Journal of Machine Learning Research 23 (2022) 1-91 Submitted 7/21; Revised 4/22; Published 6/22

A Nonconvex Framework for Structured Dynamic Covariance Recovery

Katherine Tsai KT14@ILLINOIS.EDU
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

Mladen Kolar MKOLAR@CHICAGOBOOTH.EDU
University of Chicago Booth School of Business
Chicago, IL 60637, USA

Oluwasanmi Koyejo SANMI@ILLINOIS.EDU
Department of Computer Science,
Beckman Institute for Advanced Science and Technology, and Statistics
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

Editor: Samuel Kaski

Abstract

We propose a flexible, yet interpretable model for high-dimensional data with time-varying second-order statistics, motivated and applied to functional neuroimaging data. Our approach implements the neuroscientific hypothesis of discrete cognitive processes by factorizing covariances into sparse spatial and smooth temporal components. Although this factorization results in parsimony and domain interpretability, the resulting estimation problem is nonconvex. We design a two-stage optimization scheme with a tailored spectral initialization, combined with iteratively refined alternating projected gradient descent. We prove a linear convergence rate up to a nontrivial statistical error for the proposed descent scheme and establish sample complexity guarantees for the estimator. Empirical results using simulated data and brain imaging data illustrate that our approach outperforms existing baselines.

Keywords: dynamic covariance, structured factor model, alternating projected gradient descent, time series data, functional connectivity

1. Introduction

We propose and evaluate a model for dynamic functional brain network connectivity, defined as the time-varying covariance of associations between brain regions (Fox and Raichle, 2007). Understanding the variation in brain connectivity between individuals is believed to be a crucial step towards uncovering the mechanisms of neural information processing (Sakoğlu et al., 2010; Chang et al., 2016), with potentially transformative applications in understanding and treating neurological and neuropsychiatric disorders (Calhoun et al., 2014).

In the neuroscience literature, estimators for time-varying covariances range from sliding window methods to hidden Markov models. The commonly used sliding-window sample covariance estimator is computationally efficient (Preti et al., 2017). However, this estimate is sensitive to the length of the selected window and spurious correlations may occur when the underlying window length is not specified correctly (Leonardi and Ville, 2015). Discrete-state hidden Markov models construct interpretable estimates of brain connectivity in terms of recurring connectivity patterns (Vidaurre et al., 2017), yet fail to capture the smooth nature of brain dynamics (Shine et al., 2016a,b). These shortcomings motivate a new approach. Specifically, our proposed approach implements the neuroscientific hypothesis that brain functions are interactions between cognitive processes (Posner et al., 1988), which we model as weighted combinations of low-rank components (Andersen et al., 2018). Beyond neuroscientific foundations, high-dimensional data often have a low-dimensional representation (Udell and Townsend, 2019), and low rank can help prevent overfitting (Udell et al., 2016). Specifically, we propose a structured and smooth low-rank time-varying covariance model inspired by the observed sparsity of brain factors (Eavani et al., 2012), and temporal dynamics of brain activity (Shine et al., 2016a,b). Hence, we constrain the temporal components to be smoothly varying via projection to a temporal kernel and restrict the sparsity of the spatial components via hard-thresholding, respectively.

We estimate the parameters of the resulting model using a first-order optimization scheme that is analogous to a Burer-Monteiro factorization (Burer and Monteiro, 2003, 2005). While the first-order approach reduces computational complexity as compared to semidefinite programming, the resulting optimization program is nonconvex, and special care is needed to design and analyze an optimization scheme that avoids converging to bad local optima. To this end, we build on the growing literature studying matrix estimation problems (Candès et al., 2015; Chi et al., 2019) using a two-stage algorithm. First, spectral initialization is used to find an initial point within a local region where the objective satisfies local regularity conditions. Next, the projected gradient descent is used to refine the estimate and find a stationary point of the objective.

In summary, our contributions include a novel dynamic covariance model motivated by neuroscientific models of functional brain connectivity networks. We provide an efficient procedure for the estimation along with convergence analysis and sample complexity. Specifically, under the assumption that spatial components are shared across time, we develop a structured spectral initialization method, which effectively uses available samples and provides a better spatial estimate than separate initialization per individual. We prove linear convergence of the factored gradient method to an estimate with a nontrivial statistical error and provide a non-asymptotic bound on the statistical error when data are Gaussian. Experiments show that the model successfully recovers temporal smoothness and detects temporal changes induced by task activation. The code to implement our procedure is available at: `https://github.com/koyejo-lab/dynamicCov.git`.

2. Background

In this section, we first introduce the notation used throughout the manuscript. Then, we introduce the problem formulation and related work.

2.1 Notation

The inner product of two matrices is denoted as $\langle X, Y\rangle = \mathrm{tr}(X^T Y)$. For a matrix X, $\sigma_k(X)$ denotes the kth largest singular value, $\|X\|_F^2 = \mathrm{tr}(X^T X)$ denotes the Frobenius norm, $\|X\|_2 = \sigma_1(X)$ denotes the spectral norm, and $\|X\|_\infty = \max_{i,j}|X_{i,j}|$ denotes the maximum norm. For two symmetric matrices X and Y, $X \preceq Y$ means that $Y - X$ is positive semidefinite. The pseudoinverse of X is denoted by $X^\dagger$. The set of $K \times K$ rotation matrices is denoted as $\mathcal{O}(K)$. Let $x \in \mathbb{R}^d$, $\|x\|_0 = \sum_{i=1}^d \mathbf{1}_{\{x_i \neq 0\}}$ denotes the ℓ_0 norm and $\|x\|_2 = (\sum_{i=1}^d x_i^2)^{1/2}$ denotes the ℓ_2 norm. We use $\kappa(\cdot,\cdot)$ to denote a positive definite kernel function. The function $\mathrm{diag} : \mathbb{R}^K \to \mathbb{R}^{K\times K}$ converts a K-dimensional vector to a $K \times K$ diagonal matrix. For scalars a and b, $a \vee b$ denotes $\max(a, b)$ and $a \wedge b$ denotes $\min(a, b)$. We use $a \gtrsim b$ ($a \lesssim b$) to denote that there exists a constant $C > 0$ such that $a \geq Cb$ ($a \leq Cb$). We use $a \asymp b$ to denote $a \gtrsim b$ and $a \lesssim b$. We use $[J]$ to denote the index set $\{1, \ldots, J\}$.

2.2 Problem Statement

Given samples from N subjects recorded at J time points, denoted $x_j^{(n)} \in \mathbb{R}^P$, $n \in [N]$, $j \in [J]$, let $S_{N,j} = N^{-1}\sum_{n=1}^N x_j^{(n)} x_j^{(n)T}$ be the sample covariance across subjects at time j. We assume that the population covariance takes a factorized form as

$$\mathbb{E}(S_{N,j}) = \Sigma_j^\star + E_j = V^\star \mathrm{diag}(a_j^\star) V^{\star T} + E_j, \quad j \in [J], \tag{1}$$

where $\mathbb{E}(\cdot)$ denotes the expectation, $\Sigma_j^\star$ has a rank that is at most K, E_j is a noise matrix such that the largest singular value of E_j is strictly smaller than the smallest nonzero singular value of $\Sigma_j^\star$. This structured assumption allows us to separate the components of interest $\Sigma_j^\star$ from the nuisance components E_j. This factorization employs spatial components $V^\star = (v_1^\star, \ldots, v_K^\star) \in \mathbb{R}^{P\times K}$ that are time invariant and column-wise orthonormal. The matrix V corresponds to the top-K eigenvectors of the set of J covariance matrices: $\{\mathbb{E}(S_{N,j})\}_{j\in[J]}$. Analogously, $A^\star = (a_1^\star, \ldots, a_J^\star) \in \mathbb{R}^{K\times J}$ represents the temporal components. To facilitate the estimation in a high-dimensional setting, we further assume that the columns of $V^\star$ are sparse and belong to $\mathcal{C}_V(s^\star) = \{v \in \mathbb{R}^P : \|v\|_0 \leq s^\star, \|v\|_2 = 1\}$. Let $G \in \mathbb{R}^{J\times J}$ be a positive semidefinite kernel matrix whose entries are $G_{x,y} = \kappa(x, y)$ for $x, y \in [J]$, where the kernel κ is known a priori. Denote $G^\dagger = Q\Lambda Q^T$ as the eigendecomposition of $G^\dagger$, the generalized inverse of G, with $Q \in \mathbb{R}^{J\times J}$ being a matrix with orthonormal columns and $\Lambda \in \mathbb{R}^{J\times J}$ being a diagonal matrix with nonnegative entries. The rows of $A^\star$, denoted $A^\star_{k\cdot}$, $k \in [K]$, are smooth, bounded, and belong to $\mathcal{C}_A(c^\star, \gamma^\star) = \{\alpha = Qu \in \mathbb{R}^J : 0 \le \alpha_j \le c^\star, u^T \Lambda u \le \gamma^\star\}$. The kernel κ is used to model the temporal smoothness of the rows of $A^\star$ and the box constraint ensures that $\alpha_j \ge 0$, so the covariance model is positive semidefinite and is upper bounded by a positive constant for $j \in [J]$.

The eigenvalues of the kernel matrix G may decay quickly, which can result in numerically unstable algorithms when projecting onto the set $\mathcal{C}_A$. For example, the eigenvalues of a kernel matrix corresponding to the Sobolev kernel decay at a polynomial rate, while for the Gaussian kernel they decay at an exponential polynomial rate (Schölkopf and Smola, 2001). Instead of working with the kernel matrix G, we construct a low-rank approximation, $\tilde{G}$, of G by truncating small eigenvalues. Write $Q = (\tilde{Q}, Q_1)$, where the columns of $\tilde{Q}$ are eigenvectors of G corresponding to eigenvalues greater than or equal to δ_A, and $\tilde{\Lambda}^{-1} = \operatorname{diag}(\Lambda_{jj}^{-1} \ge \delta_A \mid j \in [J])$. Then $\tilde{G}^\dagger = \tilde{Q}\tilde{\Lambda}\tilde{Q}^T$. We define $\tilde{\mathcal{C}}_A(c, \gamma) = \{\alpha = \tilde{Q}u : 0 \le \alpha_j \le c, u^T \tilde{\Lambda} u \le \gamma\}$ and the rank of $\tilde{G}$ is denoted as $r(\tilde{G})$.

