Transfer Learning for Precision Matrix Estimation via Nodewise Regression

Abstract

This paper is concerned with estimating the high-dimensional precision matrix of a target distribution when multiple auxiliary source datasets are available. We propose a transfer learning procedure for high-dimensional precision matrix estimation based on nodewise regression, which enables one to estimate the precision matrix column by column. We study the sampling properties of the resulting estimate and establish its rate of convergence. We conducted numerical studies to assess the finite sample performance of the proposed procedure and compare its performance with other existing ones. Our numerical comparison shows the superior performance of the proposed procedure. We further illustrate the proposed methodology via an empirical analysis of S&P500 stock data.

Keywords: Covariance matrix estimation, Lasso, SCAD

Mathematics Subject Classification (2020): 62H12, 62F12, 62J07

1 Introduction

Precision matrix estimation plays a crucial role in high-dimensional statistics and its applications. In multivariate analysis, the precision matrix underlies the Hotelling statistic T^2 and determines the decision boundary in discriminant analysis. In finance, it is central to Markowitz's portfolio optimization for allocating assets to maximize return for a given level of risk. In genomics, it helps infer gene regulatory networks that govern gene expression. The precision matrix is also intrinsically connected to Gaussian graphical models (GGMs), as it encodes conditional dependence among Gaussian variables: nonzero off-diagonal entries correspond to edges in the associated graph, with magnitudes reflecting the strength of conditional association.

When the dimension p is comparable to or exceeds the sample size n, the sample covariance matrix becomes nearly singular or ill-conditioned, making direct inversion unstable. In this setting, consistent precision matrix estimation requires additional structure, most commonly sparsity. This has motivated a broad class of regularized estimators. The graphical lasso (GLasso) estimates a sparse precision matrix via ℓ_1-penalized Gaussian likelihood and yields a positive-definite estimator (Yuan and Lin, 2007; Friedman et al., 2008). Another class, constrained ℓ_1-minimization for inverse matrix estimation (CLIME), casts the problem as a set of constrained optimization programs that can be solved efficiently using linear programming

routines (Cai et al., 2011, 2016). Alternatively, regression-based approaches estimate the precision matrix through nodewise regressions and have been applied to high-dimensional inference procedures (Meinshausen and Bühlmann, 2006; Yuan, 2010; van de Geer et al., 2014). Under suitable sparsity and regularity conditions, these methods achieve consistency and can recover the underlying conditional dependence structure. However, they are largely developed for a single target domain and do not address borrowing strength across heterogeneous but related studies. This work aims to fill this gap.

In high-dimensional statistics, transfer learning refers to the use of auxiliary source datasets to improve estimation and prediction in a target dataset when the underlying parameters are related but not identical. A common modeling perspective represents the source-target discrepancy as a bias term and develops procedures that borrow strength via a pooled estimator while accounting for this bias, often through an explicit correction step, thereby maintaining interpretability and enabling theoretical guarantees. For high-dimensional linear regression, Bastani (2021) formulated a single-source version of this problem and analyzed a two-step procedure for estimation and prediction. Li et al. (2022) extended this framework to accommodate multiple sources and introduced model selection aggregation to address non-informative sources. Building on these ideas, Tian and Feng (2023) generalized the approach to high-dimensional generalized linear models (GLMs) and proposed confidence interval construction via the desparsified Lasso and an algorithm-free transferable-source detection strategy based on three-fold validation. In contrast to these two-step approaches, Liu (2024) proposed the unified transfer learning framework (UTrans), which incorporates a regularizer that promotes sparsity in both the target coefficients and the source-target bias. Closely related, Liu et al. (2025) develop a UTrans-like co-regularization transfer framework for high-dimensional GLMs (CoRT), jointly learning the target coefficients and source-target contrasts to stabilize transfer beyond the classical two-step paradigm. By not restricting to the Lasso penalty, UTrans naturally accommodates other commonly used regularizers, including nonconvex penalties such as SCAD (Fan and Li, 2001) and MCP (Zhang, 2010).

In this paper, we consider the problem of estimating a target precision matrix in a high-dimensional setting when multiple auxiliary source datasets are available. Let superscripts (0) represent the target and superscript (k), $k \in \{1, ..., K\}$, represent the sources. Our goal is to estimate the target precision matrix $\mathbf{\Omega}^{(0)} = (\mathbf{\Sigma}^{(0)})^{-1}$ with target observation $\mathbf{x}^{(0)} = (X_1^{(0)}, \ldots, X_p^{(0)})^\top$ with a covariance matrix $\mathbf{\Sigma}^{(0)}$ while also incorporate source observation $\mathbf{x}^{(k)} = (X_1^{(k)}, \ldots, X_p^{(k)})^\top$ with a covariance matrix $\mathbf{\Sigma}^{(k)}$ in a way that can reduce the bias introduced with the help of transfer learning. Despite extensive work on high-dimensional precision matrix estimation, there are limited existing methods that address transfer learning for precision matrix estimation. Our proposed method is built on the notion of nodewise regression, which allows for estimating precision matrix column by column. For each column, say the j-th column, we obtain the transfer learning estimation of regression coefficients

$$\widehat{\boldsymbol{b}}_j = \underset{\boldsymbol{b} \in \mathbb{R}^{(K+1)(p-1)}}{\operatorname{argmin}} \left\{ \frac{1}{2N} \sum_{k=0}^{K} \left\| \mathbf{X}_j^{(k)} - \mathbf{X}_{-j}^{(k)} \boldsymbol{\beta}_j^{(k)} \right\|_2^2 + p_{\lambda_j}(\boldsymbol{b}) \right\} \tag{1}$$

and estimated error variance $\widehat{\tau}_j$. Here p_{λ_j} denotes a generic sparsity-inducing penalty function,

such as the Lasso (Tibshirani, 1996) and the SCAD penalty (Fan and Li, 2001). An estimate of $\mathbf{\Omega}^{(0)}$ can be constructed as $\widehat{\mathbf{\Omega}} = (\widehat{\mathbf{\Omega}}_1, \ldots, \widehat{\mathbf{\Omega}}_p)$, where

$$\begin{cases} \widehat{\Omega}_{jj} = 1/\widehat{\tau}_j \\ \widehat{\mathbf{\Omega}}_{-j,j} = -\widehat{\boldsymbol{\beta}}_j/\widehat{\tau}_j \end{cases}. \tag{2}$$

Detailed notation will be introduced at the end of this section.

In addition to the methodological development, we established sufficient conditions for statistical consistency and derived the convergence rate of the estimator. We introduce a matrix-level similarity metric

$$\|\mathbf{\Omega}^{(k)} - \mathbf{\Omega}^{(0)}\|_1 \tag{3}$$

and prove that this matrix L_1 similarity induces bounds on the corresponding regression-level bias norms, i.e. the precision-matrix level consistency and convergence rate can be guaranteed by regression-level results. We allow two variance estimators in nodewise regression chosen based on how similar the diagonal entries of source and target precision matrices are. We obtained the following convergence rate under regularity conditions

$$\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}\|_1 \lesssim \left(s^{\frac{1}{2}} + C_1\|\mathbf{\Omega}^{(0)}\|_1\right)\left(\frac{s\log p}{N}\right)^{\frac{1}{2}} + (sh)^{\frac{1}{2}}\left(\frac{\log p}{N}\right)^{\frac{1}{4}}. \tag{4}$$

The rate depends on sparsity s, transferring level h, and total sample size N. To our knowledge, this is the first work to systematically study transfer learning for precision matrix estimation via nodewise regression.

This paper is organized as follows. In Section 2, we introduce the proposed method and algorithm in detail, establish sufficient conditions for statistical consistency, and prove the rate of convergence of the estimator. In Section 3, we compare the proposed algorithm to others through simulation studies, apply our proposed algorithm to the portfolio optimization problem using S&P 500 data. A brief summary and concluding remarks are given in Section 4. Technical proofs and some additional simulation results are presented in the Appendix.

Notation: Let superscripts (0) represent the target and superscript (k), $k \in \{1, ..., K\}$, represent the sources. The target observation $\mathbf{x}^{(0)} = (X_1^{(0)}, \ldots, X_p^{(0)})^\top \in \mathbb{R}^p$ has mean $\boldsymbol{\mu}^{(0)}$ and covariance matrix $\mathbf{\Sigma}^{(0)}$, and the source observation $\mathbf{x}^{(k)} = (X_1^{(k)}, \ldots, X_p^{(k)})^\top \in \mathbb{R}^p$ has mean $\boldsymbol{\mu}^{(k)}$ and covariance matrix $\mathbf{\Sigma}^{(k)}$, $k \in \{1, ..., K\}$. A row of $\mathbf{X}$ is denoted as $\mathbf{X}_{i,\cdot}$, or $\mathbf{x}_i$ for simplicity. A column of $\mathbf{X}$ is denoted as $\mathbf{X}_{\cdot,j}$, or $\mathbf{X}_j$ for simplicity. In most cases, we use index k for source data, index j for columns, and index i for rows.

For a p-dimensional vector $\mathbf{x} = (X_1, \cdots, X_p)^\top$, $\|\mathbf{x}\|_q$ stands for the ℓ_q norm. In particular, $\|\mathbf{x}\|_0$ is the number of nonzero elements, and $\|\mathbf{x}\|_\infty$ is the maximum norm $\max_i |X_i|$. For a matrix $\mathbf{A} \in \mathbb{R}^{p\times q}$, we denote its (i,j)-th element by $\mathbf{A}_{ij}$, the j-th column of $\mathbf{A}$ by $\mathbf{A}_j$, and the submatrix of $\mathbf{A}$ with the j-th column removed by $\mathbf{A}_{-j}$. Denote $\mathbf{A}_{-i,-j}$ as the submatrix of $\mathbf{A}$ with it i-th row and the j-th column removed, $\mathbf{A}_{\mathcal{I},\mathcal{J}}$ as the submatrix of $\mathbf{A}$ with rows and columns indexed by $\mathcal{I}$ and $\mathcal{J}$. For a symmetric matrix $\mathbf{A}$, let $\lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$ denote the largest and smallest eigenvalues of $\mathbf{A}$, respectively. The matrix L_1 norm is $\|\mathbf{A}\|_1 = \max_j \sum_{i=1}^p |\mathbf{A}_{ij}|$. The spectral norm, also referred to as the operator norm or matrix L_2 norm, is $\|\mathbf{A}\|_2 = \sup_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2 =$

$\left\{\lambda_{\max}\left(\mathbf{A}^\top\mathbf{A}\right)\right\}^{1/2} = sv_1(\mathbf{A})$, the largest singular value of $\mathbf{A}$. The Frobenius norm is $\|\mathbf{A}\|_F = \left(\sum_{i=1}^p\sum_{j=1}^q \mathbf{A}_{ij}^2\right)^{1/2} = \left\{\mathrm{tr}\left(\mathbf{A}^\top\mathbf{A}\right)\right\}^{1/2}$.

Let $a_n = O_P(b_n)$ and $a_n \lesssim b_n$ denote $P(|a_n/b_n| \le c) \to 1$ for some $c < \infty$. Let $a_n = o_P(b_n)$ and $a_n \ll b_n$ denote $P(|a_n/b_n| > c) \to 0$ for any $c > 0$. Let $a_n \asymp b_n$ denote that a_n/b_n converges to some positive constant. Let $a \vee b$ denote $\max\{a, b\}$ and $a \wedge b$ denote $\min\{a, b\}$. We use $c_0, c_1, \ldots$ and $C_0, C_1, \ldots$ as generic constants which can be different at different places.

2 A New Estimator and its Sampling Properties

In this section, we propose a transfer learning framework for estimating the target precision matrix by exploiting nodewise regression representations across target and sources. We begin with an introduction to the statistical setting and modeling assumptions.

2.1 Statistical setting and models

Let $\mathbf{x}^{(0)} = (X_1^{(0)}, \ldots, X_p^{(0)})^\top \in \mathbb{R}^p$ denote the target distribution with mean $\boldsymbol{\mu}^{(0)}$ and covariance $\boldsymbol{\Sigma}^{(0)}$, and for $k \in \{1, \ldots, K\}$ let $\mathbf{x}^{(k)} = (X_1^{(k)}, \ldots, X_p^{(k)})^\top$ denote the k-th source population with mean $\boldsymbol{\mu}^{(k)}$ and covariance $\boldsymbol{\Sigma}^{(k)}$. We observe a target sample $\{\mathbf{x}_i^{(0)}\}_{i=1}^{n_0}$ and K source samples $\{\mathbf{x}_i^{(k)}\}_{i=1}^{n_k}$, and our goal is to estimate the target precision matrix $\boldsymbol{\Omega}^{(0)} = (\boldsymbol{\Sigma}^{(0)})^{-1}$ in a high-dimensional regime, in which p is allowed to exceed n_0. We adopt the nodewise regression representation, which reduces estimation of $\boldsymbol{\Omega}^{(0)}$ to p high-dimensional sparse regressions and provides a natural route for transfer learning from sources to target.

For notational convenience, consider a generic random vector $\mathbf{x} = (X_1, \ldots, X_p)^\top$, either target or source, with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Let $\mathbf{x}_{-j}$ denote the subvector excluding X_j. The linear projection of X_j onto $\mathbf{x}_{-j}$ yields the nodewise regression

$$X_j = \alpha_j + \mathbf{x}_{-j}^\top \boldsymbol{\beta}_j + \varepsilon_j, \tag{5}$$

where $\boldsymbol{\beta}_j = \boldsymbol{\Sigma}_{-j,-j}^{-1}\boldsymbol{\Sigma}_{-j,j}$, $\alpha_j = \mu_j - \boldsymbol{\beta}_j^\top \boldsymbol{\mu}_{-j}$, and the error term satisfies $\mathrm{E}(\varepsilon_j) = 0$ and $\tau_j := \mathrm{Var}(\varepsilon_j) = \Sigma_{jj} - \boldsymbol{\Sigma}_{j,-j}\boldsymbol{\Sigma}_{-j,-j}^{-1}\boldsymbol{\Sigma}_{-j,j}$ (the Schur complement). Through matrix blockwise inversion, the j-th column of $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ satisfies

$$\Omega_{jj} = 1/\tau_j, \qquad \boldsymbol{\Omega}_{-j,j} = -\boldsymbol{\beta}_j/\tau_j. \tag{6}$$

Equation (6) establishes a direct mapping from the regression parameters $(\boldsymbol{\beta}_j, \tau_j)$ to the entries of the precision matrix column $\boldsymbol{\Omega}_j$. This allows us to decompose the estimation of $\boldsymbol{\Omega}$ into p independent regression sub-problems. Leveraging this column-wise decomposability, we extend the framework to the transfer learning setting by applying nodewise regression to both target and source domains. Throughout, we assume each dataset has been centered so that $\boldsymbol{\mu}^{(k)} = \mathbf{0}$ for $k = 0, 1, \ldots, K$, and hence the nodewise regressions have no intercept. For each node $j \in [p]$,

we write

$$X_j^{(0)} = (\mathbf{x}_{-j}^{(0)})^\top \boldsymbol{\beta}_j^{(0)} + \varepsilon_j^{(0)}, \tag{7}$$

$$X_j^{(k)} = (\mathbf{x}_{-j}^{(k)})^\top \boldsymbol{\beta}_j^{(k)} + \varepsilon_j^{(k)}, \qquad k = 1, \ldots, K. \tag{8}$$

We model the distribution shift between the target and source domains via the relationship between their coefficients:

$$\boldsymbol{\beta}_j^{(k)} = \boldsymbol{\beta}_j^{(0)} + \boldsymbol{\delta}_j^{(k)}, \quad k = 1, \ldots, K, \tag{9}$$

or equivalently,

$$\boldsymbol{\delta}_j^{(k)} = \boldsymbol{\beta}_j^{(k)} - \boldsymbol{\beta}_j^{(0)} = \frac{\boldsymbol{\Omega}_{-j,j}^{(0)}}{\Omega_{jj}^{(0)}} - \frac{\boldsymbol{\Omega}_{-j,j}^{(k)}}{\Omega_{jj}^{(k)}}, \tag{10}$$

where $\boldsymbol{\delta}_j^{(k)}$ represents the source-target bias vector for node j. In the high-dimensional setting, we proceed under the assumption that the source and target domains share common structural features, meaning that $\boldsymbol{\delta}_j^{(k)}$ is to be sparse or small in magnitude.

To quantify source-target similarity at the precision-matrix level, define the discrepancy matrix $\boldsymbol{\Delta}^{(k)} = \boldsymbol{\Omega}^{(k)} - \boldsymbol{\Omega}^{(0)}$. Since our procedure operates column-by-column, we measure divergence using the matrix L_1-norm and impose the transferability condition

$$\max_{k \in \{1,\ldots,K\}} \|\boldsymbol{\Delta}^{(k)}\|_1 \le h. \tag{11}$$

This condition controls the cumulative discrepancy in each column of $\boldsymbol{\Omega}^{(k)}$ relative to $\boldsymbol{\Omega}^{(0)}$, and it aligns with the regression-based transfer learning literature where guarantees are typically stated in terms of coefficient-bias bounds, e.g., $\|\boldsymbol{\delta}_j^{(k)}\|_1 \le h_j$. Lemma 1 formalizes the connection between the matrix-level similarity condition and regression-level coefficient similarity, which enables us to transfer theoretical arguments from high-dimensional regression to precision matrix estimation.

Lemma 1. *Assume there exists a constant $M_0 > 0$ such that $M_0^{-1} \le \Omega_{jj}^{(k)} \le M_0$, $j \in \{1, \ldots, p\}$, $k \in \{0, 1, \ldots, K\}$. For each j, define the diagonal discrepancy*

$$h_{D,j} := \max_{1 \le k \le K} |\Omega_{jj}^{(k)} - \Omega_{jj}^{(0)}|,$$

and let $\boldsymbol{\Delta}^{(k)} := \boldsymbol{\Omega}^{(k)} - \boldsymbol{\Omega}^{(0)}$. Suppose that

$$\max_{1 \le k \le K} \|\boldsymbol{\Delta}^{(k)}\|_1 \le h, \qquad \text{and} \qquad \max_{1 \le j \le p} \left\{ 1 + \|\boldsymbol{\beta}_j^{(0)}\|_1 \frac{h_{D,j}}{h} \right\} \le C_0,$$

where $\beta_j^{(k)}$ is the jth nodewise regression coefficient vector under domain k and C_0 is a positive constant. Let

$$h_j := \max_{1 \le k \le K} \|\boldsymbol{\delta}_j^{(k)}\|_1, \qquad \boldsymbol{\delta}_j^{(k)} := \boldsymbol{\beta}_j^{(k)} - \boldsymbol{\beta}_j^{(0)}.$$

Then, for all j, $h_j \le Mh$ with $M := M_0 C_0$.

