## Supplementary Information A: Proofs of theorems

### Proof of Theorem 1

*Proof.* We first define the following quantities.

$$\mathbf{x}_s\mathbf{x}_s^T = \sum_i^m \mathbf{x}_i\mathbf{x}_i^T + \sum_{i<j}^m \left(\mathbf{x}_i\mathbf{x}_j^T + \mathbf{x}_j\mathbf{x}_i^T\right) \tag{4}$$

$$\mathbf{x}_{ij}\mathbf{x}_{ij}^T = \mathbf{x}_i\mathbf{x}_i^T + \mathbf{x}_j\mathbf{x}_j^T - \mathbf{x}_i\mathbf{x}_j^T - \mathbf{x}_j\mathbf{x}_i^T \tag{5}$$

$$\sum_{i<j}^m \mathbf{x}_{ij}\mathbf{x}_{ij}^T = (m-1)\sum_i^m \mathbf{x}_i\mathbf{x}_i^T - \sum_{i<j}^m \left(\mathbf{x}_i\mathbf{x}_j^T + \mathbf{x}_j\mathbf{x}_i^T\right) \tag{6}$$

$$\sum_i^m \mathbf{x}_i\mathbf{x}_i^T = \frac{1}{m}\left(\mathbf{x}_s\mathbf{x}_s^T + \sum_{i<j}^m \mathbf{x}_{ij}\mathbf{x}_{ij}^T\right) \qquad \text{From Eqs. (4) and (6)} \tag{7}$$

To prove the theorem, we visit each term in the model in Theorem 1, $\mathbf{y}^T\mathbf{\Lambda}\mathbf{y}$, $\mathbf{x}^T\mathbf{\Theta}\mathbf{y}$, $|\mathbf{\Lambda}|$, and $\exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{\Theta}\mathbf{\Lambda}^{-1}\mathbf{\Theta}^T\mathbf{x}\right)$ and re-write it as follows:

$$\begin{aligned}
\operatorname{tr}(\mathbf{y}\mathbf{y}^T\mathbf{\Lambda}) &= \operatorname{tr}(\left(\sum_i^m \mathbf{y}_i\mathbf{y}_i^T\right)(\mathbf{\Omega}+\mathbf{\Delta}) + \left(\sum_{i<j}^m \left(\mathbf{y}_i\mathbf{y}_j^T + \mathbf{y}_j\mathbf{y}_i^T\right)\right)\mathbf{\Omega}) \\
&= \operatorname{tr}(\left(\sum_i^m \mathbf{y}_i\mathbf{y}_i^T + \sum_{i<j}^m \left(\mathbf{y}_i\mathbf{y}_j^T + \mathbf{y}_j\mathbf{y}_i^T\right)\right)\mathbf{\Omega} + \left(\sum_i^m \mathbf{y}_i\mathbf{y}_i^T\right)\mathbf{\Delta}) \\
&= \operatorname{tr}(\mathbf{y}_s\mathbf{y}_s^T\mathbf{\Omega} + \left(\sum_i^m \mathbf{y}_i\mathbf{y}_i^T\right)\mathbf{\Delta}) \qquad \text{From Eq. (4)} \\
&= \operatorname{tr}(\mathbf{y}_s\mathbf{y}_s^T\mathbf{\Omega} + \frac{1}{m}\left(\mathbf{y}_s\mathbf{y}_s^T + \sum_{i<j}^m \mathbf{y}_{ij}\mathbf{y}_{ij}^T\right)\mathbf{\Delta}) \qquad \text{From Eq. (7)} \\
&= \operatorname{tr}(\mathbf{y}_s\mathbf{y}_s^T\left(\mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta}\right) + \left(\sum_{i<j}^m \mathbf{y}_{ij}\mathbf{y}_{ij}^T\right)\frac{1}{m}\mathbf{\Delta}) \\
\operatorname{tr}(\mathbf{y}\mathbf{x}^T\mathbf{\Theta}) &= \operatorname{tr}(\left(\sum_i^m \mathbf{y}_i\mathbf{x}_i^T\right)\mathbf{\Pi} + \left(\sum_i^m \mathbf{y}_i\mathbf{z}^T\right)\mathbf{\Xi}) \\
&= \operatorname{tr}(\mathbf{y}_s\mathbf{x}_s^T\left(\frac{1}{m}\mathbf{\Pi}\right) + \left(\sum_{i<j}^m \mathbf{y}_{ij}\mathbf{x}_{ij}^T\right)\frac{1}{m}\mathbf{\Pi} + \mathbf{y}_s\mathbf{z}^T\mathbf{\Xi}) \\
\operatorname{tr}(\mathbf{x}\mathbf{x}^T\mathbf{\Theta}\mathbf{\Sigma}\mathbf{\Theta}^T) &= \operatorname{tr}\Bigg(\mathbf{x}_s\mathbf{x}_s^T\left(\frac{1}{m}\mathbf{\Pi}\right)\left(\mathbf{\Omega}+\frac{1}{m}\mathbf{\Delta}\right)^{-1}\left(\frac{1}{m}\mathbf{\Pi}\right)^T \\
&\quad + \left(\sum_{i<j}^m \mathbf{x}_{ij}\mathbf{x}_{ij}^T\right)\frac{1}{m}\mathbf{\Pi}\left(\frac{1}{m}\mathbf{\Delta}\right)^{-1}\frac{1}{m}\mathbf{\Pi}^T \\
&\quad + \mathbf{z}\mathbf{z}^T\mathbf{\Xi}\left(\mathbf{\Omega}+\frac{1}{m}\mathbf{\Delta}\right)^{-1}\mathbf{\Xi}^T\Bigg) \\
\det(\mathbf{\Lambda}) &= \det(m\mathbf{\Omega}+\mathbf{\Delta})\det(\mathbf{\Delta})^{m-1}
\end{aligned}$$

The equation above on determinants follows from Theorem 3 in Silvester (2000). □

### Proof of Theorem 2

We first show that the following lemma holds, which is then used to prove Theorem 2.

**Lemma 1.** *Let* $\mathbf{K} = \mathbf{C}^{+}\mathbf{C} = \mathbf{C}^{T}\mathbf{C}^{+T} = \mathbf{K}^{T}$. *Then, for* $\mathbf{\Sigma} = \mathbf{\Lambda}^{-1} = [\mathbf{I}_{m\times m} \otimes \mathbf{\Delta} + \mathbf{1}_{m\times m} \otimes \mathbf{\Omega}]^{-1}$, *we have* $\mathbf{K\Sigma} = \mathbf{\Sigma K}$.