Under the model (1), we estimate the parameters $Z^\star = (V^{\star T}, A^\star)^T$ by minimizing the following objective

$$\min_Z f_N(Z) = \min_{\substack{v_k \in \mathcal{C}_V(s),\, k\in[K] \\ A_{k\cdot} \in \tilde{\mathcal{C}}_A(c,\gamma),\, k\in[K]}} \frac{1}{J}\sum_{j=1}^J \frac{1}{2}\|S_{N,j} - V \operatorname{diag}(a_j) V^T\|_F^2, \tag{2}$$

where $A_{k\cdot}$ is the kth row of A. Although f_N is nonconvex with respect to $Z = (V^T, A)^T$, the corresponding covariance loss $\ell_{N,j}(\Sigma_j) = \frac{1}{2}\|S_{N,j} - \Sigma_j\|_F^2$ is m-strongly convex and L-smooth with $m = L = 1$ (Nesterov, 2013). We use alternating projected gradient descent to update V and A. The selection of tuning parameters of $\mathcal{C}_V$ and $\tilde{\mathcal{C}}_A$ is discussed in Section 3.2.

2.3 Related Work

Dynamic covariance models are common for analyzing time series data in applications ranging from computational finance and economics (Engle et al., 2019) to epidemiology (Fox and Dunson, 2015) and neuroscience (Foti and Fox, 2019). Dynamic covariance models can be fully nonparametric with kernel functions encoding the temporal dependencies (Wu and Pourahmadi, 2003; Chen and Leng, 2016). In practice, however, it is common to impose an additional structure on the covariance model. For example, one can assume that the inverse covariance matrix is sparse and furthermore, follows a particular temporal dynamic. Such an approach is called dynamic graphical modeling as nonzero entries in the inverse covariance matrix encode the structure of a Markov network when data are Gaussian. In the dynamic graphical model setting, one can either impose temporal smoothness using regularization approaches (Kolar et al., 2010b; Kolar and Xing, 2012; Monti et al., 2014; Hallac et al., 2017; Gibberd and Nelson, 2017; Zhu and Koyejo, 2018; Geng et al., 2019) or using kernel smoothing (Song et al., 2009a; Kolar and Xing, 2009; Zhou et al., 2010; Kolar et al., 2010a; Kolar and Xing, 2011; Wang and Kolar, 2014; Qiu et al., 2016; Lu et al., 2018; Geng et al., 2020). We emphasize that the methods for dynamic graphical models assume that the data are sampled independently at different time points but are generated by related distributions. In contrast, functional graphical models treat the data as multivariate random functions (Li and Solea, 2018; Qiao et al., 2019; Zhao et al., 2019; Qiao et al., 2020; Zapata et al., 2021; Zhao et al., 2022, 2021). Wang et al. (2020) focused on the estimation and inference of a graph that underlies the data from a point process. Another popular approach to modeling dynamic covariance models is via factor models. Factor models can encode the temporal structure using latent kernel regularization (Paciorek, 2003; Kastner et al., 2017). For example, Andersen et al. (2018) encoded smooth temporal dynamics by introducing a latent Gaussian process prior. Li (2019) also used piecewise Gaussian process factors to capture combinations of gradual and abrupt changes. Along similar lines, our approach implements temporal and spatial structure through projection onto suitable constraint sets.

Our work is also related to dictionary learning (Olshausen and Field, 1997; Mairal et al., 2010), which can be viewed as a type of factorization where the signal is decomposed into atoms and coefficients. In this factorization, the sparsity is controlled through a sparse penalty on the coefficients. Mishne and Charles (2019) extended this approach to encode temporal data by constructing time-trace atoms with spatial coefficients. In comparison, our model has shared spatial structure and individual temporal structure.

Autoregressive models have been applied to model dynamic connectivity in fMRI (Song et al., 2009b; Qiu et al., 2016; Liégeois et al., 2019). Although autoregressive models employ modeling assumptions different from ours, they can capture smooth temporal dynamics of signals. However, autoregressive model forecasts can become unreliable in high-dimensional settings (Bańbura et al., 2010). To this end, various implementations of structured transition matrices (Davis et al., 2016; Ahelegbey et al., 2016; Skripnikov and Michailidis, 2019) have been proposed and shown to improve computational efficiency and prediction accuracy.

The optimization problem in (2) is nonconvex and is optimized by alternating minimization. Recent literature has established a linear convergence rate to global optima (Jain et al., 2013; Hardt, 2014; Gu et al., 2016; Chen et al., 2021; Yu et al., 2020a, 2018; Na et al., 2021, 2020). In particular, our work builds on Bhojanapalli et al. (2016), who showed linear convergence in V when the underlying objective function is strongly convex with respect to $X = VV^T$. Subsequently, Park et al. (2018) and Yu et al. (2020c) proved a linear convergence rate for nonsymmetric matrices. Unlike previous work, our factorization scheme $V\text{diag}(a_j)V^T$ imposes additional structure on eigenvalues, which has potential applications in regularizing graph-structured models (Kumar et al., 2020).

In nonconvex optimization, finding a good initialization in a local region is often useful to avoid convergence to bad local optima (e.g., $Z = 0$ is a trivial stationary point in our model). One common approach is to use the minimizer of a convex relaxation of the original problem as a starting point (Yu et al., 2020c). For some problems, spectral methods also provide good initialization (Chen and Candès, 2015). We employ a problem-specific spectral approach to develop a novel initialization method. After initialization, a first-order gradient descent method is sufficient to ensure convergence to the desired optima (Candès et al., 2015). Combining with structured constraints, Chen and Wainwright (2015) provided a theoretical framework for the projected gradient descent method when the constraint sets are convex. In our work, the iterates are projected onto a nonconvex set, which might increase the distance $\|V - V^{\star}R\|_F^2$. Therefore, we need a problem-specific analysis to quantify the expansion coefficient.

3. Methodology

We first introduce the proposed two-stage algorithm. Subsequently, we discuss the selection of the tuning parameters.

3.1 Two-stage Algorithm

We develop a two-stage algorithm to solve the optimization problem in (2). As the objective is nonconvex, a local iterative procedure may converge to bad local optima or saddle points. In the first stage of the algorithm, spectral decomposition is used to find an initialization point. In the second stage, projected gradient descent is used to locally refine the initial estimate and find a stationary point that is within the statistical error of the population parameters. Algorithm 1 summarizes our initialization procedure. Here, the eigendecomposition of $\{S_{N,j}\}_{j\in[J]}$ is performed to obtain initial estimates of $V^{\star}$ and $A^{\star}$. Specifically, the initialization uses the shared spatial structure of $\{\Sigma_j^{\star}\}_{j\in[J]}$ to increase the effective sample size. That is, the initial estimate V^0 is obtained from the eigenvectors corresponding to the largest K eigenvalues of the covariance matrix pooled over time, $M_N = J^{-1}\sum_{j=1}^{J} S_{N,j}$. The initial estimate of the temporal coefficients, A^0, is obtained by projecting $\{S_{N,j}\}_{j\in[J]}$ onto V^0.

```
Set $M_N = (NJ)^{-1}\sum_{j=1}^{J}\sum_{n=1}^{N} x_j^{(n)} x_j^{(n)T}$
Set $V^0 = (v_1^0, v_2^0, \ldots, v_k^0) \leftarrow$ top $K$ eigenvectors of $M_N$
For $j = 1$ to $j = J$ and $k = 1$ to $k = K$
    $a_{k,j}^0 \leftarrow v_k^{0T} S_{N,j} v_k^0$
Set $A^0 = (a_{k,j}^0)_{k\in[K],j\in[J]}$
Output $V^0$, $A^0$
```

Algorithm 1: Spectral initialization

After initialization, we iteratively refine the estimates of V and A using an alternating projected gradient descent. In each iteration, the iterates V and A are updated using the gradient of f_N, where η denotes the step size. Note that we scale down the step size for the V update by J to balance the magnitude of the gradient. After a gradient update, we project the iterates onto the constraint sets $\mathcal{C}_V$ and $\tilde{\mathcal{C}}_A$ to enforce sparsity in V and smoothness in A. Details are given in Algorithm 2.

```
Set $V^0, A^0 =$ Spectral initialization$(\{x_j^{(n)}\}_{n\in[N],j\in[J]})$
While $|f_N(Z^{i-1}) - f_N(Z^{i-2})| > \varepsilon$
  $\widehat{A}^i \leftarrow A^{i-1} - \eta\nabla_A f_N(Z^{i-1})$
  $A^i \leftarrow$ Project rows of $\widehat{A}^i$ to $\tilde{\mathcal{C}}_A$
  $\widehat{V}^i \leftarrow V^{i-1} - \frac{\eta}{J}\nabla_V f_N(Z^{i-1})$
  $V^i \leftarrow$ Project columns of $\widehat{V}^i$ to $\mathcal{C}_V$
Output $V$, $A$
```

Algorithm 2: Dynamic covariance estimation

Although $\mathcal{C}_V$ is a nonconvex set, projection onto this set can be computed efficiently by picking the top-s largest entries in magnitude and then projecting the constructed vector to the unit sphere. Despite projecting onto a nonconvex set, we are able to show that the gradient and projection step jointly result in a contraction (see Appendix A). On the other hand, the projection onto the convex set $\tilde{\mathcal{C}}_A$ can be computed efficiently via convex programming: we project onto $\tilde{\mathcal{C}}_A$ by iteratively projecting onto $\{\alpha \in \mathbb{R}^J : 0 \leq \alpha_j \leq c, j \in [J]\}$ and $\{\alpha = \tilde{Q}u : u^T\tilde{\Lambda}u \leq \gamma\}$, which gives us a point in the intersection of the sets by von Neumann's theorem (Escalante and Raydan, 2011).

3.2 Selection of Tuning Parameters

The parameters of the proposed model include the sparsity level s, the rank K, the kernel length scale l, the smoothness coefficient γ, the truncation level δ_A, and the upper bound c for the constraint. For some kernels, such as the Gaussian kernel, the Matérn five-half kernel, and other radial basis function kernels, one must also select the length scale parameter l, which captures the smoothness of the curves (i.e., $\{A^\star_{k\cdot}\}_{k\in[K]}$); for example, a Gaussian kernel function is $\kappa_l(x,y) = \sigma^2\exp\{-(x-y)^2/(2l^2)\}$, where l affects the slope of decay of the eigenvalues. We denote such kernel functions as κ_l rather than κ. Our theory suggests that δ_A should be upper bound by the magnitude of $\min_{j\in[J]}\sigma_K^2(\Sigma_j^\star)$ to obtain a good statistical error. Furthermore, δ_A is selected for numerical stability. In experiments, we find that $\delta_A = 10^{-5}$ is a good empirical choice and satisfies the sufficient conditions. In principle, we do not want to cut off any important signals, so we choose c as a value greater than $\max_{j\in[J]}\|S_{N,j}\|_2$ and $c^\star = \max_{j\in[J]}\|\Sigma_j^\star\|_2$. In terms of estimation performance, we observe that the selection of s and K has a greater effect than the selection of γ and l. Although the underselection of s and K leads to poor evaluation scores, the improper selection of l and γ has a relatively minor influence. Therefore, we adopt a two-stage approach to selecting parameters. In the first stage, we perform a grid search on s, K, γ, l and find the configuration that minimizes the Bayesian information criterion $\textsc{bic} = \log N \sum_{k=1}^{K} \|v_k\|_0 - 2\widehat{L}_N$, where $\widehat{L}_N$ is the maximized Gaussian log-likelihood function. We notice that varying γ and l have a subtle influence on BIC. Consequently, in the second stage, we fix s, K with values selected in the first stage and select γ and l using a 5-fold cross-validation with the Gaussian log-likelihood, which is motivated by prior work on nonparametric dynamic covariances (Yin et al., 2010). Empirically, we find that tuning the length scale parameter l is more effective than tuning γ in producing globally smooth temporal structures (see Appendix G.5).

