

Supplementary Materials for “A Latent Multilayer Graphical Model For Complex, Interdependent Systems”

We provide supplementary material for the paper *A Latent Multilayer Graphical Model For Complex, Interdependent Systems*. Specifically, proofs of the main paper theorems are provided in Section 9. Section 10 provides information on the computational reproducibility of our work. More detailed information on the simulation studies contained in the main article is available in Section 11 and additional simulations are considered in Section 12. More detailed information on the experimental data used in the main article is available in Section 13.

9 Theoretical analysis

9.1 Derivation of likelihood function

We assume that the inverse covariance matrix for each layer is given by $\Sigma_\alpha^{-1} = S_\alpha + L_\alpha = S_\alpha + \mathcal{P}_{\Omega_\alpha}(L)$ where $\alpha \in 1, \dots, l$. In layer α , the i th sample is denoted $\mathbf{x}_{\alpha i}$. Given the product rule for the axiom of independent events, we define the probability density function of a l layer, multivariate Gaussian distribution as

$$f(\{\mathbf{x}_{\alpha i}\}_{\alpha=1}^l) = \underbrace{\prod_{\alpha=1}^l (2\pi)^{-p_\alpha/2} \det(\Sigma_\alpha^{-1})^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x}_{\alpha i} - \bar{\mathbf{x}}_\alpha)^T (\Sigma_\alpha^{-1})(\mathbf{x}_{\alpha i} - \bar{\mathbf{x}}_\alpha)\right]}_{\text{PDF of multivariate Gaussian}}.$$

Applying the log function, we have

$$\log f(\{\mathbf{x}_{\alpha i}\}_{\alpha=1}^l) = \sum_{\alpha=1}^l \left(-\frac{p_\alpha}{2} \log(2\pi) - \frac{1}{2} \log \det(\Sigma_\alpha^{-1}) - \frac{1}{2} (\mathbf{x}_{\alpha i} - \bar{\mathbf{x}}_\alpha)^T (\Sigma_\alpha^{-1})(\mathbf{x}_{\alpha i} - \bar{\mathbf{x}}_\alpha) \right).$$

For n_α independent observations $\{\mathbf{x}_{\alpha 1}, \dots, \mathbf{x}_{\alpha n_\alpha}\}$ in each layer α , sample covariance $\tilde{\Sigma}_\alpha$, and parameters $S_\alpha + L_\alpha = \Sigma_\alpha^{-1}$, the log-likelihood function is given by

$$\mathcal{L}_M(\{S\}_{\alpha=1}^l, L) = \sum_{\alpha=1}^l \sum_{i=1}^{n_\alpha} f(\mathbf{x}_{\alpha i}) = \sum_{\alpha=1}^l \mathcal{L}(S_\alpha, L_\alpha; \tilde{\Sigma}_\alpha).$$

Assuming $\mu_\alpha = 0$, $\forall \alpha \in 1, \dots, l$, and dropping constant terms, we can simplify the layerwise likelihood functions in the typical way of Hastie et al. (2009). Finally, we have

$$\mathcal{L}_M(\{S\}_{\alpha=1}^l, L) \propto \sum_{\alpha=1}^l \log \det(S_\alpha + L_\alpha) - \text{tr}(\tilde{\Sigma}_\alpha(S_\alpha + L_\alpha)).$$

9.2 Proofs

In this section, we present detailed proofs of the proposed theorems. The sketch of our proof strategy is as follows. We are first interested in determining under what conditions we can exactly recover each S_α^* and L_α^* for all $\alpha = \{1, \dots, l\}$ (Theorem 5.2). Furthermore, we provide a proof that shows that we can recover model selection consistent estimates of S_α^* with $L1$ regularization (Theorem 5.3). After showing that we can exactly recover each L_α^* , we will show that this is sufficient to exactly recover L^* (Theorem 5.1). Lastly, we derive a recovery rate for L^* (Theorem 5.4).

To begin, we consider the consistency of \hat{S}_α and \hat{L}_α for one layer. Here, we adapt conditions from Wang and Allen (2023) which guarantee appropriate regularization (condition 4), bounded eigenvalues (conditions 2,3, and 5), and sub-Gaussianity (condition 1).

Condition 1: $\{\mathbf{x}_{\alpha i}\}_{i=1}^{n_\alpha}$ are i.i.d. $\mathcal{N}(0, \Sigma_\alpha^*)$.

Condition 2: There exists a constant $k_1 > 0$ such that $\lambda_{\min}(S_\alpha^{*-1}) \geq k_1$ (equivalently, $\lambda_{\max}(S_\alpha^*) \leq 1/k_1$).

Condition 3: There exists a constant $k_2 > 0$ such that $\lambda_{\max}(\mathbf{S}_\alpha^{*-1}) \leq k_2$.

Condition 4: The regularization parameter satisfies $\rho_\alpha \asymp \sqrt{\frac{\log p_\alpha}{n_\alpha}}$.

Condition 5: There exists a constant $k_3 > 0$ such that $\lambda_{\max}(\mathbf{L}_\alpha^*) \leq k_3$.

Lemma 9.1 (Joint $\hat{\mathbf{S}}$ and $\hat{\mathbf{L}}$ Consistency). *Assume conditions 1-5 hold. For the L1 based SLICE estimator for $\hat{\mathbf{S}}_\alpha$, w.h.p, we have*

$$\max\{\|\hat{\mathbf{S}}_{\alpha\text{off}} - \mathbf{S}_{\alpha\text{off}}^*\|_F, \|\hat{\mathbf{L}}_\alpha - \mathbf{L}_\alpha^*\|_\infty\} \lesssim C_4 \sqrt{\frac{s_\alpha \log p_\alpha}{n_\alpha}} + C_5 \sqrt{\frac{\log p_\alpha}{n_\alpha}}.$$

Proof. Let $\hat{\mathbf{L}}_\alpha = \tilde{\Sigma}_\alpha^{-1} - \mathbf{S}_\alpha$, $\mathbf{L}_\alpha^* = \Sigma_\alpha^{*-1} - \mathbf{S}_\alpha$, and $\Delta_{L_\alpha} = \mathbf{L}_\alpha^* - \hat{\mathbf{L}}_\alpha$. Notice that $\hat{\mathbf{L}}_\alpha - \mathbf{L}_\alpha^* = \tilde{\Sigma}_\alpha^{-1} - \Sigma_\alpha^{*-1}$. The inverse is approximated by the Neumann series, hence

$$(\Sigma_\alpha^* + \Delta_{\Sigma_\alpha})^{-1} = \sum_{k=0}^{\infty} (\Sigma_\alpha^{*-1} \Delta_{\Sigma_\alpha})^k \Sigma_\alpha^{*-1} = \Sigma_\alpha^{*-1} + \sum_{k=1}^{\infty} (\Sigma_\alpha^{*-1} \Delta_{\Sigma_\alpha})^k \Sigma_\alpha^{*-1}.$$

Then, after rearranging and applying the infinity norm, we have

$$\|(\Sigma_\alpha^* + \Delta_{\Sigma_\alpha})^{-1} - \Sigma_\alpha^{*-1}\|_\infty = \|\tilde{\Sigma}_\alpha^{-1} - \Sigma_\alpha^{*-1}\|_\infty = \left\| \sum_{k=1}^{\infty} (\Sigma_\alpha^{*-1} \Delta_{\Sigma_\alpha})^k \Sigma_\alpha^{*-1} \right\|_\infty.$$

For the first order approximation $k = 1$, we have

$$\|\tilde{\Sigma}_\alpha^{-1} - \Sigma_\alpha^{*-1}\|_\infty \approx \|\Sigma_\alpha^{*-1}\|_\infty^2 \cdot \|\Delta_{\Sigma_\alpha}\|_\infty \approx \mathcal{O}(\|\tilde{\Sigma}_\alpha - \Sigma_\alpha^*\|_\infty).$$

