Multi-task Learning for Gaussian Graphical Regressions with High Dimensional Covariates

Jingfei Zhang[1], Yi Li[2]

[1] Goizueta Business School, Emory University, Atlanta, GA 30322

[2] Department of Biostatistics, University of Michigan, Ann Arbor, MI 48109

Abstract

Gaussian graphical regression is a powerful approach for regressing the precision matrix of a Gaussian graphical model on covariates, which permits the response variables and covariates to outnumber the sample size. However, traditional approaches of fitting the model via separate node-wise lasso regressions overlook the network-induced structure among these regressions, leading to high error rates, particularly when the number of nodes is large. To address this issue, we propose a multi-task learning estimator for fitting Gaussian graphical regression models, which incorporates a cross-task group sparsity penalty and a within-task element-wise sparsity penalty to govern the sparsity of active covariates and their effects on the graph, respectively. We also develop an efficient augmented Lagrangian algorithm for computation, which solves subproblems with a semi-smooth Newton method. We further prove that our multi-task learning estimator has considerably lower error rates than the separate node-wise regression estimates, as the cross-task penalty enables borrowing information across tasks. We examine the utility of our method through simulations and an application to a gene co-expression network study with brain cancer patients.

Keywords: subject-specific Gaussian graphical model, graphical model with covariates, multi-task learning, concentration inequality for dependent variables, co-expression quantitative trait loci.

1 Introduction

Gaussian graphical models are an effective tool for inferring the dependence among variables of interest, such as the co-expression patterns among genes (Peng et al., 2009; Cai et al., 2012; Chen et al., 2016) and functional connectivity between brain regions (Zhang et al., 2019), because precision matrices for multivariate Gaussian variables have an interpretation of conditional dependence (Lauritzen, 1996). That is, if $(X_1, \ldots, X_p) \sim \mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma})$, then $(\mathbf{\Sigma}^{-1})_{jk} = 0$ implies that X_j and X_k are conditionally independent given all other variables. While the literature on Gaussian graphical models is expanding, most existing models assume a homogeneous population with a common graphical model (Meinshausen and Bühlmann, 2006; Yuan and Lin, 2007; Friedman et al., 2008; Peng et al., 2009) or several stratified graphical models (Guo et al., 2011; Danaher et al., 2014).

In numerous applications, graphical structures may depend on high dimensional external continuous and discrete covariates. For example, the functional connectivity of brain regions can be modulated by individual subject's gender, age and genetic variants (Zhang et al., 2022); genetic variants, clinical and environmental factors, may affect both the expression levels of individual genes and the co-expression relationships among genes (Wang et al., 2012). Finding genetic variants that alter co-expression relationships, commonly referred to as co-expression quantitative trait loci, is of scientific interest (Wang et al., 2012; van der Wijst et al., 2018). The identification of co-expression quantitative trait loci among many SNPs can be framed as a problem that involves linking graphical models with high-dimensional external covariates. An important goal is thus to ascertain how the covariates modulate the individual-level graphical structures, and to recover both the population- and subject-level graphs. This understanding of heterogeneity across individuals is critical for precision health and medicine.

Though much progress has been made for developing graphical models, less has been done for covariate-dependent graphical models. Several works (Li et al., 2012; Cai et al., 2012; Chen et al., 2016) considered covariate-dependent Gaussian graphical models, wherein the mean of the nodes depends on covariates, while the network structure is the same across all of the subjects. Guo et al. (2011) and Danaher et al. (2014) estimated several stratified graphical models by preserving the common structure among them. Liu et al. (2010) partitioned the covariate space into several subspaces and fitted separate Gaussian graphical models for each subspace. Kolar et al. (2010) nonparametrically estimated the dependence of a covariance matrix on one continuous covariate. Cheng et al. (2014) fitted a conditional Ising model for binary data. Ni et al. (2019) proposed a directed acyclic graph model that allows the graph structure to vary with a small number of covariates, and assumed a hierarchical ordering of the nodes. None of these cited works, however, can handle high-dimensional continuous and discrete covariates. More recently, Zhang and Li (2022) proposed a Gaussian graphical regression framework that regresses the precision matrix of a graphical model on high-dimensional covariates and estimates the parameters using separate node-wise regressions. However, this separate regression approach ignored the network-induced common structure among these regressions, potentially leading to large errors, especially with many nodes.

In this work, we address these limitations by making several methodological, computational and theoretical contributions. Regarding the *methodology*, we propose a novel approach within the Gaussian graphical regression framework that addresses the challenge of having a large number of covariates in relation to the sample size. Specifically, we introduce a cross-task group sparsity penalty that allows us to effectively borrow information across different tasks, while accommodating the sparsity assumption on active covariates

with nonzero effects on the graph. Additionally, we incorporate an element-wise sparsity constraint to account for the potential sparse effects of active covariates. To speed up the *computation*, we adapt an efficient inexact augmented Lagrangian algorithm to optimize the cross-task objective function with the combined sparsity penalty. Our algorithm involves invoking a semi-smooth Newton method to solve multiple subproblems; for a graphical model of dimension $p = 50$ and number of covariates $q = 300$ resulting in about 750,000 parameters in total, the joint optimization can be solved in a few seconds on a personal desktop. Our *theoretical* contributions address the main challenge that the regression tasks are entangled in a general dependence structure. We establish a new and sharp tail probability bound for dependent heavy-tailed random variables with an arbitrary dependence structure, which is meritorious even on its own. Moreover, as the combined sparsity penalty is not decomposable, common techniques using decomposable regularizers and null space properties (Negahban et al., 2012) are not applicable. Thus, our techniques may advance high-dimensional regressions with simultaneously sparse structures. Finally, we prove that, compared to Zhang and Li (2022), the error rate of the simultaneously estimated precision parameters improves by a factor of p, the number of response variables. The improvement is remarkable for a large p, as further corroborated in simulation studies.

2 Multi-task Learning for Gaussian Graphical Regressions

2.1 Gaussian Graphical Regressions

Denote by $\boldsymbol{X} = (X_1, \ldots, X_p)^\top$ the p-dimensional vector of response variables, for example, gene expression levels. The standard Gaussian graphical model assumes that

$\boldsymbol{X} \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$, which after writing $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ can be represented as regressions:

$$X_j = \sum_{k \neq j}^{p} \beta_{jk} X_k + \epsilon_j, \quad j \in [p], \tag{1}$$

where $\beta_{jk} = -\boldsymbol{\Omega}_{jk}/\boldsymbol{\Omega}_{jj}$, $\mathrm{Var}(\epsilon_j) = 1/\boldsymbol{\Omega}_{jj}$ and ϵ_j is independent of $\boldsymbol{X}_{-j} = \{X_k : k \neq j \in [p]\}$. As $\boldsymbol{\Omega}_{jk} \neq 0$ is equivalent to X_j and X_k being conditionally dependent given all other variables, estimating the conditional dependence structure reduces to a model selection problem (i.e., finding nonzero β_{jk}'s) under the regression in (1).

Let $\boldsymbol{U} = (U_1, \ldots, U_q)^\top$ be the q-dimensional vector of covariates, such as age, sex and genetic variants. We assume that

$$\boldsymbol{X} \,|\, \boldsymbol{U} = \boldsymbol{u} \sim \mathcal{N}_p(\boldsymbol{\Gamma}\boldsymbol{u}, \boldsymbol{\Sigma}(\boldsymbol{u})),$$

where $\boldsymbol{\Sigma}(\boldsymbol{u})$ is the conditional covariance, and $\boldsymbol{\Omega}(\boldsymbol{u}) = \boldsymbol{\Sigma}^{-1}(\boldsymbol{u})$ is the conditional precision matrix linked to $\boldsymbol{u}$ via

$$-\boldsymbol{\Omega}(\boldsymbol{u})_{jk} = \begin{cases} -\sigma^{jj} & j = k, \\ \beta'_{jk0} + \sum_{h=1}^{q} \beta'_{jkh} u_h & j \neq k, \end{cases} \tag{2}$$

where $\beta'_{jkh} = \beta'_{kjh}$ for al j, k, h; see an illustration in Figure 1. We assume $\boldsymbol{\Omega}(\boldsymbol{u})_{jj} = \sigma^{jj}$ to be free of $\boldsymbol{u}$, and this is discussed shortly after (3) in Remark 1. A result fundamental to our method is that the precision matrix in (2) is in fact connected to an interpretable regression representation, termed *Gaussian graphical regression* (Zhang and Li, 2022). This important result allows us to formulate the problem of estimating conditional dependence as estimating a regression model. After centering $\boldsymbol{Z} = \boldsymbol{X} - \boldsymbol{\Gamma}\boldsymbol{u} = (Z_1, \ldots, Z_p)^\top$, some algebra shows that (2) can be represented as the following regression:

$$Z_j = \sum\nolimits_{k \neq j}^{p} \beta_{jk0} Z_k + \sum\nolimits_{k \neq j}^{p} \sum\nolimits_{h=1}^{q} \underbrace{\beta_{jkh} \times u_h Z_k}_{\text{interaction term}} + \epsilon_j, \tag{3}$$

where $\beta_{jkh} = \beta'_{jkh}/\sigma^{jj}$, ϵ_j is independent of $\boldsymbol{Z}_{-j}$ and $\mathrm{Var}(\epsilon_j) = 1/\sigma^{jj}$, for all j, k and h. When $\beta_{jkh} = 0$, (3) reduces to the usual Gaussian graphical model with $\boldsymbol{Z} \,|\, \boldsymbol{U} = \boldsymbol{u} \sim$

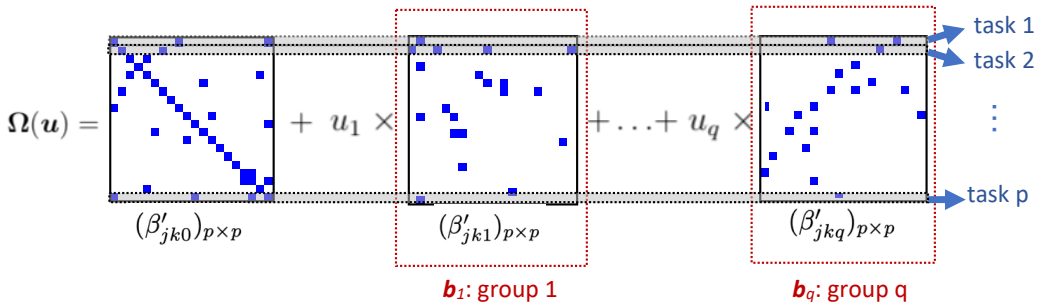

Figure 1: An illustration of multi-task learning in Gaussian graphical regression.

$\mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$. Notably, model (3) provides a regression framework for estimating the precision parameters in (2), by including the interactions between $\boldsymbol{X}_{-j}$ and $\boldsymbol{u}$. Correspondingly, the partial correlation between X_j and X_k is modeled as a function of $\boldsymbol{u}$.

Remark 1. The variance of the error term in (3) can be written as $\text{Var}(\epsilon_j) = 1/\sigma^{jj}$, where σ^{jj} is the diagonal element of $\boldsymbol{\Omega}(\boldsymbol{u})$. Correspondingly, assuming σ^{jj} to be free of $\boldsymbol{u}$ implies the residual variance of Z_j, after accounting for the effects of $\boldsymbol{u}$, $\boldsymbol{Z}_{-j}$ and their interactions, no longer varies with $\boldsymbol{u}$. This assumption is reasonable in regression contexts, as noted by Zhang and Li (2022), and greatly simplifies our methodology. In Section 7, we discuss formulations where σ^{jj} may vary with $\boldsymbol{u}$.

Remark 2. A natural sufficient condition that ensures the positive definiteness of $\boldsymbol{\Omega}(\boldsymbol{u})$'s is diagonal dominance. Given (3), diagonal dominance in $\boldsymbol{\Omega}(\boldsymbol{u})$ implies that $\max(1, \|\boldsymbol{u}\|_\infty)\|\boldsymbol{\beta}_j\|_1 < 1$ where $\boldsymbol{\beta}_j$ is a vector that collects all coefficients in (3). If we restrict $u_h^{(i)}$ to be within [-1,1] (if not, rescale it), we can simplify this sufficient condition to $\|\boldsymbol{\beta}_j\|_1 < 1$, $j \in [p]$. This implies that, to maintain diagonal dominance, magnitudes of the effects of $\boldsymbol{u}$ on partial correlations should not be too large. In finite sample cases and to ensure the positive definiteness of estimated $\boldsymbol{\Omega}(\boldsymbol{u})$'s we can consider a posthoc rescaling step; see Section 3.2 for details.

2.2 A multi-task learning approach

Let $\boldsymbol{\beta}_j = (\beta_{j10}, \ldots, \beta_{j,j-1,0}, \beta_{j,j+1,0}, \ldots, \beta_{jp0}, \ldots, \beta_{j1q}, \ldots, \beta_{j,j-1,q}, \beta_{j,j+1,q}, \ldots, \beta_{jpq})^\top$ be the vector of $(p-1)(q+1)$ coefficients in (3) and write $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \ldots, \boldsymbol{\beta}_p^\top)^\top$. To expose the key ideas, we assume a known $\boldsymbol{\Gamma}$ in the ensuing development, and focus on the estimation of $\boldsymbol{\beta}$; extensions with unknown $\boldsymbol{\Gamma}$ are straightforward, but with more involved notation.

We impose on $\boldsymbol{\beta}$ simultaneous group sparsity and element-wise sparsity, with groups illustrated in (1). First, we assume $\boldsymbol{\beta}$ is *group sparse*, stipulating that effective covariates, that is, those with nonzero effects on edges, are sparse. Importantly, this group sparse penalty is placed across p regressions, allowing us to borrow information across p regression tasks when selecting effective covariates. We further assume $\boldsymbol{\beta}$ is *element-wise sparse*. That is, effective covariates may influence only a few edges. These simultaneous sparsity assumptions are well supported by genetic studies (van der Wijst et al., 2018).

With n independent data $\mathcal{D} = \{(\boldsymbol{u}^{(i)}, \boldsymbol{x}^{(i)}), i \in [n]\}$, let $\boldsymbol{z}^{(i)} = \boldsymbol{x}^{(i)} - \boldsymbol{\Gamma}\boldsymbol{u}^{(i)}$ and $\boldsymbol{w}_{-j}^{(i)} = \boldsymbol{z}_{-j}^{(i)} \otimes \boldsymbol{u}^{(i)}$, where $\otimes$ denotes the Kronecker product. To estimate $\boldsymbol{\beta}$, we consider

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2n} \sum_{j=1}^{p} \sum_{i=1}^{n} (z_j^{(i)} - \boldsymbol{w}_{-j}^{(i)\top} \boldsymbol{\beta}_j)^2 + \lambda_1 \sum_{h=1}^{q} \|\boldsymbol{b}_h\|_2 + \lambda_2 \|\boldsymbol{\beta}\|_1, \tag{4}$$

where $\boldsymbol{b}_h = ((\boldsymbol{\beta}_1)_{(h)}, \ldots, (\boldsymbol{\beta}_p)_{(h)})$ collects all coefficients related to u_h with $(\boldsymbol{\beta}_j)_{(h)} = (\beta_{j1h}, \ldots, \beta_{j,j-1,h}, \beta_{j,j+1,h}, \ldots, \beta_{jph})$ and $\lambda_1, \lambda_2 \geq 0$ are tuning parameters. Distinguishing from the node-wise estimation in Zhang and Li (2022), (4) considers p graphical regressions simultaneously, with the group lasso penalty regulating $\boldsymbol{b}_h$ cross tasks; see Figure 1. The convex regularizing terms, $\lambda_1 \sum_{h=1}^{q} \|\boldsymbol{b}_h\|_2$ and $\lambda_2 \|\boldsymbol{\beta}\|_1$, encourage group- and element-wise sparsity, respectively, though the group sparse penalty is not applied to $\boldsymbol{b}_0$, the intercept coefficients, as it determines the population level regulatory network.

3 An Inexact Augmented Lagrangian Algorithm

The combined sparsity penalty in (4) is referred to as the *sparse group lasso* penalty (Simon et al., 2013). Our sparse group lasso optimization problem is challenging as there are an order of p^2q parameters in (4). For example, when $p = 100$ and $q = 100$, nearly 1,000,000 parameters need to be optimized. Existing algorithmic solutions to the sparse group lasso problem include, for example, gradient descent methods (e.g., Liu et al., 2009; Simon et al., 2013), block coordinate descent methods (e.g., Li et al., 2015), alternating direction method of multipliers (ADMM) (e.g., Boyd et al., 2011). These algorithms are first-order methods applied directly to the primal problem. There is another line of research on group lasso with an overlapping or hierarchical group structure (Jenatton et al., 2011; Yuan et al., 2011; Yan and Bien, 2017; Yu and Bien, 2017; Won et al., 2019; Qi and Li, 2022), these methods can be applied to solve the sparse group lasso problem but may not be computationally efficient due to the large number of $q + p$ overlapping groups. In our approach, we deal with the dual problem of (4), which naturally leads to an augmented Lagrangian algorithm (Hestenes, 1969), in conjunction with a semi-smooth Newton method (Kummer, 1988; Zhang et al., 2020). Compared with these first order methods, the semi-smooth Newton augmented Lagrangian method is more computationally efficient by exploiting the second order information while leveraging sparsity in the generalized Jacobian, and requires fewer iterations to converge (Li et al., 2018a,b; Zhang et al., 2020). We also compare our algorithm with an efficient accelerated proximal gradient descent method SLEP (Liu et al., 2009), and obtained favorable results, as reported in Section 5. We present our algorithm as follows.