4. Theory

We provide theoretical guarantees on the algorithm described in the previous section. We show that given enough samples, that is, $N \gtrsim KP(\log P + \log J)$, the estimate converges linearly to a statistically good point with high probability. A statistically good point is one that is close to the population parameters as quantified by the statistical error.

4.1 Preliminaries

Before presenting our main theoretical results, we introduce two tools that will help us establish the results.

First, we discuss orthogonalization. The spatial component V produced by Algorithm 2 is not necessarily orthonormal. However, $V^\star$ has full rank, and if $\min_{Y \in \mathcal{O}(K)} \|V - V^\star Y\|_2^2 < 1$ is guaranteed at each iteration, then V also has full rank. As a result, the subspace spanned by columns of V is equal to the subspace spanned by columns of the orthogonalized version of it. To simplify the analysis of Algorithm 2, we add a QR decomposition step that orthogonalizes V after projection onto $\mathcal{C}_V$. That is, in each iteration, we compute

$$V^i_{ortho} \leftarrow V^i (L^i)^{-1} \quad \text{(QR decomposition)},$$

where L^i is the upper triangular matrix, with diagonal entries less than or equal to 1. Note that orthogonalization of V in each iteration of Algorithm 2 is not needed in practice and is only used to establish theoretical properties. This approach is commonly used in the literature (Jain et al., 2013; Zhao et al., 2015). We further note that the addition of QR decomposition only increases the distance of the iterate V^i to $V^\star R$ by a mild constant (Stewart, 1977; Zhao et al., 2015) (see Appendix A.3). Furthermore, QR decomposition increases the number of nonzero elements of the iterate V to at most Ks. As we consider the rank K to be fixed and $P \gtrsim s$, the effect of QR decomposition is mild. Our experiments further demonstrate that optimization with and without the QR decomposition step results in comparable performance.

Next, we introduce the notion of statistical error, which allows us to quantify the distance of the population parameters from the stationary point to which the optimization algorithm converges. Note that the notion of statistical error was used in the context of M-estimation (Loh and Wainwright, 2015). Let $\mathcal{B}_t = \{v \in \mathbb{R}^P \mid \|v\|_0 \le t, \|v\|_2 \le 1\}$ and

$$\Upsilon(r,t,h,\delta_A) = \{\{\Delta_j = V\mathrm{diag}(a_j)W^T\}_{j\in J} \mid v_k \in \mathcal{B}_t, w_k \in \mathcal{B}_t, A_{k\cdot}^T\tilde{G}^{\dagger}A_{k\cdot} \le h, k \in [r]\},$$

where $\tilde{G}$ is the truncation of G at the level of δ_A. We define the statistical error as

$$\varepsilon_{stat} = \varepsilon_{stat}(2K, 2s+s^\star, 2\gamma, \delta_A) = \max_{\{\Delta_j\}_{j\in[J]}\in\Upsilon(2K,2s+s^\star,2\gamma,\delta_A)} \frac{\sum_{j=1}^{J}\langle \nabla \ell_{N,j}(\Sigma_j^\star), \Delta_j\rangle}{\left(\sum_{j=1}^{J}\|\Delta_j\|_F^2\right)^{1/2}}.$$

The statistical error describes the geometric landscape around the optimum—it quantifies the magnitude of gradient of the empirical loss function evaluated at the population parameter in the directions constrained to the set Υ.

4.2 Assumptions and Main Results

We begin by stating the assumptions needed to establish the main results. Note that V in this section is used to denote an iterate in after the QR factorization step.

An upper bound on the step size is required for convergence of Algorithm 2. Let $Z_j^0 = (V^{0T}, \mathrm{diag}(a_j^0))^T$, $j \in [J]$, denote the output of Algorithm 1.

Assumption 1 *The step size satisfies* $\eta \le \min_{j\in[J]}\ J^{1/2}/(64\|Z_j^0\|_2^2)$.

Note that the step size depends on the initial estimate, but remains constant throughout the iterations. Let $\beta = 1 - \eta/(4J\xi^2) < 1$, $\chi = 4\beta^{1/2}(1-2I_0/\sqrt{J})^{-2}(1+32\|A^\star\|_\infty^2)$, and $\tau = J^{-1}\{9/2 + (1/2 \vee K/8)\}$, where

$$I_0^2 = \left\{\frac{1}{16\xi^2}\frac{1}{(1+\|A^\star\|_\infty^2 J^{-1})} \wedge \frac{J}{4}\right\}, \quad \xi^2 = \max_{j\in[J]}\left\{\frac{16}{\sigma_K^2(\Sigma_j^\star)} + \left(1 + \frac{8c}{\sigma_K(\Sigma_j^\star)}\right)^2\right\}. \tag{3}$$

We also require that the tuning parameters be selected appropriately.

Assumption 2 *We have* $c \ge c^\star$, $\gamma \ge \gamma^\star$, $s \ge [\{4(1/\chi - 1)^{-2} + 1\} \vee 2]s^\star$. *The matrix* $\tilde{G}$ *is obtained with the truncation level* $\delta_A \le (16\gamma^\star)^{-1}\min_{j\in[J]}\sigma_K^2(\Sigma_j^\star)$.

Note that the condition on δ_A is mild. It guarantees that we do not truncate too much of the signal. Finally, we require an assumption on the statistical error.

Assumption 3 *We have* $\varepsilon_{stat}^2 \le JI_0^2\{(\beta^{1/2}-\beta)/(\tau\eta) \wedge \min_{j\in[J]} 3\|Z_j^\star\|_2^2\}$.

Assumption 3 is essentially a requirement on the sample size N, since for a sufficiently large N the assumption will be satisfied with high probability. Note that as the sample size increases, the statistical error becomes smaller, while the radius of the local region of convergence, I_0, remains constant. Furthermore, if Assumption 3 is not satisfied, this implies that the initialization point is already close enough to the population parameters and that the subsequent refinement by Algorithm 2 is not needed.

With these assumptions, we are ready to state the main result, which tells us how far the estimate obtained by Algorithms 1 and 2 is from the population parameter. Let $\Sigma_j^I = V^I \mathrm{diag}(a_j^I)(V^I)^T$, $j \in [J]$, denote the estimate of the population covariance after the Ith iteration.

Theorem 4 *Suppose Assumptions 1—3 are satisfied and $J \ge 4$. Furthermore, for a sufficiently large constant C_0, suppose that there are $N = C_0 KP\log(PJ/\delta_0)$ independent samples such that $\|x_j^{(n)}\|_2^2 \le P\|A^\star\|_\infty$ almost surely, $j \in [j]$, with zero mean and covariance as in* (1)*. Then, with probability at least $1-\delta_0$, the estimate obtained by Algorithm 1 and Algorithm 2 satisfies*

$$\sum_{j=1}^{J} \|\Sigma_j^I - \Sigma_j^\star\|_F^2 \le \beta^{I/2}(4\mu^2\xi^2)\sum_{j=1}^{J} \|\Sigma_j^0 - \Sigma_j^\star\|_F^2 + \frac{2\tau\mu^2\eta}{\beta^{1/2}-\beta}\varepsilon_{stat}^2 + 2K\gamma^\star\delta_A, \qquad (4)$$

where $\mu = \max_{j\in[J]}(17/8)\|Z_j^\star\|_2$.

The first term on the right-hand side of (4) corresponds to the optimization error, and we observe a linear convergence rate. The second and third terms of (4) correspond to the statistical and approximation errors due to the truncation of the kernel matrix, respectively. From the bound we observe a trade-off between ε_{stat} and the truncation error δ_A: if δ_A decreases, ε_{stat} increases.

The proof of Theorem 4 is given in two steps. First, we establish the convergence rate of the iterates obtained by Algorithm 2 when the initial points V^0 and A^0 lie in a neighborhood around $V^\star$ and $A^\star$ (see §4.3). Subsequently, we show in Theorem 7 that Algorithm 1 provides suitable V^0 and A^0 with high probability (see §4.5).

To give an example of Theorem 4, we consider the case where data are generated from a multivariate Gaussian distribution and for a Gaussian kernel.

Proposition 5 *Let $x_j^{(n)} \in \mathbb{R}^P$ be independent Gaussian samples with mean zero and covariance as in* (1) *with $J \ge 4$ and $N \gtrsim K(P + \log J/\delta_0)$. Suppose that G is a Gaussian kernel matrix whose eigenvalue decays at the rate $\exp(-l^2j^2)$ for some length scale $l > 0$. Let $\delta_A \asymp (\gamma^\star lN)^{-1}\{\log(\gamma^\star lN)\}^{1/2}$. Suppose that Assumptions 1—2 hold, $s^\star\log(P/s^\star) < PJ$ and $\max_{j\in[J]}\|E_j\|_2 \lesssim I_0$. Then after $I \gtrsim \log(1/\delta_1)$ iterations of Algorithm 2, with probability at least $1-\delta_0$, we have*

$$\sum_{j=1}^{J} \|\Sigma_j^I - \Sigma_j^\star\|_F^2 \lesssim \delta_1 + \frac{1}{N}\left[K\left\{\frac{1}{l}\left(\log\gamma^\star lN\right)^{1/2} + s^\star\log\frac{P}{s^\star}\right\} + \log\delta_0^{-1}\right] + J\max_{j\in[J]}\|E_j\|_2^2.$$

Condition $s^\star \log(P/s^\star) < PJ$ is mild, since $s^\star \lesssim P$, and condition $\max_{j\in[J]} \|E_j\|_2 \lesssim I_0$ is mild, since $\|E_j\|_2 < \sigma_K(\Sigma_j^\star)$, $j \in [J]$. Under the Gaussian distribution, the sample complexity is improved to $N \gtrsim K(P + \log J)$ from $N \gtrsim KP(\log P + \log J)$ in Theorem 4. Proposition 5 provides an explicit bound on the estimator that can be obtained under an assumption on the eigenvalue decay. The statistical error is comprised of two terms that correspond to errors when estimating smooth temporal components and sparse spatial components. In our choice of δ_A, the truncation error is in the same order as the statistical error induced by the smooth temporal components.

