
DNNLasso: Scalable Graph Learning for Matrix-Variate Data

Meixia Lin

Singapore University of Technology and Design

Yangjing Zhang

Chinese Academy of Sciences

Abstract

We consider the problem of jointly learning row-wise and column-wise dependencies of matrix-variate observations, which are modelled separately by two precision matrices. Due to the complicated structure of Kronecker-product precision matrices in the commonly used matrix-variate Gaussian graphical models, a sparser Kronecker-sum structure was proposed recently based on the Cartesian product of graphs. However, existing methods for estimating Kronecker-sum structured precision matrices do not scale well to large scale datasets. In this paper, we introduce DNNLasso, a diagonally non-negative graphical lasso model for estimating the Kronecker-sum structured precision matrix, which outperforms the state-of-the-art methods by a large margin in both accuracy and computational time. Our code is available at <https://github.com/YangjingZhang/DNNLasso>.

1 INTRODUCTION

In the modern big data era, matrix-variate observations (i.e., two-dimensional grids of observations) are becoming prevalent in various domains including spatial-temporal data analysis, financial markets, genomics and imaging processing. A typical example is the spatial-temporal data in weather forecasting (Stevens and Willett, 2019; Stevens et al., 2021), in which each observation contains winter precipitations of t time lags (rows) and s locations (columns). Due to the pervasiveness of matrix-variate observations, it is important for us to understand the structure encoded in these observations. In particular, the commonly used precision

matrix (also called as the inverse covariance matrix) has received a lot of attention, as it encodes conditional independence relationships among variables.

One possible approach is to learn the precision matrices associated with the rows and columns of the matrix-variate observations separately. For example, for the spatial-temporal data in weather forecasting, we can estimate the spatial precision matrix by treating the columns of all winter precipitation observations as vector-variate spatial observations, and also estimate the temporal precision matrix by treating the rows of all winter precipitation observations as vector-variate temporal observations. In high-dimensional multivariate data analysis on vector-variate observations, many statistical models have been proposed for the estimation of the precision matrix. One widely used model is the Gaussian graphical model that learns a sparse precision matrix via an ℓ_1 -norm penalized maximum likelihood approach (Yuan and Lin, 2007; Banerjee et al., 2008; Friedman et al., 2008; Rothman et al., 2008). However, it is limited in our scenario since the observations in the Gaussian graphical model are assumed to be independent and identically distributed, while the vector-variate observations can be correlated in matrix-variate data analysis. For example, in the spatial-temporal data, not only the spatial observations can be correlated, but different temporal observations can also be correlated. Therefore, it is necessary to model the correlations among both rows and columns in the observations jointly.

Given matrix-variate data where each observation Z is a $t \times s$ matrix, it may appear tempting to stack Z as a column vector $\text{vec}(Z)$ and model Z as a ts -dimensional vector. Gaussian graphical models can be used to analyze the vectorized data, while they suffer from three shortcomings. First, estimating a $ts \times ts$ precision matrix can be daunting due to the extremely high dimension. Second, the analysis based on $\text{vec}(Z)$ ignores all row and column structural information in the observations, which is useful and sometimes vital in practice. Third, learning a precision matrix without prior structural assumptions would be impractical in high-dimension low-sample regime. Alternative ap-

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

proaches that explore the matrix nature of such matrix-variate observations are therefore attractive nowadays, among which the matrix-variate Gaussian graphical model is the most famous one.

The matrix-variate Gaussian distribution (Dawid, 1981; Gupta and Nagar, 1999; Efron, 2009; Allen and Tibshirani, 2010; Leng and Tang, 2012) of Z assumes that the covariance matrix of $\text{vec}(Z)$ has the form of a Kronecker-product (KP) between two covariance matrices, separately associated with the rows and columns of matrix-variate observations. The KP assumption for the covariance implies that the precision matrix is a also KP of two precision matrices, that is $\Omega \otimes \Gamma$, where $\Omega \in \mathbb{S}_{++}^s$ models the column-wise dependencies in Z and $\Gamma \in \mathbb{S}_{++}^t$ models the row-wise dependencies in Z . This additional KP structure has been considered in many recent works by Yin and Li (2012); Leng and Tang (2012); Tsiligkaridis and Hero (2013); Tsiligkaridis et al. (2013); Zhou (2014), which allows one to provide a satisfying estimation of the precision matrix with a small sample size. However, the KP structure leads to a relatively dense graph and a nonconvex log-likelihood, which raises great challenges in the optimization of the model. To overcome the challenges, Kalaitzis et al. (2013) introduced a sparser structure for the precision matrix by imposing a novel Kronecker-sum (KS) structure instead of KP. For estimating the KS structured precision matrix, many optimization methods have been proposed recently by Kalaitzis et al. (2013); Greenewald et al. (2019); Yoon and Kim (2022).

1.1 Related Works

Kalaitzis et al. (2013) first considered a Gaussian distribution for a matrix variable with a novel KS structure, say $\Omega \oplus \Gamma = \Omega \otimes I_t + I_s \otimes \Gamma$, for the precision matrix. Here I_t denotes a t by t identity matrix. They proposed the algorithm **BiGLasso**, a block coordinate descent algorithm for optimizing Ω and Γ in the maximum likelihood estimation approach, by regarding the columns of Ω and Γ as blocks. However, they did not tackle the non-identifiability of the diagonal entries of Ω and Γ , which is one of the key challenges in estimating the precision matrices with the KS structure. The non-identifiability arises from the fact that $\Omega \oplus \Gamma = (\Omega + cI_s) \oplus (\Gamma - cI_t)$ for all $c \in \mathbb{R}$. Namely, the KS matrix $\Omega \oplus \Gamma$ does not uniquely determine the pair (Γ, Ω) , as one can modify the diagonal entries of Ω and Γ by adding and subtracting a constant c without changing their KS. Moreover, **BiGLasso** may not scale well to median-sized datasets and the convergence of **BiGLasso** was not analyzed by Kalaitzis et al. (2013).

Later, Greenewald et al. (2019) proposed a multi-way tensor generalization of the two-way KS structure for the precision matrix studied by Kalaitzis et al. (2013).

Based on the accelerated proximal gradient method of Nesterov (2013), a method **TeraLasso** with convergence guarantees was provided by Greenewald et al. (2019). Their strategy for estimating the diagonal entries is through identifiable reparameterization with additional restrictions on the traces of Ω and Γ . Although **TeraLasso** is much better than **BiGLasso** in terms of convergence properties and computational speed, **TeraLasso** seems to be limited to graphs with only a few hundreds nodes.

More recently, based on a proximal Newton’s method for a regularized log-determinant program (Hsieh et al., 2014), an efficient algorithm **EiGLasso** was proposed by Yoon and Kim (2022) for learning the KS structured precision matrix. They introduced a new scheme for identifying the unidentifiable diagonal entries of Ω and Γ via introducing an additional constraint — restricting the trace ratio of Ω and Γ to be a fixed constant such that the KS $\Omega \oplus \Gamma$ uniquely determines Ω and Γ . The numerical experiments by Yoon and Kim (2022) show that **EiGLasso** empirically has two to three orders-of-magnitude speedup compared to **TeraLasso**, while it still takes hours on datasets with graph size of around one thousand.

1.2 Contributions

Our main contributions are summarized in four parts. First, we propose the diagonally non-negative graphical lasso (**DNNLasso**) algorithm for estimating the KS precision matrix. These additional non-negative constraints on the diagonal entries of the two precision matrices Ω and Γ naturally avoid the non-identifiability issue. Second, we develop an efficient and robust algorithm based on the alternating direction method of multipliers for solving the optimization problem in **DNNLasso**, where the computational cost and memory cost are both extremely low. Third, as a key ingredient in **DNNLasso**, we deduce the explicit solution of the proximal operator associated with the negative log-determinant of KS. As far as our knowledge goes, it is the first time that the explicit formula is provided. Last, numerical experiments on both synthetic data and real data demonstrate that **DNNLasso** outperforms the state-of-the-art **TeraLasso** and **EiGLasso** by a large margin.

1.3 Notation

$\mathbb{R}^{m \times n}$ denotes the space of m by n matrices and \mathbb{S}^n denotes the space of n by n symmetric matrices. \mathbb{S}_+^n (resp. \mathbb{S}_{++}^n) denotes the space of n by n positive semidefinite (resp. definite) matrices. $\lambda_{\min}(X)$ denotes the smallest eigenvalue of a symmetric matrix X . $\text{diag}(X)$ denotes the column vector containing the diagonal elements of the matrix X . The log-

determinant function $\log |X| := \log \det(X)$ takes the logarithm of the determinant of the positive definite matrix X . $[n] := \{1, 2, \dots, n\}$. $\delta_C(\cdot)$ denotes the indicator function of the set C , i.e., $\delta_C(x) = 0$ if $x \in C$; $\delta_C(x) = +\infty$ if $x \notin C$. For any $X \in \mathbb{R}^{n \times n}$, $\|X\|_{1,\text{off}} := \sum_{1 \leq i \neq j \leq n} |X_{ij}|$. I_t denotes a t by t identity matrix. The Kronecker-sum of matrices $\Gamma \in \mathbb{S}^t$ and $\Omega \in \mathbb{S}^s$ is $\Omega \oplus \Gamma := \Omega \otimes I_t + I_s \otimes \Gamma$.

2 ESTIMATION OF A KRONECKER-SUM PRECISION MATRIX

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with a vertex set $\mathcal{V} = \{1, \dots, ts\}$ and an edge set \mathcal{E} . Each variable (e.g., a certain feature at a particular time and location in spatial-temporal data) is associated with one vertex. A random vector $z \in \mathbb{R}^{ts}$ is said to satisfy the Gaussian graphical model with graph \mathcal{G} , if $z \sim \mathcal{N}(0, \Sigma)$ is Gaussian with $(\Sigma^{-1})_{ij} = 0$ for all $(i, j) \notin \mathcal{E}$.

Given i.i.d. observations $Z^{(1)}, \dots, Z^{(n)}$ in $\mathbb{R}^{t \times s}$ such that $\text{vec}(Z^{(k)}) \sim \mathcal{N}(0, \Sigma)$ for each k , the sparse Gaussian graphical lasso estimator (Yuan and Lin, 2007; Banerjee et al., 2008; Friedman et al., 2008; Rothman et al., 2008) for the precision matrix Σ^{-1} is given by

$$\widehat{X} \in \arg \min_{X \in \mathbb{S}_{++}^{ts}} \left\{ -\log |X| + \langle C, X \rangle + \lambda_0 \|X\|_{1,\text{off}} \right\}. \quad (1)$$

Here $\lambda_0 > 0$ is a parameter that controls the strength of the penalty, and the sample covariance matrix is $C = \frac{1}{n} \sum_{k=1}^n \text{vec}(Z^{(k)} - \bar{Z})(\text{vec}(Z^{(k)} - \bar{Z}))^T \in \mathbb{S}_+^{ts}$ with $\bar{Z} = \sum_{k=1}^n Z^{(k)} / n$. The ℓ_1 regularizer $\|X\|_{1,\text{off}}$ is added to get a sparse network, as the sparsity pattern of Σ^{-1} determines the conditional independence structure of the ts variables (Lauritzen, 1996).

