Gamma^2(\delta_f\delta_f' + M\delta_f + M\delta_f').$$

Immediately after this equation, it is shown that $\delta_f < A$ and $\delta_f' < A$, where $A = \frac{\alpha M}{4-\alpha}$. Clearly, $\delta_f \leq \max\{\delta_f, \delta_f'\}$ and $\delta_f' \leq \max\{\delta_f, \delta_f'\}$. Using the definition of λ in (S5.3) and combining these bounds yields

$$\left\|\widehat{\Delta}(\lambda) - \Delta^*\right\|_\infty \leq \left\{\kappa_\Gamma + 3s\kappa_\Gamma^2 AB\right\}\left(2 + 4\max\left\{\frac{M}{A}, \frac{6sMB(2sM^2\kappa_\Gamma^2 + \kappa_\Gamma)}{\alpha}\right\}\right) \times \max\{\delta_f, \delta_f'\} + 6sM\kappa_\Gamma^2 B\max\{\delta_f, \delta_f'\},$$

where $B = A + 2M$.

Furthermore, Lemma A1(ii) in Yuan et al. (2017) specifies three conditions under which the estimator $\widehat{\Delta}(\lambda)$ has at most s nonzero elements. As shown in the section "Proofs of Theorems 1 and 3" in Yuan et al. (2017), these conditions hold under the setting of the present lemma. Consequently,

$$\left\|\widehat{\Delta}(\lambda) - \Delta^*\right\|_F \leq \sqrt{s}\left\|\widehat{\Delta}(\lambda) - \Delta^*\right\|_\infty.$$

Combining the above bounds yields

$$\left\|\widehat{\Delta}(\lambda) - \Delta^*\right\|_F \leq \sqrt{s}\left\|\widehat{\Delta}(\lambda) - \Delta^*\right\|_\infty \leq K \max\{\delta_f, \delta_f'\},$$

where

$$\begin{aligned} K =& \sqrt{s}\left\{\kappa_\Gamma + 3s\kappa_\Gamma^2 AB\right\}\left(2 + 4\max\left\{\frac{M}{A}, \frac{6sMB(2sM^2\kappa_\Gamma^2 + \kappa_\Gamma)}{\alpha}\right\}\right) \\ &+ 6s^{1.5}M\kappa_\Gamma^2 B. \end{aligned}$$

□

We now proceed to prove Theorem 2 by leveraging Lemma 4. To this end, we first recalibrate the parameters in Lemma 4 to align with our specific setting, and then verify that the conditions stated therein are satisfied. Finally, we reformulate the conclusion of Lemma 4 within our context to establish the desired result.

Based on Lemma 1, our proposed estimator satisfies the tail condition with $\nu_* = \infty$ and the function f is given by

$$f = \exp\left(\frac{m\left(\max\left\{2\delta - \mu, 0\right\}\right)^2}{32\pi^2}\right). \tag{S5.4}$$

In addition, we have:

$$\bar{\delta} = \min\left\{-1 + \sqrt{1 + (6s\kappa_\Gamma)^{-1}},\ -1 + \sqrt{2 + \tilde{M}^{-1}},\ \frac{\alpha}{4 - \alpha}\right\},$$

which follows from the definition of $\bar{\delta}$ in Lemma 4, with the simplification $M = 1$. This simplification is justified by our assumption that the

diagonal entries of the covariance matrices are equal to 1, which implies $\max\left[\|\Sigma\|_\infty, \|\Sigma'\|_\infty\right] \leq 1$. Here, $\tilde{M}$ is defined as $\frac{24s\left(2s\kappa_\Gamma^2+\kappa_\Gamma\right)}{\alpha}$.

Let we define $r = d^\eta$ for some $\eta > 2$. Since we assume $\mu < 2\bar{\delta}$, the function f, as defined in Equation (S5.4), can be simplified, allowing the expression for $m_f\left(\bar{\delta}, d^\eta\right)$ to be computed as follows:

$$m_f\left(\bar{\delta}, r\right) = \frac{32\pi^2\eta\log d}{\left(2\bar{\delta}-\mu\right)^2}, \tag{S5.5}$$

which follows from the definition of $m_f\left(\bar{\delta}, r\right)$ provided in Equation (S5.1).

From Equation (S5.4), it is obvious that the function f, as defined in Equation (S5.4), is monotonically increasing in both m and δ, and continuous with respect to δ. Consequently, it satisfies the condition (C1) in Lemma 4. Furthermore, given the assumption that $m > \frac{32\pi^2\eta\log d}{\left(2\bar{\delta}-\mu\right)^2}$, which implies $m > m_f\left(\bar{\delta}, r\right)$ when incorporating Equation (S5.5), the condition (C2) is also satisfied. Additionally, by assuming the irrepresentability condition holds, as stated in our theorem, condition (C3) is satisfied as well. Therefore, the conclusion of Lemma 4 holds in our setting.

Based on this foundation, we proceed to compute the adopted value for the regularization parameter λ and the quantities δ_f and δ'_f within our

framework. Specifically, we have:

$$\begin{aligned}
\delta_f &= \frac{\tilde{\delta}}{\sqrt{m}} + \frac{\mu}{2}, \\
\delta'_f &= \frac{\tilde{\delta}}{\sqrt{m'}} + \frac{\mu}{2}, \\
\lambda &= \max\left\{\frac{2(\mu + G_A)}{A},\ \tilde{M}\left((G_B + (1+\frac{\mu}{2})(G_A+\mu) + \frac{\mu}{4}\right)\right\},
\end{aligned}$$

where $\tilde{\delta} = \sqrt{8\pi^2 \eta \log d}$, $G_A = \frac{\tilde{\delta}}{\sqrt{m}} + \frac{\tilde{\delta}}{\sqrt{m'}}$ and $G_B = \frac{\tilde{\delta}^2}{\sqrt{mm'}}$. These expressions are derived based on the definition of δ_f, δ'_f and λ, as provided in Equations S5.2 and S5.3, respectively, together with straightforward algebraic manipulations. From the definitions of δ_f and δ'_f, it also follows that $\max\{\delta_f, \delta'_f\} = \frac{\tilde{\delta}}{\sqrt{\min\{m,m'\}}} + \frac{\mu}{2}$. As a result, by applying Lemma 4 to our specific setting, we obtain the result, i.e.,

$$\left\|\widehat{\Delta}(\lambda) - \Delta^*\right\|_F \leq \sqrt{s}\left\|\widehat{\Delta}(\lambda) - \Delta^*\right\|_\infty \leq K\left(\sqrt{\frac{8\pi^2\eta\log d}{\min\{m,m'\}}} + \frac{\mu}{2}\right),$$

with

$$\begin{aligned}
K =& \sqrt{s}\left\{\kappa_\Gamma + 3s\kappa_\Gamma^2 AB\right\}\left(2 + 4\max\left\{\frac{1}{A}, \frac{6sB(2s\kappa_\Gamma^2 + \kappa_\Gamma)}{\alpha}\right\}\right) \\
&+ 6s^{1.5}\kappa_\Gamma^2 B, \\
A =& \frac{\alpha}{4-\alpha}, \\
B =& A + 2.
\end{aligned}$$

