Curves, Connections, and Conditional Dependence

Developments in Functional Graphical Models for Neuroimaging Data

---

Alexander Petersen

September 16, 2026

IMSI Workshop: Connectomics, Non-Euclidean Data, and Statistics

Department of Statistics, Brigham Young University

Joint work with J. Zapata, S.-Y. Oh, A. Pope, G. Page, and P. Platts

Supported by NSF Awards DMS-2310943 and DMS-2128589

Outline

From Classical Graphical Models to Functional GMs

Partial Separability and the Functional GGM

Testing for Levels of Separability

Beyond Truncation: FG-Lasso and a New Relaxation

Summary and Future Directions

Outline

From Classical Graphical Models to Functional GMs

Partial Separability and the Functional GGM

Testing for Levels of Separability

Beyond Truncation: FG-Lasso and a New Relaxation

Summary and Future Directions

Gaussian Graphical Models: The Basics

Let $\theta = (\theta_1, \ldots, \theta_p)^\top$ be a mean-zero random vector.

- $\Sigma = \mathrm{Var}(\theta)$: the **covariance matrix**
- $\Omega = \Sigma^{-1}$: the **precision matrix**

Gaussian Graphical Models: The Basics

Let $\theta = (\theta_1, \ldots, \theta_p)^\top$ be a mean-zero random vector.

- $\Sigma = \mathrm{Var}(\theta)$: the **covariance matrix**
- $\Omega = \Sigma^{-1}$: the **precision matrix**

If $\theta \sim \mathcal{N}(0, \Sigma)$, then Ω encodes conditional independence:

$$\theta_j \perp\!\!\!\perp \theta_k \mid \theta_{-(j,k)} \iff \Omega_{jk} = 0.$$

Gaussian Graphical Models: The Basics

Let $\theta = (\theta_1, \ldots, \theta_p)^\top$ be a mean-zero random vector.

- $\Sigma = \mathrm{Var}(\theta)$: the **covariance matrix**
- $\Omega = \Sigma^{-1}$: the **precision matrix**

If $\theta \sim \mathcal{N}(0, \Sigma)$, then Ω encodes conditional independence:

$$\theta_j \perp\!\!\!\perp \theta_k \mid \theta_{-(j,k)} \iff \Omega_{jk} = 0.$$

This defines an undirected graph $G = (V, E)$:

- $V = \{1, \ldots, p\}$
- $E = \{(j, k) : \Omega_{jk} \neq 0\}$

Sparse estimation (glasso, neighborhood selection) makes this tractable when p is large.

From Vectors to Functions: Neuroimaging Motivation

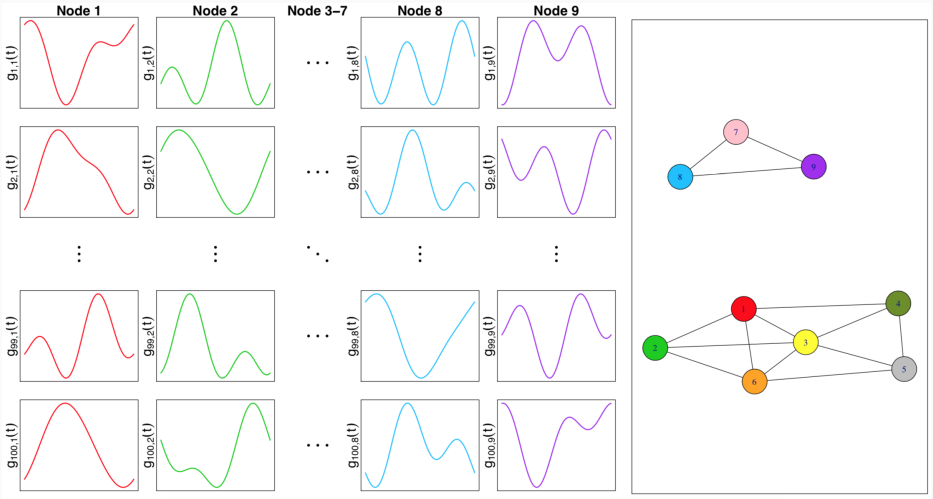
- fMRI, EEG, dynamic PET: for each subject, a time course at every brain region, sensor, or source

From Vectors to Functions: Neuroimaging Motivation

- fMRI, EEG, dynamic PET: for each subject, a time course at every brain region, sensor, or source
- Node j is no longer a scalar θ_j, but an entire curve $X_j(t)$, $t \in \mathcal{T}$

From Vectors to Functions: Neuroimaging Motivation

- fMRI, EEG, dynamic PET: for each subject, a time course at every brain region, sensor, or source
- Node j is no longer a scalar θ_j, but an entire curve $X_j(t)$, $t \in \mathcal{T}$
- Model $\boldsymbol{X}(t) = (X_1(t), \ldots, X_p(t))^\top$ as a multivariate Gaussian process; conditional independence should again correspond to a graph



Nodes as curves, edges as conditional dependence (Qiao, Guo, and James 2019).

The Functional Covariance Operator

Setting: $\boldsymbol{X}$ is a vector-valued random function in $L^2(\mathcal{T}, \mathbb{R}^p)$, $E \int_{\mathcal{T}} \|\boldsymbol{X}(t)\|^2 dt < \infty$

The Functional Covariance Operator

Setting: $\boldsymbol{X}$ is a vector-valued random function in $L^2(\mathcal{T}, \mathbb{R}^p)$, $E\int_{\mathcal{T}} \|\boldsymbol{X}(t)\|^2 dt < \infty$

- $G_{jk}(s,t) = \mathrm{Cov}(X_j(s), X_k(t))$ is the covariance **kernel** between nodes j, k; the associated operator $\mathcal{G}_{jk}$ acts by

$$(\mathcal{G}_{jk} f)(s) = \int_{\mathcal{T}} G_{jk}(s,t) f(t)\, dt.$$

The Functional Covariance Operator

Setting: $\boldsymbol{X}$ is a vector-valued random function in $L^2(\mathcal{T}, \mathbb{R}^p)$, $E\int_{\mathcal{T}} \|\boldsymbol{X}(t)\|^2 dt < \infty$

- $G_{jk}(s,t) = \mathrm{Cov}(X_j(s), X_k(t))$ is the covariance **kernel** between nodes j, k; the associated operator $\mathcal{G}_{jk}$ acts by

$$(\mathcal{G}_{jk} f)(s) = \int_{\mathcal{T}} G_{jk}(s,t) f(t)\, dt.$$

- Collecting all pairs gives the block operator $\boldsymbol{\mathcal{G}} = \{\mathcal{G}_{jk}\}_{j,k=1}^{p}$

The Functional Covariance Operator

Setting: $\boldsymbol{X}$ is a vector-valued random function in $L^2(\mathcal{T}, \mathbb{R}^p)$, $E\int_{\mathcal{T}} \|\boldsymbol{X}(t)\|^2 dt < \infty$

- $G_{jk}(s,t) = \mathrm{Cov}(X_j(s), X_k(t))$ is the covariance **kernel** between nodes j, k; the associated operator $\mathcal{G}_{jk}$ acts by

$$(\mathcal{G}_{jk} f)(s) = \int_{\mathcal{T}} G_{jk}(s,t) f(t)\, dt.$$

- Collecting all pairs gives the block operator $\boldsymbol{\mathcal{G}} = \{\mathcal{G}_{jk}\}_{j,k=1}^{p}$
- Key obstacle: $\boldsymbol{\mathcal{G}}$ is **compact** (trace-class) $\Rightarrow$ its inverse is **unbounded** — there is no functional analogue of $\Omega = \Sigma^{-1}$ without extra structure

This single fact drives most of the methodology in this talk.

Conditional Independence for Functional Data

For $\boldsymbol{X}$ a mean-zero multivariate Gaussian process, let $X_{-jk} = \{X_l : l \neq j, k\}$. Then

$$X_j \perp\!\!\!\perp X_k \mid X_{-jk} \iff G_{jk|-jk}(s,t) = 0 \ \ \forall\, s,t \in \mathcal{T},$$

where $G_{jk|-jk}$ is the conditional cross-covariance kernel of the operator $\mathcal{G}_{jk|-jk}$.

Conditional Independence for Functional Data

For $\boldsymbol{X}$ a mean-zero multivariate Gaussian process, let $X_{-jk} = \{X_l : l \neq j, k\}$. Then

$$X_j \perp\!\!\!\perp X_k \mid X_{-jk} \iff G_{jk|-jk}(s,t) = 0 \ \ \forall\, s,t \in \mathcal{T},$$

where $G_{jk|-jk}$ is the conditional cross-covariance kernel of the operator $\mathcal{G}_{jk|-jk}$.

This defines the functional graphical model $G = (V, E)$:

$$E = \{(j,k) : \mathcal{G}_{jk|-jk} \neq 0\}.$$

Conditional Independence for Functional Data

For $\boldsymbol{X}$ a mean-zero multivariate Gaussian process, let $X_{-jk} = \{X_l : l \neq j, k\}$. Then

$$X_j \perp\!\!\!\perp X_k \mid X_{-jk} \iff G_{jk|-jk}(s,t) = 0 \ \ \forall\, s,t \in \mathcal{T},$$

where $G_{jk|-jk}$ is the conditional cross-covariance kernel of the operator $\mathcal{G}_{jk|-jk}$.

This defines the functional graphical model $G = (V, E)$:

$$E = \{(j,k) : \mathcal{G}_{jk|-jk} \neq 0\}.$$

Two broad strategies appear in the literature, differing in how they cope with $\mathcal{G}$'s non-invertibility.

Two Families of Methods

With dimension reduction (truncate via FPCA, then borrow multivariate tools)

- Gaussian: functional graphical lasso (Qiao, Guo, and James 2019); Bayesian FGMs (Zhu, Strawn, and Dunson 2016); neighborhood selection (Zhao et al. 2024); partial separability (Zapata, Oh, and Petersen 2022)
- Non-Gaussian: copula-transformed scores (Solea and Li 2022); additive/nonparametric scores (Solea and Dette 2022)

Two Families of Methods

With dimension reduction (truncate via FPCA, then borrow multivariate tools)

- Gaussian: functional graphical lasso (Qiao, Guo, and James 2019); Bayesian FGMs (Zhu, Strawn, and Dunson 2016); neighborhood selection (Zhao et al. 2024); partial separability (Zapata, Oh, and Petersen 2022)
- Non-Gaussian: copula-transformed scores (Solea and Li 2022); additive/nonparametric scores (Solea and Dette 2022)

Without dimension reduction (work directly with operators)

- Correlation-operator based (Li and Solea 2018); functional graphical lasso on the full operator (Waghmare, Masak, and Panaretos 2025); CI testing via the Generalized Hilbertian Covariance Measure (Lundborg, Shah, and Peters 2022)

See Zhao and Lee (2026, WIREs Comp. Stat.) for a comprehensive review, including directed FGMs (PC-type algorithms, functional SEMs, Func-LiNGAM), set aside today.