4.3 Linear Convergence

We establish the linear convergence rate of Algorithm 2 when it is appropriately initialized. Recall that the rows of $A^\star$ belong to $\mathcal{C}_A(c^\star, \gamma^\star) \subseteq \mathcal{C}_A(c, \gamma)$, while the projected gradient descent is implemented on the set $\tilde{\mathcal{C}}_A(c, \gamma) \subset \mathcal{C}_A(c, \gamma)$. Let

$$\tilde{A}^\star = \underset{B_{k\cdot} \in \tilde{\mathcal{C}}_A(c,\gamma), k\in[K]}{\operatorname{argmin}} \|B - A^\star\|_F^2,$$

be the best approximation of $A^\star$ in $\tilde{\mathcal{C}}_A(c, \gamma)$. See Appendix E for details on the construction of $\tilde{A}^\star$. We define $\tilde{\Sigma}_j^\star = V^\star \mathrm{diag}(\tilde{a}_j^\star) V^{\star T}$, $j \in [J]$, and $\tilde{Z}^{\star T} = (V^{\star T}, \tilde{A}^\star)$. With these definitions, we establish the linear rate of convergence of the iterates to $\tilde{\Sigma}_j^\star$ and $\tilde{Z}^\star$. The convergence rate in Theorem 4 will then follow by combining the results with the truncation error.

Observe that the covariance factorization is not unique since, for any $R \in \mathcal{O}(K)$, we have $\Sigma_j = V \mathrm{diag}(a_j) V^T = V R_j {R_j}^T \mathrm{diag}(a_j) R_j {R_j}^T V^T$, $j \in [J]$. By the triangle inequality, we have

$$\sum_{j=1}^J \|\Sigma_j - \tilde{\Sigma}_j^\star\|_F^2 \leq \sum_{j=1}^J \alpha_{V,j} \|V - V^\star R\|_F^2 + \alpha_A \|\mathrm{diag}(a_j) - R^T \mathrm{diag}(\tilde{a}_j^\star) R\|_F^2, \qquad (5)$$

where $\alpha_{V,j} = 3\{\|V \mathrm{diag}(a_j)\|_2^2 + \|V^\star \mathrm{diag}(\tilde{a}_j^\star)\|_2^2\}$, $j \in [J]$, and $\alpha_A = 3\|V^\star\|_2^2 \|V\|_2^2$. This implies that if $\|V - V^\star R\|_F^2 + \|\mathrm{diag}(a_j) - R^T \mathrm{diag}(\tilde{a}_j^\star) R\|_F^2$ is small for some rotation matrix R and every $j \in [J]$, then the left-hand side will also be small. To this end, our goal is to show that the following distance metric contracts in each iteration of Algorithm 2. Let

$$R = \underset{Y \in \mathcal{O}(K)}{\operatorname{argmin}} \|V - V^\star Y\|_F^2, \quad \mathrm{dist}^2(Z, \tilde{Z}^\star) = \sum_{j=1}^J d^2(Z_j, \tilde{Z}_j^\star); \qquad (6)$$

$$d^2(Z_j, \tilde{Z}_j^\star) = \|V - V^\star R\|_F^2 + \|\mathrm{diag}(a_j) - R^T \mathrm{diag}(\tilde{a}_j^\star) R\|_F^2,$$

where $Z_j^T = (V^T, \mathrm{diag}(a_j))$ and $\tilde{Z}_j^{\star T} = (V^{\star T}, \mathrm{diag}(\tilde{a}_j^\star))$. The metric first finds the rotation matrix that aligns two subspaces and then computes the transformation of $\mathrm{diag}(\tilde{a}_j^\star)$ along the rotation R. This metric is similar to the distance metric commonly used in matrix factorization problems (Anderson and Rubin, 1956; ten Berge, 1977), but in our model the choice of R depends only on V.

To show the convergence of $\mathrm{dist}^2(Z, \tilde{Z}^\star)$, we need the following assumptions.

Assumption 6 *Suppose that Z_j^0 satisfies $d^2(Z_j^0, Z_j^\star) \leq I_0^2$, for $j \in [J]$, where I_0 is defined in* (3)*. Assume that $\|V^0 - V^\star R\|_F^2 \leq I_0^2/J$ and $\|\mathrm{diag}(a_j^0) - R^T \mathrm{diag}(a_j^\star) R\|_F^2 \leq (J-1)I_0^2/J$.*

Since $d^2(Z_j^0, \tilde{Z}_j^\star) \leq d^2(Z_j^0, Z_j^\star)$ for $j \in [J]$, Assumption 6 ensures that the distance between initial estimates and the population parameters is bounded within the ball of radius I_0. Furthermore, $I_0^2 \leq J$ ensures that $\|V - V^\star R\|_2 \leq 1$, so that V is full-rank. Intuitively, we assume that the squared distance for V is $1/(J-1)$ times smaller than the squared distance for A, because we have J times more samples to estimate V compared to A.

Theorem 7 *Assume that Assumptions 1—3, and Assumption 6 hold. After I iterations of Algorithm 2, we have*

$$dist^2(Z^I, \tilde{Z}^\star) \leq \beta^{I/2} dist^2(Z^0, \tilde{Z}^\star) + \frac{\tau\eta\varepsilon_{stat}^2}{\beta^{1/2} - \beta}.$$

Theorem 7 establishes a linear rate of convergence in $\mathrm{dist}^2(Z, \tilde{Z}^\star)$. The second term on the left-hand side denotes the constant multiple of the statistical error, which depends on the distribution of the data and the sample size. Combining with (5) gives us a linear rate of convergence in $\sum_{j=1}^J \|\Sigma_j - \tilde{\Sigma}_j^\star\|_F^2$.

4.4 Statistical Error

Theorem 7 shows the linear convergence of the algorithm to a region around the population parameters characterized by statistical error. One may wonder how large the statistical error can be. While Assumption 3 provides a condition under which convergence is guaranteed, this bound is loose, as it does not depend on the sample size. We establish a tighter bound under the Gaussian distribution.

Proposition 8 (Statistical Error for Gaussian Data) *Let $x_j^{(n)} \in \mathbb{R}^P$ be independent Gaussian samples with mean zero and covariance as in* (1)*. Then, with probability at least $1-\delta$,*

$$\varepsilon_{stat}(2K, (2m+1)s^*, 2m'\gamma^\star, \delta_A) \leq (\nu \vee \nu^2) + \sqrt{J} \max_{j\in[J]} \|E_j\|_2,$$

where

$$\nu = \frac{\|A^\star\|_\infty}{e_0} \left[\frac{1}{N} \left\{ \log\frac{1}{\delta} + Kr(\tilde{G}) + Ks^\star \log\frac{P}{s^\star} \right\} \right]^{\frac{1}{2}},$$

m, m' are positive integers, e_0 is an absolute constant depending on m and m', and $r(\tilde{G})$ is the rank of the δ_A-truncated kernel matrix $\tilde{G}$.

We interpret ε_{stat} as follows. The first term corresponds to the error in estimating the low-rank matrix, while the second term corresponds to the essential error incurred from approximating the covariance matrix by a low-rank matrix. The low-rank matrix can be estimated with the rate that converges to zero as $[K\{r(\tilde{G}) + s^\star \log P\}/N]^{-1/2}$, which corresponds to the rate of convergence of temporal and spatial components. We also highlight that truncation of G simplifies the statistical analysis because we can view the projection to $\tilde{\mathcal{C}}_A$ as restricting rows of A to a subset of a $r(\tilde{G})$-dimensional smooth subspace with $r(\tilde{G})$ much smaller than J, the original dimension.

4.5 Sample Complexity of Spectral Initialization

We discuss the sample complexity required to satisfy Assumption 6. That is, we characterize the sample size needed for Algorithm 1 to give a good initial estimate, so that Algorithm 2 outputs a solution characterized in Theorem 7. We consider a general case of a bounded distribution.

Theorem 9 (Sample Complexity of Spectral Initialization) *Let $x_j^{(n)} \in \mathbb{R}^P$ be independent zero mean samples with $\|x_j^{(n)}\|_2^2 \leq P\|A^\star\|_\infty$ almost surely, $n \in [N]$, $j \in [J]$, $J \geq 4$. Let $M^\star = J^{-1}\sum_{j=1}^J \mathbb{E}(S_{N,j})$ and $g = \sigma_K(M^\star) - \sigma_{K+1}(M^\star) > 0$ be the eigengap. Then, with probability at least $1-\delta$,*

$$dist^2(Z^0, Z^\star) \leq \phi(g, A^\star)\left\{\frac{KJP^2}{N^2}\left(\log\frac{4JP}{\delta}\right)^2 + \frac{KJP}{N}\log\frac{4JP}{\delta}\right\}; \tag{7}$$

$$\phi(g, A^\star) = 4\|A^\star\|_\infty^2\left\{\frac{5(1+16\varphi^2\|A^\star\|_\infty^2)}{g^2 J} \vee 8\varphi^2\right\},$$

where $\varphi^2 = \max_{j\in[J]}\{1 + 4\sqrt{2}\|A^\star\|_\infty/\sigma_K(\Sigma_j^\star)\}$.

From (7) we note that if $N \gtrsim P\log(PJ/\delta)$, then Assumption 6 will be satisfied with high probability. The eigengap g must be greater than 0 for the bound in (7) to be nontrivial. Moreover, since $g \leq \|A^\star\|_\infty$, the first term of $\phi(g, A^\star)$ dominates when J is small. Combining results from (5), Theorem 7, and Theorem 9, we can establish Theorem 4.

5. Simulations

We evaluate the algorithm described in Section 3 using the metric in (6) and the average log-Euclidean metric (Arsigny et al., 2006) over a variety of temporal dynamics. Table 1 collects competing methods.

We generate synthetic samples from the following Gaussian distribution: $x_j^{(n)} \sim \mathcal{N}(0, \Sigma_j^\star + \sigma I)$, $n \in [N]$, $j \in [J]$, where $\Sigma_j^\star = \sum_{k=1}^K a_{k,j}^\star v_k^\star v_k^{\star T}$ and σI is additive noise. Unless stated otherwise, we use the Matérn five-half kernel (Minasny and McBratney, 2005) as the smoothing kernel for all simulations.