We have

$$\|\hat{\mathbf{L}}_\alpha - \mathbf{L}_\alpha^*\| \approx \|\Sigma_\alpha^{*-1}\|^2 \cdot \|\tilde{\Sigma}_\alpha - \Sigma_\alpha^*\|.$$

Then, combining with Lemma 9.3, we have

$$\|\hat{\mathbf{L}}_\alpha - \mathbf{L}_\alpha^*\|_\infty \lesssim \|\Sigma_\alpha^{*-1}\|_\infty^2 \cdot 2 \max_i (\Sigma_{\alpha ii}^*) \sqrt{\frac{\log p_\alpha}{n_\alpha}} = C_1 \sqrt{\frac{\log p_\alpha}{n_\alpha}} \quad (3)$$

holds with probability at least $1 - \frac{C_1}{p_\alpha}$, where $C_1 > 0$ is a constant. Next, we determine $\|(\mathbf{S}_\alpha^* + \Delta_{L_\alpha})^{-1} - \mathbf{S}_\alpha^{*-1}\|_\infty$. Using a similar approach as before, we have

$$\begin{aligned} (\mathbf{S}_\alpha^* + \Delta_{L_\alpha})^{-1} &\approx \mathbf{S}_\alpha^{*-1} - \mathbf{S}_\alpha^{*-1} \Delta_{L_\alpha} \mathbf{S}_\alpha^{*-1} \\ \|(\mathbf{S}_\alpha^* + \Delta_{L_\alpha})^{-1} - \mathbf{S}_\alpha^{*-1}\|_\infty &\approx \|\mathbf{S}_\alpha^{*-1}\|_\infty^2 \cdot \|(\Delta_{L_\alpha})\|_\infty. \end{aligned} \quad (4)$$

Notice again, that the rate depends on Δ_{L_α} . Thus, applying 3, we have

$$\|(\mathbf{S}_\alpha^* + \Delta_{L_\alpha})^{-1} - \mathbf{S}_\alpha^{*-1}\|_\infty = \mathcal{O}\left(\sqrt{\frac{\log p_\alpha}{n_\alpha}}\right).$$

This allows us to apply Lemma 9.5, and then we have

$$\|\hat{\mathbf{S}}_{\alpha\text{off}} - \mathbf{S}_{\alpha\text{off}}^*\|_F \leq C_4 \sqrt{\frac{s_\alpha \log p_\alpha}{n_\alpha}}, \quad (5)$$

with probability at least $1 - b_1 e^{-b_2 n_\alpha \rho_\alpha^2}$ where b_1 and b_2 depend on k . To combine these results, consider the maximum deviation between the two terms $\max\{\|\hat{\mathbf{S}}_{\alpha\text{off}} - \mathbf{S}_{\alpha\text{off}}^*\|_F, \|\hat{\mathbf{L}}_\alpha - \mathbf{L}_\alpha^*\|_\infty\}$. Given the bounds of (3) and (5), the event that both bounds hold can be determined by considering the worst-case scenario. Specifically, we need to account for the higher of the two failure probabilities, $1 - \max\left\{b_1 e^{-b_2 n_\alpha \rho_\alpha^2}, \frac{C_1}{p_\alpha}\right\}$, which reflects a conservative estimate that at least one of the bounds fails. \square

Next, we turn our attention to the exact recovery of \mathbf{L}^* . We utilize the work of (Liu et al., 2017) as motivation for the next condition. Here, we require that each sub-matrix of \mathbf{L}^* formed by Ω_α , \mathbf{L}_α^* , has the same rank as \mathbf{L}^* .

Condition 6: $\mathcal{R}(\mathbf{L}^*) = \mathcal{R}(\mathbf{L}_\alpha^*)$ for all α .

This condition guarantees that we are able to reconstruct the missing parts of \mathbf{L}^* from only the observed portions. With Condition 6, we are ready to prove Theorem 5.1.

Proof of Theorem 5.1. We must first show that \mathbf{L}^* is also Ω/Ω^T -isomeric according to Definition 1. Since $\mathcal{R}(\mathbf{L}_\alpha^*) = \mathcal{R}(\mathbf{L}^*)$ for all α , we have

$$\mathcal{R}([\mathbf{L}^*]_{\Omega^{i,:}}) = \mathcal{R}(\mathbf{L}^*).$$

Given that $|\cup_{\alpha=1}^m \omega_\alpha| \geq |\omega|$, it follows that $\Omega^i \neq \emptyset$ for all $i \in \omega_\alpha$. Therefore, \mathbf{L}^* is Ω -isomeric. Due to symmetry, $\mathbf{L}^* = \mathbf{L}^{*T}$ and $\Omega = \Omega^T$. Therefore, applying Lemma 9.2 gives the desired result. \square

After this, we are ready to prove Theorem 5.2 by showing the consistency of the estimates of \mathbf{S}_α^* and \mathbf{L}^* . Here, we combine the insights from Lemma 9.1 and Theorem 5.1, as well as conditions 1-6. We leverage the fact that the layerwise recovery from Lemma 9.1 implies element-wise convergence as all layers' $n_\alpha \rightarrow \infty$.

Proof of Theorem 5.2. First, we consider the convergence of independent SLICE models applied layerwise.

For each α , the sparse and latent components are given by \mathbf{S}_α^* and $\mathcal{P}_{\Omega_\alpha}(\mathbf{L}^*) = \mathbf{L}_\alpha^*$. We can apply Lemma 9.1 and we have that w.h.p., the following statement holds

$$\max\{\|\hat{\mathbf{S}}_{\alpha\text{off}} - \mathbf{S}_{\alpha\text{off}}^*\|_F, \|\hat{\mathbf{L}}_\alpha - \mathbf{L}_\alpha^*\|_\infty\} \lesssim C_4 \sqrt{\frac{s_\alpha \log p_\alpha}{n_\alpha}} + C_5 \sqrt{\frac{\log p_\alpha}{n_\alpha}}. \quad (6)$$

As $n_\alpha \rightarrow \infty$, the terms on the right-hand side of the inequality tend to zero, which implies that $\|\hat{\mathbf{S}}_{\alpha\text{off}} - \mathbf{S}_{\alpha\text{off}}^*\|_F \xrightarrow{P} 0$ and $\|\hat{\mathbf{L}}_\alpha - \mathbf{L}_\alpha^*\|_\infty \xrightarrow{P} 0$.

Recall that $\|\hat{\mathbf{S}}_{\alpha,\text{off}} - \mathbf{S}_{\alpha,\text{off}}^*\|_F = \sqrt{\sum_{i \neq j} |(\hat{\mathbf{S}}_{\alpha ij} - \mathbf{S}_{\alpha ij}^*)|^2}$ and $\|\hat{\mathbf{L}}_\alpha - \mathbf{L}_\alpha^*\|_\infty = \max_i \sum_j |(\hat{\mathbf{L}}_{\alpha ij} - \mathbf{L}_{\alpha ij}^*)|$. With $\|\cdot\|_F$ and $\|\cdot\|_\infty$ tending to zero, this implies element-wise convergence, and so we have

$$\hat{\mathbf{S}}_{\alpha\text{off}} \xrightarrow{P} \mathbf{S}_{\alpha\text{off}}^* \text{ and } \hat{\mathbf{L}}_\alpha \xrightarrow{P} \mathbf{L}_\alpha^* \text{ as } n_\alpha \rightarrow \infty, \forall \alpha = 1 \dots l. \quad (7)$$

Given that $\Omega = \cup_{\alpha=1}^l \Omega_\alpha$, we construct $\mathcal{P}_\Omega(\hat{\mathbf{L}})$ by the following

$$\mathcal{P}_\Omega(\hat{\mathbf{L}})_{ij} = \begin{cases} \hat{\mathbf{L}}_{\alpha ij} & \text{if } (i, j) \in \Omega_\alpha, \forall \alpha \in \{1, \dots, l\} \\ \emptyset & \text{else} \end{cases}$$

which, combined with (7), gives us $\mathcal{P}_\Omega(\hat{\mathbf{L}}) \xrightarrow{P} \mathcal{P}_\Omega(\mathbf{L}^*)$ as $n_\alpha \rightarrow \infty, \forall \alpha = 1 \dots l$. From Lemma 5.1, we have that \mathcal{P}_Ω^{-1} exists, and so applying this yields $\mathcal{P}_\Omega^{-1}(\mathcal{P}_\Omega(\mathbf{L}^*)) = \mathbf{L}^*$. \square

To guarantee a sign-consistent recovery of \mathbf{S}^* in Theorem 5.3, we require assumptions on sub-Gaussianity, minimum signal strength, and irrepresentability of the sparsity pattern.

