Journal of Machine Learning Research 23 (2022) 1-39 Submitted 5/21; Revised 11/21; Published 3/22

EiGLasso for Scalable Sparse Kronecker-Sum Inverse Covariance Estimation

Jun Ho Yoon JUNHOY@CS.CMU.EDU

Seyoung Kim SSSYKIM@CS.CMU.EDU

Computational Biology Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Editor: Daniela Witten

Abstract

In many real-world data, complex dependencies are present both among samples and among features. The Kronecker sum or the Cartesian product of two graphs, each modeling dependencies across features and across samples, has been used as an inverse covariance matrix for a matrix-variate Gaussian distribution as an alternative to Kronecker-product inverse covariance matrix due to its more intuitive sparse structure. However, the existing methods for sparse Kronecker-sum inverse covariance estimation are limited in that they do not scale to more than a few hundred features and samples and that unidentifiable parameters pose challenges in estimation. In this paper, we introduce EiGLasso, a highly scalable method for sparse Kronecker-sum inverse covariance estimation, based on Newton's method combined with eigendecomposition of the sample and feature graphs to exploit the Kronecker-sum structure. EiGLasso further reduces computation time by approximating the Hessian matrix, based on the eigendecomposition of the two graphs. EiGLasso achieves quadratic convergence with the exact Hessian and linear convergence with the approximate Hessian. We describe a simple new approach to estimating the unidentifiable parameters that generalizes the existing methods. On simulated and real-world data, we demonstrate that EiGLasso achieves two to three orders-of-magnitude speed-up, compared to the existing methods.

Keywords: Kronecker sum, sparse inverse covariance estimation, Newton's method, convex optimization, L_1 regularization.

1. Introduction

In many real-world datasets, complex dependencies are found among samples as well as among features. For example, in tumor gene-expression data collected from cancer patients, gene-expression levels are correlated both among genes in the same pathways and among patients with the same or similar cancer subtypes (Dai et al., 2015). Other examples include multivariate time-series data such as temporally correlated stock prices of companies in the same sector (King, 1966) or a sequence of images in video data (Kalchbrenner et al., 2017). Gaussian graphical models with L_1-regularization have been widely used to learn a sparse inverse covariance matrix that corresponds to a graph over features (Friedman et al., 2007; Hsieh et al., 2014, 2013), but they were limited in that they did not model correlation among samples.

As an alternative, a matrix-variate Gaussian distribution has been introduced, where the dependencies across samples and across features were modeled with two separate graphs and were combined through either Kronecker-product or Kronecker-sum operators to form an inverse covariance matrix. The Kronecker product of the two graphs led to a hard-to-interpret dense graph and a non-convex log-likelihood of data (Leng and Tang, 2012; Yin and Li, 2012; Tsiligkaridis and Hero, 2013; Zhou, 2014), but its bi-convexity provided a fast flip-flop optimization method. In contrast, the Kronecker sum had the advantage of producing a sparse inverse covariance matrix from the Cartesian product of the two graphs and having a convex log-likelihood, but the existing optimization methods, such as BiGLasso and TeraLasso, did not scale to large datasets (Kalaitzis et al., 2013; Greenewald et al., 2019). BiGLasso used GLasso as a subroutine to estimate one of the two graphs while fixing the other in each iteration (Kalaitzis et al., 2013). TeraLasso significantly improved the scalability of BiGLasso with the gradient descent method (Greenewald et al., 2019), but still did not scale to problems with more than a few hundred samples and features.

Another main challenge with the Kronecker sum arises from the unidentifiable parameters in the diagonals of the matrices representing feature and sample graphs. BiGLasso did not estimate these unidentifiable parameters. TeraLasso employed a reparameterization to make the parameters identifiable and projected the parameter estimates to the reparameterized space in each iteration during optimization. In other works, where the Kronecker-sum inverse covariance matrix was used to model errors in covariates in regression models, the trace of one of the two graphs was assumed to be known (Rudelson and Zhou, 2017; Park et al., 2017; Zhang, 2020). With the Kronecker product, the parameters are unidentifiable as well, but they can be identified with a simple method after the optimization is complete (Yin and Li, 2012).

In this paper, to address these limitations, we introduce eigen graphical Lasso (EiGLasso) for efficient estimation of the Kronecker-sum inverse covariance matrix while handling the unidentifiable parameters. For efficient optimization, we extend the Newton's method in QUIC for sparse inverse covariance estimation (Hsieh et al., 2013, 2014). A naive application of QUIC to our problem would fail from expensive computation time and high memory cost, since the gradient and Hessian matrices in Kronecker-sum inverse covariance estimation are far larger with an inflated structure than in single inverse covariance estimation. To reduce time and memory cost, we use the eigendecomposition of the parameters to deflate the inflated structures in the gradient and Hessian matrices. Based on this eigendecomposition, we develop a strategy for approximating the Hessian to further improve the scalability. In our experiments, we show that compared to the previous methods, EiGLasso with the approximate Hessian achieves two to three orders-of-magnitude speed-up. In addition, we prove that the unidentifiable parameters are uniquely determined given the ratio of the traces of the two graph matrices. We show that the parameters can be optimized in the original space rather than in the reparameterized space as in TeraLasso, and can be identified once after convergence rather than in every iteration as in TeraLasso.

Our preliminary work on EiGLasso appeared in Yoon and Kim (2020). EiGLasso in our preliminary work used the flip-flop optimization of two graph parameters with no theoretical guarantee in convergence. In this paper, we optimize both of the graph parameters jointly in each iteration, as this allows us to study the properties of the exact and approximate Hessian matrices and to analyze the convergence behavior of EiGLasso using these properties. Specifically, we show that the exact Hessian is positive semi-definite but positive definite in the Kronecker-sum space, since the Kronecker-sum space exists outside of the null space of the Hessian, while the approximate Hessian is positive definite everywhere. We use these properties to extend the theoretical results on the convergence of QUIC and show that EiGLasso achieves quadratic convergence with the exact Hessian and linear convergence with the approximate Hessian. We present more extensive experimental results to demonstrate the performance of EiGLasso and to provide insights into its convergence behavior.

The rest of the paper is organized as follows. In Section 2, we review the previous work on statistical methods with the Kronecker-sum operator. We introduce our EiGLasso optimization in Section 3, study its convergence behavior in Section 4, present experimental results in Section 5, and conclude with future work in Section 6.

2. Related Work

A Gaussian distribution of a matrix random variable $\boldsymbol{Y} \in \mathbb{R}^{q\times p}$ for q samples and p features with a Kronecker-sum inverse covariance matrix (Kalaitzis et al., 2013) is given as

$$\mathrm{vec}(\boldsymbol{Y}) \sim \mathcal{N}\left(\mathrm{vec}(\boldsymbol{M}), \boldsymbol{\Omega}^{-1}\right), \tag{1}$$

where $\boldsymbol{M} \in \mathbb{R}^{q\times p}$ is the mean and $\mathrm{vec}(\cdot)$ is an operator that stacks the columns of a matrix into a vector. The $pq \times pq$ inverse covariance matrix $\boldsymbol{\Omega}$ in Eq. (1) is defined as the Kronecker-sum of two graphs, a $p \times p$ matrix $\boldsymbol{\Theta}$ for dependencies across features and a $q \times q$ matrix $\boldsymbol{\Psi}$ for dependencies across samples:

$$\boldsymbol{\Omega} = \boldsymbol{\Theta} \oplus \boldsymbol{\Psi} = \boldsymbol{\Theta} \otimes \boldsymbol{I}_q + \boldsymbol{I}_p \otimes \boldsymbol{\Psi},$$

where $\otimes$ is the Kronecker-product operator and $\boldsymbol{I}_a$ is an $a \times a$ identity matrix. A non-zero value in the (i, j)th element $[\boldsymbol{\Theta}]_{ij}$, $[\boldsymbol{\Psi}]_{ij}$, and $[\boldsymbol{\Omega}]_{ij}$ implies an edge between the ith and jth nodes in the corresponding graph. The two graphs $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ are constrained to form a positive-definite Kronecker-sum space as follows:

$$\mathbb{KS}^{p,q} = \{(\boldsymbol{\Theta}, \boldsymbol{\Psi}) | \boldsymbol{\Theta} \in \mathbb{S}^p, \boldsymbol{\Psi} \in \mathbb{S}^q, \boldsymbol{\Omega} = \boldsymbol{\Theta} \oplus \boldsymbol{\Psi} \in \mathbb{S}^{pq}_{++}\}, \tag{2}$$

where $\mathbb{S}^a$ denotes the set of all $a \times a$ symmetric matrices and $\mathbb{S}^a_{++}$ the set of all $a \times a$ positive definite matrices. Then, $\boldsymbol{\Omega}$ models a graph over pq nodes, where each node is associated with an observation for the given sample and feature pair. For the rest of this paper, we assume zero mean $\mathrm{vec}(\boldsymbol{M}) = \boldsymbol{0}$, and focus on the inverse covariance matrix $\boldsymbol{\Omega}$.

Compared to the Kronecker product $\boldsymbol{\Theta} \otimes \boldsymbol{\Psi}$, one main advantage of the Kronecker sum is that as the Cartesian product of the two graphs $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$, it leads to a more intuitively appealing sparse structure in the graph $\boldsymbol{\Omega}$ (Figure 1(a); Figure 1 in Kalaitzis et al. (2013)). In $\boldsymbol{\Omega}$ with the Kronecker sum, the sample graph $\boldsymbol{\Psi}$ is used only within the same feature (diagonal blocks in $\boldsymbol{\Omega}$ in Figure 1(a)) and the feature graph $\boldsymbol{\Theta}$ is used only within the same sample (diagonals of off-diagonal blocks in $\boldsymbol{\Omega}$ in Figure 1(a)), whereas the Kronecker-product leads to a dense graph, where the sample graph $\boldsymbol{\Psi}$ and the feature graph $\boldsymbol{\Theta}$ are used across features and samples. Because of the sparse structure, the Kronecker-sum operator has been

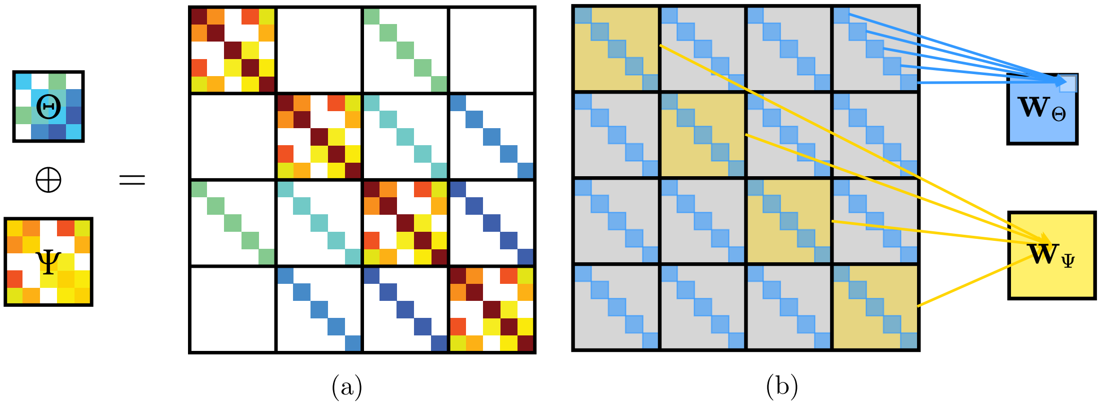


Figure 1: Illustration of the Kronecker-sum inverse covariance matrix and gradient computation in EiGLasso. (a) $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ (left) and their Kronecker sum $\boldsymbol{\Omega} = \boldsymbol{\Theta} \oplus \boldsymbol{\Psi}$ (right). $\boldsymbol{\Theta}$ appears in the diagonals of off-diagonal blocks of $\boldsymbol{\Omega}$, and $\boldsymbol{\Psi}$ appears in the diagonal blocks of $\boldsymbol{\Omega}$. (b) $\boldsymbol{W} = \boldsymbol{\Omega}^{-1}$ (left), and $\boldsymbol{W}_{\boldsymbol{\Theta}}$ and $\boldsymbol{W}_{\boldsymbol{\Psi}}$ (right) involved in the gradient computation. Computing $\boldsymbol{W}_{\boldsymbol{\Theta}}$ and $\boldsymbol{W}_{\boldsymbol{\Psi}}$ amounts to collapsing $\boldsymbol{W}$ based on the Kronecker-sum structure in Panel (a). The (i, j)th element of $\boldsymbol{W}_{\boldsymbol{\Theta}}$ is obtained by summing over the diagonals of the (i, j)th block (blue arrows), and $\boldsymbol{W}_{\boldsymbol{\Psi}}$ is obtained by summing over the diagonal blocks of $\boldsymbol{W}$ (yellow arrows).

adopted in various other statistical methods to represent sparse associations between two sets of variables, such as dependencies between genes and mutations in colorectal cancer (Haupt et al., 2021) and between microRNAs and diseases (Li et al., 2018). It also has been embedded in neural networks to model both row and column dependencies in a matrix (Zhang et al., 2018b; Gao et al., 2020).

Given data $\{\boldsymbol{Y}^1, \ldots, \boldsymbol{Y}^n\}$, where $\boldsymbol{Y}^i \in \mathbb{R}^{q \times p}$ for $i = 1, \ldots, n$ for n independent observations, BiGLasso (Kalaitzis et al., 2013) and TeraLasso (Greenewald et al., 2019) obtained a sparse estimate of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ by minimizing the L_1-regularized negative log-likelihood of data:

$$\operatorname*{argmin}_{\boldsymbol{\Theta}, \boldsymbol{\Psi}} f(\boldsymbol{\Theta}, \boldsymbol{\Psi}) \quad \text{subject to} \quad \boldsymbol{\Theta} \oplus \boldsymbol{\Psi} \succ 0, \tag{3}$$

where $\boldsymbol{A} \succ 0$ means $\boldsymbol{A}$ is positive definite. The objective function $f(\boldsymbol{\Theta}, \boldsymbol{\Psi})$ is

$$f(\boldsymbol{\Theta}, \boldsymbol{\Psi}) = g(\boldsymbol{\Theta}, \boldsymbol{\Psi}) + h(\boldsymbol{\Theta}, \boldsymbol{\Psi}),$$

with the smooth log-likelihood function $g(\boldsymbol{\Theta}, \boldsymbol{\Psi}) = q\,\mathrm{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + p\,\mathrm{tr}(\boldsymbol{T}\boldsymbol{\Psi}) - \log|\boldsymbol{\Theta} \oplus \boldsymbol{\Psi}|$ and non-smooth penalty $h(\boldsymbol{\Theta}, \boldsymbol{\Psi}) = q\gamma_{\boldsymbol{\Theta}}\|\boldsymbol{\Theta}\|_{1,\text{off}} + p\gamma_{\boldsymbol{\Psi}}\|\boldsymbol{\Psi}\|_{1,\text{off}}$, given the sample covariances $\boldsymbol{S} = \frac{1}{nq}\sum_{i=1}^n \boldsymbol{Y}^{iT}\boldsymbol{Y}^i$ and $\boldsymbol{T} = \frac{1}{np}\sum_{i=1}^n \boldsymbol{Y}^i\boldsymbol{Y}^{iT}$, the regularization parameters $\gamma_{\boldsymbol{\Theta}}$ and $\gamma_{\boldsymbol{\Psi}}$, and $\|\cdot\|_{1,\text{off}}$ for the L_1-regularization of the off-diagonal elements of the matrix.

One of the main challenges of solving Eq. (3) arises from the unidentifiable diagonals of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ given $\boldsymbol{\Omega}$. Although Greenewald et al. (2019) showed that the objective in Eq. (3) has a unique global optimum with respect to $\boldsymbol{\Omega} = \boldsymbol{\Theta} \oplus \boldsymbol{\Psi}$, this does not imply the uniqueness of the pair $(\boldsymbol{\Theta}, \boldsymbol{\Psi})$ given $\boldsymbol{\Omega}$, since the set in Eq. (2) forms equivalence classes

$$S[\boldsymbol{\Omega}] = \{(\boldsymbol{\Theta}, \boldsymbol{\Psi}) \in \mathbb{KS}^{p,q} | \, (\boldsymbol{\Theta} - c\boldsymbol{I}_p) \oplus (\boldsymbol{\Psi} + c\boldsymbol{I}_q) = \boldsymbol{\Omega}, c \in \mathbb{R}\}. \tag{4}$$

Thus, $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ whose diagonals are modified by a constant c and $-c$ lead to the same $\boldsymbol{\Omega}$. While BiGLasso did not estimate the diagonals of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$, TeraLasso formed an identifiable reparameterization $\boldsymbol{\Omega} = \bar{\boldsymbol{\Theta}} \oplus \bar{\boldsymbol{\Psi}} + \tau \boldsymbol{I}_{pq}$ with $\tau = \frac{\mathrm{tr}(\boldsymbol{\Theta} \oplus \boldsymbol{\Psi})}{pq}$, where $\bar{\boldsymbol{\Theta}}$ and $\bar{\boldsymbol{\Psi}}$ are forced to have zero traces as follows:

$$\bar{\boldsymbol{\Theta}} = \boldsymbol{\Theta} - \frac{\mathrm{tr}(\boldsymbol{\Theta})}{p} \boldsymbol{I}_p \qquad \text{and} \qquad \bar{\boldsymbol{\Psi}} = \boldsymbol{\Psi} - \frac{\mathrm{tr}(\boldsymbol{\Psi})}{q} \boldsymbol{I}_q. \tag{5}$$

Then, in each iteration of the gradient descent method, TeraLasso projected the gradient to this reparameterized space and distributed τ to $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ as $\bar{\boldsymbol{\Theta}} \leftarrow \bar{\boldsymbol{\Theta}} + \frac{1}{2}\tau \boldsymbol{I}$ and $\bar{\boldsymbol{\Psi}} \leftarrow \bar{\boldsymbol{\Psi}} + \frac{1}{2}\tau \boldsymbol{I}$. For efficient computation and projection of the gradient, TeraLasso employed an eigendecomposition of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$: $\boldsymbol{\Theta} = \boldsymbol{Q}_{\boldsymbol{\Theta}} \boldsymbol{\Lambda}_{\boldsymbol{\Theta}} \boldsymbol{Q}_{\boldsymbol{\Theta}}^T$ with the eigenvector matrix $\boldsymbol{Q}_{\boldsymbol{\Theta}}$ and the diagonal eigenvalue matrix $\boldsymbol{\Lambda}_{\boldsymbol{\Theta}}$, and similarly, $\boldsymbol{\Psi} = \boldsymbol{Q}_{\boldsymbol{\Psi}} \boldsymbol{\Lambda}_{\boldsymbol{\Psi}} \boldsymbol{Q}_{\boldsymbol{\Psi}}^T$. TeraLasso exploited the property that the eigendecomposition of the Kronecker sum of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ is

$$\boldsymbol{\Theta} \oplus \boldsymbol{\Psi} = (\boldsymbol{Q}_{\boldsymbol{\Theta}} \otimes \boldsymbol{Q}_{\boldsymbol{\Psi}})(\boldsymbol{\Lambda}_{\boldsymbol{\Theta}} \oplus \boldsymbol{\Lambda}_{\boldsymbol{\Psi}})(\boldsymbol{Q}_{\boldsymbol{\Theta}} \otimes \boldsymbol{Q}_{\boldsymbol{\Psi}})^T. \tag{6}$$

Then, the inverse of $\boldsymbol{\Theta} \oplus \boldsymbol{\Psi}$ can be obtained efficiently by inverting the diagonal eigenvalue matrix $\boldsymbol{\Lambda}_{\boldsymbol{\Theta}} \oplus \boldsymbol{\Lambda}_{\boldsymbol{\Psi}}$:

$$\boldsymbol{W} = (\boldsymbol{\Theta} \oplus \boldsymbol{\Psi})^{-1} = (\boldsymbol{Q}_{\boldsymbol{\Theta}} \otimes \boldsymbol{Q}_{\boldsymbol{\Psi}})(\boldsymbol{\Lambda}_{\boldsymbol{\Theta}} \oplus \boldsymbol{\Lambda}_{\boldsymbol{\Psi}})^{-1}(\boldsymbol{Q}_{\boldsymbol{\Theta}} \otimes \boldsymbol{Q}_{\boldsymbol{\Psi}})^T. \tag{7}$$

While TeraLasso is significantly faster than BiGLasso, its scalability is limited to graphs with a few hundred nodes.

3. EiGLasso

We introduce EiGLasso, an efficient method for estimating a sparse Kronecker-sum inverse covariance matrix. We begin by describing a simple new scheme for identifying the unidentifiable parameters in Section 3.1. We introduce EiGLasso, Newton's method with the exact Hessian in Section 3.2 and its modification with the approximate Hessian in Section 3.3. In Section 3.4, we describe a simple strategy for determining the unidentifiable parameters in EiGLasso. In Section 3.5, we discuss additional strategies for improving the computational efficiency.

3.1 Identifying Parameters with Trace Ratio

Theorem 1 below provides a simple approach to identifying the diagonal elements of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$, given $\boldsymbol{\Omega}$ and the ratio of the traces of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$.

Theorem 1 *Assume that the trace ratio of $\boldsymbol{\Psi}$ and $\boldsymbol{\Theta}$ is given as $\rho = \frac{\mathrm{tr}(\boldsymbol{\Psi})}{\mathrm{tr}(\boldsymbol{\Theta})}$. Then, a Kronecker-sum matrix $\boldsymbol{\Omega} = \boldsymbol{\Theta} \oplus \boldsymbol{\Psi}$ can be mapped to a unique pair of symmetric matrices $(\boldsymbol{\Theta}, \boldsymbol{\Psi})$ with the diagonals of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ identified as*

$$\begin{aligned} \mathrm{diag}(\boldsymbol{\Theta}) &= \frac{1}{q}\left[\mathrm{diag}\left(\sum_{i=1}^{q} (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})^T \boldsymbol{\Omega} (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})\right) - \frac{\rho}{q + \rho p} \mathrm{tr}(\boldsymbol{\Omega}) \boldsymbol{I}\right], \\ \mathrm{diag}(\boldsymbol{\Psi}) &= \frac{1}{p}\left[\mathrm{diag}\left(\sum_{j=1}^{p} (\boldsymbol{e}_{p,j} \otimes \boldsymbol{I}_q)^T \boldsymbol{\Omega} (\boldsymbol{e}_{p,j} \otimes \boldsymbol{I}_q)\right) - \frac{1}{q + \rho p} \mathrm{tr}(\boldsymbol{\Omega}) \boldsymbol{I}\right], \end{aligned} \tag{8}$$

where $\boldsymbol{e}_{q,i}$ is a $q \times 1$ one-hot vector with 1 in the ith entry and 0's elsewhere.

Proof Given $\rho = \frac{\operatorname{tr}(\boldsymbol{\Psi})}{\operatorname{tr}(\boldsymbol{\Theta})}$, $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ are unique, since for any $c \in \mathbb{R} \setminus \{0\}$, $\frac{\operatorname{tr}(\boldsymbol{\Psi})}{\operatorname{tr}(\boldsymbol{\Theta})} \neq \frac{\operatorname{tr}(\boldsymbol{\Psi}+c\boldsymbol{I}_q)}{\operatorname{tr}(\boldsymbol{\Theta}-c\boldsymbol{I}_p)} = \frac{\operatorname{tr}(\boldsymbol{\Psi})+cq}{\operatorname{tr}(\boldsymbol{\Theta})-cp}$. From the definition of Kronecker sum, we notice

$$\sum_{i=(j-1)q+1}^{jq} [\boldsymbol{\Omega}]_{ii} = q[\boldsymbol{\Theta}]_{jj} + \operatorname{tr}(\boldsymbol{\Psi}), \tag{9}$$

and

$$\operatorname{tr}(\boldsymbol{\Omega}) = q \operatorname{tr}(\boldsymbol{\Theta}) + p \operatorname{tr}(\boldsymbol{\Psi}). \tag{10}$$

From the trace ratio and Eq. (10), we have $\operatorname{tr}(\boldsymbol{\Psi}) = \frac{\rho}{q+\rho p} \operatorname{tr}(\boldsymbol{\Omega})$. Plugging this into Eq. (9), we identify $[\boldsymbol{\Theta}]_{jj}$ as

$$[\boldsymbol{\Theta}]_{jj} = \frac{1}{q}\left(\sum_{i=(j-1)q+1}^{jq} [\boldsymbol{\Omega}]_{ii} - \frac{\rho}{q+\rho p} \operatorname{tr}(\boldsymbol{\Omega}) \right).$$

The case for $\boldsymbol{\Psi}$ can be shown in a similar way. ■

Theorem 1 suggests that it is possible to identify the diagonals of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ by solving Eq. (3) with the linear constraint $\frac{\operatorname{tr}(\boldsymbol{\Psi})}{\operatorname{tr}(\boldsymbol{\Theta})} = \rho$:

$$\operatorname*{argmin}_{\boldsymbol{\Theta},\boldsymbol{\Psi}} f(\boldsymbol{\Theta}, \boldsymbol{\Psi}) \quad \text{subject to} \quad \boldsymbol{\Theta} \oplus \boldsymbol{\Psi} \succ 0,\ \boldsymbol{R}^T \begin{bmatrix} \operatorname{vec}(\boldsymbol{\Theta}) \\ \operatorname{vec}(\boldsymbol{\Psi}) \end{bmatrix} = 0, \tag{11}$$

where the second constraint with $\boldsymbol{R} = \begin{bmatrix} -\rho \operatorname{vec}(\boldsymbol{I}_p) \\ \operatorname{vec}(\boldsymbol{I}_q) \end{bmatrix}$ is equivalent to $\frac{\operatorname{tr}(\boldsymbol{\Psi})}{\operatorname{tr}(\boldsymbol{\Theta})} = \rho$. To handle this additional constraint explicitly, the substitution method or Newton's method for equality-constrained optimization problem could be used (Boyd and Vandenberghe, 2004). Instead, in Sections 3.2 and 3.4, we show that in EiGLasso, because of the special problem structure, it is sufficient to solve Eq. (3) with Newton's method, ignoring the equality constraint, and to adjust the diagonals of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ once after convergence to satisfy the constraint.