Method	Model	low-rank	smooth A	sparse V
M1	Sliding window principal component analysis	✓	✓	✗
M2	Hidden Markov model	✗	✗	✗
M3	Autoregressive hidden Markov model (Poritz, 1982)	✗	✓	✗
M4	Sparse dictionary learning (Mairal et al., 2010)	✓	✗	✓
M5	Bayesian structured learning (Andersen et al., 2018)	✓	✓	✓
M6	Lasso and kernel regularization (Daubechies et al., 2010)	✓	✓	✓
M7	Slinding window shrunk covariance (Ledoit and Wolf, 2004)	✗	✓	✗
MS	Spectral initialization (Algorithm 1)	✓	✗	✗
MR	Proposed model with random initialization (Algorithm 2)	✓	✓	✓
M**	Proposed model (Algorithm 1—2)	✓	✓	✓
MQ**	Proposed model (Algorithm 1—2) with QR decomposition	✓	✓	✓

Table 1: List of competing methods. The implementation details of M6 and MR are presented in Appendix G.1 and Appendix G.2.

5.1 Ground-truth recovery and linear convergence

We demonstrate the performance of the proposed algorithm under different smooth temporal structures. The tuning parameters are selected as described in Section 3.2. Figure 1 shows the temporal dynamics of the ground truth together with the results. The upper row corresponds to the setting of mixing temporal weights, where we have sine functions, a constant function, and a ramp function. The bottom row corresponds to the setting with different sine functions. Figure 2 shows how distance $\text{dist}^2(Z, Z^\star)$ changes with the number of subjects $N \in \{1, 5, 15, 200\}$. We see linear convergence of the distance up to some statistical error, which decreases as the sample size increases, as predicted by Theorem 7.

To compare with other methods, we use the average log-Euclidean metric (Arsigny et al., 2006): $J^{-1} \sum_{j=1}^{J} \| \log(\Sigma_j) - \log(\Sigma_j^\star) \|_F$, where $\log(\Sigma_j) = U_j \log(\Lambda_j) U_j^T$, U is the matrix of eigenvectors, and Λ is the diagonal matrix of eigenvalues of Σ_j. In practice, we truncate the eigenvalues whose magnitude is less than 10^{-5} to maintain the stability of the evaluation. Table 2 reports the average log-Euclidean metric, while Table 3 reports the average running time over 20 independent runs. The simulations under discrete switching dynamics are presented in Appendix G.3 and simulations with varying K are presented in Appendix G.4.

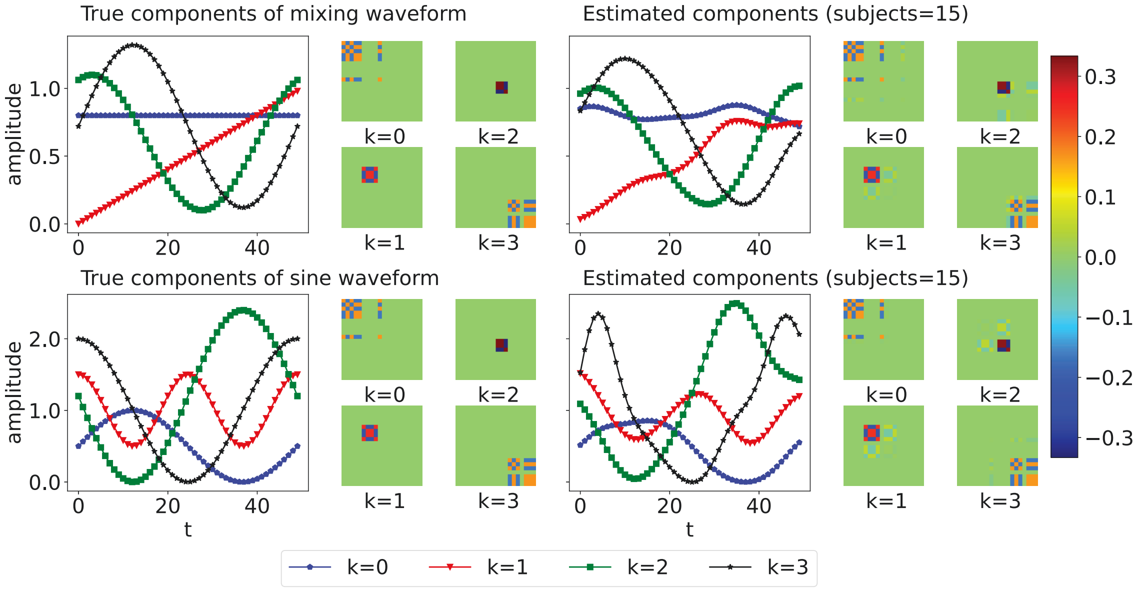

Figure 1: Covariance recovery with $K = 4, P = 20, J = 50$ and $\sigma = 0$. The left two columns show the ground truth and the right two columns show the recovery with $N = 15$. The results indicate good spatial and temporal recovery.

	Mixing waveform			Sine waveform		
Method	$N = 1$	$N = 5$	$N = 10$	$N = 1$	$N = 5$	$N = 10$
M1	0.45 ± 0.01	0.40 ± 0.01	0.38 ± 0.01	0.67 ± 0.03	0.63 ± 0.01	0.63 ± 0.01
M2	6.58 ± 0.31	0.68 ± 0.02	0.62 ± 0.01	6.22 ± 1.02	0.82 ± 0.01	0.80 ± 0.01
M3	6.91 ± 1.39	0.75 ± 0.02	0.65 ± 0.01	7.36 ± 0.24	0.87 ± 0.02	0.80 ± 0.01
M4	0.46 ± 0.01	0.43 ± 0.02	0.39 ± 0.02	0.64 ± 0.01	0.58 ± 0.03	0.54 ± 0.04
M5	0.41 ± 0.01	0.36 ± 0.01	0.34 ± 0.00	0.58 ± 0.01	0.54 ± 0.01	0.53 ± 0.01
M6	0.51 ± 0.03	0.39 ± 0.02	0.37 ± 0.02	0.67 ± 0.03	0.58 ± 0.02	0.57 ± 0.03
MS	0.89 ± 0.05	0.41 ± 0.01	0.38 ± 0.01	0.94 ± 0.06	0.58 ± 0.02	0.57 ± 0.01
MR	0.42 ± 0.01	0.41 ± 0.02	0.41 ± 0.00	0.62 ± 0.00	0.62 ± 0.00	0.62 ± 0.00
M**	0.41 ± 0.03	0.29 ± 0.03	0.30 ± 0.04	0.59 ± 0.02	0.51 ± 0.03	0.50 ± 0.02
MQ**	0.37 ± 0.02	0.31 ± 0.03	0.29 ± 0.03	0.58 ± 0.02	0.56 ± 0.02	0.56 ± 0.01

Table 2: Log-Euclidean metric averaged over 20 independent runs. Temporal dynamics are given in Figure 1 with $K = 4, P = 20, J = 50$ and $\sigma = 0.5$. For M1, we set the window length as $W = 20$. For both task cases, M** and MQ** outperform the competing methods under varying sample size. When $N = 1$, M2 and M3 have sufficiently large average log-Euclidean. This is because M2 and M3 are not designed to be low-rank models, whereas the ground truth is low-rank. Hence, when the estimated covariance matrices are not low-rank, they have many small trailing nonzero eigenvalues, which results in large average log-Euclidean metric.

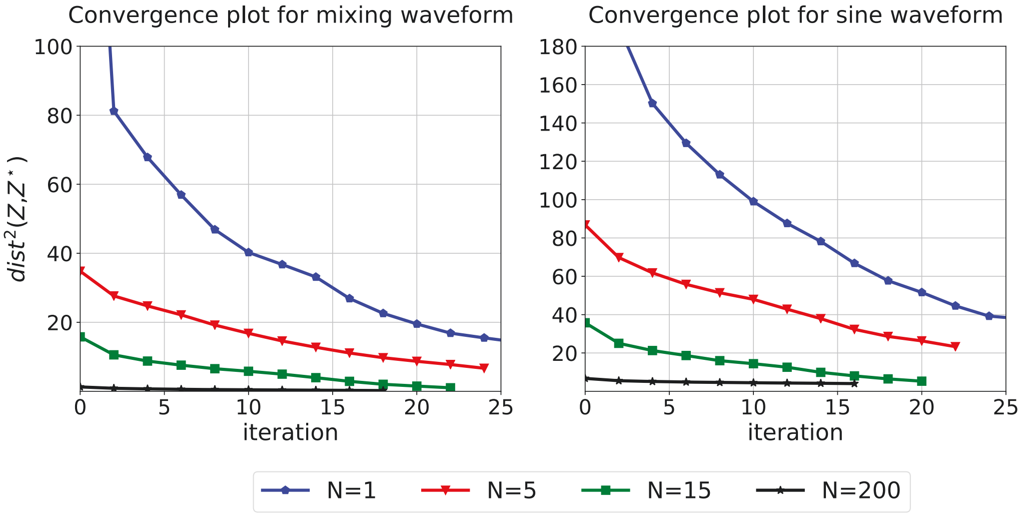

Figure 2: Convergence rate for different sample sizes with $K = 4, P = 20, J = 50$ and $\sigma = 0$. The data generating mechanism is given in Figure 1. Irrespective of the sample size, we observe linear convergence of the algorithm up to a neighborhood of the population parameters. The radius of the neighborhood is characterized by the statistical error, which depends on the sample size.

		Number of subjects (N)		
	Method	1	5	10
Mixing waveform	M1	0.5 ± 0.1	0.4 ± 0.0	0.6 ± 0.0
	M2	300.0 ± 4.0	790.0 ± 10.0	1000.0 ± 40.0
	M3	190.0 ± 30.0	2840.0 ± 10.0	2450.0 ± 70.0
	M4	60.0 ± 30.0	520.0 ± 290.0	1080.0 ± 510.0
	M5	350.0 ± 80.0	360.0 ± 130.0	380.0 ± 60.0
	M6	620.0 ± 50.0	630.0 ± 40.0	620.0 ± 60.0
	MS	0.1 ± 0.0	0.2 ± 0.0	0.2 ± 0.0
	MR	90.0 ± 3.7	220.0 ± 0.0	220.0 ± 3.5
	M**	8.2 ± 1.6	4.5 ± 0.2	5.9 ± 0.6
	MQ**	18.1 ± 2.2	14.0 ± 0.6	13.4 ± 0.3
Sine waveform	M1	0.5 ± 0.1	0.4 ± 0.0	0.5 ± 0.0
	M2	30.0 ± 4.0	800.0 ± 10.0	830.0 ± 7.9
	M3	180.0 ± 20.0	2840.0 ± 20.0	290.0 ± 4.1
	M4	70.0 ± 30.0	320.0 ± 150.0	780.0 ± 330.0
	M5	350.0 ± 80.0	360.0 ± 110.0	380.0 ± 90.0
	M6	570.0 ± 30.0	570.0 ± 30.0	570.0 ± 30.0
	MS	0.1 ± 0.0	0.2 ± 0.0	0.2 ± 0.0
	MR	73.9 ± 11.4	35.9 ± 3.2	42.9 ± 2.0
	M**	6.7 ± 1.7	4.5 ± 0.8	4.3 ± 1.2
	MQ**	15.3 ± 2.4	12.9 ± 0.8	12.0 ± 0.6

Table 3: Running time ($\times 10^{-2} s$) averaged over 20 independent runs. Temporal dynamics are given in Figure 1 with $K = 4, P = 20, J = 50$ and $\sigma = 0.5$. For M1, we set the window length as $W = 20$. When P is small, MS is the most efficient method as it only computes eigendecomposition once. M1 computes eigendecomposition multiple times and, as a result, is slower. The running time of M* is composed of the running time of MS and the running time of Algorithm 2. MQ** is slower than M** because it requires additional QR decomposition at each step.