Write $\mathcal{W} = \begin{pmatrix} \boldsymbol{W}_{-1} & \cdots & \mathbf{0}_{n\times(p-1)(q+1)} \\ \vdots & \ddots & \vdots \\ \mathbf{0}_{n\times(p-1)(q+1)} & \cdots & \boldsymbol{W}_{-p} \end{pmatrix}$, where $\boldsymbol{W}_{-j} = [\boldsymbol{w}_{-j}^{(1)}; \ldots; \boldsymbol{w}_{-j}^{(n)}]$, and

$\boldsymbol{y} = (z_1^{(1)}, \ldots, z_1^{(n)}, z_2^{(1)}, \ldots, z_p^{(n)}) \in \mathbb{R}^{np}$. Correspondingly, (4) can be written as

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2n} \|\mathcal{W}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \lambda_1 \sum_{h=1}^{q} \|\boldsymbol{b}_h\|_2 + \lambda_2 \|\boldsymbol{\beta}\|_1, \tag{5}$$

and its dual problem takes a form as specified in Zhang et al. (2020):

$$\begin{aligned} &\min_{\boldsymbol{a},\boldsymbol{g}} \ \langle \boldsymbol{y}, \boldsymbol{a} \rangle + \frac{1}{2}\|\boldsymbol{a}\|_2^2 + h^*(\boldsymbol{g}) \\ &\text{s.t. } \mathcal{W}^\top \boldsymbol{a} + \boldsymbol{g} = 0, \end{aligned} \tag{6}$$

where $\langle \cdot, \cdot \rangle$ denotes vector inner product and $h^*(\cdot)$ is the Fenchel conjugate function of $h(\boldsymbol{\beta}) := \lambda_1 \sum_{h=1}^{q} \|\boldsymbol{b}_h\|_2 + \lambda_2 \|\boldsymbol{\beta}\|_1$, defined as $h^*(\boldsymbol{g}) = \sup_{\boldsymbol{\beta}} \{\langle \boldsymbol{\beta}, \boldsymbol{g} \rangle - h(\boldsymbol{\beta})\}$. We augment the Lagrangian objective for solving (6) with

$$\langle \boldsymbol{y}, \boldsymbol{a} \rangle + \frac{1}{2}\|\boldsymbol{a}\|_2^2 + h^*(\boldsymbol{g}) + \boldsymbol{v}^\top (\mathcal{W}^\top \boldsymbol{a} + \boldsymbol{g}) + \frac{\tau}{2}\|\mathcal{W}^\top \boldsymbol{a} + \boldsymbol{g}\|_2^2, \tag{7}$$

where $\boldsymbol{v}$ is the Lagrangian multiplier corresponding to the constraint of (6) and the quadratic term $\frac{\tau}{2}\|\mathcal{W}^\top \boldsymbol{a} + \boldsymbol{g}\|_2^2$, with $\tau > 0$, is the augmentation. The augmented term does not change the original problem (6), because it is zero when satisfying the constraint. But it does make the new objective strongly convex when τ is large (Hestenes, 1969). Next, (7) can be rewritten as

$$\mathcal{L}_\tau(\boldsymbol{a}, \boldsymbol{g}; \boldsymbol{v}) = \langle \boldsymbol{y}, \boldsymbol{a} \rangle + \frac{1}{2}\|\boldsymbol{a}\|_2^2 + h^*(\boldsymbol{g}) + \frac{\tau}{2}\|\mathcal{W}^\top \boldsymbol{a} + \boldsymbol{g} - \tau^{-1}\boldsymbol{v}\|_2^2 - \frac{1}{2\tau}\|\boldsymbol{v}\|_2^2. \tag{8}$$

The augmented Lagrangian then proceeds to iteratively update $(\boldsymbol{a}, \boldsymbol{g})$ and $\boldsymbol{v}$, with the most challenging step of finding $\min_{\boldsymbol{a},\boldsymbol{g}} \mathcal{L}_\tau(\boldsymbol{a}, \boldsymbol{g}; \boldsymbol{v})$ given $\boldsymbol{v}$ and τ. To proceed, we first define the proximal mapping of any function $f(\cdot)$ to be

$$\text{Prox}_f(\boldsymbol{u}) = \arg\min_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{u}\|_2^2 \right\},$$

and it follows that, with $\boldsymbol{a}$ fixed,

$$\arg\min_{\boldsymbol{g}} \mathcal{L}_\tau(\boldsymbol{a}, \boldsymbol{g}; \boldsymbol{v}) = \text{Prox}_{h^*/\tau}(\tau^{-1}\boldsymbol{v} - \mathcal{W}^\top \boldsymbol{a}). \tag{9}$$

Algorithm 1 An augmented Lagrangian method for solving (4)

Let $\tau_0 > 0$ be given, and choose $(\boldsymbol{a}_0, \boldsymbol{g}_0, \boldsymbol{v}_0)$. Iterate the following steps for $k = 0, 1, \ldots,$ until convergence.

Step 1: compute $\boldsymbol{a}^{k+1} = \arg\min_{\boldsymbol{a}} \psi_{\tau_k}(\boldsymbol{a}; \boldsymbol{v}^k)$ and $\boldsymbol{g}^{k+1} = \mathrm{Prox}_{h^*/\tau_k}(\tau_k^{-1}\boldsymbol{v}^k - \mathcal{W}^\top \boldsymbol{a}^{k+1})$;

Step 2: compute $\boldsymbol{v}^{k+1} = \boldsymbol{v}^k - \tau_k(\mathcal{W}^\top \boldsymbol{a}^{k+1} + \boldsymbol{g}^{k+1})$;

Step 3: update τ_{k+1} such that $\tau_{k+1} > \tau_k$.

Then, with $\boldsymbol{g}$ replaced by (9) in $\mathcal{L}_\tau(\boldsymbol{a}, \boldsymbol{g}; \boldsymbol{v})$, coupled with the Moreau identity that $\mathrm{Prox}_{th}(u) + t\mathrm{Prox}_{h^*/t}(u/t) = u$, we define a subproblem of $\min_{\boldsymbol{a}} \psi_\tau(\boldsymbol{a}; \boldsymbol{v})$, where $\psi_\tau(\boldsymbol{a}; \boldsymbol{v})$ is defined as

$$\langle \boldsymbol{y}, \boldsymbol{a} \rangle + \frac{1}{2}\|\boldsymbol{a}\|_2^2 + h^*\{\mathrm{Prox}_{h^*/\tau}(\tau^{-1}\boldsymbol{v} - \mathcal{W}^\top \boldsymbol{a})\} + \frac{\tau}{2}\|\mathrm{Prox}_h(\tau^{-1}\boldsymbol{v} - \mathcal{W}^\top \boldsymbol{a})\|_2^2 - \frac{1}{2\tau}\|\boldsymbol{v}\|_2^2.$$

Correspondingly, the augmented Lagrangian is solved via iterations as summarized in Algorithm 1. We choose $\tau_{k+1} = 3\tau_k$ in Step 3 as suggested in Zhang et al. (2020), and discuss the handling of $\mathcal{W}$ in Section A1.1 of the supplement.

A main challenge of executing Algorithm 1, however, lies in solving $\min_{\boldsymbol{a}} \psi_{\tau_k}(\boldsymbol{a}; \boldsymbol{v}^k)$ in Step 1. We address this issue as follows. Because $\psi_{\tau_k}(\boldsymbol{a}; \boldsymbol{v}^k)$ is strongly convex and continuously differentiable with

$$\nabla_{\boldsymbol{a}} \psi_{\tau_k}(\boldsymbol{a}; \boldsymbol{v}^k) = \boldsymbol{y} + \boldsymbol{a} - \tau_k \mathcal{W}\mathrm{Prox}_h(\tau^{-1}\boldsymbol{v}^k - \mathcal{W}^\top \boldsymbol{a}),$$

the solution to $\min_{\boldsymbol{a}} \psi_{\tau_k}(\boldsymbol{a}; \boldsymbol{v}^k)$ can be obtained by solving $\nabla_{\boldsymbol{a}} \psi_{\tau_k}(\boldsymbol{a}; \boldsymbol{v}^k) = \boldsymbol{0}$. Although $\nabla_{\boldsymbol{a}} \psi_{\tau_k}(\boldsymbol{a}; \boldsymbol{v}^k)$ is non-smooth, it is semi-smooth (Kummer, 1988) with respect to its Clarke generalized Jacobian (Clarke, 1990), a generalization of the Jacobian for a smooth function to non-smooth functions; see Section A1.2 in the supplement for the semi-smoothness of $\nabla_{\boldsymbol{a}} \psi_{\tau_k}(\boldsymbol{a}; \boldsymbol{v}^k)$. We adopt the semi-smooth Newton method (Mifflin, 1977), a generalization of the Newton method by using the generalized Jacobian, to solve $\nabla_{\boldsymbol{a}} \psi_{\tau_k}(\boldsymbol{a}; \boldsymbol{v}^k) = \boldsymbol{0}$, as detailed in Section A1.2 of the supplement.

3.1 Convergence of Algorithm 1

The following theorem states the conditions needed for Theorem 1 to converge. See its proof and discussions on the convergence of inexact augmented Lagrangian algorithms in Section A1.3 of the supplement.

Theorem 1 *Let* $\{\boldsymbol{a}^k, \boldsymbol{g}^k, \boldsymbol{v}^k\}$ *be a sequence of estimates generated by Algorithm 1 with the stopping criterion of*

$$\|\nabla\psi_{\tau_k}(\boldsymbol{a}^{k+1};\boldsymbol{v}^k)\|_2 \le e_k/\sqrt{2\tau_k}, \quad \sum_{k=0}^{\infty} e_k < \infty.$$

Then $\boldsymbol{v}^k$ *converges to the optimal solution of the primal problem in (5) and* $\{\boldsymbol{a}^k, \boldsymbol{g}^k\}$ *converges to the optimal solution of the dual problem in (6). If, additionally, the following stopping criterion is also met:*

$$\|\nabla\psi_{\tau_k}(\boldsymbol{a}^{k+1};\boldsymbol{v}^k)\|_2 \le (\eta_k/\sqrt{2\tau_k})\|\boldsymbol{v}^{k+1}-\boldsymbol{v}^k\|_2, \quad \sum_{k=0}^{\infty}\eta_k < \infty,$$

then $\boldsymbol{v}^k$ *and* $\{\boldsymbol{a}^k, \boldsymbol{g}^k\}$ *converge to the optimal solutions of (5) and (6), respectively, with linear convergence rates when* k *is sufficiently large and both linear rates go to 0 as* τ_k *increases to* $+\infty$.

The theorem guarantees the convergence of Algorithm 1 if the subproblem $\nabla_{\boldsymbol{a}}\psi_{\tau_k}(\boldsymbol{a};\boldsymbol{v}_k) = \boldsymbol{0}$ can be solved with sufficient accuracy. The theoretical results on SSN algorithm in Li et al. (2018b) (Theorem 3) ensures that the SSN iterates in Algorithm 1 converge to the solution of $\nabla_{\boldsymbol{a}}\psi_{\tau_k}(\boldsymbol{a};\boldsymbol{v}_k) = \boldsymbol{0}$ at a superlinear rate. This, combined with Theorem 1, ensures the convergence of Algorithm 1.

3.2 Tuning and post-hoc processing

We select λ_1 and λ_2 in (4) jointly via L-fold, for example, $L = 5$, cross validation. Rewrite $\lambda_1 = (1-\alpha)\lambda_0$ and $\lambda_2 = \alpha\lambda_0$, where α reflects the weight of the lasso penalty relative to

the group lasso penalty and λ_0 reflects the total amount of regularization. We assess a set of values for $\alpha \in [0, 1]$; for each α, a grid of λ_0 values are considered in cross validation. The search can be easily parallelized for different α's. To ensure symmetry of the estimated $\boldsymbol{\Omega}(\boldsymbol{u})$, we propose a post-processing step as done in Meinshausen and Bühlmann (2006) and Cheng et al. (2014). Specifically, denote by $\hat{\beta}^0_{jkh} = -\hat{\sigma}^{jj}\hat{\beta}_{jkh}$, where $\hat{\beta}_{jkh}$ is estimated from (4) and $\hat{\sigma}^{jj}$ from the residuals of (4) for all j, k and h. We enforce symmetry by setting $\beta'_{jkh} = \beta'_{hjk} = \hat{\beta}^0_{jkh} 1_{\{|\hat{\beta}^0_{jkh}| < |\hat{\beta}^0_{kjh}|\}} + \hat{\beta}^0_{kjh} 1_{\{|\hat{\beta}^0_{jkh}| > |\hat{\beta}^0_{kjh}|\}}$.

In finite sample cases, we further adopt a post-hoc re-scaling step to ensure the positive definiteness of the final estimator. Assuming the true precision matrix is diagonal dominant and the covariates are within known ranges, the re-scaling step gives the same estimator asymptotically as guaranteed by Theorem 3. Specifically, without loss of generality, assume $u_h \in [0, 1]$ (if not, rescale u_h first). For any j such that $\|\hat{\boldsymbol{\beta}}'_j\|_1 > \hat{\sigma}^{jj}$, we set the final estimate of $\boldsymbol{\beta}_j$ to $\hat{\boldsymbol{\beta}}_j / \|\hat{\boldsymbol{\beta}}_j\|_1$. Note that this rescaling step does not alter the sparsity pattern in the estimated parameters. See simulation results in Section A2.3 of the supplement that compare the estimators with and without this step.

4 Theoretical Results

We establish the non-asymptotic ℓ_2 error rate of the sparse group lasso estimator from (4). One main challenge is that the p tasks in (4) are involved in a complicated dependence structure and this differs from the usual multi-task learning with group sparsity (Lounici et al., 2011), where the tasks are independent. To address this, we have made a key advance in Theorem 2 that gives a new tail bound for the sum of dependent heavy-tailed, such as sub-exponential, variables with an arbitrary dependence structure.

There are also a few other challenges. First, because the design matrix in (4) includes

high-dimensional interactions between $\boldsymbol{z}^{(i)}$ and $\boldsymbol{u}^{(i)}$, and the variance of $\boldsymbol{z}^{(i)}$ is a function of $\boldsymbol{u}^{(i)}$, characterizing the joint distribution of each row in $\boldsymbol{W}_{-j}$ is difficult and requires a delicate treatment. Second, as the combined penalty term $\lambda_1 \sum_{h=1}^{q} \|\boldsymbol{b}_h\|_2 + \lambda_2 \|\boldsymbol{\beta}\|_1$ is not decomposable, the classic techniques for decomposable regularizers and null space (Negahban et al., 2012) are not applicable. By utilizing a novel concentration inequality for the sum of dependent variables in (4), we derive two interrelated bounds for the stochastic term, whose combination yields a sharp upper bound of the stochastic term. We, therefore, show that our proposed estimator can have an improved ℓ_2 error bounds compared to the lasso and the group lasso when the true coefficients are simultaneously sparse, and, more importantly, the error rate of the multi-task estimates improves by a factor of p, compared to the separate node-wise estimates obtained by Zhang and Li (2022).

4.1 New concentration inequality for sum of dependent variables

The concentration results for dependent random variables are often derived under specific dependence structures, such as weak dependence (Merlevède et al., 2011) and asymptotic independence (Ko and Tang, 2008). These structures are unlikely to be applicable to our setting because the error terms in (4) from the p tasks, $\epsilon_1, \ldots, \epsilon_p$, depend on, for example, the expressions of p genes, and are dependent via a complicated co-expression network. To bound $\sum_j \epsilon_j$, we employ a novel idea that partitions the index set of p response variables into mutually exclusive subsets such that the variables within each subset are independent. This can be done by exploring the topology of a graph and solving a vertex coloring problem (Lewis, 2015); see the proof of Theorem 2 in the supplement. With that, we present a concentration inequality result under a general dependence structure.

Theorem 2 *Consider N dependent mean zero sub-exponential random variables Y_j, $j \in$*

$[N]$ *and an induced network* $G(V, E)$ *with a node set* $V = \{1, \ldots, N\}$ *and an edge set* $E = \{(j, k) : Y_j \not\perp Y_k\}$. *Denote the maximum node degree of* $G(V, E)$ *by* $d_{\max}$ *and let* $c_G = \min\left(d_{\max} + 1, \frac{1+\sqrt{8|E|+1}}{2}\right)$. *For any* $t \geq 0$ *and a constant* $c > 0$, *it holds that*

$$\mathbb{P}\left(\sum_{j=1}^{N} Y_j \geq t\right) \leq c_G \exp\left[-c \min\left\{\frac{t^2}{c_G^2 \sum_{j=1}^{N} \|Y_i\|_{\psi_1}^2}, \frac{t}{c_G \max_j \|Y_i\|_{\psi_1}}\right\}\right],$$

where $\|\cdot\|_{\psi_1}$ *is the sub-exponential norm,* $\|X_i\|_{\psi_1} = \sup_{p \geq 1} p^{-1}(E|X_i|^p)^{1/p}$.

The results are inclusive. For example, when $Y_1, \ldots, Y_N$ are mutually independent, we have $c_G = 1$, and the inequality reduces to the usual form for independent sub-exponential random variables (Vershynin, 2010); if the variables are dependent with $c_G = O(1)$, we obtain a tail probability in the same order as when the variables are independent. The theorem leads to a corollary (see Corollary 1 in the supplement) that gives a sharp bound on the sum of dependent chi-squared random variables, which is critical for our proof.