Condition 7: $\mathbf{x}_i / \sqrt{\Sigma_\alpha}$ are i.i.d. sub-Gaussian with parameter σ .

Condition 8: The irrerepresentable condition holds; that is, there exists $\eta \in (0, 1)$ such that $\max_{e \in \mathbf{S}^*} \|\mathbf{\Gamma}_e^* (\mathbf{\Gamma}^*)^{-1}\|_1 \leq 1 - \eta$ where $\mathbf{\Gamma}^* = \mathbf{S}^{*-1} \otimes \mathbf{S}^{*-1}$.

Condition 9: \mathbf{S}^* satisfies $\min_{(i,j) \in \mathbf{S}_\alpha^*} |S_{\alpha ij}^*| > C \sqrt{\frac{\log p_\alpha}{n_\alpha}}$, for some constant $C > 0$.

Condition 7 ensures that the tails of the distribution are bounded, to encourage the estimate to be well-behaved. Condition 8, the irrerepresentable or incoherence condition, is typical in L1 penalization literature, and requires that there is not too much correlation within the sparsity structure of \mathbf{S}^* (Wainwright et al., 2006; Zhao and Yu, 2006). Lastly, condition 9 requires that the signal of \mathbf{S}^* be strong enough to detect. With conditions 1-5, and 7-9, we can prove Theorem 5.3.

Proof of Theorem 5.3. From (4), we have $\|(\mathbf{S}_\alpha^* + \Delta_{L_\alpha})^{-1} - \mathbf{S}_\alpha^{*-1}\|_\infty = \mathcal{O}\left(\sqrt{\frac{\log p_\alpha}{n_\alpha}}\right)$. Given $\Sigma_\alpha^* \hat{L}_\alpha = (\mathbf{S}_\alpha^* + \Delta_{L_\alpha})^{-1} = (\Sigma_\alpha^{*-1} - \hat{L}_\alpha)^{-1}$, we can apply Lemma 9.4. Assuming that the remaining conditions of Lemma 9.4 hold, we get the desired result. \square

Lastly, we consider deriving a rate for $\hat{L} - L^*$ by leveraging the individual rates for each layer α we derived in Lemma 9.1. In particular, we assume that the rate of the observed portions in Ω is the same as the unobserved portions in Ω^c .

Condition 10: $\|[\hat{L} - L^*]_{\setminus \alpha, \alpha}\|_\infty = \mathcal{O}(\|\hat{L}_\alpha - L_\alpha^*\|_\infty)$

This condition is motivated by Liu et al. (2017) and Candes and Recht (2012). A prominent idea in the matrix completion literature is the notion of coherence, which explains to what extent the singular vector directions of a low-rank matrix are “spread out”. A great deal of the matrix completion literature utilizes a setting in which the coherence is low, which implies that the matrix is not too concentrated in any particular set of entries. Our condition can be viewed as an extension of this where $\hat{L} - L^*$ is relatively evenly “spread out”, such that the difference between \hat{L} and L^* is not too concentrated in any row since the bound we derive in Theorem 9.1 is in the infinity norm. With Conditions 1-5 and 10, we are able to prove Theorem 5.4.

Proof of Theorem 5.4. It is implied that there exists a constant C_6 such that $\|[\hat{L} - L^*]_{\setminus \alpha, \alpha}\|_\infty = C_6 \|\hat{L}_\alpha - L_\alpha^*\|_\infty \quad \forall \alpha = 1, \dots, l$.

Next, using the subadditivity of norms and combining with (6), we have that

$$\begin{aligned} \|\hat{L} - L^*\|_\infty &\lesssim \sum_{\alpha=1}^l C_6 \|\hat{L}_\alpha - L_\alpha^*\|_\infty, \\ \|\hat{L} - L^*\|_\infty &\lesssim \sum_{\alpha=1}^l C_6 C_5 \sqrt{\frac{\log p_\alpha}{n_\alpha}}, \\ \|\hat{L} - L^*\|_\infty &\lesssim \sum_{\alpha=1}^l C \sqrt{\frac{\log p_\alpha}{n_\alpha}}. \end{aligned}$$

\square

9.3 Auxiliary Lemmas

Definition 1 (Ω/Ω^T -isomerism; (Liu et al., 2017)). Let $\mathbf{M} \in \mathbb{R}^{p \times q}$ and $\Omega \subseteq \{1, 2, \dots, p\} \times \{1, 2, \dots, q\}$. Suppose that $\Omega^j \neq \emptyset$ (empty set), $\forall 1 \leq j \leq n$. Then the matrix \mathbf{M} is called Ω -isomeric iff

$$\mathcal{R}([\mathbf{M}]_{\Omega_j, :}) = \mathcal{R}(\mathbf{M}), \forall j = 1, 2, \dots, q.$$

Furthermore, if $\Omega_i \neq \emptyset$ and $\Omega^j \neq \emptyset, \forall i = 1, \dots, p, j = 1, \dots, q$. Then the matrix \mathbf{M} is Ω/Ω^T -isomeric iff \mathbf{M} is Ω -isomeric and \mathbf{M}^T is Ω^T -isomeric.

Lemma 9.2 (Invertability of \mathcal{P}_Ω ; (Liu et al., 2017)). Let $\mathbf{M} \in \mathbb{R}^{p \times q}$ and $\Omega \subseteq \{1, 2, \dots, p\} \times \{1, 2, \dots, q\}$. Let the SVD of \mathbf{M} be $U_0 \Sigma_0 V_0^T$. Denote $\mathcal{P}_{U_0}(\cdot) = U_0 U_0^T(\cdot)$ and $\mathcal{P}_{V_0}(\cdot) = (\cdot) V_0 V_0^T$. Then we have the following:

1. The linear operator $\mathcal{P}_{U_0} \mathcal{P}_\Omega \mathcal{P}_{U_0}$ is invertible iff U_0 is Ω -isomeric.
2. The linear operator $\mathcal{P}_{V_0} \mathcal{P}_\Omega \mathcal{P}_{V_0}$ is invertible iff V_0 is Ω^T -isomeric.

Lemma 9.3 (Xu et al. (2017); Rothman et al. (2008)). Let $\mathbf{x}_i \in \mathbb{R}^p$ be i.i.d. sub-Gaussian random vectors, and $\tilde{\Sigma} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^T$. Then we have

$$\left\| \tilde{\Sigma} - \Sigma^* \right\|_\infty \leq 2 \max_i (\Sigma_{ii}^*) \sqrt{\frac{\log p}{n}}$$

holds with probability at least $1 - \frac{C_1}{p}$, where $C_1 > 0$ is a constant.