		Number of subjects (N)				
	Method	1	5	15	200	1000
Mixing waveform	MS	208.12 ± 41.82	45.22 ± 2.62	15.15 ± 1.10	1.17 ± 0.10	0.23 ± 0.01
	MR	29.67 ± 0.39	29.16 ± 0.21	28.86 ± 0.12	28.68 ± 0.04	28.43 ± 0.06
	M**	18.99 ± 5.00	6.72 ± 2.14	2.22 ± 1.13	0.19 ± 0.05	0.04 ± 0.01
Sine waveform	MS	477.54 ± 95.10	103.24 ± 9.64	36.64 ± 3.39	3.53 ± 1.08	1.53 ± 1.46
	MR	95.60 ± 0.63	94.05 ± 0.36	92.86 ± 0.26	91.39 ± 0.12	90.85 ± 0.12
	M**	51.06 ± 12.28	22.27 ± 5.99	9.85 ± 4.38	2.41 ± 3.22	1.44 ± 2.13

Table 4: Average $\mathrm{dist}^2(Z, Z^\star)$ over 20 independent runs. The data generating mechanism is given in Figure 1 with $K = 4, P = 20, J = 50$ and $\sigma = 0$. M** performs the best under varying sample sizes. MR outperforms MS for small sample sizes. However, MS outperforms MR in the large sample size settings.

5.2 Importance of Spectral Initialization

We compare the spectral initialization method in Algorithm 1 (MS), random initialization method with iterative refinement (MR) (see Appendix G.1 for details), and the proposed model (M**) to demonstrate the importance of proper initialization and iterative refinement. The data are generated as in the previous experiment. For each setting, we run the simulation 20 times with $N \in \{1, 5, 15, 200, 1000\}$ and average $\mathrm{dist}^2(Z, Z^\star)$ at convergence. Table 4 indicates that the error of the MS decreases with increasing sample size, which corresponds to the result of Theorem 9. We further observe that M** outperforms MR and MS, indicating that the two-stage algorithm works better than the single-stage algorithms (MR or MS) under varying sample sizes. Figure 3 shows that the MR method converges to poor local optima that correspond to large $\mathrm{dist}^2(Z, Z^\star)$. We also visually observe that the recovered temporal dynamics is far from the population ground truth. When the sample size is small, both MR and M** outperform MS, implying that iterative refinement helps improve estimation in addition to spectral initialization. When the sample size is larger, that is, $N \in \{15, 200, 1000\}$, MS outperforms MR, implying that spectral initialization is a better option than random initialization.

5.3 Simulations with increasing P and K

We increase both the dimension of the data P and the number of components K to demonstrate the effectiveness of the proposed algorithm in a high-dimensional setting. For the data generation process, we randomly generate a sparse orthogonal matrix $V^\star \in \mathbb{R}^{P\times K}$. Specifically, we generate a diagonal block matrix, denoted $\tilde{V}^\star$, with four blocks, and the size of each block is $\lceil K/4 \rceil$. Each block matrix is generated as a random sparse matrix where the probability that an entry is nonzero is 0.4 and each nonzero entry is drawn from unif$([0, 1])$. Finally, we obtain $V^\star$ from the QR decomposition of $\tilde{V}^\star = QR$ as $V^\star = Q$. Note that the spatial components are partially overlapping in this setting, unlike in the previous setting, making the task more challenging. We generate the temporal components $A_{k\cdot}$ as follows. We select 6 knots uniformly at random in $[0, T]$ and draw the corresponding y value from unif$([0, 1])$. These values are interpolated with a cubic spline function.

Table 5 reports the results with $N = 100$ and $J = 100$, while $P \in \{50, 100, 150, 200, 300\}$ and $K \in \{10, 20, 30, 40, 50\}$. The results indicate that with a fixed N, the distance increases with rank, which can be expected since there are more parameters to estimate. Moreover, we also find that dimension P has a small influence on performance.

We compare our algorithm with competing methods in a setting where $P = 100$, since some of the other methods are not scalable to higher-dimensional problems. We set $J = 100$, $K = 10$, and the noise level to $\sigma = 0.5$. Results for $\sigma \in \{0.1, 0.2\}$ are reported in Appendix G.5. For all methods, we set the number of components to estimate as 10, that is, all methods know the true number of components in the data generation process. For the Bayesian model (M5), we draw 30 samples from the posterior distribution and compute the estimated covariance. We report the results averaged over 20 independent

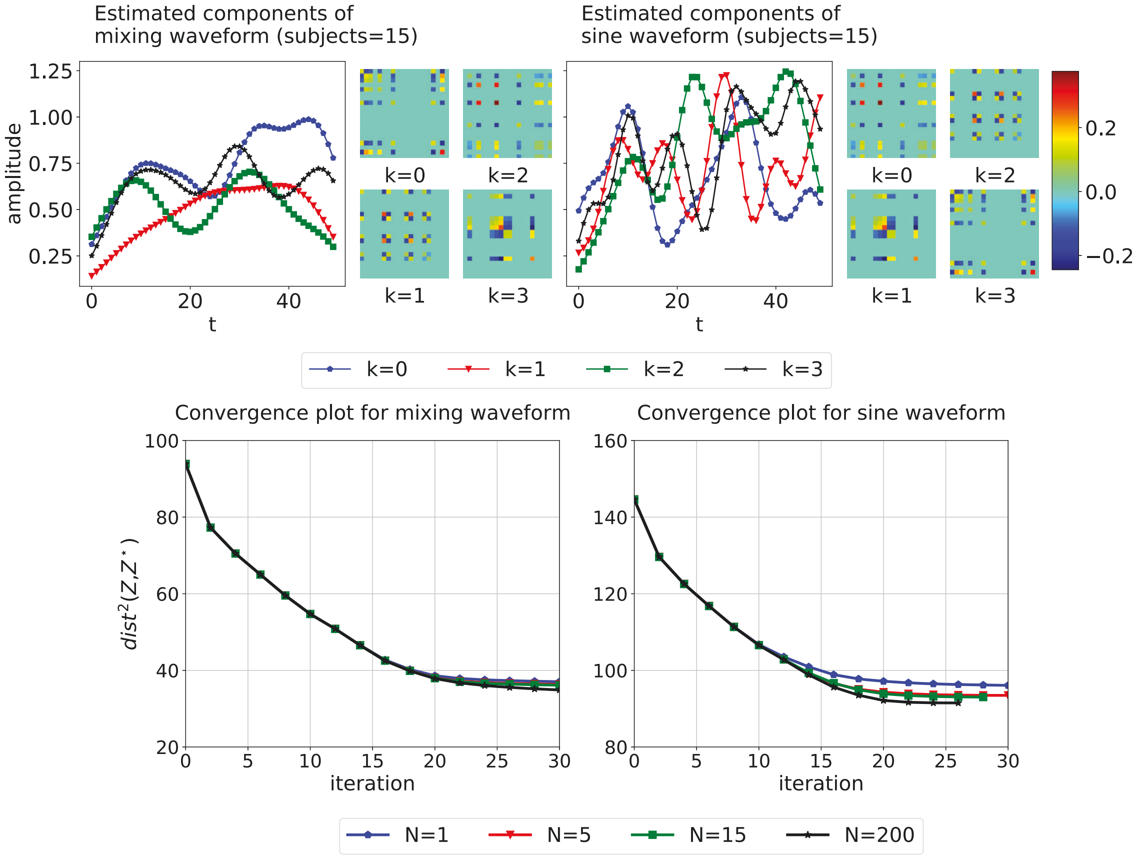

Figure 3: Performance of random initialization method (MR). The data generating mechanism is given in Figure 1. The left two columns in the top row correspond to recovery of mixing waveform, while the right two columns correspond to recovery of sine waveform. The bottom row displays the convergence with respect to the number of iterations.

simulation runs. Table 6 summarizes the results. The proposed algorithm performs the best compared to the alternatives. We also observe that the log-Euclidean metric decreases as the number of samples increases for M**, which is predicted by the results of Theorem 4 and Proposition 8. When comparing MS and M**, we see a decrease in the log-Euclidean metric resulting from Algorithm 2. The results of M** and MQ** are similar, implying that the QR decomposition does not affect the estimation much, supporting the theory that V and V_{ortho} span the same subspace. Our method delivers performance comparable to M5, although M5 uses variational inference in a Bayesian framework. This can be expected, as the underlying models are similar. Table 7 reports the running times for different methods. The running time of the proposed method remains relatively stable as the number of subjects increases. On the other hand, the running time of the M2, M3, and M4 methods increases as N increases. Our method remains efficient even in a high-dimensional setting, while many other methods become slow as the dimension increases. Finally, although the M5 and M6 methods make the same structural assumptions and achieve comparable performance, our method is more computationally efficient. In particular, when the sample size N increases, our method provides the best practical choice.

5.4 Selections of kernel functions

We vary the number of knots to see how the choice of kernel length scale affects the estimation. Moreover, we investigate the effect of the kernel function. We average simulation results over 20 independent runs with $N = 50$, $K = 10$, $P = 100$, $J = 100$ and $\sigma = 0.5$. The components are generated by the same data generation process described in Section 5.3 and we only vary the number of knots. Table 8 shows that as the number of knots increases, indicating that the temporal signal fluctuates more intensively, the optimal choice of length scale decreases. This behavior is observed with all three kernel functions.

6. Experiment on neuroimaging data

To investigate the proposed model on real data, we focus on (i) the interpretability of the model and (ii) the out-of-sample prediction. We use motor task data from the Human Connectome Project functional magnetic resonance imaging (fMRI) data (Van Essen et al., 2013). Data are preprocessed using the existing pipeline (Van Essen et al., 2013), and an additional high-pass filter with a cutoff frequency $0.015Hz$ to remove physiological noise as recommended by Smith et al. (1999). The data consist of five motor tasks: tapping the right hand, tapping the left foot, wagging the tongue, tapping the right foot, and tapping the left hand. During a session, each task is activated twice. See the activation sequence in Figure 4.