After a simple rescaling of constants, the bound can be written in the

more compact form

$$\left\|\widehat{\Delta}(\lambda) - \Delta^*\right\|_F \leq \sqrt{s}\left\|\widehat{\Delta}(\lambda) - \Delta^*\right\|_\infty \leq C\left(\sqrt{\frac{\eta \log d}{\min\{m, m'\}}} + \frac{\mu}{4\pi\sqrt{2}}\right),$$

with

$$\begin{aligned} C =& 4\pi\sqrt{2s}\left(\left\{\kappa_\Gamma + 3s\kappa_\Gamma^2\left(A^2 + 2A\right)\right\}\left(1 + 2\bar{\lambda}\right) + 3s\kappa_\Gamma^2(A+2)\right), \\ A =& \frac{\alpha}{4-\alpha}, \\ \bar{\lambda} =& \max\left\{\frac{1}{A}, \frac{6s(A+2)(2s\kappa_\Gamma^2 + \kappa_\Gamma)}{\alpha}\right\}. \end{aligned}$$

S6 Proof of Theorem 3

The proof is similar to the proof of Theorem 2 presented by Yuan et al. (2017). We know from Theorem 2 that the nonzero elements of $\widehat{\Delta}$ are a subset of the nonzero elements of Δ^*. Moreover, we derive a probabilistic upper bound on the estimation error. In addition, it is assumed that the minimum absolute value of the nonzero elements of Δ^* is greater than twice the obtained probabilistic upper bound in Theorem 2. Consequently, we can deduce the conclusion of Theorem 2.

S7 Derivation of the KKT residuals

We rewrite the regularized optimization problem in (2.1) as

$$\begin{aligned} \min_{D^+, D^- \in \mathbb{R}^{d\times d}} \quad & L_D\big(D^+ - D^-; \widehat{\Sigma}, \widehat{\Sigma}'\big) + \lambda \sum_{i=1}^{d}\sum_{j=1}^{d} \big(D^+_{i,j} + D^-_{i,j}\big) \\ \text{subject to} \quad & D^+_{i,j} \geq 0, \quad D^-_{i,j} \geq 0, \quad \forall i, j \in [d]. \end{aligned}$$

Let Δ^+ and Δ^- denote the optimal solutions corresponding to D^+ and D^-, respectively. The Karush–Kuhn–Tucker (KKT) conditions associated with this problem are as follows.

Primal feasibility.

$$\Delta^+_{i,j} \geq 0, \quad \Delta^-_{i,j} \geq 0, \quad \forall i, j \in [d].$$

Dual feasibility. Let $\Gamma^+_{i,j} \geq 0$ and $\Gamma^-_{i,j} \geq 0$ be the Lagrange multipliers associated with the nonnegativity constraints. Then

$$\Gamma^+_{i,j} \geq 0, \quad \Gamma^-_{i,j} \geq 0, \quad \forall i, j \in [d].$$

Complementary slackness.

$$\Gamma^+_{i,j}\, \Delta^+_{i,j} = 0, \quad \Gamma^-_{i,j}\, \Delta^-_{i,j} = 0, \quad \forall i, j \in [d].$$

Stationarity. The Lagrangian is given by

$$\mathcal{L}(\Delta^+, \Delta^-, \Gamma^+, \Gamma^-) = L_D(\Delta^+ - \Delta^-; \widehat{\Sigma}, \widehat{\Sigma}') + \lambda \sum_{i,j} \left(\Delta^+_{i,j} + \Delta^-_{i,j}\right)$$
$$- \sum_{i,j} \Gamma^+_{i,j}\Delta^+_{i,j} - \sum_{i,j} \Gamma^-_{i,j}\Delta^-_{i,j}.$$

For each $(i,j) \in [d] \times [d]$, the first-order optimality conditions are

$$\frac{\partial \mathcal{L}}{\partial \Delta^+_{i,j}} = \frac{1}{2}\Big(\widehat{\Sigma}_{i,:}(\Delta^+ - \Delta^-)\widehat{\Sigma}'_{:,j} + \widehat{\Sigma}'_{i,:}(\Delta^+ - \Delta^-)\widehat{\Sigma}_{:,j}\Big) - \widehat{\Sigma}_{i,j} + \widehat{\Sigma}'_{i,j} + \lambda - \Gamma^+_{i,j} = 0,$$
$$\frac{\partial \mathcal{L}}{\partial \Delta^-_{i,j}} = -\frac{1}{2}\Big(\widehat{\Sigma}_{i,:}(\Delta^+ - \Delta^-)\widehat{\Sigma}'_{:,j} + \widehat{\Sigma}'_{i,:}(\Delta^+ - \Delta^-)\widehat{\Sigma}_{:,j}\Big) + \widehat{\Sigma}_{i,j} - \widehat{\Sigma}'_{i,j} + \lambda - \Gamma^-_{i,j} = 0.$$

In vectorized form, the stationarity conditions can be written as

$$\begin{cases} \widehat{\Gamma}\left(\vec{\Delta}^+ - \vec{\Delta}^-\right) + \mathbf{v} + \vec{\lambda} - \vec{\Gamma}^+ = \mathbf{0}, \\ -\widehat{\Gamma}\left(\vec{\Delta}^+ - \vec{\Delta}^-\right) - \mathbf{v} + \vec{\lambda} - \vec{\Gamma}^- = \mathbf{0}, \end{cases}$$

where

$$\widehat{\Gamma} = \frac{1}{2}\left(\widehat{\Sigma} \otimes \widehat{\Sigma}' + \widehat{\Sigma}' \otimes \widehat{\Sigma}\right), \quad \mathbf{v} = \mathrm{vec}\Big(\widehat{\Sigma}' - \widehat{\Sigma}\Big),$$

and

$$\vec{\lambda} = \mathrm{vec}(\lambda \mathbf{1}_{d\times d}),$$

with $\mathbf{1}_{d\times d}$ denoting the $d \times d$ matrix whose entries are all equal to one.

Define

$$\mathcal{Z} = \left\{k : \vec{\Delta}^+_k = \vec{\Delta}^-_k = 0\right\}.$$

We partition the index set $\{1, \ldots, d^2\}$ into $\mathcal{Z}$ and its complement $\mathcal{Z}^c$.