Two Truncation-Based Extensions of Classical GGM Methods

Common first step: truncate each curve $X_{ij}(\cdot)$ via FPCA to K scores per node.

Two Truncation-Based Extensions of Classical GGM Methods

Common first step: truncate each curve $X_{ij}(\cdot)$ via FPCA to K scores per node.

- Functional graphical lasso (Qiao, Guo, and James 2019) extends the **graphical lasso**: apply a group-lasso penalty to the blocks of the resulting (truncated) precision matrix, so that whole functional edges are kept or removed together. Solved by block coordinate descent; requires each X_j to lie (a.s.) in a finite-dimensional space.

Two Truncation-Based Extensions of Classical GGM Methods

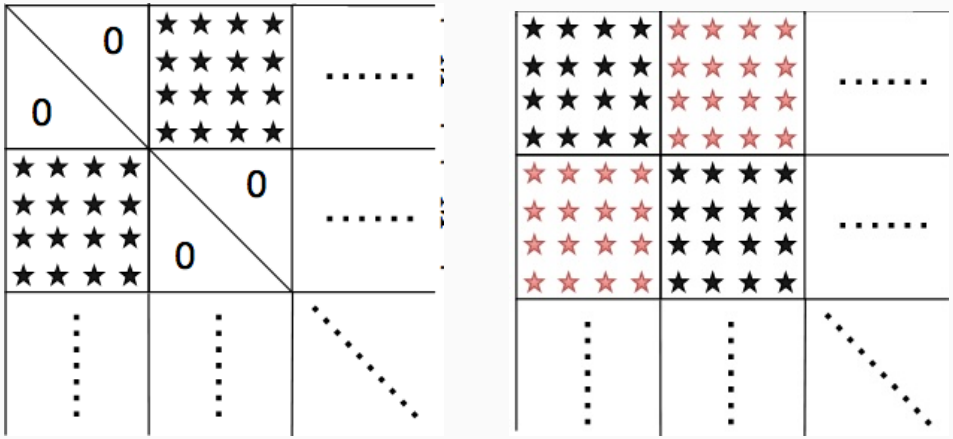
Common first step: truncate each curve $X_{ij}(\cdot)$ via FPCA to K scores per node.

- Functional graphical lasso (Qiao, Guo, and James 2019) extends the **graphical lasso**: apply a group-lasso penalty to the blocks of the resulting (truncated) precision matrix, so that whole functional edges are kept or removed together. Solved by block coordinate descent; requires each X_j to lie (a.s.) in a finite-dimensional space.
- Neighborhood selection (Zhao et al. 2024) extends **neighborhood selection**: regress each node's (truncated) curve X_{ij} on those of every other node, and select its neighbors via a group-lasso penalty across nodes. Never requires an explicit precision operator.

Both still rely on FPCA truncation at every node — with the same consequence, examined next.

The Pitfall: Truncation Induces Bias

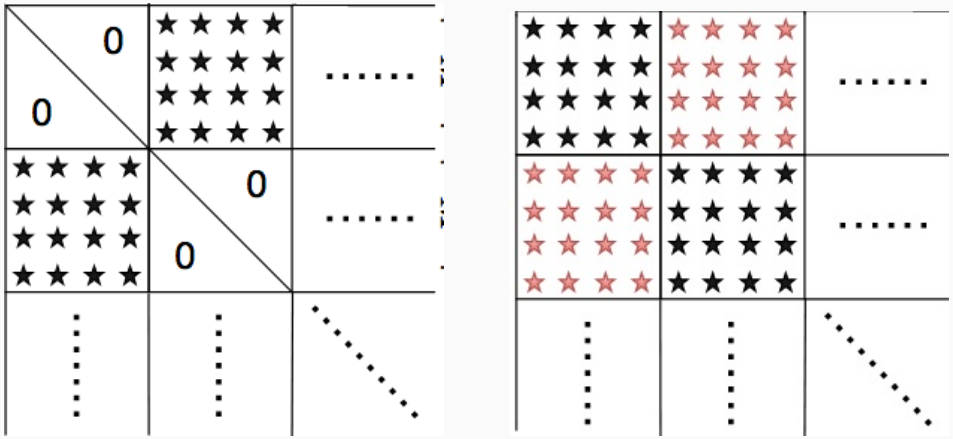
Both approaches replace the infinite-dimensional conditioning variable X_{-jk} with a K-truncated proxy. This changes the target.



Left: block structure of Σ from separate, per-node FPCA. Right: its inverse — truncation introduces spurious ($\star$) nonzero blocks not present in the true precision structure.

The Pitfall: Truncation Induces Bias

Both approaches replace the infinite-dimensional conditioning variable X_{-jk} with a K-truncated proxy. This changes the target.



Left: block structure of $\boldsymbol{\Sigma}$ from separate, per-node FPCA. Right: its inverse — truncation introduces spurious ($\star$) nonzero blocks not present in the true precision structure.

This bias need **not** vanish as $n, K \to \infty$

Outline

From Classical Graphical Models to Functional GMs

Partial Separability and the Functional GGM

Testing for Levels of Separability

Beyond Truncation: FG-Lasso and a New Relaxation

Summary and Future Directions

Separability: Viewing X as a Tensor

- $\boldsymbol{X}$ is a tensor data object: $L^2(\mathcal{T}, \mathbb{R}^p) \cong L^2(\mathcal{T}) \otimes \mathbb{R}^p$

Separability: Viewing X as a Tensor

- $\boldsymbol{X}$ is a tensor data object: $L^2(\mathcal{T}, \mathbb{R}^p) \cong L^2(\mathcal{T}) \otimes \mathbb{R}^p$
- Separability: suppose the covariance itself factors as a tensor,

$$\boldsymbol{\mathcal{G}} = \mathcal{H} \otimes \boldsymbol{A},$$

  a "time pattern" $\mathcal{H}$ and a "region pattern" $\boldsymbol{A}$

Separability: Viewing X as a Tensor

- $\boldsymbol{X}$ is a tensor data object: $L^2(\mathcal{T}, \mathbb{R}^p) \cong L^2(\mathcal{T}) \otimes \mathbb{R}^p$
- Separability: suppose the covariance itself factors as a tensor,

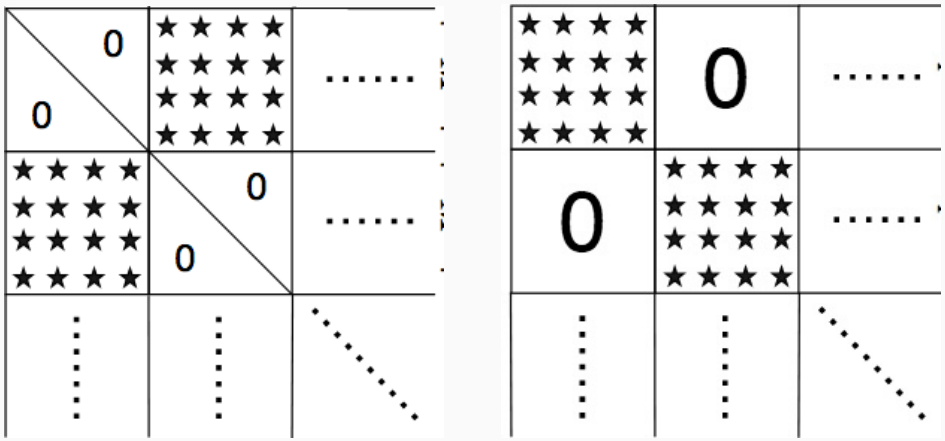
$$\boldsymbol{\mathcal{G}} = \mathcal{H} \otimes \boldsymbol{A},$$

  a "time pattern" $\mathcal{H}$ and a "region pattern" $\boldsymbol{A}$

- Expanding in the tensor basis $\{\phi_l e_j\}$ (marginal eigenfunctions ϕ_l of $\mathcal{H}$, eigenvectors e_j of $\boldsymbol{A}$) gives coefficients $\boldsymbol{\Lambda} = \{\boldsymbol{\Lambda}_{lm}\}$, $\boldsymbol{\Lambda}_{lm} = \{\lambda_{lmjk}\}_{j,k=1}^{p}$
- Separability $\iff$ $\boldsymbol{\Lambda} = B \otimes \Gamma$: a **diagonal tensor array** — very restrictive in practice

Partial Separability: A Useful Relaxation

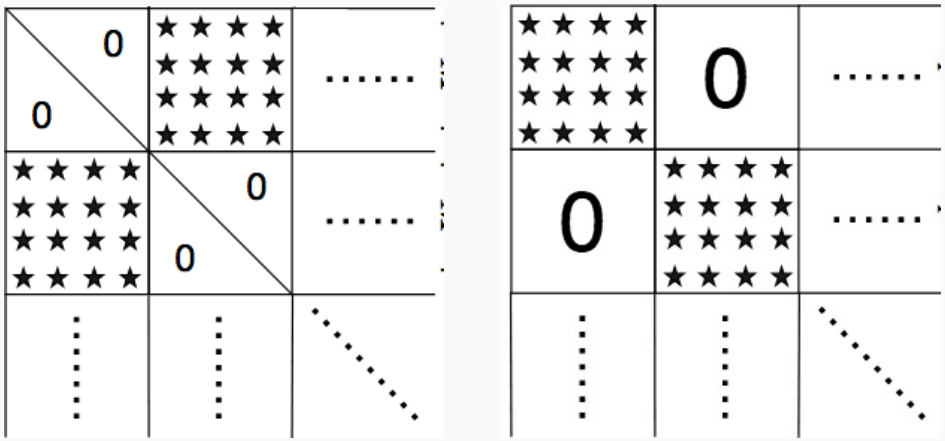
- Zapata, Oh, and Petersen (2022) relax to: $\mathbf{\Lambda}$ is **block-diagonal** ($\mathbf{\Lambda}_{lm} = 0$ for $l \neq m$), without requiring each block to be a scalar multiple of a fixed matrix



Left: generic (non-separable) block structure. Right: partial separability — off-diagonal blocks vanish, but each diagonal block $\mathbf{\Lambda}_{ll}$ is unrestricted.