When $\Omega_{jj}^{(k)} = \Omega_{jj}^{(0)}$ for all k, then $h_{D,j} = 0$, and $C_0 = 1$. When $\|\boldsymbol{\beta}_j^{(0)}\|_1$ is bounded, then C_0 can be a finite constant since $h_{D,j}/h \le 1$. For example, when $\boldsymbol{\Omega}^{(0)} = (\omega_{ij}^{(0)})$ is a banded matrix

such that $\omega_{ij}^{(0)} = 0$ if $|i-j| \geq b$, then $\|\boldsymbol{\beta}_j^{(0)}\|_1$ is bounded if b is finite. For instance, when $\boldsymbol{\Sigma}_0$ is of autoregressive correlation structure with order 1, denoted by AR(1) for short, then $\boldsymbol{\Omega}^{(0)}$ is a banded matrix with $b = 2$. Certainly, $\boldsymbol{\Sigma}^{(0)}$ can be more general than AR(1) structure.

2.2 New estimation procedures and theoretical properties

For each node $j \in [p]$, we estimate the target regression coefficient $\boldsymbol{\beta}_j^{(0)} \in \mathbb{R}^{p-1}$ by solving a transfer learning problem

$$\widehat{\boldsymbol{b}}_j = \underset{\boldsymbol{b} \in \mathbb{R}^{(K+1)(p-1)}}{\arg\min} \left\{ \frac{1}{2N} \sum_{k=0}^{K} \left\| \mathbf{X}_j^{(k)} - \mathbf{X}_{-j}^{(k)} \boldsymbol{\beta}_j^{(k)} \right\|_2^2 + p_{\lambda_j}(\boldsymbol{b}) \right\}, \tag{12}$$

where $N = \sum_{k=0}^{K} n_k$, $\mathbf{X}_j^{(k)} \in \mathbb{R}^{n_k}$ and $\mathbf{X}_{-j}^{(k)} \in \mathbb{R}^{n_k \times (p-1)}$ are the response vector and design matrix, and

$$\boldsymbol{b} = [\boldsymbol{\delta}_j^{(1)}, \ldots, \boldsymbol{\delta}_j^{(K)}, \boldsymbol{\beta}_j^{(0)}]^\top, \qquad \boldsymbol{\beta}_j^{(k)} = \boldsymbol{\beta}_j^{(0)} + \boldsymbol{\delta}_j^{(k)}, \qquad k = 1, ..., K,$$

so that $\boldsymbol{b}$ jointly collects the target coefficient and the source-target bias vectors. We extract the estimator of $\boldsymbol{\beta}_j^{(0)}$ as the last $p-1$ entries of $\widehat{\boldsymbol{b}}_j$, denoted by $\widehat{\boldsymbol{\beta}}_j = \mathrm{tail}(\widehat{\boldsymbol{b}}_j, p-1)$.

A convex instance, termed Trans-Nodewise Lasso, is obtained by choosing an ℓ_1 penalty:

$$\widehat{\boldsymbol{b}}_j = \underset{\boldsymbol{b} \in \mathbb{R}^{(K+1)(p-1)}}{\arg\min} \left\{ \frac{1}{2N} \sum_{k=0}^{K} \left\| \mathbf{X}_j^{(k)} - \mathbf{X}_{-j}^{(k)} \boldsymbol{\beta}_j^{(k)} \right\|_2^2 + \lambda_{j0} \|\boldsymbol{\beta}_j^{(0)}\|_1 + \sum_{k=1}^{K} \lambda_{jk} \|\boldsymbol{\delta}_j^{(k)}\|_1 \right\}, \tag{13}$$

where λ_{j0} and $\{\lambda_{jk}\}_{k=1}^{K}$ allow different sparsity levels in the target coefficient and the bias vectors.

Alternatively, we may use a nonconvex SCAD penalty:

$$\widehat{\boldsymbol{b}}_j = \underset{\boldsymbol{b} \in \mathbb{R}^{(K+1)(p-1)}}{\arg\min} \left\{ \frac{1}{2N} \sum_{k=0}^{K} \left\| \mathbf{X}_j^{(k)} - \mathbf{X}_{-j}^{(k)} \boldsymbol{\beta}_j^{(k)} \right\|_2^2 + p_{\lambda_{j0}}^{\mathrm{SCAD}}(\boldsymbol{\beta}_j^{(0)}) + \sum_{k=1}^{K} p_{\lambda_{jk}}^{\mathrm{SCAD}}(\boldsymbol{\delta}_j^{(k)}) \right\}, \tag{14}$$

where $p_{\lambda}^{\mathrm{SCAD}}(\cdot)$ denotes the SCAD penalty with tuning parameter λ.

Given Equation (6), estimating $\boldsymbol{\Omega}^{(0)}$ further requires an estimator of the residual variance $\tau_j^{(0)}$. We consider two MSE-based estimators. The first is a global estimator that pools all domains:

$$\widehat{\tau}_j = \frac{\left\| \mathbf{y} - \mathbf{X} \widehat{\boldsymbol{b}}_j \right\|_2^2}{N - \widehat{s}_{bj}}, \tag{15}$$

where $(\mathbf{y}, \mathbf{X})$ are defined in (19), $N = \sum_{k=0}^{K} n_k$, and $\widehat{s}_{bj}$ is the number of nonzero entries of $\widehat{\boldsymbol{b}}_j$. This pooled estimator is appropriate when the conditional noise variances are comparable across domains.

The second is a local estimator based only on the target sample:

$$\widehat{\tau}_j = \frac{\left\| \mathbf{X}_j^{(0)} - \mathbf{X}_{-j}^{(0)} \widehat{\boldsymbol{\beta}}_j \right\|_2^2}{n_0 - \widehat{s}_j}, \tag{16}$$

where $\widehat{s}_j$ is the number of nonzero entries of $\widehat{\boldsymbol{\beta}}_j$.

Overall, we obtain $\widehat{\boldsymbol{\beta}}_j$ and $\widehat{\tau}_j$ for each j to form the j-th column $\widehat{\boldsymbol{\Omega}}_j$ where

$$\begin{cases} \widehat{\Omega}_{jj} = 1/\widehat{\tau}_j \\ \widehat{\boldsymbol{\Omega}}_{-j,j} = -\widehat{\boldsymbol{\beta}}_j/\widehat{\tau}_j, \end{cases} \tag{17}$$

for $j = 1, \ldots, p$. And an estimate of $\boldsymbol{\Omega}^{(0)}$ can be constructed as $\widehat{\boldsymbol{\Omega}} = (\widehat{\boldsymbol{\Omega}}_1, \ldots, \widehat{\boldsymbol{\Omega}}_p)$. We refer to $\widehat{\boldsymbol{\Omega}}$ as the *Trans-Nodewise estimator* of $\boldsymbol{\Omega}^{(0)}$. A specific instance is determined by the choice of penalty function used in either (13) or (14) and the choice of variance estimator in either (15) or (16).

To study the sampling properties of the resulting estimator, we impose the following assumptions to facilitate our theoretical analysis.

Assumption 1. *Target data $\mathbf{x}_i^{(0)} \in \mathbb{R}^p$ are i.i.d. sub-Gaussian distributed with mean zero and covariance $\boldsymbol{\Sigma}^{(0)}$ for $i = 1, \ldots n_0$. Source data $\mathbf{x}_i^{(k)}$ are i.i.d. sub-Gaussian distributed with mean zero and covariance $\boldsymbol{\Sigma}^{(k)}$ for $i = 1, \ldots, n_k$, for $1 \le k \le K$.*

Assumption 2. *There exists a constant M_0 such that for $0 \le k \le K$,*

$$\frac{1}{M_0} \le \Lambda_{\min}\left(\boldsymbol{\Omega}^{(k)}\right) \le \Lambda_{\max}\left(\boldsymbol{\Omega}^{(k)}\right) \le M_0.$$

The constant M_0 may either be fixed or allowed to vary with n and p, depending on the specific setting.

Assumption 3. *The sample covariance matrix $\widehat{\boldsymbol{\Sigma}}^{(k)} = \frac{1}{n_k}(\boldsymbol{X}^{(k)})^\top \boldsymbol{X}^{(k)}$, for $k = 0, 1, \cdots, K$, satisfies the restricted strong convexity (RSC) condition*

$$\boldsymbol{\Delta}^\top \widehat{\boldsymbol{\Sigma}}^{(k)} \boldsymbol{\Delta} \ge \nu_k \|\boldsymbol{\Delta}\|_2^2 - \tau_k \sqrt{\frac{\log p}{n_k}} \|\boldsymbol{\Delta}\|_1,$$

for any $\boldsymbol{\Delta} \in \mathbb{R}^p$ and $\|\boldsymbol{\Delta}\|_1 \ge 1$, where $\nu_k > 0$ and $\tau_k > 0$.

Assumption 4. *The penalty p_λ used in (12) is amenable with parameters $L > 0$ and $\zeta \ge 0$:* (i) *$p_\lambda(0) = 0$ and p_λ is even;* (ii) *p_λ is nondecreasing on $[0, \infty)$ and $t \mapsto p_\lambda(t)/t$ is nonincreasing on $(0, \infty)$;* (iii) *p_λ is differentiable for $t \ne 0$ with $\lim_{t \to 0+} p'_\lambda(t) = \lambda L$; and* (iv) *$t \mapsto p_\lambda(t) + \frac{\zeta}{2} t^2$ is convex. The amenability parameters L and ζ, as well as any shape parameter associated with the penalty, are fixed constants independent of n, p, and N.*

Assumption 5. *Let $\nu_{\min} = \min_{0 \le k \le K} \nu_k n_k / N$, and let $\kappa_K = 2/(K + 2 + \sqrt{K^2 + 4K})$. The number of sources K satisfies $a := 2\kappa_K \nu_{\min} - \frac{3\zeta}{2} > 0$ with ζ as in Assumption 4. We assume that the number of sources is uniformly bounded, $K \le K_0$, and that there exists a constant $a_0 > 0$, both independent of n, p, and N, such that $a \ge a_0$.*

Assumption 1 implies sub-Gaussian designs (and hence sub-Gaussian nodewise errors). Assumption 2 ensures uniform conditioning of the precision matrices and, in particular, bounds the nodewise error variance since $\tau_j^{(k)} = 1/\Omega_{jj}^{(k)}$ and $\Lambda_{\min}(\boldsymbol{\Omega}^{(k)}) \le \Omega_{jj}^{(k)} \le \Lambda_{\max}(\boldsymbol{\Omega}^{(k)})$. In Assumption 3, we assume that the sample covariance matrices satisfy the restricted strong convexity

(RSC) condition. This assumption is crucial for the derivation of the non-asymptotic error bounds for the regression coefficients and can often be considered a property of sub-Gaussian designs. This formulation follows condition C3 of Liu (2024). The amenability conditions in Assumption 4 are those of Loh and Wainwright (2015) and hold for the Lasso, with $L = 1$ and $\zeta = 0$, as well as the SCAD penalty. Writing $P_\lambda(\boldsymbol{b}) = \sum_l p_\lambda(b_l)$ for the separable penalty in (12), Assumption 4 implies, for every $\boldsymbol{w}$,

$$P_\lambda(\boldsymbol{w}) \le \lambda L\|\boldsymbol{w}\|_1 \qquad \text{and} \qquad \lambda L\|\boldsymbol{w}\|_1 \le P_\lambda(\boldsymbol{w}) + \tfrac{\zeta}{2}\|\boldsymbol{w}\|_2^2, \tag{18}$$

by Lemma 4 of Loh and Wainwright (2015). For a nonconvex penalty we take $\widehat{\boldsymbol{b}}_j$ to be any stationary point of (12), in the sense of Loh and Wainwright (2015), and for the Lasso every stationary point is a global minimizer. Assumption 5 imposes an upper bound on the number of sources K to facilitate the technical proofs. This condition may be relaxed when the Lasso penalty or weight Lasso penalty is used since $\zeta = 0$ for convex penalties. The constant κ_K is used for the restricted strong convexity bound for $\widehat{\boldsymbol{\Sigma}} = \mathbf{X}^\top\mathbf{X}/N$ derived in Lemma 3.

We consider the parameter space

$$\mathbb{G}(s,h) = \Big\{\Big(\boldsymbol{\Omega}^{(0)}, \boldsymbol{\Omega}^{(1)}, \ldots, \boldsymbol{\Omega}^{(K)}\Big) : \max_{1\le j\le p}\Big\|\boldsymbol{\Omega}_j^{(0)}\Big\|_0 \le s, \max_{1\le k\le K}\max_{1\le j\le p}\Big\|\boldsymbol{\Delta}_j^{(k)}\Big\|_0 \le Cs, \max_{1\le k\le K}\Big\|\boldsymbol{\Delta}^{(k)}\Big\|_1 \le h\Big\},$$

where $\boldsymbol{\Delta}^{(k)} = \boldsymbol{\Omega}^{(k)} - \boldsymbol{\Omega}^{(0)}$ and $C > 0$ is a constant. In $\mathbb{G}(s,h)$, the target precision matrix is s-sparse, the disparity matrix is Cs-sparse, and the similarity is controlled by h.

We first state the convergence rate of the Trans-Nodewise estimator using the global variance estimator $\widehat{\tau}_j$ defined in (15). The corresponding result for the local variance estimator $\widehat{\tau}_j$ in (16) is summarized in the remark following Theorem 1 and stated formally as Theorem 2 in Appendix 4.

Theorem 1. *Under Assumptions 1–5, uniformly over $\mathbb{G}(s,h)$, let $\widehat{\boldsymbol{\Omega}}$ be the Trans-Nodewise estimator computed with $\lambda_{j0} = \lambda_{j1} = \cdots = \lambda_{jK} = \lambda \asymp \sqrt{\log p/N}$ and the global variance estimator. Further assume that the conditions of Lemma 1 hold. Suppose that $\lambda L/2 \ge c_\varepsilon\sqrt{\log p/N} + r_N$ for some constant c_ε, where $r_N = \tau_{\max}(\sqrt{n_{\max}\log p} + \sqrt{n_0 \log p})/N$, $\tau_{\max} = (K+1)\max_{0\le k\le K}\tau_k$, and $n_{\max} = \max_{1\le k\le K} n_k$. Assume the global pooled variance estimator satisfies $\max_{1\le j\le p}|\widehat{\tau}_j - \tau_j^{(0)}| \le c\sqrt{s_b \log p_b/N}$ with $s_b = (1+K+CK)s$, $p_b = (K+1)(p-1)$, and $cM_0\sqrt{s_b\log p_b/N} \le 1/2$. Then with probability at least $1 - cp^{-1}$,*

$$\|\widehat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^{(0)}\|_1 \lesssim M_0\Big(s^{1/2} + C_1\|\boldsymbol{\Omega}^{(0)}\|_1 + C_2M_0\Big)\sqrt{\frac{s\log p}{N}} + M_0M^{1/2}(sh)^{1/2}\Big(\frac{\log p}{N}\Big)^{1/4},$$

where $N = \sum_{k=0}^K n_k$, $h = \max_{1\le k\le K}\|\boldsymbol{\Omega}^{(k)} - \boldsymbol{\Omega}^{(0)}\|_1$, C_1 and C_2 are positive constants, and $M = M_0C_0$. Moreover, $\|\widehat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^{(0)}\|_2 \le \|\widehat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^{(0)}\|_1$ satisfies the same upper bound.

Detailed proof of Theorem 1 is provided in Appendix 4. This theorem establishes a non-asymptotic upper bound for the Trans-Nodewise estimator in the matrix L_1 norm, and hence also in the spectral norm since $\|\widehat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^{(0)}\|_2 \le \|\widehat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^{(0)}\|_1$. The bound contains two components.

The first term,

$$M_0\Big(s^{1/2} + C_1\|\mathbf{\Omega}^{(0)}\|_1 + C_2 M_0\Big)\sqrt{\frac{s\log p}{N}},$$

reflects the estimation error that would arise under effective pooling across the transferable source and target samples. The second term,

$$M_0 M^{1/2}(sh)^{1/2}\Big(\frac{\log p}{N}\Big)^{1/4},$$

captures the additional cost of source-target discrepancy through the similarity level $h = \max_{1\le k\le K}\|\mathbf{\Omega}^{(k)} - \mathbf{\Omega}^{(0)}\|_1$. Thus, the benefit of transfer learning depends on balancing the gain from the larger effective sample size N against the bias introduced by heterogeneous source precision matrices.

When M_0 and M are treated as constants, the bound simplifies to

$$\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_1 \lesssim \Big(s^{1/2} + C_1\|\mathbf{\Omega}^{(0)}\|_1\Big)\sqrt{\frac{s\log p}{N}} + (sh)^{1/2}\Big(\frac{\log p}{N}\Big)^{1/4}.$$

We can see more clearly that the leading term scales with the total sample size N, while the second term vanishes as the sources become more similar to the target. In contrast, the classical single-task nodewise Lasso based only on the target sample satisfies (van de Geer et al., 2014)

$$\max_{1\le j\le p}\|\widehat{\mathbf{\Omega}}_j - \mathbf{\Omega}_j^{(0)}\|_1 = O\left(s\sqrt{\frac{\log p}{n_0}}\right).$$

Since $\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_1 = \max_{1\le j\le p}\|\widehat{\mathbf{\Omega}}_j - \mathbf{\Omega}_j^{(0)}\|_1$, Theorem 1 shows that the proposed estimator can improve upon the single-task rate when the auxiliary sources are sufficiently similar to the target and the total sample size N is substantially larger than n_0. In particular, when $h \to 0$ and $\|\mathbf{\Omega}^{(0)}\|_1, M_0 \lesssim s^{1/2}$, the rate reduces to the pooled-data order

$$\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_1 = O\left(s\sqrt{\frac{\log p}{N}}\right).$$

This result is stated for the estimator using the global variance estimator. When the diagonal entries are comparable across domains, this estimator benefits from the larger pooled sample size N by combining residuals from all $K+1$ domains. When the diagonal entries are not comparable across domains, one may instead use the local variance estimator in (16). An analogous error bound is established in Theorem 2 in Appendix 4. Relative to Theorem 1, the main difference is that the diagonal-estimation component is then governed by the target sample size n_0 rather than the pooled sample size N, since the local variance estimator uses only target-domain residuals.