3.2 EiGLasso with Exact Hessian

We introduce EiGLasso for an efficient estimation of sparse $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ that form a Kronecker-sum inverse covariance matrix. We adopt the framework of QUIC (Hsieh et al., 2014), Newton's method for estimating sparse Gaussian graphical models. In each iteration, QUIC found a Newton direction by minimizing the L_1-regularized second-order approximation of the objective, and updated the parameters given this Newton direction and the step size found by backtracking line search (Bertsekas, 1995; Tseng and Yun, 2009). Using the same strategy, EiGLasso finds Newton directions, $D_{\boldsymbol{\Theta}}$ for $\boldsymbol{\Theta}$ and $D_{\boldsymbol{\Psi}}$ for $\boldsymbol{\Psi}$, by minimizing the following second-order approximation of the smooth part $g(\boldsymbol{\Theta}, \boldsymbol{\Psi})$ of the objective in Eq. (11) with the L_1-regularization:

$$(D_{\boldsymbol{\Theta}}, D_{\boldsymbol{\Psi}}) = \operatorname*{argmin}_{\Delta_{\boldsymbol{\Theta}},\Delta_{\boldsymbol{\Psi}}} \hat{g}(\Delta_{\boldsymbol{\Theta}}, \Delta_{\boldsymbol{\Psi}}) + h(\boldsymbol{\Theta} + \Delta_{\boldsymbol{\Theta}}, \boldsymbol{\Psi} + \Delta_{\boldsymbol{\Psi}}) \tag{12}$$

Algorithm 1: Line Search

```
input  : $0 < \sigma < 0.5$, $0 < \beta < 1$
output : step size $\alpha$
Initialize $\alpha = 1$
for $i = 0, 1, \ldots$ do
    Check the following conditions:
        1. $(\boldsymbol{\Theta} + \alpha D_{\boldsymbol{\Theta}}) \oplus (\boldsymbol{\Psi} + \alpha D_{\boldsymbol{\Psi}}) \succ 0$,
        2. $f(\boldsymbol{\Theta} + \alpha D_{\boldsymbol{\Theta}}, \boldsymbol{\Psi} + \alpha D_{\boldsymbol{\Psi}}) \leq f(\boldsymbol{\Theta}, \boldsymbol{\Psi}) + \alpha\sigma\delta$, where
           $\delta = [\mathrm{vec}(D_{\boldsymbol{\Theta}})^T, \mathrm{vec}(D_{\boldsymbol{\Psi}})^T]\, \mathrm{vec}(\boldsymbol{G}) + h(\boldsymbol{\Theta} + D_{\boldsymbol{\Theta}}, \boldsymbol{\Psi} + D_{\boldsymbol{\Psi}}) - h(\boldsymbol{\Theta}, \boldsymbol{\Psi})$.
    If satisfied, break. If not, $\alpha \leftarrow \alpha\beta$.
```

$$\text{subject to } \boldsymbol{R}^T \begin{bmatrix} \mathrm{vec}(\Delta_{\boldsymbol{\Theta}}) \\ \mathrm{vec}(\Delta_{\boldsymbol{\Psi}}) \end{bmatrix} = 0,$$

where

$$\hat{g}(\Delta_{\boldsymbol{\Theta}}, \Delta_{\boldsymbol{\Psi}}) = \mathrm{vec}(\boldsymbol{G})^T \begin{bmatrix} \mathrm{vec}(\Delta_{\boldsymbol{\Theta}}) \\ \mathrm{vec}(\Delta_{\boldsymbol{\Psi}}) \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \mathrm{vec}(\Delta_{\boldsymbol{\Theta}}) \\ \mathrm{vec}(\Delta_{\boldsymbol{\Psi}}) \end{bmatrix}^T \boldsymbol{H} \begin{bmatrix} \mathrm{vec}(\Delta_{\boldsymbol{\Theta}}) \\ \mathrm{vec}(\Delta_{\boldsymbol{\Psi}}) \end{bmatrix}$$

with the gradient $\boldsymbol{G}$ and Hessian $\boldsymbol{H}$

$$\mathrm{vec}(\boldsymbol{G}) = \begin{bmatrix} \mathrm{vec}\left(\boldsymbol{G}_{\boldsymbol{\Theta}}\right) \\ \mathrm{vec}\left(\boldsymbol{G}_{\boldsymbol{\Psi}}\right) \end{bmatrix} = \begin{bmatrix} \mathrm{vec}\left(\nabla_{\boldsymbol{\Theta}} g(\boldsymbol{\Theta}, \boldsymbol{\Psi})\right) \\ \mathrm{vec}\left(\nabla_{\boldsymbol{\Psi}} g(\boldsymbol{\Theta}, \boldsymbol{\Psi})\right) \end{bmatrix}, \tag{13}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{\Theta}} & \boldsymbol{H}_{\boldsymbol{\Theta\Psi}} \\ \boldsymbol{H}_{\boldsymbol{\Theta\Psi}}^T & \boldsymbol{H}_{\boldsymbol{\Psi}} \end{bmatrix} = \begin{bmatrix} \nabla^2_{\boldsymbol{\Theta}} g(\boldsymbol{\Theta}, \boldsymbol{\Psi}) & \nabla_{\boldsymbol{\Theta}} \nabla_{\boldsymbol{\Psi}} g(\boldsymbol{\Theta}, \boldsymbol{\Psi}) \\ \nabla_{\boldsymbol{\Theta}} \nabla_{\boldsymbol{\Psi}} g(\boldsymbol{\Theta}, \boldsymbol{\Psi})^T & \nabla^2_{\boldsymbol{\Psi}} g(\boldsymbol{\Theta}, \boldsymbol{\Psi}) \end{bmatrix}. \tag{14}$$

As we will show in Section 3.4, the objective above can be optimized without being concerned about the equality constraint. Hence, we optimize this objective with the coordinate descent method as in QUIC, ignoring the equality constraint. Given the descent directions $D_{\boldsymbol{\Theta}}$ and $D_{\boldsymbol{\Psi}}$, we update the parameters as $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} + \alpha D_{\boldsymbol{\Theta}}$ and $\boldsymbol{\Psi} \leftarrow \boldsymbol{\Psi} + \alpha D_{\boldsymbol{\Psi}}$, with the step size α found by the line-search method in Algorithm 1.

Two key challenges arise in a direct application of QUIC to Kronecker-sum inverse covariance estimation. First, as we detail in the next section, the gradient $\boldsymbol{G}$ and Hessian $\boldsymbol{H}$ involve the computation of the $pq \times pq$ matrix $\boldsymbol{W} = (\boldsymbol{\Theta} \oplus \boldsymbol{\Psi})^{-1}$, which is significantly larger than a $p \times p$ matrix required for a single graph in QUIC. Second, the constraint on the trace ratio in Eq. (12) needs to be considered to handle the unidentifiability of the diagonals of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ during the optimization. In the rest of Section 3, we describe how we address these challenges by leveraging the eigenstructure of the model, and provide the details of the EiGLasso optimization outlined in Algorithm 2.

3.2.1 Efficient Computation of Gradient and Hessian via Eigendecomposition

In Lemma 2 below, we provide the form of the gradient $\boldsymbol{G}$ in Eq. (13) and the Hessian $\boldsymbol{H}$ in Eq. (14). We show that $\boldsymbol{G}$ can be represented in a significantly more compact form than what has been previously presented in Kalaitzis et al. (2013) and Greenewald et al.

```
Algorithm 2: EiGLasso
  input  : Sample covariances $\boldsymbol{S} = \frac{1}{nq}\sum_{i=1}^{n} \boldsymbol{Y}^{iT}\boldsymbol{Y}^{i}$ and $\boldsymbol{T} = \frac{1}{np}\sum_{i=1}^{n} \boldsymbol{Y}^{i}\boldsymbol{Y}^{iT}$,
           regularization parameters $\gamma_{\boldsymbol{\Theta}}$ and $\gamma_{\boldsymbol{\Psi}}$, $K$ for Hessian approximation,
           line search parameters $\sigma$ and $0 < \beta < 1$, and trace ratio $\rho = \operatorname{tr}(\boldsymbol{\Psi})/\operatorname{tr}(\boldsymbol{\Theta})$.
  output : Parameters $\boldsymbol{\Theta}, \boldsymbol{\Psi}$.
  Initialize $\boldsymbol{\Theta}^0 \leftarrow \boldsymbol{I}_p, \boldsymbol{\Psi}^0 \leftarrow \boldsymbol{I}_q$.
  for $t = 0, 1, \dots$ do
      Eigendecompose $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$.
      Compute $\boldsymbol{G}$ and $\boldsymbol{H}$.
      Determine active sets $\mathcal{A}_{\boldsymbol{\Theta}}$ and $\mathcal{A}_{\boldsymbol{\Psi}}$.
      Compute $(D^t_{\boldsymbol{\Theta}}, D^t_{\boldsymbol{\Psi}})$ via coordinate descent over the active sets.
      Compute a step-size $\alpha_t$ using Algorithm 1.
      Check convergence.
  Adjust the diagonal elements of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ according to the trace ratio $\rho$.
```

(2019). We exploit this compact representation to further reduce the computation time via eigendecomposition of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ in Theorem 3.

Lemma 2 *Let $\boldsymbol{W} = (\boldsymbol{\Theta} \oplus \boldsymbol{\Psi})^{-1}$. The components of gradient $\boldsymbol{G}$ in Eq.* (13) *are given as*

$$\boldsymbol{G}_{\boldsymbol{\Theta}} = q\boldsymbol{S} - \boldsymbol{W}_{\boldsymbol{\Theta}}, \quad \boldsymbol{G}_{\boldsymbol{\Psi}} = p\boldsymbol{T} - \boldsymbol{W}_{\boldsymbol{\Psi}}, \tag{15}$$

where

$$\boldsymbol{W}_{\boldsymbol{\Theta}} = \sum_{i=1}^{q} (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})^T \boldsymbol{W} (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i}), \quad \boldsymbol{W}_{\boldsymbol{\Psi}} = \sum_{i=1}^{p} (\boldsymbol{e}_{p,i} \otimes \boldsymbol{I}_q)^T \boldsymbol{W} (\boldsymbol{e}_{p,i} \otimes \boldsymbol{I}_q). \tag{16}$$

The Hessian $\boldsymbol{H}$ in Eq. (14) *is given as*

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{\Theta}} & \boldsymbol{H}_{\boldsymbol{\Theta\Psi}} \\ \boldsymbol{H}^T_{\boldsymbol{\Theta\Psi}} & \boldsymbol{H}_{\boldsymbol{\Psi}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}^T_{\boldsymbol{\Theta}}(\boldsymbol{W} \otimes \boldsymbol{W})\boldsymbol{P}_{\boldsymbol{\Theta}} & \boldsymbol{P}^T_{\boldsymbol{\Theta}}(\boldsymbol{W} \otimes \boldsymbol{W})\boldsymbol{P}_{\boldsymbol{\Psi}} \\ \boldsymbol{P}^T_{\boldsymbol{\Psi}}(\boldsymbol{W} \otimes \boldsymbol{W})\boldsymbol{P}_{\boldsymbol{\Theta}} & \boldsymbol{P}^T_{\boldsymbol{\Psi}}(\boldsymbol{W} \otimes \boldsymbol{W})\boldsymbol{P}_{\boldsymbol{\Psi}} \end{bmatrix} = \boldsymbol{P}^T(\boldsymbol{W} \otimes \boldsymbol{W})\boldsymbol{P}, \tag{17}$$

where $\boldsymbol{P} = [\boldsymbol{P}_{\boldsymbol{\Theta}}, \boldsymbol{P}_{\boldsymbol{\Psi}}]$ is a $(p^2q^2) \times (p^2 + q^2)$ matrix with $\boldsymbol{P}_{\boldsymbol{\Theta}} = \sum_{i=1}^{q} \boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i} \otimes \boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i}$ and $\boldsymbol{P}_{\boldsymbol{\Psi}} = \sum_{j=1}^{p} \boldsymbol{e}_{p,j} \otimes \boldsymbol{I}_q \otimes \boldsymbol{e}_{p,j} \otimes \boldsymbol{I}_q$.

Proof Let $\boldsymbol{E}_{p,ij} = \boldsymbol{e}_{p,i}\boldsymbol{e}^T_{p,j}$ be a $p \times p$ one-hot matrix with 1 in the (i, j)th element and zero elsewhere. Then, for the gradient $\boldsymbol{G}_{\boldsymbol{\Theta}}$, we have

$$\begin{aligned} [\boldsymbol{G}_{\boldsymbol{\Theta}}]_{ij} &= q[\boldsymbol{S}]_{ij} - \operatorname{tr}\left(\boldsymbol{W}(\boldsymbol{E}_{p,ij} \otimes \boldsymbol{I}_q)\right) \\ &= q[\boldsymbol{S}]_{ij} - \operatorname{tr}\big(\boldsymbol{W}(\boldsymbol{E}_{p,ij} \otimes \sum_l \boldsymbol{e}_{q,l}\boldsymbol{e}^T_{q,l})\big) \\ &= q[\boldsymbol{S}]_{ij} - \operatorname{tr}\big(\sum_l \boldsymbol{W}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,l})(\boldsymbol{E}_{p,ij} \otimes 1)(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,l})^T\big) \\ &= q[\boldsymbol{S}]_{ij} - \operatorname{tr}\big(\sum_l (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,l})^T \boldsymbol{W} (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,l})\boldsymbol{E}_{p,ij}\big). \end{aligned}$$

By using $\operatorname{tr}(\boldsymbol{A}\boldsymbol{E}_{p,ij}) = \boldsymbol{A}_{ij}$, we collect the elements $[\boldsymbol{G}_{\boldsymbol{\Theta}}]_{ij}$ to obtain Eqs. (15) and (16). The case for $\boldsymbol{G}_{\boldsymbol{\Psi}}$ can be shown similarly. Now, for the Hessian, we have

$$\begin{aligned}
[\boldsymbol{H}_{\boldsymbol{\Theta}}]_{ip+j,lp+k} &= \operatorname{tr}\left(\boldsymbol{W}(\boldsymbol{E}_{p,ij} \otimes \boldsymbol{I}_q)\boldsymbol{W}(\boldsymbol{E}_{p,kl} \otimes \boldsymbol{I}_q)\right) \\
&= \operatorname{tr}\left(\left[\sum_r \boldsymbol{W}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,r})\boldsymbol{E}_{p,ij}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,r})^T\right]\left[\sum_s \boldsymbol{W}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,s})\boldsymbol{E}_{p,kl}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,s})^T\right]\right) \\
&= \operatorname{tr}\left(\sum_{r,s}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,s})^T\boldsymbol{W}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,r})\boldsymbol{E}_{p,ij}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,r})^T\boldsymbol{W}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,s})\boldsymbol{E}_{p,kl}\right),
\end{aligned}$$

by $\operatorname{tr}(\boldsymbol{ABCD}) = \operatorname{vec}(\boldsymbol{B}^T)^T(\boldsymbol{A}^T \otimes \boldsymbol{C})\operatorname{vec}(\boldsymbol{D})$ for matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$, and $\operatorname{vec}(\boldsymbol{E}_{p,ij}) = \boldsymbol{e}_{p^2,jp+i}$,

$$\begin{aligned}
&= \boldsymbol{e}_{p^2,ip+j}^T\left(\sum_{r,s}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,r})^T\boldsymbol{W}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,s}) \otimes (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,r})^T\boldsymbol{W}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,s})\right)\boldsymbol{e}_{p^2,lp+k} \\
&= \boldsymbol{e}_{p^2,ip+j}^T\left(\sum_{r,s}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,r} \otimes \boldsymbol{I}_p \otimes \boldsymbol{e}_{q,r})^T(\boldsymbol{W} \otimes \boldsymbol{W})(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,s} \otimes \boldsymbol{I}_p \otimes \boldsymbol{e}_{q,s})\right)\boldsymbol{e}_{p^2,lp+k}.
\end{aligned}$$

We collect the elements into a matrix

$$\boldsymbol{H}_{\boldsymbol{\Theta}} = \left(\sum_{r=1}^{q}\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,r} \otimes \boldsymbol{I}_p \otimes \boldsymbol{e}_{q,r}\right)^T(\boldsymbol{W} \otimes \boldsymbol{W})\left(\sum_{s=1}^{q}\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,s} \otimes \boldsymbol{I}_p \otimes \boldsymbol{e}_{q,s}\right).$$

The cases for $\boldsymbol{H}_{\boldsymbol{\Psi}}$ and $\boldsymbol{H}_{\boldsymbol{\Theta\Psi}}$ can be shown in a similar way. ■

According to Lemma 2, the gradient $\boldsymbol{G}$ is obtained by collapsing $\boldsymbol{W}$ as in Eq. (16) and the Hessian $\boldsymbol{H}$ is obtained by collapsing $\boldsymbol{W} \otimes \boldsymbol{W}$ as in Eq. (17). This collapse of the gradient is illustrated in Figure 1(b). While the Kronecker sum inflates $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ into $\boldsymbol{\Omega}$ (Figure 1(a)), this inflated structure gets deflated, when computing $\boldsymbol{W}_{\boldsymbol{\Theta}}$ and $\boldsymbol{W}_{\boldsymbol{\Psi}}$ in the gradients from $\boldsymbol{W} = \boldsymbol{\Omega}^{-1}$ (Figure 1(b)). This deflation can be viewed as applying the mask $\mathbb{1}_p \oplus \mathbb{1}_q$, where $\mathbb{1}_a$ is an $a \times a$ matrix of ones, to $\boldsymbol{W}$ to obtain $\boldsymbol{W}^{\text{masked}}$ such that $[\boldsymbol{W}^{\text{masked}}]_{ij} = [\boldsymbol{W}]_{ij}$ if $[\mathbb{1}_p \oplus \mathbb{1}_q]_{ij} \neq 0$, otherwise $[\boldsymbol{W}^{\text{masked}}]_{ij} = 0$, and replacing $\boldsymbol{W}$ in Eq. (16) with $\boldsymbol{W}^{\text{masked}}$. In other words, only the non-zero elements of $\boldsymbol{W}^{\text{masked}}$ contribute towards $\boldsymbol{W}_{\boldsymbol{\Theta}}$ (blue in Figure 1(b)) and $\boldsymbol{W}_{\boldsymbol{\Psi}}$ (yellow in Figure 1(b)). The collapse of $\boldsymbol{W} \otimes \boldsymbol{W}$ in Eq. (17) to obtain the Hessian can be viewed as the same type of deflation applied twice to $\boldsymbol{W} \otimes \boldsymbol{W}$ (details in Appendix A).

Lemma 2 reveals the challenge in computing $\boldsymbol{G}$ and $\boldsymbol{H}$ in a direct application of QUIC to the Kronecker-sum model: $\boldsymbol{G}$ involves computing the large $pq \times pq$ matrix $\boldsymbol{W}$ and $\boldsymbol{H}$ involves computing the even larger $p^2q^2 \times p^2q^2$ matrix $\boldsymbol{W} \otimes \boldsymbol{W}$. In Theorem 3 below, we show that $\boldsymbol{G}$ and $\boldsymbol{H}$ can be obtained via eigendecomposition of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ without explicitly constructing $\boldsymbol{W}$ and $\boldsymbol{W} \otimes \boldsymbol{W}$.

Theorem 3 *Given the eigendecomposition* $\boldsymbol{\Theta} = \boldsymbol{Q}_{\boldsymbol{\Theta}}\boldsymbol{\Lambda}_{\boldsymbol{\Theta}}\boldsymbol{Q}_{\boldsymbol{\Theta}}^T$ *and* $\boldsymbol{\Psi} = \boldsymbol{Q}_{\boldsymbol{\Psi}}\boldsymbol{\Lambda}_{\boldsymbol{\Psi}}\boldsymbol{Q}_{\boldsymbol{\Psi}}^T$ *with the kth smallest eigenvalue* $\lambda_{\boldsymbol{\Theta},k}$ *and* $\lambda_{\boldsymbol{\Psi},k}$ *in the* (k,k)*th element of* $\boldsymbol{\Lambda}_{\boldsymbol{\Theta}}$ *and* $\boldsymbol{\Lambda}_{\boldsymbol{\Psi}}$*, the gradient* *in Eq.* (15) *is given as*

$$\boldsymbol{G}_{\boldsymbol{\Theta}} = q\boldsymbol{S} - \boldsymbol{Q}_{\boldsymbol{\Theta}} \left(\sum_{k=1}^{q} \boldsymbol{\Xi}_{\boldsymbol{\Theta},k} \right) \boldsymbol{Q}_{\boldsymbol{\Theta}}^T, \qquad \boldsymbol{G}_{\boldsymbol{\Psi}} = p\boldsymbol{T} - \boldsymbol{Q}_{\boldsymbol{\Psi}} \left(\sum_{l=1}^{p} \boldsymbol{\Xi}_{\boldsymbol{\Psi},l} \right) \boldsymbol{Q}_{\boldsymbol{\Psi}}^T, \tag{18}$$

where

$$\boldsymbol{\Xi}_{\boldsymbol{\Theta},k} = \left(\boldsymbol{\Lambda}_{\boldsymbol{\Theta}} + \lambda_{\boldsymbol{\Psi},k} \boldsymbol{I}_p \right)^{-1}, \qquad \boldsymbol{\Xi}_{\boldsymbol{\Psi},l} = \left(\boldsymbol{\Lambda}_{\boldsymbol{\Psi}} + \lambda_{\boldsymbol{\Theta},l} \boldsymbol{I}_q \right)^{-1}.$$

For the Hessian, we have

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{\Theta}} & \boldsymbol{H}_{\boldsymbol{\Theta}\boldsymbol{\Psi}} \\ \boldsymbol{H}_{\boldsymbol{\Theta}\boldsymbol{\Psi}}^T & \boldsymbol{H}_{\boldsymbol{\Psi}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}_{\boldsymbol{\Theta}} \otimes \boldsymbol{Q}_{\boldsymbol{\Theta}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_{\boldsymbol{\Psi}} \otimes \boldsymbol{Q}_{\boldsymbol{\Psi}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_{\boldsymbol{H}_{\boldsymbol{\Theta}}} & \boldsymbol{\Upsilon} \\ \boldsymbol{\Upsilon}^T & \boldsymbol{\Lambda}_{\boldsymbol{H}_{\boldsymbol{\Psi}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{Q}_{\boldsymbol{\Theta}} \otimes \boldsymbol{Q}_{\boldsymbol{\Theta}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_{\boldsymbol{\Psi}} \otimes \boldsymbol{Q}_{\boldsymbol{\Psi}} \end{bmatrix}^T, \tag{19}$$

where

$$\begin{aligned} \boldsymbol{\Lambda}_{\boldsymbol{H}_{\boldsymbol{\Theta}}} &= \sum_{k=1}^{q} \boldsymbol{\Xi}_{\boldsymbol{\Theta},k} \otimes \boldsymbol{\Xi}_{\boldsymbol{\Theta},k}, \quad \boldsymbol{\Lambda}_{\boldsymbol{H}_{\boldsymbol{\Psi}}} = \sum_{l=1}^{p} \boldsymbol{\Xi}_{\boldsymbol{\Psi},l} \otimes \boldsymbol{\Xi}_{\boldsymbol{\Psi},l}, \\ \boldsymbol{\Upsilon} &= \tilde{\boldsymbol{I}}_{p^2,p} \begin{bmatrix} \lambda_{\boldsymbol{W},11}^2 & \cdots & \lambda_{\boldsymbol{W},1q}^2 \\ \vdots & \ddots & \vdots \\ \lambda_{\boldsymbol{W},p1}^2 & \cdots & \lambda_{\boldsymbol{W},pq}^2 \end{bmatrix} \tilde{\boldsymbol{I}}_{q,q^2}. \end{aligned} \tag{20}$$

$\tilde{\boldsymbol{I}}_{p^2,p}$ *and* $\tilde{\boldsymbol{I}}_{q,q^2}$ *above are stretched identity matrices,* $\tilde{\boldsymbol{I}}_{p^2,p} = \begin{bmatrix} \boldsymbol{E}_{p,11} & \boldsymbol{E}_{p,22} & \cdots & \boldsymbol{E}_{p,pp} \end{bmatrix}^T$ *and* $\tilde{\boldsymbol{I}}_{q,q^2} = \begin{bmatrix} \boldsymbol{E}_{q,11} & \boldsymbol{E}_{q,22} & \cdots & \boldsymbol{E}_{q,qq} \end{bmatrix}$, *where* $\boldsymbol{E}_{p,ij}$ *is a* $p \times p$ *one-hot matrix with* 1 *in the* (i,j)*th element and zero elsewhere, and* $\lambda_{\boldsymbol{W},lk} = (\lambda_{\boldsymbol{\Theta},l} + \lambda_{\boldsymbol{\Psi},k})^{-1}$.