Partial Separability: A Useful Relaxation

- Zapata, Oh, and Petersen (2022) relax to: $\mathbf{\Lambda}$ is **block-diagonal** ($\mathbf{\Lambda}_{lm} = 0$ for $l \neq m$), without requiring each block to be a scalar multiple of a fixed matrix



Left: generic (non-separable) block structure. Right: partial separability — off-diagonal blocks vanish, but each diagonal block $\mathbf{\Lambda}_{ll}$ is unrestricted.

Equivalently, a coupled Karhunen–Loève expansion holds:

$$\boldsymbol{X}(t) = \sum_{l=1}^{\infty} \theta_l \phi_l(t), \quad \theta_l = (\theta_{l1}, \ldots, \theta_{lp})^\top \text{ uncorrelated across } l.$$

Conditional Independence under Partial Separability

Theorem

If $\boldsymbol{X}$ is Gaussian with $\mathcal{G}$ partially separable, and $\kappa_{ljk} = \mathrm{Cov}(\theta_{lj}, \theta_{lk} \mid \theta_{l,-jk})$, *then*

$$G_{jk|-jk}(s,t) = \sum_{l=1}^{\infty} \kappa_{ljk}\, \phi_l(s)\phi_l(t).$$

Conditional Independence under Partial Separability

Theorem

If $\boldsymbol{X}$ is Gaussian with $\mathcal{G}$ partially separable, and $\kappa_{ljk} = \mathrm{Cov}(\theta_{lj}, \theta_{lk} \mid \theta_{l,-jk})$, then

$$G_{jk|-jk}(s,t) = \sum_{l=1}^{\infty} \kappa_{ljk}\, \phi_l(s)\phi_l(t).$$

Corollary

Let $\Omega_l = \Sigma_l^{-1}$ where $\Sigma_l = \mathrm{Var}(\theta_l)$. Then

$$E = \bigcup_{l=1}^{\infty} E_l, \qquad (j,k) \notin E_l \iff \{\Omega_l\}_{jk} = 0.$$

Conditional Independence under Partial Separability

Theorem

If $\boldsymbol{X}$ *is Gaussian with* $\mathcal{G}$ *partially separable, and* $\kappa_{ljk} = \mathrm{Cov}(\theta_{lj}, \theta_{lk} \mid \theta_{l,-jk})$*, then*

$$G_{jk|-jk}(s,t) = \sum_{l=1}^{\infty} \kappa_{ljk}\, \phi_l(s)\phi_l(t).$$

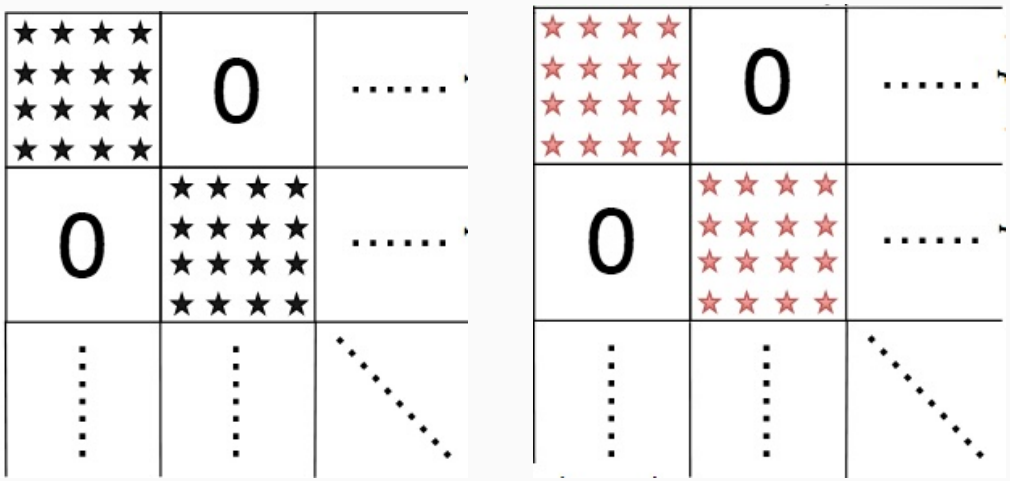
Corollary

Let $\Omega_l = \Sigma_l^{-1}$ *where* $\Sigma_l = \mathrm{Var}(\theta_l)$*. Then*

$$E = \bigcup_{l=1}^{\infty} E_l, \qquad (j,k) \notin E_l \iff \{\Omega_l\}_{jk} = 0.$$

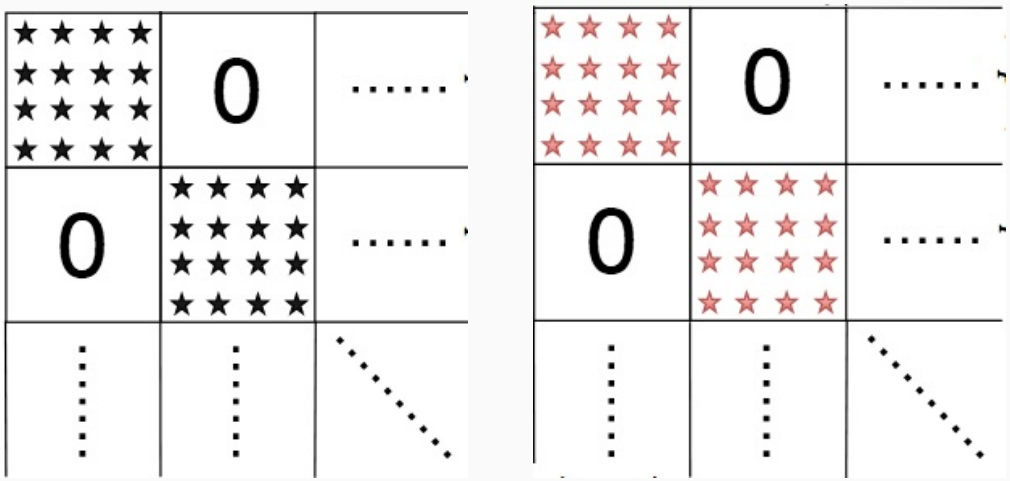
Payoff: an infinite-dimensional graphical problem becomes a **sequence of ordinary (uncorrelated) multivariate GGM problems** — one per level l.

Truncation Is Now Quantifiable



Under partial separability, Σ (left) and its inverse (right) share the same block-diagonal zero pattern — truncating at level L discards only the blocks $l > L$, with no fill-in bias in the retained blocks.

Truncation Is Now Quantifiable



Under partial separability, Σ (left) and its inverse (right) share the same block-diagonal zero pattern — truncating at level L discards only the blocks $l > L$, with no fill-in bias in the retained blocks.

Contrast with the earlier (non-PS) picture: there, truncation **fills in** spurious precision entries. Under partial separability:

- Each level contributes a clean, self-contained edge set E_l
- There always exists L_p^* with $\bigcup_{l \geq 1} E_l = \bigcup_{l \leq L_p^*} E_l$, so truncation bias is controlled and vanishes for L large enough

Estimation: From Data to Graph

Big picture: Observations $\xrightarrow{\text{FDA step}} \{\hat{R}_l\} \xrightarrow{\text{Graphical step}} \hat{E} = \bigcup_{l=1}^{M} \hat{E}_l$

Estimation: From Data to Graph

Big picture: Observations $\xrightarrow{\text{FDA step}} \{\hat{R}_l\} \xrightarrow{\text{Graphical step}} \hat{E} = \bigcup_{l=1}^{M} \hat{E}_l$

1. Estimate $\hat{\mathcal{G}}_{jk}$ for each pair, and $\hat{\mathcal{H}} = \sum_j \hat{\mathcal{G}}_{jj}$; extract eigenfunctions $\hat{\phi}_l$

Estimation: From Data to Graph

Big picture: Observations $\xrightarrow{\text{FDA step}} \{\hat{R}_l\} \xrightarrow{\text{Graphical step}} \hat{E} = \bigcup_{l=1}^{M} \hat{E}_l$

1. Estimate $\hat{\mathcal{G}}_{jk}$ for each pair, and $\hat{\mathcal{H}} = \sum_j \hat{\mathcal{G}}_{jj}$; extract eigenfunctions $\hat{\phi}_l$
2. Project to get score covariances $\hat{\Sigma}_l$, convert to correlation scale $\hat{R}_l$

Estimation: From Data to Graph

Big picture: Observations $\xrightarrow{\text{FDA step}} \{\hat{R}_l\} \xrightarrow{\text{Graphical step}} \hat{E} = \bigcup_{l=1}^{M} \hat{E}_l$

1. Estimate $\hat{\mathcal{G}}_{jk}$ for each pair, and $\hat{\mathcal{H}} = \sum_j \hat{\mathcal{G}}_{jj}$; extract eigenfunctions $\hat{\phi}_l$
2. Project to get score covariances $\hat{\Sigma}_l$, convert to correlation scale $\hat{R}_l$
3. Apply a Joint Graphical Lasso across levels $l = 1, \ldots, M$ (Danaher et al. 2014): a lasso penalty within each Ξ_l plus a fused/group penalty linking $\Xi_1, \ldots, \Xi_M$ together

Estimation: From Data to Graph

Big picture: Observations $\xrightarrow{\text{FDA step}} \{\hat{R}_l\} \xrightarrow{\text{Graphical step}} \hat{E} = \bigcup_{l=1}^{M} \hat{E}_l$

1. Estimate $\hat{\mathcal{G}}_{jk}$ for each pair, and $\hat{\mathcal{H}} = \sum_j \hat{\mathcal{G}}_{jj}$; extract eigenfunctions $\hat{\phi}_l$
2. Project to get score covariances $\hat{\Sigma}_l$, convert to correlation scale $\hat{R}_l$
3. Apply a Joint Graphical Lasso across levels $l = 1, \ldots, M$ (Danaher et al. 2014): a lasso penalty within each Ξ_l plus a fused/group penalty linking $\Xi_1, \ldots, \Xi_M$ together
4. $\hat{E} = \bigcup_l \hat{E}_l$ from the nonzero pattern of $\{\hat{\Xi}_l\}$

Theory: consistent graph recovery once $n \gtrsim C_L \log p$, with L allowed to grow slowly with n.