4.2 Error rate analysis

Let $\mathcal{S}$ be the element-wise support set of $\boldsymbol{\beta}$ and $\mathcal{G}$ be the group-wise support set of $\boldsymbol{\beta}$ such that $\mathcal{G} = \{h : \boldsymbol{b}_h \neq \mathbf{0}, h \in [q]\}$, and denote by $s_e = |\mathcal{S}|$ and $s_g = |\mathcal{G}|$, that is, s_e and s_g are the numbers of nonzero entries and nonzero groups, respectively. Without loss of generality, we assume $\sigma^{jj} = 1$. We state the needed regularity conditions.

Assumption 1 *Suppose* $\boldsymbol{u}^{(i)}$ *are independently and identically distributed mean zero random vectors with a covariance matrix satisfying* $\lambda_{\min}(Cov(\boldsymbol{u}^{(i)})) \geq 1/\phi_0$ *for some constant* $\phi_0 > 0$. *Moreover, there exists a constant* $M > 0$ *such that* $|u_h^{(i)}| \leq M$ *for all* i *and* h.

Assumption 2 *Suppose* $\phi_1 \leq \lambda_{min}(Cov(\boldsymbol{z}^{(i)})) \leq \lambda_{max}(Cov(\boldsymbol{z}^{(i)})) \leq \phi_2$ *for some constants* $\phi_1, \phi_2 > 0$.

Assumption 3 *The dimensions p, q and sparsity s_e satisfy $\log p + \log q = \mathcal{O}(n^\delta)$ and $s_e = o(n^\delta)$ for $\delta \in [0, 1/6]$. The maximum column ℓ_0 norm of $\boldsymbol{\Omega}(\boldsymbol{u})$ is bounded above by a positive constant d_0.*

Assumption 1 assumes that $\boldsymbol{u}^{(i)}$'s are element-wise bounded, which is needed in characterizing the distribution of each row in $\mathcal{W}$. This condition is not restrictive as genetic variants are often coded to be $\{0, 1\}$ or $\{0, 1, 2\}$ (Chen et al., 2016). Assumptions 1 and 2 impose bounded eigenvalues on $\mathrm{Cov}(\boldsymbol{u}^{(i)})$ and $\mathrm{Cov}(\boldsymbol{z}^{(i)})$. Assumption 3 is a sparsity condition, useful in establishing a restricted eigenvalue condition (Bickel et al., 2009) for $\mathcal{W}^\top \mathcal{W}/n$.

Let s_λ denote the number of nonzero entries in a candidate model such that $s_e < s_\lambda \leq n$. In parameter tuning, given an s_λ satisfying the conditions in Theorem 3, we choose the range of λ_0 for each α, respectively corresponding to an empty model with no variables selected and a sparse model with s_λ variables selected.

Theorem 3 *Suppose that Assumptions 1-3 hold, $s_\lambda(\log p + \log q) = \mathcal{O}(\sqrt{n})$ and $n \geq A_1\{s_g \log(eq/s_g) + s_e \log(ep)\}$ for some constant $A_1 > 0$. Then the sparse group lasso estimator $\hat{\boldsymbol{\beta}}$ in (4) with*

$$\lambda_1 = C\sqrt{\log(eq/s_g)/n + 2s_e \log(ep)/(ns_g)}, \quad \lambda_2 = \sqrt{s_g/s_e}\lambda_1 \tag{10}$$

satisfies, with probability at least $1 - C_1 \exp[-C_2\{s_g \log(eq/s_g) + s_e \log(ep)\}]$,

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2 \precsim \frac{1}{n}\{s_g \log(eq/s_g) + s_e \log(ep)\}, \tag{11}$$

where C, C_1, and C_2 are positive constants.

In Theorem 3, the condition $s_\lambda(\log p + \log q) = \mathcal{O}(\sqrt{n})$ upper bounds the number of nonzero entries in $\hat{\boldsymbol{\beta}}$, which in turn helps to bound the stochastic term $\langle \boldsymbol{\epsilon}, \mathcal{W}\Delta \rangle$ in the proof. Theorem 3 also shows that our proposed estimator enjoys an improved ℓ_2 error

bound over estimators with only a lasso or a group lasso penalty on $\boldsymbol{\beta}$. Specifically, given that the dimension of $\boldsymbol{\beta}$ is $p(p-1)(q+1)$ and $s_g \leq s_e$, applying the standard lasso regularizer $\lambda\|\boldsymbol{\beta}\|_1$ alone would yield an error bound of $(s_e/n)\log(pq)$ (Negahban et al., 2012), which is slower than that in (11) when $\log p/\log q = o(1)$ and $s_g/s_e = o(1)$. If we utilize a group lasso regularizer $\lambda_g\|\boldsymbol{\beta}\|_{1,2}$ that includes $\boldsymbol{b}_0$, the estimator would have an ℓ_2 error bound of $(s_g/n)\log q + (s_g/n)p(p-1)$ (Lounici et al., 2011), which is slower than that in (11) when $\log q/\{p(p-1)\} = o(1)$ and $s_e/s_g = o(p(p-1)/\log p)$. Notably, the separate node-wise regressions considered in Zhang and Li (2022) yield an error rate of $\frac{1}{n}\{ps_g\log(eq/s_g) + s_e\log(ep)\}$, slower than that in (11) by a factor p if $s_e\log p = O(s_g\log q)$; as p often far exceeds n, the improvement with the multi-task learning approach can be considerable.

5 Numerical Experiments

We compare the finite sample performance of our proposed method defined in (4) (referred to as `MtRegGMM`), with those of three competing solutions, namely, a benchmarking standard Gaussian graphical model estimated by the neighborhood selection method (Meinshausen and Bühlmann, 2006) (`IID`), a lasso estimator ($\texttt{Joint}_{\texttt{lasso}}$) with

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2n}\sum_{j=1}^{p}\sum_{i=1}^{n}(z_j^{(i)} - \boldsymbol{W}_{-j}^{(i)\top}\boldsymbol{\beta}_j)^2 + \lambda\|\boldsymbol{\beta}\|_1,$$

and the separate regressions (Zhang and Li, 2022) (`RegGMM`). For computational feasibility, the $\texttt{Joint}_{\texttt{lasso}}$ is estimated with the method in Section 3 by setting $\lambda_1 = 0$, which is the same algorithm as proposed in Li et al. (2018a). We have also considered the joint group lasso estimator, and reported the results in Section A2.4 of the supplement.

We simulate n samples $\{(\boldsymbol{u}^{(i)}, \boldsymbol{x}^{(i)}), i \in [n]\}$ from (2), with $\boldsymbol{x}^{(i)} \in \mathbb{R}^p$, such as genes, and external covariate $\boldsymbol{u}^{(i)} \in \mathbb{R}^q$, such as SNPs. The $\boldsymbol{u}_j^{(i)}$'s are generated independently

Table 1: Estimation accuracy of $\boldsymbol{\beta}$ and $\boldsymbol{\Omega}$ in simulations with varying sample size n, network size p and covariate dimension q. TPR and FPR represent the true and false positive rates, respectively.

n	p, q	Method	$\text{TPR}_{\boldsymbol{\beta}}$	$\text{FPR}_{\boldsymbol{\beta}}$	Error of $\boldsymbol{\beta}$	Error of $\boldsymbol{\Omega}$
100	$p = 20$ $q = 100$	`MtRegGMM`	**0.985** (0.013)	**0.0001** (0.0000)	**0.534** (0.032)	**0.405** (0.046)
		$\texttt{Joint}_{\texttt{lasso}}$	0.942 (0.015)	0.0003 (0.0000)	0.762 (0.033)	0.712 (0.067)
		`RegGMM`	0.922 (0.020)	0.0013 (0.0001)	1.146 (0.045)	1.984 (0.153)
		`IID`	-	-	-	1.764 (0.034)
	$p = 20$ $q = 200$	`MtRegGMM`	**0.983** (0.021)	**0.0001** (0.0000)	**0.549** (0.037)	**0.436** (0.060)
		$\texttt{Joint}_{\texttt{lasso}}$	0.918 (0.030)	0.0002 (0.0000)	0.848 (0.040)	0.912 (0.088)
		`RegGMM`	0.923 (0.016)	0.0009 (0.0001)	1.207 (0.036)	2.223 (0.199)
		`IID`	-	-	-	1.777 (0.026)
	$p = 100$ $q = 200$	`MtRegGMM`	**0.964** (0.013)	**0.0000** (0.0000)	**0.612 (0.034)**	**0.489** (0.055)
		$\texttt{Joint}_{\texttt{lasso}}$	0.896 (0.020)	**0.0000** (0.0000)	0.846 (0.038)	0.786 (0.069)
		`RegGMM`	0.918 (0.020)	0.0003 (0.0000)	2.620 (0.060)	5.845 (0.329)
		`IID`	-	-	-	5.119 (0.890)
200	$p = 20$ $q = 100$	`MtRegGMM`	**1.000** (0.000)	**0.0000** (0.0000)	**0.251** (0.013)	**0.099** (0.011)
		$\texttt{Joint}_{\texttt{lasso}}$	0.996 (0.003)	0.0000 (0.0000)	0.296 (0.019)	0.133 (0.017)
		`RegGMM`	0.996 (0.003)	0.0006 (0.0001)	0.584 (0.033)	1.111 (0.538)
		`IID`	-	-	-	1.637 (0.020)
	$p = 20$ $q = 200$	`MtRegGMM`	**1.000** (0.000)	**0.0000** (0.0000)	**0.251** (0.008)	**0.089** (0.006)
		$\texttt{Joint}_{\texttt{lasso}}$	0.998 (0.002)	**0.0000** (0.0000)	0.346 (0.018)	0.158 (0.015)
		`RegGMM`	0.998 (0.002)	0.0003 (0.0000)	0.626 (0.029)	0.717 (0.124)
		`IID`	-	-	-	1.642 (0.016)
	$p = 100$ $q = 200$	`MtRegGMM`	0.994 (0.006)	**0.0000** (0.0000)	**0.266 (0.039)**	**0.123** (0.039)
		$\texttt{Joint}_{\texttt{lasso}}$	**1.000** (0.000)	**0.0000** (0.0000)	0.385 (0.007)	0.280 (0.102)
		`RegGMM`	**1.000** (0.000)	0.0001 (0.0000)	1.530 (0.051)	5.636 (2.322)
		`IID`	-	-	-	2.087 (0.058)

and identically from Bernoulli(0.5). Details for generating $\boldsymbol{\Sigma}(\boldsymbol{u}^{(i)})$'s are collected in Section A2.1 of the supplement. For each simulation configuration, we generate 50 independent data sets; given $\boldsymbol{u}^{(i)}$, we determine $\boldsymbol{\Omega}(\boldsymbol{u}^{(i)})$ and $\boldsymbol{\Sigma}(\boldsymbol{u}^{(i)})$, and generate the ith sample $\boldsymbol{x}^{(i)}$ from $\mathcal{N}(\boldsymbol{\Gamma}\boldsymbol{u}^{(i)}, \boldsymbol{\Sigma}(\boldsymbol{u}^{(i)}))$, $i \in [n]$. For a fair comparison, tuning parameters in all of the methods are selected via 5-fold cross validation.

To evaluate the estimation accuracy, we report in Table 1 the estimation errors $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$. For selection accuracy, we report the true positive and false positive rates. Also reported is the average estimation error of the precision matrix defined to be $\sum_{i=1}^{n} \|\hat{\boldsymbol{\Omega}}_i - \boldsymbol{\Omega}_i\|_{F,\text{off}}^2/n$. The proposed `MtRegGMM` outperforms the alternative methods in estimation and selection accuracy for various n, p and q. The estimation errors of `MtRegGMM` increases with p and q

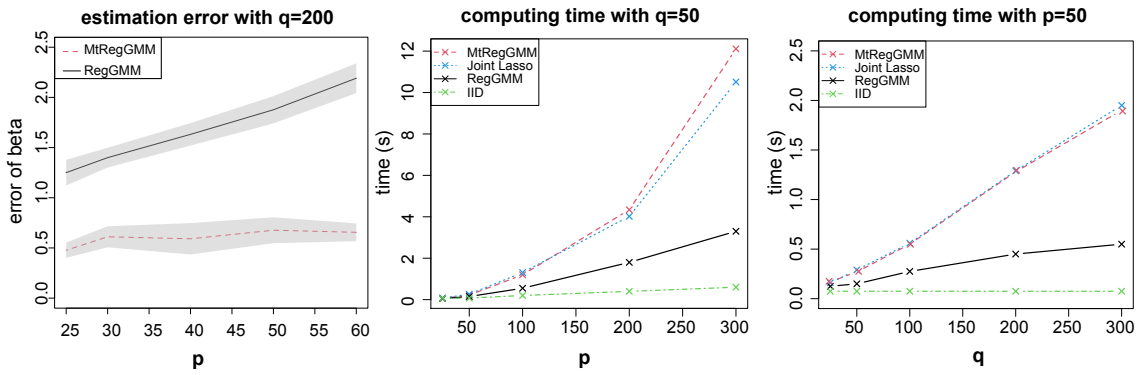

Figure 2: Estimation errors and computing time.

and decreases with n, confirming the results of Theorem 3. It seen that the error of `RegGMM` scales roughly with p while the error of `MtRegGMM` remains relatively stable as p increases, again confirming Theorem 3. The left panel of Figure 2 shows additional estimation error comparison of `RegGMM` and `MtRegGMM` across varying p. We also assess the computation cost. Figure 2 show the computation time of all methods given tuning parameter values. The simulations were run on an iMac with a 3.6 GHz Intel Core i9 processor. Because the number of parameters is in the order of p^2q, the total computing cost is expected to be in the order of p^2q, as seen in Figure 2. It is seen that the computation cost of `MtRegGMM` is on par with that of $\texttt{Joint}_{\texttt{lasso}}$.

We also consider simulations with both discrete and continuous covariates. Specifically, we set half of the covariates $(q/2)$ to be discrete and the other half $(q/2)$ to be continuous. Discrete covariates are generated independently from Bernoulli(0.5) and continuous covariates are generated independently from Uniform$[0, 1]$. Other settings remain the same as in Table 1. Table 2 reports the average criteria with standard errors in the parentheses. It is seen that `MtRegGMM` achieves the best performance in both selection and estimation accuracy. Simulation results with $p > n$ and $p > q$ are reported in A2.2 of the supplement.

Finally, we compare with `SLEP` (Liu et al., 2009), a popular accelerated proximal

Table 2: Estimation accuracy of $\boldsymbol{\beta}$ and $\boldsymbol{\Omega}$ in simulations with varying p, q while $n = 100$. TPR and FPR represent the true and false positive rates, respectively.

p, q	Method	TPR$_{\boldsymbol{\beta}}$	FPR$_{\boldsymbol{\beta}}$	Error of $\boldsymbol{\beta}$	Error of $\boldsymbol{\Omega}$
$p = 20$ $q = 100$	`MtRegGMM`	**0.951** (0.022)	**0.0003** (0.0000)	**0.612** (0.036)	**0.471** (0.042)
	`Joint`$_{\texttt{lasso}}$	0.909 (0.025)	0.0005 (0.0001)	0.824 (0.039)	0.783 (0.076)
	`RegGMM`	0.901 (0.025)	0.0016 (0.0001)	1.153 (0.036)	4.565 (1.184)
	`IID`	-	-	-	1.764 (0.034)
$p = 20$ $q = 200$	`MtRegGMM`	**0.947** (0.017)	**0.0001** (0.0000)	**0.599** (0.039)	**0.474** (0.060)
	`Joint`$_{\texttt{lasso}}$	0.893 (0.020)	**0.0001** (0.0000)	0.802 (0.033)	0.713 (0.053)
	`RegGMM`	0.898 (0.021)	0.0007 (0.0001)	1.165 (0.039)	3.016 (0.453)
	`IID`	-	-	-	5.119 (0.890)
$p = 100$ $q = 200$	`MtRegGMM`	**0.989** (0.011)	**0.0000** (0.0000)	**0.647** (0.089)	**0.645** (0.175)
	`Joint`$_{\texttt{lasso}}$	0.959 (0.023)	**0.0000** (0.0000)	0.962 (0.103)	1.105 (0.246)
	`RegGMM`	0.959 (0.023)	0.0002 (0.0000)	2.746 (0.299)	9.172 (1.225)
	`IID`	-	-	-	5.515 (1.208)

gradient algorithm that can be used to estimate sparse group lasso problems. For `MtRegGMM`, the algorithm is terminated when the relative duality gap $|\text{obj}_p - \text{obj}_d|/(1 + |\text{obj}_p| + |\text{obj}_d|)$, where obj_p and obj_d denote the primal and dual function values, respectively, and relative dual infeasibility $(\|\mathcal{W}^\top \boldsymbol{a} + \boldsymbol{g}\|_2)/(\|1 + \|\boldsymbol{a}\|_2)$ are both less than 10^{-6}. For `SLEP`, the algorithm is terminated when the relative difference of objective functions between adjacent iterations is less than 10^{-6} (`SLEP` does not give a dual sequence). Other parameters for `SLEP` are set to their default values [1], and both `MtRegGMM` and `SLEP` are implemented in MATLAB. We set $p = 50, 100$, $q = 100, 200$ and $n = 100$, while keeping the other settings same as in Table 1. We compare the two methods in computing time (in seconds) and the area under the ROC curve (ROC AUC). For a fair comparison, the same tuning parameters λ_1 and λ_2 are used in `MtRegGMM` and `SLEP` when comparing computing time and error of $\boldsymbol{\beta}$, and they are selected using cross validation with `MtRegGMM`. Table 3 reports the average criteria with standard errors in the parentheses. It is seen that `MtRegGMM` enjoys an improved computational efficiency and estimation accuracy.