When $n \ll ts$, in particular, $n = 1$, the sample covariance matrix C is a highly uncertain estimate of the truth Σ . More priors are required to obtain a satisfying estimation of the precision matrix Σ^{-1} .

2.1 Kronecker-Sum Structured Precision Matrix

A fundamental assumption in the KS model by Kalaitzis et al. (2013) is that Σ^{-1} takes the KS form of $\Sigma^{-1} = \Omega \oplus \Gamma$. Then the problem of estimating Σ^{-1} reduces to estimating Γ and Ω , which correspond to the row-wise and column-wise precision matrices in Z , respectively. Specifically, the sparse Gaussian graphical lasso estimator in (1) reduces to $\widehat{X} = \widehat{\Omega} \oplus \widehat{\Gamma}$, where

$(\widehat{\Gamma}, \widehat{\Omega})$ is an optimal solution to the problem

$$\min_{\substack{\Gamma \in \mathbb{S}_{++}^t, \\ \Omega \in \mathbb{S}_{++}^s}} \left\{ -\log |\Omega \oplus \Gamma| + \langle \Omega, W \rangle + \langle \Gamma, R \rangle \right. \\ \left. + \lambda_0 s \|\Gamma\|_{1,\text{off}} + \lambda_0 t \|\Omega\|_{1,\text{off}} \right\}. \quad (2)$$

The sample row-wise and column-wise covariance matrices are $R = \frac{1}{n} \sum_{k=1}^n Z^{(k)}(Z^{(k)})^T$ and $W = \frac{1}{n} \sum_{k=1}^n (Z^{(k)})^T Z^{(k)}$. Throughout the paper, we make the following Assumption 1.

Assumption 1. $R_{ii} > 0$, $W_{jj} > 0$, $\forall i \in [t]$, $j \in [s]$.

We take R as an example to illustrate the assumption. Suppose $R_{ii} = 0$ for some $i \in [t]$. This implies that the i -th row of $Z^{(k)}$ equals to zero for every $k \in [n]$. Consequently, we can remove the i -th rows of all matrix-variate observations prior to model construction, due to their redundant nature.

2.2 Equivalent Formulation of (2)

We are going to construct an optimal solution to the problem (2) through solving a simpler model without the positive definite constraints on Γ and Ω , since the positive definite constraints will usually raise computational challenges in designing and implementing optimization algorithms.

The basic idea is to control the nonnegativity of diagonal elements instead of forcing the positive definiteness of Γ and Ω . Specifically, we propose the diagonally non-negative graphical lasso (DNNLasso) model for estimating sparse row-wise and column-wise precision matrices simultaneously as

$$\min_{\Gamma \in \mathbb{S}^t, \Omega \in \mathbb{S}^s} \left\{ -\log |\Omega \oplus \Gamma| + \langle \Omega, W \rangle + \langle \Gamma, R \rangle \right. \\ \left. + \lambda_0 s \|\Gamma\|_{1,\text{off}} + \lambda_0 t \|\Omega\|_{1,\text{off}} \right\} \quad (3)$$

s.t. $\Omega \oplus \Gamma \in \mathbb{S}_{++}^{ts}$, $\text{diag}(\Omega) \geq 0$, $\text{diag}(\Gamma) \geq 0$,

The following proposition formally states the equivalence between problems (2) and (3). The detailed proof can be found in Appendix S1.

Proposition 2. *Problems (2) and (3) are equivalent in the following sense:*

- (a) they share the same optimal objective function value;
- (b) any optimal solution to (2) is optimal to (3);
- (c) if (Γ^*, Ω^*) is an optimal solution to (3), then

$$(\widehat{\Gamma}, \widehat{\Omega}) := \begin{cases} (\Gamma^*, \Omega^*) & \text{if } \Gamma^* \in \mathbb{S}_{++}^t, \Omega^* \in \mathbb{S}_{++}^s, \\ (\Gamma^* - cI_t, \Omega^* + cI_s) & \text{otherwise,} \end{cases} \quad (4)$$

with $c = (\lambda_{\min}(\Gamma^*) - \lambda_{\min}(\Omega^*)) / 2$, is an optimal solution to (2).

Furthermore, prior to designing an algorithm for solving (3), we present the subsequent theorem which characterizes the solution set of (3). The proof can be found in Appendix S2.

Theorem 3. *Under Assumption 1, the problem (3) admits a non-empty and bounded solution set.*

Due to the non-identifiability issue, for an optimal solution (Γ, Ω) to (3) without imposing non-negativity constraints, their diagonal entries Γ_{ii} and Ω_{jj} can possibly be extremely large values. This may cause numerical instability in optimization algorithms. However, the non-negativity constraints in (3) ensure a non-empty and bounded solution set, as demonstrated in Theorem 3. This boundedness effectively overcomes the aforementioned instability in optimization algorithms.

3 DNNLASSO

In order to obtain the DNNLasso estimator, we design an efficient and robust algorithm for solving (3). We consider an equivalent and compact form of the problem

$$\begin{aligned} \min_{\Gamma \in \mathbb{S}^t, \Omega \in \mathbb{S}^s} & \left\{ \begin{array}{l} -\log |\Omega \oplus \Gamma| + \langle \Omega, W \rangle + \langle \Gamma, R \rangle \\ + p(\Gamma) + q(\Omega) \end{array} \right\} \quad (5) \\ \text{s.t.} & \quad \Omega \oplus \Gamma \in \mathbb{S}_{++}^{ts}, \end{aligned}$$

where $p(\Gamma) = \lambda_T \|\Gamma\|_{1,\text{off}}$ if $\text{diag}(\Gamma) \geq 0$, and $+\infty$ otherwise; $q(\Omega) = \lambda_S \|\Omega\|_{1,\text{off}}$ if $\text{diag}(\Omega) \geq 0$, and $+\infty$ otherwise. The penalty function p and q necessitate the non-negativity of diagonal entries and promote the sparsity of off-diagonal entries. We can obtain the problem (3) by taking $\lambda_T = \lambda_0 s$ and $\lambda_S = \lambda_0 t$.

3.1 Alternating Direction Method of Multipliers

The alternating direction method of multipliers (ADMM) is well-suited for the problems with separable or block separable objectives and a mix of equality and inequality constraints; see Glowinski and Marroco (1975); Gabay and Mercier (1976); Eckstein and Bertsekas (1992). These characteristics align perfectly with the structure of our target problem (5) after the introduction of auxiliary variables. In fact, by introducing auxiliary variables $\Lambda \in \mathbb{S}^t$, $\Theta \in \mathbb{S}^s$, $\Xi \in \mathbb{S}^s$, we have an equivalent form of (5) as

$$\begin{aligned} \min_{\Gamma, \Lambda \in \mathbb{S}^t, \Omega, \Theta, \Xi \in \mathbb{S}^s} & \left\{ \begin{array}{l} -\log |\Omega \oplus \Gamma| + \langle \Xi, W \rangle + \langle \Gamma, R \rangle \\ + p(\Lambda) + q(\Theta) \end{array} \right\} \\ \text{s.t.} & \quad \Omega \oplus \Gamma \in \mathbb{S}_{++}^{ts}, \quad \Gamma - \Lambda = 0, \quad (6) \\ & \quad \Xi - \Theta = 0, \quad \Xi - \Omega = 0. \end{aligned}$$

Given $\sigma > 0$, for any $(\Gamma, \Omega, \Lambda, \Theta, \Xi, X, Y, U) \in \mathbb{S}^t \times \mathbb{S}^s \times \mathbb{S}^t \times \mathbb{S}^s \times \mathbb{S}^s \times \mathbb{S}^t \times \mathbb{S}^s \times \mathbb{S}^s$, the augmented Lagrangian

function associated with the above problem is

$$\begin{aligned} \mathcal{L}_\sigma(\Gamma, \Omega, \Lambda, \Theta, \Xi; X, Y, U) &= -\log |\Omega \oplus \Gamma| + \langle \Xi, W \rangle + \langle \Gamma, R \rangle + p(\Lambda) + q(\Theta) \\ &+ \delta_{\mathbb{S}_{++}^{ts}}(\Omega \oplus \Gamma) + \frac{\sigma}{2} \|\Gamma - \Lambda - \sigma^{-1} X\|_F^2 \\ &+ \frac{\sigma}{2} \|\Xi - \Theta - \sigma^{-1} Y\|_F^2 + \frac{\sigma}{2} \|\Xi - \Omega - \sigma^{-1} U\|_F^2 \\ &- \frac{1}{2\sigma} \|X\|_F^2 - \frac{1}{2\sigma} \|Y\|_F^2 - \frac{1}{2\sigma} \|U\|_F^2. \end{aligned}$$

Note that the ADMM is a primal-dual method. We are going to alternately minimize the primal variables among the two blocks (Ξ, Γ) and $(\Lambda, \Theta, \Omega)$, and then update the multipliers (X, Y, U) . See Algorithm 1 for the full algorithm. The convergence result stated in Theorem 4 follows from the well-known convergence property for the classical 2-block ADMM; see Glowinski and Marroco (1975); Gabay and Mercier (1976).

Theorem 4. *Suppose that Assumption 1 holds. Let $\{(\Gamma^k, \Omega^k, \Lambda^k, \Theta^k, \Xi^k, X^k, Y^k, U^k)\}$ be the sequence generated by Algorithm 1. Then $\{(\Gamma^k, \Omega^k)\}$ converges to an optimal solution of the problem (5).*

The bottleneck in implementing Algorithm 1 is the following steps in the k -th iteration

$$\begin{aligned} \Gamma^{k+1} &= \arg \min_{\Gamma \in \mathbb{S}^t} \left\{ -\frac{1}{\sigma} \log |\Omega^k \oplus \Gamma| + \frac{1}{2} \|\Gamma - \tilde{\Gamma}_k\|_F^2 \right\}, \\ \Omega^{k+1} &= \arg \min_{\Omega \in \mathbb{S}^s} \left\{ -\frac{1}{\sigma} \log |\Omega \oplus \Gamma^{k+1}| + \frac{1}{2} \|\Omega - \tilde{\Omega}_k\|_F^2 \right\}, \end{aligned}$$

given some $\tilde{\Gamma}_k \in \mathbb{S}^t$ and $\tilde{\Omega}_k \in \mathbb{S}^s$. In the subsequent section, we provide an efficient procedure for this.