HCP Motor Task Experiment

- Human Connectome Project: 1200 subjects, 3T fMRI

HCP Motor Task Experiment

- Human Connectome Project: 1200 subjects, 3T fMRI
- Glasser et al. (2016) atlas: $p = 360$ cortical ROIs, $s = 18$ BOLD time points during finger/foot tapping tasks

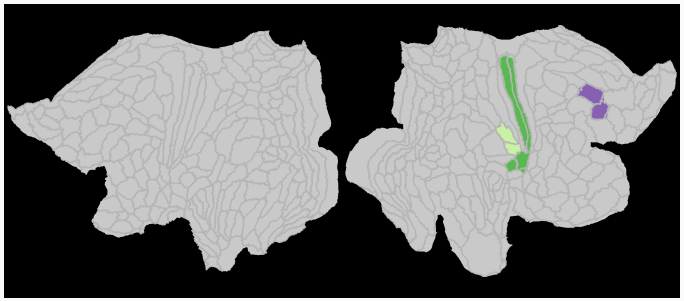
HCP Motor Task Experiment

- Human Connectome Project: 1200 subjects, 3T fMRI
- Glasser et al. (2016) atlas: $p = 360$ cortical ROIs, $s = 18$ BOLD time points during finger/foot tapping tasks
- Compare estimated connectivity during left-hand vs. right-hand tasks

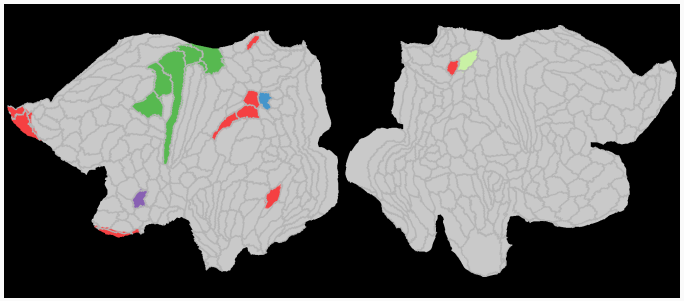
HCP Motor Task Experiment

- Human Connectome Project: 1200 subjects, 3T fMRI
- Glasser et al. (2016) atlas: $p = 360$ cortical ROIs, $s = 18$ BOLD time points during finger/foot tapping tasks
- Compare estimated connectivity during left-hand vs. right-hand tasks
- Without treating the data as functional: $\sim$21M covariance parameters. Even a low-rank ($L = 5$) functional model still has 1.6M – partial separability is what makes this tractable

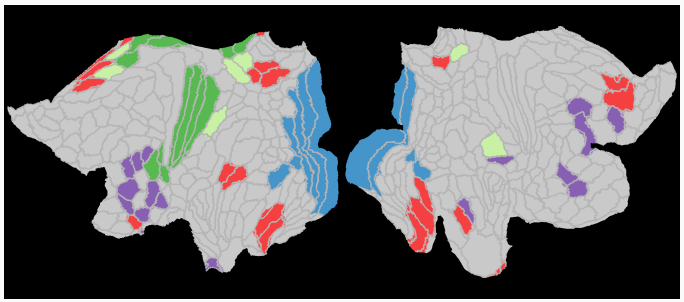
Connectivity Maps: Left- vs. Right-Hand Task



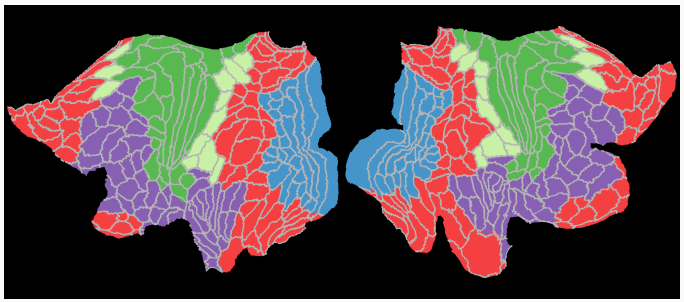
(a) Unique to left-hand



(b) Unique to right-hand



(c) Common to both



(d) Task activation map

psFGGM-estimated functionally connected ROIs, colored by functionality: visual (blue), motor (green), mixed motor (light green), mixed other (red), other (purple). Connectivity patterns respect known functional anatomy and differ meaningfully by task laterality.

Outline

From Classical Graphical Models to Functional GMs

Partial Separability and the Functional GGM

Testing for Levels of Separability

Beyond Truncation: FG-Lasso and a New Relaxation

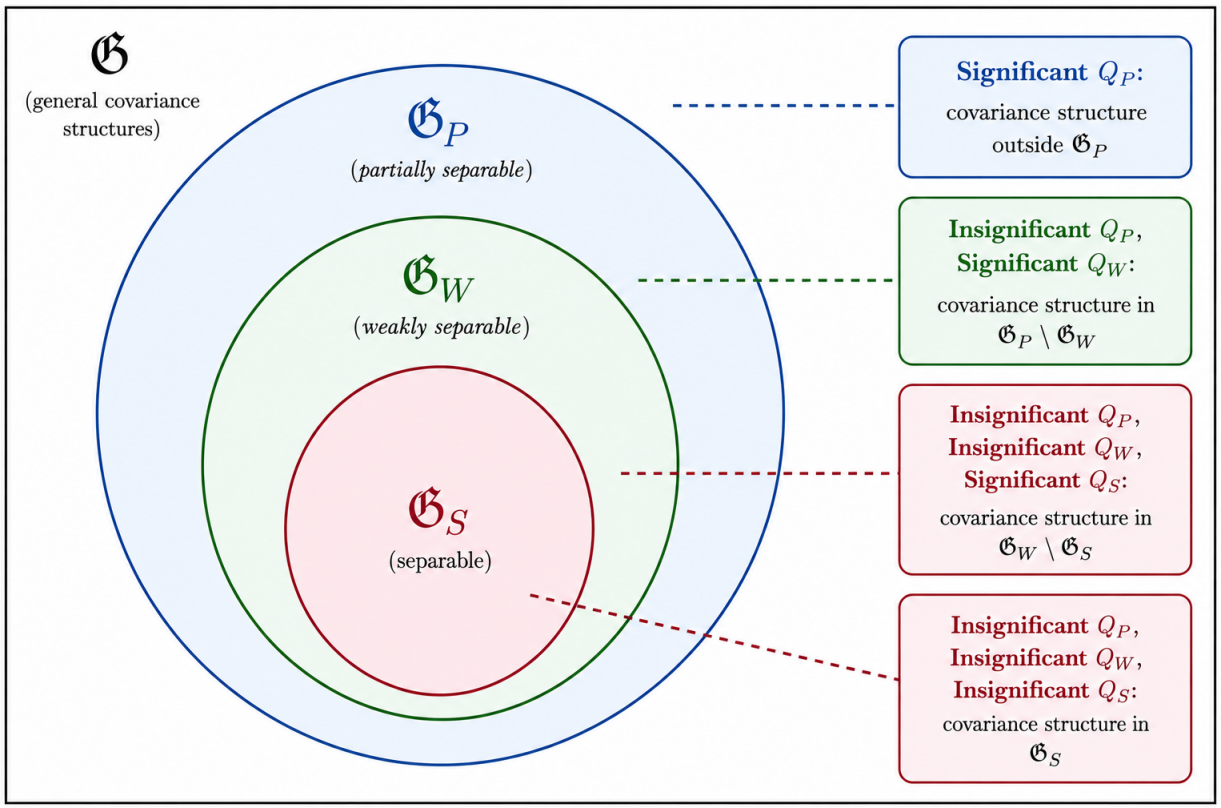
Summary and Future Directions

Which Level of Separability Applies?

- Partial separability is convenient, but is it appropriate for a given data set?

Which Level of Separability Applies?

- Partial separability is convenient, but is it appropriate for a given data set?
- There is a natural nested hierarchy of structural assumptions on $\mathcal{G}$, from most to least restrictive



The Nested Hierarchy

- Separable ($\mathfrak{G}_S$): $\mathbf{\Lambda} = B \otimes \Gamma$, a diagonal **tensor** array

The Nested Hierarchy

- Separable ($\mathfrak{G}_S$): $\mathbf{\Lambda} = B \otimes \Gamma$, a diagonal **tensor** array
- Weakly separable ($\mathfrak{G}_W$, Lynch and Chen 2018): $\mathbf{\Lambda}$ diagonal, but not necessarily a tensor product

The Nested Hierarchy

- Separable ($\mathfrak{G}_S$): $\mathbf{\Lambda} = B \otimes \Gamma$, a diagonal **tensor** array
- Weakly separable ($\mathfrak{G}_W$, Lynch and Chen 2018): $\mathbf{\Lambda}$ diagonal, but not necessarily a tensor product
- Partially separable ($\mathfrak{G}_P$, Zapata, Oh, and Petersen 2022): $\mathbf{\Lambda}$ block-diagonal

The Nested Hierarchy

- Separable ($\mathfrak{G}_S$): $\mathbf{\Lambda} = B \otimes \Gamma$, a diagonal **tensor** array
- Weakly separable ($\mathfrak{G}_W$, Lynch and Chen 2018): $\mathbf{\Lambda}$ diagonal, but not necessarily a tensor product
- Partially separable ($\mathfrak{G}_P$, Zapata, Oh, and Petersen 2022): $\mathbf{\Lambda}$ block-diagonal

Each is a genuine relaxation of the last: $\mathfrak{G}_S \subset \mathfrak{G}_W \subset \mathfrak{G}_P \subset \mathfrak{G}$.