2.3 Practical implementation issues

We implemented the proposed method in R using the package `glmnet`. For each node $j \in [p]$, the column-wise optimization can be written as a single penalized least-squares problem by stacking

the source and target regressions. Specifically, define

$$\mathbf{y} = \begin{bmatrix} \mathbf{X}_j^{(1)} \\ \vdots \\ \mathbf{X}_j^{(K)} \\ \mathbf{X}_j^{(0)} \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} \mathbf{X}_{-j}^{(1)} & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{X}_{-j}^{(1)} \\ \mathbf{0} & \mathbf{X}_{-j}^{(2)} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{X}_{-j}^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{X}_{-j}^{(K)} & \mathbf{X}_{-j}^{(K)} \\ \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} & \mathbf{X}_{-j}^{(0)} \end{bmatrix} \in \mathbb{R}^{N\times(K+1)(p-1)}, \tag{19}$$

where $\mathbf{X}_{-j}^{(k)} \in \mathbb{R}^{n_k\times(p-1)}$ and $N = \sum_{k=0}^{K} n_k$. Then (13) is equivalent to a Lasso-type regression

$$\mathbf{y} = \mathbf{X}\boldsymbol{b} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{b} = [\boldsymbol{\delta}_j^{(1)}, \ldots, \boldsymbol{\delta}_j^{(K)}, \boldsymbol{\beta}_j^{(0)}]^\top \in \mathbb{R}^{(K+1)(p-1)}. \tag{20}$$

In practice, we select the tuning parameters by High-Dimensional Bayesian Information Criterion (HBIC) separately for each j (Wang et al., 2013). When allowing different regularization levels across blocks (i.e., λ_{j0} and $\{\lambda_{jk}\}_{k=1}^{K}$), this can be implemented in `glmnet` via coefficient-specific penalty factors (or equivalently by rescaling the corresponding columns of $\mathbf{X}$). The p nodewise regressions are independently parallel and can be distributed across cores using `doParallel` and `foreach`.

Since the estimator is constructed column-by-column, we typically enforce symmetry by averaging,

$$\widehat{\boldsymbol{\Omega}}_{\mathrm{s}} = \tfrac{1}{2}(\widehat{\boldsymbol{\Omega}} + \widehat{\boldsymbol{\Omega}}^\top).$$

If a positive semidefinite (PSD) estimate is desired, we further apply eigenvalue truncation: if $\widehat{\boldsymbol{\Omega}}_{\mathrm{s}} = \mathbf{V}\,\mathrm{diag}(\lambda_1, \ldots, \lambda_p)\,\mathbf{V}^\top$, then

$$\widehat{\boldsymbol{\Omega}}_{\mathrm{s}}^{+} = \mathbf{V}\,\mathrm{diag}(\lambda_1^+, \ldots, \lambda_p^+)\,\mathbf{V}^\top, \qquad \lambda_i^+ = \max\{\lambda_i, 0\}.$$

3 Numerical Studies

In Section 3.1, we present our simulation studies to assess the finite sample performance of the proposed procedure via a numerical comparison with existing ones. In Section 3.2, we illustrate the proposed methodology by an empirical analysis of a real-world example.

3.1 Simulation

We conducted simulation studies to evaluate the finite-sample performance of the proposed transfer learning estimators for high-dimensional precision matrix estimation. The simulations are designed to assess whether incorporating auxiliary source data improves estimation accuracy and support recovery, and how such improvement depends on the amount of available source information under controlled distribution shift. In particular, we focus on the effect of the number of source domains while fixing the dimension, target sample size, source sample size, and source-target similarity level. This design allows us to examine whether additional transferable source domains lead to systematic gains and to compare the proposed transfer learning estimators with single-task estimators applied either to the target data alone or to naively pooled data.

We compare six precision matrix estimators based on nodewise regression:

- *Nodewise Lasso*, a single-task estimator applied only to the target data;
- *Pooled Nodewise Lasso*, a single-task estimator applied to the naively pooled target and source data;
- *Trans-Nodewise Lasso*, our proposed transfer learning estimator with global diagonal estimation;
- *Nodewise SCAD*, a single-task estimator applied only to the target data;
- *Pooled Nodewise SCAD*, a single-task estimator applied to the naively pooled target and source data;
- *Trans-Nodewise SCAD*, our proposed transfer learning estimator with global diagonal estimation.

Target data-generating mechanism: The target data are generated from a zero-mean Gaussian distribution with a covariance matrix $\mathbf{\Sigma}^{(0)} = (\sigma_{ij}^{(0)}) \in \mathbb{R}^{p\times p}$ following an AR(1) structure,

$$\sigma_{ij}^{(0)} = \rho^{|i-j|}, \qquad 1 \le i, j \le p,$$

with $\rho = 0.6$. The corresponding target precision matrix is given by $\mathbf{\Omega}^{(0)} = (\mathbf{\Sigma}^{(0)})^{-1}$. Under this construction, $\mathbf{\Omega}^{(0)}$ is tri-diagonal, so the target precision matrix is sparse and each variable is conditionally dependent only on its immediate neighbors. This setting is chosen to better reflect the sparse precision-matrix structure under which nodewise regression methods are expected to perform well. We then generate the target sample $\mathbf{x}_1^{(0)}, \ldots, \mathbf{x}_{n_0}^{(0)} \overset{\text{iid}}{\sim} N(\mathbf{0}, \mathbf{\Sigma}^{(0)})$.

Source data-generating mechanism: For each source domain $k = 1, \ldots, K$, we generate the source precision matrix by

$$\mathbf{\Omega}^{(k)} = \mathbf{\Omega}^{(0)} + \mathbf{a}^{(k)}\big(\mathbf{a}^{(k)}\big)^{\top},$$

where $\mathbf{a}^{(k)} \in \mathbb{R}^p$ is a sparse perturbation vector. Let $S_k \subset \{1, \ldots, p\}$ denote its support, with $|S_k| = m$, where $m = 0.1p$. For each source domain, the support S_k is selected uniformly at random by sampling m distinct indices from $\{1, \ldots, p\}$ without replacement. We consider two ways to generate $\mathbf{a}^{(k)}$ on this random support:

- For $i \in S_k$, generate

$$a_i^{(k)} \sim U\left(-\sqrt{\frac{2h(m+2)}{m(m+1)}},\ \sqrt{\frac{2h(m+2)}{m(m+1)}}\right),$$

  independently, and set $a_i^{(k)} = 0$ for $i \notin S_k$.

- For $i \in S_k$, set

$$a_i^{(k)} = \xi_i^{(k)}\sqrt{\frac{h}{m}}, \qquad \xi_i^{(k)} \in \{-1, 1\},$$

  where $\xi_i^{(k)}$ are independent Rademacher variables, and set $a_i^{(k)} = 0$ for $i \notin S_k$.

Thus, $\mathbf{\Delta}^{(k)} := \mathbf{\Omega}^{(k)} - \mathbf{\Omega}^{(0)} = \mathbf{a}^{(k)}\big(\mathbf{a}^{(k)}\big)^\top$ is a rank-one perturbation. Because $\mathbf{\Delta}^{(k)}$ is supported on $S_k \times S_k$, the perturbation can create new off-diagonal nonzero entries among variables in S_k. Hence, the source precision matrices generally do not need to share the same sparsity pattern as the target precision matrix. Moreover, $\|\mathbf{\Delta}^{(k)}\|_1 = \left\|\mathbf{a}^{(k)}\big(\mathbf{a}^{(k)}\big)^\top\right\|_1 = \|\mathbf{a}^{(k)}\|_1 \, \|\mathbf{a}^{(k)}\|_\infty$. For the first construction (Uniform), the perturbation magnitude is calibrated so that $\mathbb{E}\|\mathbf{\Delta}^{(k)}\|_1 = h$. For the second construction (Rademacher), we have $\|\mathbf{\Delta}^{(k)}\|_1 = h$ exactly. Therefore, the parameter h directly controls the source-target similarity in the sense of the matrix L_1 norm used in our theory: smaller values of h correspond to more informative sources, whereas larger values of h correspond to greater distribution shift.

This perturbation scheme also guarantees that each source precision matrix is valid. Since $\mathbf{\Omega}^{(0)}$ is positive definite and $\mathbf{a}^{(k)}(\mathbf{a}^{(k)})^\top$ is positive semidefinite, we have

$$\mathbf{\Omega}^{(k)} \succ 0 \qquad \text{for all } k = 1, \ldots, K.$$

Hence, the source covariance matrix $\mathbf{\Sigma}^{(k)} = \big(\mathbf{\Omega}^{(k)}\big)^{-1}$ is well defined and positive definite. By the Sherman–Morrison formula,

$$\mathbf{\Sigma}^{(k)} = \big(\mathbf{\Omega}^{(0)} + \mathbf{a}^{(k)}(\mathbf{a}^{(k)})^\top\big)^{-1} = \mathbf{\Sigma}^{(0)} - \frac{\mathbf{\Sigma}^{(0)}\mathbf{a}^{(k)}(\mathbf{a}^{(k)})^\top\mathbf{\Sigma}^{(0)}}{1 + (\mathbf{a}^{(k)})^\top\mathbf{\Sigma}^{(0)}\mathbf{a}^{(k)}}.$$

We then, for each source domain $k = 1, \ldots, K$, generate

$$\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_{n_s}^{(k)} \overset{\text{i.i.d.}}{\sim} N_p(\mathbf{0}, \mathbf{\Sigma}^{(k)}).$$

Experimental design. We first fix $p = 50, n_0 = 50, n_s = 400, \rho = 0.6$. We generate up to $K_{\max} = 5$ source domains per replication and evaluate $K \in \{1, 2, 3, 4, 5\}$. Let $m = \lfloor 0.1p \rfloor = 5$. We consider

$$h \in \{0.125m,\ 0.25m,\ 0.5m,\ m,\ 2m,\ 4m\} = \{0.625,\ 1.25,\ 2.5,\ 5,\ 10,\ 20\}.$$

For each perturbation mechanism and each value of h, we run 200 Monte Carlo replications. Within each replication, we first generate five source domains independently and then form the K-source setting by taking the first K domains. Thus, the comparisons across different values of K are made on a common underlying realization, which reduces Monte Carlo noise when studying the marginal benefit of adding more sources.

Parameter tuning. All nodewise regression tuning parameters are selected by HBIC, separately for each column-wise regression. For SCAD, we compute a regularization path using `ncvreg` with `penalty = "SCAD"` and otherwise default path settings. HBIC is then evaluated along the locally convex portion of this path, separately for each column-wise regression, and the final estimator is taken from the HBIC-minimizing path solution without a separate single-λ refit. For transfer learning estimators, the HBIC is evaluated using the target residuals and degrees of freedom.

Evaluation. All estimators are symmetrized by averaging the matrix with its transpose. Following our preliminary comparison of diagonal estimation strategies, we use the global diagonal version as the default transfer learning estimator and report the oracle diagonal version as a benchmark. We evaluate each estimator using support recovery and estimation error. For support recovery, we report false discovery rate (FDR) and true positive rate (TPR), where FDR is the proportion of estimated nonzero off-diagonal entries that are false positives, and TPR is the proportion of truly nonzero off-diagonal entries in $\mathbf{\Omega}^{(0)}$ that are correctly identified. Here, FDR is used purely as an edge-selection accuracy metric relative to the known support of the true precision matrix in simulation, rather than as a multiple-testing error rate associated with hypothesis testing. Throughout this evaluation, diagonal entries are excluded from these metrics, and each undirected edge is counted once using the upper triangular part of the matrix. For estimation accuracy, we report the Frobenius norm error $\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_F$ and the spectral norm error $\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_2$.

Simulation results. We report the results under the uniform perturbation mechanism here and defer the corresponding results under the Rademacher perturbation mechanism to the appendix to save space, since the qualitative conclusions are the same. For estimation accuracy, we also include two additional benchmark methods: Trans-Nodewise Lasso/SCAD (oracle). These oracle variants use the true diagonal entries of the target precision matrix and are included only as upper benchmarks to separate the effect of diagonal estimation from that of the off-diagonal transfer step. Overall, the simulations convey three main messages. First, the proposed transfer learning estimators improve both support recovery and estimation accuracy relative to target-only estimators by borrowing strength from transferable source domains. Second, increasing the number of source domains (K) generally reduces the estimation error of the proposed methods, with the largest gains occurring when the first few sources are added. Third, compared with naive pooling, the proposed methods better account for source-target heterogeneity and therefore achieve more favorable edge recovery while remaining similarly competitive in matrix estimation. In what follows, we discuss these patterns in more detail for support recovery and estimation accuracy.

Support recovery: FDR and TPR. Under the uniform perturbation mechanism, the main distinction among methods lies in false discovery control (FDR) rather than sensitivity (TPR), since the TPR is high across all estimators. As shown in Figures 1 and 2, Trans-Nodewise Lasso and Trans-Nodewise SCAD achieve TPR essentially equal to one while maintaining nearly zero FDR, especially once at least two or three source domains are available. When $K = 1$, Trans-Nodewise Lasso has slightly higher TPR but also slightly higher FDR than Trans-Nodewise SCAD. By contrast, the target-only estimators, Nodewise Lasso and Nodewise SCAD, are clearly dominated in support recovery: compared with the transfer learning estimators, they have substantially higher FDR and lower TPR. The pooled estimators, Pooled Nodewise Lasso and Pooled Nodewise SCAD, attain near-one TPR across all values of K and h, similar to the transfer learning estimators, and their FDR improves as additional source domains become available. However, their FDR remains higher than that of the transfer learning estimators, with the gap being especially substantial for Pooled Nodewise Lasso. This reflects the cost

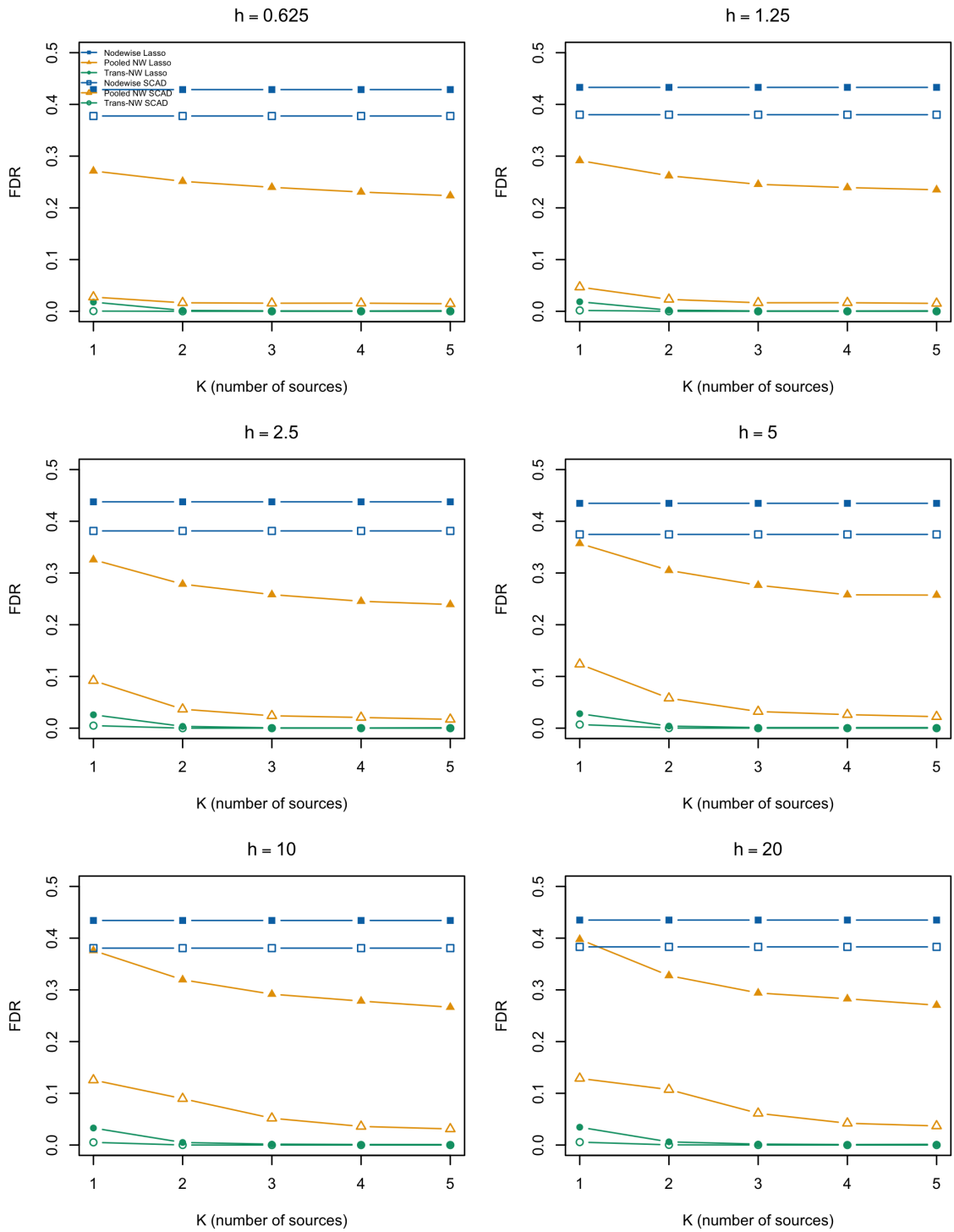

Figure 1: False discovery rate (FDR) for support recovery of the target precision matrix (mean over 200 Monte Carlo replications) under the simulation setting with dimension $p = 50$, target sample size $n_0 = 50$, source sample size $n_s = 400$, AR(1) base covariance with $\rho = 0.6$, and uniform perturbation. Panels correspond to the similarity parameter $h \in \{0.625, 1.25, 2.5, 5, 10, 20\}$ (larger h indicates greater source-target shift); within each panel, FDR is shown as a function of the number of source domains $K \in \{1, 2, 3, 4, 5\}$.

of naively aggregating heterogeneous source domains. Finally, the gains from adding source domains are strongest when moving from $K = 1$ to $K = 2$ or 3, after which support recovery performance largely stabilizes.