Proof We prove for $\boldsymbol{G}_{\boldsymbol{\Theta}}$, $\boldsymbol{H}_{\boldsymbol{\Theta}}$, and $\boldsymbol{H}_{\boldsymbol{\Theta}\boldsymbol{\Psi}}$, because the case for $\boldsymbol{G}_{\boldsymbol{\Psi}}$ and $\boldsymbol{H}_{\boldsymbol{\Psi}}$ can be proved similarly. Let $\boldsymbol{q}_{\boldsymbol{\Theta},i}$ and $\boldsymbol{q}_{\boldsymbol{\Psi},i}$ be the ith eigenvectors of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$, given as the ith columns of $\boldsymbol{Q}_{\boldsymbol{\Theta}}$ and $\boldsymbol{Q}_{\boldsymbol{\Psi}}$. Then, we can re-write Eq. (7) as

$$\boldsymbol{W} = \sum_{l=1}^{p} \sum_{k=1}^{q} \lambda_{\boldsymbol{W},lk} (\boldsymbol{q}_{\boldsymbol{\Theta},l} \otimes \boldsymbol{q}_{\boldsymbol{\Psi},k}) (\boldsymbol{q}_{\boldsymbol{\Theta},l} \otimes \boldsymbol{q}_{\boldsymbol{\Psi},k})^T. \tag{21}$$

For $\boldsymbol{G}_{\boldsymbol{\Theta}}$, we substitute $\boldsymbol{W}$ in $\boldsymbol{W}_{\boldsymbol{\Theta}}$ in Eq. (16) with Eq. (21):

$$\begin{aligned} \boldsymbol{G}_{\boldsymbol{\Theta}} &= q\boldsymbol{S} - \sum_{i=1}^{q} (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})^T \left(\sum_{l=1}^{p} \sum_{k=1}^{q} \lambda_{\boldsymbol{W},lk} (\boldsymbol{q}_{\boldsymbol{\Theta},l} \otimes \boldsymbol{q}_{\boldsymbol{\Psi},k}) (\boldsymbol{q}_{\boldsymbol{\Theta},l} \otimes \boldsymbol{q}_{\boldsymbol{\Psi},k})^T \right) (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i}) \\ &= q\boldsymbol{S} - \sum_{i=1}^{q} \sum_{l=1}^{p} \sum_{k=1}^{q} \lambda_{\boldsymbol{W},lk} ([\boldsymbol{q}_{\boldsymbol{\Psi},k}]_i \boldsymbol{q}_{\boldsymbol{\Theta},l}) ([\boldsymbol{q}_{\boldsymbol{\Psi},k}]_i \boldsymbol{q}_{\boldsymbol{\Theta},l})^T \quad \text{since } (\boldsymbol{I} \otimes \boldsymbol{e}_{q,i})^T (\boldsymbol{a} \otimes \boldsymbol{b}) = \boldsymbol{b}_i \boldsymbol{a} \\ &= q\boldsymbol{S} - \sum_{l=1}^{p} \sum_{k=1}^{q} \lambda_{\boldsymbol{W},lk} \boldsymbol{q}_{\boldsymbol{\Theta},l} \boldsymbol{q}_{\boldsymbol{\Theta},l}^T \quad \text{since } \boldsymbol{Q}_{\boldsymbol{\Psi}} \text{ is orthonormal} \\ &= q\boldsymbol{S} - \sum_{k=1}^{q} \boldsymbol{Q}_{\boldsymbol{\Theta}} (\boldsymbol{\Lambda}_{\boldsymbol{\Theta}} + \lambda_{\boldsymbol{\Psi},k} \boldsymbol{I}_p)^{-1} \boldsymbol{Q}_{\boldsymbol{\Theta}}^T. \end{aligned}$$

This is equivalent to $\boldsymbol{G}_{\boldsymbol{\Theta}}$ in Eq. (18), since the summation can be performed over $\boldsymbol{\Xi}_{\boldsymbol{\Theta},k}$, instead of over $\boldsymbol{Q}_{\boldsymbol{\Theta}}\boldsymbol{\Xi}_{\boldsymbol{\Theta},k}\boldsymbol{Q}_{\boldsymbol{\Theta}}^T$.

To show for $\boldsymbol{H}_{\boldsymbol{\Theta}}$, we again substitute $\boldsymbol{W}$ in Eq. (17) with Eq. (21).

$$\begin{aligned}
\boldsymbol{H}_{\boldsymbol{\Theta}} &= \sum_{i=1}^{q}\sum_{j=1}^{q}\Bigg(\sum_{l=1}^{p}\sum_{k=1}^{q}\lambda_{\boldsymbol{W},lk}([\boldsymbol{q}_{\boldsymbol{\Psi},k}]_i\boldsymbol{q}_{\boldsymbol{\Theta},l})([\boldsymbol{q}_{\boldsymbol{\Psi},k}]_j\boldsymbol{q}_{\boldsymbol{\Theta},l})^T \\
&\qquad\qquad\qquad\qquad \otimes\sum_{r=1}^{p}\sum_{s=1}^{q}\lambda_{\boldsymbol{W},rs}([\boldsymbol{q}_{\boldsymbol{\Psi},s}]_i\boldsymbol{q}_{\boldsymbol{\Theta},r})([\boldsymbol{q}_{\boldsymbol{\Psi},s}]_j\boldsymbol{q}_{\boldsymbol{\Theta},r})^T\Bigg) \\
&= \sum_{k=1}^{q}\Bigg(\sum_{l=1}^{p}\lambda_{\boldsymbol{W},lk}\boldsymbol{q}_{\boldsymbol{\Theta},l}\boldsymbol{q}_{\boldsymbol{\Theta},l}^T\otimes\sum_{r=1}^{p}\lambda_{\boldsymbol{W},rk}\boldsymbol{q}_{\boldsymbol{\Theta},r}\boldsymbol{q}_{\boldsymbol{\Theta},r}^T\Bigg) \\
&= \sum_{k=1}^{q}\Big(\boldsymbol{Q}_{\boldsymbol{\Theta}}(\boldsymbol{\Lambda}_{\boldsymbol{\Theta}}+\lambda_{\boldsymbol{\Psi},k}\boldsymbol{I}_p)^{-1}\boldsymbol{Q}_{\boldsymbol{\Theta}}^T\Big)\otimes\Big(\boldsymbol{Q}_{\boldsymbol{\Theta}}(\boldsymbol{\Lambda}_{\boldsymbol{\Theta}}+\lambda_{\boldsymbol{\Psi},k}\boldsymbol{I}_p)^{-1}\boldsymbol{Q}_{\boldsymbol{\Theta}}^T\Big) \\
&= (\boldsymbol{Q}_{\boldsymbol{\Theta}}\otimes\boldsymbol{Q}_{\boldsymbol{\Theta}})\Bigg(\sum_{k=1}^{q}(\boldsymbol{\Lambda}_{\boldsymbol{\Theta}}+\lambda_{\boldsymbol{\Psi},k}\boldsymbol{I}_p)^{-1}\otimes(\boldsymbol{\Lambda}_{\boldsymbol{\Theta}}+\lambda_{\boldsymbol{\Psi},k}\boldsymbol{I}_p)^{-1}\Bigg)(\boldsymbol{Q}_{\boldsymbol{\Theta}}\otimes\boldsymbol{Q}_{\boldsymbol{\Theta}})^T.
\end{aligned}$$

For $\boldsymbol{H}_{\boldsymbol{\Theta\Psi}}$, we substitute $\boldsymbol{W}$ in Eq. (17) with Eq. (21).

$$\begin{aligned}
\boldsymbol{H}_{\boldsymbol{\Theta\Psi}} &= \sum_{i=1}^{q}\sum_{j=1}^{p}\Bigg(\sum_{l=1}^{p}\sum_{k=1}^{q}\lambda_{\boldsymbol{W},lk}([\boldsymbol{q}_{\boldsymbol{\Psi},k}]_i\boldsymbol{q}_{\boldsymbol{\Theta},l})([\boldsymbol{q}_{\boldsymbol{\Theta},l}]_j\boldsymbol{q}_{\boldsymbol{\Psi},k})^T \\
&\qquad\qquad\qquad\qquad \otimes\sum_{r=1}^{p}\sum_{s=1}^{q}\lambda_{\boldsymbol{W},rs}([\boldsymbol{q}_{\boldsymbol{\Psi},s}]_i\boldsymbol{q}_{\boldsymbol{\Theta},r})([\boldsymbol{q}_{\boldsymbol{\Theta},r}]_j\boldsymbol{q}_{\boldsymbol{\Psi},s})^T\Bigg) \\
&= \sum_{l=1}^{p}\sum_{k=1}^{q}\lambda_{\boldsymbol{W},lk}^2(\boldsymbol{q}_{\boldsymbol{\Theta},l}\boldsymbol{q}_{\boldsymbol{\Psi},k}^T\otimes\boldsymbol{q}_{\boldsymbol{\Theta},l}\boldsymbol{q}_{\boldsymbol{\Psi},k}^T) \\
&= \sum_{l=1}^{p}(\boldsymbol{q}_{\boldsymbol{\Theta},l}\otimes\boldsymbol{q}_{\boldsymbol{\Theta},l})\sum_{k=1}^{q}\lambda_{\boldsymbol{W},lk}^2(\boldsymbol{q}_{\boldsymbol{\Psi},k}\otimes\boldsymbol{q}_{\boldsymbol{\Psi},k})^T \\
&= \begin{bmatrix}\boldsymbol{q}_{\boldsymbol{\Theta},1}^T\otimes\boldsymbol{q}_{\boldsymbol{\Theta},1}^T\\ \boldsymbol{q}_{\boldsymbol{\Theta},2}^T\otimes\boldsymbol{q}_{\boldsymbol{\Theta},2}^T\\ \vdots\\ \boldsymbol{q}_{\boldsymbol{\Theta},p}^T\otimes\boldsymbol{q}_{\boldsymbol{\Theta},p}^T\end{bmatrix}^T
\begin{bmatrix}\lambda_{\boldsymbol{W},11}^2 & \cdots & \lambda_{\boldsymbol{W},1q}^2\\ \vdots & \ddots & \vdots\\ \lambda_{\boldsymbol{W},p1}^2 & \cdots & \lambda_{\boldsymbol{W},pq}^2\end{bmatrix}
\begin{bmatrix}\boldsymbol{q}_{\boldsymbol{\Psi},1}^T\otimes\boldsymbol{q}_{\boldsymbol{\Psi},1}^T\\ \boldsymbol{q}_{\boldsymbol{\Psi},2}^T\otimes\boldsymbol{q}_{\boldsymbol{\Psi},2}^T\\ \vdots\\ \boldsymbol{q}_{\boldsymbol{\Psi},q}^T\otimes\boldsymbol{q}_{\boldsymbol{\Psi},q}^T\end{bmatrix} \\
&= (\boldsymbol{Q}_{\boldsymbol{\Theta}}\otimes\boldsymbol{Q}_{\boldsymbol{\Theta}})\tilde{\boldsymbol{I}}_{p^2,p}\begin{bmatrix}\lambda_{\boldsymbol{W},11}^2 & \cdots & \lambda_{\boldsymbol{W},1q}^2\\ \vdots & \ddots & \vdots\\ \lambda_{\boldsymbol{W},p1}^2 & \cdots & \lambda_{\boldsymbol{W},pq}^2\end{bmatrix}\tilde{\boldsymbol{I}}_{q,q^2}(\boldsymbol{Q}_{\boldsymbol{\Psi}}\otimes\boldsymbol{Q}_{\boldsymbol{\Psi}})^T.
\end{aligned}$$

■

Theorem 3 provides a significantly more efficient way of computing $\boldsymbol{G}$ and $\boldsymbol{H}$, compared to Lemma 2. Since $\boldsymbol{G}$ and $\boldsymbol{H}$ can be written entirely in terms of the eigenvectors and eigenvalues of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$, the operations that involve $\boldsymbol{W} = (\boldsymbol{\Theta}\oplus\boldsymbol{\Psi})^{-1}$ are replaced with the cheaper operation of eigendecomposition with time $\mathcal{O}(p^3+q^3)$ and space $\mathcal{O}(p^2+q^2)$. TeraLasso also used the eigendecomposition of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ in each iteration of the optimization. However, TeraLasso has done this for an efficient projection of the gradient to the reparameterized space, whereas EiGLasso performs the eigendecomposition for an efficient computation of $\boldsymbol{G}$ and $\boldsymbol{H}$ in the original parameter space.

It follows from Theorem 3 that $(\boldsymbol{\Theta},\boldsymbol{\Psi})$'s in the same equivalence class in Eq. (4) differ only in their eigenvalues, but not in eigenvectors. The equivalence class in Eq. (4) can be written equivalently as

$$S[\boldsymbol{\Omega}] = \{(\boldsymbol{\Theta},\boldsymbol{\Psi}) \in \mathbb{KS}^{p,q} | \, (\boldsymbol{\Lambda_\Theta} - c\boldsymbol{I}_p) \oplus (\boldsymbol{\Lambda_\Psi} + c\boldsymbol{I}_q) = \boldsymbol{\Lambda_\Omega}, c \in \mathbb{R}\}, \tag{22}$$

since $\boldsymbol{\Theta} - c\boldsymbol{I}_p = \boldsymbol{Q_\Theta \Lambda_\Theta Q_\Theta^T} - c\boldsymbol{Q_\Theta Q_\Theta^T} = \boldsymbol{Q_\Theta}(\boldsymbol{\Lambda_\Theta} - c\boldsymbol{I}_p)\boldsymbol{Q_\Theta^T}$ and similarly for $\boldsymbol{\Psi}$. We exploit this feature to handle the unidentifiable parameters during estimation in Section 3.4 and to analyze the convergence in Section 4.

In the theorem below, we prove several properties of $\boldsymbol{H}$. These properties will be used as a key to extending the theoretical results on the convergence of QUIC to EiGLasso in Section 4.

Theorem 4 *The Hessian $\boldsymbol{H}$ in EiGLasso has the following properties:*

- *It is positive semi-definite with the null space null($\boldsymbol{H}$) $= \{[\mathrm{vec}(\boldsymbol{X}_p)^T, \mathrm{vec}(\boldsymbol{X}_q)^T]^T | \boldsymbol{X}_p \oplus \boldsymbol{X}_q = 0, \boldsymbol{X}_p \in \mathbb{R}^{p\times p}, \boldsymbol{X}_p \in \mathbb{R}^{q\times q}\}$.*
- *On the Kronecker-sum space in Eq. (2), $\boldsymbol{H}$ is positive definite. The minimum eigenvalue $\lambda_{\boldsymbol{H},\mathrm{min}_0}$ outside of the null space and the maximum eigenvalue $\lambda_{\boldsymbol{H},\mathrm{max}}$ are bounded as*

$$\lambda_{\boldsymbol{H},\mathrm{min}_0} \geq \min\{p,q\}(\lambda_{\boldsymbol{\Theta},p} + \lambda_{\boldsymbol{\Psi},q})^{-2}, \quad \lambda_{\boldsymbol{H},\mathrm{max}} \leq (p+q)(\lambda_{\boldsymbol{\Theta},1} + \lambda_{\boldsymbol{\Psi},1})^{-2}.$$

Proof To prove the first property, from Lemma 2, we notice that since $\boldsymbol{W}$ is positive definite, the null space of $\boldsymbol{H}$ is given by $\boldsymbol{X}_p$ and $\boldsymbol{X}_q$ that satisfy

$$\boldsymbol{P}\begin{bmatrix}\mathrm{vec}(\boldsymbol{X}_p)\\ \mathrm{vec}(\boldsymbol{X}_q)\end{bmatrix} = 0.$$

Since the left-hand side of the above can be written as

$$\begin{aligned}\boldsymbol{P}\begin{bmatrix}\mathrm{vec}(\boldsymbol{X}_p)\\ \mathrm{vec}(\boldsymbol{X}_q)\end{bmatrix} &= \sum_{i=1}^{q} \mathrm{vec}\left((\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})\boldsymbol{X}_p(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})^T\right) + \sum_{j=1}^{p} \mathrm{vec}\left((\boldsymbol{e}_{p,j} \otimes \boldsymbol{I}_q)\boldsymbol{X}_q(\boldsymbol{e}_{p,j} \otimes \boldsymbol{I}_q)^T\right)\\ &= \mathrm{vec}\left(\boldsymbol{X}_p \otimes \boldsymbol{I}_q\right) + \mathrm{vec}\left(\boldsymbol{I}_p \otimes \boldsymbol{X}_q\right)\\ &= \mathrm{vec}\left(\boldsymbol{X}_p \oplus \boldsymbol{X}_q\right),\end{aligned}$$

the null space of $\boldsymbol{H}$ is given by $\boldsymbol{X}_p$ and $\boldsymbol{X}_q$ that satisfy $\boldsymbol{X}_p \oplus \boldsymbol{X}_q = 0$.

To prove the second property, we notice that the null space of $\boldsymbol{H}$ is outside of the constraint region $\boldsymbol{\Theta} \oplus \boldsymbol{\Psi} \succ 0$; thus, $\boldsymbol{H}$ is positive definite on the Kronecker-sum space in Eq. (2). To find the eigenvalues of $\boldsymbol{H}$, we first find the eigenvalues of

$$\boldsymbol{P}^T\boldsymbol{P} = \begin{bmatrix}\boldsymbol{P_\Theta^T P_\Theta} & \boldsymbol{P_\Theta^T P_\Psi}\\ \boldsymbol{P_\Psi^T P_\Theta} & \boldsymbol{P_\Psi^T P_\Psi}\end{bmatrix} = \begin{bmatrix}q\boldsymbol{I}_{p^2} & \mathrm{vec}(\boldsymbol{I}_p)\,\mathrm{vec}(\boldsymbol{I}_q)^T\\ \mathrm{vec}(\boldsymbol{I}_q)\,\mathrm{vec}(\boldsymbol{I}_p)^T & p\boldsymbol{I}_{q^2}\end{bmatrix},$$

by finding the solutions of the characteristic equation $\det(\boldsymbol{P}^T\boldsymbol{P} - \lambda\boldsymbol{I}) = 0$:

$$\begin{aligned}
\det(\boldsymbol{P}^T\boldsymbol{P} - \lambda\boldsymbol{I}_{p^2+q^2}) &= \det(q\boldsymbol{I}_{p^2} - \lambda\boldsymbol{I}_{p^2})\det\left(p\boldsymbol{I}_{q^2} - \lambda\boldsymbol{I}_{q^2} - \frac{p}{q-\lambda}\operatorname{vec}(\boldsymbol{I}_q)\operatorname{vec}(\boldsymbol{I}_q)^T\right)\\
&= (q-\lambda)^{p^2}\det\left((p-\lambda)\boldsymbol{I}_{q^2} - \frac{p}{q-\lambda}\operatorname{vec}(\boldsymbol{I}_q)\operatorname{vec}(\boldsymbol{I}_q)^T\right)\\
&= (q-\lambda)^{p^2-1}(p-\lambda)^{q^2-1}\big((p-\lambda)(q-\lambda) - pq\big) \qquad \text{from matrix determinant lemma}\\
&= (p-\lambda)^{q^2-1}(q-\lambda)^{p^2-1}\lambda\big(\lambda - (p+q)\big) = 0.
\end{aligned}$$

Thus, the eigenvalues of $\boldsymbol{P}^T\boldsymbol{P}$ are 0, p, q, and $(p+q)$, each with the algebraic multiplicity 1, q^2-1, p^2-1, and 1. From Eq. (17), for any unit vector $\boldsymbol{u} \in \mathbb{R}^{p^2+q^2} \setminus \operatorname{null}(\boldsymbol{H})$,

$$\lambda^2_{\boldsymbol{W},\min}(\boldsymbol{u}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{u}) \le \boldsymbol{u}^T\boldsymbol{P}^T(\boldsymbol{W}\otimes\boldsymbol{W})\boldsymbol{P}\boldsymbol{u} \le \lambda^2_{\boldsymbol{W},\max}(\boldsymbol{u}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{u}),$$

where $\lambda_{\boldsymbol{W},\min} = (\lambda_{\boldsymbol{\Theta},p} + \lambda_{\boldsymbol{\Psi},q})^{-1}$ and $\lambda_{\boldsymbol{W},\max} = (\lambda_{\boldsymbol{\Theta},1} + \lambda_{\boldsymbol{\Psi},1})^{-1}$ are the minimum and maximum eigenvalues of $\boldsymbol{W}$. Combining this with the minimum eigenvalue $\min\{p,q\}$ and the maximum eigenvalue $(p+q)$ of $\boldsymbol{P}^T\boldsymbol{P}$ outside of the null space of $\boldsymbol{H}$ results in

$$\min\{p,q\}\lambda^2_{\boldsymbol{W},\min} \le \lambda^2_{\boldsymbol{W},\min}(\boldsymbol{u}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{u}) \quad \text{and} \quad \lambda^2_{\boldsymbol{W},\max}(\boldsymbol{u}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{u}) \le (p+q)\lambda^2_{\boldsymbol{W},\max},$$

which completes the proof. ■

3.3 EiGLasso with Approximate Hessian

EiGLasso pre-computes and stores $\boldsymbol{G}$ after eigendecomposing $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ at the beginning of the coordinate descent to solve Eq. (12). However, explicitly computing and storing the $(p^2+q^2)\times(p^2+q^2)$ matrix $\boldsymbol{H}$ is still expensive for large p and q even with the eigendecomposition. To further reduce the computation time and memory, we approximate $\boldsymbol{H}$ based on its form in Theorem 3 as follows. We first drop $\boldsymbol{H}_{\boldsymbol{\Theta}\boldsymbol{\Psi}}$ while keeping only $\boldsymbol{H}_{\boldsymbol{\Theta}}$ and $\boldsymbol{H}_{\boldsymbol{\Psi}}$. Since pre-computing and storing the $p^2\times p^2$ matrix $\boldsymbol{H}_{\boldsymbol{\Theta}}$ and the $q^2\times q^2$ matrix $\boldsymbol{H}_{\boldsymbol{\Psi}}$ is still expensive, we approximate $\boldsymbol{H}_{\boldsymbol{\Theta}}$ and $\boldsymbol{H}_{\boldsymbol{\Psi}}$ by approximating their eigenvalues $\boldsymbol{\Lambda}_{\boldsymbol{H}_{\boldsymbol{\Theta}}}$ and $\boldsymbol{\Lambda}_{\boldsymbol{H}_{\boldsymbol{\Psi}}}$ in Eq. (19), while keeping the same eigenvectors $\boldsymbol{Q}_{\boldsymbol{\Theta}}\otimes\boldsymbol{Q}_{\boldsymbol{\Theta}}$ and $\boldsymbol{Q}_{\boldsymbol{\Psi}}\otimes\boldsymbol{Q}_{\boldsymbol{\Psi}}$. The resulting approximate Hessian $\hat{\boldsymbol{H}}$ is

$$\hat{\boldsymbol{H}} = \begin{bmatrix}\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}} & \boldsymbol{0}\\ \boldsymbol{0} & \hat{\boldsymbol{H}}_{\boldsymbol{\Psi}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{Q}_{\boldsymbol{\Theta}}\otimes\boldsymbol{Q}_{\boldsymbol{\Theta}} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_{\boldsymbol{\Psi}}\otimes\boldsymbol{Q}_{\boldsymbol{\Psi}}\end{bmatrix}\begin{bmatrix}\hat{\boldsymbol{\Lambda}}_{\boldsymbol{H}_{\boldsymbol{\Theta}}} & \boldsymbol{0}\\ \boldsymbol{0} & \hat{\boldsymbol{\Lambda}}_{\boldsymbol{H}_{\boldsymbol{\Psi}}}\end{bmatrix}\begin{bmatrix}\boldsymbol{Q}_{\boldsymbol{\Theta}}\otimes\boldsymbol{Q}_{\boldsymbol{\Theta}} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_{\boldsymbol{\Psi}}\otimes\boldsymbol{Q}_{\boldsymbol{\Psi}}\end{bmatrix}^T, \tag{23}$$

where

$$\begin{aligned}
\hat{\boldsymbol{\Lambda}}_{\boldsymbol{H}_{\boldsymbol{\Theta}}} &= \sum_{k=1}^{K}\boldsymbol{\Xi}_{\boldsymbol{\Theta},k}\otimes\boldsymbol{\Xi}_{\boldsymbol{\Theta},k} + (q-K)\boldsymbol{\Xi}_{\boldsymbol{\Theta},K}\otimes\boldsymbol{\Xi}_{\boldsymbol{\Theta},K},\\
\hat{\boldsymbol{\Lambda}}_{\boldsymbol{H}_{\boldsymbol{\Psi}}} &= \sum_{l=1}^{K}\boldsymbol{\Xi}_{\boldsymbol{\Psi},l}\otimes\boldsymbol{\Xi}_{\boldsymbol{\Psi},l} + (p-K)\boldsymbol{\Xi}_{\boldsymbol{\Psi},K}\otimes\boldsymbol{\Xi}_{\boldsymbol{\Psi},K}.
\end{aligned} \tag{24}$$