[1]The SLEP method is implemented using software at http://www.yelabs.net/software/SLEP/

Table 3: Computing time and estimation accuracy with varying p, q while $n = 100$. Computing time is shown in seconds and AUC represents the area under the ROC curve.

p, q	Method	Computing time	Error of $\boldsymbol{\beta}$	AUC
50,100	`MtRegGMM`	1.110 (0.054)	0.681 (0.023)	0.997 (0.002)
	`SLEP`	1.221 (0.075)	0.712 (0.024)	0.994 (0.002)
50,200	`MtRegGMM`	2.362 (0.105)	0.730 (0.021)	0.989 (0.003)
	`SLEP`	2.762 (0.140)	0.746 (0.020)	0.989 (0.003)
100,100	`MtRegGMM`	4.078 (0.144)	0.755 (0.023)	0.991 (0.012)
	`SLEP`	4.639 (0.479)	0.787 (0.020)	0.985 (0.004)

6 Co-expression Quantitative Trait Loci Analysis

Glioblastoma multiforme is the most aggressive and fatal subtype of brain cancer (Bleeker et al., 2012), and existing therapies remain largely ineffective (Bleeker et al., 2012). It is imperative to explore effective treatment, such as new gene therapies (Kwiatkowska et al., 2013), and the characterization of the molecular underpinning of the disease is the key. We analyze the REMBRANDT trial (GSE108476), with a subcohort of $n = 178$ glioblastoma multiforme patients, who had undergone microarray and single nucleotide polymorphism (SNP) chip profiling, with both gene expression and SNP data available for analysis. The raw data were pre-processed and normalized using standard pipelines (Gusev et al., 2018).

The response variables are the expression levels of $p = 73$ genes belonging to the human glioma pathway in the Kyoto Encyclopedia of Genes and Genomes database (Kanehisa and Goto, 2000); see Figure A1. The covariates include local SNPs (within 2kb upstream and 0.5kb downstream of the 73 genes), resulting in a total of 118 SNPs. SNPs are coded with "0" indicating homozygous in the major allele and "1" otherwise. Age and gender are also included in analysis. We construct the population-level gene co-expression network, and examine if and how age, gender and SNPs modulate the network.

We have used the proposed method to construct the population level network as shown in the left panel of Figure 3. Most of the connected genes in this network are known

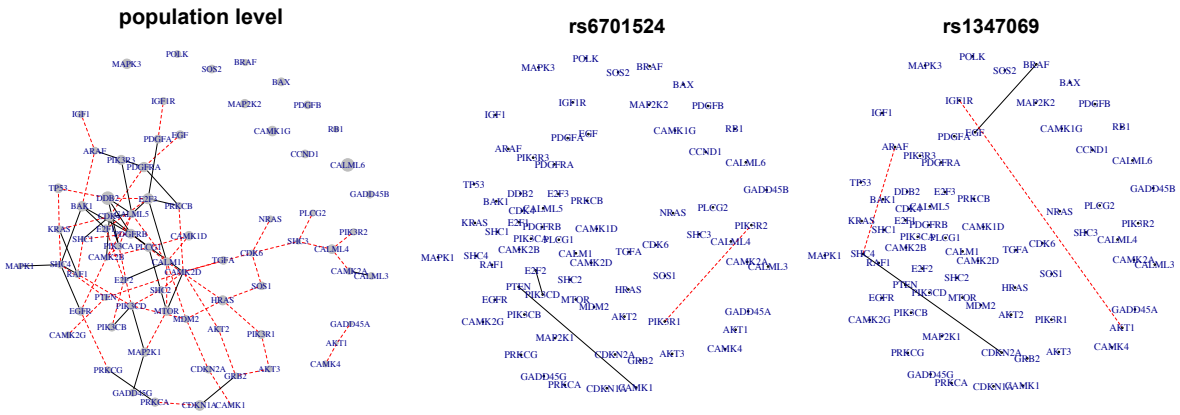

Figure 3: The population-level gene co-expression network (left), where the size of each node is proportional to the mean expression level and effects of SNPs on edges (middle and right), with edges with positive and negative partial correlations are shown in red dashed and black solid lines, respectively.

to be oncogenes. For example, PIK3CA is a protein coding gene and is one of the most highly mutated oncogenes identified in human cancers (Samuels and Velculescu, 2004). The PIK3CA gene is a part of the PI3K/AKT/MTOR signaling pathway, which is one of the core pathways in human glioblastoma multiforme and other types of cancer (Network et al., 2008). In the left panel of Figure 3, we can identify several core pathways in human glioblastoma multiforme including the PI3K/ AKT/MTOR, Ras-Raf-MEK-ERK and calcium signaling pathways. We also considered the `IID` method, estimated via neighborhood selection (Meinshausen and Bühlmann, 2006); the graphical lasso (Friedman et al., 2008) gives a similar but denser solution. Figure 4 reveals that the `IID` estimator shares many patterns similar to that from `MtRegGMM`, albeit yielding a more noisy network.

We next examine the covariate effects on the network. Identified by our method are nine co-expression quantitative trait loci, namely, `rs6701524`, `rs10519201`, `rs1347069`, `rs9303511`, `rs503314`, `rs7286558`, `rs759950`, `rs306098`, `rs25919`. The network effects of `rs6701524` are shown in the middle panel of Figure 3. This single nucleotide polymorphism, residing in MTOR, is found to affect co-expressions of genes in the PI3K/ AKT/MTOR

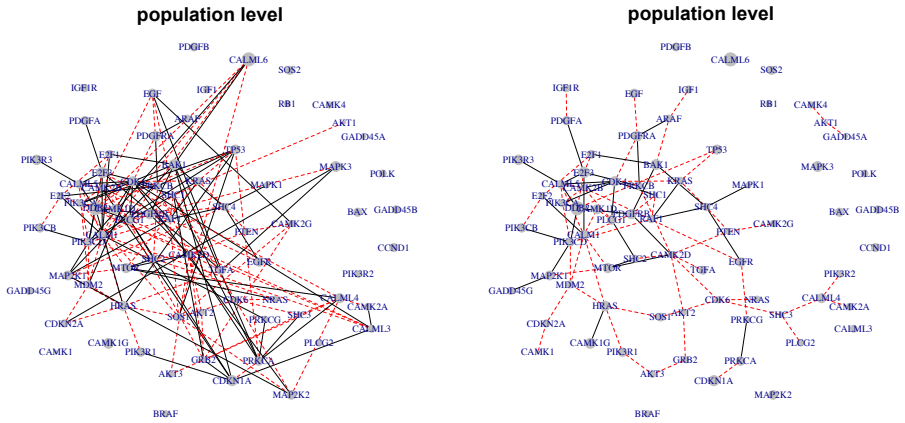

Figure 4: Comparison of the population-level gene co-expression networks estimated via `IID` (left) and `MtRegGMM` (right).

pathway. This is an interesting finding as PI3K/MTOR is a key pathway in glioblastoma multiforme development and progression, and inhibition of PI3K/MTOR signaling was found effective in increasing survival with glioblastoma multiforme (Batsios et al., 2019). This co-expression quantitative trait locus can potentially play an important role in activating the PI3K/MTOR pathway. Shown in the right panel of Figure 3 are the network effects of `rs1347069`, a variant of MAP2K1, indicating that this variant mostly affects the co-expressions of genes in the Ras-Raf-MEK-ERK pathway.

Moreover, we have found some novel co-expression quantitative trait loci. For example, `rs10519201` regulates the co-expressions of PDGFB and GADD45A; `rs9303511` is associated with the co-expressions of GADD45A and CALML4; `rs503314` influences the co-expressions of CCND1 and PLCG2, CALML3; `rs7286558` may modify the co-expressions of EGFR and PRKCA; `rs759950` modulates the co-expressions of MAP2K1 and HRAS; `rs306098` influences the co-expressions of PIK3CD and CAMK2D; `rs25919` may alter the co-expressions of CAMK2A and GADD45G.

7 Discussion

While we had assumed the diagonal σ^{jj}'s to be free of $\boldsymbol{u}$ in model (2), our modeling framework can be extended to allow σ^{jj}'s to depend on $\boldsymbol{u}$, for example, via $\sigma^{jj}(\boldsymbol{u}) = g(\boldsymbol{\nu}_j^\top \boldsymbol{u})$. Here, $g(\cdot)$ is a link function, e.g., $g(x) = \exp(x)$, and $\boldsymbol{\nu}_j$ is a vector of unknown coefficients. We can rewrite (3) as

$$Z_j \times g(\boldsymbol{\nu}_j^\top \boldsymbol{u}) = \sum_{k \neq j}^{p} \theta_{jk0} Z_k + \sum_{k \neq j}^{p} \sum_{h=1}^{q} \theta_{jkh} u_h Z_k + \tilde{\epsilon}_j, \tag{12}$$

where $\theta_{jkh} = -\beta'_{jkh}$ and $\mathrm{Var}(\tilde{\epsilon}_j) = g(\boldsymbol{\nu}_j^\top \boldsymbol{u})$. In this case, parameters $\boldsymbol{\nu}_j$ and $\boldsymbol{\theta}_j = (\theta_{j10}, \ldots, \theta_{jp0}, \ldots, \theta_{j1q}, \ldots, \theta_{jpq})$ can be estimated by considering a joint loss function:

$$\frac{1}{2n} \sum_{j=1}^{p} \sum_{i=1}^{n} \left(z_j^{(i)} \times g(\boldsymbol{\nu}_j^\top \boldsymbol{u}^{(i)}) - \boldsymbol{w}_{-j}^{(i)\top} \boldsymbol{\theta}_j \right)^2,$$

subject to regularizations on $\boldsymbol{\nu}_j$ and $\boldsymbol{\theta}_j$ as specified in (4). The extension is nontrivial, requiring an iterative estimation of $\boldsymbol{\nu}_j$ and $\boldsymbol{\beta}_j$ and a new theoretical analysis.

References

Batsios, G., P. Viswanath, E. Subramani, C. Najac, A. M. Gillespie, R. D. Santos, A. R. Molloy, R. O. Pieper, and S. M. Ronen (2019). Pi3k/mtor inhibition of idh1 mutant glioma leads to reduced 2hg production that is associated with increased survival. *Scientific Reports 9*(1), 1–15.

Bickel, P. J., Y. Ritov, and A. B. Tsybakov (2009). Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics 37*(4), 1705–1732.

Bleeker, F. E., R. J. Molenaar, and S. Leenstra (2012). Recent advances in the molecular understanding of glioblastoma. *Journal of Neuro-oncology 108*(1), 11–27.

Boyd, S., N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine learning 3*(1), 1–122.

Cai, T. T., H. Li, W. Liu, and J. Xie (2012). Covariate-adjusted precision matrix estimation with an application in genetical genomics. *Biometrika 100*(1), 139–156.

Chen, M., Z. Ren, H. Zhao, and H. Zhou (2016). Asymptotically normal and efficient estimation of covariate-adjusted gaussian graphical model. *Journal of the American Statistical Association 111*(513), 394–406.

Cheng, J., E. Levina, P. Wang, and J. Zhu (2014). A sparse ising model with covariates. *Biometrics 70*(4), 943–953.

Clarke, F. H. (1990). *Optimization and nonsmooth analysis*. SIAM.

Danaher, P., P. Wang, and D. M. Witten (2014). The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76*(2), 373–397.

Diestel, R., A. Schrijver, and P. D. Seymour (2010). Graph theory. *Oberwolfach Reports 7*(1), 521–580.

Friedman, J., T. Hastie, and R. Tibshirani (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics 9*(3), 432–441.

Guo, J., E. Levina, G. Michailidis, and J. Zhu (2011). Joint estimation of multiple graphical models. *Biometrika 98*(1), 1–15.

Gusev, Y., K. Bhuvaneshwar, L. Song, J.-C. Zenklusen, H. Fine, and S. Madhavan (2018). The rembrandt study, a large collection of genomic data from brain cancer patients. *Scientific Data 5*, 180158.

Hestenes, M. R. (1969). Multiplier and gradient methods. *Journal of optimization theory and applications 4*(5), 303–320.

Jenatton, R., J.-Y. Audibert, and F. Bach (2011). Structured variable selection with sparsity-inducing norms. *The Journal of Machine Learning Research 12*, 2777–2824.

Ko, B. and Q. Tang (2008). Sums of dependent nonnegative random variables with subexponential tails. *Journal of Applied Probability 45*(1), 85–94.

Kolar, M., A. P. Parikh, and E. P. Xing (2010). On sparse nonparametric conditional covariance selection. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 559–566.

Kummer, B. (1988). Newton's method for non-differentiable functions. *Mathematical research 45*, 114–125.

Kwiatkowska, A., M. S. Nandhu, P. Behera, E. A. Chiocca, and M. S. Viapiano (2013). Strategies in gene therapy for glioblastoma. *Cancers 5*(4), 1271–1305.

Lauritzen, S. L. (1996). *Graphical Models*, Volume 17. Clarendon Press.

Lewis, R. (2015). *A Guide to Graph Colouring*, Volume 7. Springer.

Li, B., H. Chun, and H. Zhao (2012). Sparse estimation of conditional graphical models with application to gene networks. *Journal of the American Statistical Association 107*(497), 152–167.

Li, X., D. Sun, and K.-C. Toh (2018a). A highly efficient semismooth newton augmented lagrangian method for solving lasso problems. *SIAM Journal on Optimization 28*(1), 433–458.

Li, X., D. Sun, and K.-C. Toh (2018b). On efficiently solving the subproblems of a level-set method for fused lasso problems. *SIAM Journal on Optimization 28*(2), 1842–1866.

Li, Y., B. Nan, and J. Zhu (2015). Multivariate sparse group lasso for the multivariate multiple linear regression with an arbitrary group structure. *Biometrics 71*(2), 354–363.

Liu, H., X. Chen, L. Wasserman, and J. D. Lafferty (2010). Graph-valued regression. In *Advances in Neural Information Processing Systems*, pp. 1423–1431.

Liu, J., S. Ji, J. Ye, et al. (2009). Slep: Sparse learning with efficient projections. *Arizona State University 6*(491), 7.

Lounici, K., M. Pontil, S. Van De Geer, and A. B. Tsybakov (2011). Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics 39*(4), 2164–2204.

Meinshausen, N. and P. Bühlmann (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics 34*(3), 1436–1462.

Merlevède, F., M. Peligrad, and E. Rio (2011). A bernstein type inequality and moderate deviations for weakly dependent sequences. *Probability Theory and Related Fields 151*(3), 435–474.

Mifflin, R. (1977). Semismooth and semiconvex functions in constrained optimization. *SIAM Journal on Control and Optimization 15*(6), 959–972.

Negahban, S. N., P. Ravikumar, M. J. Wainwright, and B. Yu (2012). A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science 27*(4), 538–557.

Network, C. G. A. R. et al. (2008). Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature 455*(7216), 1061.

Ni, Y., F. C. Stingo, and V. Baladandayuthapani (2019). Bayesian graphical regression. *Journal of the American Statistical Association 114*(525), 184–197.

Peng, J., P. Wang, N. Zhou, and J. Zhu (2009). Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association 104*(486), 735–746.

Qi, M. and T. Li (2022). The non-overlapping statistical approximation to overlapping group lasso. *arXiv preprint arXiv:2211.09221*.

Rockafellar, R. T. (1976). Augmented lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of operations research 1*(2), 97–116.

Samuels, Y. and V. E. Velculescu (2004). Oncogenic mutations of pik3ca in human cancers. *Cell Cycle 3*(10), 1221–1224.

Simon, N., J. Friedman, T. Hastie, and R. Tibshirani (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics 22*(2), 231–245.

van der Wijst, M. G., H. Brugge, D. H. de Vries, P. Deelen, M. A. Swertz, and L. Franke (2018). Single-cell rna sequencing identifies celltype-specific cis-eqtls and co-expression qtls. *Nature Genetics 50*(4), 493–497.

Wang, Y., S. J. Joseph, X. Liu, M. Kelley, and R. Rekaya (2012). Snpxge2: a database for human snp–coexpression associations. *Bioinformatics 28*(3), 403–410.

Won, J.-H., J. Xu, and K. Lange (2019). Projection onto minkowski sums with application to constrained learning. In *International Conference on Machine Learning*, pp. 3642–3651. PMLR.

Yan, X. and J. Bien (2017). Hierarchical Sparse Modeling: A Choice of Two Group Lasso Formulations. *Statistical Science 32*(4), 531 – 560.

Yu, G. and J. Bien (2017). Learning local dependence in ordered data. *The Journal of Machine Learning Research 18*(1), 1354–1413.

Yuan, L., J. Liu, and J. Ye (2011). Efficient methods for overlapping group lasso. *Advances in neural information processing systems 24*.

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

Zhang, J. and Y. Li (2022). High-dimensional gaussian graphical regression models with covariates. *Journal of the American Statistical Association*, 1–13.

Zhang, J., W. W. Sun, and L. Li (2019). Mixed-effect time-varying network model and application in brain connectivity analysis. *Journal of the American Statistical Association*, 1–15.