Case 1: $k \in \mathcal{Z}$. Combining dual feasibility and stationarity yields

$$\begin{cases} \widehat{\Gamma}_{k,:}\left(\vec{\Delta}^{+} - \vec{\Delta}^{-}\right) + \mathbf{v}_k + \lambda \geq 0, \\ -\widehat{\Gamma}_{k,:}\left(\vec{\Delta}^{+} - \vec{\Delta}^{-}\right) - \mathbf{v}_k + \lambda \geq 0. \end{cases}$$

Equivalently,

$$\left|\widehat{\Gamma}_{k,:}\left(\vec{\Delta}^{+} - \vec{\Delta}^{-}\right) + \mathbf{v}_k\right| \leq \lambda. \tag{S7.1}$$

Case 2: $l \in \mathcal{Z}^c$. At least one of $\vec{\Delta}_l^{+}$ or $\vec{\Delta}_l^{-}$ is strictly positive. By complementary slackness, the corresponding dual variable must be zero. Substituting into the stationarity conditions yields

$$\begin{cases} \widehat{\Gamma}_{l,:}\vec{\Delta} + \mathbf{v}_l + \lambda = 0, & \text{if } \vec{\Delta}_l > 0, \\ -\widehat{\Gamma}_{l,:}\vec{\Delta} - \mathbf{v}_l + \lambda = 0, & \text{if } \vec{\Delta}_l < 0, \end{cases} \tag{S7.2}$$

where $\vec{\Delta} = \vec{\Delta}^{+} - \vec{\Delta}^{-}$.

Combining (S7.1) and (S7.2), the KKT residuals r_i for $i = 1, \ldots, d^2$ can be expressed as

$$r_i = \begin{cases} \left|\widehat{\Gamma}_{i,:}\vec{\Delta} + \mathbf{v}_i + \lambda\,\mathrm{sign}(\vec{\Delta}_i)\right|, & \text{if } \vec{\Delta}_i \neq 0, \\ \max\left\{\left|\widehat{\Gamma}_{i,:}\vec{\Delta} + \mathbf{v}_i\right| - \lambda,\, 0\right\}, & \text{if } \vec{\Delta}_i = 0. \end{cases}$$

S8 Notation and preliminaries

For clarity and to facilitate understanding of the formulas and derivations presented in this supplementary material, we summarize the notations used throughout the manuscript and supplement in Table 1. This table includes key matrices, vectors, sets, and parameters that appear in the main method, the proposed algorithm, and theoretical results.

S9 Expanded simulation studies

This section presents details of three experimental studies evaluating the performance of the proposed method against existing approaches. The first experiment compares the accuracy and computational efficiency of the proposed method with alternative methods. The results demonstrate superior performance in both accuracy and runtime, consistent with the theoretical findings. The second experiment quantifies the differences between exact and approximate optimization. Our exact solver is compared against the iterative APGD algorithm over the same regularization path λ. Performance is assessed via five metrics: (1) the attained objective value, (2) KKT residuals (measuring optimality), (3) structural discrepancies induced by approximation, (4) memory consumption and (5) the minimum computation

time required for APGD to attain a solution without structural error. The results demonstrate that our method attains exact solutions, while APGD produces non-negligible structural errors. Eliminating these errors requires many iterations and substantial computation, especially as λ becomes small. The third experiment investigates the robustness of the proposed method when applied to heterogeneous collection of datasets, as compared with homogeneous datasets. The findings indicate that the method maintains a level of accuracy comparable to that observed in the homogeneous setting, thereby supporting the theoretical guarantees. Detailed descriptions of these experiments are provided below.

S9.1 Simulation setup and data generation

In all experiments, synthetic data are generated by constructing two subject-specific networks. The first network adopts a scale-free structure, which is commonly used to model real-world data accurately (Barabási, 2009). The second network is derived by randomly deleting and inserting 20 edges from the first network. This process allows us to identify the zero entries in the precision matrices for each group. Subsequently, we generate the non-zero entries of the precision matrices by sampling from the distribution $w = \text{sign}(v) + v$, where $v \sim \text{N}(0, 1)$. If the resulting matrices are not

positive definite, we add $(1+\gamma)I$, for a suitable value of γ, to ensure positive definiteness. This process results in two valid precision matrices, Ω and Ω', for the respective subject-specific networks.

After constructing the precision matrices Ω and Ω', we generate synthetic data from multivariate normal distributions. Specifically, for the first network we independently draw m samples from $\mathcal{N}(\mathbf{0}, \Omega^{-1})$, and for the second network we independently draw m' samples from $\mathcal{N}(\mathbf{0}, \Omega'^{-1})$. Here, $m = \sum_{r=1}^{N} m_r$ and $m' = \sum_{r'=1}^{N'} m'_{r'}$ denote the total numbers of samples across N datasets of the first network and N' datasets of the second network, respectively. The samples in each network are then partitioned into disjoint datasets. In the first network, the r-th dataset contains m_r observations, while in the second network the r'-th dataset contains $m'_{r'}$ observations. These datasets share no samples, so each dataset is generated independently.

To estimate the covariance matrices, we use a rank-based estimator described in Section 2.5 of the main manuscript. This estimator is invariant to monotone transformations of the variables in non-paranormal distributions. As a result, the simulation design mimics data generated from non-paranormal models of the form $\mathrm{NPN}(\mathbf{0}, \Omega^{-1}, \mathbf{f}_r)$ and $\mathrm{NPN}(\mathbf{0}, \Omega'^{-1}, \mathbf{f}'_{r'})$, since our differential network inference procedure requires only covariance

matrix estimates for the two groups of observations. Here, each dataset has its own set of monotone transformation functions $\mathbf{f}_r$ or $\mathbf{f}'_{r'}$. Therefore, generating multivariate normal samples and estimating the covariance matrices using the procedure in Section 2.5 of the main manuscript provides a practical way to reproduce both homogeneous and heterogeneous data scenarios considered in our methodology.

S9.2 Experiment 1: competing methods, hyperparameters, additional results

In the first experiment, we evaluate the performance of the proposed solution path method against three established approaches. (i) APGD D-trace (Yuan et al., 2017), implemented using the Dtrace function in the DiffGraph package (Zhang et al., 2018), which applies accelerated proximal gradient descent to obtain an approximate solution to the ℓ_1-penalized D-trace loss minimization problem. (ii) CrossFDTL (Wu et al., 2020), which estimates the differential network through an alternative optimization formulation that directly targets structural changes. (iii) Fused graphical lasso (FGL) (Danaher et al., 2014), a widely used baseline method implemented via the FGL function in DiffGraph (Zhang et al., 2018), which jointly estimates two precision matrices using a fused lasso penalty. Both the proposed method

and CrossFDTL are implemented in C++ using Rcpp (Eddelbuettel et al., 2011).