3.2 Proximal Operators Associated with the Negative Log-determinant KS Function

Given $\Gamma \in \mathbb{S}^t$ and $\beta > 0$, we investigate the proximal operator associated with $-\beta \log |\cdot \oplus \Gamma|$ defined by

$$\Psi_{\text{Left}, \beta, \Gamma}(\Omega) = \arg \min_{\Upsilon \in \mathbb{S}^s} \left\{ \frac{1}{2} \|\Upsilon - \Omega\|_F^2 - \beta \log |\Upsilon \oplus \Gamma| \right\},$$

for $\Omega \in \mathbb{S}^s$. The following proposition gives an efficient procedure to compute $\Psi_{\text{Left}, \beta, \Gamma}(\cdot)$.

Proposition 5. *Given $\beta > 0$ and $\Gamma \in \mathbb{S}^t$ with eigenvalues $\lambda_1, \dots, \lambda_t$. For any $\Omega \in \mathbb{S}^s$ with the eigenvalue decomposition $\Omega = Q \Sigma_\Omega Q^T$, $\Sigma_\Omega = \text{Diag}(\mu_1, \dots, \mu_s)$, we have*

$$\Psi_{\text{Left}, \beta, \Gamma}(\Omega) = Q \text{Diag}(\alpha_1, \dots, \alpha_s) Q^T,$$

where for every $j \in [s]$, α_j is the unique solution to the univariate nonlinear equation

$$\alpha_j - \mu_j - \sum_{i=1}^t \frac{\beta}{\alpha_j + \lambda_i} = 0, \quad \alpha_j > -\min_{i \in [t]} \lambda_i. \quad (7)$$

Proof. The first part of the proof is based on the orthogonally invariant property of $\log |\cdot \oplus \Gamma|$. Namely, given $\Omega \in \mathbb{S}^s$, it holds that $\log |\Omega \oplus \Gamma| = \log |(M\Omega M^T) \oplus \Gamma|$ for any orthogonal matrix M . This implies that for any $\Omega \in \mathbb{S}^s$ with eigenvalue decomposition $\Omega = Q\Sigma_\Omega Q^T$, we have

$$\begin{aligned} & \Psi_{\text{Left},\beta,\Gamma}(\Omega) \\ &= \arg \min_{\Upsilon \in \mathbb{S}^s} \left\{ \frac{1}{2} \|\Upsilon - Q\Sigma_\Omega Q^T\|_F^2 - \beta \log |\Upsilon \oplus \Gamma| \right\} \\ &= \arg \min_{\Upsilon \in \mathbb{S}^s} \left\{ \frac{1}{2} \|Q^T \Upsilon Q - \Sigma_\Omega\|_F^2 - \beta \log |(Q^T \Upsilon Q) \oplus \Gamma| \right\} \\ &= Q\Psi_{\text{Left},\beta,\Gamma}(\Sigma_\Omega)Q^T. \end{aligned}$$

The above equality for the orthogonally invariant functions can also be found in Eqn (6.11) of Parikh and Boyd (2014). Since Σ_Ω is a diagonal matrix and $\log |\cdot \oplus \Gamma|$ is orthogonally invariant, we can see that $\Psi_{\text{Left},\beta,\Gamma}(\Sigma_\Omega)$ is also a diagonal matrix. Moreover, $\Psi_{\text{Left},\beta,\Gamma}(\Sigma_\Omega) = \text{Diag}(\alpha)$ satisfies $\alpha = \arg \min_{\alpha \in \mathbb{R}^s} \left\{ \sum_{j=1}^s (\alpha_j - \mu_j)^2 / 2 - \beta \sum_{i=1}^t \sum_{j=1}^s \log(\alpha_j + \lambda_i) \right\}$, which holds due to the fact the eigenvalues of the KS of two matrices are the pairwise sums of the eigenvalues of the two matrices (Horn and Johnson, 1991). Note that the above minimization problem can be solved component-wisely. Thus α_j should satisfy (7) due to the first-order optimality condition, for every $j \in [s]$. Lastly, we prove that for any given $x \in \mathbb{R}$ the equation $h(y) = (y - x)/\beta - \sum_{i=1}^t 1/(y + \lambda_i) = 0$ admits a unique solution on the interval $(-\min_i \lambda_i, +\infty)$. It is true because h is increasing on $(-\min_i \lambda_i, +\infty)$, $\lim_{y \rightarrow (-\min_i \lambda_i)^+} h(y) = -\infty$, and $\lim_{y \rightarrow +\infty} h(y) = +\infty$. This completes the proof. \square

Given $\lambda_1, \dots, \lambda_t$ and $\beta > 0$, let the univariate function $\psi(\cdot; \lambda_1, \dots, \lambda_t, \beta) : \mathbb{R} \rightarrow \mathbb{R}$ be defined as

$$\begin{aligned} & \psi(x; \lambda_1, \dots, \lambda_t, \beta) \\ &:= \left\{ y \left| \frac{y - x}{\beta} = \sum_{i=1}^t \frac{1}{y + \lambda_i}, y > -\min_{i \in [t]} \lambda_i \right. \right\}. \quad (8) \end{aligned}$$

which is well-defined according to the proof of Proposition 5. Moreover, the function value of $\psi(\cdot; \lambda_1, \dots, \lambda_t, \beta)$ can be calculated by the Newton's method or the bisection method. By solving s univariate nonlinear equations, we obtain that $\alpha_j = \psi(\mu_j; \lambda_1, \dots, \lambda_t, \beta)$, $j \in [s]$.

Similarly, given $\Omega \in \mathbb{S}^s$ and $\beta > 0$, the proximal operator associated with $-\beta \log |\Omega \oplus \cdot|$ is

$$\Psi_{\text{Right},\beta,\Omega}(\Gamma) = \arg \min_{\Delta \in \mathbb{S}^t} \left\{ \frac{1}{2} \|\Delta - \Gamma\|_F^2 - \beta \log |\Omega \oplus \Delta| \right\},$$

for $\Gamma \in \mathbb{S}^t$. Analogous to Proposition 5, we can provide an efficient procedure to compute $\Psi_{\text{Right},\beta,\Omega}(\cdot)$. Details are in Appendix S3.

Algorithm 1 : DNNLasso

Input: Given sample covariance matrices $R \in \mathbb{S}_+^t$, $W \in \mathbb{S}_+^s$ and a parameter $\lambda_0 > 0$.

Initialization: Set $\lambda_T = \lambda_0 s$, $\lambda_S = \lambda_0 t$, $\tau = 1.618$. Set $k \leftarrow 0$. Choose $\sigma > 0$ and an initial point $(\Omega^0, \Lambda^0, \Theta^0, X^0, Y^0, U^0) \in \mathbb{S}^s \times \mathbb{S}^t \times \mathbb{S}^s \times \mathbb{S}^t \times \mathbb{S}^s \times \mathbb{S}^s$.

repeat

Step 1. Compute

$$\begin{aligned} \Gamma^{k+1} &= \Psi_{\text{Right},1/\sigma,\Omega^k}(\Lambda^k + \frac{X^k}{\sigma} - \frac{R}{\sigma}), \\ \Xi^{k+1} &= \frac{1}{2}(\Theta^k + \Omega^k + \frac{Y^k + U^k - W}{\sigma}). \end{aligned}$$

Step 2. Let $\tilde{\Lambda} = \Gamma^{k+1} - X^k/\sigma$, $\tilde{\Theta} = \Xi^{k+1} - Y^k/\sigma$. Compute $\Lambda^{k+1} \in \mathbb{S}^t$ and $\Theta^{k+1}, \Omega^{k+1} \in \mathbb{S}^s$ as

$$\begin{aligned} \Lambda_{ij}^{k+1} &= \begin{cases} \max(0, \tilde{\Lambda}_{ij}) & \text{if } i = j, \\ \text{sgn}(\tilde{\Lambda}_{ij}) \max(|\tilde{\Lambda}_{ij}| - \frac{\lambda_T}{\sigma}, 0) & \text{if } i \neq j, \end{cases} \\ \Theta_{ij}^{k+1} &= \begin{cases} \max(0, \tilde{\Theta}_{ij}) & \text{if } i = j, \\ \text{sgn}(\tilde{\Theta}_{ij}) \max(|\tilde{\Theta}_{ij}| - \frac{\lambda_S}{\sigma}, 0) & \text{if } i \neq j, \end{cases} \\ \Omega^{k+1} &= \Psi_{\text{Left},1/\sigma,\Gamma^{k+1}}(\Xi^{k+1} - \frac{U^k}{\sigma}). \end{aligned}$$

Step 3. Update the multipliers by

$$\begin{aligned} X^{k+1} &= X^k - \tau\sigma(\Gamma^{k+1} - \Lambda^{k+1}), \\ Y^{k+1} &= Y^k - \tau\sigma(\Xi^{k+1} - \Theta^{k+1}), \\ U^{k+1} &= U^k - \tau\sigma(\Xi^{k+1} - \Omega^{k+1}). \end{aligned}$$

Step 4. Set $k \leftarrow k + 1$.

until Stopping criterion is satisfied.

Output: An approximate solution $(\hat{\Gamma}, \hat{\Omega})$ computed as follows: $(\hat{\Gamma}, \hat{\Omega}) = (\Gamma^k, \Omega^k)$ if $\Gamma^k \succ 0$, $\Omega^k \succ 0$; and $(\hat{\Gamma}, \hat{\Omega}) = (\Gamma^k - cI_t, \Omega^k + cI_s)$ with $c = (\lambda_{\min}(\Gamma^k) - \lambda_{\min}(\Omega^k))/2$ otherwise.

3.3 The Full Algorithm

We provide the pseudocode of DNNLasso in Algorithm 1, where the computation of $\Psi_{\text{Right},\beta,\Omega}(\cdot)$ and $\Psi_{\text{Left},\beta,\Gamma}(\cdot)$ is described in Section 3.2. Note that the non-negative constraints on diagonal elements only bring in the computation of $\max(0, \tilde{\Lambda}_{ii})$ and $\max(0, \tilde{\Theta}_{ii})$, of which the cost is negligible.

We provide in Table 1 a comparison of DNNLasso with three existing methods BiGLasso, TeraLasso, and

EiGLasso, in terms of memory cost and computational cost per iteration.

Table 1: Comparison among algorithms in terms of memory cost and computational cost per iteration.

| | Memory cost | Computational cost |
|-----------|---------------------|--------------------------|
| BiGLasso | $O(t^2 s^2)$ | $O(N_1 t^3 + N_1 s^3)$ |
| TeraLasso | $O(ts + t^2 + s^2)$ | $O(2ts + t^3 + s^3)$ |
| EiGLasso | $O(Kt^2 + Ks^2)$ | $O(N_2 Kt^3 + N_2 Ks^3)$ |
| DNNLasso | $O(t^2 + s^2)$ | $O(t^3 + s^3)$ |

In Table 1, (1) N_1 represents the average number of iterations of the subroutines in BiGLasso (i.e., the coordinate descent procedure implemented in GLasso to estimate the precision matrix for a simple graph); (2) $K \leq \min(t, s)$ is a user-specified parameter in the Hessian approximation of EiGLasso, which is typically chosen within the range from 1 to 10. The default setting is $K = 1$ in their codes. (3) N_2 represents the average number of iterations of the subroutines in EiGLasso (i.e., the coordinate descent method to compute the Newton directions by minimizing second-order approximations of the objective function).