In $\hat{\boldsymbol{\Lambda}}_{\boldsymbol{H}_{\boldsymbol{\Theta}}}$, we keep only the components $\boldsymbol{\Xi}_{\boldsymbol{\Theta},k}\otimes\boldsymbol{\Xi}_{\boldsymbol{\Theta},k}$'s of $\boldsymbol{\Lambda}_{\boldsymbol{H}_{\boldsymbol{\Theta}}}$ in Eq. (20) that contain the K smallest eigenvalues $\lambda_{\boldsymbol{\Psi},1},\ldots,\lambda_{\boldsymbol{\Psi},K}$. These smallest eigenvalues have the largest contribution toward $\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}$, since the eigenvalues of $\boldsymbol{\Psi}$ contribute to the eigenvalues of $\boldsymbol{H}_{\boldsymbol{\Theta}}$ through their inverses. We replace the remaining $(q-K)$ components with $\boldsymbol{\Xi}_{\boldsymbol{\Theta},K} \otimes \boldsymbol{\Xi}_{\boldsymbol{\Theta},K}$, because dropping the $(q-K)$ eigenvalue components in $\boldsymbol{\Lambda}_{\boldsymbol{H}_{\boldsymbol{\Theta}}}$ would amount to assuming that $\boldsymbol{\Psi}$ has $(q-K)$ eigenvalues of infinite magnitude. This approximation leads to the following form

$$\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}} = \sum_{k=1}^{K} \boldsymbol{V}_{\boldsymbol{\Theta},k} \otimes \boldsymbol{V}_{\boldsymbol{\Theta},k} + (q-K)\boldsymbol{V}_{\boldsymbol{\Theta},K} \otimes \boldsymbol{V}_{\boldsymbol{\Theta},K},$$

$$\hat{\boldsymbol{H}}_{\boldsymbol{\Psi}} = \sum_{k=1}^{K} \boldsymbol{V}_{\boldsymbol{\Psi},k} \otimes \boldsymbol{V}_{\boldsymbol{\Psi},k} + (q-K)\boldsymbol{V}_{\boldsymbol{\Psi},K} \otimes \boldsymbol{V}_{\boldsymbol{\Psi},K},$$

where $\boldsymbol{V}_{\boldsymbol{\Theta},k} = \boldsymbol{Q}_{\boldsymbol{\Theta}} \boldsymbol{\Xi}_{\boldsymbol{\Theta},k} \boldsymbol{Q}_{\boldsymbol{\Theta}}^T$ and $\boldsymbol{V}_{\boldsymbol{\Psi},k} = \boldsymbol{Q}_{\boldsymbol{\Psi}} \boldsymbol{\Xi}_{\boldsymbol{\Psi},k} \boldsymbol{Q}_{\boldsymbol{\Psi}}^T$. Then, during the coordinate descent optimization, we pre-compute and store $\boldsymbol{V}_{\boldsymbol{\Theta},k}$'s and $\boldsymbol{V}_{\boldsymbol{\Psi},k}$'s for $k = 1, \ldots, K$, and compute the Hessian entries from $\boldsymbol{V}_{\boldsymbol{\Theta},k}$'s and $\boldsymbol{V}_{\boldsymbol{\Psi},k}$'s as needed in each coordinate descent update (details in Appendix B). We set $\boldsymbol{H}_{\boldsymbol{\Theta}\boldsymbol{\Psi}} = \boldsymbol{0}$, because the same strategies used to approximate $\boldsymbol{\Lambda}_{\boldsymbol{\Theta}}$ and $\boldsymbol{\Lambda}_{\boldsymbol{\Psi}}$ cannot be applied to approximate $\boldsymbol{\Upsilon}$ in Eq. (20). Approximating $\boldsymbol{H}_{\boldsymbol{\Theta}\boldsymbol{\Psi}}$ by replacing the $(q-K)$ or $(p-K)$ eigenvalue components with the Kth component does not guarantee the resulting approximate Hessian to be positive definite, which is a condition required to guarantee EiGLasso to converge as we show in Section 4. We assume the same K for both $\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}$ and $\hat{\boldsymbol{H}}_{\boldsymbol{\Psi}}$, though this can be relaxed. In our experiments in Section 5, we demonstrate that often $K=1$ suffices.

We provide a geometric interpretation of our approximation of $\boldsymbol{H}_{\boldsymbol{\Theta}}$ with $\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}$. Assuming Eq. (12) without the L_1-regularization, we have the descent direction $-\boldsymbol{H}_{\boldsymbol{\Theta}}^{-1}\operatorname{vec}(\boldsymbol{G}_{\boldsymbol{\Theta}})$ with the exact Hessian and $-\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}^{-1}\operatorname{vec}(\boldsymbol{G}_{\boldsymbol{\Theta}})$ with the approximate Hessian. Since $\boldsymbol{H}_{\boldsymbol{\Theta}}$ and $\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}$ share the same eigenvectors, in the coordinate system defined by these eigenvectors as bases, the descent directions become $-\boldsymbol{H}_{\boldsymbol{\Theta}}^{-1/2}\operatorname{vec}(\boldsymbol{G}_{\boldsymbol{\Theta}}) = -\boldsymbol{\Lambda}_{\boldsymbol{H}_{\boldsymbol{\Theta}}}^{-\frac{1}{2}}(\boldsymbol{Q}_{\boldsymbol{\Theta}} \otimes \boldsymbol{Q}_{\boldsymbol{\Theta}})^T\operatorname{vec}(\boldsymbol{G}_{\boldsymbol{\Theta}})$ with the exact Hessian and $-\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}^{-\frac{1}{2}}\operatorname{vec}(\boldsymbol{G}_{\boldsymbol{\Theta}}) = -\hat{\boldsymbol{\Lambda}}_{\boldsymbol{H}_{\boldsymbol{\Theta}}}^{-\frac{1}{2}}(\boldsymbol{Q}_{\boldsymbol{\Theta}} \otimes \boldsymbol{Q}_{\boldsymbol{\Theta}})^T\operatorname{vec}(\boldsymbol{G}_{\boldsymbol{\Theta}})$ with the approximate Hessian (Boyd and Vandenberghe, 2004). The latter is the former scaled in each element of the matrix. Furthermore, since $[\hat{\boldsymbol{\Lambda}}_{\boldsymbol{H}_{\boldsymbol{\Theta}}}^{-\frac{1}{2}}]_{ii} \leq [\boldsymbol{\Lambda}_{\boldsymbol{H}_{\boldsymbol{\Theta}}}^{-\frac{1}{2}}]_{ii}$, in the coordinate system defined by the eigenvectors of $\boldsymbol{H}_{\boldsymbol{\Theta}}$ and $\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}$ as bases, each element of the descent direction with $\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}$ is a convex combination of 0 and the corresponding element of the descent direction with $\boldsymbol{H}_{\boldsymbol{\Theta}}$.

The following theorem provides the properties of $\hat{\boldsymbol{H}}$, which will be used to analyze the convergence of EiGLasso in Section 4.

Theorem 5 *The approximate Hessian $\hat{\boldsymbol{H}}$ is positive definite. Furthermore, its minimum and maximum eigenvalues are*

$$\lambda_{\hat{\boldsymbol{H}},\min} = \min\left\{\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}},\min}, \lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Psi}},\min}\right\}, \quad \lambda_{\hat{\boldsymbol{H}},\max} = \max\left\{\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}},\max}, \lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Psi}},\max}\right\},$$

where

$$\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}},\min} = \sum_{i=1}^{K} (\lambda_{\boldsymbol{\Theta},p} + \lambda_{\boldsymbol{\Psi},i})^{-2} + (q-K)(\lambda_{\boldsymbol{\Theta},p} + \lambda_{\boldsymbol{\Psi},K})^{-2}$$

$$\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}},\max} = \sum_{i=1}^{K}(\lambda_{\boldsymbol{\Theta},1} + \lambda_{\boldsymbol{\Psi},i})^{-2} + (q-K)(\lambda_{\boldsymbol{\Theta},1} + \lambda_{\boldsymbol{\Psi},K})^{-2}$$
$$\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Psi}},\min} = \sum_{j=1}^{K}(\lambda_{\boldsymbol{\Theta},j} + \lambda_{\boldsymbol{\Psi},q})^{-2} + (p-K)(\lambda_{\boldsymbol{\Theta},K} + \lambda_{\boldsymbol{\Psi},q})^{-2}$$
$$\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Psi}},\max} = \sum_{j=1}^{K}(\lambda_{\boldsymbol{\Theta},j} + \lambda_{\boldsymbol{\Psi},1})^{-2} + (p-K)(\lambda_{\boldsymbol{\Theta},K} + \lambda_{\boldsymbol{\Psi},1})^{-2}.$$

Proof Since a block-diagonal matrix has the same set of eigenvalues as its diagonal blocks, the eigenvalues of $\hat{\boldsymbol{H}}$ are the same as the union of those of $\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}$ and $\hat{\boldsymbol{H}}_{\boldsymbol{\Psi}}$ in Eq. (24). The eigenvalues of $\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}$ and $\hat{\boldsymbol{H}}_{\boldsymbol{\Psi}}$ in Eq. (24) are positive, because they are derived from the eigenvalues of $\boldsymbol{W}$ that are positive. Thus, $\hat{\boldsymbol{H}}$ is positive definite. ■

3.4 Estimation with the Unidentifiable Parameters

We describe a simple approach to estimating the unidentifiable diagonals of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ in EiGLasso. We show that to solve the optimization problem with the constraint on a fixed trace ratio in Eq. (11), it is sufficient to solve the unconstrained problem in Eq. (3) and to adjust the diagonals once to enforce the given trace ratio after EiGLasso converges.

In Lemma 6 below, we show that given the current estimate of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$, the quadratic approximation in Eq. (12) for determining descent directions is uniquely defined.

Lemma 6 *Within the same equivalence class that contains the current estimate of* $(\boldsymbol{\Theta}, \boldsymbol{\Psi})$*,* $\boldsymbol{G}$*,* $\boldsymbol{H}$*, and* $\hat{\boldsymbol{H}}$ *are uniquely defined. Thus, the second-order approximation in Eq.* (12) *is uniquely defined.*

Proof According to Eq. (22), the unidentifiability in the diagonal elements of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ reduces to the unidentifiability in the eigenvalues of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$. In the gradient in Eq. (18) and Hessians in Eqs. (19) and (23), the eigenvalues of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ always appear as a pair, where the shift c cancels out as follows:

$$\big((\lambda_{\boldsymbol{\Theta},l} + c) + (\lambda_{\boldsymbol{\Psi},k} - c)\big)^{-1} = (\lambda_{\boldsymbol{\Theta},l} + \lambda_{\boldsymbol{\Psi},k})^{-1}, \forall (l,k) \in \{1,\dots,p\} \times \{1,\dots,q\}.$$

Thus, $\boldsymbol{G}$, $\boldsymbol{H}$, and $\hat{\boldsymbol{H}}$ are unaffected by the shift c in the diagonals of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$. ■

Lemma 6 allows us to solve the problem in Eq. (12) to obtain the descent directions $(D_{\boldsymbol{\Theta}}, D_{\boldsymbol{\Psi}})$, ignoring the constraint on the trace ratio. However, if $(D_{\boldsymbol{\Theta}}, D_{\boldsymbol{\Psi}})$ is a solution to the problem in Eq. (12), then $(D_{\boldsymbol{\Theta}} - c\boldsymbol{I}, D_{\boldsymbol{\Psi}} + c\boldsymbol{I})$ for $c \in \mathbb{R}$ is also a solution, forming an equivalence class

$$S[D_{\boldsymbol{\Theta}} \oplus D_{\boldsymbol{\Psi}}] = \{(D_{\boldsymbol{\Theta}}, D_{\boldsymbol{\Psi}}) |\, (D_{\boldsymbol{\Theta}} - c\boldsymbol{I}_p) \oplus (D_{\boldsymbol{\Psi}} + c\boldsymbol{I}_q) = D_{\boldsymbol{\Theta}} \oplus D_{\boldsymbol{\Psi}}, c \in \mathbb{R}\}. \qquad (25)$$

Lemma 7 *Within the equivalence class in Eq.* (25)*, the line-search method in Algorithm 1 identifies a unique step size* α*.*

Proof We only need to show that $[\mathrm{vec}(D_{\boldsymbol{\Theta}})^T, \mathrm{vec}(D_{\boldsymbol{\Psi}})^T]\,\mathrm{vec}(\boldsymbol{G})$, a term that appears in the computation of δ in Algorithm 1, is invariant within the equivalence class in Eq. (25), since $D_{\boldsymbol{\Theta}}$ and $D_{\boldsymbol{\Psi}}$ always appear as the Kronecker sum of the two in all the other parts of the line search such that c and $-c$ in Eq. (25) cancel out. We have $\mathrm{tr}(\boldsymbol{G}_{\boldsymbol{\Theta}}) = \mathrm{tr}(\boldsymbol{G}_{\boldsymbol{\Psi}})$, which can be directly verified from Eqs. (15) and (16) in Lemma 2 as $\mathrm{tr}(\boldsymbol{G}_{\boldsymbol{\Theta}}) = \sum_{i=1}^{n} \frac{1}{n}\mathrm{tr}(\boldsymbol{Y}^{iT}\boldsymbol{Y}^{i}) - \mathrm{tr}(\boldsymbol{W}) = \mathrm{tr}(\boldsymbol{G}_{\boldsymbol{\Psi}})$. From this, we have $[\mathrm{vec}(D_{\boldsymbol{\Theta}})^T, \mathrm{vec}(D_{\boldsymbol{\Psi}})^T]\,\mathrm{vec}(\boldsymbol{G}) = \mathrm{tr}(D_{\boldsymbol{\Theta}}\boldsymbol{G}_{\boldsymbol{\Theta}}) + \mathrm{tr}(D_{\boldsymbol{\Psi}}\boldsymbol{G}_{\boldsymbol{\Psi}}) = \mathrm{tr}((D_{\boldsymbol{\Theta}} - c\boldsymbol{I})\boldsymbol{G}_{\boldsymbol{\Theta}}) + \mathrm{tr}((D_{\boldsymbol{\Psi}} + c\boldsymbol{I})\boldsymbol{G}_{\boldsymbol{\Psi}}) = \mathrm{tr}(D_{\boldsymbol{\Theta}}) + \mathrm{tr}(D_{\boldsymbol{\Psi}})$, which proves $[\mathrm{vec}(D_{\boldsymbol{\Theta}})^T, \mathrm{vec}(D_{\boldsymbol{\Psi}})^T]\,\mathrm{vec}(\boldsymbol{G})$ is unaffected by the unidentifiable diagonals of $D_{\boldsymbol{\Theta}}$ and $D_{\boldsymbol{\Psi}}$. ■

With Lemmas 6 and 7, we can minimize the EiGLasso objective in Eq. (11) by updating $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ with $(D_{\boldsymbol{\Theta}}, D_{\boldsymbol{\Psi}})$ and step size α and adjusting the diagonals of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ using Theorem 1 after each update. The theorem below shows this procedure can be simplified even further.

Theorem 8 *In EiGLasso, given the trace ratio $\rho = \mathrm{tr}(\boldsymbol{\Psi})/\,\mathrm{tr}(\boldsymbol{\Theta})$, it is sufficient to identify the diagonals of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ only once after convergence. This leads to the identical estimate obtained by identifying the diagonals of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ in every iteration to maintain the trace ratio ρ. At convergence, the diagonal elements of $(\boldsymbol{\Theta}, \boldsymbol{\Psi})$ are adjusted by the scalar factor*

$$\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} - \frac{\mathrm{tr}(\boldsymbol{\Psi}) - \rho\,\mathrm{tr}(\boldsymbol{\Theta})}{q + \rho p}\boldsymbol{I}_p. \tag{26}$$

Proof Regardless of which member of the equivalence class $S[D_{\boldsymbol{\Theta}} \oplus D_{\boldsymbol{\Psi}}]$ is used to update the current estimate of $(\boldsymbol{\Theta}, \boldsymbol{\Psi})$, we arrive at the same equivalence class $S[\boldsymbol{\Theta} \oplus \boldsymbol{\Psi}]$ for the estimate of $(\boldsymbol{\Theta}, \boldsymbol{\Psi})$ after the update. From Lemma 6, given this equivalence class $S[\boldsymbol{\Theta} \oplus \boldsymbol{\Psi}]$, the problem in Eq. (12) in the next iteration is uniquely defined. Thus, regardless of whether we adjust the diagonals to meet the constraint on the trace ratio, the sequence of equivalence class $S[\boldsymbol{\Theta} \oplus \boldsymbol{\Psi}]$ over iterations is the same, and it is not necessary to identify the diagonals in each iteration of Newton's method. The adjustment in Eq. (26) is found by applying Theorem 1 to $\boldsymbol{\Theta} \oplus \boldsymbol{\Psi}$ with the current estimate of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$. ■

When we set $\rho = \frac{q}{p}$, the one-time identification of diagonal parameters with Eq. (26) in EiGLasso becomes identical to the identification of the parameters that TeraLasso (Greenewald et al., 2019) performs at the end of every iteration. At the end of each iteration, TeraLasso evenly distributes $\tau = \frac{\mathrm{tr}(\boldsymbol{\Theta} \oplus \boldsymbol{\Psi})}{pq}$ between the reparameterized $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ in Eq. (5) and performs the update $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} - \frac{\mathrm{tr}(\boldsymbol{\Theta})}{p}\boldsymbol{I}_p + \frac{\tau}{2}\boldsymbol{I}_p$ and similarly for $\boldsymbol{\Psi}$. It is straightforward to show that with $\rho = \frac{q}{p}$ and from Eq. (10), Eq. (26) reduces to this update in TeraLasso. Theorem 8 is analogous to the simple approach to handling the unidentifiability of the parameters in Kronecker-product inverse covariance estimation (Yin and Li, 2012), where $\boldsymbol{\Theta} \otimes \boldsymbol{\Psi} = (c\boldsymbol{\Theta}) \otimes (\frac{1}{c}\boldsymbol{\Psi})$ for any positive constant c. The parameters are identified by rescaling $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ as $\boldsymbol{\Theta} \leftarrow c\boldsymbol{\Theta}$ and $\boldsymbol{\Psi} \leftarrow \frac{1}{c}\boldsymbol{\Psi}$ with some constant c such that $\boldsymbol{\Theta}_{11}$ is equal to 1 after convergence, similar to the one-time identification in EiGLasso.

In practice, since our optimization algorithm is not affected by the choice of the trace ratio ρ, one can set ρ to any desired quantity and set the diagonals of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ after the optimization is complete. A possible choice for ρ is the trace ratio of the inverse of the empirical covariance matrices $\boldsymbol{S}$ and $\boldsymbol{T}$, or the inverse of the trace ratio of $\boldsymbol{S}$ and $\boldsymbol{T}$ for singular $\boldsymbol{S}$ and $\boldsymbol{T}$.

3.5 Active Set and Automatic Detection of Block-diagonal Structure

In the graphical lasso, a simple strategy for reducing computation time has been introduced that detects the block-diagonal structure in the inverse covariance parameter from the sample covariance matrix prior to estimation (Witten et al., 2011; Mazumder and Hastie, 2012). Then, only the parameters within the blocks corresponding to the connected components in the graph need to be estimated. In Theorem 9 below, using Lemma 2 and Theorem 3, we show that a similar strategy can be applied to EiGLasso with both the exact and approximate Hessian, to detect the block diagonal structures in $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ from the sufficient statistics $\boldsymbol{S}$ and $\boldsymbol{T}$.