To assess the algorithm's performance on synthetic data, we employ precision-recall analysis. Let $\widehat{\Delta}$ denote an estimator of the true differential matrix Δ^*. The precision and recall metrics are formally defined as follows:

$$\text{precision} = \frac{\sum_{i<j} \mathbb{1}\{\widehat{\Delta}_{i,j} \neq 0 \ \ \& \ \ \Delta^*_{i,j} \neq 0\}}{\sum_{i<j} \mathbb{1}\{\widehat{\Delta}_{i,j} \neq 0\}},$$
$$\text{recall} = \frac{\sum_{i<j} \mathbb{1}\{\widehat{\Delta}_{i,j} \neq 0 \ \ \& \ \ \Delta^*_{i,j} \neq 0\}}{\sum_{i<j} \mathbb{1}\{\Delta^*_{i,j} \neq 0\}},$$

where $\mathbb{1}\{.\}$ represents the indicator function.

We perform a comparative evaluation of four methods, focusing on precision–recall performance and computational efficiency. This analysis averages results over 100 randomly generated datasets, with dimensionality parameters $d \in \{50, 100\}$ and $m = m' \in \{500, 1000\}$. For both the APGD D-trace and CrossFDTL methods, 50 distinct values for the regularization parameter are sampled from the interval $[0.001, 0.8]$. For the FGL method, 50 values of the regularization parameter λ_2, which controls the sparsity of the differential network, are sampled from $[0.001, 0.2]$. To ensure a fair comparison, the regularization parameter's range is controlled by setting $c = 100$. This specific value is chosen because our experimental observations indicate that the proposed method identifies approximately

50 knots within the solution path, which is comparable to the number of distinct tuning parameter values explored in case of the APGD D-trace and CrossFDTL methods. The CrossFDTL method includes an additional tuning parameter, ρ, which governs the effect of Frobenius norm of $\widehat{\Delta}$ in its objective function. For a fair comparison, ρ is set to 0 in all evaluations. Similarly, the FGL method includes an additional parameter λ_1 that controls the sparsity of the individual group precision matrices. This parameter is also fixed at 0 throughout the experiments.

S9.3 Evaluation under hub-perturbed network structure

To examine the effect of structured changes, we conducted a variant of the first experimental setting in which the differential structure was no longer random. Instead, we introduced a hub-based perturbation pattern. Specifically, a single node was randomly selected, and 20 of its connections to randomly chosen nodes were perturbed. This design simulates a biologically relevant scenario in which structural differences arise from hub nodes. The results of this experiment are presented in Figure 2. The performance under this hub-perturbation setting is similar to that observed in the original experiment. These findings indicate that the proposed method remains robust when structural changes are concentrated around hub nodes.

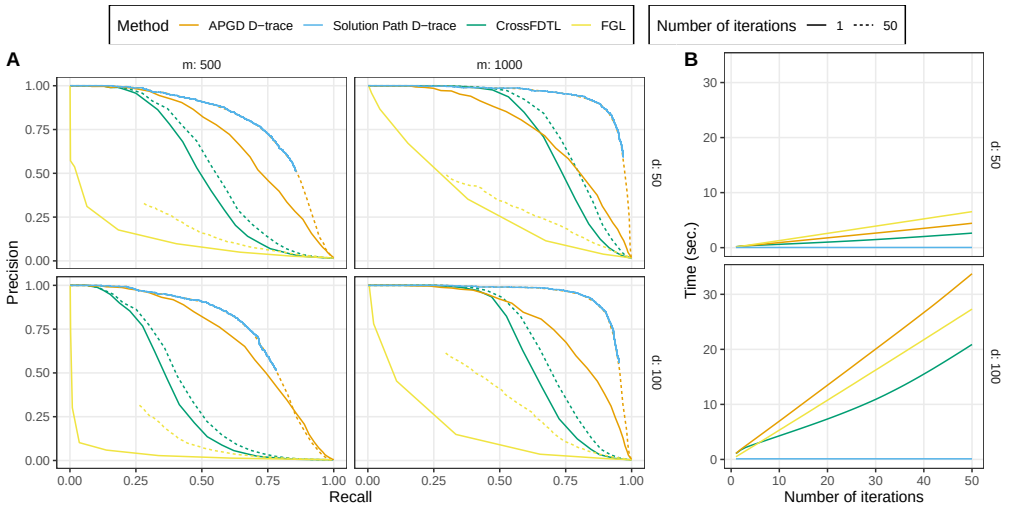

Figure 2: Performance comparison of four differential network estimation methods on synthetic data with hub-perturbed network structures. Line color denotes the method. Data are generated from networks with $d \in \{50, 100\}$ nodes and $m = m' \in \{500, 1000\}$ samples per network. Results are averaged over 100 datasets. (**A**) Precision-recall curves for detecting differential edges. Line type corresponds to the number of iterations used by iterative methods. The proposed method is non-iterative and is therefore shown only with solid lines. (**B**) Computational time for estimating the differential network across all regularization parameters. The x-axis shows the number of iterations of iterative methods and the y-axis shows runtime. The horizontal blue line indicates the average runtime of the proposed method.

S9.4 Experiment 2: exactness, KKT derivations, additional metrics

In this section, we introduce four performance metrics: the objective gap, the Karush-Kuhn-Tucker (KKT) residual, the false selection rate, and the minimum computation time required to obtain a solution without false selections. To formally define these metrics, let $\widehat{\Delta}(\lambda)$ denote the solution obtained by the exact solver (the proposed method), and let $\widehat{\Delta}(\lambda, k)$ denote the estimator produced by an approximate solver (the APGD method) after k iterations. The objective gap is defined as

$$L_D(\widehat{\Delta}(\lambda,k);\widehat{\Sigma},\widehat{\Sigma}') + \lambda\|\widehat{\Delta}(\lambda,k)\|_1 - \Big(L_D(\widehat{\Delta}(\lambda);\widehat{\Sigma},\widehat{\Sigma}') + \lambda\|\widehat{\Delta}(\lambda)\|_1\Big),$$

which quantifies the difference between the objective value achieved by the approximate solver and that attained by the exact solver.

We assess optimality via the KKT residual, defined as the magnitude of violation of the optimality conditions,

$$\begin{cases} \Gamma_{i,:}\vec{\Delta}(\lambda) + \mathbf{v}_i = -\lambda\,\mathrm{sign}\Big(\vec{\Delta}_i(\lambda)\Big), & \text{if } \vec{\Delta}_i(\lambda) \neq 0, \\ \Big|\Gamma_{i,:}\vec{\Delta}(\lambda) + \mathbf{v}_i\Big| \leq \lambda, & \text{if } \vec{\Delta}_i(\lambda) = 0. \end{cases} \tag{S9.1}$$

Here, $\vec{\Delta}(\lambda)$ denotes the vectorized form of either $\widehat{\Delta}(\lambda)$ or $\widehat{\Delta}(\lambda, k)$. The total KKT residual is defined as the sum of all individual residual magnitudes. The derivation of these optimality conditions is provided in Section S7.

