Journal of Machine Learning Research 26 (2025) 1-69 Submitted 4/23; Revised 6/25; Published 7/25

# Latent Process Models for Functional Network Data

**Peter W. MacDonald** PWMACDONALD@UWATERLOO.CA
*Department of Statistics & Actuarial Science*
*University of Waterloo*
*Waterloo, ON N2L 3G1, Canada*

**Elizaveta Levina** ELEVINA@UMICH.EDU
**Ji Zhu** JIZHU@UMICH.EDU
*Department of Statistics*
*University of Michigan*
*Ann Arbor, MI 48109-1107, USA*

**Editor:** Mladen Kolar

## Abstract

Network data are often sampled with auxiliary information or collected through the observation of a complex system over time, leading to multiple network snapshots indexed by a continuous variable. Many methods in statistical network analysis are traditionally designed for a single network, and can be applied to an aggregated network in this setting, but that approach can miss important functional structure. Here we develop an approach to estimating the expected network explicitly as a function of a continuous index, be it time or another indexing variable. We parameterize the network expectation through low dimensional latent processes, whose components we represent with a fixed, finite-dimensional functional basis. We derive a gradient descent estimation algorithm, establish theoretical guarantees for recovery of the low dimensional structure, compare our method to competitors, and apply it to a data set of international political interactions over time, showing our proposed method to adapt well to data, outperform competitors, and provide interpretable and meaningful results.

**Keywords:** Latent space model, Multilayer, Multiplex, Dynamic network, $B$-spline

## 1. Introduction

Modern data are collected in a much greater variety of forms than classical statistics considered, and require novel models and methods. Networks are one important example of a complex data structure which has received recent interest in many fields of application. In general, network data on $n$ statistical units, or nodes, describe connections, or edges, between pairs of those units. This information is stored in the $n \times n$ adjacency matrix $A$, where each entry $\{A_{ij} : i, j = 1, \ldots, n\}$ describes the connection from node $i$ to node $j$, which could be binary or real-valued. Much of the statistical network analysis literature deals with a single network, sometimes with auxiliary information, but increasingly samples of networks are also studied. Samples of networks arise in diverse applications: for instance, in neuroimaging we observe a brain connectivity network for each study subject, and in political science, we observe relations between countries over many years. In this

paper, we focus on functional network data, the setting where edges are indexed by a continuous auxiliary variable. This variable is commonly time, but it can be anything else that has a natural ordering.

Often, functional network data are collected as repeated, indexed snapshots of a single evolving network. In this regime, suppose that we observe a network on a common set of $n$ nodes, at $m$ distinct indices $\{x_k\}_{k=1}^m$ in a compact set $\mathcal{X} \subseteq \mathbb{R}$. The complete data is made up of a collection of indexed adjacency matrix snapshots, $\{A_k\}_{k=1}^m \subseteq \mathbb{R}^{n \times n}$, where each matrix entry $[A_k]_{ij}$ gives the value of edge $(i, j)$ from node $i$ to node $j$ for the snapshot corresponding to index $x_k$. These adjacency matrices may have binary edges with values in $\{0, 1\}$, or weighted edges taking any real value, and may be either undirected or directed. Examples of functional network data collected as time-indexed snapshots include dynamic friendship networks (Snijders, 2017), or dynamic networks of international conflict (cf. Section 6). However, networks may also be indexed in some other continuous covariate; for instance brain images for a collection of subjects, indexed by a single continuous task score (Shan, 2022).

It is also common that functional network data are collected in the form of time-stamped records of interactions between pairs of nodes. Examples of functional network data collected in this manner include e-mail networks, bike share networks (Matias et al., 2018), or animal interaction networks (Mersch et al., 2013). We can construct snapshots from such event data by partitioning the index set into $m$ contiguous intervals, and constructing weighted adjacency matrices which count the number or total weight of events between nodes $i$ and $j$ in a given interval.

Another way to view adjacency matrix snapshots is to treat them as a multiplex network (Kivelä et al., 2014), a multilayer network object with $m$ layers corresponding to the snapshots. However, the ordering of the layers, inherited from the ordering of the indexing variable, gives them additional structure not usually present in a multiplex network.

Much of the existing literature focuses on the dynamic network setting, with the snapshots indexed by time, rather than the general functional setting. We briefly review recent related work for dynamic networks; see Kim et al. (2018) for a more detailed review. Approaches based on the Stochastic Block Model (SBM) include Matias and Miele (2017), which assumes node community memberships follow a Markov chain and allows connection probabilities to vary; Bhattacharyya and Chatterjee (2018), which assumes community memberships are fixed but allows connection probabilities to vary; and Pensky and Zhang (2019), which allows both community memberships and connection probabilities to vary smoothly in time. Latent space or latent position models for networks have also been popular since the seminal work of Hoff et al. (2002). Latent space approaches to dynamic networks go back to Sarkar and Moore (2005), which models latent positions in discrete time with independent Gaussian random walks. A Bayesian latent space approach with similar random walk transitions on latent positions in discrete time was proposed by Sewell and Chen (2015), and extended to continuous time by Durante and Dunson (2016). Both Lee and Priebe (2011) and Padilla et al. (2022) consider extensions of the random dot product graph (RDPG) to the dynamic setting, for the purpose of changepoint detection. In all of these papers, the single network latent space framework is extended by mapping each node to a sequence or continuum of positions in the latent space, with network snapshots which are conditionally independent given the positions. More recently, Athreya et al. (2025) and

Chen et al. (2024) consider a similar functional RDPG model, and make a valuable contribution towards the statistical interpretation of a sequence of embeddings of functional network snapshots, in order to reveal the overall changes in network structure over time. In contrast, our work contributes new, improved methodology for producing such a sequence.

Direct modeling of indexed dyadic connection events has been studied by Perry and Wolfe (2013) and Kreiß et al. (2019), among others, again in the dynamic case. These two papers model edge event intensities using time-varying edge covariates, with constant and time-varying coefficients, respectively. A variational Bayes approach to fitting edge event intensities according to a latent community structure was proposed by Matias et al. (2018). A similar community-based model was proposed by Arastuie et al. (2020), focusing on the effect of self-excitation on community detection.

In this paper, we propose a new model for functional network data: a continuously indexed inner product latent position model, which we will call a *latent process model*. In contrast to many previous approaches for functional latent position models, which treat the positions as random variables, we treat the latent processes as function-valued parameters. As a result, our approach does not rely on any assumptions of an explicit, simple, and discrete transition model for the latent positions between snapshots (Matias and Miele, 2017; Sarkar and Moore, 2005; Sewell and Chen, 2015; Padilla et al., 2022). Instead, it allows for arbitrary, continuously varying latent processes, with estimation efficency depending primarily on their functional smoothness. This means that our approach can easily handle irregularly spaced snapshot indices and missing edge entries, with faster computation time than similarly flexible Bayesian methods (Durante and Dunson, 2016).

Our key contribution is to make this function estimation problem tractable by modeling latent processes using a finite, prespecified function basis. By adaptively selecting the basis based on the data, we are able to share information to efficiently estimate network structure which is shared locally between snapshots, but need not assume that that any part of that structure is common to all snapshots (Bhattacharyya and Chatterjee, 2018; Arroyo and Athreya, 2021). In contrast to approaches which first smooth network snapshots and then estimate latent network structure (Pensky and Zhang, 2019), our estimation approach performs both tasks simultaneously, hence it can adapt to smoothness in the latent structure that may not be directly detectable from the network edges. Conversely, compared to approaches which estimate latent network structure for each snapshot, then summarize or smooth the output (Sanna Passino et al., 2021; Athreya et al., 2025; Chen et al., 2024), our approach can accurately estimate the latent processes as the edge variance increases (see Figure 1), as its performance does not rely directly on the signal-to-noise ratio of the individual network snapshots.

The rest of this paper is organized as follows. In Section 2, we define the latent process network model. In Section 3, we develop an estimation algorithm using gradient descent on coordinates in a function basis. Section 4 provides theoretical guarantees for recovery of the latent processes. Section 5 investigates the proposed methods in simulation studies, and Section 6 applies them to a data set of international political interactions. Finally, Section 7 contains brief discussion and future research directions. Mathematical proofs, technical details, and additional simulation studies have been provided as supplementary material.

## 2. Latent Process Network Models

In this section, we will introduce latent process network models. To begin, we fix some notation that we will use in the remainder of this paper.

Throughout, $\|\cdot\|_F$ will denote the matrix Frobenius norm, $\|\cdot\|_2$ the matrix $\ell_2$ operator norm, and $\langle\cdot,\cdot\rangle$ the Frobenius inner product. When applied to a vector, these coincide with the standard vector $\ell_2$ norm and inner product. With some abuse of notation, we will analogously use $\|\cdot\|_F$ to denote the vector $\ell_2$ norm of the vectorization of a 3-mode tensor. The notation $[\cdot]_{ij}$ denotes the $(i,j)$-th entry of a matrix. We use $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ to denote the maximum and minimum eigenvalues of a symmetric matrix; The condition number is defined as the ratio of the maximum and minimum singular values of a matrix, it is bounded between 0 and 1; and $\mathcal{O}_d$ will denote the set of $d \times d$ orthogonal transformation matrices, which includes rotations, reflections, and combinations of the two.

Capital calligraphic letters are generally reserved for 3-mode tensors, and notation for tensor operations will follow Kolda and Bader (2009). The slices of a 3-mode tensor are constructed by fixing the first, second or third index respectively, and varying the other two. The fibers of a 3-mode tensor are constructed by fixing two indices and varying the third. For $m = 1, 2, 3$, tensor-matrix multiplication in the $m$th mode is defined by the operator $\times_m$ and computes the $m$th mode fibers of the product by premultiplying each $m$th mode fiber of the first tensor argument by the second matrix argument. For $m = 1, 2, 3$, tensor-vector multiplication in the $m$th mode is defined by the operator $\bar{\times}_m$ and matrix-vector multiplies $m$th mode slice of the first tensor argument by the second vector argument. Note that tensor-vector multiplication in the $m$th mode results in a matrix with dimensions corresponding to the other two modes of the original tensor argument.

### 2.1 Latent Functional Parameterization

We parameterize functional networks through a matrix-valued network mean function $\Theta : \mathcal{X} \to \mathbb{R}^{n\times n}$, such that $\Theta_{ij}(x_k) = \mathbb{E}([A_k]_{ij})$ for all $i, j$ and $k = 1, \ldots, m$. We assume independent edges, that is, $[A_k]_{ij}$ are independent for all $i \leq j$ and $k$. Formally, suppose $q(\cdot; \theta, \phi)$ is the edge distribution, parameterized by its mean $\theta$ and some possible nuisance parameters $\phi$. Then for $1 \leq i, j \leq n$ and $k = 1, \ldots, m$,

$$[A_k]_{ij} \overset{\text{ind}}{\sim} q(\cdot; \Theta_{ij}(x_k), \phi).$$

For instance, we could model edges with $q(\cdot; \theta, \sigma) = \mathcal{N}(\theta, \sigma^2)$, in which case the network is fully parameterized by $\Theta$ and the nuisance edge variance $\sigma^2$. For binary edge networks, we could model $q(\cdot; \theta, \phi) = \text{Bernoulli}(\theta)$, in which case the model is fully parameterized by $\Theta$. In making the independence assumption, we follow both the single network latent space literature, which typically assumes edge independence conditional on latent positions (Athreya et al., 2018), and multilayer network latent space models that also make the assumption of independence across layers (MacDonald et al., 2022). While the independence assumption is likely not exactly correct, it is a common and useful analysis tool for estimating the network structure, and, in this setting, the functional trends in this structure.

For a fixed node pair $(i, j)$, we may also view the sequence of random variables $\{[A_k]_{ij}\}_{k=1}^m$ as a univariate functional response with inputs $x_k$. The independence of edges over $k$ implies

that these responses have a mean which is a function of the continuous index $x$, and independent errors. Hence the focus of this work is on modeling functional mean structure with no temporal dependence. In general functional settings, for instance where network snapshots represent brain scans of different patients indexed by a continuous task performance score, independence across snapshots is a reasonable working assumption. In dynamic networks, independence across snapshots is not guaranteed, but still commonly assumed conditional on latent structure (e.g., Sewell and Chen, 2015). In future work, we may consider allowing within-edge autoregressive errors, or other forms of dependence. However, accurate nonparametric estimation of the underlying trend component, which we develop in this paper, will be of primary interest in many applications, and such an estimator is necessary to make individual edge sequences stationary before applying time series modeling approaches to the residuals (Fan and Yao, 2003).

Extending the latent space modeling approach, we assume that the parameter of interest $\Theta$ is determined by the trajectories of each node in a $d$-dimensional latent space, denoted by $Z^{(i)} : \mathcal{X} \to \mathbb{R}^d$ for $i = 1, \ldots, n$. We denote the component functions by $Z^{(i)} = (z_{i,1}, \ldots, z_{i,d})^\intercal$. Throughout this paper, we assume an inner product similarity function, namely that $\Theta_{ij}(x) = \{Z^{(i)}(x)\}^\intercal Z^{(j)}(x)$ for any $1 \leq i, j \leq n$ and $x \in \mathcal{X}$. We refer to this as an inner product *latent process network model*. Collecting all the latent processes $Z^{(i)}$ into rows of an $n \times d$ matrix-valued function $Z$, we can write $\Theta(x) = Z(x)Z(x)^\intercal$. For each $x$, $Z(x)Z(x)^\intercal$ is a rank $d$ matrix. The appeal of the inner product similarity is a parameterization of the network mean function which is low rank for any $x \in \mathcal{X}$.