*Proof.* Using a block inversion of a matrix, we can show $\mathbf{\Sigma} = \mathbf{I}_{m\times m} \otimes \mathbf{E} + \mathbf{1}_{m\times m} \otimes \mathbf{\Phi}$, where $\mathbf{\Phi} = -\mathbf{\Delta}^{-1}\mathbf{\Omega}(m\mathbf{\Omega} + \boldsymbol{\delta})^{-1}$ and $\mathbf{E} = \mathbf{\Delta}^{-1}$, and verify this by checking $\mathbf{\Sigma\Lambda} = \mathbf{I}$. $\mathbf{K}$ is an orthogonal projection matrix with $\mathbf{K}_{i,i} = \mathbf{K}_{i,2i} = \mathbf{K}_{2i,i} = \mathbf{K}_{2i,2i} = \frac{1}{2}$ for $i \in H$ for outputs with missing individual-level observations. For observed outputs $i \in V$, $\mathbf{K}_{i,i} = \mathbf{K}_{2i,2i} = 1$, and $\mathbf{K}_{i,2i} = \mathbf{K}_{2i,i} = 0$. Then, using the repeat block structure of $\mathbf{\Sigma}$, we define all the elements of $\mathbf{K\Sigma}$ and $\mathbf{\Sigma K}$:

$$(\mathbf{K\Sigma})_{i,i} = (\mathbf{K\Sigma})_{i,2i} = (\mathbf{K\Sigma})_{2i,i} = (\mathbf{K\Sigma})_{2i,2i} = \mathbf{\Phi}_{i,i} + \frac{1}{2}\boldsymbol{\epsilon}_i \quad \text{for } i \in H, \text{ and}$$
$$(\mathbf{K\Sigma})_{i,i} = (\mathbf{K\Sigma})_{i,2i} = (\mathbf{K\Sigma})_{2i,i} = (\mathbf{K\Sigma})_{2i,2i} = \mathbf{\Phi}_{i,i} + \boldsymbol{\epsilon}_i \quad \text{for } i \in V.$$

For $(i,j) \in \{i,j : i \neq j, i = 1,\ldots,q, j = 1,\ldots,q\}$, $(\mathbf{K\Sigma})_{i,j} = (\mathbf{K\Sigma})_{i,2j} = (\mathbf{K\Sigma})_{2i,j} = (\mathbf{K\Sigma})_{2i,2j} = \mathbf{\Phi}_{i,j}$. The elements of $\mathbf{\Sigma K}$ are identical to those of $\mathbf{K\Sigma}$. □

Now we prove Theorem 2.

*Proof.* From our model in Eq. (2), we derive the probability distribution for the observed outputs. We begin by re-writing Eq. (2) in the form of Gaussian distribution to make the marginal distribution explicitly represented:

$$p(\mathbf{y} \mid \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) \sim \mathcal{N}(-\mathbf{C}_a\mathbf{\Lambda}^{-1}\mathbf{\Theta}^{T}\mathbf{x}, \mathbf{C}_a\mathbf{\Lambda}^{-1}\mathbf{C}_a), \tag{8}$$

where $\mathbf{C}_a = \begin{bmatrix} \mathbf{I}_{q\times q} & \mathbf{0}_{q\times q} \\ \mathbf{0}_{q\times q} & \mathbf{I}_{q\times q} \end{bmatrix}$. Then, the output sum variables are also Gaussian distributed:

$$p(\mathbf{y}_s \mid \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) \sim \mathcal{N}(-\mathbf{C}_s\mathbf{\Lambda}^{-1}\mathbf{\Theta}^{T}\mathbf{x}, \mathbf{C}_s\mathbf{\Lambda}^{-1}\mathbf{C}_s^{T}), \tag{9}$$

where $\mathbf{C}_s = [\mathbf{I}_{q\times q}\ \mathbf{I}_{q\times q}]$.

We combine the marginal distributions for the observed output variables in $V$ from Eq. (8) and for output variables in $H$ with missing individual-level observations but only with group-level sum data from Eq. (9) to form a joint distribution of the observed variables $\mathbf{y}_s^H$ and $\mathbf{y}^V$:

$$p(\mathbf{y}_s^H, \mathbf{y}^V \mid \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) \sim \mathcal{N}(-\mathbf{C\Lambda}^{-1}\mathbf{\Theta}^{T}\mathbf{x}, \mathbf{C\Lambda}^{-1}\mathbf{C}^{T}), \tag{10}$$

given $\mathbf{C} = \begin{bmatrix} \mathbf{C}_s^H \\ \mathbf{C}_a^V \end{bmatrix}$. $\mathbf{C}_s^H$ and $\mathbf{C}_a^V$ are the submatrices of $\mathbf{C}_s$ and $\mathbf{C}_a$ with only the rows corresponding to the variables in $\mathbf{y}_s^H$ and $\mathbf{y}^V$. To show the distribution in Eq. (10) is a CGGM, we only need to show Eq. (10) can be written as

$$p(\mathbf{y}_s^H, \mathbf{y}^V \mid \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) = \mathcal{N}\big(-\tilde{\mathbf{\Lambda}}^{-1}\tilde{\mathbf{\Theta}}^{T}\mathbf{x}, \tilde{\mathbf{\Lambda}}^{-1}\big), \tag{11}$$

by obtaining the explicit forms of the mean and inverse covariance of the above, since expanding the quadratic term in this distribution above leads to the CGGM in Theorem 2.

We first show $\tilde{\mathbf{\Lambda}} = [\mathbf{C\Lambda}^{-1}\mathbf{C}^{T}]^{-1} = \mathbf{C}^{+T}\mathbf{\Lambda}\mathbf{C}^{+}$, where $\mathbf{C}^{+}$ is the Moore-Penrose inverse of $\mathbf{C}$, by showing $\mathbf{C\Lambda}^{-1}\mathbf{C}^{T}\mathbf{C}^{+T}\mathbf{\Lambda}\mathbf{C}^{+} = \mathbf{I}$:

$$\begin{aligned}
\mathbf{C\Sigma C}^{T}\mathbf{C}^{+T}\mathbf{\Lambda}\mathbf{C}^{+} &= \mathbf{C}\mathbf{C}^{T}\mathbf{C}^{+T}\mathbf{\Lambda}^{-1}\mathbf{\Lambda}\mathbf{C}^{+} && \text{from Lemma 1} \\
&= \mathbf{C}\mathbf{C}^{T}\mathbf{C}^{+T}\mathbf{C}^{+} && \\
&= \mathbf{C}\mathbf{C}^{+}\mathbf{C}\mathbf{C}^{+} && \text{since } \mathbf{C}^{+}\mathbf{C} \text{ is symmetric from the definition of Moore-Penrose inverse} \\
&= \mathbf{C}\mathbf{C}^{+} && \text{from } \mathbf{C}\mathbf{C}^{+} = \mathbf{I} \\
&= \mathbf{I} &&
\end{aligned}$$

Then the mean of Eq. (10) can be written as

$$\begin{aligned}
-\mathbf{C}\mathbf{\Lambda}^{-1}\mathbf{\Theta}^T\mathbf{x} &= -\mathbf{C}\mathbf{C}^+\mathbf{C}\mathbf{\Lambda}^{-1}\mathbf{\Theta}^T\mathbf{x} && \text{from } \mathbf{C} = \mathbf{C}\mathbf{C}^+\mathbf{C} \\
&= -\mathbf{C}\mathbf{\Lambda}^{-1}\mathbf{C}^+\mathbf{C}\mathbf{\Theta}^T\mathbf{x} && \text{from Lemma 1} \\
&= -\mathbf{C}\mathbf{\Lambda}^{-1}\mathbf{C}^T\mathbf{C}^{+T}\mathbf{\Theta}^T\mathbf{x} && \text{from } \mathbf{C}^+\mathbf{C} = \mathbf{C}^T\mathbf{C}^{+T} \text{ since } \mathbf{C}^+\mathbf{C} \text{ is symmetric} \\
&= -\tilde{\mathbf{\Lambda}}^{-1}\tilde{\mathbf{\Theta}}^T\mathbf{x} && \tilde{\mathbf{\Theta}} = \mathbf{\Theta}\mathbf{C}^+
\end{aligned}$$

Thus, we have Eq. (11) with $\tilde{\mathbf{\Theta}} = \mathbf{\Theta}\mathbf{C}^+$ and $\tilde{\mathbf{\Lambda}} = \mathbf{C}^{+T}\mathbf{\Lambda}\mathbf{C}^+$.