For the model interpretation experiment, we select $N = 20$ subjects. For each subject, the preprocessed time series of length $J = 284$ were extracted from $P = 375$ cortical and subcortical parcels, following (Shine et al., 2019). The regions include 333 cortical parcels

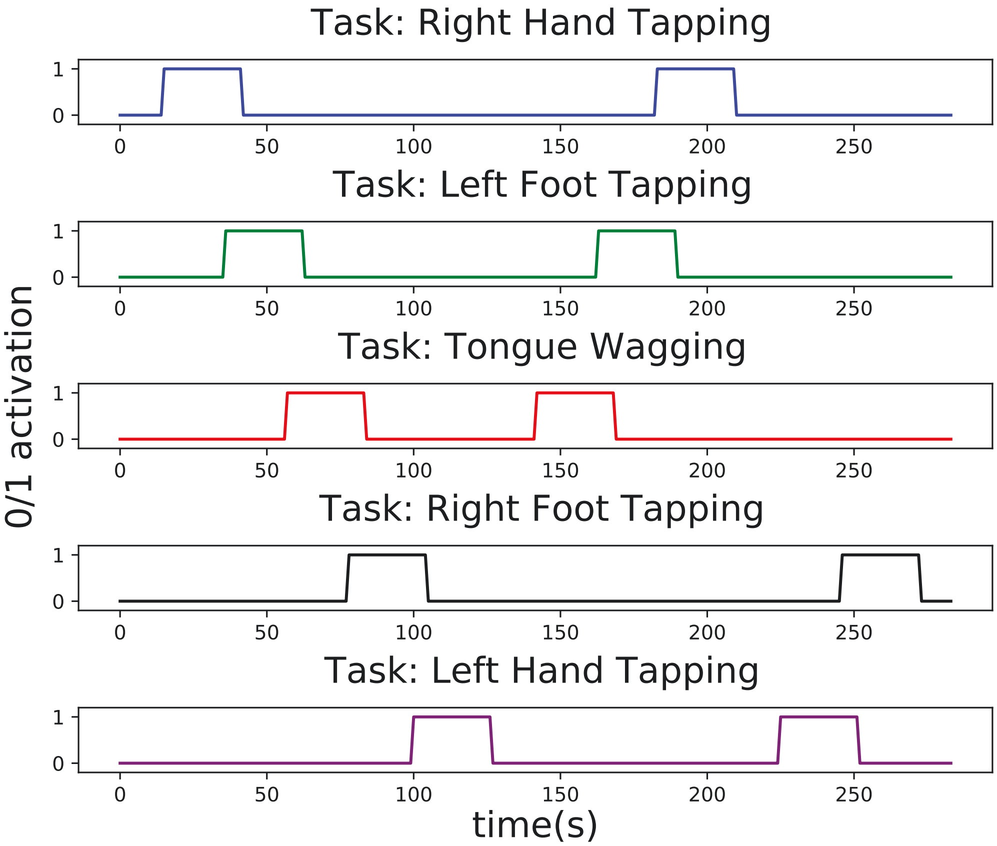

Figure 4: The activation map for the Human Connectome Project motor data set (Van Essen et al., 2013). The activation time of each task is partially overlapping with the other tasks.

(161 and 162 regions from the left and right hemispheres, respectively) using the Gordon atlas (Gordon et al., 2014), 14 subcortical regions from the Harvard–Oxford subcortical atlas (bilateral thalamus, caudate, putamen, ventral striatum, globus pallidus, amygdala, and hippocampus), and 28 cerebellar regions from the SUIT atlas 54 (Diedrichsen et al., 2009).

The goal is to analyze the corresponding dynamic connectivity. To investigate the temporal and spatial components, we compute the correlation of each weight $A_{k\cdot}$ with the onset task activation, and select the component that has the highest correlation. Figure 5 shows the results for three tasks: tapping the left foot, wagging the tongue, and tapping the left hand. Figure 12 in Appendix G.6 shows the results for tapping the right foot

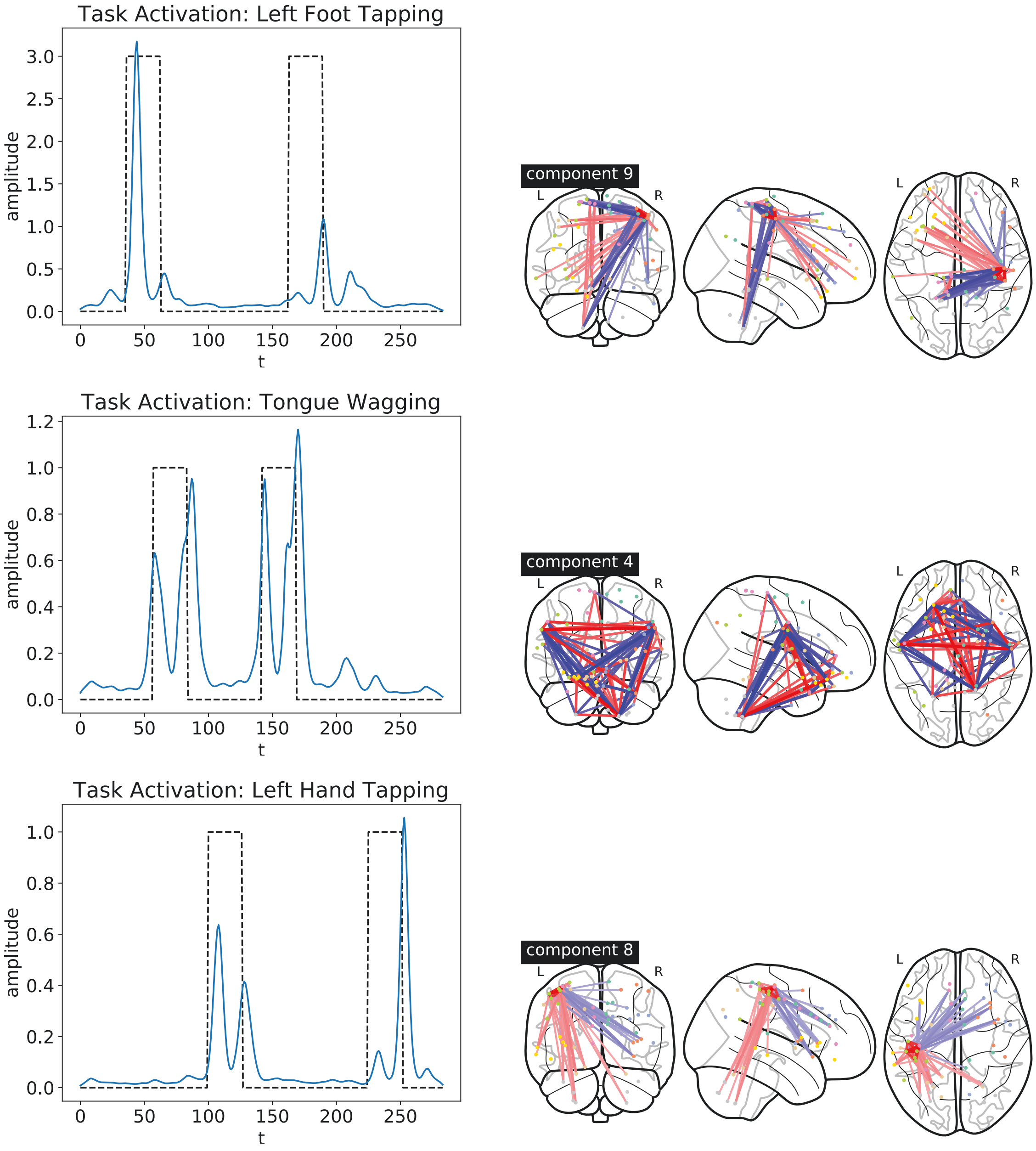

Figure 5: The left column shows the temporal components (blue solid lines) whose correlations are the largest with respect to the task activation (black dotted lines). The right column shows the corresponding brain connectivity patterns (spatial components) for the tasks. The red lines denote positive connectivity and blue lines denote negative connectivity.

	Dimension (P)				
Rank (K)	50	100	150	200	300
$K = 10$	0.35 ± 0.01	0.35 ± 0.02	0.41 ± 0.02	0.35 ± 0.01	0.34 ± 0.03
$K = 20$	0.66 ± 0.02	0.66 ± 0.02	0.69 ± 0.02	0.66 ± 0.03	0.66 ± 0.01
$K = 30$	0.82 ± 0.01	0.82 ± 0.01	0.82 ± 0.02	0.80 ± 0.01	0.79 ± 0.01
$K = 40$	0.97 ± 0.01	0.93 ± 0.01	0.93 ± 0.01	0.88 ± 0.01	0.87 ± 0.01
$K = 50$	1.11 ± 0.01	1.10 ± 0.01	1.04 ± 0.01	0.99 ± 0.01	0.99 ± 0.01

Table 5: Log-Euclidean metric averaged over 20 independent simulation runs ($\sigma = 0.5$). The data generating mechanism is described in Section 5.3. For a fixed sample size and increasing K increases, the log-Euclidean metric increases due to large number of parameters that need to be estimated. The log-Euclidean metric is only mildly affected by the dimension P, which is due to the number of nonzero entries being the same for different values of P.

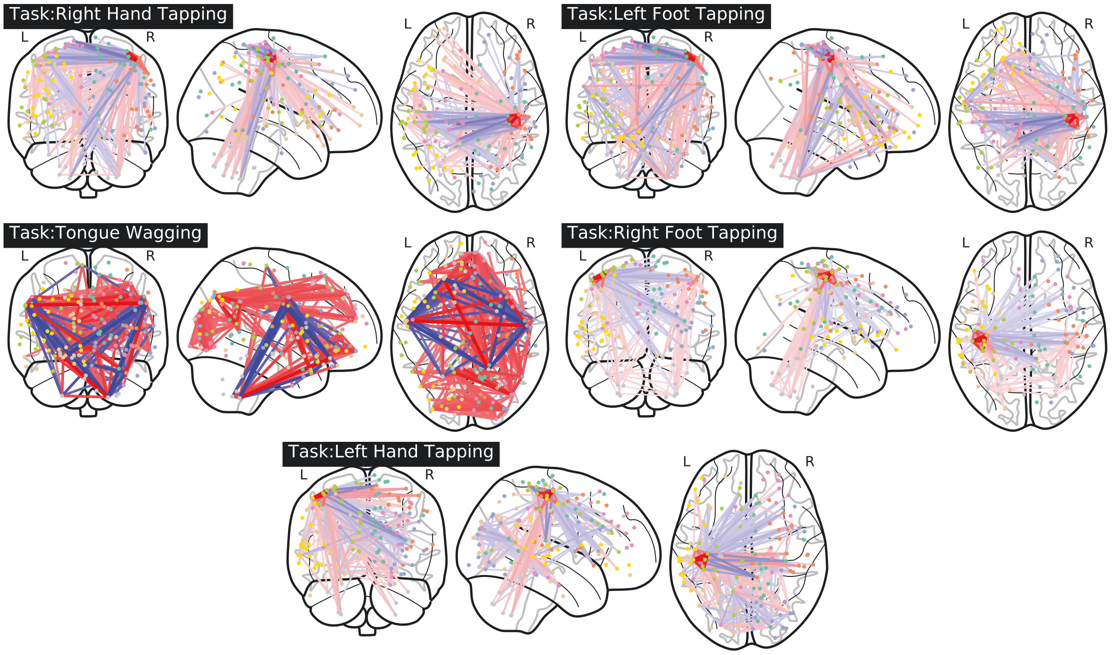

Figure 6: Each connectome is the superposition of the top three spatial components. The spatial hubs in the connectivity matrices closely match with the expected motor regions (hands, feet, tongue) as defined in the cortical homunculus (Marieb and Hoehn, 2018).