Evaluating the latent processes at the snapshot indices, we see that our function-valued parameterization produces a tensor decomposition of the $n \times n \times m$ tensor $\mathbb{E}(\mathcal{A})$ with $n \times n$ slices $\mathbb{E}(A_k)$ for $k = 1, \ldots, m$, similar to the Tucker or canonical polyadic (CP) decompositions (Kolda and Bader, 2009). However, our formulation does not force the latent structure to factorize in the third mode. Consider the simple case where $d = 1$, and compare our parameterization

$$\mathbb{E}(A_k) = Z(x_k)Z(x_k)^\intercal$$

to a representation of $\mathbb{E}(\mathcal{A})$ by a CP decomposition which is symmetric in the first two modes. The latent process representation cannot in general be reproduced by a rank 1 CP decomposition, which would require that $\mathbb{E}(A_k) = w_k \boldsymbol{z}\boldsymbol{z}^\intercal$ for an $n$-vector $\boldsymbol{z}$ and scalars $w_k$ for $k = 1, \ldots, m$. Under the rank 1 CP decomposition, the expected value of every network snapshot would share a common eigenvector. To capture the structure of the latent process representation would require a rank $m$ CP decomposition with different eigenvectors, and in general no dimension reduction or information sharing across snapshot means.

While we still assume there is a low rank representation of each snapshot mean, we treat them as functions of the index, and propose methodology with good theoretical and empirical properties when the representations are smooth in $x$. In other words, we do not force our functional network models to rely on tensor-valued extensions of matrix algebra procedures, recognizing that node modes should be treated differently from the index mode. In the two node modes with the third index mode fixed, the data are assumed to have low rank or latent position structure, a highly successful and popular approach for network

models. On the other hand, in the third index mode the data are assumed to have smooth structure as a function of the index, as in classical nonparametric regression.

### 2.2 Identifiability

Latent space models with inner product link functions are well known to be non-identifiable due to their invariance to orthogonal transformations of the latent positions, since $\mathbb{E}(A) = XX^\intercal = XO(XO)^\intercal$ for any $n \times d$ matrix $X$ and $d$-dimensional orthogonal transformation $O \in \mathcal{O}_d$. This non-identifiability also extends to continuous time: for any orthogonal matrix-valued function $Q : \mathcal{X} \to \mathcal{O}_d$, we have

$$Z(x)Z(x)^\intercal = Z(x)Q(x)\left\{Z(x)Q(x)\right\}^\intercal. \tag{1}$$

Thus, $Z$ is identifiable only up to uncountably many orthogonal transformations. In particular, for a given $Z$, we define the unidentified class of latent processes $\mathcal{T}(Z)$ by

$$\mathcal{T}(Z) = \left\{Z(x)Q(x) : Q : \mathcal{X} \to \mathcal{O}_d\right\}. \tag{2}$$

Our goal is to take advantage of smoothness in $Z$ to share information across network snapshots. The non-identifiability could mask the smoothness, because even if $Z(x)$ is a smooth function of $x$, $Z(x)Q(x)$ may not be. Although all elements of $\mathcal{T}(Z)$ lead to identically distributed network snapshots, an estimation algorithm which targets the class representative which is "maximally smooth" will have the greatest potential to share information across snapshots and improve estimation efficiency. In general, our theory will instead consider the distance between an estimate $\widehat{Z}$ and an unknown representative of $\mathcal{T}(Z)$.

## 3. Estimation With Gradient Descent

The latent process assumption reduces the parameter space from $n^2$ to $nd$ function-valued parameters. To further simplify estimation, we will restrict to a finite dimensional parameter space by assuming that each component function of each latent process, $z_{i,r}$, $i = 1, \ldots, n$, $r = 1, \ldots, d$ is well approximated by the span of a common $q$-dimensional function basis. That is, suppose that $B = (B_1, \ldots, B_q)^\intercal$ is a $q$-dimensional basis of functions each mapping from $\mathcal{X}$ to $\mathbb{R}$. We assume that every component function $z_{i,r}(x)$ is close to $\boldsymbol{w}_{i,r}^\intercal B(x)$ for some $q$-dimensional coordinate vector $\boldsymbol{w}_{i,r}$. For each $r$, we collect $\boldsymbol{w}_{i,r}$ as the rows of an $n \times q$ matrix $\boldsymbol{W}_r$. Let $\mathcal{W} = \{\boldsymbol{W}_r\}_{r=1}^d$ denote the $n \times q \times d$ tensor containing all the basis coordinates for all nodes in all latent dimensions. For such a coordinate tensor, the first mode corresponds to the nodes, the second mode to the index space after summarizing from $m$ snapshots into $q$ basis coordinates, and the third mode to the latent space dimensions.

The basis $B$ could be, for instance, a $B$-spline basis on $\mathcal{X}$, or any other similar functional basis. A $B$-spline basis of order $D \geq 0$ is defined by an increasing sequence of $K$ internal knots, as well as boundary knots. The dimension of a $B$-spline basis is $q = K + D + 1$. The span of a given $B$-spline basis is a collection of piecewise polynomial functions which are $(D-1)$-times differentiable at the internal knots and smooth elsewhere. For additional mathematical properties, we refer readers to Schumaker (2007). In this paper, we will use order 3, or cubic $B$-spline bases, although the algorithms to follow would proceed similarly

for any function basis $B$, including orthonormal function bases such as the Fourier basis. The simulations and real data analysis in Sections 5 and 6 explore the performance of $B$-splines in detail, but we also compare performance using penalized smoothing splines in Appendix E of the supplementary materials.

For simplicity, we assume a common basis $B$ for all $i = 1, \ldots, n$ and all $r = 1, \ldots, d$, although this could also be relaxed. In practice, the underlying latent processes may not exactly belong to span($B$). However, if the latent processes are smooth in $x$, we will be able to approximate them effectively with functions in span($B$). In Section 3.2, we develop an approach to choose the dimension, and thus the approximation power of $B$ adaptively from data.

We begin by defining a nonconvex least squares optimization problem equivalent to maximizing a Gaussian likelihood. In examples in Section 5, we minimize the same objective function for other edge distributions, in particular the binary edge model with $q(\theta; \phi) =$ Bernoulli($\theta$). Denote

$$\ell(\mathcal{W}) = \sum_{k=1}^{m} \|A_k - \sum_{r=1}^{d} \boldsymbol{W}_r B(x_k) B(x_k)^\intercal \boldsymbol{W}_r^\intercal\|_F^2. \tag{3}$$

Fixing the latent space dimension $d$, and a $q$-dimensional function basis $B$, we can apply gradient descent over the $n \times q \times d$ tensor-valued argument $\mathcal{W}$ with $n \times q$ slices $\boldsymbol{W}_r \in \mathbb{R}^{n \times q}$ for $r = 1, \ldots, d$. Throughout the paper, we will store the basis coordinates in this way as 3-mode tensors, where the first mode corresponds to the nodes, the second to the index space, and the third to the dimensions of the latent space. The following proposition derives the gradient of $\ell$ with respect to each $n \times q$ slice of $\mathcal{W}$. The proof is provided in Appendix A of the supplementary materials.

**Proposition 1** *Define $\ell(\mathcal{W})$ as in (3). Then*

$$\frac{\partial \ell}{\partial \boldsymbol{W}_r}(\mathcal{W}) \propto -\sum_{k=1}^{m} \left\{ A_k - \sum_{r'=1}^{d} \boldsymbol{W}_{r'} B(x_k) B(x_k)^\intercal \boldsymbol{W}_{r'}^\intercal \right\} \boldsymbol{W}_r B(x_k) B(x_k)^\intercal \in \mathbb{R}^{n \times q}$$

*for $r = 1, \ldots, d$.*

Since (3) is a nonconvex objective, gradient descent will not necessarily converge to the global optimum, and the result depends on the starting value. In Section 4, we will directly prove results for the output of the gradient descent algorithm proposed below, rather than for the global minimizer of (3). We will see in Section 4 that the starting value for gradient descent may affect the target of estimation among the unidentified class $\mathcal{T}(Z)$ of latent processes defined by (2).

We propose a gradient descent algorithm which estimates the $d$ latent dimensions concurrently. Our concurrent gradient descent algorithm takes initial coordinates

$$\widehat{\mathcal{W}}^0 = \{\widehat{\boldsymbol{W}}_r^0\}_{r=1}^d,$$

step sizes $\eta_h > 0$, and a maximum number of iterations $H$ as inputs. In general, we shall allow the step size to depend on the iteration number $h \geq 0$. The output of Algorithm 1 is

**Algorithm 1:** Concurrent gradient descent algorithm.

For $h = 1$ to $h = H$

For $r = 1$ to $r = d$

$\widehat{\boldsymbol{W}}_r^h \leftarrow \widehat{\boldsymbol{W}}_r^{h-1} - \eta_h \frac{\partial \ell}{\partial \boldsymbol{W}_r}(\widehat{\mathcal{W}}^{h-1})$

Output $\widehat{\mathcal{W}}^H = \{\widehat{\boldsymbol{W}}_r^H\}_{r=1}^d$

an $n \times q \times d$ tensor-valued coordinate estimator $\widehat{\mathcal{W}}^H$. In practice, we choose $\eta_h$ according to a backtracking search scheme. We start from fixing the maximum step size, typically $\bar{\eta} = 1/nm$, and try one gradient descent step with size $\bar{\eta}$. If the objective decreases we accept it and continue, otherwise we try the original step again with size $\bar{\eta}/2$. This is repeated until we find a step size that decreases the objective. In the next iteration, we begin with the step size accepted in the previous iteration, and repeat.

Based on an estimator $\widehat{\mathcal{W}}^H$ found using our gradient descent scheme, we can use the function basis $B$ to convert back to an estimate of the unknown latent processes. For $x \in \mathcal{X}$, define the $n \times d$ matrix

$$\widehat{Z}^H(x) = \widehat{\mathcal{W}}^H \bar{\times}_2 B(x), \tag{4}$$

where we recall that the tensor-vector product operator $\bar{\times}_2$ takes a weighted sum of the $n \times d$ slices of $\widehat{\mathcal{W}}^H$ and the elements of $B(x)$ along the second mode, corresponding to the index space. We will refer to the estimator $\widehat{Z}^H$, or simply $\widehat{Z}$, as a functional adjacency spectral embedding (FASE).

The FASE estimator, unlike many other approaches to dynamic networks, can easily produce an estimate of the unknown latent processes at a value of $x$ where no snapshot is observed. While the function basis modeling approach taken here is not well suited to forecasting (extrapolation), it should do well for predicting a snapshot at a new index $x$ in the interior of the index space (interpolation). We evaluate the performance of FASE on this task and compare it to some competing approaches in Appendix C.2 of the supplementary materials. FASE can also be easily adapted to the case where some edge variables are missing from some snapshots, similar to MacDonald et al. (2022), by ignoring those triples in the calculation of the least squares objective (3). FASE is implemented in an R package `fase` available on CRAN.

### 3.1 Initializing Gradient Descent

As most iterative methods, FASE relies on a suitable initial value. In this section, we develop a principled initialization approach based on local averaging of network snapshots.

We start with the formal definition of adjacency spectral embedding (ASE Tang et al., 2013). For an $n \times n$ symmetric matrix $M$ with eigendecomposition $Y \Lambda Y^\intercal$, define

$$\mathrm{ASE}_d(M) = Y_d \Lambda_d^{1/2} \in \mathbb{R}^{n \times d},$$

where $Y_d \in \mathbb{R}^{n\times d}$ and $\Lambda_d \in \mathbb{R}^{d\times d}$ correspond to the first $d$ eigenvectors and eigenvalues of $M$, ordered according to the absolute values of the diagonal entries of $\Lambda$. If the diagonal entries of $\Lambda$ are distinct, then the ASE is uniquely defined up to sign flips of each column.

Under the conditions of the FASE model, suppose the sets $T_\ell$ for $\ell = 1, \ldots, L$ form a partition of $\{1, \ldots, m\}$ into $L$ contiguous groups of approximately equal size. We will use this decomposition of the index set to construct our initializer for the unknown coordinates. First we embed a local mean adjacency matrix using the adjacency spectral embedding, and define, for each $\ell = 1, \ldots, L$,

$$\widehat{Z}^0_\ell = \mathrm{ASE}_d\left(\frac{1}{|T_\ell|}\sum_{k\in T_\ell} A_k\right).$$

Then, for each $\ell = 2, \ldots, L$, we perform an additional alignment step which orthogonally transforms the columns of each embedding to minimize the discrepancy with the previous embedding, as measured by the Frobenius norm. This helps produce an initial smooth set of processes, targeting a smooth representative of the unidentified class $\mathcal{T}(Z)$ and increasing the potential to share information across network snapshots.