Theorem 9 *The block-diagonal structure of $\boldsymbol{\Theta}$ can be detected by thresholding $\boldsymbol{S}$ such that $[\boldsymbol{\Theta}]_{ij} = 0$ iff $|[\boldsymbol{S}]_{ij}| \leq \gamma_{\boldsymbol{\Theta}}$. Similarly, the block-diagonal structure of $\boldsymbol{\Psi}$ can be detected by thresholding $\boldsymbol{T}$ such that $[\boldsymbol{\Psi}]_{ij} = 0$ iff $|[\boldsymbol{T}]_{ij}| \leq \gamma_{\boldsymbol{\Psi}}$.*

Proof Let $\partial|\cdot|$ denote the subgradient of the L_1 norm of a matrix, i.e., $[\partial|\boldsymbol{A}|]_{ij}$ is 1 if $[\boldsymbol{A}]_{ij} > 0$, -1 if $[\boldsymbol{A}]_{ij} < 0$, and $[\partial|\boldsymbol{A}|]_{ij} \in [-1, 1]$ if $[\boldsymbol{A}]_{ij} = 0$. Then, the Karush-Kuhn-Tucker conditions (Boyd and Vandenberghe, 2004; Witten et al., 2011) for $\boldsymbol{\Theta}$ in Eq. (3) is

$$\boldsymbol{W}_{\boldsymbol{\Theta}} - q\boldsymbol{S} - q\gamma_{\boldsymbol{\Theta}}\partial|\boldsymbol{\Theta}| = 0, \tag{27}$$

where $\boldsymbol{W}_{\boldsymbol{\Theta}}$ is given in Eq. (16). If $\boldsymbol{\Theta}$ is block-diagonal, $\boldsymbol{W}_{\boldsymbol{\Theta}}$ is also block-diagonal, since $\boldsymbol{\Theta}$ and $\boldsymbol{W}_{\boldsymbol{\Theta}}$ have the same eigenvectors according to Lemma 2 and Theorem 3. This implies that if $[\boldsymbol{\Theta}]_{ij} = 0$ in the off-diagonal blocks, $|[\boldsymbol{S}]_{ij}| \leq \gamma_{\boldsymbol{\Theta}}$ in Eq. (27). The case for $\boldsymbol{\Psi}$ can be proved similarly. ■

Hsieh et al. (2014) further showed that their active-set strategy in QUIC amounts to detecting a block-diagonal structure in the first iteration, if the parameters are initialized to a diagonal matrix. This strategy can be extended to EiGLasso with the approximate Hessian. To reduce the computation time, as in QUIC, in each Newton iteration, EiGLasso detects the active set of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$

$$\mathcal{A}_{\boldsymbol{\Theta}} = \{(i,j) \mid [\boldsymbol{\Theta}]_{ij} \neq 0 \text{ or } |[\boldsymbol{G}_{\boldsymbol{\Theta}}]_{ij}| > q\gamma_{\boldsymbol{\Theta}}\}, \;\; \mathcal{A}_{\boldsymbol{\Psi}} = \{(i,j) \mid [\boldsymbol{\Psi}]_{ij} \neq 0 \text{ or } |[\boldsymbol{G}_{\boldsymbol{\Psi}}]_{ij}| > p\gamma_{\boldsymbol{\Psi}}\},$$

and update only the parameters in the active sets during the coordinate descent optimization, while setting those in the fixed set to zero. When $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ are initialized to diagonal matrices, the approximate Hessian $\hat{\boldsymbol{H}}$ in iteration 1 is diagonal, since the eigenvector matrices $\boldsymbol{Q}_{\boldsymbol{\Theta}}$ and $\boldsymbol{Q}_{\boldsymbol{\Psi}}$ are diagonal. Then, the optimization problem in Eq. (12) decouples into a set of optimization problems, each of which involves a single element of $(D_{\boldsymbol{\Theta}}, D_{\boldsymbol{\Psi}})$ and has a closed-form solution for $[D_{\boldsymbol{\Theta}}]_{ij}$ as

$$[D_{\boldsymbol{\Theta}}]_{ij} = -q\left(\sum_{k=1}^{K}[\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{ii}[\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{jj} + (q-K)[\boldsymbol{V}_{\boldsymbol{\Theta},K}]_{ii}[\boldsymbol{V}_{\boldsymbol{\Theta},K}]_{jj}\right)^{-1} \mathcal{S}\left([\boldsymbol{S}]_{ij}, \gamma_{\boldsymbol{\Theta}}\right), \quad \forall i \neq j.$$

The soft-thresholding operator $\mathcal{S}(a, b)$ above is defined as $\mathcal{S}(a, b) = \text{sign}(a)(|a| - b)_+$, where $(c)_+ = 0$ if $c < 0$ and $(c)_+ = c$ if $c > 0$. This closed-form solution is $[D_{\boldsymbol{\Theta}}]_{ij} = 0$ if $|[\boldsymbol{S}]_{ij}| < \gamma_{\boldsymbol{\Theta}}$, which is equivalent to the condition for detecting the block-diagonal structure in $\boldsymbol{\Theta}$ from $\boldsymbol{S}$ in Theorem 9. The case for $\boldsymbol{\Psi}$ can be shown similarly.

4. Convergence Analysis

We examine the properties of the line-search method in Algorithm 1 and analyze the global and local convergence of EiGLasso. We use the properties of the exact and approximate Hessian matrices that we proved in Section 3 to extend the previous results on the convergence behavior of QUIC (Hsieh et al., 2014).

4.1 Line Search Properties

EiGLasso inherits some of the line-search properties shown for QUIC (Hsieh et al., 2014). This is because our objective in Eq. (3) can be written in terms of $\boldsymbol{\Omega} = \boldsymbol{\Theta} \oplus \boldsymbol{\Psi}$ in the form that resembles the objective of QUIC

$$f(\boldsymbol{\Omega}) = \text{tr}(\boldsymbol{U}\boldsymbol{\Omega}) - \log|\boldsymbol{\Omega}| + \|\boldsymbol{\Gamma} \circ \boldsymbol{\Omega}\|_{1,\text{off}},$$

where $\boldsymbol{U} = \frac{1}{n}\sum_{i=1}^{n} \text{vec}(\boldsymbol{Y}^i)^T \text{vec}(\boldsymbol{Y}^i)$, $\boldsymbol{\Gamma} = \gamma_{\boldsymbol{\Theta}} \mathbb{1}_p \oplus \gamma_{\boldsymbol{\Psi}} \mathbb{1}_q$, and $\circ$ is an element-wise multiplication operator. Then, the EiGLasso's update $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} + \alpha D_{\boldsymbol{\Theta}}$ and $\boldsymbol{\Psi} \leftarrow \boldsymbol{\Psi} + \alpha D_{\boldsymbol{\Psi}}$ can be written as $\boldsymbol{\Omega} \leftarrow \boldsymbol{\Omega} + \alpha D_{\boldsymbol{\Omega}}$, where $D_{\boldsymbol{\Omega}} = D_{\boldsymbol{\Theta}} \oplus D_{\boldsymbol{\Psi}}$, since $\boldsymbol{\Omega} + \alpha D_{\boldsymbol{\Omega}} = (\boldsymbol{\Theta} + \alpha D_{\boldsymbol{\Theta}}) \oplus (\boldsymbol{\Psi} + \alpha D_{\boldsymbol{\Psi}})$. We extend these results for $\boldsymbol{\Omega}$, to prove the results for individual $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$, where $(\boldsymbol{\Theta}, \boldsymbol{\Psi}) \in \mathbb{KS}^{p,q}$, when unlike in QUIC the exact Hessian $\boldsymbol{H}$ is not positive definite everywhere (Theorem 4), and when the approximate Hessian $\hat{\boldsymbol{H}}$ is used.

We begin by showing that both $\boldsymbol{H}$ and $\hat{\boldsymbol{H}}$ have bounded eigenvalues in the level set $\mathcal{U} = \{(\boldsymbol{\Theta}, \boldsymbol{\Psi}) \mid \boldsymbol{\Theta} \oplus \boldsymbol{\Psi} \in \mathbb{S}_{++}^{pq},\ \ f(\boldsymbol{\Theta}, \boldsymbol{\Psi}) \le f(\boldsymbol{\Theta}^0, \boldsymbol{\Psi}^0)\}$ and are Lipschitz-continuous. This result will be used to prove the line-search properties and global and local convergence.

Lemma 10 *In the level set $\mathcal{U}$, $\boldsymbol{H}$ and $\hat{\boldsymbol{H}}$ are Lipschitz-continuous and have bounded eigenvalues:*

$$\begin{aligned} \min\{p, q\}\bar{\lambda}^{-2}\boldsymbol{I}_{p^2+q^2} \preceq & \boldsymbol{H} \preceq (p+q)\underline{\lambda}^{-2}\boldsymbol{I}_{p^2+q^2}, \\ \min\{p, q\}\bar{\lambda}^{-2}\boldsymbol{I}_{p^2+q^2} \preceq & \hat{\boldsymbol{H}} \preceq \max\{p, q\}\underline{\lambda}^{-2}\boldsymbol{I}_{p^2+q^2}, \end{aligned}$$

for some constants $\underline{\lambda}, \bar{\lambda} > 0$ that depend on $\gamma_{\boldsymbol{\Theta}}$, $\gamma_{\boldsymbol{\Psi}}$, $f(\boldsymbol{\Theta}^0, \boldsymbol{\Psi}^0)$, and $\{\boldsymbol{Y}^1, \dots, \boldsymbol{Y}^n\}$.

Proof We bound the eigenvalues of $\boldsymbol{\Omega} = \boldsymbol{\Theta} \oplus \boldsymbol{\Psi}$, $\boldsymbol{\Theta}$, and $\boldsymbol{\Psi}$, and use these bounds to bound the eigenvalues of $\boldsymbol{H}$ and $\hat{\boldsymbol{H}}$. When the diagonals are not identified, it directly follows from Lemma 2 in Hsieh et al. (2014) that all EiGLasso iterates of $(\boldsymbol{\Theta}, \boldsymbol{\Psi})$ are contained in the set with bounded eigenvalues of $\boldsymbol{\Theta} \oplus \boldsymbol{\Psi}$

$$\mathcal{C} = \{(\boldsymbol{\Theta}, \boldsymbol{\Psi}) \mid \underline{\lambda}\boldsymbol{I} \preceq \boldsymbol{\Theta} \oplus \boldsymbol{\Psi} \preceq \bar{\lambda}\boldsymbol{I}\}. \tag{28}$$

Next, given a fixed trace ratio $\rho = \frac{\text{tr}(\boldsymbol{\Psi})}{\text{tr}(\boldsymbol{\Theta})}$, we bound the eigenvalues of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$. Since Eq. (28) implies $\underline{\lambda}\boldsymbol{I} \preceq \boldsymbol{\Lambda}_{\boldsymbol{\Theta}} \oplus \boldsymbol{\Lambda}_{\boldsymbol{\Psi}} \preceq \bar{\lambda}\boldsymbol{I}$, we apply Theorem 1 to $\boldsymbol{\Lambda}_{\boldsymbol{\Omega}} = \boldsymbol{\Lambda}_{\boldsymbol{\Theta}} \oplus \boldsymbol{\Lambda}_{\boldsymbol{\Psi}}$ to identify $\boldsymbol{\Lambda}_{\boldsymbol{\Theta}}$

and $\boldsymbol{\Lambda_\Psi}$ in the equivalence class in Eq. (22) given ρ, thus, identifying the diagonals of $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$. We bound each of the two terms in Eq. (8) for $\boldsymbol{\Theta}$ as

$$q\underline{\lambda}\boldsymbol{I} \preceq \sum_{i=1}^{q}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})^T \boldsymbol{\Lambda_\Omega}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i}) \preceq q\bar{\lambda}\boldsymbol{I} \quad \text{and} \quad \frac{\rho pq}{q+\rho p}\underline{\lambda}\boldsymbol{I} \preceq \frac{\rho}{q+\rho p}\operatorname{tr}(\boldsymbol{\Lambda_\Omega})\boldsymbol{I} \preceq \frac{\rho pq}{q+\rho p}\bar{\lambda}\boldsymbol{I},$$

and combine these to obtain the eigenvalue bounds for $\boldsymbol{\Theta}$ and similarly for $\boldsymbol{\Psi}$

$$\begin{aligned}\mathcal{C}_\rho = \Big\{(\boldsymbol{\Theta}, \boldsymbol{\Psi}) \mid \Big(\underline{\lambda} - \frac{\rho p}{q+\rho p}\bar{\lambda}\Big)\boldsymbol{I} \preceq \boldsymbol{\Theta} \preceq \Big(\bar{\lambda} - \frac{\rho p}{q+\rho p}\underline{\lambda}\Big)\boldsymbol{I}, \\ \Big(\underline{\lambda} - \frac{q}{q+\rho p}\bar{\lambda}\Big)\boldsymbol{I} \preceq \boldsymbol{\Psi} \preceq \Big(\bar{\lambda} - \frac{q}{q+\rho p}\underline{\lambda}\Big)\boldsymbol{I}\Big\}.\end{aligned} \tag{29}$$

From Eq. (28) and Theorem 4, we obtain the bounds on the eigenvalues of $\boldsymbol{H}$. From Eq. (28) and Theorem 5, we obtain the bounds on the eigenvalues of $\hat{\boldsymbol{H}}$, since $\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}},\min} \geq q\bar{\lambda}^{-2}$, $\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}},\max} \leq q\underline{\lambda}^{-2}$, $\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Psi}},\min} \geq p\bar{\lambda}^{-2}$, $\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Psi}},\max} \leq p\underline{\lambda}^{-2}$. On the set in Eq. (29), since the log-determinant is a continuous function of class C^∞, and a continuous function on a compact set is bounded, both $\boldsymbol{H}$ and $\hat{\boldsymbol{H}}$ are locally Lipschitz-continuous. ■

Now, for EiGLasso with the exact and approximate Hessian matrices, we show that the following three line-search properties hold: the line search method is guaranteed to terminate for any symmetric matrices $D_{\boldsymbol{\Theta}}$ and $D_{\boldsymbol{\Psi}}$, as the two line search conditions in Algorithm 1 are satisfied for some step size α (Lemma 11); the update with the Newton direction is guaranteed to decrease the objective (Lemma 12); and EiGLasso with the exact Hessian is guaranteed to enter pure-Newton phase where the step size $\alpha = 1$ is always chosen. We state and prove the first two properties. The proof of the last property follows directly from the proof in Tseng and Yun (2009) and Hsieh et al. (2014).

Lemma 11 *For any $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$, where $\boldsymbol{\Theta} \oplus \boldsymbol{\Psi} \succ 0$, and symmetric matrices $D_{\boldsymbol{\Theta}}$ and $D_{\boldsymbol{\Psi}}$ for descent directions found with either the exact or approximate Hessian, there exists a step size $\tilde{\alpha} \in (0, 1]$ such that for all $\alpha < \tilde{\alpha}$ the two conditions in the line search in Algorithm 1 are satisfied.*

Proof If $\alpha < \lambda_{\min}(\boldsymbol{\Theta} \oplus \boldsymbol{\Psi})/\|D_{\boldsymbol{\Theta}} \oplus D_{\boldsymbol{\Psi}}\|_2$, the updated estimates $\boldsymbol{\Theta} + \alpha D_{\boldsymbol{\Theta}}$ and $\boldsymbol{\Psi} + \alpha D_{\boldsymbol{\Psi}}$ satisfy $(\boldsymbol{\Theta} + \alpha D_{\boldsymbol{\Theta}}) \oplus (\boldsymbol{\Psi} + \alpha D_{\boldsymbol{\Psi}}) \succ 0$, since $(\boldsymbol{\Theta} + \alpha D_{\boldsymbol{\Theta}}) \oplus (\boldsymbol{\Psi} + \alpha D_{\boldsymbol{\Psi}}) = \boldsymbol{\Theta} \oplus \boldsymbol{\Psi} + \alpha(D_{\boldsymbol{\Theta}} \oplus D_{\boldsymbol{\Psi}})$ and we have $\boldsymbol{\Theta} \oplus \boldsymbol{\Psi} \succ 0$ and $\|\alpha(D_{\boldsymbol{\Theta}} \oplus D_{\boldsymbol{\Psi}})\|_2 < \lambda_{\min}(\boldsymbol{\Theta} \oplus \boldsymbol{\Psi})$. Thus, the first line-search condition in Algorithm 1 is satisfied. From Lemma 1 in Tseng and Yun (2009) and Proposition 3 in Hsieh et al. (2014), it is straightforward to show the second condition in Algorithm 1 is satisfied. ■

Lemma 12 *Let $\operatorname{vec}(D) = [\operatorname{vec}(D_{\boldsymbol{\Theta}})^T, \operatorname{vec}(D_{\boldsymbol{\Psi}})^T]^T$ for all symmetric $D_{\boldsymbol{\Theta}}$ and $D_{\boldsymbol{\Psi}}$. With the exact Hessian $\boldsymbol{H}$, if not at the optimum, δ in Algorithm 1 is upper bounded,*

$$\delta \leq -\operatorname{vec}(D)^T \boldsymbol{H} \operatorname{vec}(D) \leq -\lambda_{\boldsymbol{H},\min_0}\|\operatorname{vec}(D)\|_2^2 < 0, \tag{30}$$

where $\|\operatorname{vec}(D)\|_2^2 = \|D_{\boldsymbol{\Theta}}\|_F^2 + \|D_{\boldsymbol{\Psi}}\|_F^2$*, and* $\lambda_{\boldsymbol{H},\min_0}$ *is given in Theorem 4. With the approximate Hessian* $\hat{\boldsymbol{H}}$*,* δ *is upper bounded everywhere*

$$\delta \leq -\operatorname{vec}(D)^T \hat{\boldsymbol{H}} \operatorname{vec}(D) \leq -\lambda_{\hat{\boldsymbol{H}},\min} \|\operatorname{vec}(D)\|_2^2 < 0, \tag{31}$$

where $\lambda_{\hat{\boldsymbol{H}},\min}$ *is given in Theorem 5.*

Proof For $\boldsymbol{H}$, the first inequality in Eq. (30) can be shown by a straightforward application of Lemma 1 and Theorem 1 in Tseng and Yun (2009) and Proposition 4 in Hsieh et al. (2014). To prove the second and third inequalities, since $\boldsymbol{H}$ is not positive definite everywhere, we need to show that $\operatorname{vec}(D)$ is outside of the null space of $\boldsymbol{H}$ described in Theorem 4, unless EiGLasso is at the optimum, where $D = 0$. The null space of $\boldsymbol{H}$, $\{\operatorname{vec}(D) | \operatorname{vec}(D_{\boldsymbol{\Theta}} \oplus D_{\boldsymbol{\Psi}}) = 0\}$, is equivalent to $\{\operatorname{vec}(D) | D_{\boldsymbol{\Theta}} = c\boldsymbol{I}, D_{\boldsymbol{\Psi}} = -c\boldsymbol{I} \text{ for } c \in \mathbb{R}\}$, which is the equivalence class of the optimality condition $D = 0$. This proves the third inequality in Eq. (30) that holds except when $D = 0$. For $\hat{\boldsymbol{H}}$, the first inequality in Eq. (31) can be again shown from Lemma 1 and Theorem 1 in Tseng and Yun (2009) and Proposition 4 in Hsieh et al. (2014). The second and third inequalities hold since $\hat{\boldsymbol{H}}$ is positive definite. ■

4.2 Convergence Analysis

To show the global convergence of EiGLasso, as in QUIC, we use a more general non-smooth optimization framework, the block coordinate descent studied in Tseng and Yun (2009). EiGLasso satisfies the following two conditions required to guarantee that the block coordinate descent algorithm converges to the global optimum. First, the objective function of EiGLasso has exact or approximate Hessian matrices that are positive definite with bounded eigenvalues in the level set $\mathcal{U}$: $a\boldsymbol{I} \preceq \boldsymbol{H}, \hat{\boldsymbol{H}} \preceq b\boldsymbol{I}$ for some positive constants a and b, according to Theorems 4 and 5, and Lemma 10. Second, EiGLasso with the exact or approximate Hessian chooses a subset of variables to be updated in each iteration according to the *Gauss-Seidel* rule: with $T = 2$, at iteration t, EiGLasso updates one of the two subsets of variables, $\mathcal{J}^{2t} = \mathcal{A}^t_{\boldsymbol{\Theta}} \sqcup \mathcal{A}^t_{\boldsymbol{\Psi}}$ and $\mathcal{J}^{2t+1} = \mathcal{J} \setminus \mathcal{A}^t_{\boldsymbol{\Theta}} \sqcup \mathcal{A}^t_{\boldsymbol{\Psi}}$, where $\mathcal{J} = \mathcal{J}^{2t} \cup \mathcal{J}^{2t+1}$ is the entire set of variables and $\mathcal{A}^t_{\boldsymbol{\Theta}}$ and $\mathcal{A}^t_{\boldsymbol{\Psi}}$ are active sets in Algorithm 2. Therefore, EiGLasso is guaranteed to converge to the global optimum according to Tseng and Yun (2009).

Now we analyze the local convergence of EiGLasso. We adopt a similar strategy used in QUIC (Hsieh et al., 2014): convergence analysis on a smooth function is applied near the global optimum, where the L_1-regularized non-smooth objective becomes locally smooth. The following theorem uses the properties of the Hessian matrices from Section 3 to extend the results for QUIC to EiGLasso.

Theorem 13 *Near the optimum, where step size* $\alpha = 1$ *is chosen, EiGLasso with the exact Hessian* $\boldsymbol{H}$ *converges to the optimum quadratically. EiGLasso with the approximate Hessian* $\hat{\boldsymbol{H}}$ *converges to the optimum linearly.*

Proof Since $\boldsymbol{H}$ is Lipschitz-continuous from Lemma 10, the proof for EiGLasso with the exact Hessian follows from Lemma 2.5 and Theorem 3.1 of Dunn (1980). While the convergence analysis in Dunn (1980) assumes that the Hessian is positive definite, the exact

Hessian in EiGLasso is not positive definite everywhere. However, according to Theorem 4, $\boldsymbol{H}$ is positive definite for all iterates $(\boldsymbol{\Theta}^t, \boldsymbol{\Psi}^t) \in \mathbb{KS}^{pq}_{++}$, since $\mathbb{KS}^{pq}_{++}$ is outside of the null space of $\boldsymbol{H}$, so the analysis in Dunn (1980) can be applied to EiGLasso. The proof for EiGLasso with $\hat{\boldsymbol{H}}$ follows from the analysis of steepest descent with the quadratic norm (Boyd and Vandenberghe, 2004), given the bounded eigenvalues of $\boldsymbol{H}$ in Lemma 10. ■

5. Experiments

We compare the performance of EiGLasso with that of TeraLasso (Greenewald et al., 2019) on simulated data and on real-world data from genomics and finance. TeraLasso is the state-of-the art method for Kronecker-sum inverse covariance estimation and has been shown to be substantially more efficient than BiGLasso (Kalaitzis et al., 2013), so we did not include BiGLasso in our experiments. We implemented EiGLasso in C++ with the sequential version of Intel Math Kernel Library. We downloaded the authors' implementation of TeraLasso and modified it to perform more iterations during line search when the safe-step approach suggested by the authors failed to find a step-size that satisfies the positive definite condition on $\boldsymbol{\Theta} \oplus \boldsymbol{\Psi}$. All experiments were run on a single core of Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz. In all of our experiments, we selected the regularization parameters $\gamma = \gamma_{\boldsymbol{\Theta}} = \gamma_{\boldsymbol{\Psi}}$ for EiGLasso and used the selected γ for TeraLasso because EiGLasso is significantly faster than TeraLasso and minimizes the same objective as TeraLasso. To assess convergence, we used the criterion that the decrease in the objective function value f^t at iteration t satisfies the condition $\left|\frac{f^t - f^{t-1}}{f^t}\right| < \epsilon$ for three consecutive iterations.

5.1 Simulated Data

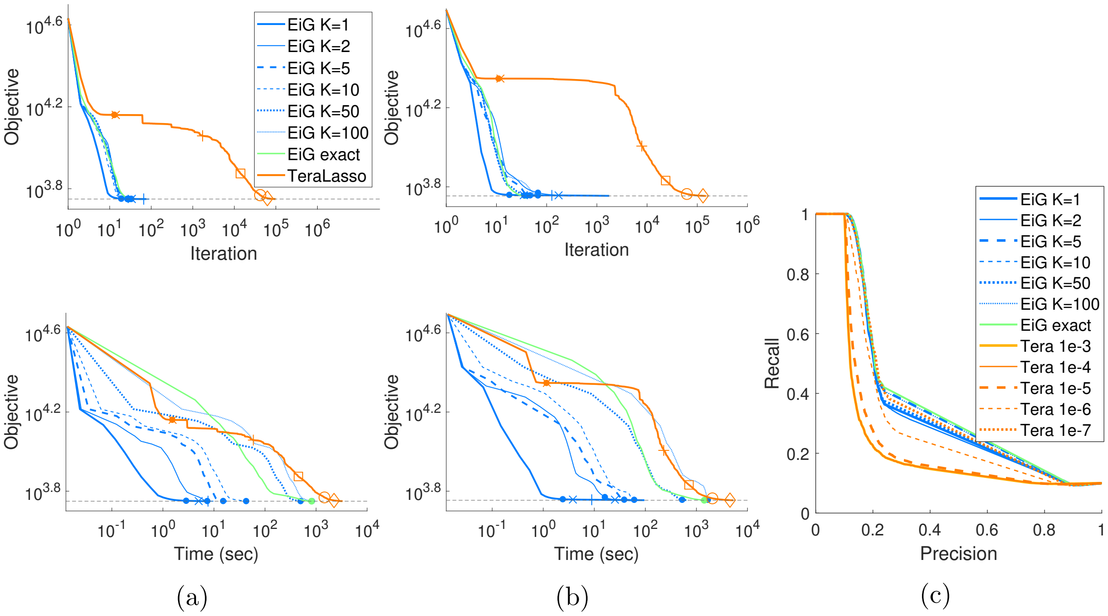
We compared EiGLasso and TeraLasso on data simulated from known $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$. We used the true $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ of different sizes ($p, q = 100$, 200, 500, 1000, 2000, and 5000), assuming two types of graph structures.

- **Random graph**: To set the ground-truth $\boldsymbol{\Theta}$, we first generated a sparse $p \times p$ matrix $\boldsymbol{A}$ by assigning -1, 0, or 1 to each element with probabilities $\frac{1-\kappa}{2}$, κ, and $\frac{1-\kappa}{2}$, respectively. We chose κ such that the number of non-zero elements of $\boldsymbol{\Theta}$ is $10p$. To ensure $\boldsymbol{\Theta}$ is positive definite, we set $\boldsymbol{\Theta}$ to $\boldsymbol{A}\boldsymbol{A}^T$ after adding $\eta + 10^{-4}$ with $\eta \sim \text{Unif}(0, 0.1)$ to each diagonal element of $\boldsymbol{A}\boldsymbol{A}^T$.