4 NUMERICAL EXPERIMENTS

We compare our DNNLasso with TeraLasso¹ (Greenwald et al., 2019) and EiGLasso² (Yoon and Kim, 2022) on both synthetic and real data, and we use their default settings for parameters and initialization. All experiments were conducted in Matlab (version 9.11) on a Windows workstation (32-core, Intel Xeon Gold 6226R @ 2.90GHz, 128 Gigabytes of RAM). Since the ADMM is a primal-dual method, we terminate it when the relative KKT error is less than a given tolerance, for example, 10^{-6} . Details are in Appendix S4.

As a warm-start of our implementation, we first run a simpler variant of DNNLasso by eliminating the auxiliary variables Ξ . As we can see from (6), Ξ is a “duplicate” of the column-wise precision matrix Ω , and at the optimal point they should be identical, i.e., $\Xi = \Omega$. Without Ξ , this variant is simpler and has less variables compared with DNNLasso. The pseudocode of this variant is given in Appendix S5.

4.1 Synthetic Data

We use two types of graph structures by Yoon and Kim (2022) for generating the ground truth $\Gamma \in \mathbb{S}_{++}^t$ (the same for $\Omega \in \mathbb{S}_{++}^s$). And we sample n observations from the Gaussian distribution $\mathcal{N}(0, (\Omega \oplus \Gamma)^{-1})$.

¹<https://github.com/kgreenwald/teralasso>

²<https://github.com/SeyoungKimLab/EiGLasso>

Type 1. We first generate a sparse matrix $A \in \mathbb{R}^{t \times t}$ where $P(A_{ij} = -1) = \frac{1}{2}(1 - \rho)$, $P(A_{ij} = 1) = \frac{1}{2}(1 - \rho)$, $P(A_{ij} = 0) = \rho$, and $\rho \geq 0$ is chosen such that A roughly has $10t$ nonzero entries. Then we set $\Gamma = AA^T + 10^{-4}I_t + \text{diag}(d_1, \dots, d_t)$ with d_i uniformly random on $[0, 0.1]$.

Type 2. We set $\Gamma \in \mathbb{R}^{t \times t}$ to be block diagonal which contains 10 blocks and each block is generated as a graph in **Type 1**. In this case we choose ρ such that there are t nonzero entries in each block.

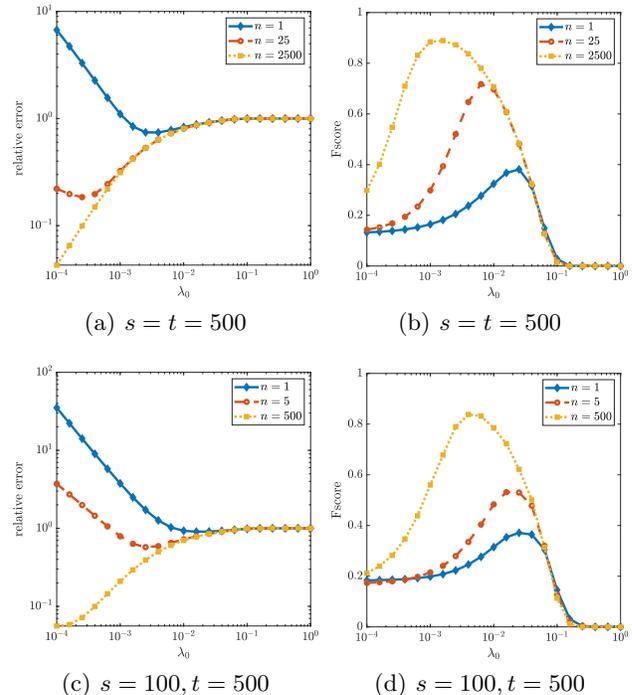


Figure 1: Relative error (a,c) / Fscore (b,d) against λ_0 for synthetic graphs of **Type 2** with dimension $s = t = 500$ (a,b) / $s = 100, t = 500$ (c,d), and sample size $n = 1, st/10000$ or $st/100$.

We start from medium-size graphs by considering a balanced graph size $s = t = 500$ and an unbalanced one $s = 100, t = 500$. We vary the sample size n in $\{1, st/10000, st/100\}$ and select the regularization parameter λ_0 in (2) from the candidate set $\{10^{-4}, 10^{-3.9}, 10^{-3.8}, \dots, 10^{-0.1}, 10^0\}$. We apply DNNLasso for solving (2) with tolerance 10^{-6} to obtain an estimated solution $(\tilde{\Gamma}, \tilde{\Omega})$. We compute the relative error of the estimated solution $(\tilde{\Gamma}, \tilde{\Omega})$ with respect to the ground truth (Γ, Ω) via $(\|\tilde{\Gamma}_{\text{off}} - \Gamma_{\text{off}}\| / \|\Gamma_{\text{off}}\| + \|\tilde{\Omega}_{\text{off}} - \Omega_{\text{off}}\| / \|\Omega_{\text{off}}\|) / 2$, where the matrix Γ_{off} is constructed from Γ by setting its diagonal entries to be zero. In addition, to measure the accuracy in identifying edges, we report the averaged FScore, that is $(\text{Fscore}(\tilde{\Gamma}_{\text{off}}, \Gamma_{\text{off}}) + \text{Fscore}(\tilde{\Omega}_{\text{off}}, \Omega_{\text{off}})) / 2$. Here

$\text{Fscore}(\tilde{\Gamma}_{\text{off}}, \Gamma_{\text{off}}) = \frac{2\text{tp}}{2\text{tp} + \text{fp} + \text{fn}}$, where tp, fp, and fn denote the number of true positive, false positive, and false negative edges between the truth Γ_{off} and the estimator $\tilde{\Gamma}_{\text{off}}$, respectively.

Figure 1 plots the relative error and Fscore against λ_0 obtained by DNNLasso for different dimensions and sample sizes. Overall, we can see that the relative error is smaller and the Fscore is higher for a larger sample size. In particular, when the sample size n is $st/100$, Figures 1(b) and 1(d) show that the best Fscore is larger than 0.8, which is close to the ideal value 1.

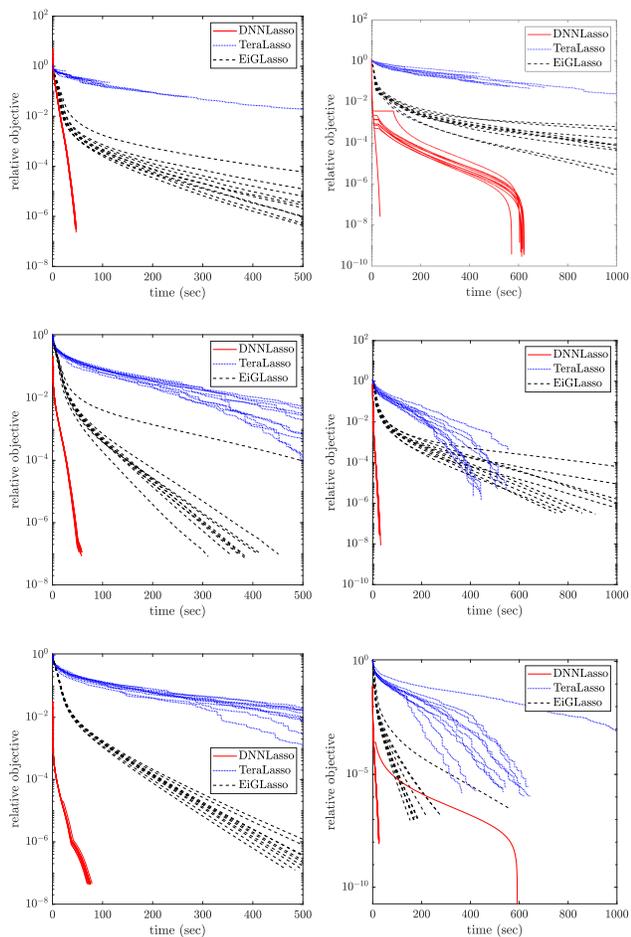


Figure 2: Relative objective function value $(f^k - f^*)/f^*$ against time on synthetic graphs of **Type 2** with dimension $s = t = 500$ (left column) / $s = 100, t = 500$ (right column), and sample size $n = 1, st/10000$ or $st/100$ (rows from upper to lower).

We fix λ_0 to be the best parameter from the candidate set which achieves the highest Fscore (the best value can be seen from Figure 1(b) and Figure 1(d)). We compare the three methods DNNLasso, TeraLasso, and EiGLasso on synthetic graphs of **Type 2**. We use the solution of DNNLasso with tolerance 10^{-8} as a bench-

mark and denote the corresponding objective function value as f^* . For the objective function value f^k at the k -th iteration of one method, we compute the relative objective function value $(f^k - f^*)/f^*$. In Figure 2, we show the relative objective function value against computational time for different methods on 10 replications. We can see from Figure 2 that our DNNLasso always achieves a better objective value within a shorter time. Besides, EiGLasso seems to be faster than TeraLasso for a large majority of instances.

Next we compare the three methods on **Type 1** graphs with relatively large balanced size $s = t = 1000$ and unbalanced size $s = 1000, t = 400$. Since we are interested in the efficiency of each algorithm for low-sample cases, we fix the sample size $n = 1$ and choose parameters $\lambda_0 = 10^{-2}$ or $10^{-1.6}$. Figure 3 illustrates the relative objective function value against computational time in different scenarios. We can see that DNNLasso outperforms TeraLasso and EiGLasso by a large margin.

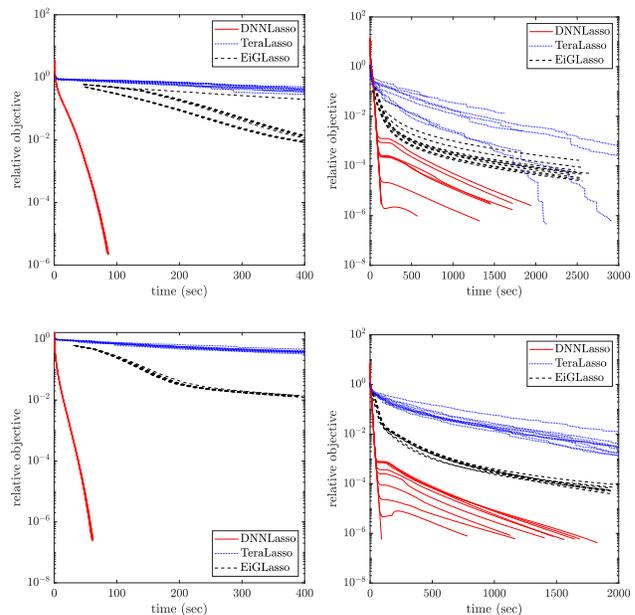


Figure 3: Relative objective function values against time on graphs of **Type 1** with sample size $n = 1$ and dimensions $s = t = 1000$ (upper row) / $s = 1000, t = 400$ (bottom row), $\lambda_0 = 10^{-2}$ (left column) / $\lambda_0 = 10^{-1.6}$ (right column).