Then we set these embeddings as a piecewise constant estimator of the true processes, according to the partition $\{T_\ell\}_{\ell=1}^L$. For each $k = 1, \ldots, m$, define

$$\widehat{Z}^0(x_k) = \{\widehat{Z}_\ell : k \in T_\ell\},$$

and denote them together as an $n \times m \times d$ tensor $\widehat{\mathcal{Z}}^0$. As for the coordinate tensors, the first mode corresponds to nodes, the second to the index space now exactly corresponding to the snapshots, and the third mode to the latent space dimensions.

The initial coordinates for each $i = 1, \ldots, n$ and $r = 1, \ldots, d$ are given by the coordinates of the least squares solution in the corresponding spline basis. Let

$$\boldsymbol{B} = \begin{pmatrix} B(x_1) & \cdots & B(x_m)\end{pmatrix}^\intercal, \tag{5}$$

an $m \times q$ matrix which we will refer to as the $B$-spline *design matrix*. Then define

$$\widehat{\mathcal{W}}^0 = \widehat{\mathcal{Z}}^0 \times_2 (\boldsymbol{B}^\intercal\boldsymbol{B})^{-1}\boldsymbol{B}^\intercal, \tag{6}$$

where the tensor-matrix product $\times_2$ along the second index space mode maps each of the $m$-dimensional fibers of $\widehat{\mathcal{Z}}^0$ to their best fitting $q$-dimensional basis coordinates. This procedure is written formally in Algorithm 2.

In Section 4.2, we will identify conditions under which this estimator $\mathcal{W}^0$ is close enough to a corresponding target to provide a good initializer for gradient descent for FASE. We also use this result to formally motivate a choice of $L$, the number of sets in the partition.

### 3.2 Parameter Tuning

Up to this point, we have treated both the latent space dimension $d$, and function basis $B$ as fixed. In practice these will have to be selected based on data. To simplify this tuning problem, we consider only cubic $B$-spline bases with knots in $\mathcal{X}$ placed at equally spaced

**Algorithm 2:** Local average embedding initialization algorithm.

Partition $\{1, \ldots, m\}$ into $L$ contiguous subsets $\{T_\ell\}_\ell^L$

For $\ell = 1$ to $\ell = L$

Set $\widehat{Z}_\ell^0 = \text{ASE}_d\left(\frac{1}{|T_\ell|}\sum_{k=1}^m A_k\right)$

**Optional:** If $\ell > 1$ then

Set $Q_\ell = \text{argmin}_{Q \in \mathcal{O}_d} \|\widehat{Z}_\ell^0 Q - \widehat{Z}_{\ell-1}^0\|_F^2$

Update $\widehat{Z}_\ell^0 \leftarrow \widehat{Z}_\ell^0 Q_\ell$

For $k = 1, \ldots, m$

Set $\widehat{Z}^0(x_k) = \{\widehat{Z}_\ell^0 : k \in T_\ell\}$

Set $\widehat{\mathcal{Z}}^0 = \left\{\widehat{Z}^0(x_k)\right\}_{k=1}^m \in \mathbb{R}^{n \times m \times d}$

Set $\widehat{\mathcal{W}}^0 = \widehat{\mathcal{Z}}^0 \times_2 (\boldsymbol{B}^\intercal \boldsymbol{B})^{-1} \boldsymbol{B}^\intercal$

Output $\widehat{\mathcal{W}}^0$

quantiles of the snapshot indices $\{x_1, \ldots, x_m\}$. Thus, basis selection is fully determined by an integer $q$ which is a function of the number of knots. The parameter $d$ controls static complexity: flexibility of the network mean structure for each fixed $x \in \mathcal{X}$, while the parameter $q$ controls functional complexity: flexibility of each latent process as a function of the index.

To select these two tuning parameters, we derive a penalized least squares criterion based on the total squared error (3) for the observed network snapshots, and the number of parameters ($nqd$) used to define the latent process model. The details of this derivation are given in Appendix B of the supplementary materials.

In short, the objective (3) can be decomposed as the sum of squared residuals for $2n$ linear regression problems, comprising the incoming and outgoing edge values for each node. Each of these problems is based on $nm/2$ independent observations, and $qd$ unknown coefficients. We evaluate the generalized cross validation (GCV Golub et al., 1979) criterion for each problem and take the mean, resulting in an overall *network GCV* (NGCV) criterion equivalent to

$$\text{NGCV}(q, d) = \log\left\{\frac{\ell(\widehat{\mathcal{W}})}{mn^2}\right\} - 2\log\left(1 - \frac{2qd}{nm}\right) \tag{7}$$

for an estimated $n \times q \times d$ tensor $\widehat{\mathcal{W}}$ of basis coordinates. Then, parameter tuning will proceed by minimizing NGCV over a grid $\{(q, d) : q_{\min} \leq q \leq q_{\max}, \quad 1 \leq d \leq d_{\max}\}$, with user specified lower and upper bounds on each parameter.

While minimization of NGCV only requires the model to be fit once for each $(q, d)$ pair, this can still be computationally costly for large $m$ and $n$. For comparison, we also propose

a more efficient heuristic approach to optimizing over the grid using coordinate descent. In the coordinate descent scheme, we initialize $d = 1$, and perform alternating minimization for $q$ and $d$ by evaluating NGCV on a grid and treating the other as fixed.

We evaluate our tuning approach on synthetic data in Section 5.2. It appears to consistently recover the ground truth parameters when the signal-to-noise ratio is sufficiently high. In other cases, it still tends to select parameters with good performance in terms of recovery of the latent processes.

## 4. Theoretical Guarantees

In this section, we establish theoretical guarantees for the error of the gradient descent estimators of the true latent processes, averaged over nodes and snapshots, . All proofs are provided in Appendix A of the supplementary materials. As in Section 2, denote the true latent processes by $Z$, an $n \times d$ matrix-valued function of $\mathcal{X}$. The latent space dimension $d$ will be treated as fixed. Suppose that after centering, the edge distribution is sub-Gaussian with parameter $\sigma$, and for simplicity assume that the networks are undirected and self loops are allowed.

We also define some notation for the important spectral quantities related to the true latent processes. Define

$$\gamma_Z^2 = \min_{k=1,\dots,m} \left[\lambda_{\min}\left\{Z(x_k)^\intercal Z(x_k)\right\}\right],$$

and

$$\kappa = \gamma_Z^{-2}\left\{\max_{k=1,\dots,m}\left(\lambda_{\max}\left\{Z(x_k)^\intercal Z(x_k)\right\}\right)\right\} \geq 1. \tag{8}$$

Throughout this section, constants may depend on $d$ or $\kappa$, which are treated as fixed, but will always be free of $n$, $m$, $q$, $\sigma$, and $\gamma_Z$ which will be tracked in the final bounds. Estimation will proceed by Algorithm 1 on $\ell(\mathcal{W})$ with fixed $q$-dimensional $B$-spline basis $B$, and design matrix $\boldsymbol{B}$ given by (5). Throughout the section, we will make the following assumptions on the $B$-spline basis and associated design matrix.

**Assumption 1**

*(A) Assume that for each $k$, $B(x_k) \geq 0$ element-wise, $\|B(x_k)\|_1 = 1$, and $B(x_k)$ has at most $2D + 1$ nonzero entries.*

*(B) Assume that $\boldsymbol{B}$ satisfies*

$$\frac{c_B m}{q} \leq \lambda_{\min}(\boldsymbol{B}^\intercal \boldsymbol{B}) \leq \lambda_{\max}(\boldsymbol{B}^\intercal \boldsymbol{B}) \leq \frac{C_B m}{q}$$

*for constants $C_B > c_B > 0$.*

Part (A) is satisfied by $B$-spline bases of fixed order $D \geq 0$. Moreover, because $B$-splines are locally supported, if the snapshot indices and basis knots are evenly spread across $\mathcal{X}$, it follows that $\mathrm{tr}(\boldsymbol{B}^\intercal \boldsymbol{B}) \sim m$. Thus part (B) simply requires that $\boldsymbol{B}^\intercal \boldsymbol{B}$ has a condition number of a constant order, a condition which holds if the support of every basis function contains on the order of $m/q$ snapshot indices, for growing $m$ and $q$.

### 4.1 Results for Algorithm 1

In this subsection, we will establish an asymptotic bound on the error of the FASE estimator as computed by Algorithm 1, up to an unknown orthogonal transformation. Recall that in general we can only identify an unknown representative of $\mathcal{T}(Z)$. Thus, as in single layer latent space approaches (Ma et al., 2020), for a given iteration $h$ of gradient descent, there may be a different representative of $\mathcal{T}(Z)$ which minimizes the Frobenius norm error over the unidentified class.

In the following, we will only need the true processes evaluated at the snapshot indices, so we store these in an $n \times m \times d$ tensor $\mathcal{Z}$ with $n \times d$ slices given by $Z(x_k)$ for $k = 1, \ldots, m$. We also define the unidentified class of process snapshots by

$$\mathcal{T}^m(\mathcal{Z}) = \{Z(x_k)Q_k : Q_k \in \mathcal{O}_d, k = 1, \ldots, m\} \subset \mathbb{R}^{n \times m \times d}.$$

These process snapshots produce the same expected adjacency matrix snapshots as the original $\mathcal{Z}$ and are thus indistinguishable with respect to the least squares objective (3). As in Section 3.1, the first mode corresponds to nodes, the second to the index space, and the third mode to the latent space dimensions.

To prove a result about Algorithm 1, we must control the intrinsic approximation error introduced by restricting our estimates to a finite-dimensional function basis. This error will depend on $\mathcal{T}^m(\mathcal{Z})$ and the functions in $\text{span}(B)$, as well as, due to nonidentifiability of $\mathcal{Z}$, on the current iterate of gradient descent.

Define a map from a set of coordinates to the space of snapshots,

$$\mathcal{R}_{\boldsymbol{B},\mathcal{Z}}(W) = \text{argmin}_{Z \in \mathcal{T}^m(\mathcal{Z})} \|W - Z \times_2 (\boldsymbol{B}^\intercal \boldsymbol{B})^{-1} \boldsymbol{B}^\intercal\|_F^2.$$

In words, $\mathcal{R}_{\boldsymbol{B},\mathcal{Z}}$ takes a set of coordinates and finds the best-aligned orthogonal transformation of the true processes. The tensor product in the second term inside the norm can be interpreted as finding a least squares solution with respect to the $B$-spline design matrix, as in (6).

Then $\mathcal{R}_{\boldsymbol{B},\mathcal{Z}}(\widehat{\mathcal{W}}^h)$ is an $n \times m \times d$ tensor which we call the *snapshot target* at iteration $h$; it is an element of $\mathcal{T}^m(\mathcal{Z})$, so for each $k = 1, \ldots, m$ and $h \geq 0$, we can find $Q_k^{*,h} \in \mathcal{O}_d$ such that $Z(x_k)Q_k^{*,h}$ is the $k$th $n \times d$ slice of $\mathcal{R}_{\boldsymbol{B},\mathcal{Z}}(\widehat{W}^h)$. We also define

$$\mathcal{W}^{*,h} = \mathcal{R}_{\boldsymbol{B},\mathcal{Z}}(\widehat{\mathcal{W}}^h) \times_2 (\boldsymbol{B}^\intercal \boldsymbol{B})^{-1} \boldsymbol{B}^\intercal \in \mathbb{R}^{n \times q \times d}, \tag{9}$$

which we call the *coordinate target* at iteration $h$. Control of the approximation error at each iteration will naturally depend on how well the coordinate target approximates the snapshot target, through

$$\varepsilon_{\text{approx},2}^{(h)} = \frac{1}{m} \sum_{k=1}^{m} \|\mathcal{W}^{*,h} \bar{\times}_2 B(x_k) - Z(x_k)Q_k^{*,h}\|_F^2$$

$$\varepsilon_{\text{approx},\infty}^{(h)} = \max_{k=1,\ldots,m} \|\mathcal{W}^{*,h} \bar{\times}_2 B(x_k) - Z(x_k)Q_k^{*,h}\|_F^2$$

for $h \geq 0$, which describe average and maximum approximation errors over the snapshot times, respectively. In the case of $B$-splines, the approximation errors will be small if the target in snapshot space is made up of latent processes which are smooth in $x$ (Schumaker, 2007).

Our main result will require that the approximation errors are controlled uniformly over $h$ with high probability. While this condition cannot be verified, empirically we find these quantities do not tend to increase in $h$, and thus if the initializer induces a sufficiently smooth snapshot target, so too will the gradient descent iterates.

With these definitions in hand, we are now ready to state the assumptions required for our main result. First, a condition on a properly scaled signal-to-noise ratio.

**Assumption 2**

$$\frac{\sigma^2 q^5 n \log q}{m \gamma_Z^4} = o(1).$$

Additionally, we require asymptotic conditions on the approximation and initialization errors.