Finally, we define the false selection rate as

$$\frac{|\mathcal{F}_{\mathrm{p}} \cup \mathcal{F}_{\mathrm{n}}|}{|\mathcal{E}|},$$

where

$$\mathcal{F}_{\mathrm{n}} = \{(i,j) : \widehat{\Delta}_{i,j}(\lambda) \neq 0 \quad \& \quad \widehat{\Delta}_{i,j}(\lambda, k) = 0\},$$

$$\mathcal{F}_{\mathrm{p}} = \{(i,j) : \widehat{\Delta}_{i,j}(\lambda) = 0 \quad \& \quad \widehat{\Delta}_{i,j}(\lambda, k) \neq 0\},$$

and

$$\mathcal{E} = \{(i,j) : \widehat{\Delta}_{i,j}(\lambda) \neq 0\}.$$

S9.5 Supplementary information for experiment 2

In the main body of the manuscript, we introduce the second experiment, in which we illustrate the elimination of approximation error achieved by the proposed method and examine the effect of approximation on feature selection accuracy. In this section, we provide an additional evaluation of this experiment by assessing computational resource usage. Specifically, we compare the proposed method with the APGD method in terms of memory consumption and computation time. Figure 3 summarizes these comparisons. Panel (A) of Figure 3 reports the memory usage of both methods. For the proposed method, labeled "Solution Path (Exact)" in the figure, memory usage reflects the resources required to compute the entire solution path.

For the APGD method, labeled "APGD (Approximate)", memory usage is measured on a per-iteration basis for each value of the regularization parameter. As expected, the per-iteration memory consumption of the APGD method remains approximately constant across different regularization values. However, the memory required to compute the full solution path using the proposed method is substantially lower than the memory required for a single iteration of the APGD method at a given regularization value. Panel (B) of Figure 3 reports computation time for both methods. For the proposed method, the reported time is the total time required to compute the complete solution path. For the APGD method, the reported time is the minimum time required, for each regularization value, to obtain a solution with no false positives or false negatives relative to the exact solution of the optimization problem. The results show that the computation time of the APGD method increases rapidly as the regularization parameter decreases. Across a range of regularization values considered, the total time required by the proposed method to compute the entire solution path is lower than the time required by the APGD method to obtain an accurate solution for a single regularization value, particularly for small values of λ.

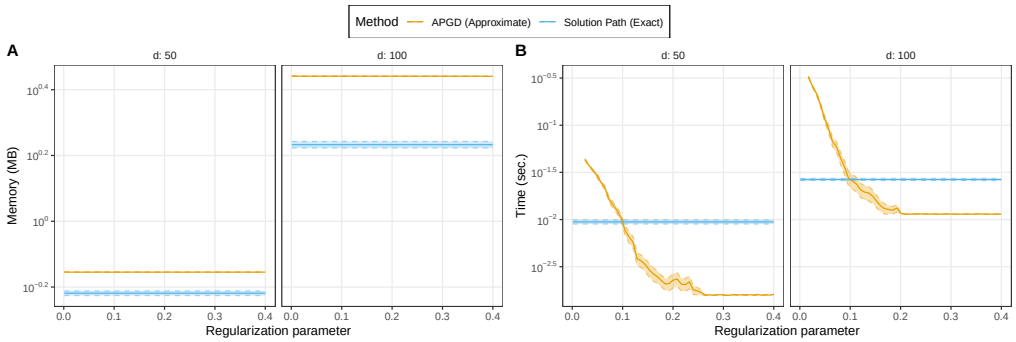

Figure 3: Comparison of computational resource usage between the proposed solution path method and the APGD-based approximate method across regularization parameters and problem dimensions. Panel (**A**) reports memory consumption as a function of the regularization parameter λ. For the proposed method, memory usage corresponds to the resources required to compute the entire solution path, whereas for the APGD method it is measured per iteration for each value of λ. Panel (**B**) reports computation time on a logarithmic scale. The reported time for the proposed method corresponds to the total time required to compute the full solution path, while for the APGD method it represents the minimum time required, for each λ, to obtain a solution without false positives or false negatives. Shaded regions indicate $\pm$ one standard error around the mean. Results are shown for different problem dimensions d.

S10 Real data applications details

S10.1 Workflow

Our proposed method produces a solution path—a collection of candidate solutions corresponding to different values of the regularization parameter.

Consequently, selecting an appropriate solution from this collection is a critical step. To address this, we adopt a stability-based model selection approach inspired by the StARS method (Liu et al., 2010). Specifically, we generate random subsamples comprising 80 percent of the data from each group and apply our method to each subsample to obtain a solution path. This procedure is repeated 10 times, and the instability of the inferred networks is computed across different regularization parameter values. By setting the instability threshold to 0.001, we determine the optimal regularization parameter and, subsequently, the final differential network inferred from the full dataset.

S10.2 Ovarian cancer: analysis details

We analyze gene expression data from ovarian cancer cases available in The Cancer Genome Atlas (TCGA) (Network et al., 2011), comprising two groups categorized by drug response according to the criterion proposed by Nabavi et al. (2016). Specifically, we select 242 platinum-sensitive and 98 platinum-resistant tumors. Genes expression for each case was measured using three different platforms, resulting in three datasets per group. To enhance computational efficiency and facilitate model interpretability, we restrict our analysis to 551 genes associated with relevant biological path-

ways, as proposed by Dilruba and Kalayda (2016) and Burris (2013).

Figure 4 illustrates the inferred differential network, comprising 77 edges among 127 genes. Notably, we identified several high-degree genes in the network for which strong literature support exists, including CDKN1A (Xia et al., 2011; Lincet et al., 2000; Koster et al., 2010), MAP2K1 (Pénzváltó et al., 2014), MNAT1 (Qiu et al., 2020), and PRKCA (Mohanty et al., 2005; Arrighetti et al., 2016; Basu and Weixel, 1995).

To systematically assess the biological relevance of the inferred differential network, we performed pathway and transcription factor regulon enrichment analyses using over-representation analysis (Khatri et al., 2012), with the 551 candidate genes serving as the background universe. P-values were adjusted using the Benjamini-Hochberg procedure (Benjamini and Hochberg, 2018). Reactome pathway analysis (Gillespie et al., 2022) identified significant enrichment of transcription-related pathways, including Gene expression (Transcription) (54 of 120 genes; adjusted $p = 0.018$), Generic Transcription Pathway (53 of 120 genes; adjusted $p = 0.018$), and RNA Polymerase II Transcription (53 of 120 genes; adjusted $p = 0.018$). To further characterize these findings, we conducted transcription factor regulon enrichment analysis using the DoRothEA database (Garcia-Alonso et al., 2019). Among all tested regulators, only the FOXM1 regulon showed

significant enrichment (12 of 87 genes; adjusted $p = 0.014$). FOXM1 is a well-established transcription factor involved in cell cycle regulation, and recent studies have implicated it in cancer drug sensitivity (Koo et al., 2012). These results support the biological plausibility of the detected structural differences.