Lemma 9.4 (Wang and Allen (2023); Ravikumar et al. (2011)). *Let $\mathbf{x}_i/\sqrt{\Sigma^*}$ be sub-Gaussian with parameter σ . Assume the irrepresentable condition, $\mathbf{\Gamma}^* := \nabla_{\hat{\mathbf{S}}}^2 g(\hat{\mathbf{S}})|_{\hat{\mathbf{S}}=\mathbf{S}^*} = (\mathbf{S}^*)^{-1} \otimes (\mathbf{S}^*)^{-1} = (\mathbf{S}^* \otimes \mathbf{S}^*)^{-1}$. There exists some $\alpha \in (0, 1)$ such that $\max_{e \in \mathbf{S}^*} \|\mathbf{\Gamma}_e^* (\mathbf{\Gamma}_{\mathbf{S}^*}^*)^{-1}\|_1 \leq (1 - \alpha)$. Assume a minimum signal strength*

$$\mathbf{S}_{\min}^* := \min_{(i,j) \in \mathbf{S}^*} \{|\mathbf{S}_{ij}^*|\} > 16\sqrt{2(1+4\alpha^2)} \max_i \Sigma_{ii}^* (1 + 12\alpha^{-1}\kappa^{-1})^{1/2} \sqrt{\frac{\log p + \log 4}{n}}$$

and that the sample size n satisfies the bound $n > C_2 d^2 (1 + \frac{12}{\alpha} + C_3)^2 (\log p + \log 4)$. If $\|(\mathbf{S}^ + \mathbf{\Delta}_L)^{-1} - (\mathbf{S}^*)^{-1}\|_\infty = \mathcal{O}\left(\sqrt{\frac{\log p}{n}}\right)$, then with probability greater than $1 - 1/p^{\tau-2}$, the graphical Lasso estimator $\hat{\mathbf{S}}^\rho$ with regularization parameter $\rho = (\frac{12}{\alpha}) \delta_f(n, p^\tau)$ is model selection consistent as $p \rightarrow \infty$,*

$$\mathbb{P}\left(\text{sign}(\hat{\mathbf{S}}_{ij}^\rho) = \text{sign}(\mathbf{S}_{ij}^*), \forall i, j \in \hat{\mathbf{S}}^\rho\right) \geq 1 - \frac{1}{p^{\tau-2}} \rightarrow 1.$$

Here, $C_2 = \{48\sqrt{2}(1 + 4\delta^2) \max_i (\Sigma_{ii}^) \max\{\kappa_{\mathbf{S}^* - 1T_r^*}^3, \kappa_{\mathbf{S}^* - 1T_r^*}^2\}\}^2$, $C_3 = \|(\mathbf{S}^* + \mathbf{L}^*)^{-1} - (\mathbf{S}^*)^{-1}\|_\infty / (\frac{\alpha\rho}{12})$, and $\delta_f(n, p^\tau) = \sqrt{128(1 + 4\delta^2) \max_i (\Sigma_{ii}^*)^2 \sqrt{\frac{\log p + \log 4}{n}}}$.*

Lemma 9.5 (Wang and Allen (2023); Ravikumar et al. (2011)). *Let \mathbf{x}_i be an i.i.d. $\mathcal{N}(0, \Sigma^*|\hat{\mathbf{L}})$ where $\Sigma^*|\hat{\mathbf{L}} = (\mathbf{S}^* + \mathbf{\Delta}_L)^{-1}$. Let $\lambda_{\min}(\mathbf{S}^{*-1}) \geq k > 0$, or $\lambda_{\max}(\mathbf{S}^*) \leq 1/k$, and $\lambda_{\max}(\mathbf{S}^{*-1}) \leq k$. If $\|(\mathbf{S}^* + \mathbf{\Delta}_L)^{-1} - (\mathbf{S}^*)^{-1}\|_\infty = \mathcal{O}\left(\sqrt{\frac{\log p}{n}}\right)$ and $\rho \asymp \sqrt{\frac{\log p}{n}}$, there exists a C_4 such that the graphical Lasso estimate $\hat{\mathbf{S}}$ satisfies*

$$\|\hat{\mathbf{S}}_{\text{off}} - \mathbf{S}_{\text{off}}^*\|_F \leq C_4 \sqrt{\frac{s \log p}{n}},$$

with probability at least $1 - b_1 e^{-b_2 n \rho^2}$ where b_1 and b_2 depend on k .

10 Reproducibility

Code implementations are provided on Github. The main methodology in the *Core* directory contains the `multislice` function, which takes the following arguments:

- `Sigmas` a list of the l layer symmetric sample covariance matrices $\{\tilde{\Sigma}_\alpha\}_{\alpha=1}^l$.
- `rhos` either a single ρ value, or vector of ρ values of length l , to enforce sparsity in \hat{S}_α .
- `rank` the rank of the low-rank joint space, r , for \hat{L} .
- `Sest` the type of sparse estimator to use; either “glasso”, “clime”, “gscad”, or “huge_glasso”. By default, “glasso” is used.
- `tol` the tolerance for the stopping criteria. By default, 1e-3 is used.
- `maxiter` the maximum number of iterations for the algorithm(s). By default, 100 is used.

11 Main simulation studies

In this section, we further detail model implementation and hyperparameter selection. We also provide definitions of the evaluation criteria and details of the full simulation results.

11.1 Implementation details

Table 3 provides a summary of all methods with details on the implementation source and hyperparameter selection. In general, we use the default option for the selection of hyperparameters in the respective code.

For the CNJGL, BJEMGM, and CFR methods, we are unable to obtain estimates of \hat{L} as they only provide estimates for \hat{S}_α . For coglasso, we use the entries in Ω_α for \hat{S}_α , and then the remaining entries for \hat{L} . For LRGQ, we compute the Moore-Penrose pseudoinverse of the completed low-rank covariance for \hat{L} . We also compute the graphical lasso on the completed low-rank covariance, selecting the entries in Ω_α for \hat{S} . With BANS, we set \hat{S}_α as the undirected-edge coefficients, α , and \hat{L} as the directed edge coefficients, b . For MLGGM, we set \hat{S}_α as the estimated intralayer edges Θ and \hat{L} as the interlayer edges B . For JMMLE, we use the same definitions as for MLGGM.

Method	Implementation	Hyperparameter Selection
CNJGL	Github	λ_1 and λ_2 are chosen based on the sampling rate of n_1 and n_2 .
MLGGM	Github	“Lasso” initialization with screening and stability selection, $\alpha = 0.05$, and 20 bootstrapped samples.
BJEMGM	Github	BIC based selection over a grid of v_0 and v_1 .
CFR	Github	AIC based selection over a grid of λ_1 and λ_2 .
BANS	Github	Default values for hyper parameters are used, $\lambda = 3$, and $\delta = 3$.
JMMLE	Github	By default, $\lambda = 0.5\sqrt{\frac{\log p}{n}}$, and γ is chosen via BIC selection.
LRGQ	Github	Rank for matrix completion is set to the ground truth $\mathcal{R}(L^*)$.
coglasso	CRAN	λ_w , λ_b , and c are selected via Stability Approach to Regularization Selection (StARS).
multiSLICE	Github	Grid search and three-fold cross-validation (CV) is used to select ρ , whereas rank for is set to $\mathcal{R}(L^*)$. Model is selected using the highest average cross-validated likelihood.

Table 3: An overview of each method’s implementation and how the hyperparameters were selected.