**Assumption 3**

$$\sup_{h \geq 0} \varepsilon^{(h)}_{\text{approx},2} = o_{\mathbb{P}}(\gamma_Z^2/q)$$
$$\sup_{h \geq 0} \varepsilon^{(h)}_{\text{approx},\infty} = o_{\mathbb{P}}(\gamma_Z^2)$$

**Assumption 4**

$$\|\widehat{\mathcal{W}}^0 - \mathcal{W}^{*,0}\|_F^2 = o_{\mathbb{P}}(\gamma_Z^2)$$

All three of these assumptions appear in the proof to ensure a contraction property of the iterates in coordinate space, in particular that the discrepancy

$$\|\widehat{\mathcal{W}}^h - \mathcal{W}^{*,h}\|_F^2$$

between the current estimate and the current coordinate target shrinks as $h$ increases. Empirically, we see that gradient descent does indeed converge as $h$ increases, but in theory this contraction does not guarantee convergence of $\widehat{\mathcal{W}}^h$. Hence, our result is written in terms of a lim sup, which exists even if gradient descent does not converge.

Assumption 3 puts a requirement on the approximation errors, and Assumption 4 puts a requirement on the initialization error. In the simplest case for a basis of dimension $q = 1$, the latent processes are modeled as constant in $x$. Then, $\varepsilon^{(h)}_{\text{approx},2}$ is the total sample variance of the snapshot target processes around their means, and we require that this is asymptotically smaller than the squared magnitude of the process. For larger $q$, we need additional parameters to proportionally reduce this variation. Similarly, we require that the maximum squared discrepancy over $k = 1, \ldots, m$ is smaller than the squared magnitude of the process. Finally, Assumption 4 requires that as $n$ increases, the initialization error is small compared to the magnitude of the processes. Validity of this assumption is addressed in Section 4.2.

With these assumptions, we are ready to state the main result of this section. Recall that in this asymptotic regime, we have $n \to \infty$, and allow $m$, $q$, $\gamma_Z$, and $\sigma$ to possibly grow as functions of $n$.

**Theorem 2** *Suppose $\{A_k\}_{k=1}^m$ are generated from a latent process network model, with independent sub-Gaussian edges with parameter $\sigma$. Suppose we compute a FASE estimator using Algorithm 1 with $q$-dimensional $B$-spline basis $B$ and step size $\eta_h \equiv \eta' q/m\gamma_Z^2$ for a constant $\eta'$. Suppose Assumptions 1, 2, 3, and 4 hold. Then if the sequence of average approximation errors $\varepsilon_{\text{approx},2}^{(h)}$ satisfies*

$$\limsup_{h\to\infty} \varepsilon_{\text{approx},2}^{(h)} = O_{\mathbb{P}}(\alpha_n), \tag{10}$$

*it follows that the estimation errors satisfy*

$$\limsup_{h\to\infty} \frac{1}{mn}\sum_{k=1}^{m} \|\widehat{Z}^h(x_k) - Z(x_k)Q_k^{*,h}\|_F^2 = O_{\mathbb{P}}\left(\frac{\sigma^2 q^4 \log q}{\gamma_Z^2 m} + \frac{\alpha_n}{n}\right), \tag{11}$$

*where $Z(x_k)Q_k^{*,h}$ is the $k$th $n \times d$ slice of the snapshot target at iteration $h$, and $\widehat{Z}^h(x_k)$ is defined in* (4).

The left hand side of (11) is an upper bound on the error of the limiting FASE estimator output by Algorithm 1, averaged over the snapshot indices and nodes. The right hand side of (11) can be interpreted as a statistical error term and an approximation bias term. The constant $\eta'$ is derived explicitly in the proof, and depends on the problem dimension, basis design, and spectral properties of $Z$. In practice we do not attempt to use this step size, instead selecting it adaptively as described in Section 3.

In this setting, assuming that each latent process is in span($B$) will not help establish consistency, as in general this assumption will no longer hold after applying the unknown orthogonal transformation, leading to nonzero approximation bias. However, in the limit the average approximation bias may be bounded above by $\limsup_{h\to\infty} \varepsilon_{\text{approx},2}^{(h)}$. Thus (11) shows that the error in estimating the unknown latent processes inherits the rate $\alpha_n$ from the intrisic basis approximation error. Based on the normalization of $\varepsilon_{\text{approx},2}^{(h)}$, we expect $\alpha_n/n$ to be approximately constant in $n$, $m$, and $\sigma$; but decrease in $q$ as the function basis becomes more flexible. Hence, there should be an optimal choice of $q$ for which the right hand side of (11) goes to zero asymptotically, and the FASE estimator is consistent. To better understand this tradeoff, we state a more interpretable result in a special case where $d = 1$.

When $d = 1$, the unknown rotations at each $x_k$ reduce to sign flips, and we can show that when the initialization error is small, gradient descent will contract towards a special, deterministic coordinate target. Define

$$\boldsymbol{W}_1^* = \text{argmin}_{W\in\mathbb{R}^{n\times q}} \sum_{k=1}^{m} \|WB(x_k) - Z_1(x_k)\|_2^2$$

and state slightly altered assumptions on the approximation and initialization error in terms of this new $\boldsymbol{W}_1^*$.

**Assumption 5**

$$\frac{1}{m}\sum_{k=1}^{m}\|\boldsymbol{W}_1^* B(x_k) - Z_1(x_k)\|_2^2 = o(\gamma_Z^2/q),$$
$$\max_{k=1,\ldots,m}\|\boldsymbol{W}_1^* B(x_k) - Z_1(x_k)\|_2^2 = o(\gamma_Z^2).$$

**Assumption 6**

$$\|\widehat{\boldsymbol{W}}_1^0 - \boldsymbol{W}_1^*\|_F^2 = o_{\mathbb{P}}(\gamma_Z^2).$$

Then we have the following stronger result, stated as a corollary of Theorem 2.

**Corollary 3** *Suppose $\{A_k\}_{k=1}^m$ are generated from a latent process network model with $d = 1$, and independent sub-Gaussian edges with parameter $\sigma$. Suppose we compute a FASE estimator using Algorithm 1 with $q$-dimensional $B$-spline basis $B$ and step size $\eta_h \equiv \eta'' q/m\gamma_Z^2$ for a constant $\eta''$. Suppose Assumptions 1, 2, 5, and 6 hold. Then if the approximation error satisfies*

$$\frac{1}{m}\sum_{k=1}^{m}\|\boldsymbol{W}_1^* B(x_k) - Z_1(x_k)\|_2^2 = O(\alpha_n'), \tag{12}$$

*it follows that the estimation error satisfies*

$$\limsup_{h\to\infty}\frac{1}{mn}\sum_{k=1}^{m}\|\widehat{Z}_1^h(x_k) - Z_1(x_k)\|_F^2 = O_{\mathbb{P}}\left(\frac{\sigma^2 q^4 \log q}{\gamma_Z^2 m} + \frac{\alpha_n'}{n}\right), \tag{13}$$

*where $\widehat{Z}_1^h(x_k)$ is defined in* (4)*.*

If we make a parametric assumption that each orthogonalized latent process is in span$(B)$, we have $\boldsymbol{W}_1^* B(x_k) = Z_1(x_k)$ for all $k = 1, \ldots, m$, so that $\alpha_n' = 0$, and get a consistent estimator on average over the nodes and snapshot indices. In the corresponding nonparametric setting, suppose that each component function $z_{i,1}$ is twice differentiable. Then by approximation results for cubic $B$-splines (Schumaker, 2007), there exists $w \in \mathbb{R}^q$ such that

$$\sup_{x\in\mathcal{X}}|z_{i,1}(x) - w^\intercal B(x)| \quad \lesssim \frac{1}{q^2}\cdot\sup_{x\in\mathcal{X}}\left|\frac{\partial^2 z_{i,1}(x)}{\partial x^2}\right|.$$

Thus, if the true processes have uniformly bounded second derivatives, we have $\alpha_n' \lesssim n/q^2$, and there exists a theoretically optimal $q$ which grows with $m$ and $n$ such that the FASE estimator is consistent for $Z_1$ on average over the nodes and snapshot indices.

This guarantee on the alignment of the true and estimated latent processes for $d = 1$ motivates a sequential estimator of FASE for higher-dimensional models, which estimates one latent dimension at a time. We describe this approach in Appendix F of the supplementary materials, and implement it in the R package `fase`. However, the sequential estimator relies on strong assumptions on separation of the singular values of each $Z(x)$ (uniformly in $x$), so in general we recommend the concurrent gradient descent estimator for FASE presented in Section 3, which we use for all the simulation and real data results to follow.

### 4.2 Results for Initialization

In this section we state the main result for our proposed initializer, a high probability upper bound for recovery of the initial coordinate target, as defined in Section 4.1. Then, we derive an optimal choice for $L$, the number of sets in the partition of the index set, as defined in Section 3.1.

For simplicity, in this section we assume the snapshot indices are equally spaced on $\mathcal{X} = [0, 1]$, that is, $x_k = k/m$ for $k = 1, \ldots, m$, and $T_\ell$ splits the indices into equal sized contiguous subsets. Supposing $m/L$ is an integer, $|T_\ell| = m/L$ for all $\ell = 1, \ldots, L$, and two indices in the same $T_\ell$ are separated by at most $1/L$ in the index space.

We also assume a Lipschitz condition on the latent processes $z_{i,r}(x)$ as a function of $x$.

**Assumption 7** *Suppose that for $i = 1, \ldots, n$ and $r = 1, \ldots, d$, each latent process $z_{i,r}(x)$ satisfies*

$$\sup_{x,y \in \mathcal{X}} |z_{i,r}(x) - z_{i,r}(y)| \leq K_1 |x - y|$$

*for a uniform a constant $K_1 > 0$.*

Note that this condition need only hold for a particular orthogonal transformation of the latent processes, not for any element of the unidentified class $\mathcal{T}(Z)$.

With these assumptions, we can state the main result of this section.

**Proposition 4** *Suppose $\{A_k\}_{k=1}^m$ are generated from a latent process network model, with independent sub-Gaussian edges with parameter $\sigma$. Define the initializer $\widehat{\mathcal{W}}^0$ as in (6), with parameter $L$. Suppose Assumption 7 holds. Then the initializer $\widehat{\mathcal{W}}^0$ satisfies*

$$\|\widehat{\mathcal{W}}^0 - \mathcal{W}^{*,0}\|_F^2 \leq \left\{ \frac{C_B q}{c_B^2} \left( \frac{(2\sqrt{10d} + 1) K_1 \sqrt{dn}}{\gamma_Z L} + \frac{c_{\text{prob}} \sigma \sqrt{10 d L n}}{\gamma_Z^2 \sqrt{m}} \right)^2 \right\} \gamma_Z^2$$

*with probability at least $1 - 4L \exp(-n)$, where $\mathcal{W}^{*,0}$ is the associated coordinate target for $\widehat{\mathcal{W}}^0$, as defined in (9), and $c_{\text{prob}}$ is a universal constant.*

Towards justifying the asymptotic rate in Assumption 4, we derive an optimal choice of $L$ (up to constant factors) and analyze the resulting effect on initialization error. Treating $C_B$, $c_B$, $K_1$, and $d$ as constants, we have, with high probability,

$$\frac{1}{\gamma_Z} \|\widehat{\mathcal{W}}^0 - \mathcal{W}^{*,0}\|_F \lesssim \frac{\sqrt{qn}}{L \gamma_Z} + \frac{\sigma \sqrt{qLn}}{\gamma_Z^2 \sqrt{m}}.$$

Some algebra shows that the first two terms of the upper bound are minimized for

$$\hat{L} \sim \left( \frac{\gamma_Z \sqrt{m}}{\sigma} \right)^{2/3},$$

and plugging this in gives optimized upper bound

$$\frac{1}{\gamma_Z} \|\widehat{\mathcal{W}}^0 - \mathcal{W}^{*,0}\|_F \lesssim \frac{\sigma^{2/3} (qn)^{1/2}}{m^{1/3} \gamma_Z^{5/3}}.$$

If this upper bound on relative error goes to zero, then the initialization assumption will hold asymptotically with high probability.

## 5. Evaluation on Synthetic Networks

In this section, we will evaluate FASE against competing methods for functional or dynamic network embedding, by applying them to simulated functional networks.

### 5.1 Latent Process Recovery

First, we compare our FASE estimator to existing methods for similar inner product latent space models for both weighted and binary edge networks, in terms of recovery of the latent processes snapshots. We will implement FASE (Algorithm 1) with both an oracle and an adaptive NGCV tuning scheme, as well as three ASE-based approaches that have been applied in the past to functional, in particular dynamic, network data (Sanna Passino et al., 2021). First, we apply the usual $d$-dimensional ASE to each of the $m$ adjacency matrix snapshots. Second, we apply the omnibus ASE (OMNI) (Levin et al., 2017), which finds a $d$-dimensional embedding at each snapshot index based on the ASE of the so-called *omnibus* matrix given by

$$\begin{pmatrix} A_1 & \frac{A_1+A_2}{2} & \cdots & \frac{A_1+A_m}{2} \\ \frac{A_1+A_2}{2} & A_2 & & \vdots \\ \vdots & & \ddots & \\ \frac{A_1+A_m}{2} & \cdots & & A_m \end{pmatrix}.$$

Third, we apply the multiple ASE (COSIE) (Arroyo and Athreya, 2021), which assumes that the expectations of the adjacency matrix snapshots share a common invariant subspace. For these baseline estimators, we assume oracle knowledge of $d$.