□

The corollary below follows from the theorem above.

**Corollary 2.** *Learning the collapsed multi-level CGGM in Eq. (3) with randomly missing individual-level output data is equivalent to learning the full multi-level model with imputed data* $\tilde{\mathbf{y}} = \mathbf{C}^+ \begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix}$.

*Proof.* From Theorem 2, given $\tilde{\mathbf{\Lambda}} = \mathbf{C}^{+T}\mathbf{\Lambda}\mathbf{C}^+$ and $\tilde{\mathbf{\Theta}}^T = \mathbf{C}^{+T}\mathbf{\Theta}^T$, we can re-write the log probability as

$$\begin{aligned}
\log p(\mathbf{y}_s^H, \mathbf{y}^V \mid \mathbf{x}; \mathbf{\Lambda}, \mathbf{\Theta}) &= -\frac{1}{2}\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}^V \end{bmatrix}^T \tilde{\mathbf{\Lambda}} \begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}^V \end{bmatrix} - \mathbf{x}^T\tilde{\mathbf{\Theta}}\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix} - \log Z(\mathbf{x}) \\
&= -\frac{1}{2}\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix}^T \mathbf{C}^{+T}\mathbf{\Lambda}\mathbf{C}^+ \begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix} - \mathbf{x}^T\mathbf{\Theta}\mathbf{C}^+\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix} - \log Z(\mathbf{x}) \\
&= -\frac{1}{2}\left[\mathbf{C}^+\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix}\right]^T \mathbf{\Lambda}\left[\mathbf{C}^+\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix}\right] - \mathbf{x}^T\mathbf{\Theta}\left[\mathbf{C}^+\begin{bmatrix} \mathbf{y}_s^H \\ \mathbf{y}_v \end{bmatrix}\right] - \log Z(\mathbf{x}) \\
&= -\frac{1}{2}\tilde{\mathbf{y}}^T\mathbf{\Lambda}\tilde{\mathbf{y}} - \mathbf{x}^T\mathbf{\Theta}\tilde{\mathbf{y}} - \log Z(\mathbf{x}).
\end{aligned}$$

□

## Supplementary Information B: Details of the Optimization Method for Multi-level Conditional Gaussian Graphical Models

### Alternating Newton coordinate descent

We provide the details of the learning algorithm for our multi-level model, using the representation of the model in Theorem 1. The $L_1$-regularized negative log-likelihood is given as

$$\begin{aligned}
\underset{\mathbf{\Omega}\succ 0, \mathbf{\Delta}, \mathbf{\Xi}, \mathbf{\Pi}}{\operatorname{argmin}} \; & -\log|\mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta}| + \operatorname{tr}\Big(\mathbf{S}_{\mathbf{y}_s}(\mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta}) + 2\mathbf{S}_{\mathbf{y}\mathbf{x}_s}\frac{1}{m}\mathbf{\Pi} + 2\mathbf{S}_{\mathbf{y}\mathbf{z}}\mathbf{\Xi} \\
& + (\mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta})^{-1}\big(\frac{1}{m^2}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\mathbf{\Pi} + \frac{1}{m}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}\mathbf{z}}\mathbf{\Xi} + \frac{1}{m}\mathbf{\Xi}^T\mathbf{S}_{\mathbf{x}\mathbf{z}}^T\mathbf{\Pi} + \mathbf{\Xi}^T\mathbf{S}_{\mathbf{z}}\mathbf{\Xi}\big)\Big) \\
& - (m-1)\log|\mathbf{\Delta}| + \operatorname{tr}\Big(\mathbf{S}_{\mathbf{y}_d}\frac{1}{m}\mathbf{\Delta} + 2\mathbf{S}_{\mathbf{y}\mathbf{x}_d}\frac{1}{m}\mathbf{\Pi} + \frac{1}{m}\mathbf{\Delta}^{-1}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_d}\mathbf{\Pi}\Big) \\
& + \lambda_{\mathbf{\Omega}}\|\mathbf{\Omega}\|_1 + \lambda_{\mathbf{\Delta}}\|\mathbf{\Delta}\|_1 + \lambda_{\mathbf{\Pi}}\|\mathbf{\Pi}\|_1 + \lambda_{\mathbf{\Xi}}\|\mathbf{\Xi}\|_1\,,
\end{aligned} \tag{12}$$

using components of covariances $\mathbf{S}_{\mathbf{y}\mathbf{w}_s} = [\mathbf{S}_{\mathbf{y}\mathbf{x}_s}\mathbf{S}_{\mathbf{y}\mathbf{z}}]$ and $\mathbf{S}_{\mathbf{w}} = \begin{bmatrix} \mathbf{S}_{\mathbf{x}_s} & \mathbf{S}_{\mathbf{x}\mathbf{z}_s} \\ \mathbf{S}_{\mathbf{x}\mathbf{z}_s}^T & \mathbf{S}_{\mathbf{z}} \end{bmatrix}$.

In each iteration, we alternately optimize for each of the parameters $\mathbf{\Omega}$, $\mathbf{\Delta}$, $\mathbf{\Xi}$, and $\mathbf{\Pi}$ by solving the optimization problem above for each of the parameter given the previous estimates of all the other parameters. We maintain and update $\mathbf{V} = \mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta}$ after each update of $\mathbf{\Omega}$ and $\mathbf{\Delta}$ in order to avoid repeatedly computing $\mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta}$. In the rest of this section, we write $\mathbf{V} = \mathbf{\Omega} + \frac{1}{m}\mathbf{\Delta}$ whenever this term occurs.

**Algorithm 1:** Alternating Newton Coordinate Descent for Multi-level Conditional Gaussian Graphical Models

**Input:** $\mathbf{S}_{\mathbf{y}_d}, \mathbf{S}_{\mathbf{yx}_d}, \mathbf{S}_{\mathbf{x}_d}, \mathbf{Y}_s, \mathbf{X}_s, \mathbf{X}_d, \mathbf{Z}$, regularization parameters $\lambda_{\mathbf{\Omega}}, \lambda_{\mathbf{\Pi}}, \lambda_{\mathbf{\Xi}}$, and $\alpha$
**Initialize:** $\mathbf{\Pi}, \mathbf{\Xi} \leftarrow 0, \mathbf{\Omega}, \mathbf{\Delta} \leftarrow \mathbf{I}_{q\times q}$
**for** $t = 0, 1, \ldots$ **do**
  Determine active sets $\mathcal{S}_{\mathbf{\Omega}}, \mathcal{S}_{\mathbf{\Delta}}, \mathcal{S}_{\mathbf{\Pi}}, \mathcal{S}_{\mathbf{\Xi}}$
  Find Newton directions via coordinate descent
    $D_{\mathbf{\Omega}} = \underset{\delta_{\mathbf{\Omega}}}{\operatorname{argmin}}\, \bar{g}(\mathbf{\Omega} + \delta_{\mathbf{\Omega}}, \mathbf{\Delta}, \mathbf{\Pi}, \mathbf{\Xi}) + h(\mathbf{\Omega} + \delta_{\mathbf{\Omega}}, \mathbf{\Delta}, \mathbf{\Pi}, \mathbf{\Xi})$
  Update $\mathbf{\Omega}^+ = \mathbf{\Omega} + \alpha D_{\mathbf{\Omega}}$, where $\alpha$ is found via line search
    $D_{\mathbf{\Delta}} = \underset{\delta_{\mathbf{\Delta}}}{\operatorname{argmin}}\, \bar{g}(\mathbf{\Omega}, \mathbf{\Delta} + \delta_{\mathbf{\Delta}}, \mathbf{\Pi}, \mathbf{\Xi}) + h(\mathbf{\Omega}, \mathbf{\Delta} + \delta_{\mathbf{\Delta}}, \mathbf{\Pi}, \mathbf{\Xi})$
  Update $\mathbf{\Delta}^+ = \mathbf{\Delta} + \alpha D_{\mathbf{\Delta}}$, where $\alpha$ is found via line search
  Solve via coordinate descent:
    $\mathbf{\Pi}^+ = \underset{\mathbf{\Pi}}{\operatorname{argmin}}\, g(\mathbf{\Pi}) + h(\mathbf{\Pi})$
    $\mathbf{\Xi}^+ = \underset{\mathbf{\Xi}}{\operatorname{argmin}}\, g(\mathbf{\Xi}) + h(\mathbf{\Xi})$