11.2 Evaluation criteria

To compare the estimate \hat{S} , to the true value S^* , we use the F1 score. To begin, we first obtain the upper triangular elements of each to obtain the vectors $\mathcal{U}(\hat{S}) = \hat{s}_u$ and $\mathcal{U}(S^*) = s_u^*$ for the estimated and true matrices, respectively. Next, we compute for these vectors,

$$TPR = \frac{TP}{TP + FN}, \quad \text{Precision} = \frac{TP}{TP + FP}, \quad F_1 = 2 \times \frac{\text{Precision} \times TPR}{\text{Precision} + TPR}$$

TP is the count of $s_u^* = 1$ and $\hat{s}_u = 1$ TN is the count of $s_u^* = 0$ and $\hat{s}_u = 0$
 FP is the count of $s_u^* = 0$ and $\hat{s}_u = 1$ FN is the count of $s_u^* = 1$ and $\hat{s}_u = 0$

To compare \hat{L} with L^* , we compute the sin angle between the first singular vectors. This is given by

$$\sin \theta(\hat{u}_1, u_1^*) = \sqrt{1 - \langle \hat{u}_1, u_1^* \rangle^2}.$$

Here, \hat{u}_1 and u_1^* are the first singular vectors of \hat{L} and L^* , respectively.

11.3 Full simulation results

In this section, we provide the full numerical results for the simulations in the main article. We only give results from methods that provided non-trivial (non-zero or non-constant over increasing n_α) estimates of \hat{S}_α and \hat{L} . For simulation 1, this corresponds to multiSLICE, MLGGM, and JMMLE for \hat{S}_α , and multiSLICE for \hat{L} . For simulation 2, this corresponds to multiSLICE, coglasso, MLGGM, and JMMLE for \hat{S}_α , and multiSLICE for \hat{L} . Table 6 shows the mean F1 score for JMMLE, MLGGM, and multiSLICE methods for various values of l and n in simulation 1. Table 4 shows the mean $\sin \theta$ for multiSLICE for various values of l and n in simulation 1. For simulation 1, $\mathcal{R}(L^*)$ is set to 2, and each simulation is repeated 100 times. Similarly, Tables 7 and 5 show the mean F1 score and $\sin \theta$ for simulation 2 where the performance of the methods is studied for various values of $\mathcal{R}(L^*)$ and n . Here, l is set to 2 for all simulations, and each simulation is repeated 100 times. Unsurprisingly, the mean F1 scores are relatively low, given the difficulty of estimating non-zero elements of precision matrices. However, multiSLICE has the best performance compared to the other state-of-the-art methods.

n	$l = 2$	$l = 3$	$l = 4$	$l = 5$
100	0.9198 (0.1015)	0.9241 (0.0935)	0.9519 (0.0608)	0.9511 (0.0553)
200	0.3468 (0.3101)	0.3837 (0.3161)	0.3989 (0.3244)	0.4706 (0.3401)
300	0.2985 (0.3121)	0.3242 (0.3179)	0.3758 (0.3454)	0.4208 (0.3491)
400	0.2809 (0.3189)	0.3306 (0.3427)	0.3542 (0.3453)	0.4162 (0.3483)
500	0.2388 (0.2907)	0.3023 (0.3357)	0.3393 (0.3434)	0.3652 (0.3434)

Table 4: Mean $\sin \theta$ (standard deviation) in simulation 1 for multiSLICE across for different l and n . $\mathcal{R}(L^*)$ is set to 2 for all simulations, and each simulation is repeated 100 times.

n	$\mathcal{R}(L^*) = 2$	$\mathcal{R}(L^*) = 3$	$\mathcal{R}(L^*) = 4$	$\mathcal{R}(L^*) = 5$
100	0.9198 (0.1015)	0.9430 (0.0770)	0.9549 (0.0566)	0.9550 (0.0504)
200	0.3468 (0.3101)	0.5500 (0.3504)	0.6785 (0.2976)	0.7892 (0.2626)
300	0.2985 (0.3121)	0.4946 (0.3591)	0.6387 (0.3276)	0.7329 (0.3032)
400	0.2809 (0.3189)	0.5522 (0.3936)	0.6473 (0.3427)	0.7446 (0.3108)
500	0.2388 (0.2907)	0.4642 (0.3924)	0.5947 (0.3609)	0.6833 (0.3574)

Table 5: Mean $\sin \theta$ (standard deviation) in simulation 1 for multiSLICE across $\mathcal{R}(L^*)$ and n . l is set to 2 for all simulations, and each simulation is repeated 100 times.

Layers	n	JMMLE	MLGGM	multiSLICE
2	100	–	0.052 (0.011)	0.089 (0.041)
	200	–	0.089 (0.018)	0.173 (0.035)
	300	0.031 (0.046)	0.135 (0.018)	0.258 (0.021)
	400	0.030 (0.045)	0.182 (0.018)	0.297 (0.042)
	500	0.051 (0.068)	0.220 (0.016)	0.337 (0.039)
3	100	–	0.054 (0.009)	0.090 (0.039)
	200	–	0.081 (0.010)	0.172 (0.039)
	300	–	0.113 (0.013)	0.259 (0.016)
	400	0.014 (0.002)	0.143 (0.013)	0.291 (0.039)
	500	0.022 (0.028)	0.170 (0.012)	0.335 (0.038)
4	100	–	0.056 (0.008)	0.089 (0.040)
	200	–	0.077 (0.008)	0.180 (0.019)
	300	–	0.102 (0.009)	0.259 (0.013)
	400	0.009 (0.001)	0.126 (0.010)	0.285 (0.041)
	500	0.036 (0.047)	0.145 (0.009)	0.341 (0.031)
5	100	–	0.058 (0.007)	0.089 (0.037)
	200	–	0.076 (0.007)	0.178 (0.027)
	300	–	0.095 (0.008)	0.259 (0.013)
	400	–	0.114 (0.007)	0.289 (0.041)
	500	0.065 (0.060)	0.130 (0.008)	0.337 (0.036)

Table 6: Mean F1 scores (standard deviation) for JMMLE, MLGGM, and multiSLICE methods for various values of l and n in simulation 1. Bold values represent the best (highest) mean F1 score for each combination of l and n across all methods.

$\mathcal{R}(\mathbf{L}^*)$	n	JMMLE	MLGGM	coglasso	multiSLICE
2	100	–	0.052 (0.011)	–	0.089 (0.041)
	200	–	0.089 (0.018)	–	0.173 (0.035)
	300	0.031 (0.046)	0.135 (0.018)	–	0.258 (0.021)
	400	0.030 (0.045)	0.182 (0.018)	–	0.297 (0.042)
	500	0.051 (0.068)	0.220 (0.016)	–	0.337 (0.039)
3	100	–	0.052 (0.012)	–	0.079 (0.043)
	200	0.014 (–)	0.086 (0.016)	0.200 (0.003)	0.147 (0.052)
	300	0.049 (0.062)	0.132 (0.017)	0.218 (0.015)	0.251 (0.030)
	400	0.045 (0.060)	0.175 (0.016)	0.244 (0.036)	0.295 (0.040)
	500	0.090 (0.098)	0.214 (0.018)	0.216 (0.066)	0.321 (0.054)
4	100	–	0.049 (0.010)	–	0.078 (0.039)
	200	0.010 (–)	0.083 (0.014)	0.192 (0.033)	0.145 (0.052)
	300	0.030 (0.038)	0.128 (0.017)	0.240 (0.023)	0.235 (0.041)
	400	0.053 (0.059)	0.167 (0.018)	0.259 (0.028)	0.288 (0.037)
	500	0.097 (0.102)	0.204 (0.019)	0.242 (0.064)	0.320 (0.050)
5	100	–	0.050 (0.011)	–	0.087 (0.034)
	200	0.048 (0.049)	0.083 (0.015)	0.164 (0.009)	0.137 (0.058)
	300	0.043 (0.043)	0.122 (0.017)	0.190 (0.036)	0.230 (0.035)
	400	0.033 (0.037)	0.160 (0.018)	0.242 (0.051)	0.282 (0.035)
	500	0.092 (0.093)	0.194 (0.017)	0.226 (0.047)	0.322 (0.052)

Table 7: Mean F1 scores (standard deviation) for JMMLE, MLGGM, coglasso, and multiSLICE methods for various values of $\mathcal{R}(\mathbf{L}^*)$ and n . Bold values represent the best (highest) mean F1 score for each combination of $\mathcal{R}(\mathbf{L}^*)$ and n across all methods.