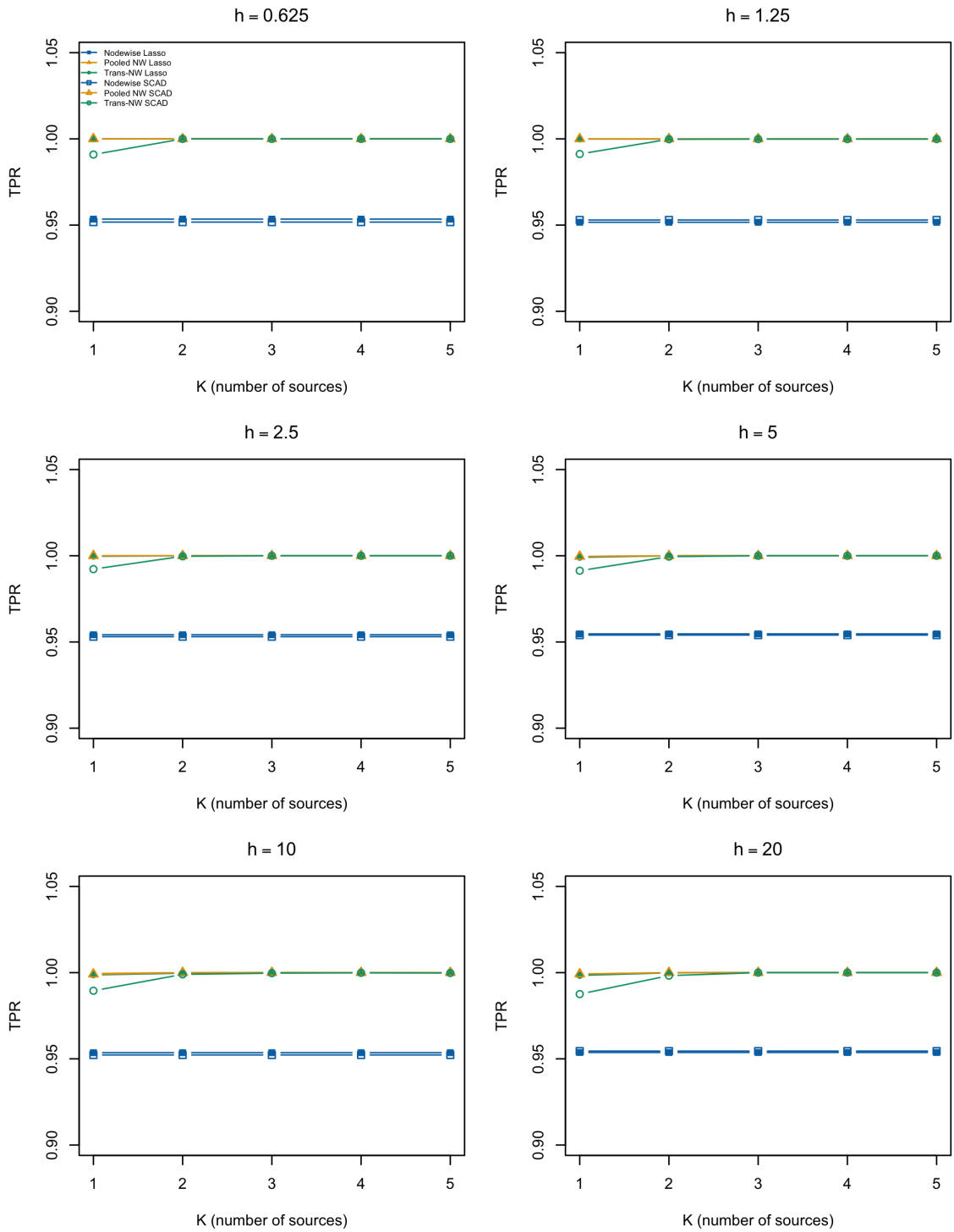

Figure 2: True positive rate (TPR) for support recovery of the target precision matrix (mean over 200 Monte Carlo replications) under the simulation setting with dimension $p = 50$, target sample size $n_0 = 50$, source sample size $n_s = 400$, AR(1) base covariance with $\rho = 0.6$, and uniform perturbation. Panels correspond to the similarity parameter $h \in \{0.625, 1.25, 2.5, 5, 10, 20\}$ (larger h indicates greater source-target shift); within each panel, TPR is shown as a function of the number of source domains $K \in \{1, 2, 3, 4, 5\}$.

Estimation accuracy: spectral and Frobenius norms. The estimation results tell a complementary story to the support recovery results. As shown in Figures 3 and 4, the target-only estimators are invariant to K, whereas both the pooled and transfer learning estimators gener-

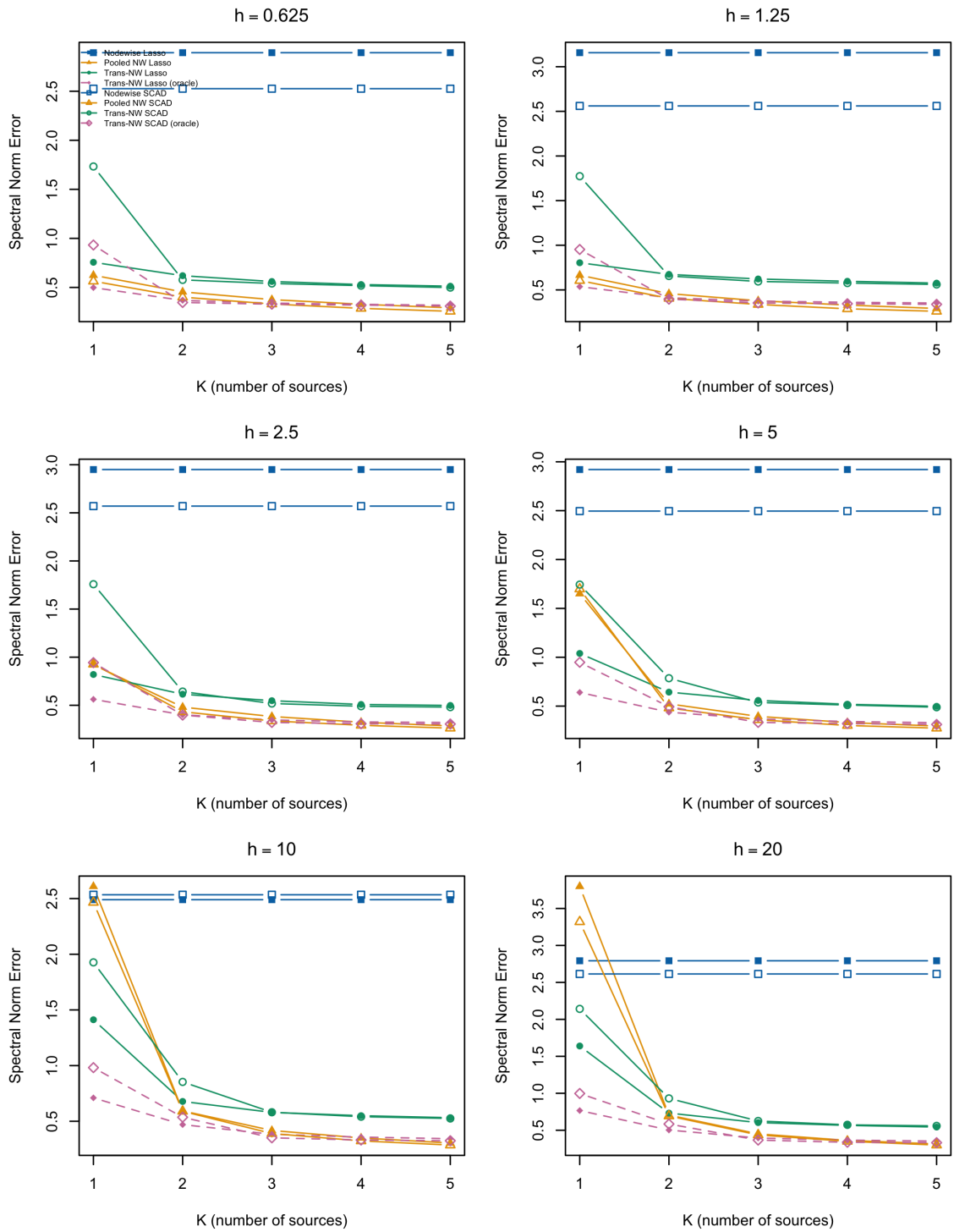

Figure 3: Spectral-norm estimation error for the target precision matrix (mean over 200 Monte Carlo replications) under the simulation setting with dimension $p = 50$, target sample size $n_0 = 50$, source sample size $n_s = 400$, AR(1) base covariance with $\rho = 0.6$, and uniform perturbation. Panels correspond to the similarity parameter $h \in \{0.625, 1.25, 2.5, 5, 10, 20\}$, where larger h indicates greater source-target shift. Within each panel, error is shown as a function of the number of source domains $K \in \{1, 2, 3, 4, 5\}$. The curves compare target-only, pooled, and transfer learning estimators for both Lasso and SCAD, including oracle-diagonal variants for the transfer learning estimators.

ally improve as more source domains are added. The largest gains typically occur when moving from $K = 1$ to $K = 2$ or 3, after which the marginal improvement becomes more modest. Under

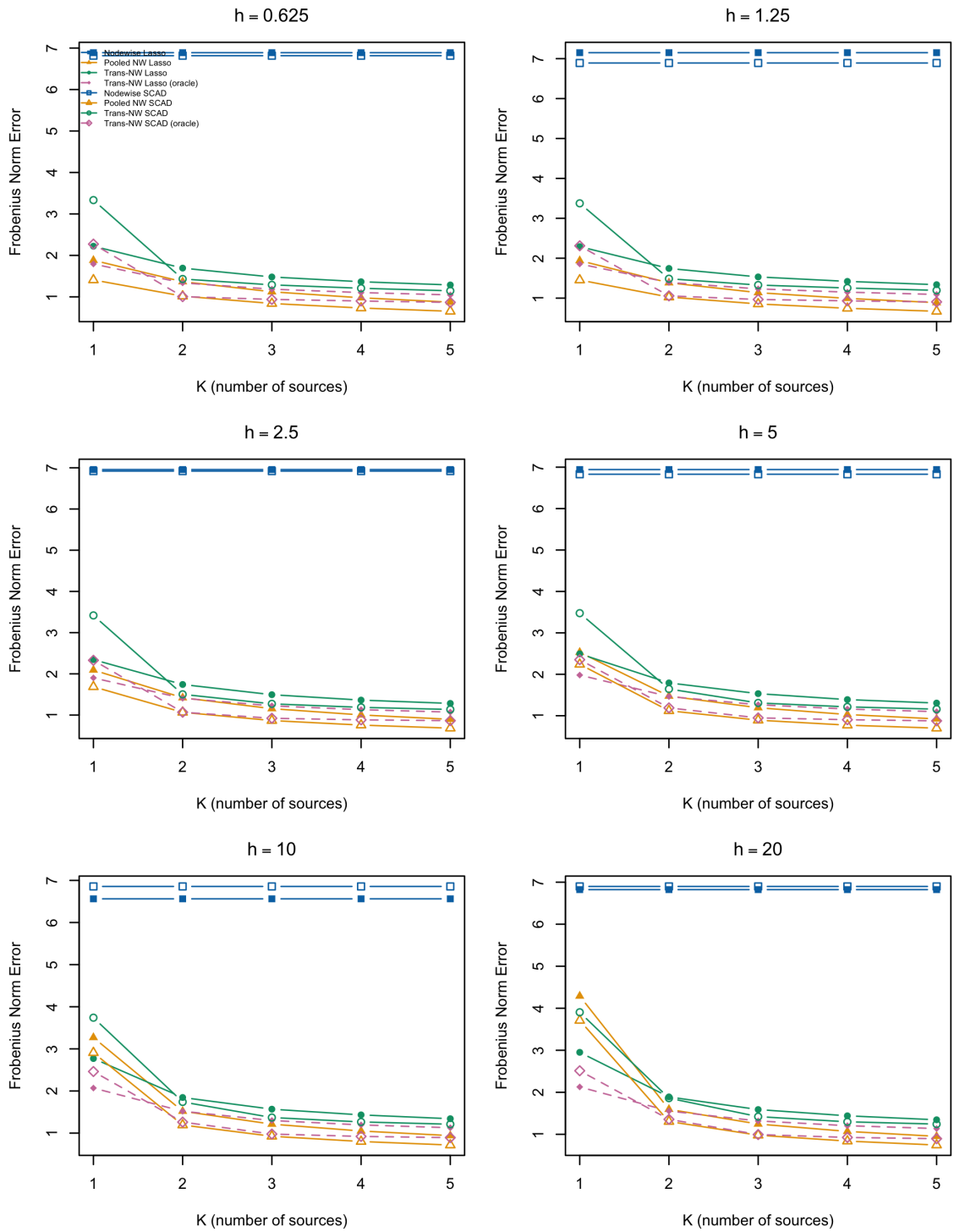

Figure 4: Frobenius-norm estimation error for the target precision matrix (mean over 200 Monte Carlo replications) under the simulation setting with dimension $p = 50$, target sample size $n_0 = 50$, source sample size $n_s = 400$, AR(1) base covariance with $\rho = 0.6$, and uniform perturbation. Panels correspond to the similarity parameter $h \in \{0.625, 1.25, 2.5, 5, 10, 20\}$, where larger h indicates greater source-target shift. Within each panel, error is shown as a function of the number of source domains $K \in \{1, 2, 3, 4, 5\}$. The curves compare target-only, pooled, and transfer learning estimators for both Lasso and SCAD, including oracle-diagonal variants for the transfer learning estimators.

both spectral and Frobenius norms, the transfer learning estimators substantially reduce estimation error relative to their target-only counterparts. Among the non-oracle methods, pooled

Nodewise SCAD often achieves the smallest absolute estimation error, especially once multiple source domains are available, although its performance can deteriorate when $K = 1$ and h is large. In contrast, the global-diagonal transfer learning estimators remain clearly competitive. Comparing the global-diagonal and oracle-diagonal transfer learning estimators shows that improved diagonal estimation can further reduce both spectral and Frobenius norm errors, in some cases bringing the transfer learning estimators close to the best non-oracle performance. This suggests that a substantial part of the remaining gap is attributable to diagonal estimation rather than to the off-diagonal transfer step.

Table 1: Average runtime (seconds) for the estimators under $p = 50$ with uniform perturbations.

Lasso				
K	Nodewise	Pooled Nodewise	Trans-Nodewise	Oracle Trans-Nodewise
1	2.30	2.33	2.39	2.38
2	2.34	2.34	2.44	2.44
3	2.35	2.38	2.52	2.52
4	2.38	2.41	2.60	2.60
5	2.36	2.40	2.67	2.66
SCAD				
K	Nodewise	Pooled Nodewise	Trans-Nodewise	Oracle Trans-Nodewise
1	0.59	0.56	2.78	2.79
2	0.59	0.59	3.78	3.80
3	0.61	0.63	5.56	5.56
4	0.62	0.68	8.14	8.14
5	0.60	0.69	11.96	11.99

Computing time. We report the computing time of these experiments on the Pennsylvania State University ROAR Collab cluster (ICDS). Each job used 50 CPU cores in parallel on a compute node with Intel Gold 6430 processors. Table 1 shows that the Lasso-based estimators have very similar computational cost: target-only and pooled estimators remain around 2.3–2.4 seconds across all values of K, while Trans-Nodewise Lasso increases only slightly, from 2.39 seconds at $K = 1$ to 2.67 seconds at $K = 5$. By contrast, the SCAD-based transfer learning estimators are substantially more expensive in terms of computational cost, and their runtime increases markedly with the number of source domains. In particular, Trans-Nodewise SCAD rises from 2.78 seconds at $K = 1$ to 11.96 seconds at $K = 5$, whereas Nodewise SCAD and pooled Nodewise SCAD remain below 0.7 seconds throughout. These results indicate that transfer learning under the Lasso penalty incurs only modest additional computational cost, whereas the SCAD-based transfer learning estimators become increasingly expensive as more sources are incorporated.