4.2 COIL100 Video Data

In this section, we adopt the data from the Columbia Object Image Library (COIL) (Nene et al., 1996). The data contains 100 objects, and for each object, it contains $s = 72$ frames (color images with the resolution of $t = 128 \times 128$ pixels) of the rotating object from dif-

ferent angles (every 5°). Our goal is to jointly recover the conditional dependency structure over the frames and the structure over the pixels. In consideration of the computational complexity, we choose to reduce the resolution of each frame. Likewise, Kalaitzis et al. (2013) consider the reduced resolution of 8×8 . We pick one object (a box of cold medicine) from the data, which is illustrated in Figure 4. From Figure 4, we can roughly recognize the object from the compressed images of 32×32 pixels in the second row, but it is hard to recognize the object from the compressed images of 8×8 pixels in the third row. This implies that the reduced resolution of 32×32 might be a better choice for graph learning than the reduced resolution of 8×8 .

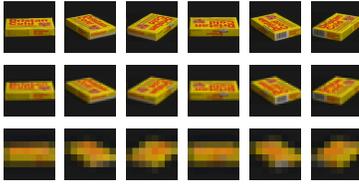


Figure 4: A rotating box of cold medicine in COIL100 video data. First row: original resolution of 128×128 pixels. Second (resp. third) row: reduced resolution of 32×32 (resp. 8×8) pixels.

We first conduct experiments on $s = 72$ frames with the reduced resolution of $t = 32 \times 32$ pixels. We select parameter λ_0 from the set $\{10^{-3.5}, 10^{-3.4}, \dots, 10^{-2.1}, 10^{-2}\}$ under the Bayesian information criterion (BIC), and then compare our DNNLasso with TeraLasso and EiGLasso. We terminate DNNLasso with tolerance 5×10^{-3} .

Figure 5(a) plots the BIC value and sparsity level against λ_0 . For an estimated pair $(\tilde{\Gamma}, \tilde{\Omega})$, the BIC value is computed as $\text{BIC}(\tilde{\Gamma}, \tilde{\Omega}) = -\log |\tilde{\Omega} \oplus \tilde{\Gamma}| + \langle \tilde{\Omega}, W \rangle + \langle \tilde{\Gamma}, R \rangle + (0.5 \log(n)/n + 0.2 \log(st))(\|\tilde{\Omega}\|_{0,\text{off}} + \|\tilde{\Gamma}\|_{0,\text{off}})$, and the sparsity level is computed as $(\|\tilde{\Omega}\|_{0,\text{off}} + \|\tilde{\Gamma}\|_{0,\text{off}})/(s(s-1) + t(t-1))$, where $\|\cdot\|_{0,\text{off}}$ denotes the number of nonzero off-diagonal entries in a matrix. We can see from Figure 5(a) that the sparsity level is roughly decreased from 15% to 3% as λ_0 increases and $\lambda_0 = 10^{-2.4}$ achieves the best BIC. Figure 5(b) illustrates the objective function value against computational time with $\lambda_0 = 10^{-2.4}$. We do not include TeraLasso in this instance since it failed to return a positive definite solution $\Omega \oplus \Gamma$. From Figure 5(b) we can see that DNNLasso took less than 20 seconds and achieved a much better objective function value than EiGLasso after more than 1500 seconds. In Figure 5(c,d), we demonstrate the sparsity pattern of the matrix $\tilde{\Omega} \in \mathbb{S}^s$ estimated by DNNLasso, namely the relationship graph of frames from different angles. Here, we only show the off diagonal entries of $\tilde{\Omega}$, whereby zero

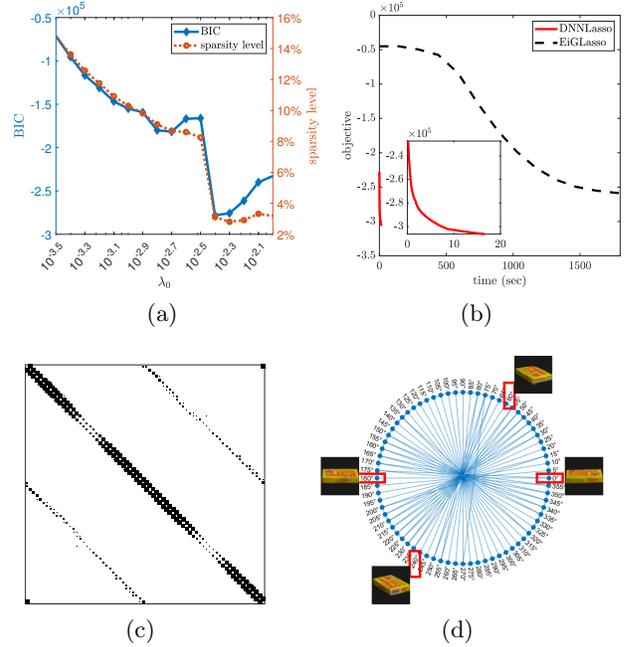


Figure 5: On $s = 72$ frames with $t = 32 \times 32$ pixels. (a) The BIC and sparsity level against λ_0 . (b) The relative objective function value against computational time. (c) Sparsity pattern of the matrix $\tilde{\Omega} \in \mathbb{S}^s$ estimated by DNNLasso (i.e., the correlation pattern among frames from different angles). (d) Relationship graph of frames from different angles.

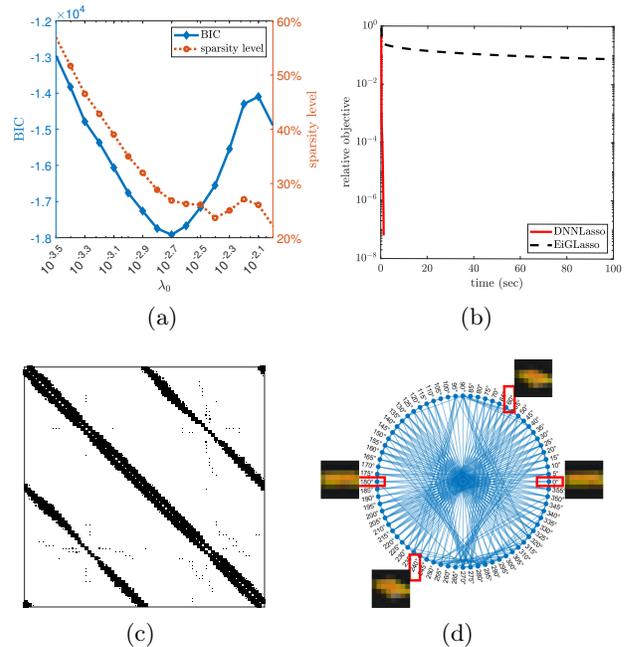


Figure 6: On $s = 72$ frames with $t = 8 \times 8$ pixels.

values ($\tilde{\Omega}_{ij} = 0$) are white, nonzero values ($\tilde{\Omega}_{ij} \neq 0$)

are represented by black squares, and the size of a black square is proportional to the weight $|\hat{\Omega}_{ij}|$. The relationship graph in Figure 5(c) indicates that an image observed from x° is connected not only to adjacent images from $(x \pm 5)^\circ$, but also to images from $(x \pm 180)^\circ$. This observation is intuitively right as the object is a box. The two images from 0° and 180° (and also those from 60° and 240°) have similar exteriors, as plotted in Figure 5(d).

Besides, we report the experimental results on the reduced resolution data with $t = 8 \times 8$ pixels. We find that there are some unexpected correlations among frames shown in Figure 6(d), compared with Figure 5(d). One possible reason is that images of 8×8 pixels are too blur to identify.

More results on the synthetic data and video data are in Appendix S6 and Appendix S7, respectively.

5 CONCLUSION

In this paper, we propose DNNLasso, an efficient framework for estimating the KS-structured precision matrix for matrix-variate data. We develop an efficient and robust ADMM based algorithm for solving it and derive an explicit solution of proximal operators associated with the negative log-determinant of KS function for the first time. Numerical experiments demonstrate that DNNLasso is superior to the existing methods by a large margin. However, our algorithm still relies on the eigenvalue decompositions in each iteration. In future work, we will consider partial or certain economical eigenvalue decomposition to further reduce computational cost. Additionally, we acknowledge that our algorithm is a first-order method that may not be efficient enough for obtaining highly accurate solutions. Therefore, we may include some second-order information into the algorithm to further speed up the process.

Acknowledgements

Meixia Lin is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 grant call (MOE-T2EP20123-0013) and the Singapore University of Technology and Design under MOE Tier 1 Grant SKI 2021_02_08. Yangjing Zhang is supported by the National Natural Science Foundation of China under grant number 12201617.

References

G. I. Allen and R. Tibshirani. Transposable regularized covariance models with an application to missing data imputation. *The Annals of Applied Statistics*, 4(2):764, 2010.

- O. Banerjee, L. E. Ghaoui, and A. d’Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.
- A. P. Dawid. Some matrix-variate distribution theory: Notational considerations and a Bayesian application. *Biometrika*, 68(1):265–274, 1981.
- J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992.
- B. Efron. Are a set of microarrays independent of each other? *The Annals of Applied Statistics*, 3(3):922, 2009.
- J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.
- D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers and Mathematics with Applications*, 2(1):17–40, 1976.
- R. Glowinski and A. Marroco. Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de dirichlet non linéaires. *Revue française d’automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(R2):41–76, 1975.
- 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.
- A. K. Gupta and D. K. Nagar. *Matrix Variate Distributions*. Chapman and Hall/CRC, 1999.
- R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- 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(1):2911–2947, 2014.
- A. Kalaitzis, J. Lafferty, N. D. Lawrence, and S. Zhou. The bigraphical lasso. In *International Conference on Machine Learning*, pages 1229–1237, 2013.
- S. L. Lauritzen. *Graphical Models*, volume 17. Clarendon Press, 1996.
- C. Leng and C. Y. Tang. Sparse matrix graphical models. *Journal of the American Statistical Association*, 107(499):1187–1200, 2012.
- S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (COIL-20). *Technical report UCUCS-005-96*, 1996.

- Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.
- N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.
- A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.
- A. Stevens and R. Willett. Graph-guided regularization for improved seasonal forecasting. *Climate Informatics*, 2019.
- A. Stevens, R. Willett, A. Mamalakis, E. Foufoula-Georgiou, A. Tejedor, J. T. Randerson, P. Smyth, and S. Wright. Graph-guided regularized regression of pacific ocean climate variables to increase predictive skill of southwestern US winter precipitation. *Journal of Climate*, 34(2):737–754, 2021.
- 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.
- T. Tsiligkaridis, A. O. Hero III, and S. Zhou. On convergence of Kronecker graphical lasso algorithms. *IEEE Transactions on Signal Processing*, 61(7):1743–1755, 2013.
- 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 for scalable sparse Kronecker-sum inverse covariance estimation. *Journal of Machine Learning Research*, 23(110):1–39, 2022.
- M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.
- S. Zhou. Gemini: Graph estimation with matrix variate normal instances. *The Annals of Statistics*, 42(2):532–562, 2014.

DNNLasso: Scalable Graph Learning for Matrix-Variate Data: Supplementary Materials

For ease of reference, sections and equations belonging to this supplementary material are prefixed with an ‘S’.

S1 PROOF OF PROPOSITION 2

Proof. We first give some notations. Denote the function

$$\begin{aligned} f_0(\Gamma, \Omega) &= -\log |\Omega \oplus \Gamma| + \langle \Omega, W \rangle + \langle \Gamma, R \rangle + \lambda_0 s \|\Gamma\|_{1,\text{off}} + \lambda_0 t \|\Omega\|_{1,\text{off}} \\ &= -\log |\Omega \oplus \Gamma| + \langle \Omega \oplus \Gamma, C \rangle + \lambda_0 \|\Omega \oplus \Gamma\|_{1,\text{off}}, \end{aligned}$$

where C is the sample covariance matrix in (1). Denote the optimal objective function values of (2) and (3) as f_2^* and f_3^* , respectively. Denote the feasible set of (2) as

$$\mathcal{F}_2 = \{(\Gamma, \Omega) \mid \Gamma \in \mathbb{S}_{++}^t, \Omega \in \mathbb{S}_{++}^s\},$$

and the feasible set of (3) as

$$\mathcal{F}_3 = \{(\Gamma, \Omega) \mid \Omega \oplus \Gamma \in \mathbb{S}_{++}^{st}, \text{diag}(\Omega) \geq 0, \text{diag}(\Gamma) \geq 0\}.$$

(a) Obviously, we can see from the definition that $\mathcal{F}_2 \subseteq \mathcal{F}_3$, and thus

$$f_2^* \geq f_3^*.$$

Suppose (Γ^*, Ω^*) is an optimal solution to (3) and then $f_3^* = f(\Gamma^*, \Omega^*)$. By our construction of $(\widehat{\Gamma}, \widehat{\Omega})$, it follows from the non-identifiability of diagonals that

$$\widehat{\Omega} \oplus \widehat{\Gamma} = \Omega^* \oplus \Gamma^*$$

and therefore $f(\widehat{\Gamma}, \widehat{\Omega}) = f(\Gamma^*, \Omega^*)$. Moreover,

$$\lambda_{\min}(\Omega^* \oplus \Gamma^*) = \lambda_{\min}(\Omega^*) + \lambda_{\min}(\Gamma^*) > 0,$$

and by the choice of c , we have $\widehat{\Gamma} \in \mathbb{S}_{++}^t, \widehat{\Omega} \in \mathbb{S}_{++}^s$. Namely, $(\widehat{\Gamma}, \widehat{\Omega}) \in \mathcal{F}_2$ is a feasible point to (2). Then

$$f_2^* \leq f(\widehat{\Gamma}, \widehat{\Omega}) = f(\Gamma^*, \Omega^*) = f_3^*.$$

We have proved that $f_2^* = f_3^*$.

(b) The statement holds naturally since $\mathcal{F}_2 \subseteq \mathcal{F}_3$ and $f_2^* = f_3^*$.

(c) As shown in (a), $(\widehat{\Gamma}, \widehat{\Omega})$ is optimal to (2) as it is feasible and attains the optimal objective function value, that is,

$$(\widehat{\Gamma}, \widehat{\Omega}) \in \mathcal{F}_2, \quad f(\widehat{\Gamma}, \widehat{\Omega}) = f(\Gamma^*, \Omega^*) = f_3^* = f_2^*.$$

The proof is completed. □

S2 PROOF OF THEOREM 3

Proof. First of all, we can prove that the function $(\Gamma, \Omega) \rightarrow -\log |\Omega \oplus \Gamma|$ is lower semi-continuous. In fact, it follows from the fact that $\alpha \geq -\log |\Omega \oplus \Gamma|$ whenever $\alpha = \lim \alpha_k$, $\Omega = \lim \Omega_k$, $\Gamma = \lim \Gamma_k$ for sequences $\{\alpha_k\}$, $\{\Omega_k\}$, $\{\Gamma_k\}$ such that $\alpha_k \geq -\log |\Omega_k \oplus \Gamma_k|$ for every k . Denote the objective function in (3) as

$$f(\Gamma, \Omega) = \begin{cases} -\log |\Omega \oplus \Gamma| + \langle \Omega, W \rangle + \langle \Gamma, R \rangle + \lambda_0 s \|\Gamma\|_{1,\text{off}} + \lambda_0 t \|\Omega\|_{1,\text{off}} & \text{if } \text{diag}(\Omega) \geq 0, \text{diag}(\Gamma) \geq 0 \\ +\infty & \text{otherwise} \end{cases}.$$

Then it can be seen that $f(\cdot, \cdot)$ is lower semi-continuous, convex, proper in $\mathbb{S}^t \times \mathbb{S}^s$.

Next we compute the recession function of $f(\cdot, \cdot)$ based on Rockafellar (1996, Theorem 8.5). We have that

$$\begin{aligned} (f0_+)(\Gamma, \Omega) &= \lim_{\alpha \rightarrow +\infty} \frac{f(I_t + \alpha\Gamma, I_s + \alpha\Omega) - f(I_t, I_s)}{\alpha} \\ &= \begin{cases} \langle \Omega, W \rangle + \langle \Gamma, R \rangle + \lambda_0 s \|\Gamma\|_{1,\text{off}} + \lambda_0 t \|\Omega\|_{1,\text{off}} & \text{if } \text{diag}(\Gamma) \geq 0, \text{diag}(\Omega) \geq 0, \Omega \oplus \Gamma \in \mathbb{S}_+^{st} \\ +\infty & \text{otherwise} \end{cases}, \end{aligned}$$

where the last equality holds since when $\text{diag}(\Gamma) \geq 0$, $\text{diag}(\Omega) \geq 0$, $\Omega \oplus \Gamma \in \mathbb{S}_+^{st}$, we have

$$\begin{aligned} &\lim_{\alpha \rightarrow +\infty} \frac{f(I_t + \alpha\Gamma, I_s + \alpha\Omega) - f(I_t, I_s)}{\alpha} \\ &= \lim_{\alpha \rightarrow +\infty} \frac{-\log |I_s \oplus I_t + \alpha(\Omega \oplus \Gamma)| + \log |I_s \oplus I_t|}{\alpha} + \langle \Omega, W \rangle + \langle \Gamma, R \rangle + \lambda_0 s \|\Gamma\|_{1,\text{off}} + \lambda_0 t \|\Omega\|_{1,\text{off}} \\ &= \langle \Omega, W \rangle + \langle \Gamma, R \rangle + \lambda_0 s \|\Gamma\|_{1,\text{off}} + \lambda_0 t \|\Omega\|_{1,\text{off}}. \end{aligned}$$

Therefore, the recession cone of $f(\cdot, \cdot)$ is

$$\begin{aligned} &\{(\Gamma, \Omega) \mid \text{diag}(\Gamma) \geq 0, \text{diag}(\Omega) \geq 0, \Omega \oplus \Gamma \in \mathbb{S}_+^{st}, \langle \Omega, W \rangle + \langle \Gamma, R \rangle + \lambda_0 s \|\Gamma\|_{1,\text{off}} + \lambda_0 t \|\Omega\|_{1,\text{off}} \leq 0\} \\ &= \{(\Gamma, \Omega) \mid \text{diag}(\Gamma) \geq 0, \text{diag}(\Omega) \geq 0, \Omega \oplus \Gamma \in \mathbb{S}_+^{st}, \langle \Omega \oplus \Gamma, C \rangle = 0, \|\Gamma\|_{1,\text{off}} = 0, \|\Omega\|_{1,\text{off}} = 0\} \\ &= \{(\Gamma, \Omega) \mid \Omega = \text{Diag}(\alpha_1, \dots, \alpha_s), \Gamma = \text{Diag}(\gamma_1, \dots, \gamma_t), \alpha_i \geq 0, \gamma_j \geq 0, (\alpha_i + \gamma_j) \sum_{k=1}^n (Z_{ji}^{(k)})^2 = 0\}, \end{aligned}$$

where the first equality follows from that $\langle \Omega, W \rangle + \langle \Gamma, R \rangle = \langle \Omega \oplus \Gamma, C \rangle \geq 0$ as both $\Omega \oplus \Gamma$ and the sample covariance matrix C in (1) are positive semidefinite; the second equality uses the expression of diagonal entries of C ; $\text{Diag}(\alpha_1, \dots, \alpha_s)$ returns a square diagonal matrix with the elements α_i on the main diagonal.

We prove by contradiction to see that $\alpha_i + \gamma_j = 0$, for all i, j . Suppose there exist i_1 and j_1 such that $\alpha_{j_1} + \gamma_{i_1} > 0$, then $\sum_{k=1}^n (Z_{i_1 j_1}^{(k)})^2 = 0$. Under Assumption 1, we have $R_{i_1 i_1} > 0$ and thus $Z_{i_1 \cdot}^{(k)} \neq 0$ for some k . Namely, there exists $j_2 \neq j_1$ such that $Z_{i_1 j_2}^{(k)} \neq 0$, which implies $\sum_{k=1}^n (Z_{i_1 j_2}^{(k)})^2 \neq 0$ and then $\alpha_{j_2} + \gamma_{i_1} = 0$. Similarly, under Assumption 1, we have $W_{j_1 j_1} > 0$, which implies that $\alpha_{j_1} + \gamma_{i_2} = 0$ for some $i_2 \neq i_1$. Therefore, $\alpha_{j_1} + \gamma_{i_1} = \alpha_{j_1} - \alpha_{j_2} > 0$ and $\alpha_{j_2} + \gamma_{i_2} = \alpha_{j_2} - \alpha_{j_1} < 0$, which is contradictory to $\alpha_i + \gamma_j \geq 0$. Therefore, all α_i 's and γ_j 's are zero and the recession cone contains zero alone.