12 Additional simulation studies

In this section, we provide additional simulation studies that are not considered in the main article. In particular, these simulations consider the empirical convergence of Algorithm 2 for multiSLICE, hyperparameter selection of ρ and r , the effect of using other sparse estimation procedures, and the computational runtime of multiSLICE.

Figure 8 shows the empirical convergence of multiSLICE’s Algorithm 2 for various values of $\mathcal{R}(\mathbf{L}^*)$ for fixed parameters $l = 2$ and $n = 10,000$. The matrix completion step requires only a single pass through all layers, hence we omit it for the purposes of this comparison. We find that increasing $\mathcal{R}(\mathbf{L}^*)$ increases the number of iterations required for the algorithm to converge. In general, the algorithm converges quickly and requires only a few iterations to complete.

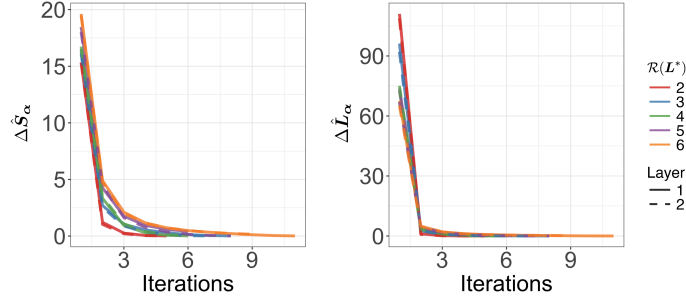


Figure 8: A simulation to study the convergence of multiSLICE for various values of $\mathcal{R}(\mathbf{L}^*)$. $\Delta\hat{\mathbf{S}}_\alpha$ and $\Delta\hat{\mathbf{L}}_\alpha$ are the change in Frobenius norm over subsequent iterations of the algorithm.

We also study the effect of hyperparameter selection on accuracy performance. Here, $\mathcal{R}(\mathbf{L}^*)$ is set to 7, $l = 3$ and $p = 50$, with an overlap of 10 in each layer. We use three-fold CV over a grid of r and ρ values. Figure 9 shows the results of the model selection over 100 iterations of simulated data. We find that the most commonly chosen hyperparameters are $\rho = 0.0108, 0.0139$ and $r = 6, 7$. The alignment between the selected ranks ($r = 6, 7$) and the true rank $\mathcal{R}(\mathbf{L}^*) = 7$ demonstrates that three-fold CV is an effective method for hyperparameter selection.

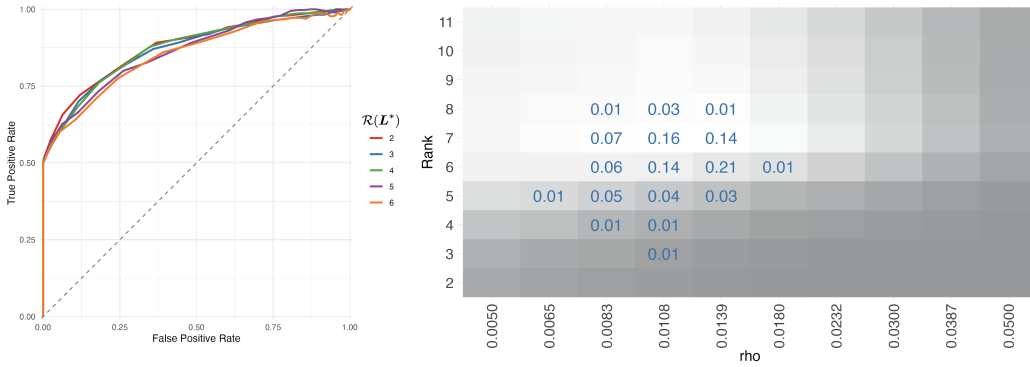


Figure 9: Left: ROC curves computed over a range of ρ and $\mathcal{R}(\mathbf{L}^*)$ values. Right: A study of the effect of model selection (ρ and r). The value in each tile is the proportion of occasions the associated r and ρ value were chosen, and tiles are colored lighter for higher average likelihoods.

Next, we study the performance of multiSLICE for various values of ρ and $\mathcal{R}(\mathbf{L}^*)$. For this simulation, we use the following parameters: $p = 100$, $l = 2$ and $n = 10,000$. Figure 9 shows the ROC curves for various values of ρ and $\mathcal{R}(\mathbf{L}^*)$. We find that $\mathcal{R}(\mathbf{L}^*)$ does not appear to have a large effect on the ROC curves; the balance between TPR and FPR is consistent. At lower $\mathcal{R}(\mathbf{L}^*)$, there is slightly better performance, although this effect appears to be quite minimal.

We also investigate the estimation performance of other sparse procedures in multiSLICE. In particular, we simulate a three-layer system with $\mathcal{R}(\mathbf{L}^*) = 2$, and observe the differences in estimates between

GLASSO, GSCAD, and CLIME. Figure 10 shows heatmaps of the true graphical structures and the graphical structures estimated by multiSLICE for different sparse estimators. GLASSO and GSCAD are very similar and at the same level of ρ produce estimates that are slightly more sparse than CLIME. However, CLIME produces less sparse estimates for the same ρ , but appears to be closer to the ground truth.

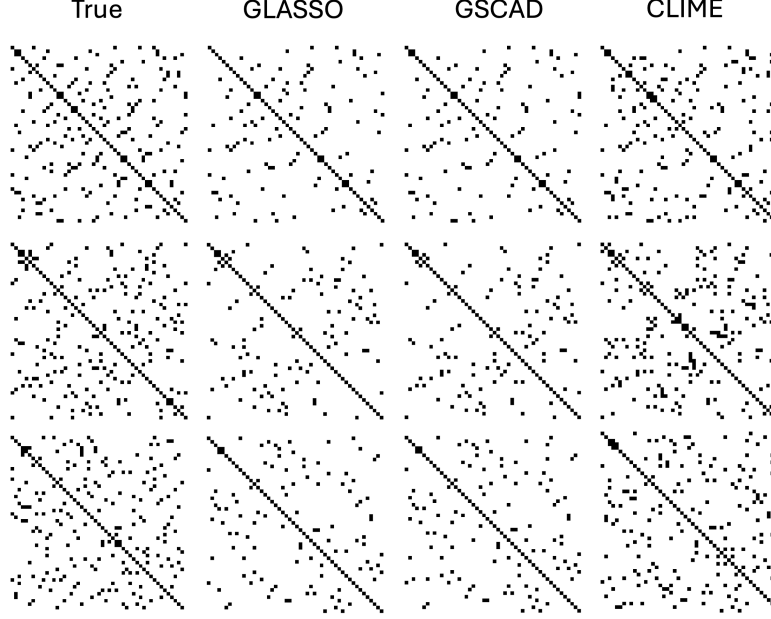


Figure 10: Heatmaps of the true graphical structures and the graphical structures estimated by multiSLICE for different sparse estimators GLASSO, GSCAD and CLIME. Each row shows the estimated sparse components for layers one to three.