We generate instances of the latent process network model under three scenarios.

(i) Parametric Gaussian network with $B$-spline processes. In this scenario we generate each component process $z_{i,r}$ for $i = 1, \ldots, n$ and $r = 1, \ldots, d$ from a cubic $B$-spline basis $B$ on $[0, 1]$, with equally spaced knots and dimension 10. In particular we generate $\boldsymbol{w}_{i,r} \sim \mathcal{N}_{10}(0, I_{10})$, and then define $z_{i,r}(x) = \boldsymbol{w}_{i,r}^\intercal B(x)$ for $x \in [0, 1]$. Then for equally spaced snapshot indices $0 = x_1 < \cdots < x_m = 1$, we set

$$A_k = Z(x_k)Z(x_k)^\intercal + E_k,$$

where each $E_k$ is a symmetric matrix of independent Gaussian random variables with variance $\sigma^2$.

(ii) Nonparametric Gaussian network with sinusoidal processes. In this scenario we generate each component process as

$$z_{i,r}(x) = \frac{3\sin[2\pi(2x - U_{i,r})]}{1 + 5[x + B_{i,r}(1 - 2x)]} + G_{i,r}$$

where $U_{i,r} \overset{\text{iid}}{\sim} \text{Unif}[0,1]$, $B_{i,r} \overset{\text{iid}}{\sim} \text{Bernoulli}(1/2)$, and $G_{i,r} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/4)$. Each process is a shifted (according to $G$) sine function which goes through 2 full cycles from a random starting point (controlled by $U$), with amplitude either increasing or decreasing (controlled by $B$) from 3 to 1/2 or 1/2 to 3. Then for equally spaced snapshot indices $0 = x_1 < \cdots < x_m = 1$, we set

$$A_k = Z(x_k)Z(x_k)^\intercal + E_k,$$

where each $E_k$ is a symmetric matrix of independent Gaussian random variables with variance $\sigma^2$.

(iii) Parametric RDPG network with $B$-spline processes. In this scenario we generate each component process $z_{i,r}$ for $i = 1, \ldots, n$ and $r = 1, \ldots, d$ from a cubic $B$-spline basis $B$ on $[0,1]$, with equally spaced knots and dimension 10. We generate each $d$-dimensional fiber of the full $n \times 10 \times d$ coordinate tensor $\mathcal{W}$ as an independent Dirichlet random variable with parameter $[0.1 \cdots 0.1]^\intercal$. The coordinates are then rescaled to control the overall network density. Then for $i \leq j$, we generate

$$[A_k]_{ij} \sim \text{Bernoulli}\left\{\sum_{r=1}^{d} z_{i,r}(x_k) z_{j,r}(x_k)\right\}$$

and set $[A_k]_{ji}$ to make each $A_k$ symmetric.

To compare the performance of FASE against the baseline estimators, we evaluate error for recovery of the latent processes up to orthogonal transformation, averaged over the snapshot indices:

$$\text{Err}_Z(\widehat{Z}) = \left\{\frac{1}{ndm}\sum_{k=1}^{m} \min_{Q_k \in \mathcal{O}_d} \|\widehat{Z}(x_k) - Z(x_k)Q_k\|_F^2\right\}^{1/2}.$$

This error metric is similar to the error bounded in the conclusion of Theorem 2. If our adaptive implementation of FASE selects $d$ incorrectly, then either the true or estimated latent processes are given additional columns of all zeros so that the dimensions match. We report two errors for the FASE estimator found using Algorithm 1, one for the adaptive version which selects $d$ and $q$ using a grid search with candidates $q = 6, 8, \ldots, 16$ and $d = 1, 2, \ldots, 6$, and the NGCV criterion defined in Section 3.2 (FASE (NGCV)); and another oracle FASE estimator which fits the model with ground truth $d$, and $q$ selected to minimize $\text{Err}_Z$ (FASE (ORC)). Implementation details for these two dimension selection procedures are given in Section 5.2). In all of the following plots, vertical lines at each point denote plus and minus 2 standard errors over the independent replications.

In Figures 1 and 2, we report results for scenario (i) generated with $\sigma \in \{2, 4, 6, 8\}$. In Figure 1 we vary the number of snapshots $m \in \{20, 40, \ldots, 200\}$ for fixed $n = 100$ and $d = 2$, and in Figure 2 we vary the number of nodes $n \in \{80, 120, \ldots, 400\}$ for fixed $m = 80$ and $d = 2$. In all settings, $\text{Err}_Z$ is averaged over 50 independent replications. In Figure 1, in all four panels, none of the baseline ASE estimators show an improvement with increasing $m$, while FASE does. Note that for $\sigma = 4$ the plotted points for COSIE are not visible,

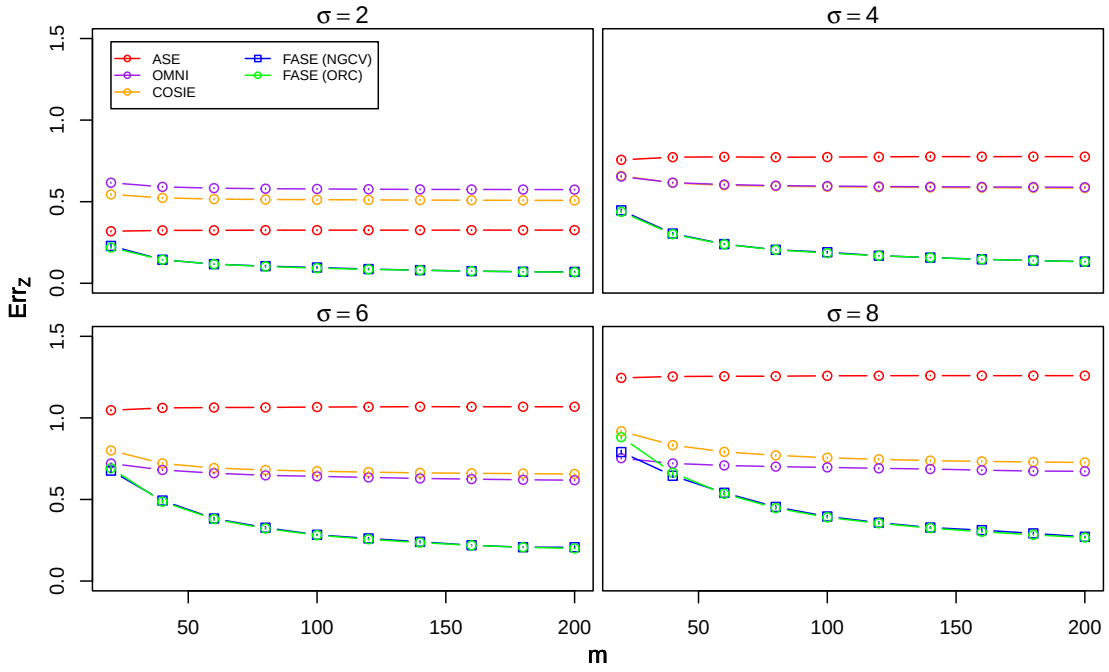


Figure 1: Mean of $\text{Err}_Z$, varying $m$, the number of snapshots. Scenario (i), parametric Gaussian networks. Plots are labeled by edge standard deviation $\sigma$.

as the performance coincides with that of OMNI. In Figure 2, only FASE and ASE show improvement with increasing $n$. While ASE improves at a faster rate than FASE, it never outperforms it, even for the largest values of $n$ considered. In almost all settings, FASE performs the best of all methods. This is true regardless of whether tune $d$ and $q$ adaptively, as the errors for FASE (NGCV) and FASE (ORC) are nearly indistinguishable in these plots. Among the baselines, the errors for ASE, which is unbiased, are by far the most sensitive to $\sigma$. On the other hand, COSIE and OMNI, which share information globally, incur a lot of bias in this setting, where the latent processes are only similar locally in the index variable. In Figure 1 for $\sigma = 8$ and $m = 20$, the adaptive FASE estimator is able to outperform the oracle on average. This phenomenon, which we discuss in more detail in Section 5.2, can occur when the signal is low and the adaptive estimator selects a value of $d$ which achieves better error than the ground truth $d$.

In Figures 3 and 4, we report results for scenario (ii) generated with $\sigma \in \{2, 4, 6, 8\}$. In Figure 3 we vary the number of snapshots $m \in \{20, 40, \ldots, 200\}$ for fixed $n = 100$ and $d = 2$, and in Figure 4 we vary the number of nodes $n \in \{80, 120, \ldots, 400\}$ for fixed $m = 80$ and $d = 2$. In all settings, $\text{Err}_Z$ is averaged over 50 independent replications. Results in these plots look similar to those in scenario (i), confirming that FASE is not relying on any parametric assumptions made on the true latent processes. In fact, even in the low signal to noise parameter settings for scenario (i) where OMNI outperformed FASE, FASE now outperforms all of its competitors. In Figure 4, for $\sigma = 2$ the errors for ASE are very close

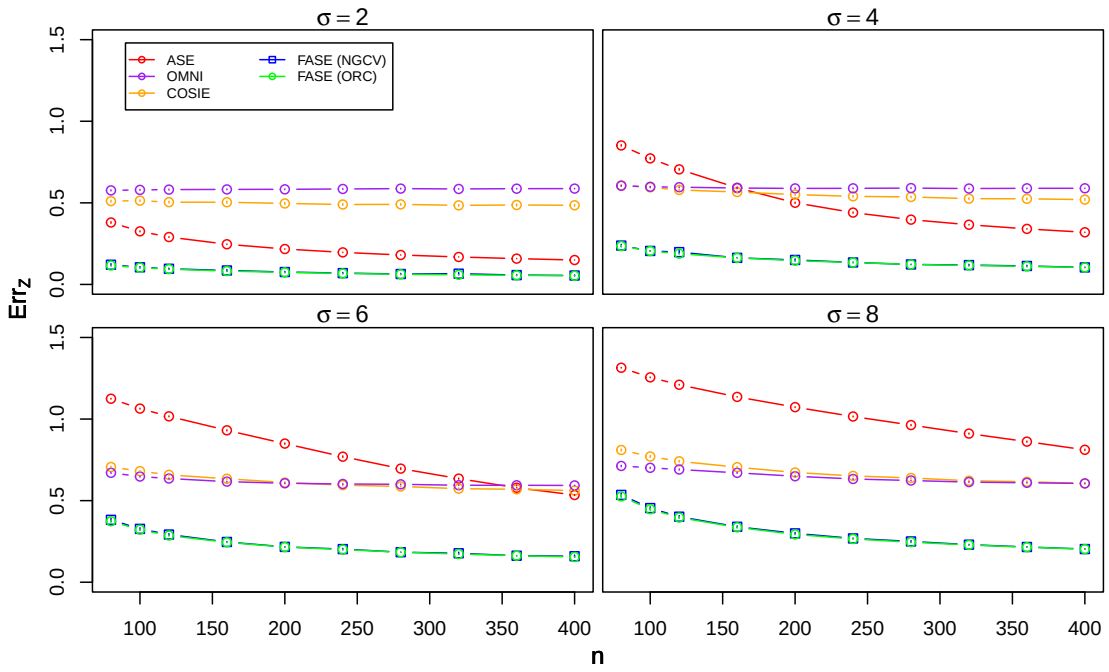


Figure 2: Mean of $\mathrm{Err}_Z$, varying $n$, the number of nodes. Scenario (i), parametric Gaussian networks. Plots are labeled by edge standard deviation $\sigma$.