- **Graph with clusters**: We set $\boldsymbol{\Theta}$ to a block-diagonal matrix such that each block corresponds to a cluster. For graphs with $p = 100$ and 200, we assumed five blocks, each with size $\frac{p}{5} \times \frac{p}{5}$. For larger graphs with $p = 500$, 1000, 2000, and 5000, we assumed 10 blocks, each with size $\frac{p}{10} \times \frac{p}{10}$. Each block was generated as a random graph described above, setting κ so that we have p non-zero elements in the block.

The ground-truth $\boldsymbol{\Psi}$ was set similarly. Given these parameters, we simulated matrix-variate data from Gaussian distribution with zero mean and inverse covariance $\boldsymbol{\Theta} \oplus \boldsymbol{\Psi}$.

First, we evaluated TeraLasso and EiGLasso on simulated data with graph size $p, q = 100$. The two methods were compared in terms of the computation time and the number of

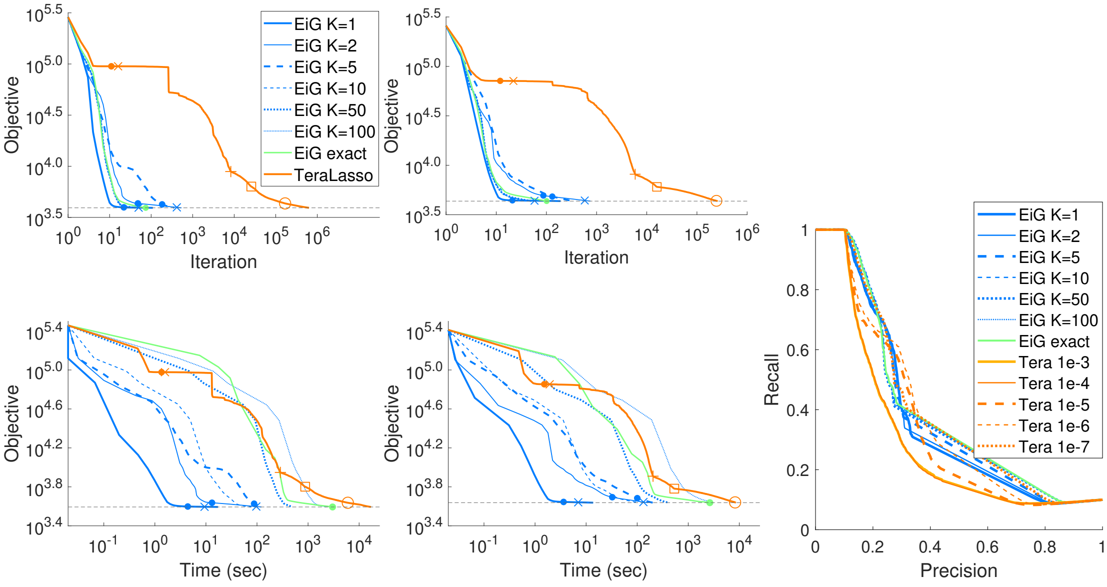


(a) (b) (c)

Figure 2: Comparison of the convergence of EiGLasso and TeraLasso on data simulated from random graphs. In (a) and (b), objective values over iterations (top) and over time (bottom) are shown for two datasets. All methods were run until they reached the objective that EiGLasso with the exact Hessian reached with the convergence criterion $\epsilon = 10^{-3}$. The '•', '×', '+', '□', '◯' and '◇' mark the points that satisfy $\epsilon = 10^{-3}$, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, and 10^{-8}, respectively. In (c), precision-recall curves for EiGLasso at $\epsilon = 10^{-3}$ and TeraLasso at different ϵ's averaged over 10 datasets are shown. Graph size $p, q = 100$ was used.

iterations required to reach the same level of optimality, which we define as the objective value that EiGLasso with the exact Hessian converged to with the convergence criterion $\epsilon = 10^{-3}$. The regularization parameters were selected such that the number of non-zero elements of the estimated parameters roughly matched that of the true parameters. EiGLasso with approximate Hessian was run with different K's ranging from $K = 1$ to $K = 100$. Results are shown for four datasets, two from random graphs (Figures 2(a) and 2(b)) and two from graphs with clusters (Figures 3(a) and 3(b)). Regardless of the degree of Hessian approximation, EiGLasso required significantly fewer iterations than TeraLasso, since methods that use the second-order information in general converge in fewer iterations than first-order methods. With the exact Hessian and approximate Hessian with large K, EiGLasso took longer in each iteration than TeraLasso but overall required two to three times less computation time than TeraLasso. As we reduced K, the time taken by EiGLasso decreased substantially, and EiGLasso with $K = 1$ achieved two to three orders-of-magnitude speed-up, compared to TeraLasso.

We compared TeraLasso and EiGLasso on the accuracy of the recovered non-zero elements in $\mathbf{\Theta}$ and $\mathbf{\Psi}$, using 10 datasets simulated as above (Figures 2(c) and 3(c)). The precision-recall curves averaged over the 10 simulated datasets were obtained for EiGLasso with different K's in the approximate Hessian and with the exact Hessian at $\epsilon = 10^{-3}$ and for TeraLasso with different ϵ's. EiGLasso with smaller K's suffered slightly on accuracy,



(a) (b) (c)

Figure 3: Comparison of the convergence of EiGLasso and TeraLasso on data simulated from graphs with clusters. In (a) and (b), objective values over iterations (top) and over time (bottom) are shown for two datasets. All methods were run until they reached the objective that EiGLasso with the exact Hessian reached with the convergence criterion $\epsilon = 10^{-3}$. The '●', '×', '+', '□', and '◯' mark the points that satisfy $\epsilon = 10^{-3}$, 10^{-4}, 10^{-5}, 10^{-6}, and 10^{-7}, respectively. In (c), precision-recall curves for EiGLasso at $\epsilon = 10^{-3}$ and TeraLasso at different ϵ's averaged over 10 datasets are shown. Graph size $p, q = 100$ was used.

compared to EiGLasso with larger K's, indicating a trade-off between the accuracy and the computation time across different K's. However, even EiGLasso with small K's provided more accurate estimates than TeraLasso with more stringent convergence criteria, and the EiGLasso estimates at $\epsilon = 10^{-3}$ for all K's were significantly better than the TeraLasso estimates at $\epsilon = 10^{-3}$ and 10^{-4}. TeraLasso needed $\epsilon = 10^{-6}$ to achieve the accuracy similar to that of EiGLasso at $\epsilon = 10^{-3}$.

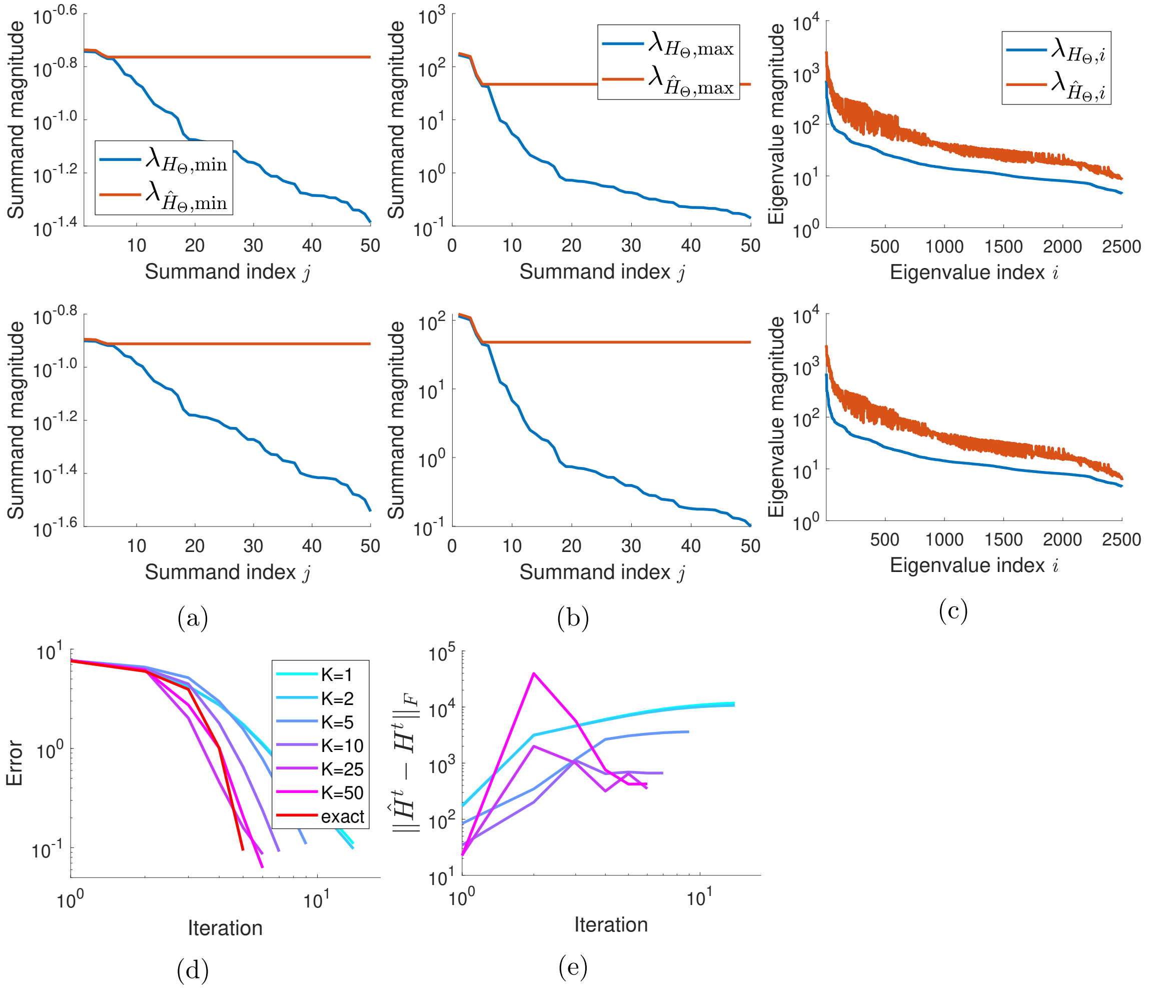
Next, we compared the computation time of TeraLasso and EiGLasso on data simulated from larger graphs (Table 1). For datasets with size $p, q = 100, 200, 500, 1000, 2000$, and 5000, we ran TeraLasso and EiGLasso with varying K's and recorded time taken by each method if it converged within 24 hours. We used $\epsilon = 10^{-3}$ as convergence criterion for EiGLasso with all K's, but ran TeraLasso until it reached the similar objective value that EiGLasso reached with $\epsilon = 10^{-3}$, which typically required $\epsilon = 10^{-6}$, 10^{-7}, or 10^{-8}. Across the different graph types and sizes, EiGLasso with different K's almost always converged faster than TeraLasso. In particular, EiGLasso with $K = 1$ and 2 was consistently two to three orders-of-magnitude faster than TeraLasso. For example, EiGLasso with $K = 1$ took three hours for graphs with $p, q = 5000$, while TeraLasso did not converge on the smaller graphs with $p, q = 500$ in 24 hours.

Graph	p, q	EiGLasso						TeraLasso
		$K = 1$	$K = 2$	$K = 5$	$K = 10$	$K = 50$	$K = 100$	
Random Graphs	100	1	4	6	40	729	1019	4194
	200	7	164	551	3059	27049	70311	16765
	500	29	155	2252	5994	31826		
	1000	515	11601	39568				
	2000	6361	28539					
	5000	10458	62685					
Graphs with Clusters	100	2	21	28	629	6516	12443	21115
	200	11	410	930	1042	11281	59813	67669
	500	28	5231	9252	10549			
	1000	2169	12614	58168				
	2000	9310	41524					
	5000	10461	67608					

Table 1: Comparison of computation time (in seconds) on simulated data.

We empirically evaluated the effects of Hessian approximation on the convergence behavior of EiGLasso by using data simulated from a random graph of size $p, q = 50$. In $\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}$ and $\hat{\boldsymbol{H}}_{\boldsymbol{\Psi}}$ of the approximate Hessian $\hat{\boldsymbol{H}}$, the components of the eigenvalues of $\boldsymbol{H}_{\boldsymbol{\Theta}}$ and $\boldsymbol{H}_{\boldsymbol{\Psi}}$ are modified from the summands in Eq. (20) to the summands in Eq. (24). For EiGLasso with $K = 5$, these summands in the eigenvalues before and after the modification are shown for the minimum and maximum eigenvalues of $\boldsymbol{H}_{\boldsymbol{\Theta}}$ at the 5th and last iterations in Figures 4(a) and 4(b). The full spectrums of the same $\boldsymbol{H}_{\boldsymbol{\Theta}}$ and $\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}$ are shown in Figure 4(c). Next, we examined the effects of such Hessian approximation on the convergence of EiGLasso (Figures 4(d) and 4(e)). While EiGLasso with the exact Hessian converged quadratically, with the approximate Hessian, the convergence slowed down with smaller K. However, even with $K = 1$, the convergence was not significantly slower than EiGLasso with the exact Hessian. Even if $\|\hat{\boldsymbol{H}} - \boldsymbol{H}\|_F$ did not approach zero over iterations (Figure 4(e)), EiGLasso with the approximate Hessian quickly reached the same level of optimality as EiGLasso with the exact Hessian (Figure 4(d)). We obtained similar results from a graph with clusters (Figure 5).

To leverage the fast convergence with $K = 1$ during early iterations and with $K > 1$ during final iterations, we explored the approach of using EiGLasso with $K = 1$ to reach near optimum and then switching to EiGLasso with larger K's to reach the optimum. Using the same simulated datasets in Figures 2(a) and 2(b) and in Figures 3(a) and 3(b), we initialized EiGLasso with $K \geq 1$ using the estimates from EiGLasso with $K = 1$ at $\epsilon = 10^{-3}$ and ran it until it reached the same objective value obtained by EiGLasso with $K = 1$ at $\epsilon = 10^{-6}$. With this initialization scheme, EiGLasso with smaller K's required more iterations but often less computation time (Figure 6). EiGLasso with relatively small K's such as $K = 1, 2, 5$, and 10 required less computation time than EiGLasso with larger K's. This suggests that EiGLasso with $K = 1$ is sufficiently fast during both initial and final



(a)

(b)

(c)

(d)

(e)

Figure 4: Effects of Hessian approximation on the convergence of EiGLasso for data simulated from a random graph. (a) The summands in $\lambda_{\boldsymbol{H_\Theta},\min} = \sum_{j=1}^{q}(\lambda_{\boldsymbol{\Theta},p} + \lambda_{\boldsymbol{\Psi},j})^{-2}$ and $\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}},\min} = \sum_{j=1}^{K}(\lambda_{\boldsymbol{\Theta},p} + \lambda_{\boldsymbol{\Psi},j})^{-2} + (q-K)(\lambda_{\boldsymbol{\Theta},p} + \lambda_{\boldsymbol{\Psi},K})^{-2}$ for EiGLasso with $K = 5$. (b) The summands in $\lambda_{\boldsymbol{H_\Theta},\max}$ and $\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}},\max}$ as in Panel (a). (c) The eigenvalues of $\boldsymbol{H_\Theta}$ and $\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}$, sorted with the eigenvalues of $\boldsymbol{H_\Theta}$. In Panels (a)-(c), the results are shown at the 5th iteration (top) and at the last iteration (bottom). (d) Error measured as $\left\| \begin{bmatrix} \text{vec}(\boldsymbol{\Theta}^t - \boldsymbol{\Theta}^*) \\ \text{vec}(\boldsymbol{\Psi}^t - \boldsymbol{\Psi}^*) \end{bmatrix} \right\|_2$ over iterations. (e) $\|\hat{\boldsymbol{H}} - \boldsymbol{H}\|_F$ over iterations. Graph size $p, q = 50$ was used.

iterations and that the strategy of running EiGLasso with $K = 1$ during initial iterations and with a small increase in K during final iterations gives EiGLasso the optimal performance.

5.2 Mouse Gene Expression Data from RNA-seq

We compared EiGLasso and TeraLasso using the gene expression levels obtained from RNA sequencing of brain tissues for multiple related mice from the same pedigree (Gonzales

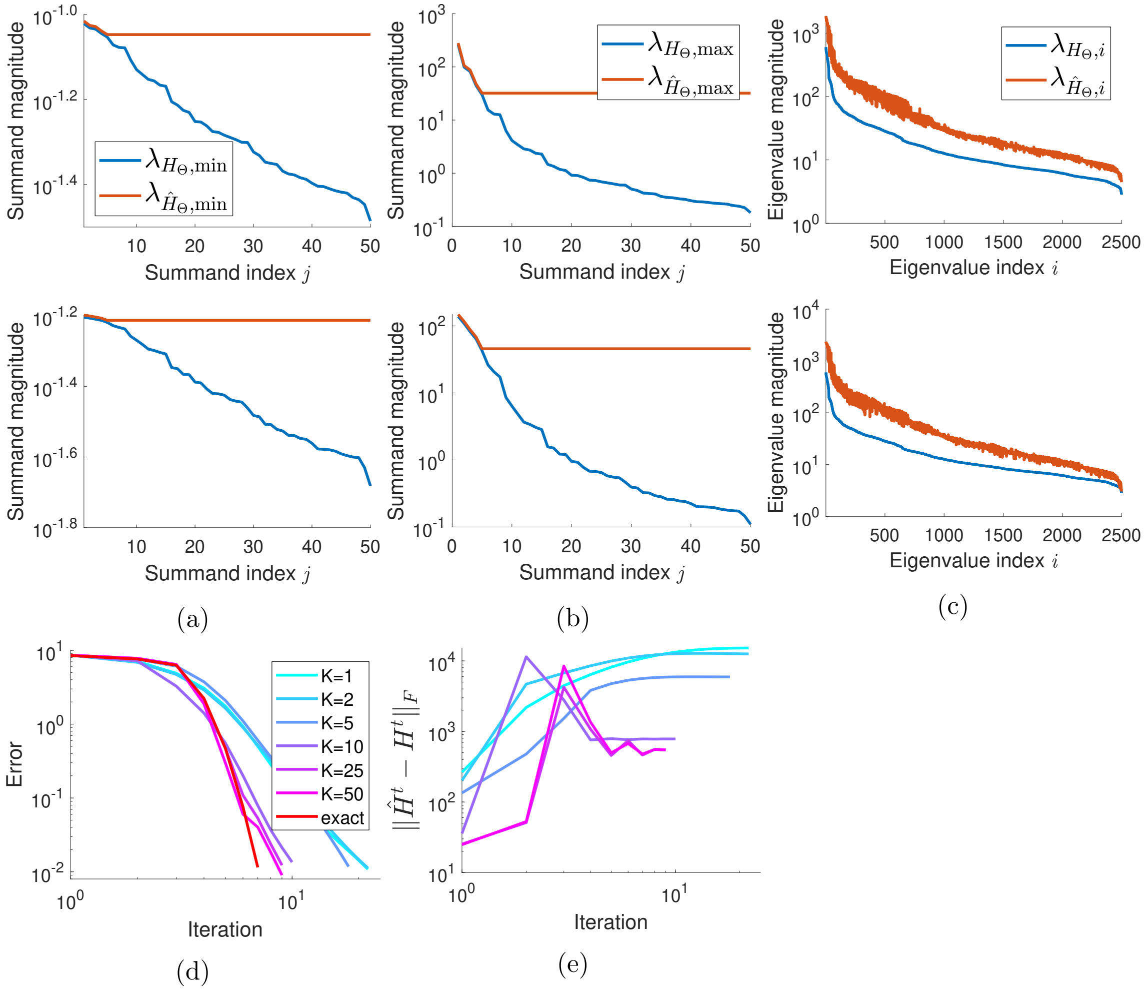
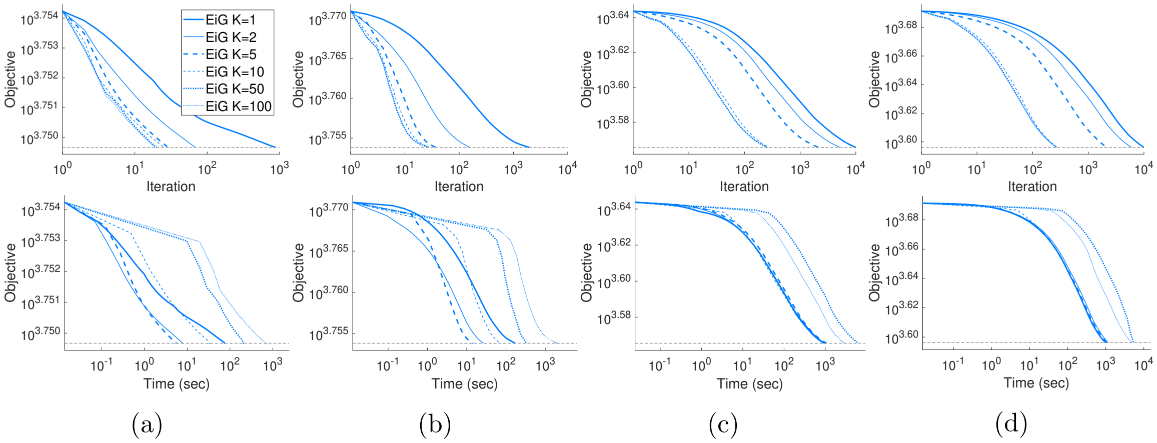


Figure 5: Effects of Hessian approximation on the convergence of EiGLasso for data simulated from a graph with clusters. (a) The summands in $\lambda_{\boldsymbol{H}_{\boldsymbol{\Theta}},\min} = \sum_{j=1}^{q}(\lambda_{\boldsymbol{\Theta},p} + \lambda_{\boldsymbol{\Psi},j})^{-2}$ and $\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}},\min} = \sum_{j=1}^{K}(\lambda_{\boldsymbol{\Theta},p} + \lambda_{\boldsymbol{\Psi},j})^{-2} + (q-K)(\lambda_{\boldsymbol{\Theta},p} + \lambda_{\boldsymbol{\Psi},K})^{-2}$ for EiGLasso with $K = 5$. (b) The summands in $\lambda_{\boldsymbol{H}_{\boldsymbol{\Theta}},\max}$ and $\lambda_{\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}},\max}$, as in Panel (a). (c) The eigenvalues of $\boldsymbol{H}_{\boldsymbol{\Theta}}$ and $\hat{\boldsymbol{H}}_{\boldsymbol{\Theta}}$, sorted with the eigenvalues of $\boldsymbol{H}_{\boldsymbol{\Theta}}$. In Panels (a)-(c), the results are shown at the 5th iteration (top) and at the last iteration (bottom). (d) Error measured as $\left\|\begin{bmatrix}\mathrm{vec}(\boldsymbol{\Theta}^t - \boldsymbol{\Theta}^*)\\ \mathrm{vec}(\boldsymbol{\Psi}^t - \boldsymbol{\Psi}^*)\end{bmatrix}\right\|_2$ over iterations. (e) $\|\hat{\boldsymbol{H}} - \boldsymbol{H}\|_F$ over iterations. Graph size $p, q = 50$ was used.

et al., 2018). The brain tissue samples were collected from generations 50-56 of the LG/J × SM/J advanced intercross line of mice. RNA was extracted from three brain tissue types: prefrontal cortex (PFC; 185 mice), striatum (STR; 169 mice), and hippocampus (HIP; 208 mice). Several genes from these tissues have been found relevant to psychiatric and metabolic disorders (Gonzales et al., 2018).