Lastly, we review the computational runtime of multiSLICE. Table 8 shows the runtime of multiSLICE with different sparse estimators on a single iteration of the simulated data. HUGE GLASSO refers to the implementation of GLASSO from the HUGE R package (Zhao et al., 2012), and the remaining methods are implemented using code from their respective papers. All computations are performed sequentially (no parallelization), on a M1 Macbook Pro with 16GB of RAM and R 4.3.3. In lower dimensions, the base GLASSO implementation works well and only at $p = 1600$ does HUGE GLASSO have a shorter computation time. GSCAD is approximately 2-3x slower than GLASSO for each p above $p = 100$, and CLIME is by far the slowest. For higher dimensions, $p = 3200$ GSCAD does not complete within the 10 minute window, and CLIME does not complete within the 10 minute window for $p = 800, 1600$, and 3200 . Our results suggest that multiSLICE estimation is feasible on standard consumer-grade machines, even in high dimensions. Furthermore, computation speed could be improved by using parallelization across layers or other approaches for ultra-high dimensions.

p	GLASSO	HUGE GLASSO	GSCAD	CLIME
100	0.1048	8.1892	0.0525	0.7824
200	0.0922	5.8666	0.1825	12.7534
400	0.9844	7.9362	1.5141	74.4429
800	4.9708	8.0310	12.4409	NA
1600	32.1637	19.5103	69.9466	NA
3200	321.6993	177.8931	NA	NA

Table 8: A study of the computational runtime of multiSLICE using different sparse estimators for one iteration of simulated data with $l = 2$, $\mathcal{R}(\mathbf{L}^*) = 2$. Time is reported in seconds and DNF means the method did not finish within 10 minutes. Bold values represent the shortest time for each p .

13 Multimodal neuroimaging data

In Wakeman and Henson (2015), fMRI data were acquired with voxel sizes ranging from $3 \times 3 \times 3.75$ mm to $3 \times 3 \times 4.05$ mm using a 3 Tesla scanner, with a Repetition Time (TR) of 2000 ms and Echo Time (TE) of 30 ms. Structural MRI (sMRI) used a T1-weighted MPRAGE sequence with 1 mm isotropic resolution. MEG and EEG recordings were collected simultaneously using a sampling rate of 1100 Hz. The EEG setup involved a 70-channel cap, while MEG employed magnetometers and planar gradiometers.

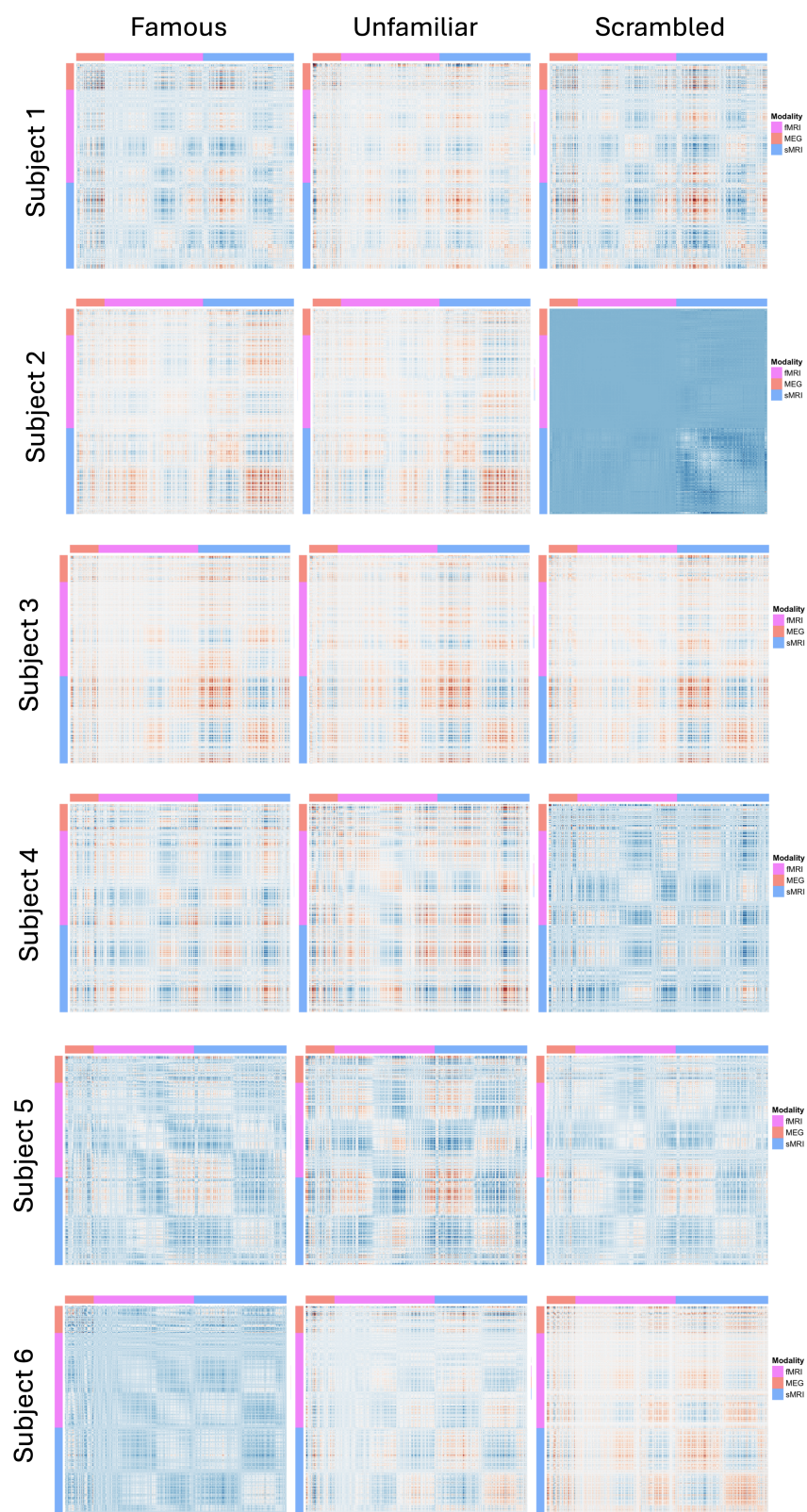
We follow a standard pre-processing procedure that is derived from Wakeman and Henson (2015) and uses statistical parametric mapping (SPM: <https://www.fil.ion.ucl.ac.uk/spm/>). For illustrative purposes, examples of stimuli presented to subjects for the three conditions (“Famous”, “Unfamiliar”, and “Scrambled”) are shown in Figure 11.