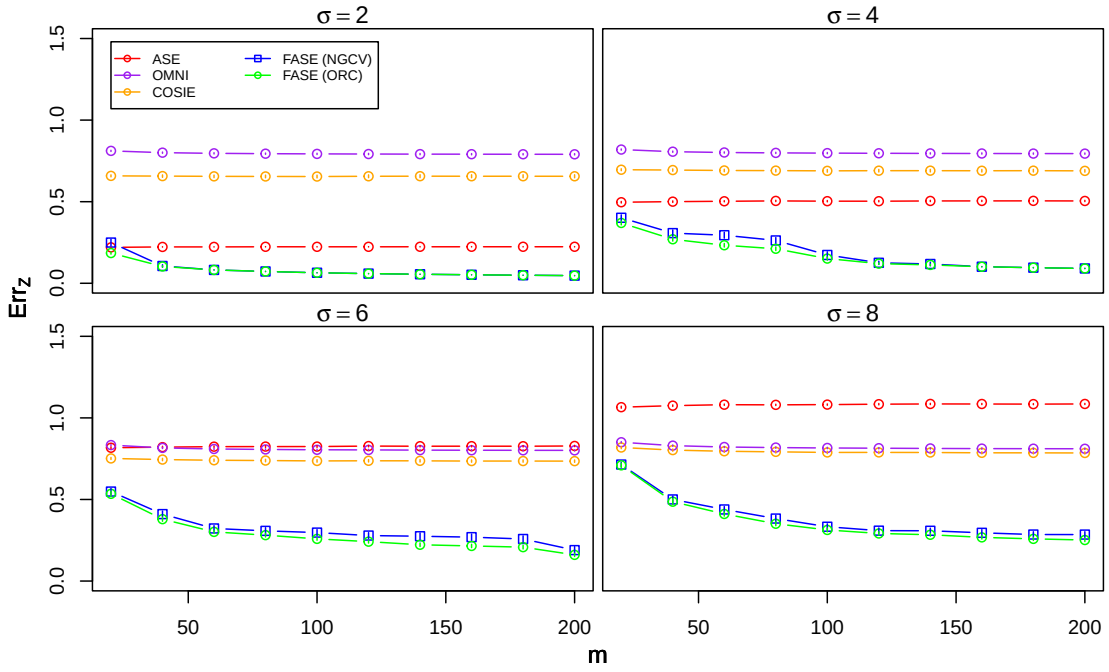


Figure 3: Mean of $\text{Err}_Z$, varying $m$, the number of snapshots. Scenario (ii), nonparametric Gaussian networks. Plots are labeled by edge standard deviation $\sigma$.

to those for FASE, but this is only because of the very high signal to noise ratio, and even in relative terms, the performance of ASE is comparatively worse as $\sigma$ increases.

In Figures 5 and 6, we report results for scenario (iii) generated with edge densities 0.1, 0.25 and 0.5. In Figure 5 we vary the number of snapshots $m \in \{20, 40, \ldots, 200\}$ for fixed $n = 100$ and $d = 2$, and in Figure 6 we vary the number of nodes $n \in \{80, 120, \ldots, 400\}$ for fixed $m = 80$ and $d = 2$. In all settings, $\text{Err}_Z$ is averaged over 50 independent replications. Once again, none of the baseline ASE estimators show an improvement with increasing $m$, while FASE does, and only ASE and FASE improve with increasing $n$. In all settings, FASE performs the best of all methods. For these RDPG networks, we see that while the more conservative COSIE and OMNI approaches improve as the density decreases, the unbiased ASE approach gets substantially worse, as the signal is decreasing. Similarly, FASE gets slightly worse for decreasing density, but still always outperforms COSIE and OMNI.

Finally, we report some brief results on convergence and runtime of our FASE estimator. In these simulations, we set the convergence criterion for Algorithm 1 to the relative decrease in the objective function dropping below $10^{-5}$. In 99% of replications this occurs in less than 200 iterations, and it always occurs in less than 600 iterations. As an example benchmark, fitting FASE with $n = m = 100$, $d = 2$ and $q = 10$ to data generated as in scenario (i) takes about 4 seconds on a computer with an Apple M2 Pro Chip and 16GB RAM.

Taken together, we see that among spectral embedding approaches for functional network data, FASE shows state of the art performance for recovery of the underlying latent

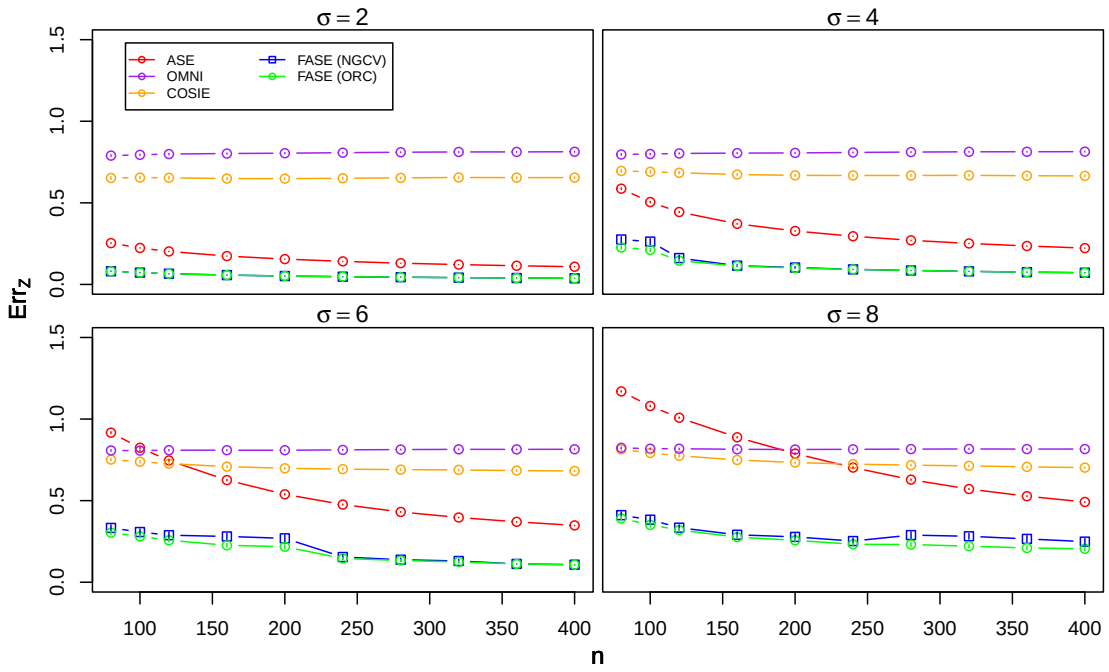


Figure 4: Mean of $\mathrm{Err}_Z$, varying $n$, the number of nodes. Scenario (ii), nonparametric Gaussian networks. Plots are labeled by edge standard deviation $\sigma$.

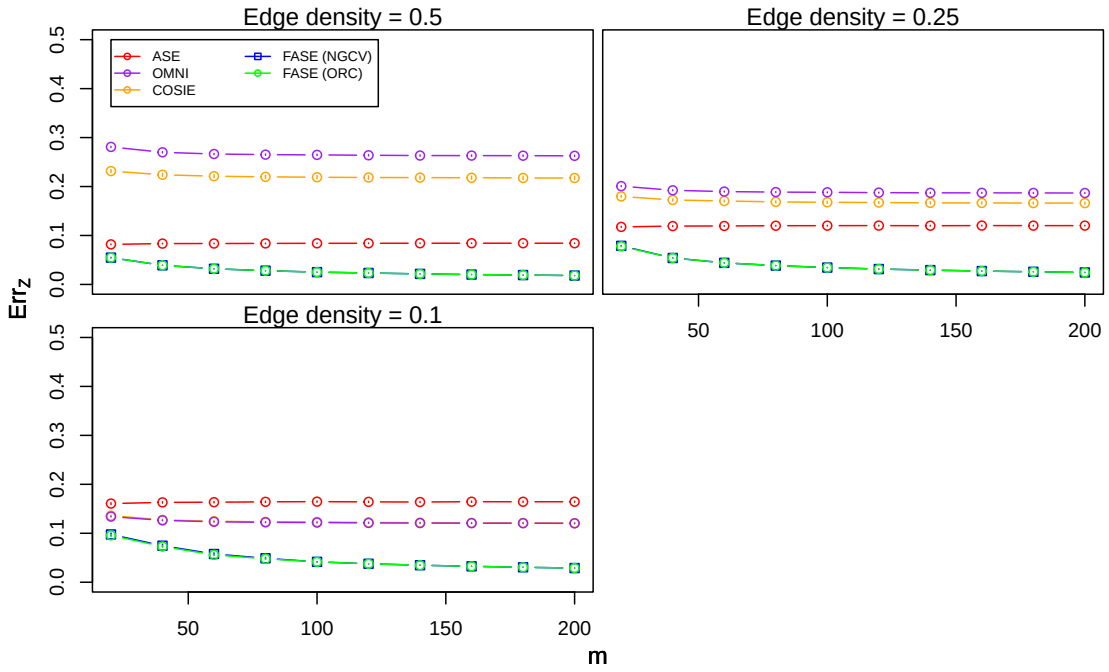


Figure 5: Mean of $\text{Err}_Z$, varying $m$, the number of snapshots. Scenario (iii), parametric RDPG networks. Plots are labeled by edge density.

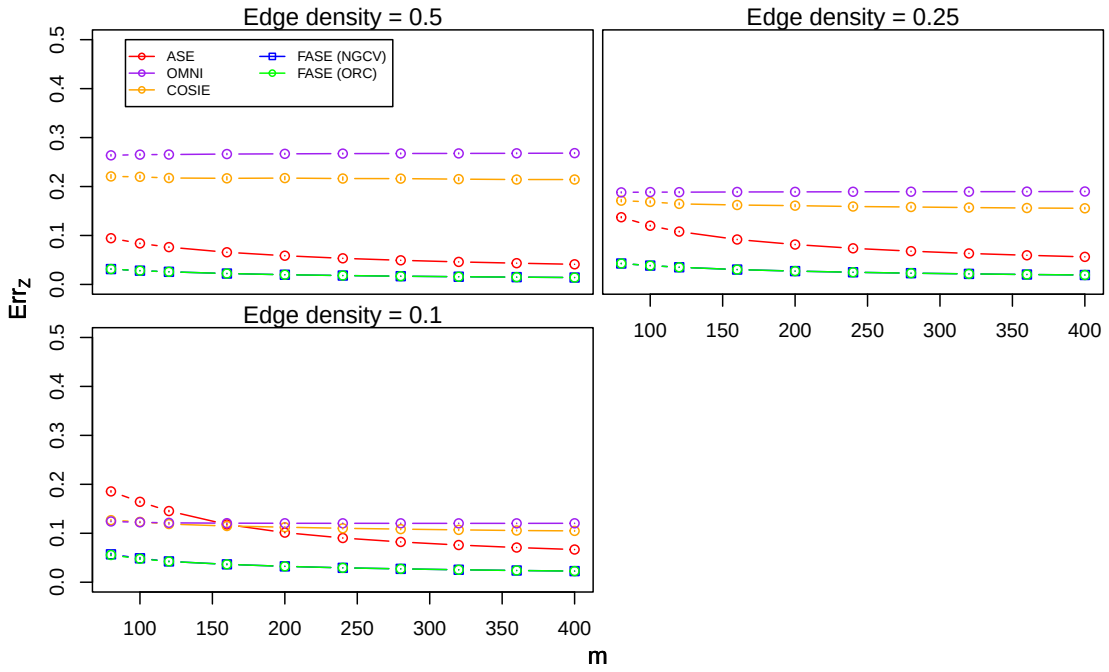


Figure 6: Mean of $\mathrm{Err}_Z$, varying $n$, the number of nodes. Scenario (iii), parametric RDPG networks. Plots are labeled by edge density.

process structure, up to unknown rotations. As suggested by Theorem 2, in both RDPG and Gaussian settings, we do not see the errors for FASE vanishing to zero, instead they appear to be bounded below by an intrinsic approximation error term. Even in scenarios (i) and (iii), when we generate the true latent processes from a $B$-spline basis, since we cannot necessarily remove the unknown orthogonal transformation, we essentially revert to a nonparametric regime, and benefit from the fact that FASE only requires smoothness of the latent processes for efficient recovery.

### 5.2 Tuning With the NGCV Criterion

We evaluate the performance of our new NGCV model selection criterion on latent process network models generated from scenarios (i), (ii), and (iii) as defined in Section 5.1. We compare the quality of selection and the quality of the eventual fitted model compared to an oracle.

To evaluate quality of selection of $d$, we see if our selected model matches the ground truth used to generate the model. To evaluate the quality of selection of $q$, as there is not always a ground truth parameter, we see if our selected model matches the oracle $q$, denoted by $q_{\text{ORC}}$, and which minimizes the process recovery error up to orthogonal transformation:

$$q_{\text{ORC}} = \underset{q_{\min} \le q' \le q_{\max}}{\operatorname{argmin}} \operatorname{Err}_Z\left(\widehat{Z}^{(q')}\right),$$

where $\widehat{Z}^{(q')}$ is a FASE estimator fit with latent space dimension $d$ and basis dimension $q'$. In Tables 1-3, we report the proportion of replications in which the NGCV selection matches the ground truth value for $d$ (d-Prop) and the proportion of replications in which it matches both values in the best pair $(q_{\text{ORC}}, d)$ (Prop). To evaluate the quality of the fitted model, Tables 1-3 also display the ratio between $\text{Err}_{\text{Z}}$ for the FASE estimator fit with $(\hat{q}, \hat{d})$ selected according to NGCV, and the FASE estimator fit with the best pair $(q_{\text{ORC}}, d)$ (Ratio).

As in Section 5.1, we consider three scenarios: (i) parametric Gaussian networks with $B$-spline latent processes, (ii) nonparametric Gaussian networks with sinusoidal latent processes, and (iii) parametric RDPG networks with $B$-spline latent processes. In all scenarios, we search for $(q, d)$ pairs over a $6 \times 6$ grid with $q = 6, 8, \ldots, 16$ and $d = 1, 2, \ldots, 6$. We will perform selection either by fitting models over the entire grid, or by coordinate descent (CD), as described in Section 3.2. Typically, coordinate descent converges in 3 or 4 univariate searches, meaning that it fits around 1/2 to 2/3 as many models compared to the full grid search over the $6 \times 6$ grid. The computational improvement of coordinate descent over a full grid search will be more pronounced for larger grids.