Existing Wald-Type Tests

- Asymptotic normality of $\hat{\mathbf{\Lambda}}$ gives Wald-type statistics: square and sum estimated parameters that should vanish under the null

Existing Wald-Type Tests

- Asymptotic normality of $\hat{\mathbf{\Lambda}}$ gives Wald-type statistics: square and sum estimated parameters that should vanish under the null
- Aston, Pigoli, and Tavakoli (2017): test for separability
  - Only checks *diagonal* elements of $\mathbf{\Lambda}$ — no power if the diagonal looks tensor-like but off-diagonal blocks do not vanish
  - Standardization is difficult; bootstrap calibration required

Existing Wald-Type Tests

- Asymptotic normality of $\hat{\mathbf{\Lambda}}$ gives Wald-type statistics: square and sum estimated parameters that should vanish under the null
- Aston, Pigoli, and Tavakoli (2017): test for separability
  - Only checks *diagonal* elements of $\mathbf{\Lambda}$ — no power if the diagonal looks tensor-like but off-diagonal blocks do not vanish
  - Standardization is difficult; bootstrap calibration required
- Lynch and Chen (2018): test for weak separability
  - Even marginal standardization reduces power

Existing Wald-Type Tests

- Asymptotic normality of $\hat{\mathbf{\Lambda}}$ gives Wald-type statistics: square and sum estimated parameters that should vanish under the null
- Aston, Pigoli, and Tavakoli (2017): test for separability
  - Only checks *diagonal* elements of $\mathbf{\Lambda}$ — no power if the diagonal looks tensor-like but off-diagonal blocks do not vanish
  - Standardization is difficult; bootstrap calibration required
- Lynch and Chen (2018): test for weak separability
  - Even marginal standardization reduces power
- Tests against partial separability (or weak vs. partial) can be built the same way, by isolating the relevant components of $\mathbf{\Lambda}$

An Empirical Likelihood Alternative

- Wald-type tests don't compare estimates *across* hypotheses, so they fail to self-normalize

An Empirical Likelihood Alternative

- Wald-type tests don't compare estimates *across* hypotheses, so they fail to self-normalize
- Empirical likelihood ratio (ELR) tests instead compare the feasibility of parameter values under nested constraints — no working likelihood required

An Empirical Likelihood Alternative

- Wald-type tests don't compare estimates *across* hypotheses, so they fail to self-normalize
- Empirical likelihood ratio (ELR) tests instead compare the feasibility of parameter values under nested constraints — no working likelihood required
- Constrained estimation under H_0 (separable / weakly separable / partially separable) is implemented via `melt`-style linear-hypothesis machinery, adapted to our setting

An Empirical Likelihood Alternative

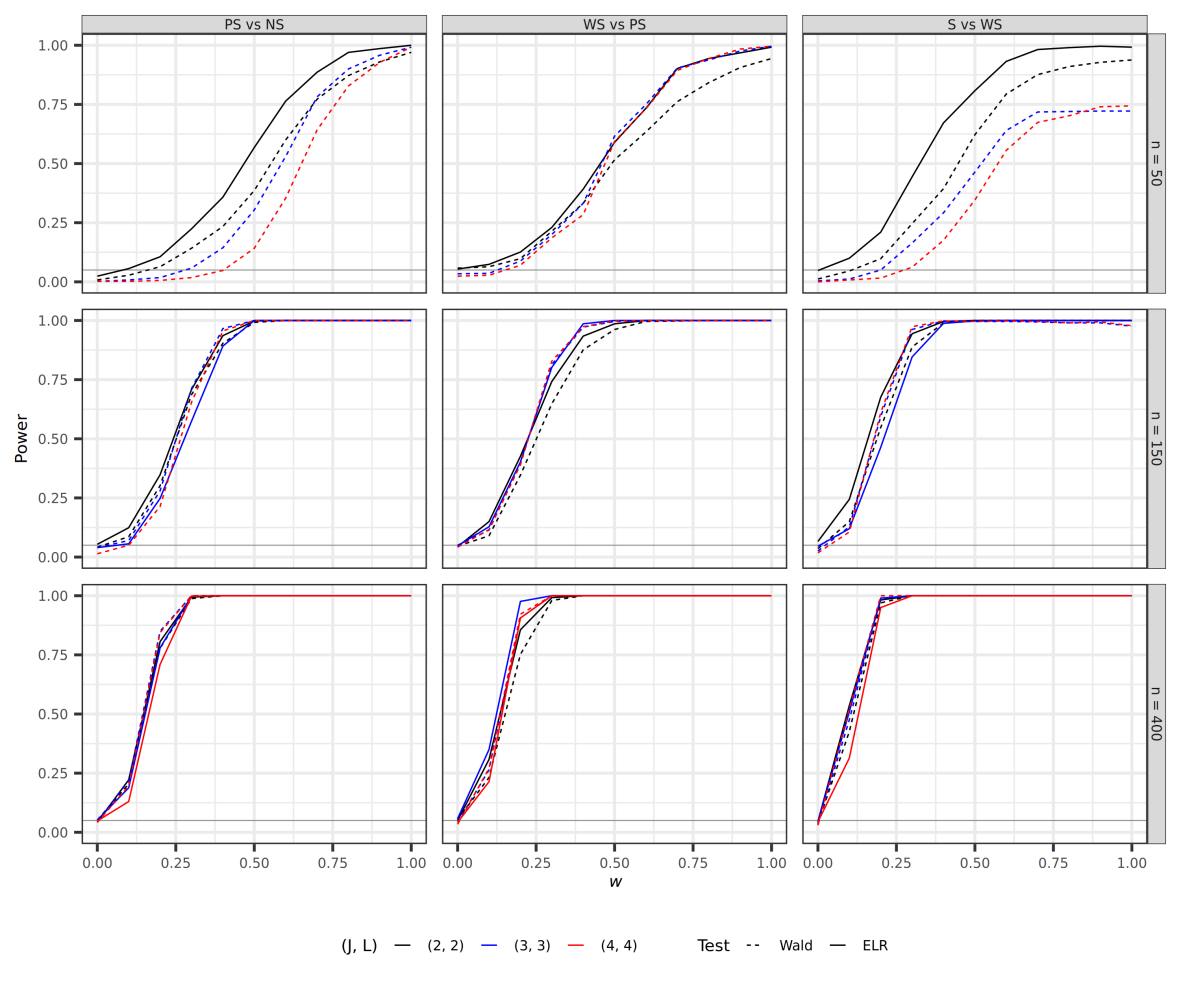
- Wald-type tests don't compare estimates *across* hypotheses, so they fail to self-normalize
- Empirical likelihood ratio (ELR) tests instead compare the feasibility of parameter values under nested constraints — no working likelihood required
- Constrained estimation under H_0 (separable / weakly separable / partially separable) is implemented via `melt`-style linear-hypothesis machinery, adapted to our setting
- Produces a genuine sequential testing procedure: test S vs. W, then W vs. P, then P vs. non-separable, to select the appropriate level

Simulation Design

- Interpolate between structures: $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{G}}_0 + w\,\boldsymbol{\mathcal{G}}_1$, with $w \in [0, 1]$ controlling distance from the null structure
- Compare Wald vs. ELR tests, at truncation levels $(J, L) \in \{(2, 2), (3, 3), (4, 4)\}$
- Sample sizes $n \in \{50, 150, 400\}$
- Two error distributions: Gaussian and multivariate t (heavier tails)

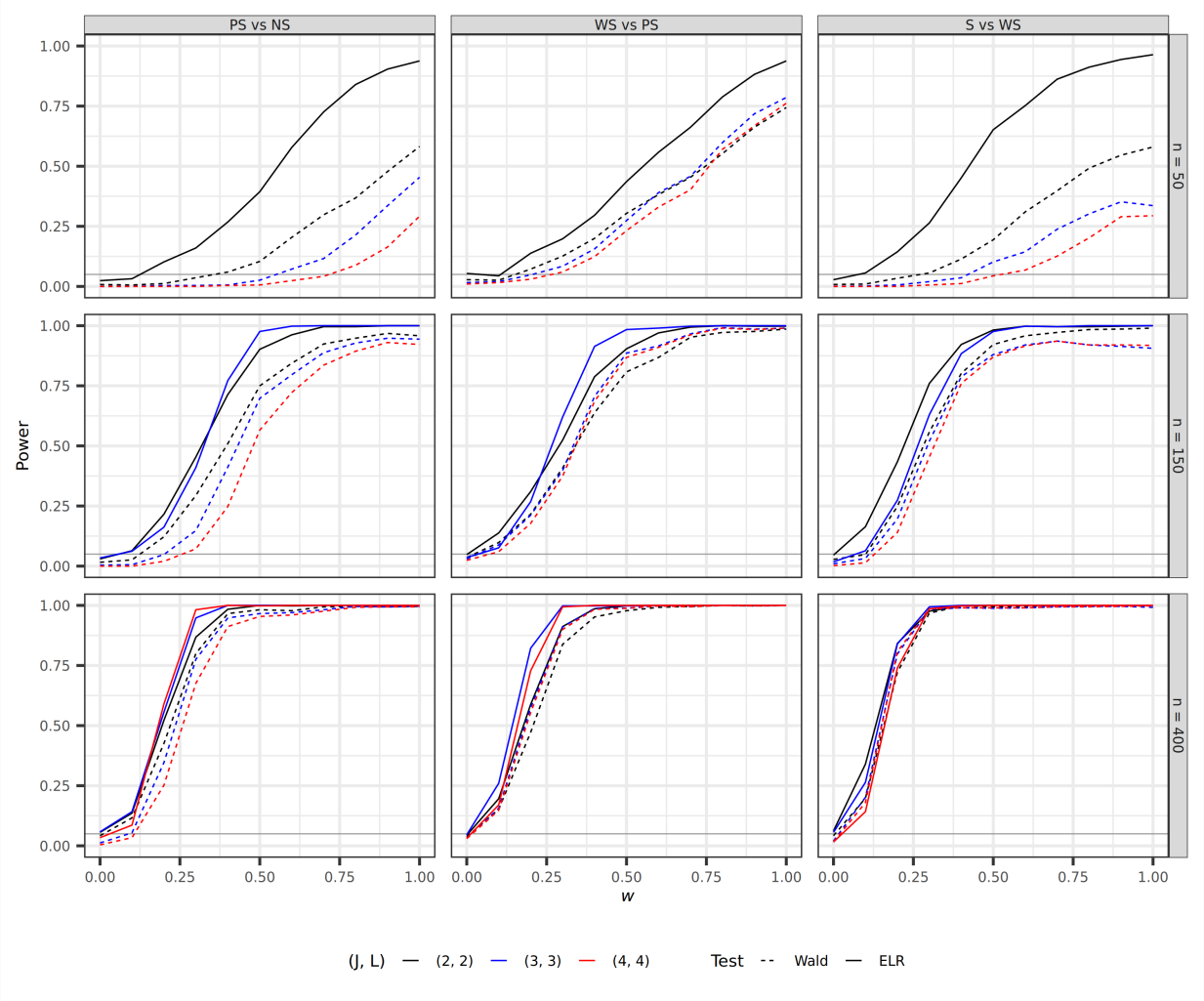
Three panels per figure: testing S vs. W, W vs. P, and P vs. non-separable.