To further evaluate the biological relevance of our findings, we compared the inferred differential network with a reference set of 912 genes previously implicated in platinum resistance in cancer (Huang et al., 2021). The genes in the differential network were classified into two categories: (1) non-isolated genes, which participate in at least one interaction, and (2) connector genes, which engage in more than one interaction (degree greater than one). Fisher's exact tests revealed significant enrichment of platinum resistance–associated genes in both categories, with p-values of 0.01 for non-isolated genes and 0.003 for connector genes. These results indicate that the genes identified by our method are closely linked to established platinum resistance markers, thereby providing additional evidence for the biological validity of the inferred differential network.

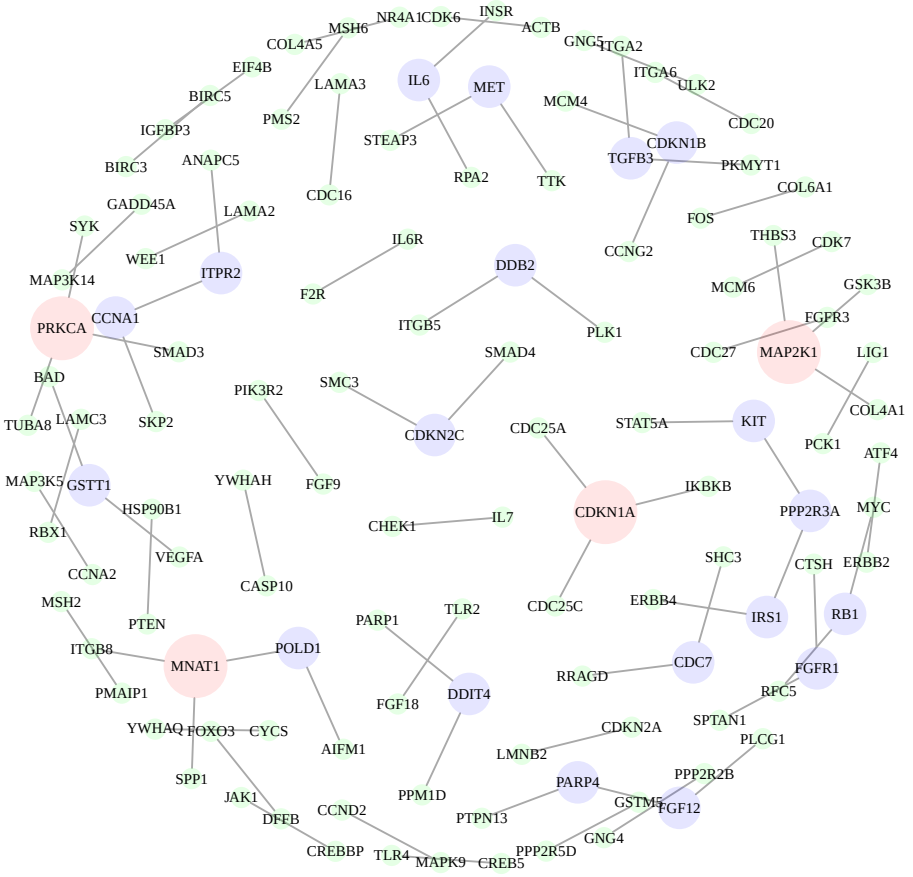

Figure 4: Differential gene network between platinum-sensitive (n=242) and platinum-resistant (n=98) ovarian tumors from TCGA, restricted to 551 genes in relevant biological pathways. The network was estimated using the proposed method, with the regularization parameter selected via StARS (instability threshold = 0.001, 10 random 80% subsamples per group). Nodes represent genes, with size and color proportional to degree.

S10.3 Breast cancer: analysis details

Breast cancer remains one of the leading causes of cancer-related mortality among women worldwide and comprises four principal molecular subtypes: luminal A, luminal B, basal-like, and HER2-enriched (Network et al., 2012). These subtypes exhibit distinct clinical outcomes and genomic characteristics (Schnitt, 2010). Characterizing differences in gene regulatory architecture across molecular subtypes may provide insight into the biological mechanisms underlying subtype heterogeneity. In this study, we estimate differential gene networks between the luminal A and basal-like subtypes using Breast Invasive Carcinoma (BRCA) data obtained from The Cancer Genome Atlas (TCGA) (Network et al., 2012) through the Genomic Data Commons via the curatedTCGAData (v2.1.1) Bioconductor package (Ramos et al., 2020). Both RNASeqGene and mRNAArray assays were included. The RNA-seq dataset contains 247 luminal A samples and 99 basal-like samples, while the microarray dataset contains 264 luminal A samples and 106 basal-like samples. In this study, we restricted the analysis to 145 genes from the breast cancer pathway (hsa05224) obtained from the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto, 2000).

We applied the proposed method to the breast cancer dataset following

the same procedure described in the ovarian cancer application (Section S10.1). The inferred differential network is shown in Figure 5, comprising 14 edges among 17 genes. Because no ground-truth differential network between luminal A and basal-like subtypes is available, direct evaluation of estimation accuracy is not feasible. Instead, we assess the biological plausibility of the estimated network through functional relevance of the identified genes, following the strategy suggested by Tu et al. (2021). In Tu et al. (2021), six key genes were highlighted based on strong literature support and their analytical findings. These genes include IGF1R, CDK6, ESR1, CCND1, RB1, and LEF1. All except LEF1 recovered in our estimated differential network, indicating substantial agreement between our results and previously reported biologically meaningful findings. Notably, ESR1 is also identified as a hub in our model.

To further evaluate biological relevance, we conducted over-representation analysis using the MSigDB C2 curated gene sets (Subramanian et al., 2005; Liberzon et al., 2011, 2015). The C2 collection comprises manually curated gene sets derived from pathway databases, published studies, and domain experts, and is widely used for hypothesis-driven functional interpretation of gene lists. The 145 candidate genes considered in the network analysis were used as the background universe, and p-values

were adjusted using the Benjamini–Hochberg procedure (Benjamini and Hochberg, 2018). The analysis revealed significant enrichment for the gene set SMID_BREAST_CANCER_BASAL_UP (6 of 17 genes; adjusted p-value = 0.024), which consists of genes upregulated in basal-like breast cancer. This enrichment indicates that a substantial proportion of the genes identified by our method are associated with transcriptional programs characteristic of the basal-like subtype. Given that our analysis contrasts luminal A and basal-like tumors, the enrichment of a basal-upregulated signature provides independent biological validation that the inferred differential network captures meaningful subtype-specific molecular differences.