Effect of source sample size. To complement the experiment that varies the number of source domains, we also investigate the effect of the per-source sample size n_s, while fixing $p = 50$, $n_0 = 50$, $K = 3$, and $h = 5$. Within each replication, source datasets are generated at the maximum size $n_s = 800$, and smaller values of n_s are obtained by taking the first n_s observations. Figure 5 presents the results in four panels: the top row reports support recovery through FDR and TPR, whereas the bottom row reports estimation accuracy through spectral-

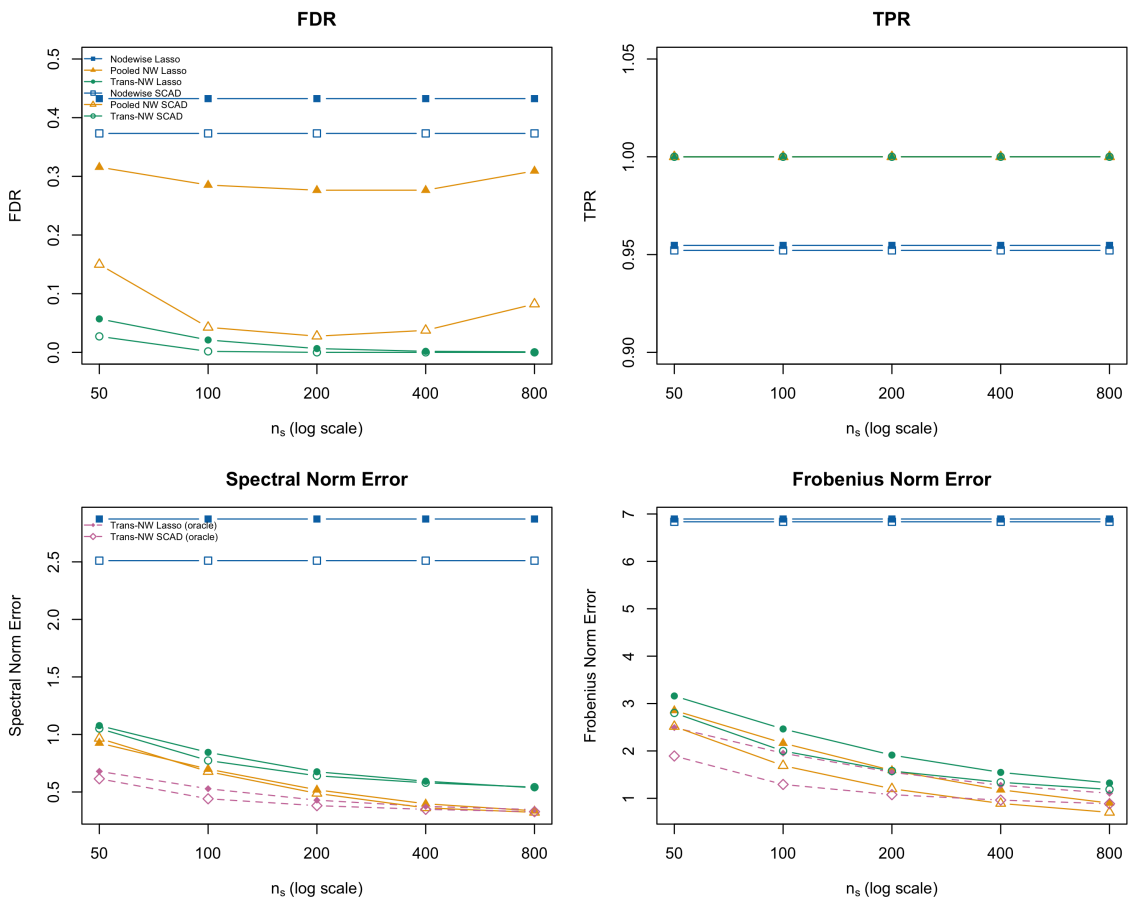

Figure 5: Performance as a function of the per-source sample size n_s under the simulation setting with dimension $p = 50$, target sample size $n_0 = 50$, number of source domains $K = 3$, similarity parameter $h = 5$, AR(1) base covariance with $\rho = 0.6$, and uniform perturbation. Panels report false discovery rate (FDR), true positive rate (TPR), spectral-norm error, and Frobenius-norm error, each averaged over 200 Monte Carlo replications.

and Frobenius-norm errors. In the top panels, increasing n_s has little effect on TPR, which is already near the ceiling for all methods. However, it substantially improves FDR for the transfer learning estimators, with both Trans-Nodewise Lasso and Trans-Nodewise SCAD approaching nearly zero FDR as n_s increases. The target-only estimators remain unchanged across n_s, as expected, since they use only the target data. For the pooled estimators, FDR decreases initially but increases again once n_s becomes large, suggesting that enlarging the source sample size is not uniformly beneficial under naive pooling. In the bottom panels, we also include oracle-diagonal transfer learning estimators as benchmarks. Both pooled and transfer learning estimators improve as n_s increases, with substantial reductions in both spectral- and Frobenius-norm errors. Notably, the pooled estimators can achieve smaller matrix estimation errors for large n_s, but this comes at the cost of noticeably worse FDR. Overall, these results show that, for a fixed number of source domains, larger source samples improve the proposed transfer estimators while preserving their advantage in support recovery.

3.2 Real Data Example

We illustrate the proposed transfer learning estimator through a Markowitz portfolio optimization problem, where accurate estimation of the precision matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ is essential. Let $R = (R_1, \ldots, R_p)^\top \in \mathbb{R}^p$ denote the vector of monthly excess returns for p assets, with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. For a portfolio exposure vector $w = (w_1, \ldots, w_p)^\top \in \mathbb{R}^p$, the excess portfolio return is $w^\top R$, with variance $w^\top \boldsymbol{\Sigma} w$. In Markowitz portfolio theory, the variance of the portfolio return is used as a measure of investment risk, and its square root is the standard deviation, commonly referred to as volatility. This risk can be reduced through diversification by allocating exposure across assets whose returns are weakly or imperfectly correlated.

To construct the minimum-variance portfolio exposure with expected excess return at least μ_0, we solve

$$\min_w \; w^\top \boldsymbol{\Sigma} w \quad \text{subject to} \quad w^\top \boldsymbol{\mu} \geq \mu_0, \tag{21}$$

where $\mu_0 > 0$, short positions are allowed, and no budget constraint is imposed. The optimizer has the closed form

$$w_{\text{opt}} = \frac{\mu_0 \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}}{\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}} = \frac{\mu_0 \boldsymbol{\Omega} \boldsymbol{\mu}}{\boldsymbol{\mu}^\top \boldsymbol{\Omega} \boldsymbol{\mu}}. \tag{22}$$

In practice, we estimate $\boldsymbol{\mu}$ by the sample mean $\widehat{\boldsymbol{\mu}}$ and estimate $\boldsymbol{\Omega}$ using a precision matrix estimator $\widehat{\boldsymbol{\Omega}}$, and then plug $(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Omega}})$ into (22).

In this real data example, we consider constituents of the S&P 500 index as potential assets. A key challenge in estimating $\boldsymbol{\mu}$ and $\boldsymbol{\Omega}$ from historical data is that market dynamics can change over time, so older observations may not follow the same distribution as the target investment period. Such distribution shifts may affect both the mean and covariance structure of stock returns, making direct use of historical data potentially less reliable. Motivated by this concern, we use transfer learning to incorporate historical source data while allowing for distributional differences between the source and target periods.

We use monthly excess returns of S&P 500 constituents from the Center for Research in Security Prices (CRSP) from January 2000 to December 2023, where excess returns are computed by subtracting the 3-month Treasury bill rate from raw returns. We restrict attention to the 196 companies that remain in the index throughout the sample period. This restriction facilitates comparison across time, but the results should be interpreted as a methodological illustration rather than as a fully investable backtest. Figure 6 illustrates the time-series variation in the equal-weighted excess return and the data partition used in the empirical analysis. Specifically, observations from January 2000 to December 2017 are used as source data, observations from January 2018 to December 2020 are used as target training data, and observations from January 2021 to December 2023 are used as out-of-time test data. The source period is treated as a single source domain, so that the transfer learning estimators use $K = 1$. This construction preserves chronological ordering and avoids look-ahead bias, while allowing us to evaluate portfolio performance over multiple test horizons within the same out-of-time period.

For each method, we estimate $\widehat{\boldsymbol{\mu}}$ using the target training period and estimate $\widehat{\boldsymbol{\Omega}}$ using Lasso- and SCAD-based nodewise estimators. Under each penalty, we compare Trans-Nodewise, Pooled Nodewise, and target-only Nodewise estimators. For the transfer learning estimator, all observations before the target period are treated as one auxiliary source domain; for the

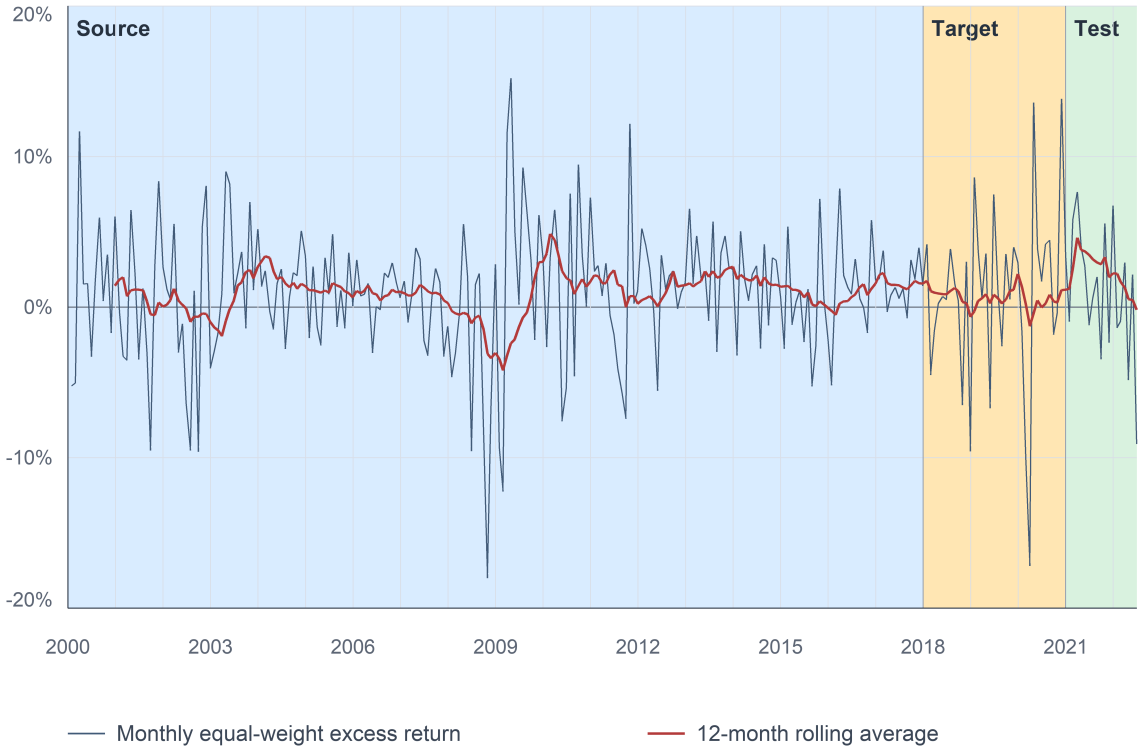

Figure 6: Equal-weighted monthly excess return for the retained S&P 500 constituents from January 2000 through June 2022. Shaded regions indicate the estimation design used in the empirical analysis: source period (January 2000–December 2017), target training period (January 2018–December 2020), and out-of-time test period (January 2021–June 2022). The red line shows the 12-month rolling average.

pooled estimator, the source observations are stacked with the target training observations without adjustment for distributional differences; and for the target-only estimator, only the target training observations are used. Across all precision matrix estimators, we use the same target-period sample mean $\widehat{\boldsymbol{\mu}}$, so that differences in portfolio performance are attributable to the estimated precision matrix $\widehat{\boldsymbol{\Omega}}$. Portfolio exposures are computed by plugging $(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Omega}})$ into (22). Since $\widehat{\boldsymbol{\mu}}$ is estimated from monthly excess returns, we use a monthly target excess return of $\mu_0 = 0.10/12$, corresponding to a target annualized excess return of 10%. The resulting exposures are held fixed throughout the test period. We then evaluate the realized monthly excess returns over horizons of 3, 6, 12 and 18 months from the start of the out-of-time test period, and report annualized mean excess return, annualized volatility, and annualized Sharpe ratio (SR). The monthly mean returns and volatilities are annualized by multiplying by 12 and $\sqrt{12}$, respectively, and the Sharpe ratio is annualized by multiplying the monthly Sharpe ratio by $\sqrt{12}$.

Table 2 summarizes the out-of-time portfolio performance across the four test horizons. At the 3-month horizon, Trans-Nodewise SCAD attains the highest Sharpe ratio among all methods, with an annualized SR of 1.3126. Trans-Nodewise Lasso ranks second, followed by target-only Nodewise SCAD and target-only Nodewise Lasso. In contrast, the pooled estimators perform less favorably over this horizon, with Pooled Nodewise Lasso producing a negative annualized return

Table 2: Annualized mean return, risk, and Sharpe ratio (SR) of S&P 500 portfolios constructed from excess returns, for target annualized excess return 10%.

	Test horizon			
Method	3-month	6-month	12-month	18-month
		Annualized return		
Trans-Nodewise Lasso	0.0842	0.0927	0.1105	0.0230
Pooled Nodewise Lasso	-0.0420	0.0146	0.0628	-0.0005
Nodewise Lasso	0.0818	0.0908	0.1149	0.0247
Trans-Nodewise SCAD	0.0926	0.1030	0.1174	0.0309
Pooled Nodewise SCAD	0.0623	0.0816	0.1041	0.0207
Nodewise SCAD	0.0794	0.0962	0.1150	0.0297
		Annualized volatility		
Trans-Nodewise Lasso	0.0695	0.0506	0.0621	0.0716
Pooled Nodewise Lasso	0.0548	0.0445	0.0536	0.0581
Nodewise Lasso	0.0757	0.0528	0.0648	0.0732
Trans-Nodewise SCAD	0.0706	0.0502	0.0623	0.0720
Pooled Nodewise SCAD	0.0629	0.0462	0.0584	0.0666
Nodewise SCAD	0.0691	0.0501	0.0619	0.0712
		Annualized Sharpe ratio		
Trans-Nodewise Lasso	*1.2107*	*1.8330*	*1.7796*	0.3211
Pooled Nodewise Lasso	-0.7668	0.3290	1.1704	-0.0082
Nodewise Lasso	1.0804	1.7194	1.7748	*0.3374*
Trans-Nodewise SCAD	**1.3126**	**2.0542**	**1.8834**	**0.4298**
Pooled Nodewise SCAD	0.9906	1.7642	1.7807	0.3113
Nodewise SCAD	1.1496	1.9230	1.8585	0.4170

and Sharpe ratio. At the 6- and 12-month horizons, Trans-Nodewise SCAD continues to attain the highest Sharpe ratio among all methods. Trans-Nodewise Lasso also improves substantially over Pooled Nodewise Lasso and performs comparably to the target-only Nodewise Lasso at the 12-month horizon. These results indicate that the transfer learning estimators are competitive over shorter horizons, with the strongest performance observed for the SCAD-based transfer estimator. As the horizon extends to 18 months, the realized Sharpe ratios decline sharply for all methods, reflecting the inclusion of the 2022 market downturn in the evaluation period. Nevertheless, Trans-Nodewise SCAD continues to deliver the highest Sharpe ratio within the SCAD-based group and remains the best overall method at both the 18-month horizons.

Across horizons, the pooled estimators, especially Pooled Nodewise Lasso, tend to have lower volatility but also much lower realized returns, leading to inferior Sharpe ratios. Overall, these results suggest that transfer learning can improve risk-adjusted portfolio performance in this setting, particularly under the SCAD penalty.

4 Discussion

In this paper, we develop a transfer learning estimation procedure for high-dimensional precision matrices to improve the accuracy of the estimation, based on nodewise regression techniques to utilize multiple auxiliary source data sets. We establish the rate of convergence of the proposed estimates and our numerical comparison demonstrates the superior performance of the proposed procedures to the alternative ones. Our empirical analysis of a real-world data set illustrates the advantage of the proposed procedures.

Supplemental Material

Supplemental material includes additional simulation studies and results.

Code Availability

All analysis code is available on the GitHub repository: https://github.com/username/repository, which is an anonymous link in the doubled submission.

References

Hamsa Bastani. Predicting with proxies: Transfer learning in high dimension. *Management Science*, 67(5):2964–2984, 2021.

T. Tony Cai, Weidong Liu, and Harrison H. Zhou. Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation. *The Annals of Statistics*, 44(2):455 – 488, 2016.

Tony Cai, Weidong Liu, and Xi Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Sai Li, T Tony Cai, and Hongzhe Li. Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):149–173, 2022.

Shuo Shuo Liu. Unified transfer learning in high-dimensional linear regression. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 1036–1044. PMLR, 02–04 May 2024.

Shuo Shuo Liu, Haotian Lin, Matthew Reimherr, and Runze Li. Co-regularization enhances knowledge transfer in high dimensions. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Po-Ling Loh and Martin J. Wainwright. Regularized M-estimators with nonconvexity: statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16(1):559–616, 2015.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436 – 1462, 2006.

Ye Tian and Yang Feng. Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association*, 118(544):2684–2697, 2023.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

Sara van de Geer, Peter Bühlmann, Ya'acov Ritov, and Ruben Dezeure. On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42(3):1166 – 1202, 2014.

Lan Wang, Yongdai Kim, and Runze Li. Calibrating nonconvex penalized regression in ultra-high dimension. *The Annals of Statistics*, 41(5):2505–2536, 2013.

Ming Yuan. High dimensional inverse covariance matrix estimation via linear programming. *The Journal of Machine Learning Research*, 11:2261–2286, 2010.

Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894 – 942, 2010.

Appendix: Technical Proofs

This section contains the proofs of our theoretical results in Section 2. We start with the proof of Lemma 1 which formalizes the link between the matrix-level similarity condition and regression-level coefficient similarity.