**Update for $\mathbf{\Xi}$:** Optimizing for $\mathbf{\Xi}$ given all the other parameters corresponds to solving the following optimization problem:

$$\underset{\mathbf{\Xi}}{\operatorname{argmin}}\, g_{\mathbf{\Omega},\mathbf{\Delta},\mathbf{\Pi}}(\mathbf{\Xi}) + \lambda_{\mathbf{\Xi}}\|\mathbf{\Xi}\|_1\,,$$

where

$$g_{\mathbf{\Omega},\mathbf{\Delta},\mathbf{\Pi}}(\mathbf{\Xi}) = \operatorname{tr}\Big(2\mathbf{S}_{\mathbf{yz}}\mathbf{\Xi} + \mathbf{V}^{-1}\big(\frac{1}{m}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{xz}}\mathbf{\Xi} + \frac{1}{m}\mathbf{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\mathbf{\Pi} + \mathbf{\Xi}^T\mathbf{S}_{\mathbf{z}}\mathbf{\Xi}\big)\Big).$$

**Update for $\mathbf{\Pi}$:** Optimizing for $\mathbf{\Pi}$ given all the other parameters corresponds to solving the following optimization problem:

$$\underset{\mathbf{\Pi}}{\operatorname{argmin}}\, g_{\mathbf{\Omega},\mathbf{\Delta},\mathbf{\Xi}}(\mathbf{\Pi}) + \lambda_{\mathbf{\Pi}}\|\mathbf{\Pi}\|_1\,,$$

where

$$g_{\mathbf{\Omega},\mathbf{\Delta},\mathbf{\Xi}}(\mathbf{\Pi}) = \operatorname{tr}\Big(2\mathbf{S}_{\mathbf{yx}_s}\frac{1}{m}\mathbf{\Pi} + 2\mathbf{S}_{\mathbf{yx}_d}\frac{1}{m}\mathbf{\Pi} + \mathbf{V}^{-1}\big(\frac{1}{m^2}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\mathbf{\Pi} + \frac{1}{m}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{xz}}\mathbf{\Xi} + \frac{1}{m}\mathbf{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\mathbf{\Pi}\big) + \frac{1}{m}\mathbf{\Delta}^{-1}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_d}\mathbf{\Pi}\Big).$$

The two problems above for $\mathbf{\Xi}$ and $\mathbf{\Pi}$ are Lasso, which can be solved efficiently using a coordiante descent algorithm.

**Update for $\mathbf{\Omega}$:** To optimize Eq. (12) for $\mathbf{\Omega}$ given all the other parameters, we use the Newton's method. We first find the Newton descent direction by minimizing the second-order approximation of the objective in Eq. (12) with respect to $\mathbf{\Omega}$ and then update $\mathbf{\Omega}$ using this Newton direction and step size found by line search. The Newton direction $D_{\mathbf{\Omega}}$ is found by solving the following optimization problem:

$$D_{\mathbf{\Omega}} = \underset{\delta_{\mathbf{\Omega}}}{\operatorname{argmin}}\, \bar{g}_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\delta_{\mathbf{\Omega}}) + \lambda_{\mathbf{\Omega}}\|\mathbf{\Omega} + \delta_{\mathbf{\Omega}}\|_1\,,$$

where $\bar{g}_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\delta_{\mathbf{\Omega}})$ is the second-order Taylor expansion of the data log-likelihood in Eq. (12) with respect to $\mathbf{\Omega}$ and is given as

$$\bar{g}_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\delta_{\mathbf{\Omega}}) = \operatorname{vec}(\nabla_{\mathbf{\Omega}} g_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\mathbf{\Omega}))^T\operatorname{vec}(\delta_{\mathbf{\Omega}}) + \frac{1}{2}\operatorname{vec}(\delta_{\mathbf{\Omega}})^T\nabla^2_{\mathbf{\Omega}} g_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\mathbf{\Omega})\operatorname{vec}(\delta_{\mathbf{\Omega}}),$$

with the gradient and Hessian

$$\nabla_{\mathbf{\Omega}} g_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\mathbf{\Omega}) = \mathbf{S}_{\mathbf{y}_s} - \mathbf{V}^{-1} - \mathbf{V}^{-1}\big(\frac{1}{m^2}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\mathbf{\Pi} + \frac{1}{m}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{xz}}\mathbf{\Xi} + \frac{1}{m}\mathbf{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\mathbf{\Pi} + \mathbf{\Xi}^T\mathbf{S}_{\mathbf{z}}\mathbf{\Xi}\big)\mathbf{V}^{-1}$$

$$\nabla^2_{\mathbf{\Omega}} g_{\mathbf{\Delta},\mathbf{\Xi},\mathbf{\Pi}}(\mathbf{\Omega}) = \mathbf{V}^{-1} \otimes \left(\mathbf{V}^{-1} + 2\mathbf{V}^{-1}\big(\frac{1}{m^2}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\mathbf{\Pi} + \frac{1}{m}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{xz}}\mathbf{\Xi} + \frac{1}{m}\mathbf{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\mathbf{\Pi} + \mathbf{\Xi}^T\mathbf{S}_{\mathbf{z}}\mathbf{\Xi}\big)\mathbf{V}^{-1}\right).$$

The above optimization problem is a Lasso problem and can be solved using a coordinate descent method.