References

Arrighetti, N., G. Cossa, L. De Cecco, S. Stucchi, N. Carenini, E. Corna, P. Gandellini, N. Zaffaroni, P. Perego, and L. Gatti (2016). Pkc-alpha modulation by mir-483-3p in platinum-resistant ovarian carcinoma cells. *Toxicology and applied pharmacology 310*, 9–19.

Barabási, A.-L. (2009). Scale-free networks: a decade and beyond. *science 325*(5939), 412–413.

Basu, A. and K. M. Weixel (1995). Comparison of protein kinase c activity and isoform expression in cisplatin-sensitive and-resistant ovarian carcinoma cells. *International journal of cancer 62*(4), 457–460.

Beck, A. (2017). *Chapter 3: Subgradients*, pp. 35–86. Philadelphia, PA: Society for Industrial

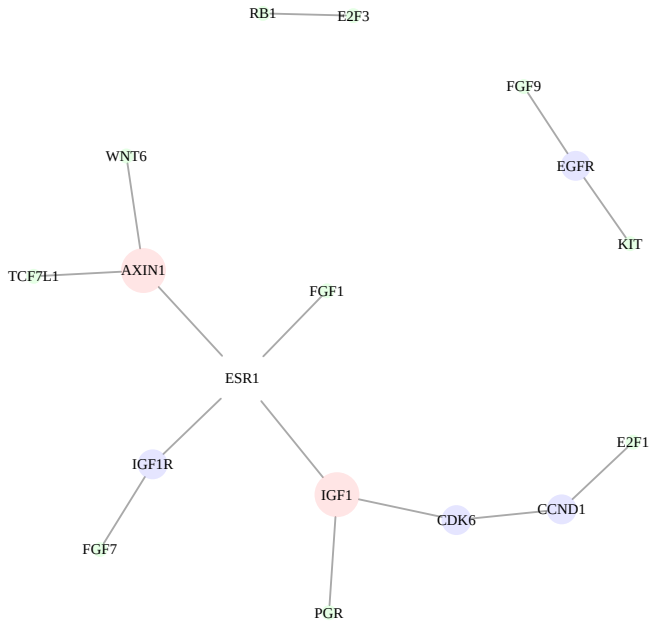

Figure 5: Differential gene network between luminal A (n=247 RNA-seq, 264 microarray) and basal-like (n=99 RNA-seq, 106 microarray) breast cancer subtypes from TCGA, restricted to 145 genes in the KEGG breast cancer pathway (hsa05224). The network was estimated using the proposed solution path method, with the regularization parameter selected via StARS (10 random 80% subsamples, instability threshold = 0.001). Nodes represent genes, with size and color proportional to degree.

and Applied Mathematics.

Benjamini, Y. and Y. Hochberg (2018, 12). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological) 57*(1), 289–300.

REFERENCES

Burris, H. A. (2013). Overcoming acquired resistance to anticancer therapy: focus on the pi3k/akt/mtor pathway. *Cancer chemotherapy and pharmacology 71*(4), 829–842.

Danaher, P., P. Wang, and D. M. Witten (2014). The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society. Series B, Statistical methodology 76*(2), 373.

Dilruba, S. and G. V. Kalayda (2016). Platinum-based drugs: past, present and future. *Cancer chemotherapy and pharmacology 77*(6), 1103–1124.

Eddelbuettel, D., R. François, J. Allaire, K. Ushey, Q. Kou, N. Russel, J. Chambers, and D. Bates (2011). Rcpp: Seamless r and c++ integration. *Journal of statistical software 40*(8), 1–18.

Garcia-Alonso, L., C. H. Holland, M. M. Ibrahim, D. Turei, and J. Saez-Rodriguez (2019). Benchmark and integration of resources for the estimation of human transcription factor activities. *Genome research 29*(8), 1363–1375.

Gillespie, M., B. Jassal, R. Stephan, M. Milacic, K. Rothfels, A. Senff-Ribeiro, J. Griss, C. Sevilla, L. Matthews, C. Gong, C. Deng, T. M. Varusai, E. Ragueneau, Y. Haider, B. May, V. Shamovsky, J. Weiser, T. Brunson, N. Sanati, L. Beckman, X. Shao, A. Fabregat, K. Sidiropoulos, J. Murillo, G. Viteri, J. Cook, S. Shorser, G. D. Bader, E. Demir, C. Sander, R. Haw, G. Wu, L. Stein, H. Hermjakob, and P. D'Eustachio (2022). The reactome pathway knowledgebase 2022. *Nucleic Acids Res. 50*(D1), 687–692.

Huang, D., S. R. Savage, A. P. Calinawan, C. Lin, B. Zhang, P. Wang, T. K. Starr, M. J.

Birrer, and A. G. Paulovich (2021). A highly annotated database of genes associated with platinum resistance in cancer. *Oncogene 40*(46), 6395–6405.

Kanehisa, M. and S. Goto (2000). Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research 28*(1), 27–30.

Kendall, M. G. (1948). *Rank correlation methods.* Griffin.

Khatri, P., M. Sirota, and A. J. Butte (2012). Ten years of pathway analysis: Current approaches and outstanding challenges. *PLoS Comput. Biol. 8*(2).

Koo, C.-Y., K. W. Muir, and E. W.-F. Lam (2012). Foxm1: From cancer initiation to progression and treatment. *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms 1819*(1), 28–37.

Koster, R., A. Di Pietro, H. Timmer-Bosscha, J. H. Gibcus, A. Van Den Berg, A. J. Suurmeijer, R. Bischoff, J. A. Gietema, S. De Jong, et al. (2010). Cytoplasmic p21 expression levels determine cisplatin resistance in human testicular cancer. *The Journal of clinical investigation 120*(10), 3594–3605.

Kruskal, W. H. (1958). Ordinal measures of association. *Journal of the American Statistical Association 53*(284), 814–861.

Liberzon, A., C. Birger, H. Thorvaldsdóttir, M. Ghandi, J. P. Mesirov, and P. Tamayo (2015). The molecular signatures database hallmark gene set collection. *Cell systems 1*(6), 417–425.

Liberzon, A., A. Subramanian, R. Pinchback, H. Thorvaldsdóttir, P. Tamayo, and J. P. Mesirov

(2011). Molecular signatures database (msigdb) 3.0. *Bioinformatics 27*(12), 1739–1740.