Power Curves: Gaussian Data



ELR (solid) uniformly dominates Wald (dashed) across all three nested tests, truncation levels, and sample sizes; both control size near the nominal 0.05 *level (horizontal line).*

Power Curves: Heavier-Tailed (Multivariate t) Data



The ELR advantage persists, and widens, under departures from Gaussianity — particularly at small n, where Wald tests lose substantial power.

Discussion

- A principled workflow: test $S \to W \to P \to$ none, and use the selected level to simplify downstream functional graphical modeling

Discussion

- A principled workflow: test $S \to W \to P \to$ none, and use the selected level to simplify downstream functional graphical modeling
- ELR advantage: better size control and power, especially away from Gaussianity, without the standardization difficulties of Wald-type statistics

Discussion

- A principled workflow: test $S \to W \to P \to$ none, and use the selected level to simplify downstream functional graphical modeling
- ELR advantage: better size control and power, especially away from Gaussianity, without the standardization difficulties of Wald-type statistics
- Calibration relies on a Rademacher-symmetrized, operator-level bootstrap — details omitted here

Outline

From Classical Graphical Models to Functional GMs

Partial Separability and the Functional GGM

Testing for Levels of Separability

Beyond Truncation: FG-Lasso and a New Relaxation

Summary and Future Directions

Recall: Methods Without Dimension Reduction

- Truncation-based methods are fast but their bias is hard to control absent special structure like partial separability

Recall: Methods Without Dimension Reduction

- Truncation-based methods are fast but their bias is hard to control absent special structure like partial separability
- An alternative: work directly with the operators, no FPCA truncation at all

Recall: Methods Without Dimension Reduction

- Truncation-based methods are fast but their bias is hard to control absent special structure like partial separability
- An alternative: work directly with the operators, no FPCA truncation at all
- Li and Solea (2018) proposed using the correlation operator rather than the covariance operator to define a functional precision structure — but a fully operator-level, sparse, regularized estimator remained open until recently

The Functional Precision Operator

- Decompose $\boldsymbol{\mathcal{G}} = \mathcal{D}^{1/2}\boldsymbol{\mathcal{R}}\,\mathcal{D}^{1/2}$ with $\mathcal{D} = \mathrm{diag}(\boldsymbol{\mathcal{G}})$

The Functional Precision Operator

- Decompose $\boldsymbol{\mathcal{G}} = \mathcal{D}^{1/2}\boldsymbol{\mathcal{R}}\,\mathcal{D}^{1/2}$ with $\mathcal{D} = \mathrm{diag}(\boldsymbol{\mathcal{G}})$
- $\boldsymbol{\mathcal{R}}$ may be **invertible** even though $\boldsymbol{\mathcal{G}}$ is not: $\boldsymbol{\mathcal{R}}^{-1} = \mathcal{I} + \mathcal{H}$

The Functional Precision Operator

- Decompose $\boldsymbol{\mathcal{G}} = \mathcal{D}^{1/2}\boldsymbol{\mathcal{R}}\,\mathcal{D}^{1/2}$ with $\mathcal{D} = \mathrm{diag}(\boldsymbol{\mathcal{G}})$
- $\boldsymbol{\mathcal{R}}$ may be **invertible** even though $\boldsymbol{\mathcal{G}}$ is not: $\boldsymbol{\mathcal{R}}^{-1} = \mathcal{I} + \mathcal{H}$
- $\mathcal{H}$ is the functional precision operator; under Gaussianity, $\mathcal{H}_{jk} = 0$ corresponds to $X_j \perp\!\!\!\perp X_k \mid X_{-jk}$

The Functional Precision Operator

- Decompose $\boldsymbol{\mathcal{G}} = \mathcal{D}^{1/2} \boldsymbol{\mathcal{R}} \mathcal{D}^{1/2}$ with $\mathcal{D} = \mathrm{diag}(\boldsymbol{\mathcal{G}})$
- $\boldsymbol{\mathcal{R}}$ may be **invertible** even though $\boldsymbol{\mathcal{G}}$ is not: $\boldsymbol{\mathcal{R}}^{-1} = \mathcal{I} + \mathcal{H}$
- $\mathcal{H}$ is the functional precision operator; under Gaussianity, $\mathcal{H}_{jk} = 0$ corresponds to $X_j \perp\!\!\!\perp X_k \mid X_{-jk}$
- $\mathcal{H}$ may even be Hilbert–Schmidt – a legitimate target for sparse estimation

A True Functional Graphical Lasso

Waghmare, Masak, and Panaretos (2025) minimize, over self-adjoint $\mathcal{H}$ with $\mathcal{I} + \mathcal{H} \succ 0$,

$$\mathrm{tr}\{\mathcal{H}(\hat{\mathcal{R}} - \mathcal{I})\} - \log \det_2(\mathcal{I} + \mathcal{H}) + \gamma \sum_{j \neq k} \|\mathcal{H}_{jk}\|_{HS},$$

where $\det_2$ is the Carleman–Fredholm determinant (needed since $\mathcal{H}$ is infinite-dimensional).

A True Functional Graphical Lasso

Waghmare, Masak, and Panaretos (2025) minimize, over self-adjoint $\mathcal{H}$ with $\mathcal{I} + \mathcal{H} \succ 0$,

$$\mathrm{tr}\{\mathcal{H}(\hat{\mathcal{R}} - \mathcal{I})\} - \log\det{}_2(\mathcal{I} + \mathcal{H}) + \gamma \sum_{j \neq k} \|\mathcal{H}_{jk}\|_{HS},$$

where $\det_2$ is the Carleman–Fredholm determinant (needed since $\mathcal{H}$ is infinite-dimensional).

- Solved by ADMM on a fine discretization of every operator

A True Functional Graphical Lasso

Waghmare, Masak, and Panaretos (2025) minimize, over self-adjoint $\mathcal{H}$ with $\mathcal{I} + \mathcal{H} \succ 0$,

$$\mathrm{tr}\{\mathcal{H}(\hat{\mathcal{R}} - \mathcal{I})\} - \log\det{}_2(\mathcal{I} + \mathcal{H}) + \gamma \sum_{j \neq k} \|\mathcal{H}_{jk}\|_{HS},$$

where $\det_2$ is the Carleman–Fredholm determinant (needed since $\mathcal{H}$ is infinite-dimensional).

- Solved by ADMM on a fine discretization of every operator
- Bottleneck: each iteration needs a full eigendecomposition of the entire $m \times m$ discretized system ($m = \sum_j m_j$) — the $\log\det_2$ term couples *all* channels together

A True Functional Graphical Lasso

Waghmare, Masak, and Panaretos (2025) minimize, over self-adjoint $\mathcal{H}$ with $\mathcal{I} + \mathcal{H} \succ 0$,

$$\mathrm{tr}\{\mathcal{H}(\hat{\mathcal{R}} - \mathcal{I})\} - \log\det{}_2(\mathcal{I} + \mathcal{H}) + \gamma \sum_{j \neq k} \|\mathcal{H}_{jk}\|_{HS},$$

where $\det_2$ is the Carleman–Fredholm determinant (needed since $\mathcal{H}$ is infinite-dimensional).

- Solved by ADMM on a fine discretization of every operator
- Bottleneck: each iteration needs a full eigendecomposition of the entire $m \times m$ discretized system ($m = \sum_j m_j$) — the $\log\det_2$ term couples *all* channels together
- Cost $O(m^3)$ per iteration, with **no column-wise parallelism** — prohibitive once p (and hence m) grows, e.g. dense-array EEG

LOCO: ℓ_1 Operator-Constrained Optimization

Multiplying the FGL stationarity condition through by $(\mathcal{I} + \mathcal{H})$ removes the operator inverse from the constraint. This yields:

LOCO: ℓ_1 Operator-Constrained Optimization

Multiplying the FGL stationarity condition through by $(\mathcal{I} + \mathcal{H})$ removes the operator inverse from the constraint. This yields:

$$\hat{\mathcal{H}}_{\text{LOCO}} = \underset{\mathcal{H}}{\text{argmin}} \sum_{j,k=1}^{p} \|\mathcal{H}_{jk}\|_{HS} \quad \text{s.t.} \quad \left\| \mathcal{I} - \hat{\mathcal{R}}(\mathcal{I} + \mathcal{H}) \right\|_{\infty} \leq \gamma.$$

LOCO: ℓ_1 Operator-Constrained Optimization

Multiplying the FGL stationarity condition through by $(\mathcal{I} + \mathcal{H})$ removes the operator inverse from the constraint. This yields:

$$\hat{\mathcal{H}}_{\mathrm{LOCO}} = \underset{\mathcal{H}}{\operatorname{argmin}} \sum_{j,k=1}^{p} \|\mathcal{H}_{jk}\|_{HS} \quad \text{s.t.} \quad \left\| \mathcal{I} - \hat{\mathcal{R}}(\mathcal{I} + \mathcal{H}) \right\|_{\infty} \leq \gamma.$$

- Feasible set always nonempty: $\mathcal{H}^* = \hat{\mathcal{R}}^{-1} - \mathcal{I}$ is feasible for every $\gamma \geq 0$

LOCO: ℓ_1 Operator-Constrained Optimization

Multiplying the FGL stationarity condition through by $(\mathcal{I} + \mathcal{H})$ removes the operator inverse from the constraint. This yields:

$$\hat{\mathcal{H}}_{\mathrm{LOCO}} = \underset{\mathcal{H}}{\operatorname{argmin}} \sum_{j,k=1}^{p} \|\mathcal{H}_{jk}\|_{HS} \quad \text{s.t.} \quad \left\| \mathcal{I} - \hat{\mathcal{R}}(\mathcal{I} + \mathcal{H}) \right\|_{\infty} \leq \gamma.$$

- Feasible set always nonempty: $\mathcal{H}^* = \hat{\mathcal{R}}^{-1} - \mathcal{I}$ is feasible for every $\gamma \geq 0$
- The constraint decouples into p independent, parallelizable column problems — symmetrize post hoc

LOCO: ℓ_1 Operator-Constrained Optimization

Multiplying the FGL stationarity condition through by $(\mathcal{I} + \mathcal{H})$ removes the operator inverse from the constraint. This yields:

$$\hat{\mathcal{H}}_{\mathrm{LOCO}} = \operatorname*{argmin}_{\mathcal{H}} \sum_{j,k=1}^{p} \|\mathcal{H}_{jk}\|_{HS} \quad \text{s.t.} \quad \left\| \mathcal{I} - \hat{\mathcal{R}}(\mathcal{I} + \mathcal{H}) \right\|_{\infty} \leq \gamma.$$