**Update for $\boldsymbol{\Delta}$:** To optimize Eq. (12) for $\boldsymbol{\Delta}$ given all the other parameters, we again use the Newton's method. We find the Newton descent direction $D_{\boldsymbol{\Delta}}$ by minimizing a second-order approximation of the objective in Eq. (12) with respect to $\boldsymbol{\Delta}$ and update $\boldsymbol{\Delta}$ with this Newton direction and a step size found by line search. The Newton descent direction $D_{\boldsymbol{\Delta}}$ can be found by solving the following:

$$D_{\boldsymbol{\Delta}} = \underset{\delta_{\boldsymbol{\Delta}}}{\operatorname{argmin}}\, \bar{g}_{\boldsymbol{\Omega},\boldsymbol{\Xi},\boldsymbol{\Pi}}(\delta_{\boldsymbol{\Delta}}),$$

where

$$\bar{g}_{\boldsymbol{\Omega},\boldsymbol{\Xi},\boldsymbol{\Pi}}(\delta_{\boldsymbol{\Delta}}) = \operatorname{vec}(\nabla_{\boldsymbol{\Delta}} g_{\boldsymbol{\Omega},\boldsymbol{\Xi},\boldsymbol{\Pi}}(\boldsymbol{\Delta}))^T \operatorname{vec}(\delta_{\boldsymbol{\Delta}}) + \frac{1}{2}\operatorname{vec}(\delta_{\boldsymbol{\Delta}})^T \nabla^2_{\boldsymbol{\Delta}} g_{\boldsymbol{\Omega},\boldsymbol{\Xi},\boldsymbol{\Pi}}(\boldsymbol{\Delta}) \operatorname{vec}(\delta_{\boldsymbol{\Delta}}),$$

with the gradient and Hessian

$$\begin{aligned}
\nabla_{\boldsymbol{\Delta}} g_{\boldsymbol{\Omega},\boldsymbol{\Xi},\boldsymbol{\Pi}}(\boldsymbol{\Delta}) = \frac{1}{m}\Big(&\mathbf{S}_{\mathbf{y}_s} + \mathbf{S}_{\mathbf{y}_d} - \mathbf{V}^{-1} - (m-1)\boldsymbol{\Gamma}^{-1} \\
&- \mathbf{V}^{-1}\big(\frac{1}{m^2}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\boldsymbol{\Pi} + \frac{1}{m}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{xz}}\boldsymbol{\Xi} + \frac{1}{m}\boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\boldsymbol{\Pi} + \boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{z}}\boldsymbol{\Xi}\big)\mathbf{V}^{-1} \\
&- \frac{1}{m^2}\boldsymbol{\Delta}^{-1}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{x}_d}\boldsymbol{\Pi}\boldsymbol{\Delta}^{-1}\Big) \\
\nabla^2_{\boldsymbol{\Delta}} g_{\boldsymbol{\Omega},\boldsymbol{\Xi},\boldsymbol{\Pi}}(\boldsymbol{\Delta}) = \frac{1}{m^2}\Bigg[&\mathbf{V}^{-1} \otimes \big(\mathbf{V}^{-1} + 2\mathbf{V}^{-1}\big(\frac{1}{m^2}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\boldsymbol{\Pi} + \frac{1}{m}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{xz}}\boldsymbol{\Xi} + \frac{1}{m}\boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\boldsymbol{\Pi} + \boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{z}}\boldsymbol{\Xi}\big)\mathbf{V}^{-1}\big) \\
&+ \boldsymbol{\Gamma}^{-1} \otimes \left((m-1)\boldsymbol{\Gamma}^{-1} + \frac{2}{m^2}\boldsymbol{\Delta}^{-1}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{x}_d}\boldsymbol{\Pi}\boldsymbol{\Delta}^{-1}\right)\Bigg].
\end{aligned}$$

This optimization problem for finding a Newton direction is again a Lasso problem, which can be solved using a coordinate descent algorithm.

In order to improve the efficiency of the coordinate descent algorithm, we extend the strategies used in McCarter and Kim (2016). First, we restrict the coordinate descent updates to the active set of variables given as

$$\begin{aligned}
\mathcal{S}_{\boldsymbol{\Omega}} &= \{(\delta_{\boldsymbol{\Omega}})_{ij} : |\left(\nabla_{\boldsymbol{\Omega}} g(\boldsymbol{\Omega})\right)_{ij}| > \lambda_{\boldsymbol{\Omega}} \vee \boldsymbol{\Omega}_{ij} \neq 0\} \\
\mathcal{S}_{\boldsymbol{\Pi}} &= \{\boldsymbol{\Pi}_{ij} : |\left(\nabla_{\boldsymbol{\Pi}} g(\boldsymbol{\Pi})\right)_{ij}| > \lambda_{\boldsymbol{\Pi}} \vee \boldsymbol{\Pi}_{ij} \neq 0\} \\
\mathcal{S}_{\boldsymbol{\Xi}} &= \{\boldsymbol{\Xi}_{ij} : |\left(\nabla_{\boldsymbol{\Xi}} g(\boldsymbol{\Xi})\right)_{ij}| > \lambda_{\boldsymbol{\Xi}} \vee \boldsymbol{\Xi}_{ij} \neq 0\} \\
\mathcal{S}_{\boldsymbol{\Delta}} &= \{(\delta_{\boldsymbol{\Delta}})_{ij} : |\left(\nabla_{\boldsymbol{\Delta}} g(\boldsymbol{\Delta})\right)_{ij}| > \lambda_{\boldsymbol{\Delta}} \vee \boldsymbol{\Delta}_{ij} \neq 0\}.
\end{aligned}$$

Second, we compute and store the intermediate results at the beginning of the coordinate descent updates for $\boldsymbol{\Xi}$ and $\boldsymbol{\Pi}$ and for $\delta_{\boldsymbol{\Omega}}$ and $\delta_{\boldsymbol{\Delta}}$ in each Newton iteration. We store the intermediate results $U_{\boldsymbol{\Xi}} \coloneqq \boldsymbol{\Xi}\mathbf{V}^{-1}$ for $\boldsymbol{\Xi}$, $U_{\boldsymbol{\Pi}_1} \coloneqq \boldsymbol{\Pi}\mathbf{V}^{-1}$ and $U_{\boldsymbol{\Pi}_2} \coloneqq \boldsymbol{\Pi}\boldsymbol{\Delta}^{-1}$ for $\boldsymbol{\Pi}$, $U_{\boldsymbol{\Omega}} \coloneqq \delta_{\boldsymbol{\Omega}}\mathbf{V}^{-1}$ for $\delta_{\boldsymbol{\Omega}}$, and $U_{\boldsymbol{\delta}_1} \coloneqq \delta_{\boldsymbol{\Delta}}\mathbf{V}^{-1}$ and $U_{\boldsymbol{\delta}_2} \coloneqq \delta_{\boldsymbol{\Delta}}\boldsymbol{\Delta}^{-1}$ for $\delta_{\boldsymbol{\Delta}}$. When $\boldsymbol{\Xi}_{ij}$ and $\boldsymbol{\Pi}_{ij}$ are updated, the $i$th row of the corresponding intermediate results is updated. After $(\delta_{\boldsymbol{\Omega}})_{ij}$ update, the $i$th and $j$th rows of $U_{\boldsymbol{\Omega}}$ are updated and after $(\delta_{\boldsymbol{\Delta}})_{ij}$ update, the $i$th and $j$th rows of $U_{\boldsymbol{\Delta}}$ is updated.

### Alternating Newton block coordinate descent

We further adopt the block-wise optimization method in McCarter and Kim (2016) to remove the memory requirement for storing large dense matrices during coordinate descent optimization. The coordinate descent updates require precomputing and storing the large dense matrices, such as $\mathbf{S}_{\mathbf{x}_s}$, $\mathbf{S}_{\mathbf{x}_d}$, and $\mathbf{V}^{-1}$. Instead, we perform a block-wise update of $\boldsymbol{\Xi}$, $\boldsymbol{\Pi}$, $\delta_{\boldsymbol{\Omega}}$, and $\delta_{\boldsymbol{\Delta}}$, precomputing and reusing the portions of the large matrices that are required for updating the current block of the parameters.