Lincet, H., L. Poulain, J.-S. Remy, E. Deslandes, F. Duigou, P. Gauduchon, and C. Staedel (2000). The p21cip1/waf1 cyclin-dependent kinase inhibitor enhances the cytotoxic effect of cisplatin in human ovarian carcinoma cells. *Cancer letters 161*(1), 17–26.

Liu, H., F. Han, M. Yuan, J. Lafferty, L. Wasserman, et al. (2012). High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics 40*(4), 2293–2326.

Liu, H., K. Roeder, and L. Wasserman (2010). Stability approach to regularization selection (stars) for high dimensional graphical models. *Advances in neural information processing systems 24*(2), 1432.

McDiarmid, C. et al. (1989). On the method of bounded differences. *Surveys in combinatorics 141*(1), 148–188.

Mohanty, S., J. Huang, and A. Basu (2005). Enhancement of cisplatin sensitivity of cisplatin-resistant human cervical carcinoma cells by bryostatin 1. *Clinical cancer research 11*(18), 6730–6737.

Nabavi, S., D. Schmolze, M. Maitituoheti, S. Malladi, and A. H. Beck (2016). Emdomics: a robust and powerful method for the identification of genes differentially expressed between heterogeneous classes. *Bioinformatics 32*(4), 533–541.

Network, C. G. A. et al. (2012). Comprehensive molecular portraits of human breast tumors. *Nature 490*(7418), 61.

Network, C. G. A. R. et al. (2011). Integrated genomic analyses of ovarian carcinoma. *Na-*

ture 474(7353), 609.

Pénzváltó, Z., A. Lánczky, J. Lénárt, N. Meggyesházi, T. Krenács, N. Szoboszlai, C. Denkert, I. Pete, and B. Győrffy (2014). Mek1 is associated with carboplatin resistance and is a prognostic biomarker in epithelial ovarian cancer. *BMC cancer 14*(1), 1–10.

Qiu, C., W. Su, N. Shen, X. Qi, X. Wu, K. Wang, L. Li, Z. Guo, H. Tao, G. Wang, et al. (2020). Mnat1 promotes proliferation and the chemo-resistance of osteosarcoma cell to cisplatin through regulating pi3k/akt/mtor pathway. *BMC cancer 20*(1), 1–12.

Ramos, M., L. Geistlinger, S. Oh, L. Schiffer, R. Azhar, H. Kodali, I. de Bruijn, J. Gao, V. J. Carey, M. Morgan, and L. Waldron (2020). Multiomic integration of public oncology databases in bioconductor. *JCO Clinical Cancer Informatics 1*(4), 958–971. PMID: 33119407.

Rosset, S. and J. Zhu (2007). Piecewise linear regularized solution paths. *The Annals of Statistics*, 1012–1030.

Schnitt, S. J. (2010). Classification and prognosis of invasive breast cancer: from morphology to molecular taxonomy. *Modern pathology 23*, S60–S64.

Still, G. (2018). Lectures on parametric optimization: An introduction. *Optimization Online*, 2.

Subramanian, A., P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression

profiles. *Proceedings of the national academy of sciences 102*(43), 15545–15550.

Tu, J.-J., L. Ou-Yang, Y. Zhu, H. Yan, H. Qin, and X.-F. Zhang (2021). Differential network analysis by simultaneously considering changes in gene interactions and gene expression. *Bioinformatics 37*(23), 4414–4423.

Wu, Y., T. Li, X. Liu, and L. Chen (2020). Differential network inference via the fused d-trace loss with cross variables. *Electronic Journal of Statistics 14*(1), 1269–1301.

Xia, X., Q. Ma, X. Li, T. Ji, P. Chen, H. Xu, K. Li, Y. Fang, D. Weng, Y. Weng, et al. (2011). Cytoplasmic p21 is a potential predictor for cisplatin sensitivity in ovarian cancer. *BMC cancer 11*(1), 1–9.

Yuan, H., R. Xi, C. Chen, and M. Deng (2017). Differential network analysis via lasso penalized d-trace loss. *Biometrika 104*(4), 755–770.

Zhang, X.-F., L. Ou-Yang, S. Yang, X. Hu, and H. Yan (2018). Diffgraph: an r package for identifying gene network rewiring using differential graphical models. *Bioinformatics 34*(9), 1571–1573.

Zhao, B., Y. S. Wang, and M. Kolar (2022). Fudge: A method to estimate a functional differential graph in a high-dimensional setting. *Journal of Machine Learning Research 23*(82), 1–82.

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

Zhao, T., K. Roeder, and H. Liu (2014). Positive semidefinite rank-based correlation matrix

estimation with application to semiparametric graph estimation. *Journal of Computational and Graphical Statistics 23*(4), 895–922.

Table 1: Notation used in the manuscript and supplementary material.

Symbol	Definition
d	Number of variables (dimensions) in the graphical model
$[d]$	The set $\{1, 2, \ldots, d\}$
$\mathbf{x} = [x_1, \ldots, x_d]^\intercal$	d-dimensional random vector
Σ, Σ'	Covariance matrices of two groups / states
$\Delta^* = \Sigma^{-1} - (\Sigma')^{-1}$	True differential network
$\mathcal{A}^* = \{(i,j) : \Delta^*_{i,j} \neq 0\}$	Support of the differential network
$s = \lvert\mathcal{A}^*\rvert$	Sparsity of the differential network
$\widehat{\Sigma}, \widehat{\Sigma}'$	Estimated covariance matrices (PSD)
$\tilde{\Sigma}$	Initial correlation matrix estimate via weighted Kendall's tau
$\widehat{\Sigma}(\mu)$	PSD-projected covariance estimate
$L_D(\Delta; \widehat{\Sigma}, \widehat{\Sigma}')$	D-trace loss function
λ	Regularization parameter in the lasso-penalized D-trace loss
$\widehat{\Delta}(\lambda)$	Solution of the lasso-penalized D-trace optimization problem
$\Gamma = (\widehat{\Sigma} \otimes \widehat{\Sigma}' + \widehat{\Sigma}' \otimes \widehat{\Sigma})/2$	Hessian matrix (Kronecker form)
$\mathbf{v} = \mathrm{vec}(\widehat{\Sigma} - \widehat{\Sigma}')$	Vectorized difference of covariance matrices
$\mathcal{A}_t$	Active set at segment t along the solution path
$\mathbf{s}_t$	Sign pattern of active set at segment t
λ_t	Knot point (regularization value) at segment t
c	Active-set bound controlling maximum cardinality of $\mathcal{A}_t$
N, N'	Number of heterogeneous datasets in each group
m_r	Number of samples in dataset $\mathcal{D}_r$
$\tau^{(r)}_{k,l}, \widehat{\tau}^{(w)}_{k,l}$	Kendall's tau (dataset r / weighted across datasets)