Zhang, J., W. W. Sun, and L. Li (2022). Generalized connectivity matrix response regression with applications in brain connectivity studies. *Journal of Computational and Graphical Statistics* (just-accepted), 1–30.

Zhang, Y., N. Zhang, D. Sun, and K.-C. Toh (2020). An efficient hessian based algorithm for solving large-scale sparse group lasso problems. *Mathematical Programming 179*(1), 223–263.

Supplementary Materials for "Multi-task Learning for Gaussian Graphical Regressions with High Dimensional Covariates"

A1 Details of Algorithm 1

A1.1 The matrix $\mathcal{W}$ in Algorithm 1

We introduce $\mathcal{W}$ for notational ease. Our implementation indeed avoids the creation of such a large matrix of $\mathcal{W}$. Instead, it represents $\mathcal{W}$ as a sparse matrix, wherein only the non-zero elements, along with their corresponding row and column indices, are stored. When operating matrix-vector multiplications, only the non-zero elements in $\mathcal{W}$ are multiplied with the corresponding components of the target vector. As a result, the computational complexity and memory efficiency of computing $\mathcal{W}^\top \boldsymbol{a}$ is the same as $\sum_{j=1}^{p} \boldsymbol{W}_{-j}^\top \boldsymbol{a}_k$, where $\boldsymbol{a} = (\boldsymbol{a}_1^\top, \ldots, \boldsymbol{a}_p^\top)^\top$.

A1.2 Semi-smoothness and semi-smooth Newton

The following definition on semi-smooth is adopted from Kummer (1988). Let $\mathcal{O} \in \mathbb{R}^n$ be an open set and $F : \mathcal{O} \to \mathbb{R}^m$ be a locally Lipschitz continuous function on $\mathcal{O}$. We call F semi-smooth on $\mathcal{O}$ with respect to a set-valued mapping $\mathcal{K} : \mathbb{R}^n \to \mathbb{R}^{m\times n}$, if for any $\boldsymbol{x} \in \mathcal{O}$ and $\boldsymbol{V} \in \mathcal{K}(\boldsymbol{x} + \Delta\boldsymbol{x})$ with $\Delta\boldsymbol{x} \to 0$, F is directionally differentiable at $\boldsymbol{x}$ and also $F(\boldsymbol{x}+\Delta\boldsymbol{x}) - F(\boldsymbol{x}) - \boldsymbol{V}\Delta\boldsymbol{x} = o(\|\Delta\boldsymbol{x}\|_2)$. Additionally, F is said to be strongly semi-smooth if for any $\boldsymbol{x} \in \mathcal{O}$ and $\boldsymbol{V} \in \mathcal{K}(\boldsymbol{x} + \Delta\boldsymbol{x})$ with $\Delta\boldsymbol{x} \to 0$, F is directionally differentiable at $\boldsymbol{x}$ and also $F(\boldsymbol{x}+\Delta\boldsymbol{x}) - F(\boldsymbol{x}) - \boldsymbol{V}\Delta\boldsymbol{x} = O(\|\Delta\boldsymbol{x}\|_2^2)$. For example, continuous piecewise linear functions are strongly semi-smooth everywhere. The notion of semi-smooth functions enables the application of certain efficient algorithms commonly used for smooth optimization problems, including the Newton method.

Recall that Step 1 in Algorithm 1 requires solving $F(\boldsymbol{a}) = \mathbf{0}$, where

$$F(\boldsymbol{a}) := \nabla_{\boldsymbol{a}} \psi_{\tau_k}(\boldsymbol{a}; \boldsymbol{v}^k) = \boldsymbol{y} + \boldsymbol{a} - \tau_k \mathcal{W} \mathrm{Prox}_h(\tau^{-1} \boldsymbol{v}^k - \mathcal{W}^\top \boldsymbol{a}),$$

and $h(\boldsymbol{\beta}) = \lambda_1 \sum_{h=1}^q \|\boldsymbol{b}_h\|_2 + \lambda_2 \|\boldsymbol{\beta}\|_1$, $\boldsymbol{b}_h = ((\boldsymbol{\beta}_1)_{(h)}, \ldots, (\boldsymbol{\beta}_p)_{(h)})^\top$. Due to the combined sparsity structure in $h(\cdot)$, it is challenging to derive the generalized Jacobian of Prox_h. Instead, Zhang et al. (2020) derived a multifunction $\mathcal{D}$ that serves as a surrogate of the generalized Jacobian of Prox_h and showed that Prox_h is strongly semi-smooth with respect to $\mathcal{D}$. Correspondingly, $F(\boldsymbol{a})$ is semi-smooth with respect to $\boldsymbol{I} + \tau_k \mathcal{W} \boldsymbol{D} \mathcal{W}^\top$, where $\boldsymbol{D} \in \mathcal{D}$ is evaluated at $\tau_k^{-1} \boldsymbol{v}^k - \mathcal{W}^\top \boldsymbol{a}$. As a result, one can employ the following semi-smooth Newton method (Mifflin, 1977) to solve for $F(\boldsymbol{a}) = \mathbf{0}$,

$$\boldsymbol{a}^{k+1} = \boldsymbol{a}^k + \alpha_k \boldsymbol{d}_k,$$

where α_k is the step size selected using line search and $\boldsymbol{d}_k$ is the solution to

$$(\boldsymbol{I} + \tau_k \mathcal{W} \boldsymbol{D} \mathcal{W}^\top) \boldsymbol{d} = F(\boldsymbol{a}^k).$$

The semi-smooth Newton (Mifflin, 1977) is a generalization of the Newton's method for solving non-smooth equations. Next, we give more details on the construction of surrogate function $\boldsymbol{D}$.

Write $h_1(\boldsymbol{\beta}) = \lambda_1 \sum_{h=1}^q \|\boldsymbol{b}_h\|_2$ and $h_2(\boldsymbol{\beta}) = \lambda_2 \|\boldsymbol{\beta}\|_1$. For a vector $\boldsymbol{a}$, we use $\mathrm{Diag}(\boldsymbol{a})$ to denote a diagonal matrix whose diagonal elements are the components in $\boldsymbol{a}$. For square matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n$, we use $\mathrm{Diag}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n)$ to denote a block diagonal matrix whose diagonals are $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n$. Given any vector $\boldsymbol{\alpha} \in \mathbb{R}^{p(p-1)(q+1)}$, denote $\boldsymbol{\beta} = \mathrm{Prox}_{h_2}(\boldsymbol{\alpha})$ and $\boldsymbol{\Theta} \in \partial \mathrm{Prox}_{h_2}(\boldsymbol{\alpha})$, where $\partial \mathrm{Prox}_{h_2}(\boldsymbol{\alpha})$ is the Clarke generalized Jacobian of $\mathrm{Prox}_{h_2}(\boldsymbol{\alpha})$ and $\boldsymbol{\Theta}$ can be calculated as $\boldsymbol{\Theta} = \mathrm{Diag}(\boldsymbol{\theta})$ with $\theta_j = \begin{cases} 0, & \text{if } |\alpha_i| \le \lambda_2, \\ 1, & \text{otherwise,} \end{cases}$. Next, define

$\boldsymbol{\Lambda} = \text{Diag}(\boldsymbol{\Lambda}_1, \ldots, \boldsymbol{\Lambda}_q)$, where

$$\boldsymbol{\Lambda}_h = \begin{cases} \frac{\lambda_1}{\|\boldsymbol{b}_h\|_2} \left(\boldsymbol{I} - \frac{\boldsymbol{b}_h \boldsymbol{b}_h^\top}{\|\boldsymbol{b}_h\|_2^2} \right), & \text{if } \|\boldsymbol{b}_h\|_2 \geq \lambda_1, \\ \boldsymbol{I}, & \text{otherwise.} \end{cases}$$

Let $\boldsymbol{L} = [\boldsymbol{L}_1; \cdots; \boldsymbol{L}_q]$, where $\boldsymbol{L}_h$ is a binary matrix such that $\boldsymbol{L}_h \boldsymbol{\beta} = \boldsymbol{b}_h$. Then, the surrogate function $\boldsymbol{D}(\boldsymbol{\alpha})$ is defined as $\boldsymbol{D}(\boldsymbol{\alpha}) = (\boldsymbol{I} - \boldsymbol{L}^\top \boldsymbol{\Lambda} \boldsymbol{L}) \boldsymbol{\Theta}$. Here, $\boldsymbol{I} - \boldsymbol{L}^\top \boldsymbol{\Lambda} \boldsymbol{L}$ is a surrogate for the generalized Jacobian of Prox_{h_1} and $\boldsymbol{\Theta}$ is the Clarke generalized Jacobian of Prox_{h_2}. By Proposition 2.1 in Zhang et al. (2020) that shows $\text{Prox}_h(\boldsymbol{\beta}) = \text{Prox}_{h_1}(\text{Prox}_{h_2}(\boldsymbol{\beta}))$, $\boldsymbol{D}$ then serves as a generalized Jacobian of Prox_h. As $\boldsymbol{L}$ and $\boldsymbol{\Theta}$ are both binary and sparse and $\boldsymbol{\Lambda}$ is block diagonal, $\boldsymbol{d}_k$ in the semi-smooth Newton can be solved efficiently from $(\boldsymbol{I} + \tau_k \mathcal{W} \boldsymbol{D} \mathcal{W}^\top) \boldsymbol{d} = F(\boldsymbol{a}^k)$ and the overall computational cost depends on the number of nonzero elements in the parameters, greatly reducing the computational cost. We refer to Section 4.3 in Zhang et al. (2020) for numerical techniques that can quickly solve for $\boldsymbol{d}_k$.

A1.3 Convergence of Algorithm 1

Algorithm 1 is an inexact augmented Lagrangian (Rockafellar, 1976) as we have to solve the subproblem numerically within the augmented Lagrangian algorithm. The global and local convergence of inexact augmented Lagrangian iterates has been investigated in the optimization literature (Rockafellar, 1976; Zhang et al., 2020). In particular, if the following condition is met when solving the subproblem in Algorithm 1:

$$\psi_{\tau_k}(\boldsymbol{a}^{k+1}; \boldsymbol{v}^k) - \inf_{\boldsymbol{a}} \psi_{\tau_k}(\boldsymbol{a}; \boldsymbol{v}^k) \leq e_k^2 / 2\tau_k, \quad \sum_{k=0}^{\infty} e_k < \infty, \tag{A1}$$

then the inexact augmented Lagrangian iterates from Algorithm 1 converge to the optimal solution (Rockafellar, 1976); additionally, if the following condition is also met:

$$\psi_{\tau_k}(\boldsymbol{a}^{k+1}; \boldsymbol{v}^k) - \inf_{\boldsymbol{a}} \psi_{\tau_k}(\boldsymbol{a}; \boldsymbol{v}^k) \leq (\eta_k^2 / 2\tau_k) \|\boldsymbol{v}^{k+1} - \boldsymbol{v}^k\|_2^2, \quad \sum_{k=0}^{\infty} \eta_k < \infty, \tag{A2}$$

then these iterates also have a linear convergence rate when k is sufficiently large (Zhang et al., 2020). Moreover, as $\psi_{\tau_k}(;\boldsymbol{v})$ is strongly convex, satisfying

$$\psi_{\tau_k}(\boldsymbol{a}^{k+1};\boldsymbol{v}^k) - \inf_{\boldsymbol{a}} \psi_{\tau_k}(\boldsymbol{a};\boldsymbol{v}^k) \leq \|\nabla\psi_{\tau_k}(\boldsymbol{a}^{k+1};\boldsymbol{v}^k)\|_2^2,$$

(A1)-(A2) can be replaced by two easy-to-check criteria, as shown in Theorem 1.

In particular, in Theorem 1, the statement on the global convergence directly follows from Theorem 5 in Rockafellar (1976) and the results on the local convergence follow from Theorem 4.1 in Zhang et al. (2020). We omit the proof.

A2 Additional Numerical Results

A2.1 Algorithm for generating $\boldsymbol{\Sigma}(\boldsymbol{u}_i)$

When generating $\boldsymbol{\Sigma}(\boldsymbol{u}_i)$, we randomly select 3 covariates to have nonzero effects. In the graphs for these covariates and the population-level graph, five edges are randomly selected to be nonzero. We set $\sigma^{jj} = 1$ for $j \in [p]$. The initial nonzero coefficients β_{jkh} are set to 0.3 and then rescaled to ensure positive definiteness. The rescaling procedure encompasses three steps.

Step 1: generate $\{\boldsymbol{u}_i\}_{i\in[n]}$ and $\{\beta^{\text{init}}_{jkh}\}_{k\neq j\in[p],h\in\{0\}\cup[q]}$.

Step 2: For each j, calculate $\tilde{\beta}_{jkh} = \beta^{\text{init}}_{jkh} / \sum_{k\neq j\in[p],h\in\{0\}\cup[q]} |\beta^{\text{init}}_{jkh}|$ for all k, h.

Step 3: set $\beta_{jkh} = \beta_{kjh} = (\tilde{\beta}_{jkh} + \tilde{\beta}_{kjh})/2$ for all j, k, h.

Briefly, Step 2 ensures the diagonal dominance of the generated precision matrices and Step 3 ensures their symmetry. This algorithm is designed in a similar spirit as in Peng et al. (2009), where diagonal dominance was used to ensure positive definiteness of covariance matrices.

A2.2 Simulations with a larger p

We have expanded our simulations by adding the cases with $p > q$ and $n < p$. Specifically, we set $p = 100, 200$, $q = 20$ and $n = 100$ and compare different methods in terms of true positive rate, false positive rate, area under the ROC curve (ROC AUC) and estimation error. Other settings remain the same as in Table 1. Table A1 reports the average criteria with standard errors in the parentheses. It is seen that `MtRegGMM` achieves a good performance when $p > q$ and $n < p$, outperforming the other methods in both selection and estimation accuracy. The ROC AUC values for `MtRegGMM` are close to 1 under both settings.

Table A1: Estimation accuracy of $\boldsymbol{\beta}$ in simulations with $p = 100, 200$, $q = 20$ and $n = 100$. TPR and FPR represent the true and false positive rates, respectively, and AUC represents the area under the ROC curve.

p, q, n	Method	$\text{TPR}_{\boldsymbol{\beta}}$	$\text{FPR}_{\boldsymbol{\beta}}$	AUC	Error of $\boldsymbol{\beta}$
100,20,100	`MtRegGMM`	**0.960** (0.011)	**0.0000** (0.0000)	**0.992** (0.004)	**0.599** (0.038)
	$\texttt{Joint}_{\texttt{lasso}}$	0.942 (0.014)	**0.0000** (0.0000)	0.985 (0.003)	0.789 (0.054)
	`RegGMM`	0.949 (0.011)	0.0017 (0.0001)	0.984 (0.005)	2.625 (0.051)
200,20,100	`MtRegGMM`	**0.936** (0.022)	**0.0000** (0.0000)	**0.981** (0.010)	**0.722** (0.069)
	$\texttt{Joint}_{\texttt{lasso}}$	0.916 (0.025)	**0.0000** (0.0000)	0.977 (0.010)	0.807 (0.046)
	`RegGMM`	0.917 (0.029)	0.0009 (0.0000)	0.971 (0.012)	3.708 (0.072)

A2.3 Positive definiteness in finite sample

We have evaluated the performance of the post-hoc rescaling procedure in Section 3.2 under the same setting as in Table 1. The results are shown in Table A2, where the rescaled $\hat{\boldsymbol{\beta}}$ is denoted as $\hat{\boldsymbol{\beta}}^r$. It is seen that the rescaled estimators have slightly smaller estimation errors.

Table A2: Estimation accuracy of $\boldsymbol{\beta}$'s and $\boldsymbol{\beta}^r$'s in simulations with $n = 100$, varying network size p and covariate dimension q.

	$p = 20$ $q = 100$	$p = 20$ $q = 200$	$p = 100$ $q = 200$
$\boldsymbol{\beta}_{\text{error}}$	0.534 (0.032)	0.549 (0.037)	0.612 (0.034)
$\boldsymbol{\beta}^r_{\text{error}}$	0.530 (0.031)	0.504 (0.031)	0.545 (0.029)

Table A3: Estimation accuracy of $\boldsymbol{\beta}$ and $\boldsymbol{\Omega}$ in simulations with $p = 5, q = 200$ and $n = 100, 200$. TPR and FPR represent the true and false positive rates, respectively.

n	Method	$\text{TPR}_{\boldsymbol{\beta}}$	$\text{FPR}_{\boldsymbol{\beta}}$	Error of $\boldsymbol{\beta}$	Error of $\boldsymbol{\Omega}$
100	`MtRegGMM`	**0.946** (0.020)	**0.0008** (0.0002)	**0.474** (0.036)	**0.324** (0.052)
	$\texttt{Joint}_{\texttt{group}}$	0.873 (0.036)	0.0132 (0.0005)	1.170 (0.037)	1.515 (0.081)
	$\texttt{Joint}_{\texttt{lasso}}$	0.907 (0.023)	0.0021 (0.0003)	0.657 (0.031)	0.505 (0.060)
	`RegGMM`	0.887 (0.024)	0.0034 (0.0004)	0.725 (0.033)	0.744 (0.104)
	`IID`	-	-	-	1.100 (0.023)
200	`MtRegGMM`	**1.000** (0.000)	**0.0001** (0.0000)	**0.206** (0.009)	**0.061** (0.006)
	$\texttt{Joint}_{\texttt{group}}$	0.993 (0.007)	0.0134 (0.0001)	0.693 (0.017)	0.553 (0.030)
	$\texttt{Joint}_{\texttt{lasso}}$	**1.000** (0.000)	0.0004 (0.0001)	0.258 (0.012)	0.087 (0.006)
	`RegGMM`	**1.000** (0.000)	0.0009 (0.0002)	0.312 (0.016)	0.134 (0.014)
	`IID`	-	-	-	1.114 (0.014)

A2.4 Comparison with the joint group lasso estimator

We consider the standard multi-tasking learning method that solves the p regression tasks jointly with a group lasso, referred to as $\texttt{Joint}_{\texttt{group}}$. We set $p = 5, q = 200$ and $n = 100, 200$, and compared with $\texttt{Joint}_{\texttt{group}}$. Other settings remain the same as in Table 1. We set $p = 5$, as $\texttt{Joint}_{\texttt{group}}$ requires samples at least in the order of $O(s_g p^2)$. The results are summarized in Table A3. It is seen that `MtRegGMM` achieves a better performance when compared to $\texttt{Joint}_{\texttt{group}}$, as `MtRegGMM` exploits both the within group sparsity (lasso) and across group sparsity (group lasso).