For scenario (i), we fix $n = 100$, $m = 80$, $q = 10$, and vary $\sigma = 2, 4, 6, 8$ and $d = 2, 4$. The results, averaged over 50 replications, are given in Table 1. With both grid selection and coordinate descent, even when the selected parameters do not match the oracle, the average error for the selected model is at most 10% greater than the average oracle error. Moreover, despite fitting fewer models, the coordinate descent approach almost always agrees with the full grid selection. In both settings of $d$, selection becomes more challenging for large values of $\sigma$. For $\sigma = 6$, this affects selection of $q$, while for $\sigma = 8$ it affects selection of $d$. As the noise level increases, both the oracle and NGCV tend to select smaller values of $q$. However, NGCV is more conservative in this respect: its choice of $q$ has already decreased for $\sigma = 6$,

| $d$ | $\sigma$ | $d$-Prop (grid) | Prop (grid) | Ratio (grid) | $d$-Prop (CD) | Prop (CD) | Ratio (CD) |
|---|---|---|---|---|---|---|---|
| 2 | 2 | 0.98 | 0.98 | 1.020 | 0.98 | 0.98 | 1.020 |
| 2 | 4 | 1.00 | 0.96 | 1.003 | 1.00 | 0.96 | 1.003 |
| 2 | 6 | 0.96 | 0.58 | 1.015 | 0.94 | 0.58 | 1.021 |
| 2 | 8 | 1.00 | 0.60 | 1.016 | 0.86 | 0.58 | 1.039 |
| 4 | 2 | 0.84 | 0.84 | 1.094 | 0.88 | 0.88 | 1.072 |
| 4 | 4 | 0.92 | 0.92 | 1.024 | 0.96 | 0.92 | 1.013 |
| 4 | 6 | 0.96 | 0.48 | 1.012 | 0.96 | 0.48 | 1.015 |
| 4 | 8 | 0.66 | 0.54 | 1.017 | 0.58 | 0.46 | 1.026 |

Table 1: Parameter tuning results for scenario (i).

| $d$ | $\sigma$ | $d$-Prop (grid) | Prop (grid) | Ratio (grid) | $d$-Prop (CD) | Prop (CD) | Ratio (CD) |
|---|---|---|---|---|---|---|---|
| 2 | 2 | 1.00 | 0.92 | 1.003 | 1.00 | 0.92 | 1.003 |
| 2 | 4 | 1.00 | 0.82 | 1.005 | 1.00 | 0.82 | 1.005 |
| 2 | 6 | 0.34 | 0.26 | 1.197 | 0.46 | 0.28 | 1.157 |
| 2 | 8 | 0.54 | 0.48 | 1.085 | 0.56 | 0.44 | 1.093 |
| 4 | 2 | 0.98 | 0.94 | 1.024 | 1.00 | 0.94 | 1.002 |
| 4 | 4 | 0.94 | 0.90 | 1.026 | 0.94 | 0.90 | 1.026 |
| 4 | 6 | 0.24 | 0.24 | 1.158 | 0.28 | 0.24 | 1.150 |
| 4 | 8 | 0.54 | 0.48 | 1.057 | 0.64 | 0.52 | 1.043 |

Table 2: Parameter tuning results for scenario (ii).

while the oracle choice does not decrease until $\sigma = 8$, which explains why selection of $q$ is better for $\sigma = 8$ compared to $\sigma = 6$.

For scenario (ii) we fix $n = 100$, $m = 80$, and vary $\sigma = 2, 4, 6, 8$ and $d = 2, 4$. The results, averaged over 50 replications, are given in Table 2. With both grid selection and coordinate descent, even when the selected parameters do not match the oracle, the average error for the selected model is at most 20% greater than the average oracle error. Compared to scenario (i), selection is more difficult in this scenario. When incorrectly selected, $d$ is typically chosen to be larger than the true value, likely due to the unknown orthogonal rotations. Wrong selection of $d$ has a large relative effect on the error, especially with lower $\sigma$, as the true $Z$ must be padded with zeros to match the dimensions of the two objects. However, we can see that especially for grid selection, if we restrict to cases where the ground truth $d$ is selected according to NGCV with grid selection, it is likely that it will also correctly select $q_{\text{ORC}}$. Conversely, when $d$ is chosen to be larger than the truth, the NGCV criterion tends to compensate by choosing $q$ smaller than $q_{\text{ORC}}$.

For scenario (iii) we fix $n = 100$, $m = 80$, $q = 10$, and vary the edge density in $0.5, 0.25, 0.1$ and $d = 2, 4$. Under the Dirichlet-based simulation scheme described in the previous section, we cannot generate RDPG networks with $d = 4$ and density 0.5, so this combination is omitted. In brief, note that the coordinates for different nodes are generated independently from a Dirichlet distribution on the $d$-dimensional probability simplex,

| $d$ | Density | $d$-Prop (grid) | Prop (grid) | Ratio (grid) | $d$-Prop (CD) | Prop (CD) | Ratio (CD) |
|---|---|---|---|---|---|---|---|
| 2 | 1/2 | 1.00 | 1.00 | 1.000 | 1.00 | 1.00 | 1.000 |
| 2 | 1/4 | 1.00 | 1.00 | 1.000 | 1.00 | 1.00 | 1.000 |
| 2 | 1/10 | 0.96 | 0.74 | 1.020 | 0.96 | 0.74 | 1.020 |
| 4 | 1/4 | 1.00 | 1.00 | 1.000 | 1.00 | 1.00 | 1.000 |
| 4 | 1/10 | 0.84 | 0.82 | 1.008 | 0.74 | 0.72 | 1.020 |

Table 3: Parameter tuning results for scenario (iii).

centered at $(1/d \ \cdots \ 1/d)^\intercal$. For two such variables $X$ and $Y$, $\mathbb{E}(X^\intercal Y) = 1/d$. To reduce the overall network density, we can rescale the positions by a constant $0 < \rho \leq 1$, however $\rho > 1$ will produce many pairs of positions with inner product greater than 1, outside the parameter space of the Bernoulli edge distribution. The results, averaged over 50 replications, are given in Table 3. In this scenario, both grid selection and coordinate descent give good selection performance, even for smaller edge densities with weaker signal. Although NGCV typically chooses $q$ smaller than $q_{\text{ORC}}$ for $d = 2$ and density 1/10, we see that this has a small relative effect on the error, as the average selected model error is at most about 2% greater compared to the average oracle error.

## 6. Analysis of International Political Interactions

As an application to real functional network data, we apply FASE to data collected by the Integrated Crisis Early Warning System (ICEWS) (Lautenschlager et al., 2015). In this aggregated data set of international political interactions, we have $m = 108$ monthly snapshots of interaction networks on the $n = 50$ most active countries from January 2005 to December 2013 in terms of total absolute edge weight. An undirected edge $[A_k]_{ij}$ describes the total "weight" of bilateral interaction between country $i$ and country $j$ in month $k$. "Weight" is a signed measure of the intensity and nature of interactions, calculated by the ICEWS. Weights can be both positive, corresponding to cooperative interactions such as giving aid; or negative, corresponding to hostile interactions such as military action. To calculate a weight, the ICEWS automatically scrapes and assigns signed weights to news articles, with edge weights calculated by summing all the news articles for a given month.

As the distribution of edge weights is highly skewed, we apply FASE after a log transformation given by

$$\text{sign}([A_k]_{ij}) \log(1 + |[A_k]_{ij}|)$$

for $k = 1, \ldots, m$ and $1 \leq i < j \leq n$. We use a cubic $B$-spline basis with equally spaced knots, and select $\hat{q} = 5$ and $\hat{d} = 8$ using NGCV. The details of this tuning procedure, including a plot of the grid of NGCV criteria are provided in Appendix D of the supplementary materials.

For interpretability of plots, as a post-processing step we perform a Procrustes alignment of each embedded snapshot to the previous snapshot's embedding. The resulting plotted latent processes are still in the unidentified class $\mathcal{T}(\widehat{Z})$. In Figures 7 and 8, we show an exploratory plot of the FASE at four time points for a subset of the latent dimensions. The

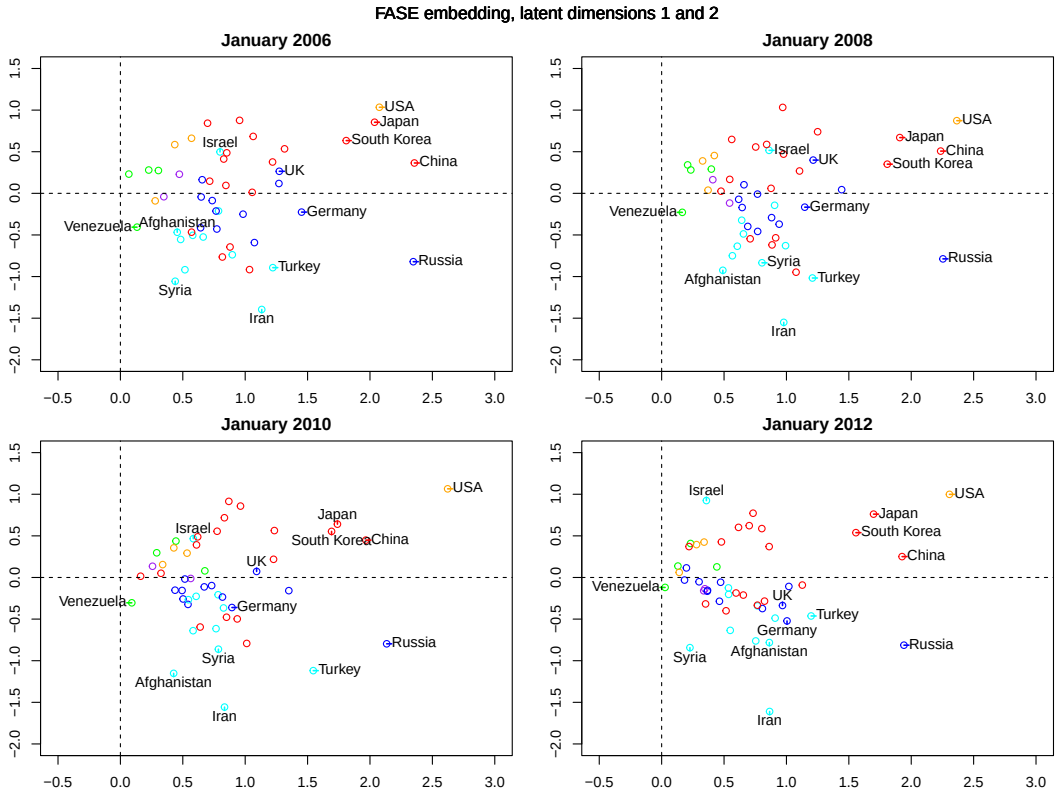


Figure 7: First (horizontal axis) and second (vertical axis) dimensions of FASE evaluated at four times: January 2006, January 2008, January 2010, and January 2012. Points are colored by geographical region. Purple: Africa, Red: Asia-Pacific, Blue: Europe, Cyan: Middle East, Orange: North America, Green: South America.

remaining dimensions are plotted in Appendix D of the supplementary materials. Figure 7 plots the first latent dimension against the second latent dimension, and Figure 8 plots the third latent dimension against the fourth latent dimension. These plots show the FASE estimates at four distinct time snapshots; a detailed view of the estimated latent processes as they evolve in continuous time can be seen in videos available online at `github.com/peterwmacd/fase/tree/main/videos`.