- Feasible set always nonempty: $\mathcal{H}^* = \hat{\mathcal{R}}^{-1} - \mathcal{I}$ is feasible for every $\gamma \geq 0$
- The constraint decouples into p independent, parallelizable column problems — symmetrize post hoc
- Matches the functional graphical lasso's convergence rate under sub-Gaussian tails, and (unlike it) attains the same rate under polynomial tails, with no irrepresentability-type condition needed

Why the Optimization Is Still Hard

Even after decoupling by column, each column problem is a genuinely operator-level program:

- Each column's constraint is a norm constraint coupling every channel block of that column — effectively a second-order cone program with p nonlinear constraints

Why the Optimization Is Still Hard

Even after decoupling by column, each column problem is a genuinely operator-level program:

- Each column's constraint is a norm constraint coupling every channel block of that column — effectively a second-order cone program with p nonlinear constraints
- Off-the-shelf interior-point / conic solvers do not scale to EEG-level discretizations ($p \sim 64$ channels, $m \sim 2000$ discretized dimensions)

Why the Optimization Is Still Hard

Even after decoupling by column, each column problem is a genuinely operator-level program:

- Each column's constraint is a norm constraint coupling every channel block of that column — effectively a second-order cone program with p nonlinear constraints
- Off-the-shelf interior-point / conic solvers do not scale to EEG-level discretizations ($p \sim 64$ channels, $m \sim 2000$ discretized dimensions)
- Developed a custom first-order (primal–dual) solver with closed-form iterations

EEG Experiment

- UCI alcoholism EEG data: $p = 64$ scalp channels, each curve discretized to $m_j = 32$ points

EEG Experiment

- UCI alcoholism EEG data: $p = 64$ scalp channels, each curve discretized to $m_j = 32$ points
- Two groups analyzed *separately*: control ($n = 45$), alcoholic ($n = 77$)

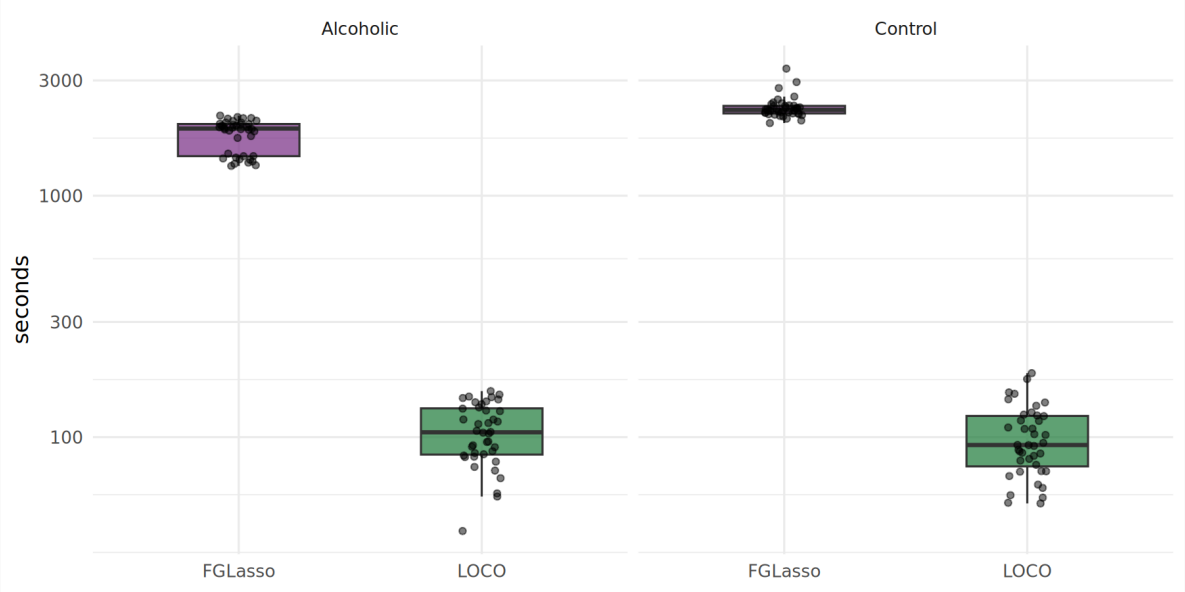
EEG Experiment

- UCI alcoholism EEG data: $p = 64$ scalp channels, each curve discretized to $m_j = 32$ points
- Two groups analyzed *separately*: control ($n = 45$), alcoholic ($n = 77$)
- $T = 40$ independent 60/40 train/test splits; both LOCO and the functional graphical lasso (FGLasso) fit along a full sparsity path (0–20% edge density)

EEG Experiment

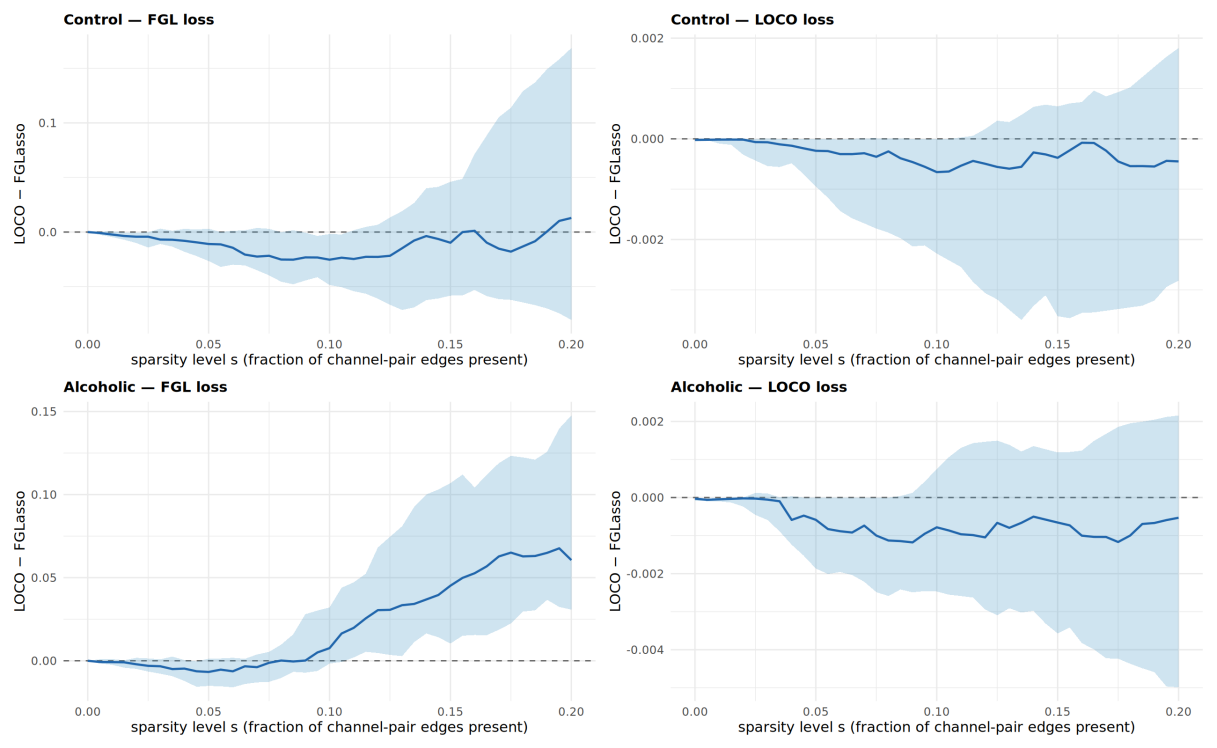
- UCI alcoholism EEG data: $p = 64$ scalp channels, each curve discretized to $m_j = 32$ points
- Two groups analyzed *separately*: control ($n = 45$), alcoholic ($n = 77$)
- $T = 40$ independent 60/40 train/test splits; both LOCO and the functional graphical lasso (FGLasso) fit along a full sparsity path (0–20% edge density)
- Out-of-sample evaluation under **both** methods' own loss functions — neither is favored

Computational Cost: LOCO vs. FGLasso



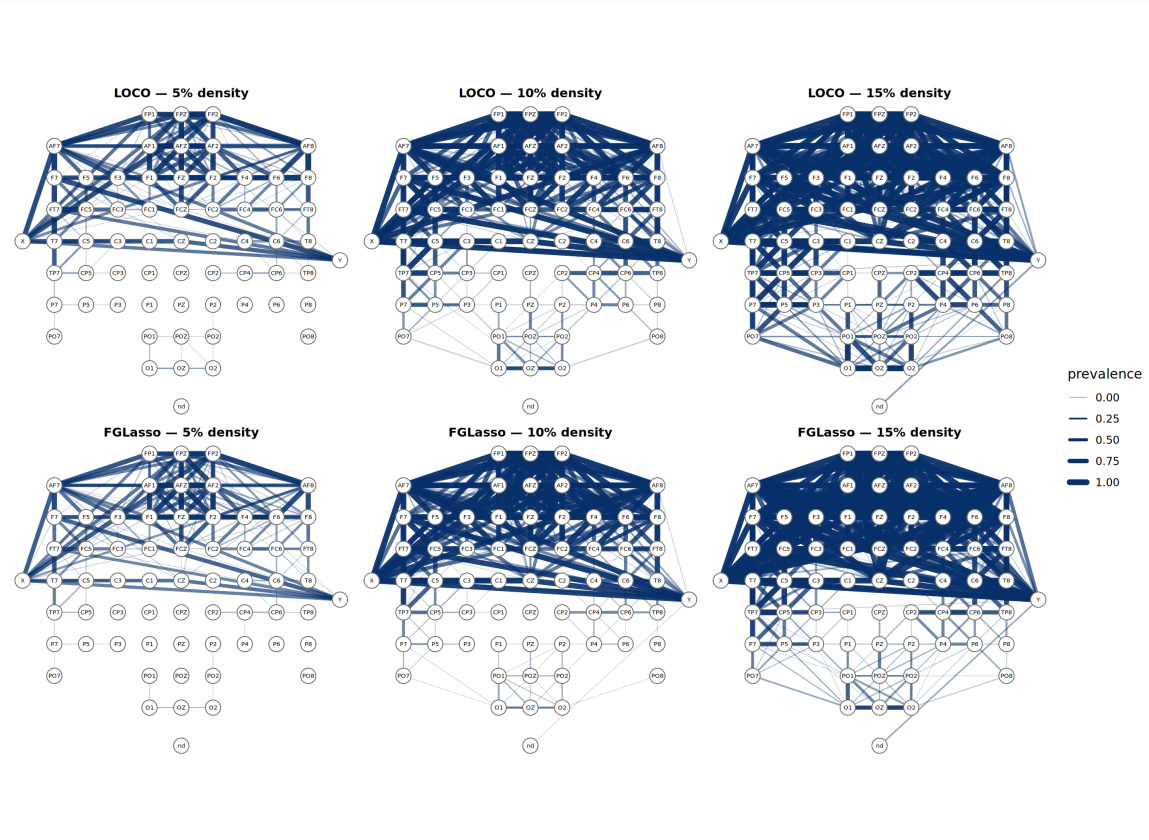
Per-split compute time (log scale) across the 40 splits, both groups: one warm-started LOCO γ-path vs. one FGLasso ADMM fit on the same grid. LOCO is roughly an order of magnitude faster.