Proof. By the nodewise regression representation of a precision matrix,

$$\boldsymbol{\beta}_j^{(k)} = -\frac{\boldsymbol{\Omega}_{-j,j}^{(k)}}{\Omega_{jj}^{(k)}}, \qquad k = 0, 1, \ldots, K,$$

where $\boldsymbol{\Omega}_{-j,j}^{(k)}$ denotes the jth column of $\boldsymbol{\Omega}^{(k)}$ with the jth entry removed. Hence,

$$\begin{aligned}\boldsymbol{\delta}_j^{(k)} = \boldsymbol{\beta}_j^{(k)} - \boldsymbol{\beta}_j^{(0)} &= -\frac{\boldsymbol{\Omega}_{-j,j}^{(k)}}{\Omega_{jj}^{(k)}} + \frac{\boldsymbol{\Omega}_{-j,j}^{(0)}}{\Omega_{jj}^{(0)}} \\ &= -\frac{\boldsymbol{\Omega}_{-j,j}^{(k)} - \boldsymbol{\Omega}_{-j,j}^{(0)}}{\Omega_{jj}^{(k)}} - \boldsymbol{\Omega}_{-j,j}^{(0)} \left(\frac{1}{\Omega_{jj}^{(k)}} - \frac{1}{\Omega_{jj}^{(0)}} \right).\end{aligned}$$

Taking ℓ_1 norms yields

$$\|\boldsymbol{\delta}_j^{(k)}\|_1 \le \frac{\|\boldsymbol{\Omega}_{-j,j}^{(k)} - \boldsymbol{\Omega}_{-j,j}^{(0)}\|_1}{|\Omega_{jj}^{(k)}|} + \|\boldsymbol{\Omega}_{-j,j}^{(0)}\|_1 \cdot \frac{|\Omega_{jj}^{(k)} - \Omega_{jj}^{(0)}|}{|\Omega_{jj}^{(k)}|\,|\Omega_{jj}^{(0)}|}.$$

Using $\boldsymbol{\Omega}_{-j,j}^{(0)} = -\boldsymbol{\beta}_j^{(0)}\Omega_{jj}^{(0)}$, we have $\|\boldsymbol{\Omega}_{-j,j}^{(0)}\|_1 = \|\boldsymbol{\beta}_j^{(0)}\|_1\,|\Omega_{jj}^{(0)}|$, so the second term becomes

$$\|\boldsymbol{\Omega}_{-j,j}^{(0)}\|_1 \cdot \frac{|\Omega_{jj}^{(k)} - \Omega_{jj}^{(0)}|}{|\Omega_{jj}^{(k)}|\,|\Omega_{jj}^{(0)}|} = \|\boldsymbol{\beta}_j^{(0)}\|_1 \cdot \frac{|\Omega_{jj}^{(k)} - \Omega_{jj}^{(0)}|}{|\Omega_{jj}^{(k)}|}.$$

Therefore,

$$\|\boldsymbol{\delta}_j^{(k)}\|_1 \le \frac{1}{|\Omega_{jj}^{(k)}|}\left\{\|\boldsymbol{\Omega}_{-j,j}^{(k)} - \boldsymbol{\Omega}_{-j,j}^{(0)}\|_1 + \|\boldsymbol{\beta}_j^{(0)}\|_1\,|\Omega_{jj}^{(k)} - \Omega_{jj}^{(0)}|\right\}.$$

Taking the maximum over k and multiplying both sides by $\min_{1\le k\le K}|\Omega_{jj}^{(k)}|$ gives

$$\begin{aligned} h_j \cdot \min_{1\le k\le K}|\Omega_{jj}^{(k)}| &\le \max_{1\le k\le K}\left\{\|\boldsymbol{\Omega}_{-j,j}^{(k)} - \boldsymbol{\Omega}_{-j,j}^{(0)}\|_1 + \|\boldsymbol{\beta}_j^{(0)}\|_1\,|\Omega_{jj}^{(k)} - \Omega_{jj}^{(0)}|\right\} \\ &\le \max_{1\le k\le K}\|\boldsymbol{\Omega}_{-j,j}^{(k)} - \boldsymbol{\Omega}_{-j,j}^{(0)}\|_1 + \|\boldsymbol{\beta}_j^{(0)}\|_1\,h_{D,j}. \end{aligned}$$

Next, we have

$$\|\boldsymbol{\Omega}_{-j,j}^{(k)} - \boldsymbol{\Omega}_{-j,j}^{(0)}\|_1 \le \|\boldsymbol{\Omega}_j^{(k)} - \boldsymbol{\Omega}_j^{(0)}\|_1 \le \|\boldsymbol{\Omega}^{(k)} - \boldsymbol{\Omega}^{(0)}\|_1,$$

where $\boldsymbol{\Omega}_j^{(k)}$ denotes the jth column of $\boldsymbol{\Omega}^{(k)}$ and hence

$$\max_{1\le k\le K}\|\boldsymbol{\Omega}_{-j,j}^{(k)} - \boldsymbol{\Omega}_{-j,j}^{(0)}\|_1 \le h.$$

Thus, by using $\min_k|\Omega_{jj}^{(k)}| \ge M_0^{-1}$, we obtain

$$h_j \le M_0\left(h + \|\boldsymbol{\beta}_j^{(0)}\|_1\,h_{D,j}\right).$$

Finally, by the assumption $\max_{1\le j\le p}\left\{1 + \|\boldsymbol{\beta}_j^{(0)}\|_1\,h_{D,j}/h\right\} \le C_0$, we have $h + \|\boldsymbol{\beta}_j^{(0)}\|_1 h_{D,j} \le C_0 h$ for every j, which implies

$$h_j \le M_0 C_0\,h = Mh.$$

□

To prove Theorem 1, we start with the following lemma on the RSC condition for nodewise regression.

Lemma 2. *Suppose the sample covariance matrix $\widehat{\boldsymbol{\Sigma}}^{(k)}$ satisfies the restricted strong convexity condition in Assumption 3. Then, for any $1 \le j \le p$, the principal submatrix $\widehat{\boldsymbol{\Sigma}}_{-j,-j}^{(k)}$ also satisfies*

$$\boldsymbol{\Delta}^\top\widehat{\boldsymbol{\Sigma}}_{-j,-j}^{(k)}\boldsymbol{\Delta} \ge \nu_k\|\boldsymbol{\Delta}\|_2^2 - \tau_k\sqrt{\frac{\log p}{n_k}}\|\boldsymbol{\Delta}\|_1,$$

for any $\boldsymbol{\Delta} \in \mathbb{R}^{p-1}$ with $\|\boldsymbol{\Delta}\|_1 \ge 1$, where $\nu_k > 0$ and $\tau_k > 0$.

Proof. Let $\boldsymbol{\Delta}' \in \mathbb{R}^{p-1}$ satisfy $\|\boldsymbol{\Delta}'\|_1 \geq 1$. Define $\boldsymbol{\Delta} = (\Delta_1, \ldots, \Delta_p)^\top \in \mathbb{R}^p$ by

$$\Delta_i = \begin{cases} \Delta'_i, & i < j, \\ 0, & i = j, \\ \Delta'_{i-1}, & i > j. \end{cases}$$

Then $\|\boldsymbol{\Delta}\|_1 = \|\boldsymbol{\Delta}'\|_1 \geq 1$ and $\|\boldsymbol{\Delta}\|_2 = \|\boldsymbol{\Delta}'\|_2$. Moreover,

$$(\boldsymbol{\Delta}')^\top \widehat{\boldsymbol{\Sigma}}^{(k)}_{-j,-j} \boldsymbol{\Delta}' = \boldsymbol{\Delta}^\top \widehat{\boldsymbol{\Sigma}}^{(k)} \boldsymbol{\Delta}.$$

Therefore, by the RSC condition for $\widehat{\boldsymbol{\Sigma}}^{(k)}$,

$$(\boldsymbol{\Delta}')^\top \widehat{\boldsymbol{\Sigma}}^{(k)}_{-j,-j} \boldsymbol{\Delta}' = \boldsymbol{\Delta}^\top \widehat{\boldsymbol{\Sigma}}^{(k)} \boldsymbol{\Delta} \geq \nu_k \|\boldsymbol{\Delta}\|_2^2 - \tau_k \sqrt{\frac{\log p}{n_k}} \|\boldsymbol{\Delta}\|_1 = \nu_k \|\boldsymbol{\Delta}'\|_2^2 - \tau_k \sqrt{\frac{\log p}{n_k}} \|\boldsymbol{\Delta}'\|_1.$$

This proves the result. □

Lemma 2 shows that the sample covariance matrix of $\mathbf{X}^{(k)}_{-j}$ satisfies the RSC condition for $1 \leq j \leq p$, provided that the sample covariance matrix of $\mathbf{X}^{(k)}$ does. Building on this result, we present the following two lemmas. Lemma 3 establishes the RSC condition for the reorganized design matrix $\mathbf{X}$ in Equation (19). Subsequently, Lemma 4 gives the estimation error bound for the target coefficient vector in the jth nodewise regression.

Lemma 3. *Let $\widehat{\boldsymbol{\Sigma}} = \mathbf{X}^\top \mathbf{X}/N$, where $\mathbf{X}$ is defined in Equation (19). If, for each $k = 0, 1, \ldots, K$, the submatrix $\widehat{\boldsymbol{\Sigma}}^{(k)}_{-j,-j}$ satisfies the restricted strong convexity condition with parameters ν_k and τ_k, then*

$$\boldsymbol{\Delta}^\top \widehat{\boldsymbol{\Sigma}} \boldsymbol{\Delta} \geq \kappa_K \, \nu_{\min} \|\boldsymbol{\Delta}\|_2^2 - \tau_{\max} \left(\sqrt{\frac{n_{\max} \log p}{N^2}} + \sqrt{\frac{n_0 \log p}{N^2}} \right) \|\boldsymbol{\Delta}\|_1$$

for any $\boldsymbol{\Delta} \in \mathbb{R}^{(K+1)(p-1)}$, where $\nu_{\min} = \min_{0 \leq k \leq K} \nu_k n_k / N$, $\tau_{\max} = (K+1) \max_{0 \leq k \leq K} \tau_k$, $n_{\max} = \max_{1 \leq k \leq K} n_k$, and $\kappa_K := 2/(K + 2 + \sqrt{K^2 + 4K})$.

Proof. Let

$$\widetilde{\mathbf{X}} := \begin{bmatrix} \mathbf{X}^{(1)}_{-j} & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{X}^{(2)}_{-j} & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & \mathbf{X}^{(K)}_{-j} & 0 \\ 0 & 0 & \cdots & 0 & \mathbf{X}^{(0)}_{-j} \end{bmatrix} \in \mathbb{R}^{N \times (K+1)(p-1)}$$

and let T be the $(K+1)(p-1) \times (K+1)(p-1)$ block matrix

$$T := \begin{bmatrix} \mathbf{I}_{p-1} & 0 & \cdots & 0 & \mathbf{I}_{p-1} \\ 0 & \mathbf{I}_{p-1} & \cdots & 0 & \mathbf{I}_{p-1} \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & \mathbf{I}_{p-1} & \mathbf{I}_{p-1} \\ 0 & 0 & \cdots & 0 & \mathbf{I}_{p-1} \end{bmatrix} = \begin{pmatrix} \mathbf{I}_K & \mathbf{1}_K \\ \mathbf{0}_K^\top & 1 \end{pmatrix} \otimes \mathbf{I}_{p-1}.$$

The stacked design $\mathbf{X}$ defined in (19) and the block-diagonal design $\widetilde{\mathbf{X}}$ are related by $\mathbf{X} = \widetilde{\mathbf{X}}\,T$. We have

$$\widehat{\boldsymbol{\Sigma}} \;=\; \frac{\mathbf{X}^\top\mathbf{X}}{N} \;=\; T^\top\,\widetilde{\boldsymbol{\Sigma}}\,T, \quad \text{where } \widetilde{\boldsymbol{\Sigma}} \;:=\; \frac{\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}}}{N} = \operatorname{diag}\big(\tfrac{n_1}{N}\widehat{\boldsymbol{\Sigma}}^{(1)}_{-j,-j},\ldots,\tfrac{n_K}{N}\widehat{\boldsymbol{\Sigma}}^{(K)}_{-j,-j},\tfrac{n_0}{N}\widehat{\boldsymbol{\Sigma}}^{(0)}_{-j,-j}\big). \tag{23}$$

The per-source RSC of Lemma 2 can be extended from $\{\|\boldsymbol{w}\|_1 \geq 1\}$ to all of $\mathbb{R}^{p-1}$ by a rescaling argument (apply Lemma 2 to $\boldsymbol{w}/\|\boldsymbol{w}\|_1$ when $\|\boldsymbol{w}\|_1 < 1$, multiply by $\|\boldsymbol{w}\|_1^2$, and use $\|\boldsymbol{w}\|_1^2 \leq \|\boldsymbol{w}\|_1$). Hence, for any $\boldsymbol{w} \in \mathbb{R}^{p-1}$ and any k,

$$\boldsymbol{w}^\top\widehat{\boldsymbol{\Sigma}}^{(k)}_{-j,-j}\boldsymbol{w} \;\geq\; \nu_k\|\boldsymbol{w}\|_2^2 - \tau_k\sqrt{\tfrac{\log p}{n_k}}\|\boldsymbol{w}\|_1.$$

For an arbitrary $\boldsymbol{\xi} = (\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_K,\boldsymbol{\xi}_0) \in \mathbb{R}^{(K+1)(p-1)}$, applying this block by block to $\widetilde{\boldsymbol{\Sigma}}$ and using $n_k\nu_k/N \geq \nu_{\min}$,

$$\boldsymbol{\xi}^\top\widetilde{\boldsymbol{\Sigma}}\boldsymbol{\xi} \;=\; \sum_{k=0}^{K}\frac{n_k}{N}\,\boldsymbol{\xi}_k^\top\widehat{\boldsymbol{\Sigma}}^{(k)}_{-j,-j}\boldsymbol{\xi}_k \;\geq\; \nu_{\min}\|\boldsymbol{\xi}\|_2^2 \;-\; \sum_{k=0}^{K}\tau_k\sqrt{\tfrac{n_k\log p}{N^2}}\,\|\boldsymbol{\xi}_k\|_1. \tag{24}$$

From the factorization (23),

$$\boldsymbol{\Delta}^\top\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Delta} \;=\; \boldsymbol{\Delta}^\top T^\top\widetilde{\boldsymbol{\Sigma}}T\boldsymbol{\Delta} \;=\; (T\boldsymbol{\Delta})^\top\widetilde{\boldsymbol{\Sigma}}(T\boldsymbol{\Delta}),$$

so applying (24) with $\boldsymbol{\xi} = T\boldsymbol{\Delta}$ gives

$$\boldsymbol{\Delta}^\top\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Delta} \;\geq\; \nu_{\min}\|T\boldsymbol{\Delta}\|_2^2 \;-\; \mathcal{R}(\boldsymbol{\Delta}) \;=\; \nu_{\min}\,\boldsymbol{\Delta}^\top T^\top T\boldsymbol{\Delta} \;-\; \mathcal{R}(\boldsymbol{\Delta}),$$

where, writing $\boldsymbol{\Delta} = (\boldsymbol{u}^{(1)\top},\ldots,\boldsymbol{u}^{(K)\top},\boldsymbol{v}^\top)^\top$,

$$\mathcal{R}(\boldsymbol{\Delta}) \;:=\; \sum_{k=0}^{K}\tau_k\sqrt{\tfrac{n_k\log p}{N^2}}\,\|(T\boldsymbol{\Delta})_k\|_1 \;=\; \tau_0\sqrt{\tfrac{n_0\log p}{N^2}}\|\boldsymbol{v}\|_1 + \sum_{k=1}^{K}\tau_k\sqrt{\tfrac{n_k\log p}{N^2}}\|\boldsymbol{u}^{(k)}+\boldsymbol{v}\|_1.$$

Bounding $\boldsymbol{\Delta}^\top T^\top T\boldsymbol{\Delta} \geq \kappa_K\|\boldsymbol{\Delta}\|_2^2$ via $\lambda_{\min}(T^\top T) = 2/(K+2+\sqrt{K^2+4K}) = \kappa_K$ yields

$$\boldsymbol{\Delta}^\top\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Delta} \;\geq\; \kappa_K\,\nu_{\min}\,\|\boldsymbol{\Delta}\|_2^2 \;-\; \mathcal{R}(\boldsymbol{\Delta}). \tag{25}$$

By the triangle inequality and $\sum_{k=1}^{K}\|\boldsymbol{u}^{(k)}\|_1 = \|\boldsymbol{\Delta}\|_1 - \|\boldsymbol{v}\|_1$,

$$\sum_{k=1}^{K}\|\boldsymbol{u}^{(k)}+\boldsymbol{v}\|_1 \leq \sum_{k=1}^{K}(\|\boldsymbol{u}^{(k)}\|_1 + \|\boldsymbol{v}\|_1) = \|\boldsymbol{\Delta}\|_1 + (K-1)\|\boldsymbol{v}\|_1 \leq K\|\boldsymbol{\Delta}\|_1.$$

Using $\tau_k \leq \tau_{\max}/(K+1)$ and $\sqrt{n_k\log p/N^2} \leq \sqrt{n_{\max}\log p/N^2}$ for $k \geq 1$, together with $\|\boldsymbol{v}\|_1 \leq$

$\|\mathbf{\Delta}\|_1$,

$$\begin{aligned}\mathcal{R}(\mathbf{\Delta}) &\le \frac{\tau_{\max}}{K+1}\sqrt{\tfrac{n_0 \log p}{N^2}}\,\|\mathbf{\Delta}\|_1 + \frac{\tau_{\max}}{K+1}\sqrt{\tfrac{n_{\max}\log p}{N^2}}\cdot K\|\mathbf{\Delta}\|_1 \\ &\le \frac{K\tau_{\max}}{K+1}\left(\sqrt{\tfrac{n_{\max}\log p}{N^2}} + \sqrt{\tfrac{n_0\log p}{N^2}}\right)\|\mathbf{\Delta}\|_1 \\ &\le \tau_{\max}\left(\sqrt{\tfrac{n_{\max}\log p}{N^2}} + \sqrt{\tfrac{n_0\log p}{N^2}}\right)\|\mathbf{\Delta}\|_1,\end{aligned}$$

the second inequality using $1 \le K$ on the target term and the third using $K/(K+1) \le 1$. Plugging into (25) completes the proof. □

Lemma 4. *Suppose Assumptions 1–5 hold and* $(\mathbf{\Omega}^{(0)}, \mathbf{\Omega}^{(1)}, \ldots, \mathbf{\Omega}^{(K)}) \in \mathbb{G}(s,h)$. *Let* $\widehat{\boldsymbol{b}}_j$ *be a stationary point of* (12) *computed with* $\lambda_{j0} = \cdots = \lambda_{jK} = \lambda$, *and let* $\widehat{\boldsymbol{\beta}}_j = \mathrm{tail}(\widehat{\boldsymbol{b}}_j, p-1)$. *Set*

$$a = 2\kappa_K \nu_{\min} - \tfrac{3\zeta}{2}, \qquad s_b = (1+K+CK)s, \qquad r_N = \frac{\tau_{\max}}{N}(\sqrt{n_{\max}\log p} + \sqrt{n_0 \log p}),$$

with κ_K, $\nu_{\min}$, *and* $\tau_{\max}$ *as in Lemma 3. Assume* $\log p = o(N)$ *and* $\lambda L/2 \ge c_\varepsilon\sqrt{\log p/N} + r_N$ *for some constant* c_ε. *Then there exist positive constants* c, c_1, c_2, c_3 *such that, for all sufficiently large* p, *with probability at least* $1 - cp^{-1}$,

$$\|\widehat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j^{(0)}\|_1 \le \frac{c_1 L\sqrt{s_b s}}{a}\left(\frac{\log p}{N}\right)^{1/2} + \frac{c_2\sqrt{LKs_b h_j}}{\sqrt{a}}\left(\frac{\log p}{N}\right)^{1/4} + \frac{c_3\,\zeta L s_b}{a^2}\left(\frac{\log p}{N}\right)^{1/2}. \tag{26}$$

For the Lasso the last term vanishes, since $\zeta = 0$.