We applied EiGLasso and TeraLasso to this data to estimate $\boldsymbol{\Theta}$ for the relationship among the mice and $\boldsymbol{\Psi}$ for the gene network. We selected $10,000$ genes from each tissue



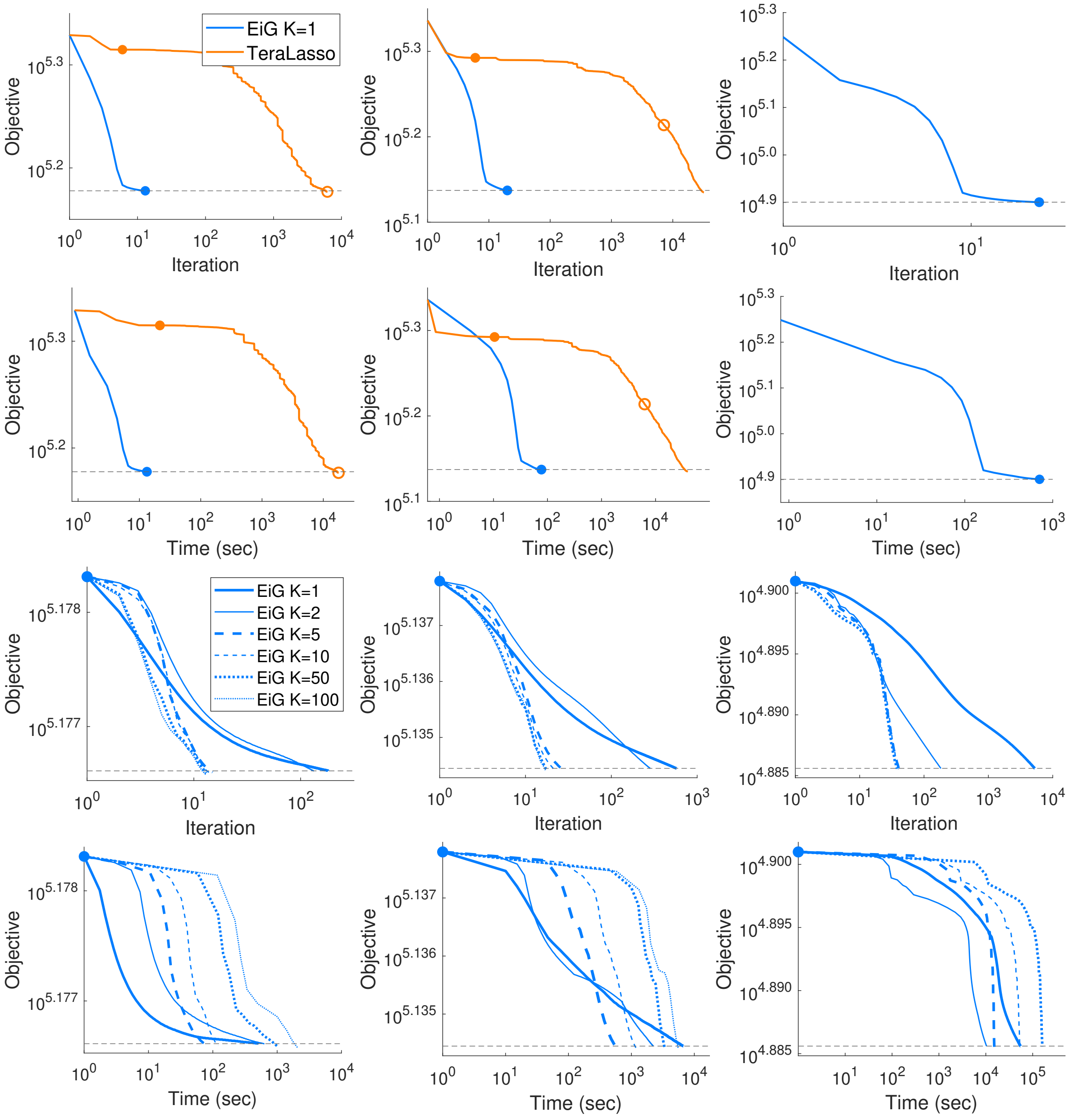
(a) (b) (c) (d)

Figure 6: Convergence of EiGLasso near optimum on simulated data. In (a) and (b), objectives over iterations (top) and over time (bottom) are shown for the same two datasets from random graphs used in Figures 2(a) and 2(b). In (c) and (d), results are shown for the same two datasets from graphs with clusters used in Figures 3(a) and 3(b). EiGLasso with different K's was initialized with the estimates from EiGLasso with $K = 1$ at $\epsilon = 10^{-3}$.

Tissue	Genes (q)	EiGLasso (sec)	TeraLasso (sec)
PFC ($p = 185$)	436	2	887
	877	13	18194
	5108	6141	
	10000	25360	
STR ($p = 169$)	447	9	6917
	977	77	39645
	5007	32977	
	10000	65797	
HIP ($p = 208$)	508	90	54454
	879	708	
	5529	3106	
	10000	20104	

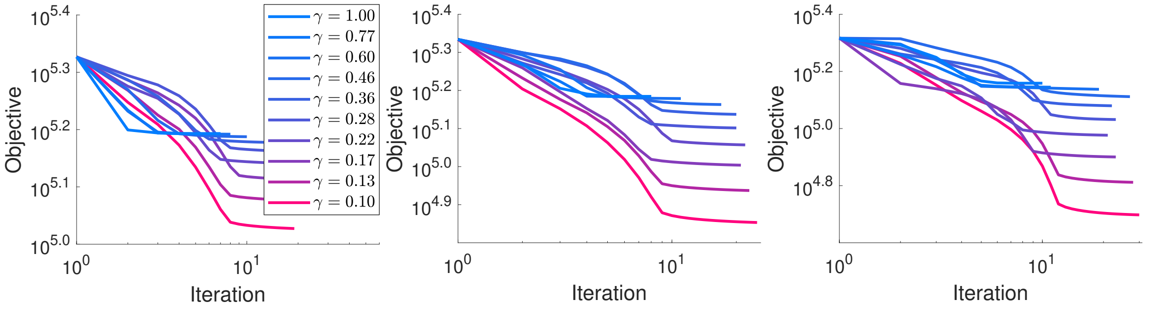
Table 2: Comparison of the computation time of EiGLasso with $K = 1$ and TeraLasso on mouse gene-expression data.

type by excluding genes with low expression variance across the mice. As TeraLasso did not converge within 24 hours on this dataset with $q = 10,000$, to compare the computation time of EiGLasso and TeraLasso, we formed three smaller datasets for each tissue type by performing hierarchical clustering and selecting clusters of genes. We selected the regularization parameters from the range $[0.1, 1.0]$ by using Bayesian Information Criterion (BIC). EiGLasso was run with $K = 1$ and $\epsilon = 10^{-3}$. TeraLasso was run up to 24 hours, until



(a) (b) (c)

Figure 7: Comparison of the convergence of EiGLasso and TeraLasso on mouse gene-expression data. (a) PFC with $q = 877$ genes. (b) STR with $q = 977$ genes. (c) HIP with $q = 879$ genes. In top two rows, the convergence over iterations and over time for EiGLasso with $K = 1$ at $\epsilon = 10^{-3}$ and for TeraLasso converging to the same objective value is shown. The '•' and '◯' mark the points that satisfy $\epsilon = 10^{-3}$ and 10^{-7}, respectively. In bottom two rows, the convergence over iterations and over time for EiGLasso with different K's is shown, after initializing with the estimates from EiGLasso with $K = 1$ at $\epsilon = 10^{-3}$ and running them until they reach the objective value that EiGLasso with $K = 1$ reached at $\epsilon = 10^{-6}$.



(a)

(b)

(c)

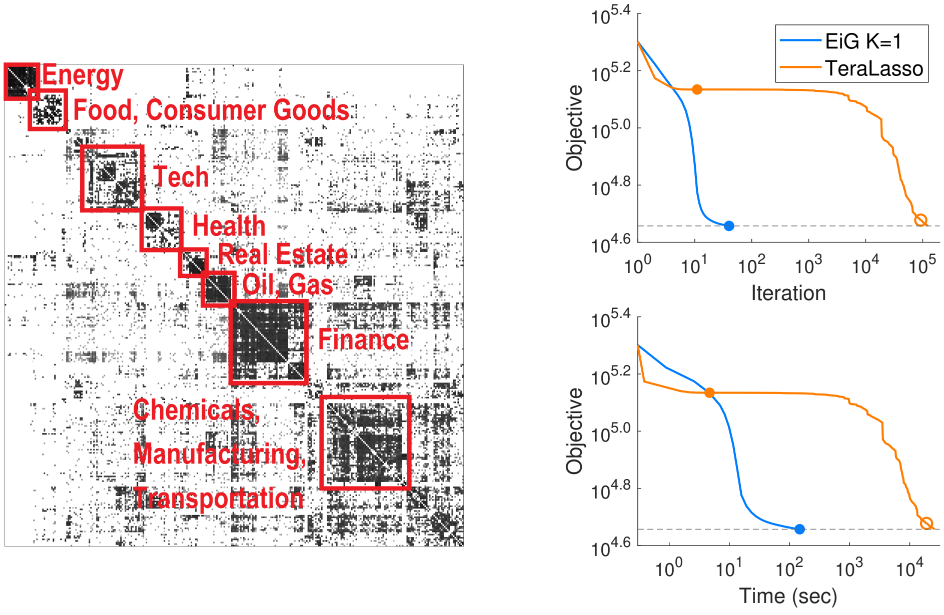
Figure 8: Convergence of EiGLasso on mouse gene-expression data for different regularization parameters. For ten different values of the regularization parameter $\gamma = \gamma_{\mathbf{\Theta}} = \gamma_{\mathbf{\Psi}}$, the objective values over iterations are shown for (a) PFC with $q = 877$ genes, (b) STR with $q = 977$ genes, and (c) HIP with $q = 879$ genes.

it reached similar objective values to those from EiGLasso, which corresponded to $\epsilon = 10^{-7}$ or 10^{-8}.

Across all datasets with different tissue types and with different sizes, EiGLasso was significantly faster than TeraLasso (Table 2). On smaller datasets with less than 1,000 genes, EiGLasso was two to three orders-of-magnitude faster than TeraLasso. On HIP tissue with 879 genes, within 24 hours, TeraLasso was not able to converge to the similar objective value that EiGLasso reached, while EiGLasso converged in about 12 minutes. On the larger datasets with more than 5,000 genes, TeraLasso was not able to reach the same objective as EiGLasso with $K = 1$ within 24 hours. On the full dataset with 10,000 genes, EiGLasso took six and seven hours for the HIP and PFC tissue types and 18 hours for the STR tissue type.

On the second smallest dataset from each tissue type in Table 2, we examined the convergence of EiGLasso over iterations and over time (Figure 7). In all tissue types, EiGLasso with $K = 1$ converged at least two orders-of-magnitude faster and required far fewer iterations than TeraLasso (Figure 7, top two rows). TeraLasso needed more stringent convergence criterion $\epsilon = 10^{-7}$ or 10^{-8} to reach the similar objective value that EiGLasso reached with $\epsilon = 10^{-3}$. For EiGLasso, we considered switching from $K = 1$ to larger K's near optimum for faster convergence. Specifically, we initialized EiGLasso with different K's with the estimates from EiGLasso with $K = 1$ at $\epsilon = 10^{-3}$ and ran it until it reached the same objective value obtained by EiGLasso with $K = 1$ at $\epsilon = 10^{-6}$ (Figure 7, bottom two rows). With a larger K, the number of iterations decreased due to the faster convergence rate, though this did not translate to reduced computation time due to the higher cost of computing the approximate Hessian. In fact, EiGLasso with relatively small K's required the least computation time: $K = 5$ for PFC and STR, and $K = 2$ for HIP. Overall, we found that EiGLasso with $K = 1$ or other small K's was the most efficient, which is consistent with what we observed in our simulation study. Finally, the number of iterations required by EiGLasso was not affected significantly by different regularization parameters γ (Figure 8).

On the second largest dataset from each tissue type in Table 2, we examined the networks over genes estimated by EiGLasso. The estimated gene network for each tissue type was clustered by using the Markov clustering algorithm (Morris et al., 2011) available in



(a) (b)

Figure 9: Results on S&P 500 data. (a) Graph over S&P 500 companies estimated by EiGLasso from the stock closing price data. Data from 306 companies over 2,517 days were used. The off-diagonals of the estimated graph are shown. (b) Comparison of the convergence of EiGLasso and TeraLasso on the data from 306 companies over 500 days. The objective values over iterations (top) and time (bottom) are shown. The '•' and '○' mark the points that satisfy the convergence criteria $\epsilon = 10^{-3}$ and 10^{-7}, respectively. EiGLasso with $K = 1$ was used.

Cytoscape (Shannon et al., 2003). We then ranked the clusters by their mean edge weights and analyzed the top cluster from each tissue type with Gene Ontology (Ashburner et al., 2000) and PANTHER (Thomas et al., 2003). In PFC, 6 out of 35 genes in the cluster were enriched for axon ensheathment (p-value$= 5.26 \times 10^{-8}$), and 24 other genes were known to affect axon ensheathment via genetic pathways related to myelination and oligodendrocyte (Dugas et al., 2006; Mattan et al., 2010; Edgar et al., 2013; McKenzie et al., 2017; Liu et al., 2018; Zhang et al., 2018a; Siems et al., 2020; Szklarczyk et al., 2021; Zhang et al., 2021). In STR, 20 out of 56 genes were enriched for cilium (p-value$= 4.15 \times 10^{-16}$) and, out of the 20 genes, 13 genes for cilium organization (p-value$= 2.68 \times 10^{-12}$) and 9 genes for cilium movement (p-value$= 8.74 \times 10^{-10}$). For HIP, the gene network was clustered after removing edges with the magnitudes of weights below 0.01. In the top cluster for HIP, out of 83 genes, 43 genes were enriched for positive regulation of cellular process (p-value$= 1.17 \times 10^{-6}$), four genes for canonical Wnt signaling pathway (p-value$= 2.85 \times 10^{-4}$), and three genes for positive regulation of JUN kinase activity (p-value$= 1.36 \times 10^{-3}$), which is known to be related to memory (Blum et al., 1999; Bevilaqua et al., 2003; Fortress et al., 2013).

5.3 Financial data

We applied EiGLasso and TeraLasso to the historical daily stock price data of the S&P 500 constituents to model the relationship among companies and dependencies across time points. We obtained the daily stock closing prices for 306 companies that remained in the index from 2/16/2010 to 2/13/2020 for 2,523 days. We excluded data for six days when stocks for only a small subset of the constituents were traded. We normalized the data by computing the proportion of price change on day t from the previous day as $\frac{\text{price}^t - \text{price}^{t-1}}{\text{price}^{t-1}}$. We ran EiGLasso with $K = 1$ and $\epsilon = 10^{-3}$ and ran TeraLasso until it reached the similar objective as EiGLasso. We used BIC to select the optimal regularization parameter from 10 different values in the range of $[0.01, 1.0]$.

EiGLasso with $K = 1$ took 46 hours, whereas TeraLasso did not converge within 48 hours. Clusters obtained by applying hierarchical clustering to the graph over 306 companies matched the known sectors (Figure 9(a)). We compared the computation time of EiGLasso with $K = 1$ and TeraLasso on a small subset over 500 days for the same 306 companies. On this smaller dataset, EiGLasso took 148 seconds to converge and TeraLasso took 25,739 seconds, showing that EiGLasso was again two orders-of-magnitude faster than TeraLasso (Figure 9(b)).

6. Conclusion

We introduced EiGLasso, an efficient algorithm for estimating an inverse covariance matrix as the Kronecker sum of two matrices, each representing a graph over samples and a graph over features. Extending the second-order optimization method used in QUIC for Gaussian graphical models (Hsieh et al., 2014), EiGLasso employed the eigendecomposition of the two matrices for individual graphs during optimization to avoid an expensive computation of large gradient and Hessian matrices with an inflated structure. Based on this eigendecomposition, EiGLasso approximated the Hessian to further reduce the computation time. We showed the quadratic convergence with the exact Hessian and the linear convergence with the approximate Hessian. In our experiments, EiGLasso achieved two to three orders-of-magnitude speed-up over the previous state-of-the-art method, TeraLasso (Greenewald et al., 2019).

In addition, we introduced a new approach for identifying and estimating the unidentifiable diagonal elements of the matrices for the feature and sample graphs. We showed that given the ratio of the traces of the two matrices, the diagonal parameters can be identified uniquely. EiGLasso performed the optimization without identifying these parameters, since all quantities involved in the optimization are invariant to the trace ratio. We further showed that it is sufficient to identify the parameters with the fixed trace ratio once after the optimization is complete. Thus, EiGLasso used a significantly simpler strategy for estimating the unidentifiable parameters than the previous methods.

There are several possible future directions. One natural extension of EiGLasso is to adopt Big&QUIC (Hsieh et al., 2013), an extension of QUIC to remove the memory requirement. Unlike QUIC, EiGLasso uses the eigendecomposition of the parameter matrices, which in itself has the space complexity cubic in the graph sizes. Thus, further research is needed to incorporate the eigendecomposition within the block-wise update used in

Big&QUIC. Another future direction is to develop a Hessian approximation technique with faster theoretical convergence. While we showed that EiGLasso with our approximate Hessian converges linearly to the optimum, it may be possible to achieve superlinear convergence with an approximate Hessian that approaches the exact Hessian over iterations.

Acknowledgments

This work was supported by NIH 1R21HG011116 and 1R21HG010948. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by NSF ACI-1548562 and ACI-1445606.

References

M. Ashburner, C. Ball, J. Blake, D. Botstein, H. L. Butler, J. Cherry, A. P. Davis, K. Dolinski, S. Dwight, J. Eppig, M. Harris, D. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. Matese, J. Richardson, M. Ringwald, G. Rubin, and G. Sherlock. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

D. P. Bertsekas. *Nonlinear programming.* Athena Scientific, 1995.

L. R. M. Bevilaqua, D. S. Kerr, J. H. Medina, I. Izquierdo, and M. Cammarota. Inhibition of hippocampal Jun N-terminal kinase enhances short-term memory but blocks long-term memory formation and retrieval of an inhibitory avoidance task. *European Journal of Neuroscience*, 17(4):897–902, 2003.

S. Blum, A. N. Moore, F. Adams, and P. K. Dash. A mitogen-activated protein kinase cascade in the CA1/CA2 subfield of the dorsal hippocampus is essential for long-term spatial memory. *Journal of Neuroscience*, 19(9):3535–3544, 1999.

S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

X. Dai, T. Li, Z. Bai, Y. Yang, X. Liu, J. Zhan, and B. Shi. Breast cancer intrinsic subtype classification, clinical use and future trends. *American Journal of Cancer Research*, 5: 2929–43, 2015.

J. C. Dugas, Y. C. Tai, T. P. Speed, J. Ngai, and B. A. Barres. Functional genomic analysis of oligodendrocyte differentiation. *Journal of Neuroscience*, 26(43):10967–10983, 2006.

J. Dunn. Newton's method and the Goldstein step length rule for constrained minimization. In *Proceedings of the 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, pages 17–22. IEEE, 1980.

R. Edgar, Y. Mazor, A. Rinon, J. Blumenthal, Y. Guan-Golan, E. Buzhor, I. Livnat, S. Ben-Ari, I. Lieder, A. Shitrit, Y. Gilboa, A. Ben-Yehudah, O. Edri, N. Blondheim Shraga, Y. Bogoch, L. Leshansky, S. Aharoni, M. West, D. Warshawsky, and R. Shtrichman. Lifemap Discoveryn: The embryonic development, stem cells, and regenerative medicine research portal. *PLoS ONE*, 8:e66629, 2013.

A. M. Fortress, S. L. Schram, J. J. Tuscher, and K. M. Frick. Canonical Wnt signaling is necessary for object recognition memory consolidation. *Journal of Neuroscience*, 33(31): 12619–12626, 2013.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2007.

H. Gao, Z. Wang, and S. Ji. Kronecker attention networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 229–237, 2020.

N. M. Gonzales, J. Seo, A. I. Hernandez Cordero, C. L. St. Pierre, J. S. Gregory, M. G. Distler, M. Abney, S. Canzar, A. Lionikas, and A. A. Palmer. Genome-wide association analysis in a mouse advanced intercross line. *Nature Communications*, 9(1):5162, 2018.

K. Greenewald, S. Zhou, and A. Hero III. Tensor graphical lasso (TeraLasso). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(5):901–931, 2019.

S. Haupt, A. Zeilmann, A. Ahadova, H. Bläker, M. von Knebel Doeberitz, M. Kloor, and V. Heuveline. Mathematical modeling of multiple pathways in colorectal carcinogenesis using dynamical systems with Kronecker structure. *PLOS Computational Biology*, 17(5): 1–31, 05 2021. doi: 10.1371/journal.pcbi.1008970.

C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. Ravikumar, and R. A. Poldrack. BIG & QUIC: Sparse inverse covariance estimation for a million variables. In *Advances in Neural Information Processing Systems*, volume 26, pages 3165–3173, 2013.

C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. QUIC: Quadratic approximation for sparse inverse covariance estimation. *Journal of Machine Learning Research*, 15(83): 2911–2947, 2014.

A. Kalaitzis, J. Lafferty, N. D. Lawrence, and S. Zhou. The bigraphical lasso. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1229–1237. PMLR, 2013.

N. Kalchbrenner, A. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1771–1779. PMLR, 2017.

B. F. King. Market and industry factors in stock price behavior. *The Journal of Business*, 39(1):139–190, 1966.

C. Leng and C. Y. Tang. Sparse matrix graphical models. *Journal of the American Statistical Association*, 107(499):1187–1200, 2012.

G. Li, J. Luo, Q. Xiao, C. Liang, and P. Ding. Prediction of microRNA–disease associations with a Kronecker kernel matrix dimension reduction model. *Royal Society of Chemistry*, 8:4377–4385, 2018.

M. Liu, P. Xu, Z. Guan, X. Qian, P. Dockery, U. Fitzgerald, T. O'Brien, and S. Shen. Ulk4 deficiency leads to hypomyelination in mice. *Glia*, 66:175–190, 2018.

N. S. Mattan, C. A. Ghiani, M. Lloyd, R. Matalon, D. Bok, P. Casaccia, and J. de Vellis. Aspartoacylase deficiency affects early postnatal development of oligodendrocytes and myelination. *Neurobiology of disease*, 40(2):432–443, 2010.

R. Mazumder and T. Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *Journal of Machine Learning Research*, 13(27):781–794, 2012.

A. T. McKenzie, S. Moyon, M. Wang, I. Katsyv, W.-M. Song, X. Zhou, E. B. Dammer, D. M. Duong, J. Aaker, Y. Zhao, et al. Multiscale network modeling of oligodendrocytes reveals molecular components of myelin dysregulation in Alzheimer's disease. *Molecular Neurodegeneration*, 12(1):1–20, 2017.

J. H. Morris, L. Apeltsin, A. M. Newman, J. Baumbach, T. Wittkop, G. Su, G. D. Bader, and T. E. Ferrin. ClusterMaker: a multi-algorithm clustering plugin for Cytoscape. *BMC Bioinformatics*, 12(1):1–14, 2011.

S. Park, K. Shedden, and S. Zhou. Non-separable covariance models for spatio-temporal data, with applications to neural encoding analysis. *arXiv preprint arXiv:1705.05265*, 2017.

M. Rudelson and S. Zhou. Errors-in-variables models with dependent measurements. *Electronic Journal of Statistics*, 11(1):1699–1797, 2017.

P. Shannon, A. Markiel, O. Ozier, N. Baliga, J. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13:2498–504, 2003.

S. B. Siems, O. Jahn, M. A. Eichel, N. Kannaiyan, L. M. N. Wu, D. L. Sherman, K. Kusch, D. Hesse, R. B. Jung, R. Fledrich, M. W. Sereda, M. J. Rossner, P. J. Brophy, and H. B. Werner. Proteome profile of peripheral myelin in healthy mice and in a neuropathy model. *eLife*, 9:e51406, 2020.

D. Szklarczyk, A. L. Gable, K. C. Nastou, D. Lyon, R. Kirsch, S. Pyysalo, N. T. Doncheva, M. Legeay, T. Fang, P. Bork, L. Jensen, and C. V. Mering. The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Research*, 49:D605–D612, 2021.

P. D. Thomas, M. Campbell, A. Kejariwal, H. Mi, B. Karlak, R. Daverman, K. Diemer, A. Muruganujan, and A. Narechania. PANTHER: a library of protein families and subfamilies indexed by function. *Genome research*, 13(9):2129–41, 2003.

P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1):387–423, 2009.

T. Tsiligkaridis and A. O. Hero. Covariance estimation in high dimensions via Kronecker product expansions. *IEEE Transactions on Signal Processing*, 61(21):5347–5360, 2013.

D. M. Witten, J. H. Friedman, and N. Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4):892–900, 2011.

J. Yin and H. Li. Model selection and estimation in the matrix normal graphical model. *Journal of Multivariate Analysis*, 107:119–140, 2012.

J. H. Yoon and S. Kim. EiGLasso: Scalable estimation of Cartesian product of sparse inverse covariance matrices. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, pages 1248–1257. PMLR, 2020.

M. Zhang, J. Wang, K. Zhang, G. Lu, Y. Liu, K. Ren, W. Wang, D. Xin, L. Xu, H. Mao, et al. Ten-eleven translocation 1 mediated-dna hydroxymethylation is required for myelination and remyelination in the mouse brain. *Nature communications*, 12(1):1–21, 2021.