Lastly, by Rockafellar (1996, Theorem 27.1), the minimum set of f is a non-empty bounded set. To this end, we have proven that problem (3) admits a non-empty and bounded solution set. \square

S3 PROXIMAL OPERATOR ASSOCIATED WITH $-\beta \log |\Omega \oplus \cdot|$

The following proposition provides an efficient procedure to compute $\Psi_{\text{Right}, \beta, \Omega}(\cdot)$. The proof is omitted as it is similar to the case in Proposition 5.

Proposition S1. *Given $\beta > 0$ and $\Omega \in \mathbb{S}^s$ with eigenvalues μ_1, \dots, μ_s . For any $\Gamma \in \mathbb{S}^t$ with the eigenvalue decomposition $\Gamma = P \Sigma_\Gamma P^T$, $\Sigma_\Gamma = \text{Diag}(\lambda_1, \dots, \lambda_t)$, we have*

$$\Psi_{\text{Right}, \beta, \Omega}(\Gamma) = P \text{Diag}(\alpha_1, \dots, \alpha_t) P^T,$$

where for every $i = 1, \dots, t$, α_i is the unique solution to the univariate nonlinear equation

$$\alpha_i - \lambda_i - \sum_{j=1}^s \frac{\beta}{\alpha_i + \mu_j} = 0, \quad \alpha_i > -\min_{j=1, \dots, s} \mu_j.$$

S4 RELATIVE KKT ERROR

Here is a remark on the stopping criterion of DNNLasso. Note that the Karush-Kuhn-Tucker (KKT) optimality conditions of (6) are given as follows:

$$\begin{cases} -\nabla\Gamma + R - X = 0 & \text{where } \nabla\Gamma = \frac{d \log |\Omega \oplus \Gamma|}{d\Gamma} \\ -\nabla\Omega + U = 0 & \text{where } \nabla\Omega = \frac{d \log |\Omega \oplus \Gamma|}{d\Omega} \\ \Lambda - \text{Prox}_p(\Lambda - X) = 0 \\ \Theta - \text{Prox}_q(\Theta - Y) = 0 \\ W - Y - U = 0 \\ \Gamma - \Lambda = 0 \\ \Xi - \Theta = 0 \\ \Xi - \Omega = 0 \end{cases}.$$

It can be proved that if Γ has the eigenvalues $\lambda_1, \dots, \lambda_t$ and the corresponding eigenvectors $u_1, \dots, u_t \in \mathbb{R}^t$, and Ω has the eigenvalues μ_1, \dots, μ_s and the corresponding eigenvectors $v_1, \dots, v_s \in \mathbb{R}^s$, we will have

$$\begin{aligned} \frac{d \log |\Omega \oplus \Gamma|}{d\Gamma} &= \sum_{i=1}^t \left(\sum_{j=1}^s \frac{1}{\mu_j + \lambda_i} \right) u_i u_i^T, \\ \frac{d \log |\Omega \oplus \Gamma|}{d\Omega} &= \sum_{j=1}^s \left(\sum_{i=1}^t \frac{1}{\mu_j + \lambda_i} \right) v_j v_j^T. \end{aligned}$$

We terminate DNNLasso when the relative KKT error is less than a given tolerance, for example, 10^{-6} . Here the relative KKT error refers to the degree to which the KKT optimality conditions are violated. It is a commonly used metric for assessing the accuracy of approximate solutions obtained from primal-dual methods. The relative KKT error refers to the maximum value of the following quantities:

$$\frac{\|-\nabla\Gamma + R - X\|_F}{1 + \|\nabla\Gamma\|_F + \|R\|_F + \|X\|_F}, \quad \text{where } \nabla\Gamma = \frac{d \log |\Omega \oplus \Gamma|}{d\Gamma}, \quad (\text{S1})$$

$$\frac{\|-\nabla\Omega + U\|_F}{1 + \|\nabla\Omega\|_F + \|U\|_F}, \quad \text{where } \nabla\Omega = \frac{d \log |\Omega \oplus \Gamma|}{d\Omega}, \quad (\text{S2})$$

$$\frac{\|\Lambda - \text{Prox}_p(\Lambda - X)\|_F}{1 + \|\Lambda\|_F + \|\text{Prox}_p(\Lambda - X)\|_F}, \quad (\text{S3})$$

$$\frac{\|\Theta - \text{Prox}_q(\Theta - Y)\|_F}{1 + \|\Theta\|_F + \|\text{Prox}_q(\Theta - Y)\|_F}, \quad (\text{S4})$$

$$\frac{\|W - Y - U\|_F}{1 + \|W\|_F + \|Y\|_F + \|U\|_F}, \quad (\text{S5})$$

$$\frac{\|\Gamma - \Lambda\|_F}{1 + \|\Gamma\|_F + \|\Lambda\|_F}, \quad (\text{S6})$$

$$\frac{\|\Xi - \Theta\|_F}{1 + \|\Xi\|_F + \|\Theta\|_F}, \quad (\text{S7})$$

$$\frac{\|\Xi - \Omega\|_F}{1 + \|\Xi\|_F + \|\Omega\|_F}. \quad (\text{S8})$$

These quantities include primal feasibility residuals (equations (S6), (S7), and (S8)), dual feasibility residuals (equations (S1), (S2), and (S5)), as well as complementarity slackness between primal and dual variables (equations (S3) and (S4)). Therefore, in our proposed algorithm, we use the relative KKT error to evaluate the optimality of the obtained approximate solutions.

For better illustration, we attach one example of the primal and dual residual plot on a Type 2 synthetic graph with $s = t = 500$ in Figure S1. Moreover, we also plot the corresponding complementarity slackness between primal and dual variables.

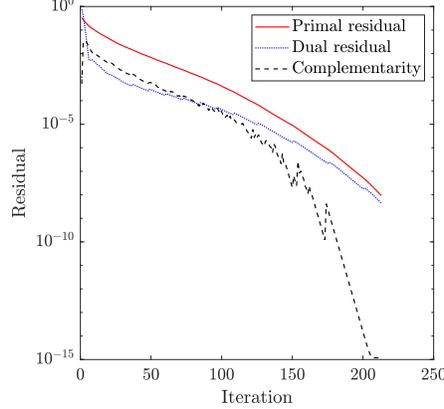


Figure S1: Relative KKT error decreasing on a **Type 2** synthetic graph with $s = t = 500$, $\lambda_0 = 10^{-2}$.

S5 A SIMPLER VARIANT OF DNNLASSO

This section introduces a simpler variant of DNNLasso (Algorithm S1) by eliminating the auxiliary variables Ξ . As we can see from (6), Ξ is a “duplicate” of the column-wise precision matrix Ω , and at the optimal point they should be identical, i.e., $\Xi = \Omega$. Without Ξ , this variant is simpler and has less variables compared with DNNLasso.

Algorithm S1 : A variant of DNNLasso

Input: Given sample covariance matrices $R \in \mathbb{S}_+^t$, $W \in \mathbb{S}_+^s$ and a parameter $\lambda_0 > 0$.

Initialization: Set $\lambda_T = \lambda_0 s$, $\lambda_S = \lambda_0 t$ and $\tau = 1.618$. Choose $\sigma > 0$. Choose an initial point $(\Omega^0, \Lambda^0, \Theta^0, X^0, Y^0) \in \mathbb{S}^s \times \mathbb{S}^t \times \mathbb{S}^s \times \mathbb{S}^t \times \mathbb{S}^s$. Set $k \leftarrow 0$.

repeat

Step 1. Compute

$$\Gamma^{k+1} = \Psi_{\text{Right}, 1/\sigma, \Omega^k}(\Lambda^k + \frac{X^k}{\sigma} - \frac{R}{\sigma}), \quad \Omega^{k+1} = \Psi_{\text{Left}, 1/\sigma, \Gamma^{k+1}}(\Theta^k + \frac{Y^k}{\sigma} - \frac{W}{\sigma}).$$

Step 2. Let $\tilde{\Lambda} = \Gamma^{k+1} - X^k/\sigma$, $\tilde{\Theta} = \Omega^{k+1} - Y^k/\sigma$. Compute $\Lambda^{k+1} \in \mathbb{S}^t$ and $\Theta^{k+1} \in \mathbb{S}^s$ as

$$\Lambda_{ij}^{k+1} = \begin{cases} \max(0, \tilde{\Lambda}_{ii}) & \text{if } i = j \\ \text{sgn}(\tilde{\Lambda}_{ij}) \max(|\tilde{\Lambda}_{ij}| - \frac{\lambda_T}{\sigma}, 0) & \text{if } i \neq j \end{cases}, \quad \Theta_{ij}^{k+1} = \begin{cases} \max(0, \tilde{\Theta}_{ii}) & \text{if } i = j \\ \text{sgn}(\tilde{\Theta}_{ij}) \max(|\tilde{\Theta}_{ij}| - \frac{\lambda_S}{\sigma}, 0) & \text{if } i \neq j \end{cases}.$$

Step 3. Update the multipliers by

$$X^{k+1} = X^k - \tau\sigma(\Gamma^{k+1} - \Lambda^{k+1}), \quad Y^{k+1} = Y^k - \tau\sigma(\Omega^{k+1} - \Theta^{k+1}).$$

Step 4. Set $k \leftarrow k + 1$.

until Stopping criterion is satisfied.

Output: An approximate solution $(\hat{\Gamma}, \hat{\Omega})$ computed as follows: $(\hat{\Gamma}, \hat{\Omega}) = (\Gamma^k, \Omega^k)$ if $\Gamma^k \succ 0$, $\Omega^k \succ 0$; and $(\hat{\Gamma}, \hat{\Omega}) = (\Gamma^k - cI_t, \Omega^k + cI_s)$ with $c = (\lambda_{\min}(\Gamma^k) - \lambda_{\min}(\Omega^k))/2$ otherwise.

S6 MORE NUMERICAL RESULTS ON LARGER SYNTHETIC DATA

To better demonstrate the superior performance of **DNNLasso**, we show the comparison of **DNNLasso**, **TeraLasso** and **EiGLasso** for learning the KS-structured precision matrices on larger synthetic data sets in the following Table S1. Specifically, we run our experiments on two types of graphs with dimensions $s = t = 1500, 2000, 3000, 4000, 5000$. We set the maximum computational time of each method as 2 hours.