	Number of subjects (N)				
Methods	10	20	30	40	50
M1	0.49 ± 0.01	0.46 ± 0.01	0.45 ± 0.01	0.45 ± 0.01	0.44 ± 0.01
M2	1.22 ± 0.01	1.04 ± 0.01	1.00 ± 0.01	0.98 ± 0.01	0.97 ± 0.01
M3	71.50 ± 6.46	1.90 ± 0.26	1.12 ± 0.01	1.14 ± 0.01	1.12 ± 0.01
M4	0.94 ± 0.03	0.46 ± 0.01	0.41 ± 0.01	0.39 ± 0.01	0.38 ± 0.01
M5	0.51 ± 0.01	0.46 ± 0.01	0.43 ± 0.01	0.42 ± 0.01	0.41 ± 0.01
M6	0.43 ± 0.01	0.41 ± 0.01	0.40 ± 0.01	0.40 ± 0.01	0.39 ± 0.01
MS	0.43 ± 0.01	0.40 ± 0.01	0.40 ± 0.01	0.39 ± 0.01	0.39 ± 0.01
M**	0.42 ± 0.03	0.35 ± 0.05	0.36 ± 0.04	0.33 ± 0.04	0.32 ± 0.02
MQ**	0.40 ± 0.04	0.40 ± 0.01	0.35 ± 0.04	0.32 ± 0.03	0.32 ± 0.02

Table 6: Log-Euclidean metric averaged over 20 independent simulation runs. The data generating mechanism is described in Section 5.3 and $\sigma = 0.5$. For M1, we set the window length to be $W = 20$.

	Number of subjects (N)				
Methods	10	20	30	40	50
M1	0.8 ± 0.4	0.5 ± 0.3	1.2 ± 0.6	1.5 ± 0.5	1.8 ± 0.5
M2	151.9 ± 17.9	222.6 ± 51.0	376.7 ± 53.6	498.8 ± 40.0	635.5 ± 58.2
M3	422.0 ± 33.3	729.7 ± 161.6	1154.4 ± 53.6	1427.4 ± 64.4	1751.7 ± 52.0
M4	267.0 ± 112.1	378.4 ± 148.5	846.2 ± 358.8	872.0 ± 440.8	1841.5 ± 697.2
M5	2243.8 ± 33.3	2263.3 ± 38.4	2273.7 ± 36.0	2259.8 ± 34.2	2278.9 ± 35.1
M6	84.9 ± 37.5	195.2 ± 62.6	201.8 ± 31.9	191.6 ± 51.6	218.3 ± 15.6
MS	0.1 ± 0.0	0.1 ± 0.1	0.2 ± 0.1	0.2 ± 0.1	0.2 ± 0.1
M**	1.2 ± 0.6	1.3 ± 0.7	2.9 ± 1.4	3.9 ± 0.8	3.6 ± 0.7
MQ**	2.2 ± 1.3	1.4 ± 0.7	2.5 ± 1.0	3.8 ± 0.7	3.5 ± 0.6

Table 7: Running time in *seconds* averaged over 20 independent simulation runs. The data generating mechanism is described in Section 5.3 and $\sigma = 0.5$. For M1, we set the window length to be $W = 20$.

and tapping the right hand. Our results show that the temporal fluctuations of the top components coincide with the task activation. Following the hypothesis that neural activity is the consequence of multiple components rather than a single component (Posner et al., 1988), for each task, we select three components with the highest correlations and plot the connectivity patterns in Figure 6. The spatial hubs in the connectivity matrices closely match the expected motor regions as defined in the cortical homunculus (Marieb and Hoehn, 2018). Thus, the results indicate that the proposed algorithm can separate and identify the components of each task and that each task has a unique connectivity pattern.

As the ground truth is unknown and motivated by the hypothesis that each task has a different activation pattern, we design a classification task as a surrogate experiment to evaluate the algorithm. Previous work also indicated that task fMRI data share similar connectivity patterns among test subjects (Zalesky et al., 2012; Calhoun et al., 2014).

	Number of knots in $J = 100$			
Methods	5	10	15	20
Radial-basis function ($l = 5$)	0.16 ± 0.04	0.26 ± 0.05	0.52 ± 0.07	0.91 ± 0.17
Radial-basis function ($l = 10$)	0.12 ± 0.04	0.18 ± 0.05	0.33 ± 0.07	0.69 ± 0.18
Radial-basis function ($l = 50$)	0.09 ± 0.04	0.16 ± 0.05	0.76 ± 0.09	1.17 ± 0.11
Radial-basis function ($l = 200$)	0.08 ± 0.04	0.30 ± 0.05	0.84 ± 0.08	1.27 ± 0.16
Matérn five-half ($l = 5$)	0.15 ± 0.04	0.23 ± 0.05	0.44 ± 0.06	0.80 ± 0.18
Matérn five-half ($l = 10$)	0.13 ± 0.04	0.20 ± 0.05	0.36 ± 0.06	0.70 ± 0.18
Matérn five-half ($l = 50$)	0.10 ± 0.04	0.15 ± 0.05	0.31 ± 0.06	0.59 ± 0.12
Matérn five-half ($l = 200$)	0.09 ± 0.04	0.14 ± 0.05	0.31 ± 0.07	0.62 ± 0.19
Rational quadratic ($l = 5$)	0.14 ± 0.04	0.24 ± 0.05	0.51 ± 0.07	0.89 ± 0.18
Rational quadratic ($l = 10$)	0.17 ± 0.04	0.29 ± 0.05	0.61 ± 0.07	1.03 ± 0.18
Rational quadratic ($l = 50$)	0.10 ± 0.04	0.13 ± 0.05	0.43 ± 0.07	1.07 ± 0.18
Rational quadratic ($l = 200$)	0.08 ± 0.04	0.22 ± 0.05	0.81 ± 0.08	1.26 ± 0.16

Table 8: Average distance $\mathrm{dist}^2(Z, Z^\star)/J$ over 20 independent runs with $\sigma = 0.5$. Results are comparable for different kernel functions when the number of knots is smaller than $J = 15$. When $J = 20$, Matérn five-half kernel function is more effective in capturing the temporal smoothness compared to the other two kernels.

Therefore, if we can recover the functional connectivity patterns of the training subjects, then similar patterns exist in the test subjects. We partition 103 subjects in the Human Connectome Project motor task data set (Van Essen et al., 2013) randomly into a training and testing set. The duration of each task is identical, 27 time points for each activation, and 2 activations in each session. Since each task partially overlaps with others (see Figure 4), we predict the task based on activation blocks rather than on a single time point. We group the estimated covariances $\{\Sigma_j\}_{j\in[J]}$ and the test data based on the task activation map and perform a nearest-neighbor search. Clustered covariances are denoted as $\Sigma_{\text{task},i}$, where task $\in$ {tapping the right hand, tapping the left foot, wagging the tongue, tapping the right foot, tapping the left hand} and $i \in [54]$. The task score for each test data block is defined as

$$score_{task}(\{x_i\}_{i\in[54]}) = \sum_{i=1}^{54} \|x_i x_i^T - \Sigma_{task,i}\|_F^2,$$

where $\{x_i\}_{i\in[54]}$ is a block of test data. We predict the task of the block data by choosing the task with the minimum score. We repeat the experiment 10 times. In each run, we randomly split the data into a training and testing set. We select the number of training subjects to be $N \in \{10, 20, 30, 40, 50\}$ and set the remaining subjects as test sets. The results are shown in Table 9. Note that the Markov model (M2) performs worst even if we increase the number of states to 60. The dictionary learning model (M4) performs similarly to our model when the sample size is large, but our model performs better with small sample sizes.

7. Discussion

Several directions are worthy of further investigation. We plan to explore a more flexible spatial structure. Previous work (Gibberd and Nelson, 2017; Hallac et al., 2017) applied fused graphical lasso and group graphical lasso to encourage similar sparse structures for time-varying graphical models. These approaches did not restrict the spatial components to be identical but only similar, and thus are more flexible compared to the proposed model. To this end, one idea is to build factor models that encourage similar, but not identical, spatial structures while retaining low rank. Our work has focused on modeling data sampled at fixed intervals. We also plan to explore models with samples obtained at irregular time intervals (Tank et al., 2019; Qiao et al., 2020), as this setting is common in multimodal data (Tsai et al., 2022). Finally, our work has focused on the estimation of parameters in a flexible covariance model, while the question of how to quantify the statistical uncertainty remains open. There has been a growing literature on inference for parameters in high-dimensional models, including linear models (Zhang and Zhang, 2014; van de Geer et al., 2014; Javanmard and Montanari, 2014; Zhao et al., 2014; Bradic and Kolar, 2017; Dai and Kolar, 2021; Wang et al., 2021), nonparametric models (Kozbur, 2021; Lu et al., 2020), and graphical models (Ren et al., 2015; Wasserman et al., 2014; Janková and van de Geer, 2015, 2017; Barber and Kolar, 2018; Wang and Kolar, 2016; Yu et al., 2016, 2020b; Xia et al., 2015; Kim et al., 2021). The model considered in our paper is comprised of a sparse spatial and a smooth nonparametric temporal component that will require the development of new inferential techniques.

Acknowledgments

We thank James M. Shine for help with data preprocessing and helpful modeling discussions. The research project is partially funded by the U.S.A. National Institutes of Health 1R01MH116226-01A, National Science Foundation Graduate Research Fellowships Program, NSF IIS 2046795 and Strategic Research Initiatives Grainger College of Engineering, the University of Illinois at Urbana-Champaign. Data were provided by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.