A3 Technical Lemmas

We state the technical lemmas that will be used in the proofs.

Lemma 1 (Lemma 1 in Bellec et al. (2018)) *Let* $pen : \mathbb{R}^d \to \mathbb{R}$ *be any convex function and* $\hat{\boldsymbol{\beta}}$ *be defined by*

$$\hat{\boldsymbol{\beta}} \in \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \left\{ \|\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\beta}\|_2^2 + pen(\boldsymbol{\beta}) \right\},$$

where $\boldsymbol{W} \in \mathbb{R}^{n\times d}$, $\boldsymbol{y} \in \mathbb{R}^n$. *Then for* $\boldsymbol{\beta} \in \mathbb{R}^d$,

$$\|\boldsymbol{y} - \boldsymbol{W}\hat{\boldsymbol{\beta}}\|_2^2 + pen(\hat{\boldsymbol{\beta}}) + \|\boldsymbol{W}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|_2^2 \le \|\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\beta}\|_2^2 + pen(\boldsymbol{\beta}).$$

Lemma 2 (Theorem F in Graybill and Marsaglia (1957)) *Let* $\boldsymbol{\epsilon}_j \sim \mathcal{N}_p(\boldsymbol{0}, \sigma^2\boldsymbol{I})$ *and* $\boldsymbol{A}$ *be an* $p \times p$ *idempotent matrix with rank equals to* $r \le p$. *Then,* $\boldsymbol{\epsilon}_j^\top \boldsymbol{A}\boldsymbol{\epsilon}_j/\sigma^2$ *follows a* χ^2 *distribution with* r *degrees of freedom.*

Lemma 3 (Lemma 1 in Laurent and Massart (2000)) *Suppose that* U *follows a* χ^2 *distribution with* r *degrees of freedom. For any* $x > 0$, *it holds that*

$$P(U - r \ge 2\sqrt{rx} + 2x) \le \exp(-x).$$

Lemma 4 (Proposition 5.16 in Vershynin (2010)) *Let* $X_1, \ldots, X_n$ *be independent centered sub-exponential random variables. Let* $v_1 = \max_i \|X_i\|_{\psi_1}$, *where* $\|X_i\|_{\psi_1} = \sup_{d\ge 1} d^{-1}(E|X_i|^d)^{1/d}$ *denotes the sub-exponential norm. There exists a constant* c *such that, for any* $t > 0$,

$$P\left(\sum_{i=1}^n X_i \ge t\right) \le \exp\left\{-c\min\left(\frac{t^2}{v_1^2 n}, \frac{t}{v_1}\right)\right\}.$$

Lemma 5 (Theorem 4.1 in Kuchibhotla and Chakrabortty (2018)) *Let* $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ *be independent random vectors in* $\mathbb{R}^p$. *Assume each element of* $\boldsymbol{X}_i$ *is sub-exponential with* $\|X_{i,j}\|_{\psi_1} < K_2$, $i \in [n], j \in [p]$. *Let* $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{X}} = \boldsymbol{X}^\top\boldsymbol{X}/n$ *and* $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \mathbb{E}(\boldsymbol{X}^\top\boldsymbol{X}/n)$. *Define*

$$\Upsilon_{n,k} = \max_{j,k} \frac{1}{n}\sum_{i=1}^n Var\{X_{i,j}X_{i,k}\}.$$

Then for any $t > 0$, with probability at least $1 - \mathcal{O}(p^{-1})$,

$$\sup_{\|\boldsymbol{v}\|_0 \le k,\, \|\boldsymbol{v}\|_2 \le 1} \left| \boldsymbol{v}^\top (\hat{\boldsymbol{\Sigma}}_{\boldsymbol{X}} - \boldsymbol{\Sigma}_{\boldsymbol{X}}) \boldsymbol{v} \right| \precsim k \sqrt{\frac{\Upsilon_{n,k} log\, p}{n}} + K_2^2 \frac{k (log\, n log\, p)^2}{n}.$$

Lemma 6 (Lemma 12 in Loh and Wainwright (2011)) *For any symmetric matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$ and if $|\boldsymbol{v}^\top \boldsymbol{\Sigma} \boldsymbol{v}| \le \delta_1$ for any $\boldsymbol{v} \in \{\boldsymbol{v} : \|\boldsymbol{v}\|_0 \le 2s \, and \, \|\boldsymbol{v}\|_2 = 1\}$, then*

$$|\boldsymbol{v}^\top \boldsymbol{\Sigma} \boldsymbol{v}| \le 27 \delta_1 (\|\boldsymbol{v}\|_2^2 + \frac{1}{s} \|\boldsymbol{v}\|_1^2), \text{ for any } \quad \boldsymbol{v} \in \mathbb{R}^p.$$

A4 Proofs of Main Results

We introduce the notation used in our proof. We denote the true parameters by $\boldsymbol{\beta}_j$, $j \in [p]$, and in some places and without ambiguities, we also use them to denote the corresponding parameters or the arguments in functions. We write $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \ldots, \boldsymbol{\beta}_p^\top)^\top$. The index set $\{1, \ldots, (p-1)(q+1)\}$ is partitioned into $q+1$ groups, indexed by $(0), (1), \ldots, (q) \subset \{1, \ldots, (p-1)(q+1)\}$. For a group index subset $\mathcal{G} \in \{1, \ldots, q\}$, we let $(\mathcal{G}) = \cup_{h \in \mathcal{G}} (h)$, $(\mathcal{G}^c) = \cup_{h \notin \mathcal{G}} (h)$. For $\mathcal{G} \in \{1, \ldots, q\}$, we use $\boldsymbol{\beta}_{(\mathcal{G})}$ to denote $((\boldsymbol{\beta}_1)_{(\mathcal{G})}, \ldots, (\boldsymbol{\beta}_p)_{(\mathcal{G})})$, where $(\boldsymbol{\beta}_j)_{(\mathcal{G})}$ represent a subvector of $\boldsymbol{\beta}_j$ indexed by $(\mathcal{G})$. Moreover, we write $\boldsymbol{b}_h = ((\boldsymbol{\beta}_1)_{(h)}, \ldots, (\boldsymbol{\beta}_p)_{(h)})$. Let $\mathcal{S}_j$ be the element-wise support set of $\boldsymbol{\beta}_j$, that is, the collection of indices of the non-zero components of $\boldsymbol{\beta}_j$, $\mathcal{S}$ be the element-wise support set of $\boldsymbol{\beta}$ and $\mathcal{G}$ be the group-wise support set of $\boldsymbol{\beta}$. Moreover, let $s_j = |\mathcal{S}_j|$, $s_e = \sum_{j=1}^p s_j$, $s_g = |\mathcal{G}|$.

Corollary 1 *Consider N dependent χ^2 variables $Y_j \sim \chi^2_{d_j}$, $j \in [N]$ and an induced network $G(V, E)$ defined as in Theorem 2. Denote the maximum node degree of $G(V, E)$ by $d_{\max}$ and let $c_G = \min\left(d_{\max} + 1, \frac{1 + \sqrt{8|E|+1}}{2}\right)$. For any $t \ge 0$, it holds that*

$$\mathbb{P}\left(\sum_{j=1}^N Y_j - \sum_{j=1}^N d_j \ge t\right) \le c_G \exp\left[-\frac{\left\{t - (c_G - 1)\sum_{j=1}^N d_j\right\}^2}{4 c_G (t + \sum_{j=1}^N d_j)}\right].$$

A4.1 Proof of Theorem 2 and Corollary 1

In Corollary 1, we derive a new bound for the sum of dependent chi-squared random variables with a sparse dependency structure. We give a detailed proof as the results will be directly useful for our later Theorem 3. On the other hand, the results of Theorem 2 on the sum of dependent sub-exponential random variables are more general. Its proof follows the same arguments as in the proof of Corollary 1, except that it sets $d_j = 0, j \in [N]$, and is omitted.

Consider the network $G(V, E)$ with a node set $V = \{1, \ldots, N\}$ and an edge set $E = \{(j, k) : Y_j \not\perp Y_k\}$. The chromatic number of $G(V, E)$, denoted as h_G, is the smallest number of colors needed to color G so that nodes with the same color are independent, for example, two nodes connected by an edge cannot have the same color. It has been shown that h_G can be upper bounded by $\min\left(d_{\max} + 1, \frac{2+\sqrt{8|E|+1}}{2}\right)$ (Diestel et al., 2010), where $d_{\max}$ is the maximum node degree and $|E|$ is the number of edges.

Correspondingly, we may decompose the node set V as $V = \cup_{k=1}^{h_G} V_k$ such that all nodes in V_k are independent and $V_k \cap V_j = \varnothing$ for all $k \neq j$. For such defined V_k, $k \in [h_G]$, we let $E_k = \sum_{j \in V_k} Y_j$. As Y_j, $j \in V_k$ are independent, E_k follows a chi-squared distribution with degree of freedom $\sum_{j \in V_k} d_j$.

Next, setting $c_G = \min\left(d_{\max} + 1, \frac{2+\sqrt{8|E|+1}}{2}\right)$, it holds that

$$\begin{aligned}
\mathbb{P}\left(\sum_{j=1}^{N} Y_j - \sum_{j=1}^{N} d_j \geq t\right) &= \mathbb{P}\left(\sum_{k=1}^{h_G} E_k \geq t + \sum_{j=1}^{N} d_j\right) \leq \sum_{k=1}^{h_G} \mathbb{P}\left\{E_k \geq \left(t + \sum_{j=1}^{N} d_j\right)/c_G\right\} \\
&\leq \sum_{k=1}^{h_G} \exp\left[-\frac{\left\{\left(t + \sum_{j=1}^{N} d_j\right)/c_G - \sum_{j \in V_k} d_j\right\}^2}{4\left(t + \sum_{j=1}^{N} d_j\right)/c_G}\right] \\
&\leq c_G \exp\left[-\frac{\left\{t - (c_G - 1)\sum_{j=1}^{N} d_j\right\}^2}{4c_G\left(t + \sum_{j=1}^{N} d_j\right)}\right],
\end{aligned}$$

where the second inequality is due to Lemma 3. □

A4.2 Proof of Theorem 3

As $\hat{\boldsymbol{\beta}}$ is a minimizer of the objective function (4) and the joint sparse group penalty function in (4) is convex, Lemma 1 implies that

$$\begin{aligned} & \frac{1}{2n}\sum_{j=1}^{p}\|\boldsymbol{z}_j - \boldsymbol{W}_{-j}\hat{\boldsymbol{\beta}}_j\|_2^2 + \lambda_1\sum_{h=1}^{q}\|\hat{\boldsymbol{b}}_h\|_2 + \lambda_2\sum_{h=0}^{q}\|\hat{\boldsymbol{b}}_h\|_1 + \frac{1}{2n}\|\boldsymbol{W}_{-j}(\hat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j)\|_2^2 \\ \leq\ & \frac{1}{2n}\sum_{j=1}^{p}\|\boldsymbol{z}_j - \boldsymbol{W}_{-j}\boldsymbol{\beta}_j\|_2^2 + \lambda_1\sum_{h=1}^{q}\|\boldsymbol{b}_h\|_2 + \lambda_2\sum_{h=0}^{q}\|\boldsymbol{b}_h\|_1, \end{aligned}$$

where $\hat{\boldsymbol{b}}_h = ((\hat{\boldsymbol{\beta}}_1)_{(h)}, \ldots, (\hat{\boldsymbol{\beta}}_p)_{(h)})$, $h = \{0\} \cup [q]$. Writing $\boldsymbol{\Delta}_j = \hat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j$, $\boldsymbol{\epsilon}_j = \boldsymbol{z}_j - \mathbb{E}(\boldsymbol{z}_j)$ and reorganizing terms in the above inequality gives

$$\frac{1}{n}\sum_{j=1}^{p}\|\boldsymbol{W}_{-j}\boldsymbol{\Delta}_j\|_2^2 + \lambda_1\sum_{h=1}^{q}\|\hat{\boldsymbol{b}}_h\|_2 + \lambda_2\sum_{h=0}^{q}\|\hat{\boldsymbol{b}}_h\|_1 \leq \frac{1}{n}\sum_{j=1}^{p}\langle\boldsymbol{\epsilon}_j, \boldsymbol{W}_{-j}\boldsymbol{\Delta}_j\rangle + \lambda_1\sum_{h=1}^{q}\|\boldsymbol{b}_h\|_2 + \lambda_2\sum_{h=0}^{q}\|\boldsymbol{b}_h\|_1.$$

Let $\mathcal{W} = \begin{pmatrix} \boldsymbol{W}_{-1} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W}_{-2} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{W}_{-p} \end{pmatrix}$, $\boldsymbol{\epsilon} = (\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_p)$ and $\boldsymbol{\Delta} = (\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_p)$. Using the fact that

$$\begin{aligned} & \sum_{h=0}^{q}\|\hat{\boldsymbol{b}}_h\|_1 = \|\hat{\boldsymbol{\beta}}_{\mathcal{S}}\|_1 + \|\hat{\boldsymbol{\beta}}_{\mathcal{S}^c}\|_1, \quad \sum_{h=0}^{q}\|\boldsymbol{b}_h\|_1 = \|\boldsymbol{\beta}_{\mathcal{S}}\|_1 \\ & \sum_{h=1}^{q}\|\hat{\boldsymbol{b}}_h\|_2 = \sum_{j=1}^{p}\sum_{h\in\mathcal{G}}\|(\hat{\boldsymbol{\beta}}_j)_{(h)}\|_2 + \sum_{j=1}^{p}\sum_{h\in\mathcal{G}^c}\|(\hat{\boldsymbol{\beta}}_j)_{(h)}\|_2, \quad \sum_{h=1}^{q}\|\boldsymbol{b}_h\|_2 = \sum_{j=1}^{p}\sum_{h\in\mathcal{G}}\|(\boldsymbol{\beta}_j)_{(h)}\|_2 \end{aligned}$$

and applying the triangle inequalities to $\|\cdot\|_1$ and $\|\cdot\|_2$, we arrive at the following inequality

$$\frac{1}{n}\|\mathcal{W}\boldsymbol{\Delta}\|_2^2 + \lambda_1\|\boldsymbol{\Delta}_{(\mathcal{G}^c)}\|_{1,2} + \lambda_2\|\boldsymbol{\Delta}_{\mathcal{S}^c}\|_1 \leq \frac{1}{n}\langle\boldsymbol{\epsilon}, \mathcal{W}\boldsymbol{\Delta}\rangle + \lambda_1\|\boldsymbol{\Delta}_{(\mathcal{G})}\|_{1,2} + \lambda_2\|\boldsymbol{\Delta}_{\mathcal{S}}\|_1, \qquad \text{(A3)}$$

where $\|\boldsymbol{\Delta}_{(\mathcal{G})}\|_{1,2} = \sum_{j=1}^{p}\sum_{h\in\mathcal{G}}\|(\boldsymbol{\Delta}_j)_{(h)}\|_2$ and $\|\boldsymbol{\Delta}_{(\mathcal{G}^c)}\|_{1,2} = \sum_{j=1}^{p}\sum_{h\in\mathcal{G}^c}\|(\boldsymbol{\Delta}_j)_{(h)}\|_2$.