Table S1: Comparison of three methods on large synthetic data with $\lambda_0 = 0.01$.

| Graph | (s, t) | DNNLasso | | TeraLasso | | EiGLasso | |
|--------|--------------|----------|-----------|-----------|-----------|----------|-----------|
| | | Time (s) | Obj | Time (s) | Obj | Time (s) | Obj |
| Type 1 | (1500, 1500) | 186 | -2.2596e6 | 7289 | -2.2322e6 | 7337 | -2.2576e6 |
| | (2000, 2000) | 405 | -3.8564e6 | 7256 | -3.5893e6 | 7488 | -3.8375e6 |
| | (3000, 3000) | 583 | -8.0733e6 | 7244 | -6.1875e6 | 7362 | -7.5311e6 |
| | (4000, 4000) | 1549 | -1.3884e7 | – | – | – | – |
| | (5000, 5000) | 3386 | -2.1174e7 | – | – | – | – |
| Type 2 | (1500, 1500) | 240 | -2.5098e6 | 7208 | -2.4604e6 | 7474 | -2.5064e6 |
| | (2000, 2000) | 526 | -4.2782e6 | 7282 | -3.9906e6 | 7638 | -4.2661e6 |
| | (3000, 3000) | 610 | -8.9069e6 | 7251 | -7.2176e6 | 7670 | -8.1600e6 |
| | (4000, 4000) | 1558 | -1.5239e7 | – | – | – | – |
| | (5000, 5000) | 3338 | -2.3122e7 | – | – | – | – |

We can see from Table S1 that, our proposed **DNNLasso** performs better than the other two estimators by a large margin, in the sense that we take much less time but get much better objective function values. Moreover, we can see that even for the smallest data set with $s = t = 1500$, **TeraLasso** and **EiGLasso** can not achieve satisfactory performance within 2 hours, while our proposed **DNNLasso** is able to solve the largest problem with $s = t = 5000$ within one hour.

S7 MORE EXPERIMENTAL RESULTS ON COIL100 VIDEO DATA

We report more experimental results on COIL100 Video Data here for illustration. We pick another object (a cargo) from the data, which is illustrated in Figure S2. From Figure S2, we again find that the reduced resolution of 32×32 is a good representation of the original 128×128 resolution, while the reduced resolution of 8×8 may not provide enough information. That is one evidence why we need an efficient and robust algorithm for estimating the large-scale KS-structured precision matrix otherwise it is impossible for us to deal with the case for $t = 32 \times 32 = 1024$ pixels within seconds.

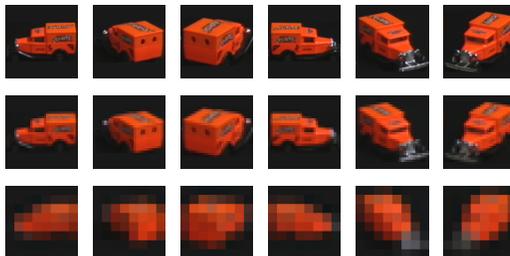


Figure S2: A rotating box of a cargo in COIL100 video data. First row: original resolution of 128×128 pixels. Second (resp. third) row: reduced resolution of 32×32 (resp. 8×8) pixels.

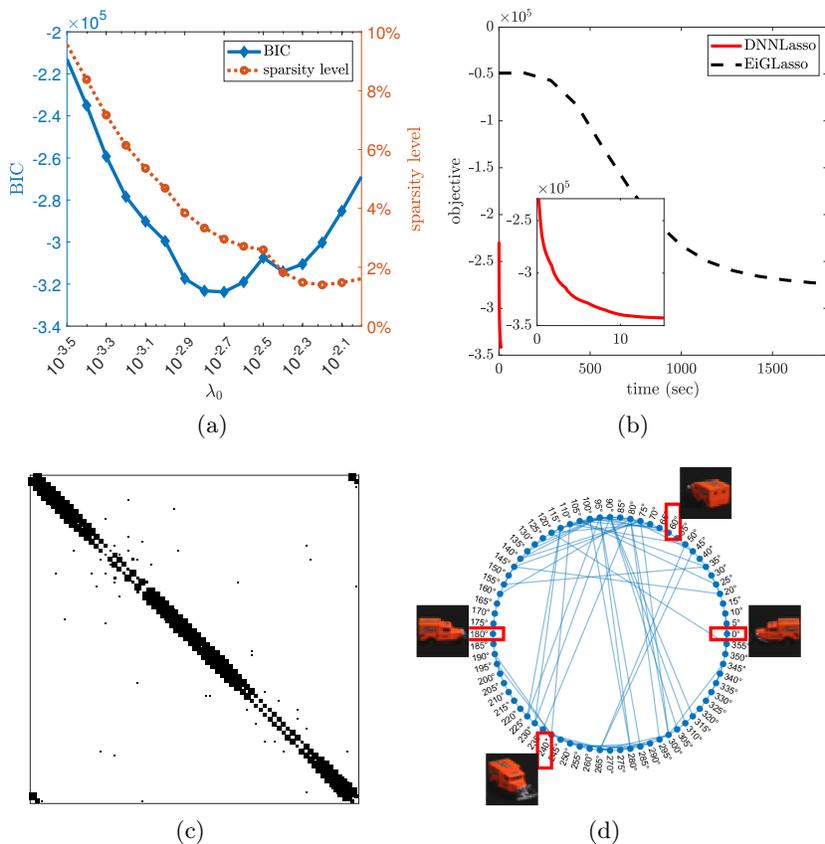


Figure S3: On $s = 72$ frames with $t = 32 \times 32$ pixels. (a) The BIC and sparsity level against λ_0 . (b) The relative objective function value against computational time. (c) Sparsity pattern of the matrix $\tilde{\Omega} \in \mathbb{S}^s$ estimated by DNNLasso (i.e., the correlation pattern among frames from different angles). (d) Relationship graph of frames from different angles.

We first conduct experiments on $s = 72$ frames with the reduced resolution of $t = 32 \times 32$ pixels. We select parameter λ_0 from the set $\{10^{-3.5}, 10^{-3.4}, \dots, 10^{-2.1}, 10^{-2}\}$ under the Bayesian information criterion (BIC), and then compare our DNNLasso with TeraLasso and EiGLasso. We terminate DNNLasso with tolerance 5×10^{-3} . Figure S3(a) plots the BIC value and sparsity level against λ_0 and Figure S3(b) illustrates the objective function value against computational time with λ_0 with the best BIC. TeraLasso is not included in the figure as it failed to return a positive definite solution $\Omega \oplus \Gamma$. From Figure S3(b) we can see that the objective function value obtained by DNNLasso after 10 seconds is far much better than that obtained by EiGLasso after more than 1600 seconds. In Figure S3(c,d), we demonstrate the sparsity pattern of the matrix $\tilde{\Omega} \in \mathbb{S}^s$ estimated by DNNLasso, namely the relationship graph of frames from different angles. The relationship graph in Figure S3(c) indicates a manifold-like structure where image observed from x° and $(x + 360)^\circ$ join, which is expected from a 360° rotation. The interesting different structure between Figure S3(c) and Figure 5(c) come from the natural observations from the objects: the box of cold medicine admits 180 degree symmetry while the cargo doesn't.

In addition, we also report the experimental results on the reduced resolution data with $t = 8 \times 8$ pixels in Figure S4. We again find that there are some unexpected correlations among frames shown in Figure S4(d), compared with Figure S3(d), which implies that images of 8×8 pixels are too blur to identify.

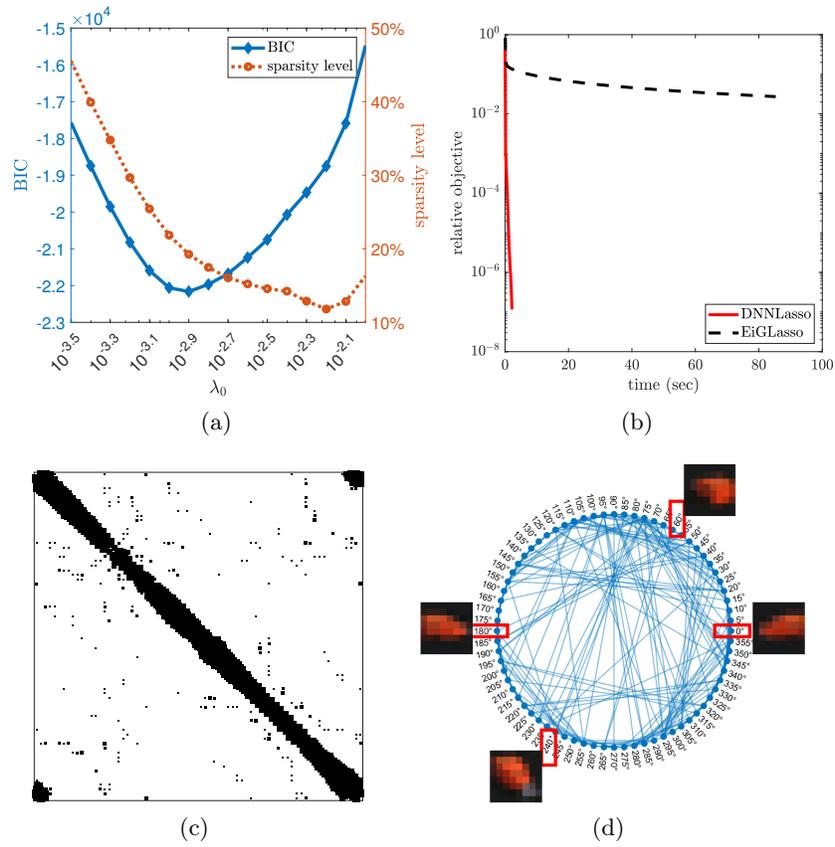


Figure S4: On $s = 72$ frames with $t = 8 \times 8$ pixels.

References

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1996.

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes/No/Not Applicable]
Yes. See e.g., Section 3.
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes/No/Not Applicable]
Yes. See e.g., Theorem 4 and Table 1.
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable]
Not Applicable.
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable]
Yes. Theorems 3 and 4 are under Assumption 1 and we clearly stated the assumption.
 - (b) Complete proofs of all theoretical results. [Yes/No/Not Applicable]
No. We have complete proofs for Proposition 2 and Theorem 3 in the appendix. And complete proofs for Proposition 5 are in given in the main paper. However, Theorem 4 follows from the well-known convergence property for the classical 2-block ADMM and we omit the proof.
 - (c) Clear explanations of any assumptions. [Yes/No/Not Applicable]
Yes.
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable]
Yes.
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/Not Applicable]
Yes.
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes/No/Not Applicable]
Yes.
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/Not Applicable]
Yes. See the first paragraph of Section 4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Yes/No/Not Applicable]
Yes.
 - (b) The license information of the assets, if applicable. [Yes/No/Not Applicable]
Not Applicable.
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/Not Applicable]
Not Applicable.
 - (d) Information about consent from data providers/curators. [Yes/No/Not Applicable]
Not Applicable.
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/Not Applicable]
Not Applicable.
5. If you used crowdsourcing or conducted research with human subjects, check if you include:

- (a) The full text of instructions given to participants and screenshots. [Yes/No/Not Applicable]
Not Applicable.
- (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/Not Applicable]
Not Applicable.
- (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/Not Applicable]
Not Applicable.