**Blockwise optimization for $\boldsymbol{\Omega}$:** A coordinate-descent optimization for $(\delta_{\boldsymbol{\Omega}})_{ij}$ requires computing the $i$th and $j$th columns of large dense $q \times q$ matrices $\mathbf{V}^{-1}$ and $\mathbf{V}^{-1}\left(\frac{1}{m^2}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\boldsymbol{\Pi} + \frac{1}{m}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{xz}}\boldsymbol{\Xi} + \frac{1}{m}\boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\boldsymbol{\Pi} + \boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{z}}\boldsymbol{\Xi}\right)\mathbf{V}^{-1}$ found in the gradient and Hessian of the objective with respect to $\boldsymbol{\Omega}$. We represent $\mathbf{V}^{-1}\left(\frac{1}{m^2}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{x}_s}\boldsymbol{\Pi} + \frac{1}{m}\boldsymbol{\Pi}^T\mathbf{S}_{\mathbf{xz}}\boldsymbol{\Xi} + \frac{1}{m}\boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{xz}}^T\boldsymbol{\Pi} + \boldsymbol{\Xi}^T\mathbf{S}_{\mathbf{z}}\boldsymbol{\Xi}\right)\mathbf{V}^{-1}$ as $(R_1 + R_2 + R_3)^T(R_1 + R_2 + R_3)$, where

$R_1 = Q_1\mathbf{V}^{-1}$, $R_2 = Q_2\mathbf{V}^{-1}$, and $R_3 = Q_3\mathbf{V}^{-1}$. We precompute and store $n \times q$ matrices, $Q_1 = \frac{1}{m}\mathbf{X}_s\mathbf{\Pi}$, $Q_2 = \mathbf{X}_s\mathbf{\Xi}$, and $Q_3 = \mathbf{Z}\mathbf{\Xi}$, which are used to update $R_1$, $R_2$, and $R_3$ after each update of $\mathbf{V}$. Instead of storing $\mathbf{V}^{-1}$ and $(R_1 + R_2 + R_3)^T(R_1 + R_2 + R_3)$ persistently, we perform blockwise coordinate descent, whereby the active set $\mathcal{S}_{\mathbf{\Omega}}$ of $\delta_{\mathbf{\Omega}}$ is clustered into smaller blocks, and we perform coordinate-descent updates for elements in the active set in each block. We cluster the rows and columns of $\delta_{\mathbf{\Omega}}$ by partitioning $\{1, \ldots, q\}$ into $k_{\mathbf{\Omega}}$ sets, $C_1, \ldots, C_{k_{\mathbf{\Omega}}}$. For each $(C_r, C_z)$ block of $\delta_{\mathbf{\Omega}}$, we do blockwise computation for $(\mathbf{V}^{-1})_{(:,C_r)}$ and $(\mathbf{V}^{-1})_{(:,C_z)}$, and compute the $i$th column in each block as $\mathbf{V}(\mathbf{V}^{-1})_{(:,i)} = \mathbf{e}_i$ using conjugate gradient method, where $\mathbf{e}_i$ is a vector of $q$ 0's except for 1 in the $i$th element. Similarly for $[(R_1 + R_2 + R_3)^T(R_1 + R_2 + R_3)]_{(:,C_r)}$, and $[(R_1 + R_2)^T(R_1 + R_2)]_{(:,C_z)}$, we obtain the $i$th column in each block using $(R_1 + R_2 + R_3)^T(R_1 + R_2 + R_3)_{(:,i)}$. After updating each $(\delta_{\mathbf{\Omega}})_{ij}$, we update the corresponding $i$th and $j$th rows of the intermediate results $(U_{\mathbf{\Omega}})_{(:,C_z)}$ and $(U_{\mathbf{\Omega}})_{(:,C_r)}$.

**Blockwise optimization for $\mathbf{\Delta}$:** A coordinate descent optimization for $(\delta_{\mathbf{\Delta}})_{ij}$ requires computing the $i$th and $j$th columns of large dense $q \times q$ matrices $\mathbf{V}^{-1}$, $(R_1 + R_2 + R_3)^T(R_1 + R_2 + R_3)$, $\mathbf{\Delta}^{-1}$, and $\mathbf{\Delta}^{-1}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_d}\mathbf{\Pi}\mathbf{\Delta}^{-1}$ found in the gradient and Hessian of the objective with respect to $\mathbf{\Delta}$. The optimization method for $\mathbf{\Delta}$ is similar to the one for $\mathbf{\Omega}$. We represent $\frac{1}{m^2}\mathbf{\Delta}^{-1}\mathbf{\Pi}^T\mathbf{S}_{\mathbf{x}_d}\mathbf{\Pi}\mathbf{\Delta}^{-1}$ as $R_4^T R_4$, where $R_4 = Q_4\mathbf{\Delta}^{-1}$. We precompute and store a $n \times q$ matrix, $Q_4 = \frac{1}{m}\mathbf{X}_d\mathbf{\Pi}$, which is used to update $R_4$ after each update of $\mathbf{\Delta}$.

**Blockwise optimization for $\mathbf{\Xi}$:** The coordinate descent update for $(\mathbf{\Xi})_{ij}$ requires the $i$th column of $\mathbf{S}_{\mathbf{x}_s}$ and $\mathbf{S}_{\mathbf{z}}$ and the $j$th column of $\mathbf{V}^{-1}$. Instead of storing $\mathbf{S}_{\mathbf{x}_s}$, $\mathbf{S}_{\mathbf{z}}$, and $\mathbf{V}$ persistently, we perform blockwise coordinate descent. We cluster the columns of $\mathbf{\Xi}$ by partitioning $\{1, \ldots, q\}$ into $k_{\mathbf{\Xi}}$ sets, $C_1, \ldots, C_{k_{\mathbf{\Xi}}}$. For each block $(i, C_r)$, where $i \in \{1, \ldots, p + r\}$, we compute $(\mathbf{S}_{\mathbf{x}_s})_{(:,i)}$, $(\mathbf{S}_{\mathbf{z}})_{(:,i)}$, and $(\mathbf{V}^{-1})_{(:,C_r)}$. After updating each $(\mathbf{\Xi})_{ij}$, we update the corresponding $i$th row of the intermediate result $(U_{\mathbf{\Xi}})_{(:,C_r)}$.

**Blockwise optimization for $\mathbf{\Pi}$:** The coordinate descent update for $(\mathbf{\Pi})_{ij}$ requires the $i$th column of $\mathbf{S}_{\mathbf{x}_s}$ and $\mathbf{S}_{\mathbf{x}_d}$ and the $j$th column of $\mathbf{V}^{-1}$ and $\mathbf{\Delta}^{-1}$. Instead of storing $\mathbf{S}_{\mathbf{x}_s}$, $\mathbf{S}_{\mathbf{x}_d}$, $\mathbf{V}$, and $\mathbf{\Delta}$ persistently, we perform blockwise coordinate descent. We cluster the columns of $\mathbf{\Pi}$ by partitioning $\{1, \ldots, q\}$ into $k_{\mathbf{\Pi}}$ sets, $C_1, \ldots, C_{k_{\mathbf{\Pi}}}$. For each block $(i, C_r)$, where $i \in \{1, \ldots, p\}$, we compute $(\mathbf{S}_{\mathbf{x}_s})_{(:,i)}$, $(\mathbf{S}_{\mathbf{x}_d})_{(:,i)}$, $(\mathbf{V}^{-1})_{(:,C_r)}$, and $(\mathbf{\Delta}^{-1})_{(:,C_r)}$. After updating each $(\mathbf{\Pi})_{ij}$, we update the corresponding $i$th row of the intermediate results $(U_{\mathbf{\Pi}_1})_{(:,C_r)}$ and $(U_{\mathbf{\Pi}_2})_{(:,C_r)}$.