Proof. Write the objective in (12) as $\mathcal{L}(\boldsymbol{b}) + P_\lambda(\boldsymbol{b})$ with $\mathcal{L}(\boldsymbol{b}) = \frac{1}{2N}\|\mathbf{y} - \mathbf{X}\boldsymbol{b}\|_2^2$ and $\nabla\mathcal{L}(\boldsymbol{b}) = \widehat{\mathbf{\Sigma}}\boldsymbol{b} - N^{-1}\mathbf{X}^\top\mathbf{y}$, where $(\mathbf{y}, \mathbf{X})$ are the stacked response and design in (19) and $\widehat{\mathbf{\Sigma}} = \mathbf{X}^\top\mathbf{X}/N$. Let $\boldsymbol{b}_j$ denote the joint truth at node j and $\widehat{\mathbf{\Delta}} = \widehat{\boldsymbol{b}}_j - \boldsymbol{b}_j$, so that the target error is the last block of $\widehat{\mathbf{\Delta}}$ and $\|\widehat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j^{(0)}\|_1 \le \|\widehat{\mathbf{\Delta}}\|_1$. Stationarity of $\widehat{\boldsymbol{b}}_j$ gives $\langle \nabla\mathcal{L}(\widehat{\boldsymbol{b}}_j) + \nabla P_\lambda(\widehat{\boldsymbol{b}}_j), \boldsymbol{b}_j - \widehat{\boldsymbol{b}}_j\rangle \ge 0$ for a fixed subgradient selection $\nabla P_\lambda(\widehat{\boldsymbol{b}}_j)$. Writing $\mathbf{y} = \mathbf{X}\boldsymbol{b}_j + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ stacks the per-domain nodewise residuals $\boldsymbol{\varepsilon}_j^{(k)}$, one has $N^{-1}\mathbf{X}^\top\mathbf{y} - \widehat{\mathbf{\Sigma}}\boldsymbol{b}_j = N^{-1}\mathbf{X}^\top\boldsymbol{\varepsilon}$, and the stationarity inequality together with Hölder's inequality yields

$$\widehat{\mathbf{\Delta}}^\top\widehat{\mathbf{\Sigma}}\widehat{\mathbf{\Delta}} \le \|N^{-1}\mathbf{X}^\top\boldsymbol{\varepsilon}\|_\infty\|\widehat{\mathbf{\Delta}}\|_1 - \langle \nabla P_\lambda(\widehat{\boldsymbol{b}}_j), \widehat{\mathbf{\Delta}}\rangle. \tag{27}$$

The l-th coordinate of $N^{-1}\mathbf{X}^\top\boldsymbol{\varepsilon}$ is $N^{-1}\sum_{i=1}^N X_{il}\varepsilon_i$, a sum of independent terms with $\mathrm{E}[X_{il}\varepsilon_i] = 0$. The mean-zero property holds because $\boldsymbol{\beta}_j^{(k)} = (\mathbf{\Sigma}_{-j,-j}^{(k)})^{-1}\mathbf{\Sigma}_{-j,j}^{(k)}$ gives $\mathrm{E}[\mathbf{x}_{-j}^{(k)}\varepsilon_j^{(k)}] = \mathbf{\Sigma}_{-j,j}^{(k)} - \mathbf{\Sigma}_{-j,-j}^{(k)}\boldsymbol{\beta}_j^{(k)} = \mathbf{0}$, of which $\mathrm{E}[X_{il}\varepsilon_i] = 0$ is the entry indexed by (l,i). Under Assumption 1 the regressors X_{il} and the residuals ε_i are sub-Gaussian, so each product $X_{il}\varepsilon_i$ is sub-exponential with ψ_1-norm bounded by a constant $M < \infty$. By the Bernstein inequality, there is a constant $c_0 > 0$ such that, for every $t > 0$,

$$\mathrm{P}\Big(\big|N^{-1}\textstyle\sum_{i=1}^N X_{il}\varepsilon_i\big| \ge t\Big) \le 2\exp\Big[-c_0 N\min(t^2/M^2,\, t/M)\Big].$$

Take $t = c_\varepsilon\sqrt{\log p/N}$. As $\log p = o(N)$, the minimum is attained at t^2/M^2 and

$$\mathrm{P}\Big(|N^{-1}\textstyle\sum_{i=1}^N X_{il}\varepsilon_i| \geq c_\varepsilon\sqrt{\frac{\log p}{N}}\Big) \leq 2\exp\!\big(-c_0\, c_\varepsilon^2 \log p/M^2\big) \leq 2p^{-2},$$

where the final bound holds once c_ε is large enough that $c_0\, c_\varepsilon^2/M^2 \geq 2$. A union bound over the $(K+1)(p-1) = O(p)$ coordinates, using $K = O(1)$, gives with probability at least $1 - cp^{-1}$

$$\|N^{-1}\mathbf{X}^\top\boldsymbol{\varepsilon}\|_\infty \leq c_\varepsilon\sqrt{\tfrac{\log p}{N}}. \tag{28}$$

By Lemma 3, $\widehat{\boldsymbol{\Delta}}^\top\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\Delta}} \geq \kappa_K\nu_{\min}\|\widehat{\boldsymbol{\Delta}}\|_2^2 - r_N\|\widehat{\boldsymbol{\Delta}}\|_1$. Convexity of $P_\lambda + \frac{\zeta}{2}\|\cdot\|_2^2$ from amenability (iv) gives $-\langle\nabla P_\lambda(\widehat{\boldsymbol{b}}_j), \widehat{\boldsymbol{\Delta}}\rangle \leq P_\lambda(\boldsymbol{b}_j) - P_\lambda(\widehat{\boldsymbol{b}}_j) + \frac{\zeta}{2}\|\widehat{\boldsymbol{\Delta}}\|_2^2$. Substituting these and (28) into (27), together with $c_\varepsilon\sqrt{\log p/N} + r_N \leq \lambda L/2$, and then applying the second bound of (18) to $\widehat{\boldsymbol{\Delta}}$ to control $\frac{\lambda L}{2}\|\widehat{\boldsymbol{\Delta}}\|_1 \leq \frac{1}{2}P_\lambda(\widehat{\boldsymbol{\Delta}}) + \frac{\zeta}{4}\|\widehat{\boldsymbol{\Delta}}\|_2^2$, we obtain

$$\kappa_K\nu_{\min}\|\widehat{\boldsymbol{\Delta}}\|_2^2 \leq P_\lambda(\boldsymbol{b}_j) - P_\lambda(\widehat{\boldsymbol{b}}_j) + \tfrac{1}{2}P_\lambda(\widehat{\boldsymbol{\Delta}}) + \tfrac{3\zeta}{4}\|\widehat{\boldsymbol{\Delta}}\|_2^2. \tag{29}$$

Each summand p_λ is even with $p_\lambda(0) = 0$, satisfies $p_\lambda(t) \leq \lambda L|t|$ by the first bound of (18), and satisfies $p_\lambda(u+v) \leq p_\lambda(u) + p_\lambda(v)$ because $t \mapsto p_\lambda(t)/t$ is nonincreasing. For any index set T, splitting $P_\lambda(\boldsymbol{b}_j) - P_\lambda(\widehat{\boldsymbol{b}}_j) + \frac{1}{2}P_\lambda(\widehat{\boldsymbol{\Delta}})$ coordinatewise with these two properties bounds it by $\frac{3\lambda L}{2}\|\widehat{\boldsymbol{\Delta}}_T\|_1 + 2\lambda L\|\boldsymbol{b}_{j,T^c}\|_1 - \frac{1}{2}P_\lambda(\widehat{\boldsymbol{\Delta}}_{T^c})$. Substituting this into (29) and recalling $a = 2\kappa_K\nu_{\min} - \frac{3\zeta}{2}$,

$$\tfrac{a}{2}\|\widehat{\boldsymbol{\Delta}}\|_2^2 + \tfrac{1}{2}P_\lambda(\widehat{\boldsymbol{\Delta}}_{T^c}) \leq \tfrac{3\lambda L}{2}\|\widehat{\boldsymbol{\Delta}}_T\|_1 + 2\lambda L\|\boldsymbol{b}_{j,T^c}\|_1, \tag{30}$$

which holds for every T and has nonnegative left-hand side since $a > 0$.

Let S be the support of $\boldsymbol{\beta}_j^{(0)}$ within the target block, so $|S| \leq s$ and $\|\boldsymbol{b}_{j,S^c}\|_1 = \sum_{k=1}^K \|\boldsymbol{\delta}_j^{(k)}\|_1 \leq Kh_j$. Taking $T = S$ in (30), discarding $\frac{1}{2}P_\lambda(\widehat{\boldsymbol{\Delta}}_{S^c}) \geq 0$, and using $\|\widehat{\boldsymbol{\Delta}}_S\|_1 \leq \sqrt{s}\,\|\widehat{\boldsymbol{\Delta}}\|_2$ gives $\frac{a}{2}\|\widehat{\boldsymbol{\Delta}}\|_2^2 \leq \frac{3\lambda L}{2}\sqrt{s}\,\|\widehat{\boldsymbol{\Delta}}\|_2 + 2\lambda LKh_j$, hence

$$\|\widehat{\boldsymbol{\Delta}}\|_2 \leq \frac{3\lambda L\sqrt{s}}{a} + 2\sqrt{\frac{\lambda LKh_j}{a}}. \tag{31}$$

Under $\mathbb{G}(s,h)$ each contrast $\boldsymbol{\delta}_j^{(k)}$ has at most $(1+C)s$ nonzero entries, so $\boldsymbol{b}_j$ is supported on a set $\widetilde{S}$ with $|\widetilde{S}| \leq K(1+C)s + s = s_b$. Taking $T = \widetilde{S}$ in (30), where $\boldsymbol{b}_{j,\widetilde{S}^c} = \mathbf{0}$ and $\|\widehat{\boldsymbol{\Delta}}_{\widetilde{S}}\|_1 \leq \sqrt{s_b}\,\|\widehat{\boldsymbol{\Delta}}\|_2$, we have

$$\|\widehat{\boldsymbol{\Delta}}\|_2 \leq \frac{3\lambda L\sqrt{s_b}}{a}, \tag{32}$$

and, after applying the second bound of (18) to $\widehat{\boldsymbol{\Delta}}_{\widetilde{S}^c}$,

$$\|\widehat{\boldsymbol{\Delta}}\|_1 \leq 4\sqrt{s_b}\,\|\widehat{\boldsymbol{\Delta}}\|_2 + \frac{\zeta}{2\lambda L}\|\widehat{\boldsymbol{\Delta}}\|_2^2. \tag{33}$$

Bounding the first term of (33) by (31) and the second term by (32),

$$\|\widehat{\boldsymbol{\Delta}}\|_1 \leq \frac{12\lambda L\sqrt{s_b s}}{a} + 8\sqrt{\frac{\lambda LKs_b h_j}{a}} + \frac{9\zeta\lambda Ls_b}{2a^2}.$$

Substituting $\lambda \asymp \sqrt{\log p/N}$ and using $\|\widehat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j^{(0)}\|_1 \le \|\widehat{\boldsymbol{\Delta}}\|_1$ yields (26), with c_1, c_2, c_3 absorbing the constants and the stated probability that of (28). $\square$

The next two lemmas provide convergence rates for the diagonal elements of the target precision matrix estimator. Because $\Omega_{jj}^{(0)}$ is linked to the nodewise regression residual variance through $\Omega_{jj}^{(0)} = 1/\tau_j^{(0)}$, the analysis reduces to control the estimation error of $\widehat{\tau}_j$ and then apply a simple inversion argument.

Lemma 5. *Under Assumptions 1–3, suppose that for each $j = 1, \ldots, p$, $M_0^{-1} \le \Omega_{jj}^{(0)} \le M_0$, and*

$$|\widehat{\tau}_j - \tau_j^{(0)}| \le c\sqrt{\frac{s_b \log p_b}{N}},$$

where $\tau_j^{(0)} = 1/\Omega_{jj}^{(0)}$, $\widehat{\tau}_j = \|\mathbf{y} - \mathbf{X}\widehat{\mathbf{b}}_j\|_2^2/(N - \widehat{s}_{bj})$, with $s_b = (1+K+CK)s$ and $p_b = (K+1)(p-1)$. Then, provided $cM_0\sqrt{s_b \log p_b/N} \le \frac{1}{2}$, we have

$$\max_{1\le j\le p} |\widehat{\Omega}_{jj} - \Omega_{jj}^{(0)}| \le 2cM_0^2\sqrt{\frac{s_b \log p_b}{N}} \lesssim M_0^2\sqrt{\frac{s \log p}{N}}.$$

Moreover, for each $j = 1, \ldots, p$,

$$\left|\frac{\widehat{\Omega}_{jj}}{\Omega_{jj}^{(0)}} - 1\right| \le 2cM_0\sqrt{\frac{s_b \log p_b}{N}}, \quad \text{and} \quad \widehat{\Omega}_{jj} \le M_0\left(1 + 2cM_0\sqrt{\frac{s_b \log p_b}{N}}\right).$$

Proof. Let $r = \sqrt{s_b \log p_b/N}$. Since

$$\tau_j^{(0)} = \frac{1}{\Omega_{jj}^{(0)}} \qquad \text{and} \qquad \widehat{\tau}_j = \frac{1}{\widehat{\Omega}_{jj}},$$

the assumed bound implies

$$\left|\frac{1}{\widehat{\Omega}_{jj}} - \frac{1}{\Omega_{jj}^{(0)}}\right| = |\widehat{\tau}_j - \tau_j^{(0)}| \le cr.$$

Multiplying both sides by $\Omega_{jj}^{(0)}$, we obtain

$$\left|\frac{\Omega_{jj}^{(0)}}{\widehat{\Omega}_{jj}} - 1\right| \le cr\,\Omega_{jj}^{(0)} \le crM_0.$$

Hence

$$1 - crM_0 \le \frac{\Omega_{jj}^{(0)}}{\widehat{\Omega}_{jj}} \le 1 + crM_0.$$

Since $crM_0 \le 1/2$, we have

$$\frac{1}{1 + crM_0} \le \frac{\widehat{\Omega}_{jj}}{\Omega_{jj}^{(0)}} \le \frac{1}{1 - crM_0} \le 1 + 2crM_0,$$

and therefore

$$\left| \frac{\widehat{\Omega}_{jj}}{\Omega_{jj}^{(0)}} - 1 \right| \leq 2crM_0.$$

Multiplying by $\Omega_{jj}^{(0)} \leq M_0$ yields

$$|\widehat{\Omega}_{jj} - \Omega_{jj}^{(0)}| \leq 2crM_0\,\Omega_{jj}^{(0)} \leq 2cM_0^2 r.$$

Taking maximum over $1 \leq j \leq p$, we get

$$\max_{1\leq j\leq p} |\widehat{\Omega}_{jj} - \Omega_{jj}^{(0)}| \leq 2cM_0^2 r.$$

Finally, since $s_b = (1 + K + CK)s$ and $p_b = (K + 1)(p - 1)$, and K is treated as fixed,

$$r = \sqrt{\frac{s_b \log p_b}{N}} \lesssim \sqrt{\frac{s \log p}{N}}.$$

This proves the first claim. The upper bound

$$\widehat{\Omega}_{jj} \leq \Omega_{jj}^{(0)}(1 + 2crM_0) \leq M_0\,(1 + 2cM_0 r)$$

follows immediately. □

Lemma 6. *Under Assumptions 1–3, suppose $M_0^{-1} \leq \Omega_{jj}^{(0)} \leq M_0$ for $j = 1, \ldots, p$, and*

$$|\widehat{\tau}_j - \tau_j^{(0)}| \lesssim \sqrt{\frac{s \log p}{n_0}},$$

for $\widehat{\tau}_j = \|X_j^{(0)} - X_{-j}^{(0)}\widehat{\boldsymbol{\beta}}_j\|_2^2/(n_0 - \widehat{s}_j)$, then

$$\max_{1\leq j\leq p} |\widehat{\Omega}_{jj} - \Omega_{jj}^{(0)}| \lesssim M_0^2 \sqrt{\frac{s \log p}{n_0}}.$$

Proof. Let $r = \sqrt{s \log p/n_0}$. Under the assumed bound

$$|\widehat{\tau}_j - \tau_j^{(0)}| \leq cr$$

for some constant $c > 0$, and using

$$\tau_j^{(0)} = \frac{1}{\Omega_{jj}^{(0)}}, \qquad \widehat{\tau}_j = \frac{1}{\widehat{\Omega}_{jj}},$$

we obtain

$$\left| \frac{1}{\widehat{\Omega}_{jj}} - \frac{1}{\Omega_{jj}^{(0)}} \right| = |\widehat{\tau}_j - \tau_j^{(0)}| \leq cr.$$

Multiplying both sides by $\Omega_{jj}^{(0)}$ gives

$$\left|\frac{\Omega_{jj}^{(0)}}{\widehat{\Omega}_{jj}} - 1\right| \leq cr\,\Omega_{jj}^{(0)} \leq crM_0.$$

Hence,

$$1 - crM_0 \leq \frac{\Omega_{jj}^{(0)}}{\widehat{\Omega}_{jj}} \leq 1 + crM_0.$$

For n_0 sufficiently large so that $crM_0 \leq 1/2$, it follows that

$$\frac{\widehat{\Omega}_{jj}}{\Omega_{jj}^{(0)}} \leq \frac{1}{1 - crM_0} \leq 1 + 2crM_0,$$

and similarly,

$$\frac{\widehat{\Omega}_{jj}}{\Omega_{jj}^{(0)}} \geq \frac{1}{1 + crM_0} \geq 1 - crM_0.$$

Therefore,

$$|\widehat{\Omega}_{jj} - \Omega_{jj}^{(0)}| \leq 2crM_0\,\Omega_{jj}^{(0)} \leq 2cM_0^2 r.$$

Taking the maximum over $1 \leq j \leq p$, we conclude that

$$\max_{1\leq j\leq p} |\widehat{\Omega}_{jj} - \Omega_{jj}^{(0)}| \lesssim M_0^2\sqrt{\frac{s\log p}{n_0}}.$$

□

We are now ready to derive the convergence rates for the off-diagonal entries of the proposed precision matrix estimator. Combining the regression error bound in Lemma 4, the diagonal error bounds in Lemmas 5 and 6, and the similarity relation in Lemma 1, the following two lemmas establish columnwise ℓ_1 convergence rates under the global and local variance estimators, respectively.