Defining $\hat{\mathcal{S}} = \{l : (\hat{\boldsymbol{\beta}})_l \neq 0, l \in [p(p-1)(q+1)]\}$ and letting $\tilde{\mathcal{S}} = \mathcal{S} \cup \hat{\mathcal{S}}$, we may write

$$\begin{aligned} & \langle \boldsymbol{\epsilon}, \mathcal{W}\boldsymbol{\Delta} \rangle = \langle \boldsymbol{\epsilon}, \boldsymbol{\mathcal{P}}_{\mathcal{W}_{\tilde{\mathcal{S}}}} \mathcal{W}_{\tilde{\mathcal{S}}} \boldsymbol{\Delta}_{\tilde{\mathcal{S}}} \rangle \qquad \text{(A4)} \\ = \quad & \langle \boldsymbol{\mathcal{P}}_{\mathcal{W}_{\tilde{\mathcal{S}}}} \boldsymbol{\epsilon}, \mathcal{W}\boldsymbol{\Delta} \rangle \leq \frac{1}{2a_1} \|\mathcal{W}\boldsymbol{\Delta}\|_2^2 + \frac{a_1}{2} \|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\tilde{\mathcal{S}}}} \boldsymbol{\epsilon}\|_2^2, \end{aligned}$$

where $\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\tilde{\mathcal{S}}}}$ is the orthogonal projection matrix onto the column space of $\mathcal{W}_{\tilde{\mathcal{S}}}$, the first and second equalities hold as the nonzero support of $\boldsymbol{\Delta}$ is $\tilde{\mathcal{S}}$ and the last inequality comes from that $2ab \leq ta^2 + b^2/t$ for any $t > 0$. Due to the diagonal block structure of $\mathcal{W}$, some straightforward algebra gives that

$$\|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\tilde{\mathcal{S}}}} \boldsymbol{\epsilon}\|_2^2 = \sum_{j=1}^{p} \|\boldsymbol{\mathcal{P}}_{\boldsymbol{W}_{\tilde{\mathcal{S}}_j}} \boldsymbol{\epsilon}_j\|_2^2,$$

where $\boldsymbol{\mathcal{P}}_{\boldsymbol{W}_{\tilde{\mathcal{S}}_j}}$ is the orthogonal projection matrix onto the column space of $(\boldsymbol{W}_{-j})_{\tilde{\mathcal{S}}_j}$. Finding a tight bound on the stochastic term $\sum_{j=1}^{p} \|\boldsymbol{\mathcal{P}}_{\boldsymbol{W}_{\tilde{\mathcal{S}}_j}} \boldsymbol{\epsilon}_j\|_2^2$ is challenging as the p terms to be summed are dependent in a complex manner. This is different from the standard multi-task learning problem where the stochastic terms from p separate regressions $\boldsymbol{\mathcal{P}}_{\boldsymbol{W}_{\tilde{\mathcal{S}}_1}} \boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\mathcal{P}}_{\boldsymbol{W}_{\tilde{\mathcal{S}}_p}} \boldsymbol{\epsilon}_p$ are mutually independent. Under our setting, $\boldsymbol{\epsilon}_i$ dependent on $\boldsymbol{\mathcal{P}}_{\boldsymbol{W}_{\tilde{\mathcal{S}}_j}} \boldsymbol{\epsilon}_j$, $j \neq i$, through $\boldsymbol{\mathcal{P}}_{\boldsymbol{W}_{\tilde{\mathcal{S}}_j}}$, as there is $\boldsymbol{z}_i$ in $\boldsymbol{W}_{-j}$ and also through $\boldsymbol{\epsilon}_j$, when $\boldsymbol{\Omega}(\boldsymbol{u})_{ij} \neq 0$. Moreover, the group lasso penalty term in (4) is not decomposable with respect to $\mathcal{S}$; hence, classic techniques based on the decomposable regularizers and null space properties are not applicable. Our analysis utilizes a novel result in Corollary 1 that gives a new tail bound for the sum of dependent chi-squared variables, and a delicate treatment of the stochastic term using the statistical properties and the computational optimality of the sparse group lasso estimator. We divide our arguments into the following steps.

Step 1: Given any $\mathcal{J} \subset [p(p-1)(q+1)]$ and $\boldsymbol{\gamma} \in \{0,1\}^{p(p-1)(q+1)}$ satisfying $\boldsymbol{\gamma}_{\mathcal{J}} = \mathbf{1}$ and $\boldsymbol{\gamma}_{\mathcal{J}^c} = \mathbf{0}$, we let $\mathcal{G}(\mathcal{J}) = \{h : (\boldsymbol{\gamma})_{(h)} \neq \mathbf{0}, h \in [q]\}$. In this step, we aim to show that, given

any $0 \leq s_g \leq q$ and $0 \leq s_e \leq p(p-1)(q+1)$, the following holds

$$\mathbb{P}\left(\sup_{|\mathcal{J}|=s_e,\, |\mathcal{G}(\mathcal{J})|=s_g} \|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\mathcal{J}}}\boldsymbol{\epsilon}\|_2^2 \geq 6\left[s_g \log(eq/s_g) + 2s_e \log(ep)\right] + t_0\right) \leq c_1 \exp(-c_2 t_0),$$

where c_1, c_2 are positive constants.

First, we find the size of $\{\mathcal{J} \subset [p(p-1)(q+1)], |\mathcal{J}| = s_e,\, |\mathcal{G}(\mathcal{J})| = s_g\}$ by considering (i) $s_g = s_e$ and (ii) $s_g < s_e$, the only two possible scenarios because the number of nonzero elements s_e cannot be less than the number of nonzero groups s_g.

case (i): $s_g = s_e$. In this case, the set $\{\mathcal{J} \subset [p(p-1)(q+1)], |\mathcal{J}| = s_e,\, |\mathcal{G}(\mathcal{J})| = s_g\}$ contains $\binom{q}{s_g}\{p(p-1)\}^{s_e}$ elements. It follows from Stirling's approximation that $\log\binom{q}{s_g} \leq s_g \log(eq/s_g)$. Therefore, $\log\left[\binom{q}{s_g}[p(p-1)]^{s_e}\right] \leq s_g \log(eq/s_g) + 2s_e \log p$.

case (ii): $s_g < s_e$. The number of elements in $\{\mathcal{J} \subset [p(p-1)(q+1)], |\mathcal{J}| = s_e, |\mathcal{G}(\mathcal{J})| = s_g\}$ is bounded above by $\binom{q}{s_g}\binom{p(p-1)+p(p-1)s_g}{s_e}$. By Stirling's approximation, we have

$$\log\binom{p(p-1)(s_g+1)}{s_e} \leq s_e \log(ep(p-1)(s_g+1)/s_e) \leq 2s_e \log(ep).$$

Therefore, we have $\log\left\{\binom{q}{s_g}\binom{p(p-1)(s_g+1)}{s_e}\right\} \leq s_g \log(eq/s_g) + 2s_e \log(ep)$.

Combining these two cases, we conclude that $\log|\{\mathcal{J} \subset [p(p-1)(q+1)], |\mathcal{J}| = s_e, |\mathcal{G}(\mathcal{J})| = s_g\}|$ is bounded above by $s_g \log(eq/s_g) + 2s_e \log(ep)$.

Let $\mathcal{J} = \mathcal{J}_1 \cup \cdots \cup \mathcal{J}_p$ such that $\boldsymbol{\beta}_{\mathcal{J}_j}$ is a sub-vector of $\boldsymbol{\beta}_j$. As the projection matrix $\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\mathcal{J}_j}}$ is idempotent, Lemma 2 implies that

$$\|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\mathcal{J}_j}}\boldsymbol{\epsilon}\|_2^2 \sim \chi^2_{d_j},$$

where $d_j = \min(|\mathcal{J}_j|, n)$. The above result hold as $\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\mathcal{J}_j}}\boldsymbol{\epsilon}$ involves only a subset of $\boldsymbol{\epsilon}_j$, and the elements of $\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\mathcal{J}_j}}\boldsymbol{\epsilon}$ have equal variances. Letting $h_n = s_g \log(eq/s_g) + 2s_e \log(ep)$, we

have

$$
\begin{aligned}
& \mathbb{P}\left[\sup_{|\mathcal{J}|=s_e,|\mathcal{G}(\mathcal{J})|=s_g} \|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\mathcal{J}}}\boldsymbol{\epsilon}\|_2^2 \geq 6(d_0+1)\left\{s_g\log(eq/s_g)+2s_e\log(ep)\right\}+t_0\right] \\
\leq\ & \exp[s_g\log(eq/s_g)+2s_e\log(ep)]\times \\
& \mathbb{P}\left[\sum_{j=1}^{p}\|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\mathcal{J}_j}}\boldsymbol{\epsilon}\|_2^2 \geq 6(d_0+1)\left\{s_g\log(eq/s_g)+2s_e\log(ep)\right\}+t_0\right] \\
\leq\ & c_G\exp\left[h_n-\frac{\{6(d_0+1)h_n+t_0-(c_G-1)s_e\}^2}{4c_G\{6(d_0+1)h_n+t_0+s_e\}}\right] \\
\leq\ & c_G\exp\left[h_n-\frac{\{(5d_0+6)h_n+t_0\}^2}{4c_G\{(6d_0+7)h_n+t_0\}}\right] \\
\leq\ & c_G\exp\left[h_n-\frac{\{(5d_0+6)h_n+t_0\}^2}{4c_G\{(6d_0+36/5)h_n+(6/5)t_0\}}\right] \\
\leq\ & c_G\exp\left[h_n-\frac{(5d_0+6)h_n+t_0}{(24/5)(d_0+1)}\right] \\
\leq\ & c_1\exp(-c_2t_0),
\end{aligned}
$$

for some positive constants c_1, c_2 . In the above derivations, the first inequality holds due to the union bound; the second inequality holds by applying Corollary 1, where we set $t=t_0+6(d_0+1)h_n$ and note that $\sum_j d_j=s_e$, $c_G\leq d_0+1$. In particular, $\|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\mathcal{J}_j}}\boldsymbol{\epsilon}\|_2^2$'s are correlated chi-squared variables with $\mathrm{Cov}(\|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\mathcal{J}_j}}\boldsymbol{\epsilon}\|_2^2,\|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\mathcal{J}_k}}\boldsymbol{\epsilon}\|_2^2)\neq 0$ when $\boldsymbol{\Omega}(\boldsymbol{u})_{j,k}\neq 0$, and the maximum number of nonzero entries in the columns of $\boldsymbol{\Omega}(\boldsymbol{u})$ is bounded by d_0, as specified in Assumption 3. The third and fifth inequalities use the fact that $h_n\geq s_e$ and $c_G\leq d_0+1$.

Step 2: Using the result from Step 1, we next find an upper bound for $\|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\tilde{\mathcal{S}}}}\boldsymbol{\epsilon}\|_2^2$. First, define

$$
r_{s_e,s_g}=\left[\sup_{|\mathcal{J}|=s_e,|\mathcal{G}(\mathcal{J})|=s_g}\|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\mathcal{J}}}\boldsymbol{\epsilon}\|_2^2-9(d_0+1)\left\{s_g\log(eq/s_g)+2s_e\log(ep)\right\}\right]_+,
$$

and $r=\sup_{1\leq s_e\leq p(p-1)(q+1),\,0\leq s_g\leq q} r_{s_e,s_g}$. It holds that

$$
\begin{aligned}
\|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\tilde{\mathcal{S}}}}\boldsymbol{\epsilon}\|_2^2 &\leq 9(d_0+1)\left\{\tilde{s}_g\log(eq/s_g)+2\tilde{s}_e\log(ep)\right\}+r \qquad \text{(A5)}\\
&\leq 9(d_0+1)\{(s_g+\hat{s}_g)\log(eq/s_g)+2(s_e+\hat{s}_e)\log(ep)\}+r.
\end{aligned}
$$

The result from Step 1 gives

$$\begin{aligned}\mathbb{P}\{r \geq t_0\} &\leq \sum_{s_e=1}^{p(p-1)(q+1)} \sum_{s_g=0}^{q} \mathbb{P}\{r_{s_e,s_g} \geq t_0\} \\ &\leq \sum_{s_e=1}^{p(p-1)(q+1)} \sum_{s_g=0}^{q} c_1 \exp[-c_2 t_0 - 9(d_0+1)c_2 \{s_g \log(eq/s_g) + 2s_e \log(ep)\}].\end{aligned}$$

Step 3: We derive an inequality for $\|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\hat{\mathcal{S}}}}\boldsymbol{\epsilon}\|_2^2$ by utilizing the computational optimality of $\hat{\boldsymbol{\beta}}$ in this step. Since the objective function is convex, $\hat{\boldsymbol{\beta}}$ is a stationary point of

$$\frac{1}{2n}\sum_{j=1}^{p}\|\boldsymbol{z}_j - \boldsymbol{W}_{-j}\boldsymbol{\beta}_j\|^2 + \lambda_1 \sum_{h=1}^{q}\|\boldsymbol{b}_h\|_2 + \lambda_2 \sum_{h=0}^{q}\|\boldsymbol{b}_h\|_1.$$

By the KKT conditions, for any $l \in \hat{\mathcal{S}}_j \cap (0)$, $(\hat{\boldsymbol{\beta}})_l$ must satisfy that

$$\lambda_2 \text{sign}\{(\hat{\boldsymbol{\beta}})_l\} = \frac{1}{n}\langle \mathcal{W}_{\cdot l}, \boldsymbol{z}_j - \boldsymbol{W}_{-j}\hat{\boldsymbol{\beta}}_j\rangle. \tag{A6}$$

Similarly, for any $l \in \hat{\mathcal{S}}_j \cap (h)$, $h \in [q]$, $(\hat{\boldsymbol{\beta}})_l$ must satisfy that

$$\lambda_2 \text{sign}\{(\hat{\boldsymbol{\beta}})_l\} + \lambda_1 \frac{(\hat{\boldsymbol{\beta}})_l}{\|\boldsymbol{b}_h\|_2} = \frac{1}{n}\langle \mathcal{W}_{\cdot l}, \boldsymbol{z}_j - \boldsymbol{W}_{-j}\hat{\boldsymbol{\beta}}_j\rangle. \tag{A7}$$

Squaring both sides of (A6) and (A7) and summing over all $l \in \hat{\mathcal{S}}$ gives

$$\lambda_1^2 \hat{s}_g + \lambda_2^2 \hat{s}_e \leq \frac{1}{n^2}\sum_{j=1}^{p}\|\boldsymbol{W}_{\hat{\mathcal{S}}_j}^{\top}(\boldsymbol{z}_j - \boldsymbol{W}_{-j}\hat{\boldsymbol{\beta}}_j)\|_2^2,$$

where we have used the fact that $\text{sign}\{(\hat{\boldsymbol{\beta}})_l\}(\hat{\boldsymbol{\beta}})_l \geq 0$.

Next, consider $\boldsymbol{W}_{i\cdot}^{\top}\boldsymbol{v}$, where $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p)$, $\boldsymbol{v}_l \in \mathbb{R}^{q+1}$ and $\|\boldsymbol{v}\| = 1$. We have $\boldsymbol{W}_{i\cdot} = (z_1^{(i)}, z_1^{(i)}u_1^{(i)}, \ldots, z_1^{(i)}u_q^{(i)}, \ldots, z_p^{(i)}, z_p^{(i)}u_1^{(i)}, \ldots, z_p^{(i)}u_q^{(i)})$. With slight overuse of notation, we include the intercept term into $\boldsymbol{u}^{(i)}$ in the subsequent development. Letting $\boldsymbol{V} = [\boldsymbol{v}_1^{\top}, \ldots, \boldsymbol{v}_p^{\top}] \in \mathbb{R}^{(q+1)\times p}$, we can reexpress $\boldsymbol{W}_{i\cdot}\boldsymbol{v}$ as $\boldsymbol{W}_{i\cdot}\boldsymbol{v} = \boldsymbol{u}^{(i)\top}\boldsymbol{V}\boldsymbol{z}^{(i)}$, $i \in [n]$.

Consequently, by the law of total expectation and Assumption 1, we have

$$\begin{aligned}
\mathbb{E}\left(\boldsymbol{v}^\top \frac{\boldsymbol{W}^\top \boldsymbol{W}}{n}\boldsymbol{v}\right) &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left(\boldsymbol{u}^{(i)\top}\boldsymbol{V}\boldsymbol{z}^{(i)}\right)^2 \\
&= \mathbb{E}\left[\mathbb{E}\left\{\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left(\boldsymbol{u}^{(i)\top}\boldsymbol{V}\boldsymbol{z}^{(i)}\right)^2 \Big| \{\boldsymbol{u}^{(i)}\}_{i\in[n]}\right\}\right] \\
&= \mathbb{E}\left\{\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{u}^{(i)\top}\boldsymbol{V}\boldsymbol{\Sigma}(\boldsymbol{u}^{(i)})\boldsymbol{V}^\top\boldsymbol{u}^{(i)}\right\} \\
&\geq \phi_1 \times \mathbb{E}\left\{\mathrm{tr}\left(\boldsymbol{u}^{(1)\top}\boldsymbol{V}\boldsymbol{V}^\top\boldsymbol{u}^{(1)}\right)\right\} \\
&\geq \phi_1 \times \lambda_{\min}\left(\mathrm{Cov}(\boldsymbol{u}^{(1)})\right)\mathrm{tr}(\boldsymbol{V}\boldsymbol{V}^\top) = \phi_1/\phi_0,
\end{aligned} \tag{A8}$$

where we have used the fact $\mathrm{tr}(\boldsymbol{A}\boldsymbol{B}) \geq \lambda_{\min}(\boldsymbol{A})\mathrm{tr}(\boldsymbol{B})$ for positive semi-definite matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ and $\mathrm{tr}(\boldsymbol{V}\boldsymbol{V}^\top) = 1$. By the Cauchy–Schwarz inequality, we deduce

$$\max_{j,k}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\{(\boldsymbol{W}_{ij}\boldsymbol{W}_{ik})^2\} = \max_{l_1,l_2,l_3,l_4}\mathbb{E}\left(z_{l_1}^{(1)^2}z_{l_2}^{(1)^2}u_{l_3}^{(1)^2}u_{l_4}^{(1)^2}\right) = \mathcal{O}(1), \tag{A9}$$

where we have used the fact that $z_{l_1}^{(i)}$ and $u_{l_2}^{(i)}$ have bounded eighth moments, as they are both sub-Gaussian with a bounded sub-Gaussian norm. Recall $|\hat{\mathcal{S}}| < s_\lambda$. By the condition on s_λ in Theorem 3, it then follows from Lemma 5, (A8) and Assumption 3 that, with probability at least $1 - C_0\exp\{-(\log p + \log q)\}$, we have $\|\boldsymbol{W}_{\hat{\mathcal{S}}}\|^2/n \leq M_1$ for some $M_1 > 0$. Since $\hat{\mathcal{S}} \in \tilde{\mathcal{S}}$, it holds that

$$\begin{aligned}
\lambda_1^2\hat{s}_g + \lambda_2^2\hat{s}_e &\leq \frac{M_1}{n}\sum_{j=1}^{p}\|\boldsymbol{W}_{-j}\boldsymbol{\Delta}_j + \boldsymbol{\mathcal{P}}_{\boldsymbol{W}_{\tilde{\mathcal{S}}_j}}\boldsymbol{\epsilon}_j\|_2^2 \\
&\leq \frac{2M_1}{n}\|\mathcal{W}\boldsymbol{\Delta}\|_2^2 + \frac{2M_1}{n}\|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\tilde{\mathcal{S}}}}\boldsymbol{\epsilon}\|_2^2,
\end{aligned} \tag{A10}$$

with probability $1 - d_1\exp(\log p - d_2 n)$.