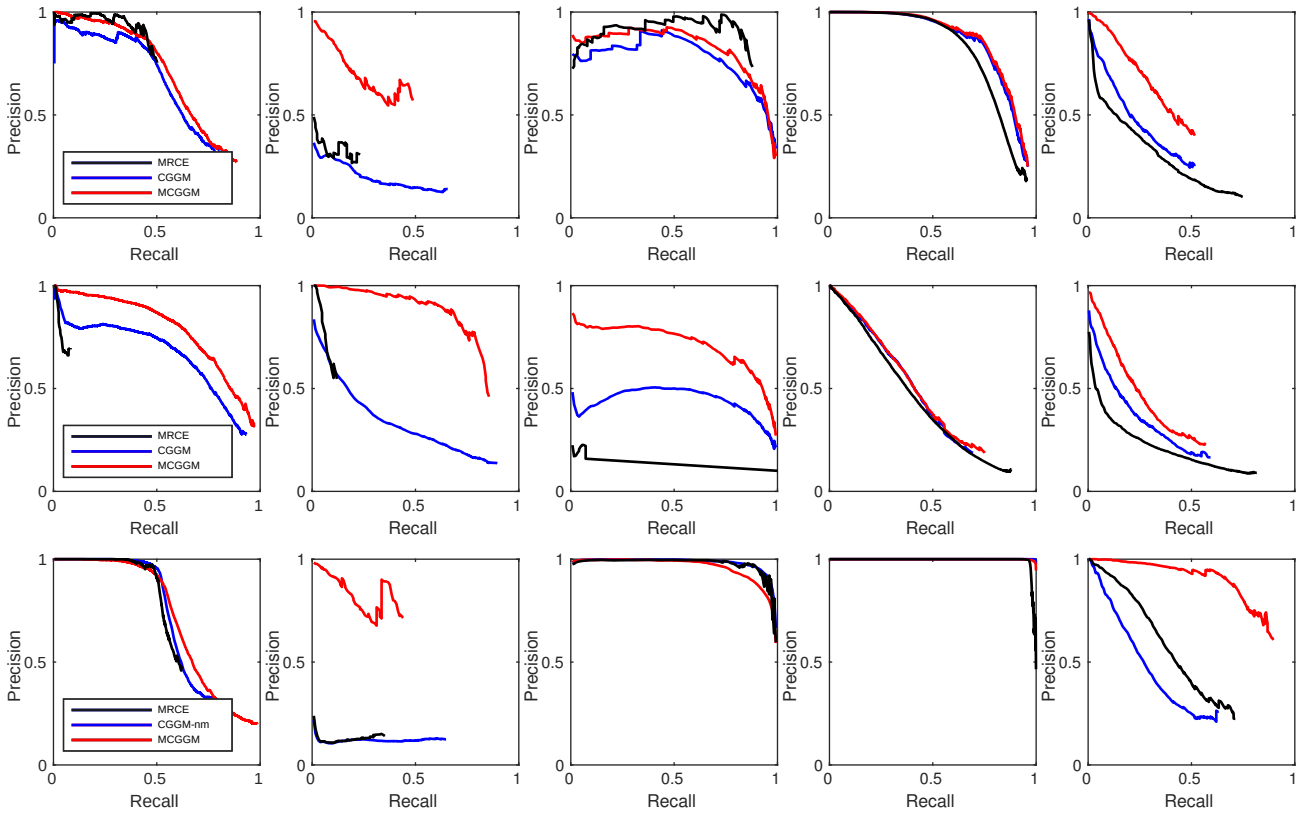


Figure 4: Comparison of methods on simulated data. Precision-recall curves for the recovery of (a) $\mathbf{\Omega} + \mathbf{\Delta}$, (b) $\mathbf{\Omega}$, and (c) $\mathbf{\Delta}$, (d) $\mathbf{\Pi}$, and (e) $\mathbf{\Xi}$. Top row for Case 1, middle row for Case 2, and bottom row for Case 3 with group size $m = 6$. $\mathbf{\Omega}$ is set as a clustered network.

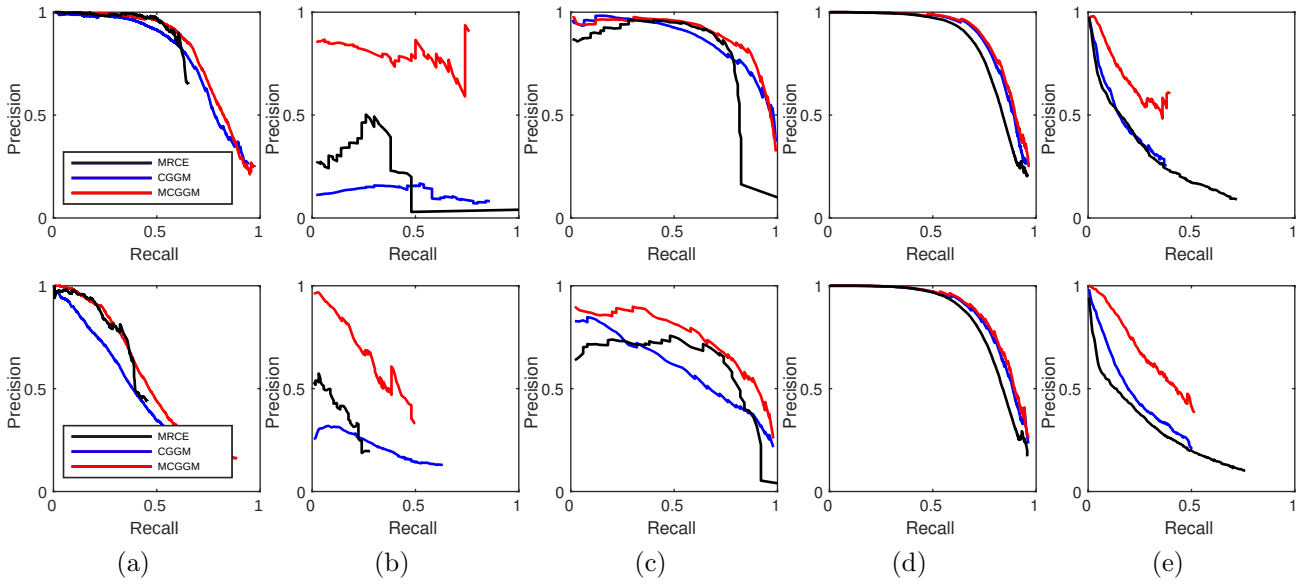


Figure 5: Comparison of methods on simulated data. Precision-recall curves for the recovery of (a) $\mathbf{\Omega} + \mathbf{\Delta}$, (b) $\mathbf{\Omega}$, and (c) $\mathbf{\Delta}$, (d) $\mathbf{\Pi}$, and (e) $\mathbf{\Xi}$. The results from Case 4 (top) and Case 5 (bottom) are shown for group size $m = 6$ and for scale-free networks for $\mathbf{\Omega}$ and $\mathbf{\Delta}$.