S. Zhang, X. Zhu, X. Gui, C. Croteau, L. Song, J. Xu, A. Wang, P. Bannerman, and F. Guo. Sox2 is essential for oligodendroglial proliferation and differentiation during postnatal brain myelination and CNS remyelination. *The Journal of Neuroscience*, 38:1802 – 1820, 2018a.

T. Zhang, W. Zheng, Z. Cui, and Y. Li. Tensor graph convolutional neural network. *arXiv preprint arXiv:1803.10071*, 2018b.

X. Zhang. *Statistical analysis for network data using matrix variate models and latent space models.* Ph.D. disseration, University of Michigan, 2020.

S. Zhou. Gemini: Graph estimation with matrix variate normal instances. *Annals of Statistics*, 42(2):532–562, 2014.

Appendix A. Hessian Computation

We show that the collapse of $\boldsymbol{W} \otimes \boldsymbol{W}$ in Eq. (17) to obtain the Hessian can be viewed as the same type of deflation for the gradient in Figure 1(b) applied twice to $\boldsymbol{W} \otimes \boldsymbol{W}$.

Let $\boldsymbol{A}$ denote a matrix of size $m_1 \times m_2$. Then, the elements of the Kronecker product $\boldsymbol{A} \otimes \boldsymbol{A}$ are given as $[\boldsymbol{A} \otimes \boldsymbol{A}]_{(i-1)m_1+k,\,(j-1)m_2+l} = [\boldsymbol{A}]_{ij}[\boldsymbol{A}]_{kl}$ for $i, k = 1, \ldots, m_1$, and $j, l = 1, \ldots, m_2$. As a representation of $\boldsymbol{A} \otimes \boldsymbol{A}$ with different row/column indices, we consider $\mathrm{vec}(\boldsymbol{A})\,\mathrm{vec}(\boldsymbol{A})^T$ whose elements are given as $[\mathrm{vec}(\boldsymbol{A})\,\mathrm{vec}(\boldsymbol{A})^T]_{(i-1)m_1+j,\,(k-1)m_1+l} = [\boldsymbol{A}]_{ij}[\boldsymbol{A}]_{kl}$. We set up an operator

$$\mathcal{P}_{m_1,m_2}(\boldsymbol{A} \otimes \boldsymbol{A}) = \mathrm{vec}(\boldsymbol{A})\,\mathrm{vec}(\boldsymbol{A})^T,$$

which maps the elements of $\boldsymbol{A} \otimes \boldsymbol{A}$ to those of $\mathrm{vec}(\boldsymbol{A})\,\mathrm{vec}(\boldsymbol{A})^T$. More generally, we define an operator on $\sum_{i=1}^N \boldsymbol{A}_i \otimes \boldsymbol{A}_i$, where $\boldsymbol{A}_i \in \mathbb{R}^{m_1 \times m_2}$,

$$\mathcal{P}_{m_1,m_2}(\sum_{i=1}^N \boldsymbol{A}_i \otimes \boldsymbol{A}_i) = \sum_{i=1}^N \mathrm{vec}(\boldsymbol{A}_i)\,\mathrm{vec}(\boldsymbol{A}_i)^T. \tag{32}$$

Collapsing $\boldsymbol{W} \otimes \boldsymbol{W}$ into $\boldsymbol{H}_{\boldsymbol{\Theta}}$ amounts to applying the same type of collapse as in the gradient (Figure 1(b)) twice on $\mathcal{P}_{p^2q^2,p^2q^2}(\boldsymbol{W} \otimes \boldsymbol{W})$ to obtain $\mathcal{P}_{p,p}(\boldsymbol{H}_{\boldsymbol{\Theta}})$, $\mathcal{P}_{q,q}(\boldsymbol{H}_{\boldsymbol{\Psi}})$, and $\mathcal{P}_{p,q}(\boldsymbol{H}_{\boldsymbol{\Theta\Psi}})$. To see this, we re-write $\boldsymbol{H}_{\boldsymbol{\Theta}}$ as

$$\boldsymbol{H}_{\boldsymbol{\Theta}} = \boldsymbol{P}_{\boldsymbol{\Theta}}^T(\boldsymbol{W} \otimes \boldsymbol{W})\boldsymbol{P}_{\boldsymbol{\Theta}}$$

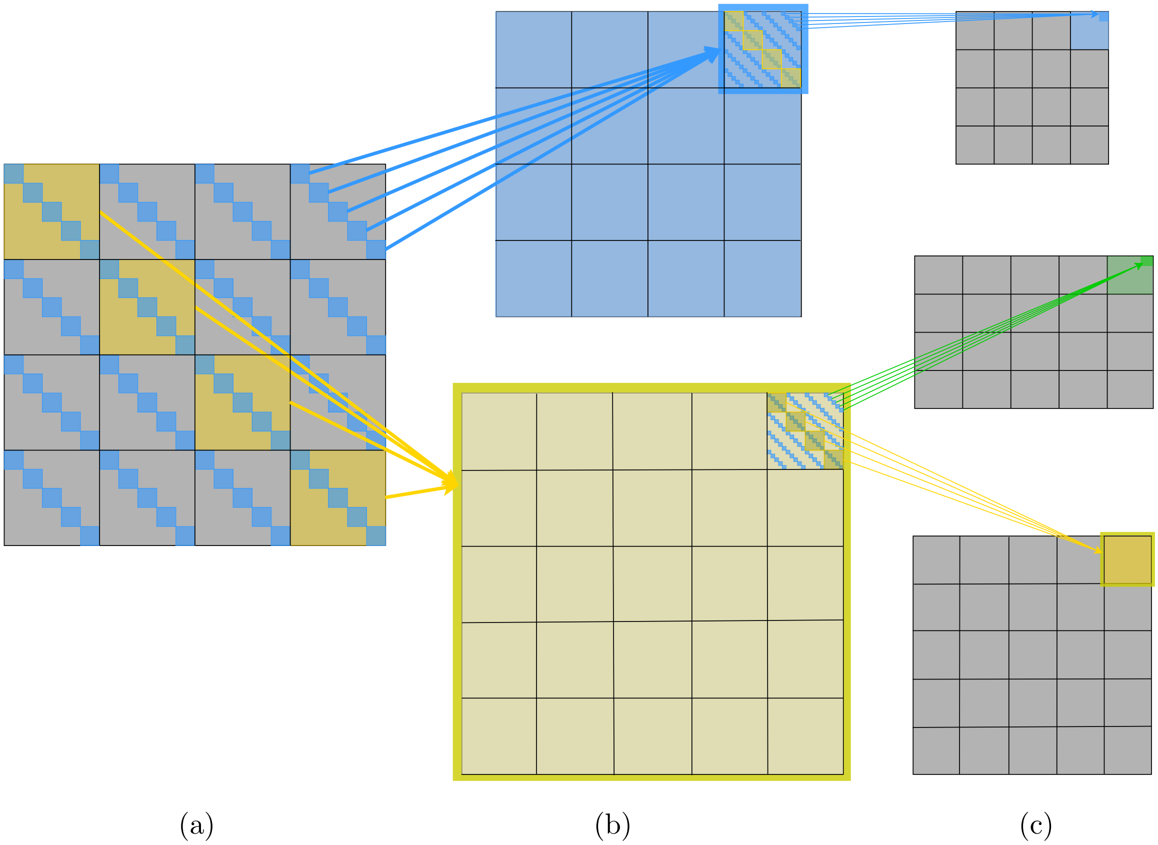


Figure 10: Illustration of the Hessian computation in EiGLasso. A two-stage collapse of $\mathcal{P}_{pq,pq}(\boldsymbol{W} \otimes \boldsymbol{W})$ into $\mathcal{P}_{p,p}(\boldsymbol{H}_{\boldsymbol{\Theta}})$, $\mathcal{P}_{p,q}(\boldsymbol{H}_{\boldsymbol{\Theta\Psi}})$, and $\mathcal{P}_{q,q}(\boldsymbol{H}_{\boldsymbol{\Psi}})$ is shown for the same $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}$ with $p = 4$ and $q = 5$ in Figure 1. (a) $\mathcal{P}_{pq,pq}(\boldsymbol{W} \otimes \boldsymbol{W})$ of size $p^2q^2 \times p^2q^2$ with a $pq \times pq$ matrix in each cell. Only the cells with non-zero elements in the Kronecker-sum-like mask $(\mathbb{1}_p \oplus \mathbb{1}_q) \otimes \mathbb{1}_{pq} = (\mathbb{1}_p \otimes \boldsymbol{I}_q + \boldsymbol{I}_p \otimes \mathbb{1}_q) \otimes \mathbb{1}_{pq}$ contribute to the Hessian. The two components of this mask, $(\mathbb{1}_p \otimes \boldsymbol{I}_q) \otimes \mathbb{1}_{pq}$ and $(\boldsymbol{I}_p \otimes \mathbb{1}_q) \otimes \mathbb{1}_{pq}$, are shown with blue and yellow, respectively. (b) Two intermediate matrices after the first collapse, each with size $p^2q \times p^2q$ (top) and size $pq^2 \times pq^2$ (bottom), with a $pq \times pq$ matrix in each block. Only the elements of these matrices that correspond to non-zero elements in the Kronecker-sum-like mask $\mathbb{1}_p \otimes (\mathbb{1}_p \oplus \mathbb{1}_q)$ (top) and $\mathbb{1}_q \otimes (\mathbb{1}_p \oplus \mathbb{1}_q)$ (bottom) contribute to the Hessian. This is equivalent to applying the mask $\mathbb{1}_p \oplus \mathbb{1}_q$ to each block. (c) $\mathcal{P}_{p,p}(\boldsymbol{H}_{\boldsymbol{\Theta}})$ of size $p^2 \times p^2$ with a $p \times p$ matrix in each block (top), $\mathcal{P}_{p,q}(\boldsymbol{H}_{\boldsymbol{\Theta\Psi}})$ of size $p^2 \times q^2$ with a $p \times q$ matrix in each block (middle), and $\mathcal{P}_{q,q}(\boldsymbol{H}_{\boldsymbol{\Psi}})$ of size $q^2 \times q^2$, with a $q \times q$ matrix in each block (bottom). The two collapses from Panel (a) to Panel (b) and from Panel (b) to Panel (c) are shown with arrows that indicate the sum of the cells in the larger matrix into the cell in the smaller matrix.

$$= \sum_{i=1}^{q}\sum_{j=1}^{q}((\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i}) \otimes (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i}))^T(\boldsymbol{W} \otimes \boldsymbol{W})((\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j}) \otimes (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j}))$$

$$= \sum_{i=1}^{q}\sum_{j=1}^{q} \Big[(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})^T \boldsymbol{W}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j})\Big] \otimes \Big[(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})^T \boldsymbol{W}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j})\Big].$$

Then, we apply the operator in Eq. (32) to $\boldsymbol{H}_{\boldsymbol{\Theta}}$:

$$\begin{aligned}
\mathcal{P}_{p,p}(\boldsymbol{H}_{\boldsymbol{\Theta}}) &= \sum_{i=1}^{q}\sum_{j=1}^{q} \mathrm{vec}\left((\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})^T \boldsymbol{W}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j})\right) \mathrm{vec}\left((\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})^T \boldsymbol{W}(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j})\right)^T \\
&= \sum_{i=1}^{q}\sum_{j=1}^{q} (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j} \otimes \boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})^T \,\mathrm{vec}(\boldsymbol{W})\,\mathrm{vec}(\boldsymbol{W})^T (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j} \otimes \boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i}) \\
&= \sum_{i=1}^{q}\sum_{j=1}^{q} \Bigg(\Big[(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j} \otimes \boldsymbol{I}_p \otimes \boldsymbol{I}_q)(\boldsymbol{I}_p \otimes 1 \otimes \boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})\Big]^T \\
&\qquad\qquad \mathcal{P}_{pq,pq}(\boldsymbol{W} \otimes \boldsymbol{W}) \Big[(\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j} \otimes \boldsymbol{I}_p \otimes \boldsymbol{I}_q)(\boldsymbol{I}_p \otimes 1 \otimes \boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})\Big] \Bigg) \\
&= \sum_{i=1}^{q} (\boldsymbol{I}_p \otimes (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i}))^T \Bigg[\sum_{j=1}^{q} ((\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j}) \otimes \boldsymbol{I}_{pq})^T \\
&\qquad\qquad \mathcal{P}_{pq,pq}(\boldsymbol{W} \otimes \boldsymbol{W})((\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j}) \otimes \boldsymbol{I}_{pq}) \Bigg] (\boldsymbol{I}_p \otimes (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})). \\
&= \sum_{i=1}^{q} (\boldsymbol{I}_p \otimes (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i}))^T \boldsymbol{M}_{\boldsymbol{\Theta}} (\boldsymbol{I}_p \otimes (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})), \qquad (33)
\end{aligned}$$

where $\boldsymbol{M}_{\boldsymbol{\Theta}} = \sum_{j=1}^{q} ((\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j}) \otimes \boldsymbol{I}_{pq})^T \mathcal{P}_{pq,pq}(\boldsymbol{W} \otimes \boldsymbol{W})((\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,j}) \otimes \boldsymbol{I}_{pq})$. Similarly,

$$\begin{aligned}
\mathcal{P}_{q,q}(\boldsymbol{H}_{\boldsymbol{\Psi}}) &= \sum_{i=1}^{p} (\boldsymbol{I}_q \otimes (\boldsymbol{e}_{p,i} \otimes \boldsymbol{I}_q))^T \boldsymbol{M}_{\boldsymbol{\Psi}} (\boldsymbol{I}_q \otimes (\boldsymbol{e}_{p,i} \otimes \boldsymbol{I}_q)), \qquad (34) \\
\mathcal{P}_{p,q}(\boldsymbol{H}_{\boldsymbol{\Theta}\boldsymbol{\Psi}}) &= \sum_{i=1}^{q} (\boldsymbol{I}_p \otimes (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i}))^T \boldsymbol{M}_{\boldsymbol{\Psi}} (\boldsymbol{I}_q \otimes (\boldsymbol{I}_p \otimes \boldsymbol{e}_{q,i})), \qquad (35)
\end{aligned}$$

where $\boldsymbol{M}_{\boldsymbol{\Psi}} = \sum_{j=1}^{p} ((\boldsymbol{e}_{p,j} \otimes \boldsymbol{I}_q) \otimes \boldsymbol{I}_{pq})^T \mathcal{P}_{pq,pq}(\boldsymbol{W} \otimes \boldsymbol{W})((\boldsymbol{e}_{p,j} \otimes \boldsymbol{I}_q) \otimes \boldsymbol{I}_{pq})$. Eqs. (33)-(35) show the two-stage collapse for Hessian computation: $\mathcal{P}_{pq,pq}(\boldsymbol{W} \otimes \boldsymbol{W})$ (Figure 10(a)) is collapsed to $\boldsymbol{M}_{\boldsymbol{\Theta}}$ and $\boldsymbol{M}_{\boldsymbol{\Psi}}$ (Figure 10(b)), which are then collapsed to $\mathcal{P}_{p,p}(\boldsymbol{H}_{\boldsymbol{\Theta}})$, $\mathcal{P}_{p,q}(\boldsymbol{H}_{\boldsymbol{\Theta}\boldsymbol{\Psi}})$, and $\mathcal{P}_{q,q}(\boldsymbol{H}_{\boldsymbol{\Psi}})$ (Figure 10(c)). The first collapse of $\mathcal{P}_{pq,pq}(\boldsymbol{W} \otimes \boldsymbol{W})$ to $\boldsymbol{M}_{\boldsymbol{\Theta}}$ and $\boldsymbol{M}_{\boldsymbol{\Psi}}$ is equivalent to applying the same collapse for the gradient in Figure 1(b) at the level of cells in Figure 10(a) each with size $pq \times pq$. The second collapse from $\boldsymbol{M}_{\boldsymbol{\Theta}}$ and $\boldsymbol{M}_{\boldsymbol{\Psi}}$ to $\mathcal{P}_{p,p}(\boldsymbol{H}_{\boldsymbol{\Theta}})$, $\mathcal{P}_{p,q}(\boldsymbol{H}_{\boldsymbol{\Theta}\boldsymbol{\Psi}})$, and $\mathcal{P}_{q,q}(\boldsymbol{H}_{\boldsymbol{\Psi}})$ is equivalent to applying the same collapse again on each cell with size $pq \times pq$ of $\boldsymbol{M}_{\boldsymbol{\Theta}}$ and $\boldsymbol{M}_{\boldsymbol{\Psi}}$ in Figure 10(b) at the level of matrix elements. Notice that not all elements of $\mathcal{P}_{pq,pq}(\boldsymbol{W} \otimes \boldsymbol{W})$ contribute to $\mathcal{P}_{p,p}(\boldsymbol{H}_{\boldsymbol{\Theta}})$, $\mathcal{P}_{p,q}(\boldsymbol{H}_{\boldsymbol{\Theta}\boldsymbol{\Psi}})$, and $\mathcal{P}_{q,q}(\boldsymbol{H}_{\boldsymbol{\Psi}})$: Only the cells of $\mathcal{P}_{pq,pq}(\boldsymbol{W} \otimes \boldsymbol{W})$ corresponding to the non-zero elements of the inflated Kronecker-sum mask $(\mathbb{1}_p \oplus \mathbb{1}_q) \otimes \mathbb{1}_{pq}$ and the elements of $\boldsymbol{M}_{\boldsymbol{\Theta}}$ and $\boldsymbol{M}_{\boldsymbol{\Psi}}$, each masked with $\mathbb{1}_p \otimes (\mathbb{1}_p \oplus \mathbb{1}_q)$ and $\mathbb{1}_q \otimes (\mathbb{1}_p \oplus \mathbb{1}_q)$, are used to compute $\mathcal{P}_{p,p}(\boldsymbol{H}_{\boldsymbol{\Theta}})$, $\mathcal{P}_{p,q}(\boldsymbol{H}_{\boldsymbol{\Theta}\boldsymbol{\Psi}})$, and $\mathcal{P}_{q,q}(\boldsymbol{H}_{\boldsymbol{\Psi}})$.

Appendix B. Computing the Newton Direction

We provide details on the coordinate descent updates for $D_{\boldsymbol{\Theta}}$. Updating $D_{\boldsymbol{\Psi}}$ can be done in a similar manner.

B1. Exact Hessian

Solving Eq. (12) for $[D_{\boldsymbol{\Theta}}]_{ij}$ with an assumption that all the other elements of $D_{\boldsymbol{\Theta}}$ are fixed amounts to solving the following optimization problem:

$$\begin{aligned}\underset{\mu}{\operatorname{argmin}} \quad & \mu\Big(q[\boldsymbol{S}]_{ij} - [\boldsymbol{W}_{\boldsymbol{\Theta}}]_{ij} + \sum_{k=1}^{q} \boldsymbol{v}_{k,i}^T D_{\boldsymbol{\Theta}} \boldsymbol{v}_{k,j} + \sum_{l=1}^{p}\sum_{k=1}^{q} \lambda_{\boldsymbol{W},lk}^2 [\boldsymbol{Q}_{\boldsymbol{\Theta}}]_{il}[\boldsymbol{Q}_{\boldsymbol{\Theta}}]_{jl} \boldsymbol{q}_{\boldsymbol{\Psi},k}^T D_{\boldsymbol{\Psi}} \boldsymbol{q}_{\boldsymbol{\Psi},k}\Big) \\ & + \frac{\mu^2}{2}\Bigg(\sum_{k=1}^{q} [\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{ij}^2 + [\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{ii}[\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{jj}\Bigg) + q\gamma_{\boldsymbol{\Theta}} \left|[\boldsymbol{\Theta}]_{ij} + [D_{\boldsymbol{\Theta}}]_{ij} + \mu\right|,\end{aligned}$$

where $\boldsymbol{v}_{k,i}$ is the ith column of $\boldsymbol{V}_{\boldsymbol{\Theta},k} = \boldsymbol{Q}_{\boldsymbol{\Theta}} \boldsymbol{\Xi}_{\boldsymbol{\Theta},k} \boldsymbol{Q}_{\boldsymbol{\Theta}}^T$. This problem has a closed-form solution

$$\mu = -c + \mathcal{S}\left(c - \frac{b}{a}, \frac{q\gamma_{\boldsymbol{\Theta}}}{a}\right),$$

where

$$\begin{aligned} a &= \sum_{k=1}^{q} [\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{ij}^2 + [\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{ii}[\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{jj}, \\ b &= q[\boldsymbol{S}]_{ij} - [\boldsymbol{W}_{\boldsymbol{\Theta}}]_{ij} + \sum_{k=1}^{q} \boldsymbol{v}_{k,i}^T D_{\boldsymbol{\Theta}} \boldsymbol{v}_{k,j} + \sum_{l=1}^{p}\sum_{k=1}^{q} \lambda_{\boldsymbol{W},lk}^2 [\boldsymbol{Q}_{\boldsymbol{\Theta}}]_{il}[\boldsymbol{Q}_{\boldsymbol{\Theta}}]_{jl} \boldsymbol{q}_{\boldsymbol{\Psi},k}^T D_{\boldsymbol{\Psi}} \boldsymbol{q}_{\boldsymbol{\Psi},k}, \\ c &= [\boldsymbol{\Theta}]_{ij} + [D_{\boldsymbol{\Theta}}]_{ij}, \end{aligned}$$

and $\mathcal{S}(z, r) = \operatorname{sign}(z) \max\{|z| - r, 0\}$ is the soft-thresholding function.

B2. Approximate Hessian

Similarly, solving Eq. (12) with $\hat{\boldsymbol{H}}$ instead of $\boldsymbol{H}$ for $[D_{\boldsymbol{\Theta}}]_{ij}$ while fixing all the other elements of $D_{\boldsymbol{\Theta}}$ amounts to solving the following optimization problem:

$$\begin{aligned}\underset{\mu}{\operatorname{argmin}} \quad & \mu\Big(q[\boldsymbol{S}]_{ij} - [\boldsymbol{W}_{\boldsymbol{\Theta}}]_{ij} + \sum_{k=1}^{K} \boldsymbol{v}_{k,i}^T D_{\boldsymbol{\Theta}} \boldsymbol{v}_{k,j} + (q-K)\boldsymbol{v}_{K,i}^T D_{\boldsymbol{\Theta}} \boldsymbol{v}_{K,j}\Big) \\ & + \frac{\mu^2}{2}\Bigg(\sum_{k=1}^{K} [\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{ij}^2 + [\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{ii}[\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{jj} + (q-K)\left([\boldsymbol{V}_{\boldsymbol{\Theta},K}]_{ij}^2 + [\boldsymbol{V}_{\boldsymbol{\Theta},K}]_{ii}[\boldsymbol{V}_{\boldsymbol{\Theta},K}]_{jj}\right)\Bigg) \\ & + q\gamma_{\boldsymbol{\Theta}} \left|[\boldsymbol{\Theta}]_{ij} + [D_{\boldsymbol{\Theta}}]_{ij} + \mu\right|,\end{aligned}$$

with a closed-form solution

$$\mu = -c + \mathcal{S}\left(c - \frac{b}{a}, \frac{q\gamma_{\boldsymbol{\Theta}}}{a}\right),$$

where

$$
\begin{aligned}
a &= \sum_{k=1}^{K} [\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{ij}^2 + [\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{ii}[\boldsymbol{V}_{\boldsymbol{\Theta},k}]_{jj} + (q-K)\left([\boldsymbol{V}_{\boldsymbol{\Theta},K}]_{ij}^2 + [\boldsymbol{V}_{\boldsymbol{\Theta},K}]_{ii}[\boldsymbol{V}_{\boldsymbol{\Theta},K}]_{jj}\right), \\
b &= q[\boldsymbol{S}]_{ij} - [\boldsymbol{W}_{\boldsymbol{\Theta}}]_{ij} + \sum_{k=1}^{K} \boldsymbol{v}_{k,i}^T D_{\boldsymbol{\Theta}} \boldsymbol{v}_{k,j} + (q-K)\boldsymbol{v}_{K,i}^T D_{\boldsymbol{\Theta}} \boldsymbol{v}_{K,j}, \\
c &= [\boldsymbol{\Theta}]_{ij} + [D_{\boldsymbol{\Theta}}]_{ij}.
\end{aligned}
$$

Online Appendix

The EiGLasso software is available at `https://github.com/SeyoungKimLab/EiGLasso`. The mouse gene-expression data are publicly available from Gonzales et al. (2018).