Combining (A5) and (A10) and letting

$$\lambda_1 = C\sqrt{\log(eq/s_g)/n + 2s_e\log(ep)/(ns_g)}, \quad \lambda_2 = \sqrt{s_g/s_e}\lambda_1,$$

where $C = 3M_1\{(d_0+1)a_2\}^{1/2}$ for some $a_2 > 0$, we obtain that

$$(1 - \frac{2}{a_2})\|\boldsymbol{\mathcal{P}}_{\mathcal{W}_{\tilde{\mathcal{S}}}}\boldsymbol{\epsilon}\|_2^2 \leq 9d_0\left\{s_g\log(eq/s_g) + 2s_e\log(ep)\right\} + \frac{2}{a_2}\|\mathcal{W}\boldsymbol{\Delta}\|_2^2 + r \tag{A11}$$

holds with probability $1 - d_1 \exp(\log p - d_2 n)$. This, together with (A3) and (A4), implies that

$$\begin{aligned} & \frac{\|\mathcal{W}\boldsymbol{\Delta}\|_2^2}{n} + \lambda_1 \|\boldsymbol{\Delta}_{(\mathcal{G}^c)}\|_{1,2} + \lambda_2 \|\boldsymbol{\Delta}_{\mathcal{S}^c}\|_1 \qquad \text{(A12)} \\ \leq \; & \frac{1}{2a_1} \frac{\|\mathcal{W}\boldsymbol{\Delta}\|_2^2}{n} + \frac{9a_1 a_2}{2(a_2-2)} \frac{s_g \log(eq/s_g) + 2s_e \log(ep)}{n} \\ & + \frac{a_1}{a_2 - 2} \frac{\|\mathcal{W}\boldsymbol{\Delta}\|_2^2}{n} + \frac{a_1 a_2}{2(a_2-2)n} r + \lambda_1 \|\boldsymbol{\Delta}_{(\mathcal{G})}\|_{1,2} + \lambda_2 \|\boldsymbol{\Delta}_{\mathcal{S}}\|_1 \end{aligned}$$

holds with probability $1 - d_1 \exp(\log p - d_2 n)$.

Let $\boldsymbol{\Sigma}_{\mathcal{W}} = \mathbb{E}(\mathcal{W}^\top \mathcal{W}/n)$. It is easy to see from the definition of $\mathcal{W}$ that $\lambda_{min}(\boldsymbol{\Sigma}_{\mathcal{W}}) \geq \lambda_{min}(\boldsymbol{\Sigma}_{\boldsymbol{W}}) \geq 1/\phi_1$ in (A8). Next, we have that

$$\frac{\|\boldsymbol{\Delta}_{(\mathcal{G})}\|_{1,2}}{\sqrt{s_g}} + \frac{\|\boldsymbol{\Delta}_{\mathcal{S}}\|_1}{\sqrt{s_e}} \leq \|\boldsymbol{\Delta}_{(\mathcal{G})}\|_2 + \|\boldsymbol{\Delta}_{\mathcal{S}}\|_2 \leq 2\sqrt{\phi_1} \|\boldsymbol{\Sigma}_{\mathcal{W}}^{1/2} \boldsymbol{\Delta}\|_2,$$

where the first inequality is due to that $\|\boldsymbol{\Delta}_{(\mathcal{G})}\|_{1,2} \leq \sqrt{s_g}\|\boldsymbol{\Delta}_{(\mathcal{G})}\|_2$, $\|\boldsymbol{\Delta}_{\mathcal{S}}\|_1 \leq \sqrt{s_e}\|\boldsymbol{\Delta}_{\mathcal{S}}\|_2$ and the second inequality holds because $\|\boldsymbol{\Delta}_{(\mathcal{G})}\|_2 + \|\boldsymbol{\Delta}_{\mathcal{S}}\|_2 < 2\|\boldsymbol{\Delta}\|_2$. Consequently,

$$\lambda_1 \|\boldsymbol{\Delta}_{(\mathcal{G})}\|_{1,2} + \lambda_2 \|\boldsymbol{\Delta}_{\mathcal{S}}\|_1 \;\; \leq \;\; 2C\sqrt{\phi_1 e_n} \|\boldsymbol{\Sigma}_{\mathcal{W}}^{1/2} \boldsymbol{\Delta}\|_2 \leq a_3 C \phi_1 e_n + \frac{1}{a_3} \|\boldsymbol{\Sigma}_{\mathcal{W}}^{1/2} \boldsymbol{\Delta}\|_2^2,$$

where $e_n = \{s_g \log(eq/s_g) + 2s_e \log(ep)\}/n$ and the last inequality comes from that $2ab \leq ta^2 + b^2/t$ for any $t > 0$. Plugging this into (A12), we obtain

$$\begin{aligned} & \left\{1 - \frac{1}{2a_1} - \frac{a_1}{a_2 - 2}\right\} \frac{\|\mathcal{W}\boldsymbol{\Delta}\|_2^2}{n} \qquad \text{(A13)} \\ \leq \; & \left\{\frac{9a_1 a_2}{2(a_2-2)} + C a_3 \phi_1\right\} \frac{s_g \log(eq/s_g) + 2s_e \log(ep)}{n} + \frac{1}{a_3} \|\boldsymbol{\Sigma}_{\mathcal{W}}^{1/2} \boldsymbol{\Delta}\|_2^2 + \frac{a_1 a_2}{2(a_2-2)n} r \end{aligned}$$

holds with probability $1 - d_1 \exp(\log p - d_2 n)$.

Finally, we bound the distance between $\|\mathcal{W}\boldsymbol{\Delta}\|_2^2/n$ and $\|\boldsymbol{\Sigma}_{\mathcal{W}}^{1/2}\boldsymbol{\Delta}\|_2^2$. For $\boldsymbol{v} \in \mathbb{R}^{p(p-1)(q+1)}$, we write $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p)$ with $\boldsymbol{v}_j \in \mathbb{R}^{(p-1)(q+1)}$. To proceed, we first show with probability $1 - C' \exp\{-(\log p + \log q)\}$,

$$\sup_{\boldsymbol{v} \in \mathbb{K}_0(C_{\boldsymbol{\beta}}; s_e)} \left| \boldsymbol{v}^\top \left(\frac{\mathcal{W}^\top \mathcal{W}}{n} - \boldsymbol{\Sigma}_{\mathcal{W}} \right) \boldsymbol{v} \right| \leq 1/L, \qquad \text{(A14)}$$

where L is a sufficiently large constant and $\mathbb{K}_0(C_{\boldsymbol{\beta}}; s_e) = \{\boldsymbol{v} : \|\boldsymbol{v}_j\|_0 \leq 2C_{\boldsymbol{\beta}} s_j\,, \sum_{j=1}^p s_j = s_e, \text{and } \|\boldsymbol{v}\|_2 = 1\}$ for some positive constant $C_{\boldsymbol{\beta}}$.

Given (A9), it then follows from Lemma 5 and Assumption 3 that with probability $1 - C'\exp\{-(\log p + \log q)\}$,

$$\begin{aligned} & \left| \boldsymbol{v}^\top \left(\frac{\mathcal{W}^\top \mathcal{W}}{n} - \boldsymbol{\Sigma}_{\mathcal{W}} \right) \boldsymbol{v} \right| \\ = & \sum_{j=1}^p \left| \boldsymbol{v}_j^\top \left\{ \frac{\boldsymbol{W}_{-j}^\top \boldsymbol{W}_{-j}}{n} - \mathbb{E}\left(\frac{\boldsymbol{W}_{-j}^\top \boldsymbol{W}_{-j}}{n} \right) \right\} \boldsymbol{v}_j \right| \\ \precsim & \max_j \left\{ \sqrt{\frac{s_j \log(pq)}{n}} + \frac{s_j \log n \log(pq)}{n} \right\} = o(1), \end{aligned}$$

where we have used the fact that $\sum_{j=1}^p \|\boldsymbol{v}_j\|_2^2 = 1$. Thus, we have shown (A14) holds with probability at least $1 - C'\exp\{-(\log p + \log q)\}$. Combing this with the result in Lemma 6, we have, with probability at least $1 - C'\exp\{-(\log p + \log q)\}$,

$$\left| \boldsymbol{\Delta}^\top \left(\frac{\mathcal{W}^\top \mathcal{W}}{n} - \boldsymbol{\Sigma}_{\mathcal{W}} \right) \boldsymbol{\Delta} \right| \leq \frac{1}{L'} \left(\|\boldsymbol{\Delta}\|_2^2 + \frac{1}{s_e} \|\boldsymbol{\Delta}\|_1^2 \right), \tag{A15}$$

where L' is a sufficiently large positive constant. Plugging (A15) into (A13) and choosing proper constants a_1, a_2 and a_3, such as $a_1 = 2$, $a_2 = 6$ and $a_3 = 6$, we have

$$\frac{1}{2} \|\boldsymbol{\Sigma}_{\mathcal{W}}^{1/2} \boldsymbol{\Delta}\|_2^2 \precsim \frac{s_g \log(eq/s_g) + 2 s_e \log(ep)}{n} + \frac{1}{L'} \left(\|\boldsymbol{\Delta}\|_2^2 + \frac{1}{s_e} \|\boldsymbol{\Delta}\|_1^2 \right), \tag{A16}$$

with probability $1 - c_1 \exp[-c_2'\{s_e \log(ep) + s_g \log(eq/s_g)\}]$, due to that

$$\begin{aligned} & \mathbb{P}\left(r \geq M_0 \{ s_g \log(eq/s_g) + 2 s_e \log(ep) \} \right) \\ \leq & \; c_1 \exp(-c_2' \{ s_g \log(eq/s_g) + 2 s_e \log(ep) \}), \end{aligned}$$

for a large positive constant M_0.

Next, taking $a_1 = 2 - \sqrt{2}$ and $a_2 = 6$ in (A12) and using the expressions for λ_1, λ_2, we have with probability at least $1 - c_1 \exp(-c_2'\{s_g \log(eq/s_g) + 2 s_e \log(ep)\})$ that

$$\frac{\|\boldsymbol{\Delta}_{(\mathcal{G}^c)}\|_{1,2}}{\sqrt{s_g}} + \frac{\|\boldsymbol{\Delta}_{\mathcal{S}^c}\|_1}{\sqrt{s_e}} \leq \sqrt{\frac{s_g \log(eq/s_g) + 2 s_e \log(ep)}{n}} + \frac{\|\boldsymbol{\Delta}_{(\mathcal{G})}\|_{1,2}}{\sqrt{s_g}} + \frac{\|\boldsymbol{\Delta}_{\mathcal{S}}\|_1}{\sqrt{s_e}}. \tag{A17}$$

Adding $\|\boldsymbol{\Delta}_{\mathcal{S}}\|_1/\sqrt{s_e}$ to both sides of (A17), we have that

$$\frac{\|\boldsymbol{\Delta}\|_1}{\sqrt{s_e}} \leq \sqrt{e_n} + 3\|\boldsymbol{\Delta}\|_2. \tag{A18}$$

Plugging (A18) into (A16) and with $\lambda_{min}(\boldsymbol{\Sigma}_{\mathcal{W}}) \geq 1/\phi_1 > 0$ in Assumption 2, we have

$$\|\boldsymbol{\Delta}\|_2^2 \precsim \frac{1}{n}\{s_g \log(eq/s_g) + s_e \log(ep)\},$$

with probability at least $1 - C_1 \exp[-C_2\{s_g \log(eq/s_g) + s_e \log(ep)\}]$, for some positive constants C_1, C_2.

□

A5 Additional Results from Real Data Analysis

A5.1 The Kyoto Encyclopedia of Genes and Genomes Human Glioma Pathway

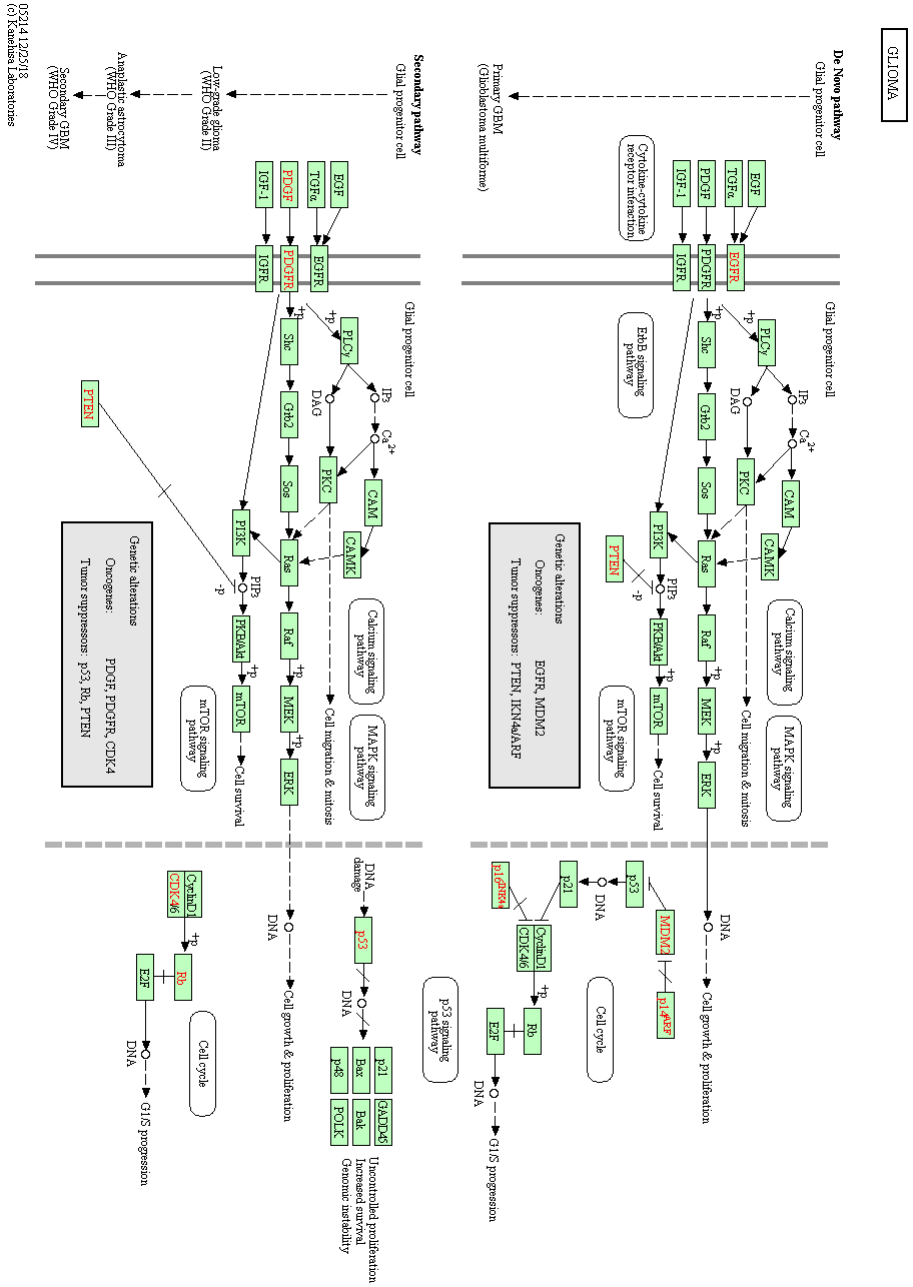

Figure A1: The Kyoto Encyclopedia of Genes and Genomes human glioma pathway. This figure is downloaded from `https://www.genome.jp/kegg/` (Kanehisa and Goto, 2000).

Additional references

Bellec, P. C., A. S. Dalalyan, E. Grappin, and Q. Paris (2018). On the prediction loss of the lasso in the partially labeled setting. *Electronic journal of statistics 12*(2), 3443–3472.

Graybill, F. A. and G. Marsaglia (1957). Idempotent matrices and quadratic forms in the general linear hypothesis. *The Annals of Mathematical Statistics 28*(3), 678–686.

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

Kuchibhotla, A. K. and A. Chakrabortty (2018). Moving beyond sub-gaussianity in high-dimensional statistics: Applications in covariance estimation and linear regression. *arXiv preprint arXiv:1804.02605*.

Laurent, B. and P. Massart (2000). Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, 1302–1338.

Loh, P.-L. and M. J. Wainwright (2011). High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. *Advances in neural information processing systems 24*.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.