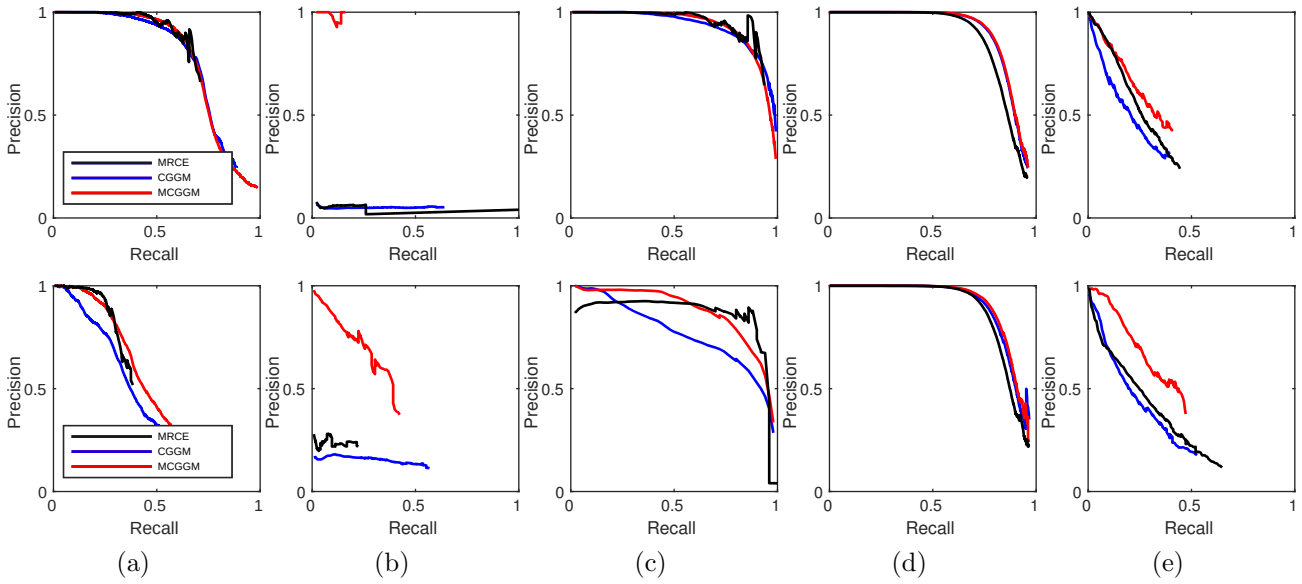


Figure 6: Comparison of methods on simulated data. Precision-recall curves for the recovery of (a) $\mathbf{\Omega} + \mathbf{\Delta}$, (b) $\mathbf{\Omega}$, and (c) $\mathbf{\Delta}$, (d) $\mathbf{\Pi}$, and (e) $\mathbf{\Xi}$. Results from Case 4 (top) and Case 5 (bottom) are shown for group size $m = 6$ and for $\mathbf{\Omega}$ with a clustered network.

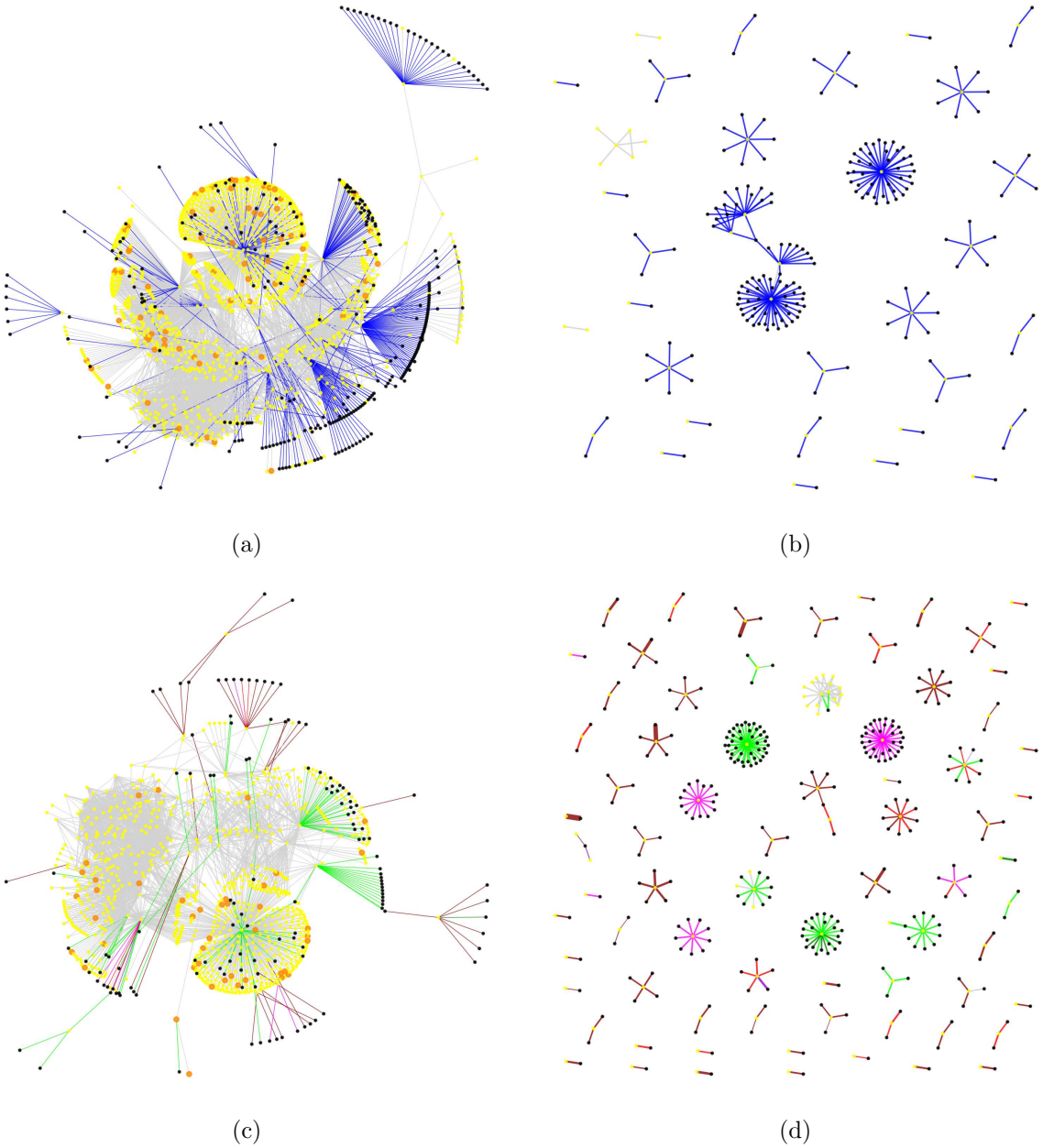

Figure 7: Single-level and multi-level conditional Gaussian graphical models estimated from GTEx data. The estimated single-level model is shown for (a) the largest connected component and (b) the rest of the gene networks, where gene network edges are shown as gray and the effects of variants on gene expressions are shown as blue. The estimated multi-level model is shown for (c) the largest connected component and (d) the rest of the gene networks. In the multi-level model, the influence of *cis*-acting variants is shown as brown if they have been previously annotated as having impact on gene expression, red if they have not been annotated, and pink if the distance to the gene it is influencing is greater than 1 megabase. The influence of *trans*-acting edges is shown as green. Known transcription factors are colored as big orange dots.