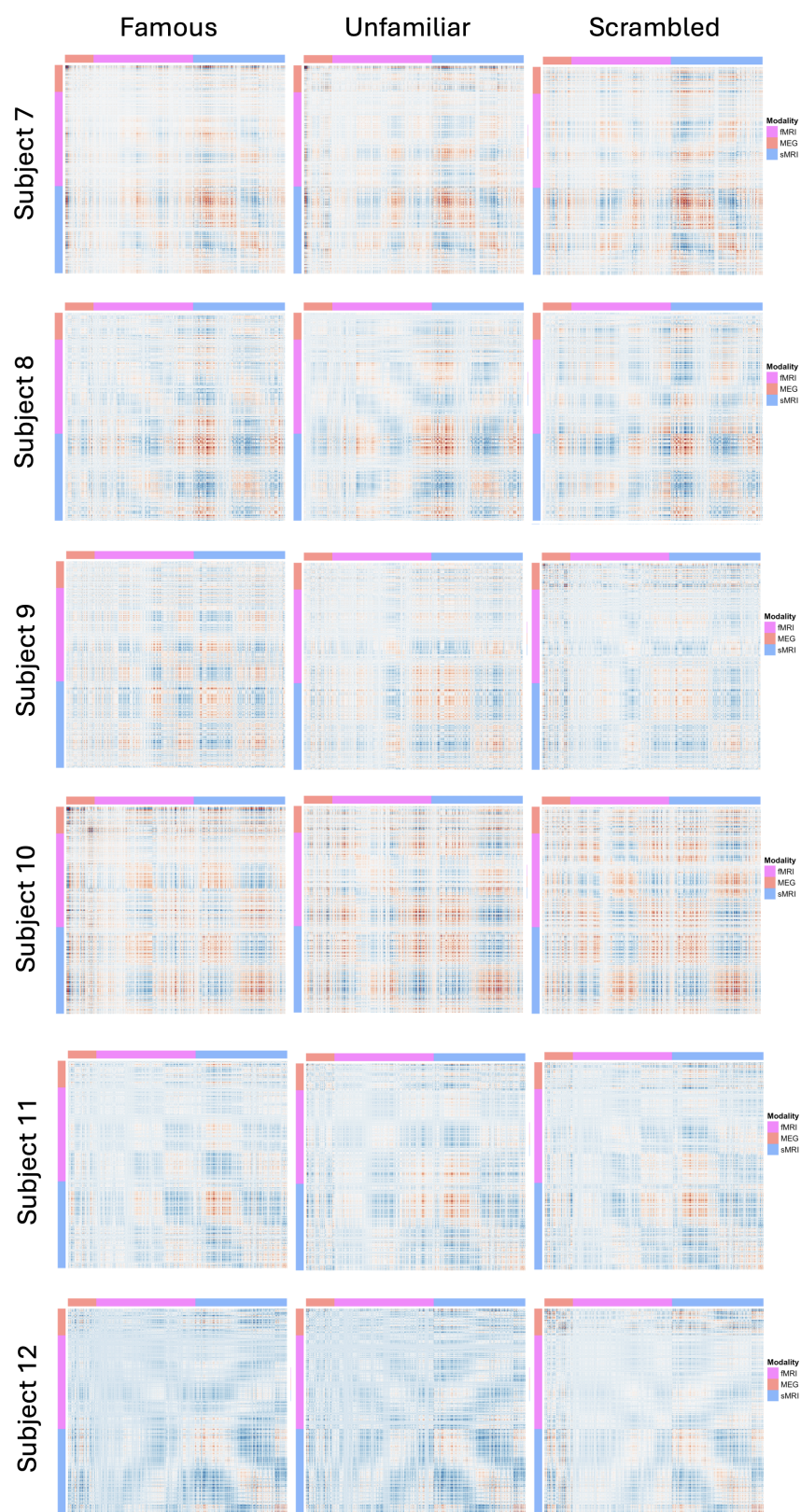


Figure 11: Example stimuli from experimental data. From left to right shown are “Famous”, “Unfamiliar”, and “Scrambled” stimuli. Images are obtained directly from the Wakeman and Henson (2015) dataset.

The next step is to define the regions of interest (ROIs), scaled according to the spatial resolution of the modality. MRI offers superior spatial resolution (approximately 1 mm) compared to M/EEG (approximately 1 cm) (Wakeman and Henson, 2015). Given the higher spatial resolution of fMRI and sMRI, we use 20 mm cubes for both structural magnetic resonance imaging (sMRI) and functional magnetic resonance imaging (fMRI) data, and 40 mm cubes for magnetoencephalography (MEG) data. All spatial measurements are referenced to the Montreal Neurological Institute (MNI) space. For MEG and fMRI data, we take ROIs and average voxel signals within the respective cube. This is carried out across all time points to yield a time series for all ROIs for each modality. For sMRI, we utilize a morphometric similarity network approach as described in Seidlitz et al. (2018). Here, we compute 15 attributes for each voxel in lieu of samples, namely summary statistics of gray and white matter, and mesh-based surface statistics which are extracted through the computational anatomy toolbox (CAT: <https://neuro-jena.github.io/cat/>) for SPM.

For each node within each modality, we regress out the global mean across all nodes in that modality. To mitigate autocorrelation, we apply a first-order lagged difference. ROIs are then finally scaled to $\mu = 0$ and unit variance. The final dimensions for each subject are $p = 52, 182$, and 188 , and $n = 101, 276$, and 15 for each subject, for MEG, fMRI, and sMRI, respectively. Figure 13 shows the recovered latent graphs for the 16 subjects. We find that each subject has a unique low-rank structure, which remains relatively consistent between the facial stimuli tasks. This may suggest that for each subject there is some variability in how facial stimuli are being processed. For each subject, between tasks, there are some small differences, which indicate task-related variability in facial processing. Generally speaking, the observations we made about subject 1 in the main paper hold for the remaining subjects in the experiment.





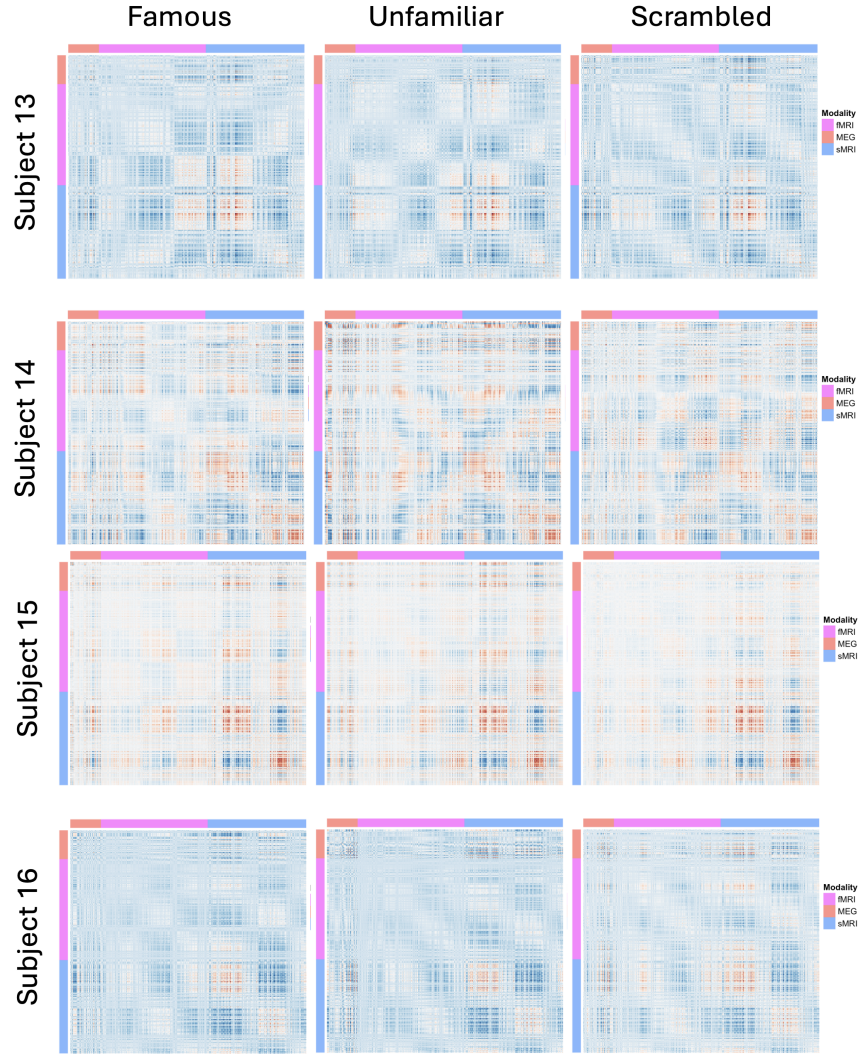


Figure 12: The recovered \hat{L} from multiSLICE for the 16 subjects and all facial stimuli (Famous, Unfamiliar, Scrambled).