Lemma 7. *Under Assumptions 1–5 and the conditions in Lemmas 1, 4 and 5, for the global estimator $\widehat{\tau}_j$, we have*

$$\max_{1\leq j\leq p} \|\widehat{\boldsymbol{\Omega}}_{-j,j} - \boldsymbol{\Omega}_{-j,j}^{(0)}\|_1 \lesssim M_0\left(s^{\frac{1}{2}} + C_1\|\boldsymbol{\Omega}^{(0)}\|_1\right)\left(\frac{s\log p}{N}\right)^{\frac{1}{2}} + M_0 M^{\frac{1}{2}}(sh)^{\frac{1}{2}}\left(\frac{\log p}{N}\right)^{\frac{1}{4}},$$

for some constant C_1, where $h = \max_{1\leq k\leq K}\|\boldsymbol{\Omega}^{(k)} - \boldsymbol{\Omega}^{(0)}\|_1$ and $M = M_0C_0$.

Proof. By the nodewise regression representation,

$$\widehat{\boldsymbol{\Omega}}_{-j,j} = -\widehat{\boldsymbol{\beta}}_j\,\widehat{\Omega}_{jj}, \qquad \boldsymbol{\Omega}_{-j,j}^{(0)} = -\boldsymbol{\beta}_j^{(0)}\,\Omega_{jj}^{(0)}.$$

Hence,

$$\begin{aligned}
\|\widehat{\boldsymbol{\Omega}}_{-j,j} - \boldsymbol{\Omega}^{(0)}_{-j,j}\|_1 &= \| - \widehat{\boldsymbol{\beta}}_j \widehat{\Omega}_{jj} + \boldsymbol{\beta}^{(0)}_j \Omega^{(0)}_{jj}\|_1 \\
&= \|(\widehat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}^{(0)}_j)\widehat{\Omega}_{jj} + \boldsymbol{\beta}^{(0)}_j(\widehat{\Omega}_{jj} - \Omega^{(0)}_{jj})\|_1 \\
&\le \|\widehat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}^{(0)}_j\|_1 \, \widehat{\Omega}_{jj} + |\widehat{\Omega}_{jj} - \Omega^{(0)}_{jj}| \, \|\boldsymbol{\beta}^{(0)}_j\|_1 \\
&= \|\widehat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}^{(0)}_j\|_1 \, \widehat{\Omega}_{jj} + \left| \frac{\widehat{\Omega}_{jj}}{\Omega^{(0)}_{jj}} - 1 \right| \|\boldsymbol{\Omega}^{(0)}_{-j,j}\|_1.
\end{aligned}$$

By Lemma 4, Assumption 4, and Assumption 5,

$$\|\widehat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}^{(0)}_j\|_1 \le (C_{\beta 1}\sqrt{s_b s} + C_{\beta 3} s_b) \left(\frac{\log p}{N}\right)^{1/2} + C_{\beta 2}\sqrt{s_b h_j} \left(\frac{\log p}{N}\right)^{1/4},$$

for some positive constants $C_{\beta 1}$, $C_{\beta 2}$, and $C_{\beta 3}$. By Lemma 5,

$$\left| \frac{\widehat{\Omega}_{jj}}{\Omega^{(0)}_{jj}} - 1 \right| \le 2cM_0 \sqrt{\frac{s_b \log p_b}{N}}, \qquad \widehat{\Omega}_{jj} \le M_0 \left(1 + 2cM_0 \sqrt{\frac{s_b \log p_b}{N}}\right).$$

Together with $\|\boldsymbol{\Omega}^{(0)}_{-j,j}\|_1 \le \|\boldsymbol{\Omega}^{(0)}\|_1$, these bounds yield

$$\begin{aligned}
\|\widehat{\boldsymbol{\Omega}}_{-j,j} - \boldsymbol{\Omega}^{(0)}_{-j,j}\|_1 \le \; & \left(\left(C_{\beta 1}\sqrt{s_b s} + C_{\beta 3} s_b\right) \left(\frac{\log p}{N}\right)^{\frac{1}{2}} + C_{\beta 2}\sqrt{s_b h_j} \left(\frac{\log p}{N}\right)^{\frac{1}{4}} \right) M_0 \left(1 + 2cM_0 \sqrt{\frac{s_b \log p_b}{N}}\right) \\
& + 2cM_0 \sqrt{\frac{s_b \log p_b}{N}} \, \|\boldsymbol{\Omega}^{(0)}\|_1.
\end{aligned}$$

Since K is uniformly bounded and C is fixed, we have

$$s_b = (1 + K + CK)s \lesssim s, \qquad p_b = (K+1)(p-1) \asymp p, \qquad \log p_b \asymp \log p.$$

Moreover, the condition

$$cM_0 \sqrt{\frac{s_b \log p_b}{N}} \le \frac{1}{2}$$

in Lemma 5 implies

$$1 + 2cM_0 \sqrt{\frac{s_b \log p_b}{N}} \le 2.$$

It therefore follows that

$$\|\widehat{\boldsymbol{\Omega}}_{-j,j} - \boldsymbol{\Omega}^{(0)}_{-j,j}\|_1 \lesssim M_0 s \left(\frac{\log p}{N}\right)^{\frac{1}{2}} + M_0 (s h_j)^{\frac{1}{2}} \left(\frac{\log p}{N}\right)^{\frac{1}{4}} + M_0 \|\boldsymbol{\Omega}^{(0)}\|_1 \left(\frac{s \log p}{N}\right)^{\frac{1}{2}}.$$

By Lemma 1, $h_j \le Mh$. Hence,

$$M_0 (s h_j)^{\frac{1}{2}} \left(\frac{\log p}{N}\right)^{\frac{1}{4}} \le M_0 M^{\frac{1}{2}} (sh)^{\frac{1}{2}} \left(\frac{\log p}{N}\right)^{\frac{1}{4}}.$$

Therefore,

$$\|\widehat{\boldsymbol{\Omega}}_{-j,j} - \boldsymbol{\Omega}^{(0)}_{-j,j}\|_1 \lesssim M_0 \left(s^{\frac{1}{2}} + C_1 \|\boldsymbol{\Omega}^{(0)}\|_1\right) \left(\frac{s \log p}{N}\right)^{\frac{1}{2}} + M_0 M^{\frac{1}{2}} (sh)^{\frac{1}{2}} \left(\frac{\log p}{N}\right)^{\frac{1}{4}}.$$

Taking the maximum over $1 \le j \le p$ completes the proof. □

Lemma 8. *Under Assumptions 1–5 and the conditions in Lemmas 1, 4 and 6, for the local estimator $\widehat{\tau}_j$, we have*

$$\max_{1 \le j \le p} \|\widehat{\boldsymbol{\Omega}}_{-j,j} - \boldsymbol{\Omega}^{(0)}_{-j,j}\|_1 \lesssim M_0 s \left(\frac{\log p}{N}\right)^{\frac{1}{2}} + M_0 M^{\frac{1}{2}} (sh)^{\frac{1}{2}} \left(\frac{\log p}{N}\right)^{\frac{1}{4}} + \|\boldsymbol{\Omega}^{(0)}\|_1 M_0 \left(\frac{s \log p}{n_0}\right)^{\frac{1}{2}},$$

Proof. By the nodewise regression representation,

$$\widehat{\boldsymbol{\Omega}}_{-j,j} = -\widehat{\boldsymbol{\beta}}_j \, \widehat{\Omega}_{jj}, \qquad \boldsymbol{\Omega}^{(0)}_{-j,j} = -\boldsymbol{\beta}^{(0)}_j \, \Omega^{(0)}_{jj},$$

and we have

$$\|\widehat{\boldsymbol{\Omega}}_{-j,j} - \boldsymbol{\Omega}^{(0)}_{-j,j}\|_1 \le \|\widehat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}^{(0)}_j\|_1 \, \widehat{\Omega}_{jj} + \left| \frac{\widehat{\Omega}_{jj}}{\Omega^{(0)}_{jj}} - 1 \right| \|\boldsymbol{\Omega}^{(0)}_{-j,j}\|_1.$$

By Lemma 4, Assumption 4, and Assumption 5,

$$\|\widehat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}^{(0)}_j\|_1 \le (C_{\beta 1} \sqrt{s_b s} + C_{\beta 3} s_b) \left(\frac{\log p}{N}\right)^{1/2} + C_{\beta 2} \sqrt{s_b h_j} \left(\frac{\log p}{N}\right)^{1/4},$$

for some positive constants $C_{\beta 1}$, $C_{\beta 2}$, and $C_{\beta 3}$. By Lemma 6,

$$\left| \frac{\widehat{\Omega}_{jj}}{\Omega^{(0)}_{jj}} - 1 \right| \le 2cM_0 \sqrt{\frac{s \log p}{n_0}}, \qquad \widehat{\Omega}_{jj} \le M_0 \left(1 + 2cM_0 \sqrt{\frac{s \log p}{n_0}}\right).$$

With $\|\boldsymbol{\Omega}^{(0)}_{-j,j}\|_1 \le \|\boldsymbol{\Omega}^{(0)}\|_1$, and treating a, L, K, and ζ as constant, we obtain,

$$\begin{aligned}
\|\widehat{\boldsymbol{\Omega}}_{-j,j} - \boldsymbol{\Omega}^{(0)}_{-j,j}\|_1 \lesssim & \left(\left(C_{\beta 1} \sqrt{s_b s} + C_{\beta 3} s_b\right) \left(\frac{\log p}{N}\right)^{\frac{1}{2}} + C_{\beta 2} \sqrt{s_b h_j} \left(\frac{\log p}{N}\right)^{\frac{1}{4}} \right) M_0 \left(1 + 2cM_0 \sqrt{\frac{s \log p}{n_0}}\right) \\
& + 2cM_0 \sqrt{\frac{s \log p}{n_0}} \, \|\boldsymbol{\Omega}^{(0)}\|_1 \\
\lesssim & \, M_0 s \left(\frac{\log p}{N}\right)^{\frac{1}{2}} + M_0 (sh_j)^{\frac{1}{2}} \left(\frac{\log p}{N}\right)^{\frac{1}{4}} + \|\boldsymbol{\Omega}^{(0)}\|_1 M_0 \left(\frac{s \log p}{n_0}\right)^{\frac{1}{2}}.
\end{aligned}$$

Using $h_j \le Mh$ from Lemma 1, we obtain

$$\|\widehat{\boldsymbol{\Omega}}_{-j,j} - \boldsymbol{\Omega}^{(0)}_{-j,j}\|_1 \lesssim M_0 s \left(\frac{\log p}{N}\right)^{\frac{1}{2}} + M_0 M^{\frac{1}{2}} (sh)^{\frac{1}{2}} \left(\frac{\log p}{N}\right)^{\frac{1}{4}} + \|\boldsymbol{\Omega}^{(0)}\|_1 M_0 \left(\frac{s \log p}{n_0}\right)^{\frac{1}{2}}.$$

Since the bound holds uniformly over j, taking the maximum over $1 \le j \le p$ completes the proof.

□

Based on the convergence rate of diagonal elements and off-diagonal elements, we have Theorem 1, the proof of which is detailed below.

Proof of Theorem 1. By definition of the matrix L_1 norm,

$$\begin{aligned}\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_1 &= \max_{1\le j\le p} \|\widehat{\mathbf{\Omega}}_j - \mathbf{\Omega}_j^{(0)}\|_1 \\ &\le \max_{1\le j\le p} \|\widehat{\mathbf{\Omega}}_{-j,j} - \mathbf{\Omega}_{-j,j}^{(0)}\|_1 + \max_{1\le j\le p} |\widehat{\Omega}_{jj} - \Omega_{jj}^{(0)}|.\end{aligned}$$

For the first term, Lemma 7 gives

$$\max_{1\le j\le p} \|\widehat{\mathbf{\Omega}}_{-j,j} - \mathbf{\Omega}_{-j,j}^{(0)}\|_1 \lesssim M_0 \left(s^{\frac{1}{2}} + C_1\|\mathbf{\Omega}^{(0)}\|_1\right)\left(\frac{s\log p}{N}\right)^{\frac{1}{2}} + M_0 M^{\frac{1}{2}}(sh)^{\frac{1}{2}}\left(\frac{\log p}{N}\right)^{\frac{1}{4}}.$$

For the second term, Lemma 5 yields

$$\max_{1\le j\le p} |\widehat{\Omega}_{jj} - \Omega_{jj}^{(0)}| \lesssim M_0^2\sqrt{\frac{s\log p}{N}}.$$

Combining the two bounds, we obtain

$$\begin{aligned}\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_1 \lesssim{}& M_0 \left(s^{\frac{1}{2}} + C_1\|\mathbf{\Omega}^{(0)}\|_1\right)\left(\frac{s\log p}{N}\right)^{\frac{1}{2}} + M_0 M^{\frac{1}{2}}(sh)^{\frac{1}{2}}\left(\frac{\log p}{N}\right)^{\frac{1}{4}} \\ &+ M_0^2\sqrt{\frac{s\log p}{N}}.\end{aligned}$$

Absorbing the last term into the first-order term gives

$$\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_1 \lesssim M_0\Big(s^{1/2} + C_1\|\mathbf{\Omega}^{(0)}\|_1 + C_2 M_0\Big)\sqrt{\frac{s\log p}{N}} + M_0 M^{1/2}(sh)^{1/2}\Big(\frac{\log p}{N}\Big)^{1/4},$$

for some constant C_2.

Finally, since for any symmetric matrix $\mathbf{A}$,

$$\|\mathbf{A}\|_2 \le \|\mathbf{A}\|_1,$$

we have

$$\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_2 \le \|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_1,$$

and therefore the same upper bound also holds for the spectral norm. □

We also have a local-version theorem parallel to Theorem 1.

Theorem 2. *Under Assumptions 1–3, uniformly over $\mathbb{G}(s,h)$, let $\widehat{\mathbf{\Omega}}$ be the Trans-Nodewise estimator computed with $\lambda_{j0} = \lambda_{j1} = \cdots = \lambda_{jK} \asymp \sqrt{\log p/N}$ and the local variance estimator. Further assume that the conditions in Lemmas 1, 4 and 6, hold. Then with probability at least $1 - cp^{-1}$,*

$$\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_1 \lesssim M_0 s\left(\frac{\log p}{N}\right)^{\frac{1}{2}} + M_0 M^{\frac{1}{2}}(sh)^{\frac{1}{2}}\left(\frac{\log p}{N}\right)^{\frac{1}{4}} + \left(\|\mathbf{\Omega}^{(0)}\|_1 + M_0\right) M_0\left(\frac{s\log p}{n_0}\right)^{\frac{1}{2}},$$

where $N = \sum_{k=0}^{K} n_k$, $h = \max_{1\le k\le K} \|\mathbf{\Omega}^{(k)} - \mathbf{\Omega}^{(0)}\|_1$, *and* $M = M_0 C_0$. *Moreover,* $\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_2 \le \|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_1$ *satisfies the same upper bound.*

Proof. By definition of the matrix L_1 norm,

$$\begin{aligned}\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_1 &= \max_{1\le j\le p} \|\widehat{\mathbf{\Omega}}_j - \mathbf{\Omega}_j^{(0)}\|_1 \\ &\le \max_{1\le j\le p} \|\widehat{\mathbf{\Omega}}_{-j,j} - \mathbf{\Omega}_{-j,j}^{(0)}\|_1 + \max_{1\le j\le p} |\widehat{\Omega}_{jj} - \Omega_{jj}^{(0)}|.\end{aligned}$$

For the off-diagonal part, Lemma 8 yields

$$\max_{1\le j\le p} \|\widehat{\mathbf{\Omega}}_{-j,j} - \mathbf{\Omega}_{-j,j}^{(0)}\|_1 \lesssim M_0 s \left(\frac{\log p}{N}\right)^{\frac{1}{2}} + M_0 M^{\frac{1}{2}} (sh)^{\frac{1}{2}} \left(\frac{\log p}{N}\right)^{\frac{1}{4}} + \|\mathbf{\Omega}^{(0)}\|_1 M_0 \left(\frac{s\log p}{n_0}\right)^{\frac{1}{2}}.$$

For the diagonal part, Lemma 6 gives

$$\max_{1\le j\le p} |\widehat{\Omega}_{jj} - \Omega_{jj}^{(0)}| \lesssim M_0^2 \left(\frac{s\log p}{n_0}\right)^{\frac{1}{2}}.$$

Therefore,

$$\begin{aligned}\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_1 \lesssim{}& M_0 s \left(\frac{\log p}{N}\right)^{\frac{1}{2}} + M_0 M^{\frac{1}{2}} (sh)^{\frac{1}{2}} \left(\frac{\log p}{N}\right)^{\frac{1}{4}} \\ &+ \|\mathbf{\Omega}^{(0)}\|_1 M_0 \left(\frac{s\log p}{n_0}\right)^{\frac{1}{2}} + M_0^2 \left(\frac{s\log p}{n_0}\right)^{\frac{1}{2}}.\end{aligned}$$

Since $M_0^2 \le M_0(1 + \|\mathbf{\Omega}^{(0)}\|_1)$ after absorbing constants, the last two terms can be combined, giving

$$\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_1 \lesssim M_0 s \left(\frac{\log p}{N}\right)^{\frac{1}{2}} + M_0 M^{\frac{1}{2}} (sh)^{\frac{1}{2}} \left(\frac{\log p}{N}\right)^{\frac{1}{4}} + \left(\|\mathbf{\Omega}^{(0)}\|_1 + M_0\right) M_0 \left(\frac{s\log p}{n_0}\right)^{\frac{1}{2}}.$$

Finally, since for any symmetric matrix $\mathbf{A}$,

$$\|\mathbf{A}\|_2 \le \|\mathbf{A}\|_1,$$

we have

$$\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_2 \le \|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^{(0)}\|_1,$$

so the same upper bound also holds for the spectral norm. □