Statistical Performance Is Comparable



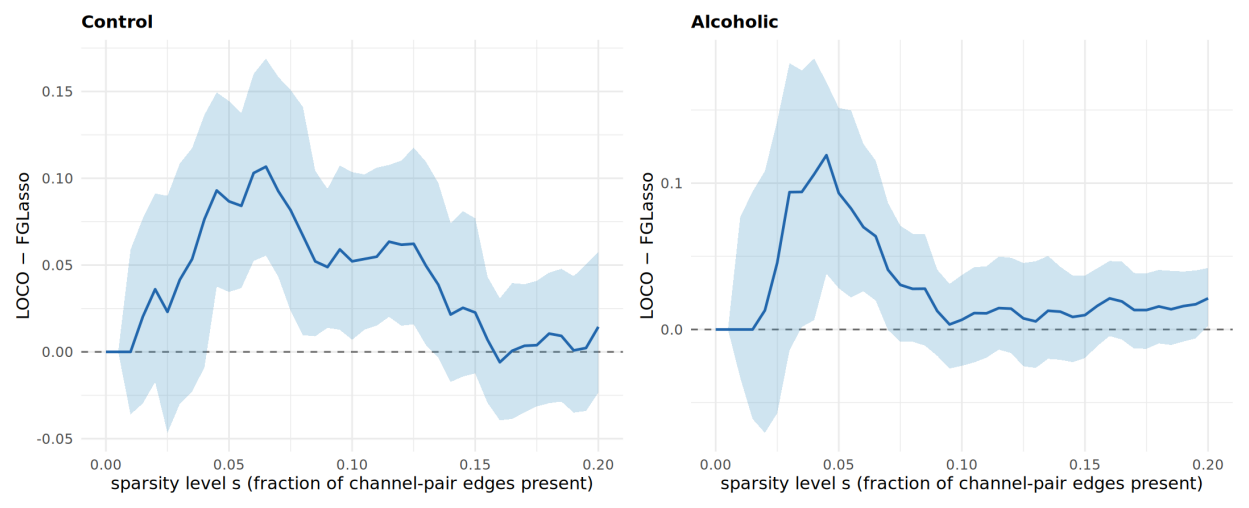
Paired per-split difference (LOCO − FGLasso) in test loss vs. sparsity, under each method's own loss (median line, IQR band). Differences are small relative to variability across splits.

Recovered Networks Agree



Edge prevalence across 40 splits at matched target densities (5/10/15%), alcoholic group: LOCO (top) vs. FGLasso (bottom). Line opacity/width = fraction of splits with that edge present — the two methods recover closely matching structure despite very different optimization criteria.

LOCO Graphs Are (Slightly) More Stable



Paired difference (LOCO − FGLasso) in split-to-split Jaccard overlap of estimated edge sets, vs. sparsity. Positive values favor LOCO; LOCO is modestly more stable across most of the sparsity range in both groups.

Summary: LOCO

- A Dantzig-selector-type relaxation of the functional graphical lasso, built as the direct operator analogue of CLIME

Summary: LOCO

- A Dantzig-selector-type relaxation of the functional graphical lasso, built as the direct operator analogue of CLIME
- Matches known convergence rates under sub-Gaussian tails, extends cleanly to heavier tails, avoids an irrepresentability-type condition

Summary: LOCO

- A Dantzig-selector-type relaxation of the functional graphical lasso, built as the direct operator analogue of CLIME
- Matches known convergence rates under sub-Gaussian tails, extends cleanly to heavier tails, avoids an irrepresentability-type condition
- An order-of-magnitude faster in practice, via a column-parallel first-order solver, with statistical performance and recovered network structure comparable to the full functional graphical lasso

Outline

From Classical Graphical Models to Functional GMs

Partial Separability and the Functional GGM

Testing for Levels of Separability

Beyond Truncation: FG-Lasso and a New Relaxation

Summary and Future Directions

Where This Leaves Us

- Functional graphical models extend conditional-independence graphs to entire component processes — but the key obstacle throughout is that functional covariance operators are compact and non-invertible

Where This Leaves Us

- Functional graphical models extend conditional-independence graphs to entire component processes — but the key obstacle throughout is that functional covariance operators are compact and non-invertible
- Truncation-based methods are simple but can carry bias that does not vanish with n; partial separability makes this bias precisely quantifiable and yields a clean sequence of finite-dimensional GGM problems

Where This Leaves Us

- Functional graphical models extend conditional-independence graphs to entire component processes — but the key obstacle throughout is that functional covariance operators are compact and non-invertible
- Truncation-based methods are simple but can carry bias that does not vanish with n; partial separability makes this bias precisely quantifiable and yields a clean sequence of finite-dimensional GGM problems
- Formal, self-normalizing (ELR) tests let us choose the right level of separability rather than assume it

Where This Leaves Us

- Functional graphical models extend conditional-independence graphs to entire component processes — but the key obstacle throughout is that functional covariance operators are compact and non-invertible
- Truncation-based methods are simple but can carry bias that does not vanish with n; partial separability makes this bias precisely quantifiable and yields a clean sequence of finite-dimensional GGM problems
- Formal, self-normalizing (ELR) tests let us choose the right level of separability rather than assume it
- Operator-level methods (LOCO) avoid truncation bias altogether, at a computational cost that recent relaxations make tractable at real neuroimaging scale

Open Directions

- Beyond time: extending FGMs to spatial and spatio-temporal functional domains (environmental, ecological, and other non-Euclidean data)

Open Directions

- Beyond time: extending FGMs to spatial and spatio-temporal functional domains (environmental, ecological, and other non-Euclidean data)
- Dynamic FGMs: allowing the dependence structure itself to evolve over time

Open Directions

- Beyond time: extending FGMs to spatial and spatio-temporal functional domains (environmental, ecological, and other non-Euclidean data)
- Dynamic FGMs: allowing the dependence structure itself to evolve over time
- Adaptive truncation: principled, data-driven choice of K/L that respects the theoretical requirement that truncation levels diverge with n

Open Directions

- Beyond time: extending FGMs to spatial and spatio-temporal functional domains (environmental, ecological, and other non-Euclidean data)
- Dynamic FGMs: allowing the dependence structure itself to evolve over time
- Adaptive truncation: principled, data-driven choice of K/L that respects the theoretical requirement that truncation levels diverge with n
- Non-Gaussian, operator-level estimation: combining LOCO-type computational strategies with copula or nonparametric formulations

See Zhao and Lee (2026, *WIREs Computational Statistics*) for a full discussion of these and other open problems.

Acknowledgments

- **Partial separability & HCP application**: Javier Zapata (OCC), Sang-Yun Oh (UC Santa Barbara)
- **Testing levels of separability**: Andrew Pope, Garritt Page (Brigham Young University)
- **LOCO**: Patric Platts (Brigham Young University)

Supported by the National Science Foundation, Awards DMS-2310943 and DMS-2128589.

Thanks also to the organizers of *Connectomics, Non-Euclidean Data, and Statistics* at IMSI.

Key References i

- Aston, J. A. D., Pigoli, D., and Tavakoli, S. (2017). Tests for separability in nonparametric covariance operators of random surfaces. *Annals of Statistics*, 45(4), 1431–1461.
- Cai, T., Liu, W., and Luo, X. (2011). A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *JASA*, 106(494), 594–607.
- Danaher, P., Wang, P., and Witten, D. M. (2014). The joint graphical lasso for inverse covariance estimation across multiple classes. *JRSS-B*, 76(2), 373–397.
- Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3), 432–441.
- Li, B., and Solea, E. (2018). A nonparametric graphical model for functional data with application to brain networks based on fMRI. *JASA*, 113(524), 1637–1655.
- Lundborg, A. R., Shah, R. D., and Peters, J. (2022). Conditional independence testing in Hilbert spaces with applications to functional data analysis. *JRSS-B*, 84(5), 1821–1850.

Key References ii

- Lynch, B., and Chen, K. (2018). A test of weak separability for multi-way functional data, with application to brain connectivity studies. *Biometrika*, 105(4), 815–831.
- Qiao, X., Guo, S., and James, G. M. (2019). Functional graphical models. *JASA*, 114(525), 211–222.
- Solea, E., and Dette, H. (2022). Nonparametric and high-dimensional functional graphical models. *Electronic Journal of Statistics*, 16(2), 6175–6231.
- Solea, E., and Li, B. (2022). Copula Gaussian graphical models for functional data. *JASA*, 117(538), 781–793.
- Waghmare, K. G., Masak, T., and Panaretos, V. M. (2025). The functional graphical lasso. *Annals of Statistics*, 53(5), 1857–1885.
- Zapata, J., Oh, S.-Y., and Petersen, A. (2022). Partial separability and functional graphical models for multivariate Gaussian processes. *Biometrika*, 109(3), 665–681.
- Zhao, B., Zhai, P. S., Wang, Y. S., and Kolar, M. (2024). High-dimensional functional graphical model structure learning via neighborhood selection approach. *Electronic Journal of Statistics*, 18(1), 1042–1129.

Key References iii

- Zhao, Y., and Lee, K.-Y. (2026). A comprehensive review of functional graphical models. *WIREs Computational Statistics*, 18(2), e70067.
- Zhu, H., Strawn, N., and Dunson, D. B. (2016). Bayesian graphical models for multivariate functional data. *JMLR*, 17(204), 1–48.

Thank you!

petersen@stat.byu.edu