In Figure 7, most countries have positive coordinates in the first latent dimension, corresponding to the total weight and sign of interactions. Countries like the USA, China and Russia, have large positive values in both dimensions at all four of the plotted times. Most Asian countries, plotted in red, have positive coordinates in the second latent dimension, while most European countries, plotted in blue, have negative coordinates. The four Asian nations with large negative coordinates in January 2006, January 2008, and January 2010 are Armenia, Azerbaijan, Georgia and Kazakhstan, four former Soviet republics which are geographically Asian but have more political ties with Europe than with East Asia (Engvall

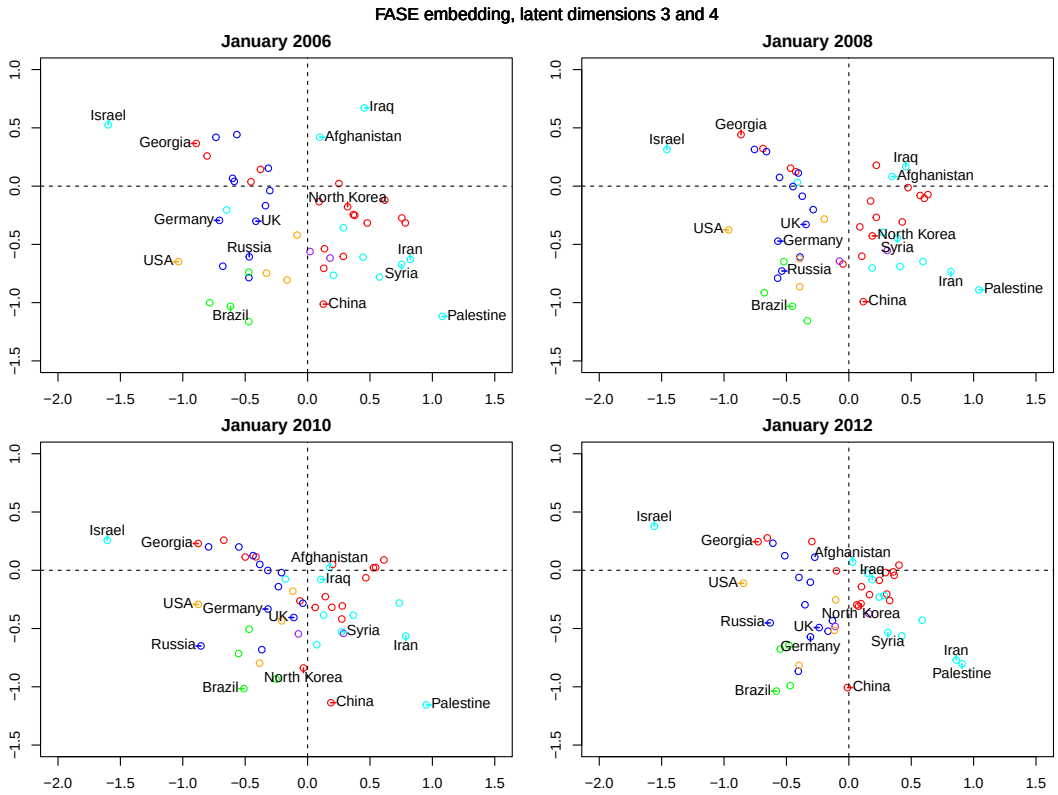


Figure 8: Third (horizontal axis) and fourth (vertical axis) dimensions of FASE evaluated at four times: January 2006, January 2008, January 2010, and January 2012. Points are colored by geographical region. Purple: Africa, Red: Asia-Pacific, Blue: Europe, Cyan: Middle East, Orange: North America, Green: South America.

and Cornell, 2017). We see some dynamic behavior in these plots as well. The first latent coordinate for Syria moves substantially between January 2010 and January 2012, possibly a consequence of the Syrian civil war, which began in late 2011 (BBC News, 2011). There is also consistent movement in the first latent coordinate among countries in the European Union. In January 2006, their mean first latent coordinate is about 0.92, while in January 2012 it is 0.53, reflecting an overall decrease in cooperative relationships during this time period.

In Figure 8, we again see a regional split between Asian countries with mostly positive third latent coordinates; and European countries with mostly negative coordinates. The top left quadrant and the bottom right quadrant separate countries with respect to the Israel-Palestine conflict, which accounts for the largest magnitude negative edges in this network. We see that this conflict appears to pit Israel against most other Middle Eastern nations, while Europe and the USA tend towards the Israeli side of the conflict. Again, there are key dynamic shifts in these plots. In January 2006, the top right and bottom left quadrants appear to separate countries with respect to conflicts between the USA and Iraq, and between the USA and Afghanistan. However, by January 2012 all three countries' latent coordinates are again much closer to the bulk of the cloud.

To further evaluate the dynamic behavior in this network, for each node we calculate the total distance traversed by its latent process in the 8-dimensional latent space. In Figure 9, we show the 20 countries with the greatest distance traversed. Due to boundary effects around the beginning and end of the time interval, we restrict to distance traversed between January 2006 and December 2012.

Taking a closer look at countries with the most dynamic behavior, we see that the civil war in Syria appeared to have implications for its relations with many countries, and its latent position changes substantially in the first, as well as the fifth and seventh latent dimensions during this time period. Afghanistan and Iraq's coordinates both move in the third latent dimension, as well as the seventh, apparently as a result of improving relations with the USA during this time period. Afghanistan's position also moves in the sixth latent dimension, possibly in response to border skirmishes with both Iran (Reuters, 2008) and Pakistan (Reuters, 2013) during this time period. The latent position for North Korea moves substantially in the first and third latent dimensions, in both cases reaching a local minimum around January 2010. As a result, the latent processes for Syria, Afghanistan, Iraq, and North Korea move the most of the countries in the network, despite not being extremely active in terms of total absolute edge weight. They are the 21st, 15th, 22nd, and 14th most active countries, respectively. These findings are consistent with the major world events of that time period.

## 7. Discussion

In this paper, we have introduced a new latent process network model for functional network data collected as either adjacency matrix snapshots or aggregated indexed events. We provide a fitting algorithm using $B$-spline approximation and gradient descent, leading to the FASE estimator. We give theoretical guarantees and demonstrate the efficacy of our method on simulated and real data with both weighted and binary edges, comparing it to existing ASE-based approaches from the literature.

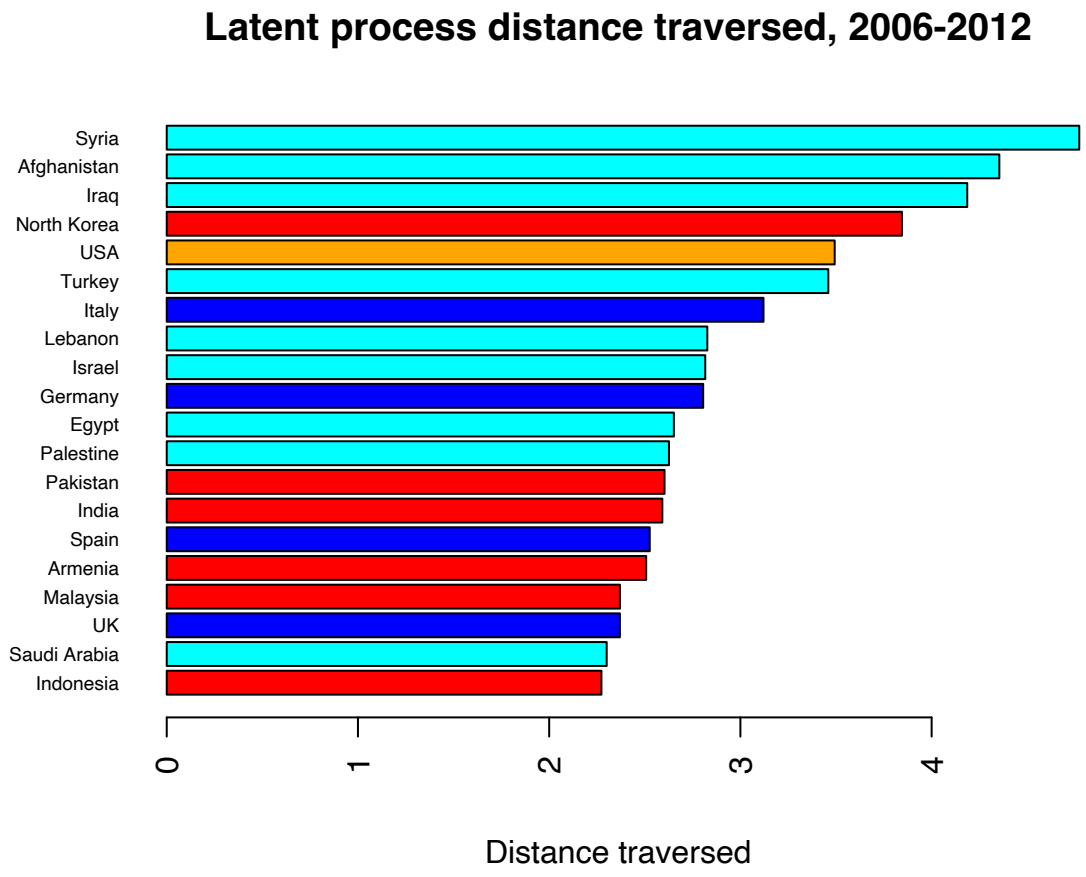


Figure 9: Distance traversed by the estimated latent processes, restricted to the 20 countries with the greatest distance traversed. Bars are colored by geographical region. Red: Asia-Pacific, Blue: Europe, Cyan: Middle East, Orange: North America.

Identifiability remains a challenge for latent process network models of this type. Without strong conditions on eigenvalue separation, even if the orthogonalized latent processes truly belong to span($B$), because of the unknown orthogonal transformation, we cannot take advantage of their parametric form in estimation. Despite this, we have provided theoretical guarantees up to orthogonal transformation, and demonstrated that for smooth latent processes, sharing of information across network snapshots can still lead to more efficient recovery of the underlying network structure.

Future directions include extending the model to accommodate some dependence, in particular autoregressive edge variables, and theory for more general basis functions, including periodic bases which can be used to model seasonality in dynamic networks. Another important direction is developing inference for $Z$ or $\mathcal{W}$, with a view towards finding confidence bands for the latent processes, or testing whether a latent dimension is homogeneous across index values.

## Acknowledgments

This research has been partially funded by the U.S. National Sciences Foundation grants DMS-191622 and DMS-2052918, and a pre-doctoral fellowship from the Rackham Graduate School. There are no financial conflicts of interest to declare.

## Appendix A. Technical Proofs

### A.1 Proof of Proposition 1

**Proof** [Proposition 1] To find the gradient for a given latent dimension $r = 1, \ldots, d$, we can rewrite the objective as

$$\sum_{k=1}^{m} \| \left\{ A_k - \sum_{r' \neq r} \boldsymbol{W}_{r'} B(x_k) B(x_k)^\intercal \boldsymbol{W}_{r'}^\intercal \right\} - \boldsymbol{W}_r B(x_k) B(x_k)^\intercal \boldsymbol{W}_r^\intercal \|_F^2. \tag{14}$$

where the matrix in braces is free of $\boldsymbol{W}_r$. Thus it is sufficient to analyze (14) in the special case $d = 1$, where the objective can be written as

$$\begin{aligned}
&\min_{\boldsymbol{W}} \left\{ \sum_{k=1}^{m} \|A_k - \boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal\|_F^2 \right\} \\
=&\sum_{k=1}^{m} \operatorname{tr}\left( [A_k - \boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal]^2 \right) \\
=&\sum_{k=1}^{m} \operatorname{tr}\left( A_k^2 - A_k \boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal - \boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal A_k \right. \\
&\quad \left. + \boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal \boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal \right) \\
\propto&\sum_{k=1}^{m} \Big\{ - \operatorname{tr}\left(A_k \boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal\right) - \operatorname{tr}\left(\boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal A_k\right) \\
&\quad + \operatorname{tr}\left(\boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal \boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal\right) \Big\},
\end{aligned}$$

where in the final expression we drop the term not depending on $\boldsymbol{W}$. Now take a derivative of each term with respect to $\boldsymbol{W}$. First,

$$\frac{\partial}{\partial \boldsymbol{W}} \operatorname{tr}\left(A_k \boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal\right) = 2 A_k \boldsymbol{W} B(x_k) B(x_k)^\intercal$$

and the other cross term is the same. Then,

$$\frac{\partial}{\partial \boldsymbol{W}} \operatorname{tr}\left(\boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal \boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal\right) = 4 \boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal \boldsymbol{W} B(x_k) B(x_k)^\intercal.$$

The entire gradient with respect to $\boldsymbol{W}$ is

$$-4 \sum_{k=1}^{m} \left(A_k - \boldsymbol{W} B(x_k) B(x_k)^\intercal \boldsymbol{W}^\intercal\right) \boldsymbol{W} B(x_k) B(x_k)^\intercal.$$

Thus for the general case with $d > 1$ we see that the gradient with respect to $\boldsymbol{W}_r$ is the desired

$$-4 \sum_{k=1}^{m} \left\{ A_k - \sum_{r'=1}^{d} \boldsymbol{W}_{r'} B(x_k) B(x_k)^\intercal \boldsymbol{W}_{r'}^\intercal \right\} \boldsymbol{W}_r B(x_k) B(x_k)^\intercal.$$

■

### A.2 Preliminaries for Proofs of Theorems

In this section we will introduce notation as well as some preliminary results which we will use in the proof of Theorems 2.

We continue to use matrix and tensor notation introduced in Section 2, as well as the matrix nuclear norm denoted by $\|\cdot\|_*$, and the Frobenius inner product denoted by $\langle \cdot, \cdot \rangle$.

The Frobenius inner product is given by

$$\langle M, R\rangle = \mathrm{tr}(M^\intercal R)$$

and satisfies $\langle M, M\rangle = \|M\|_F^2$ for matrices $M$ and $R$ of the same dimensions. Recall that with some abuse of notation, we will use $\|\cdot\|_F$ and $\langle\cdot,\cdot\rangle$ to denote the vector $\ell_2$ norm and Euclidean inner product of the vectorization of a 3-mode tensor.

Auxiliary results will typically be referenced below mathematical displays in which they are used. We also use some well known matrix algebra results, including the submultiplicative property of matrix norms, and Cauchy-Schwarz inequality for both the Euclidean and Frobenius inner products. The matrix norms we consider satisfy

$$\frac{1}{\mathrm{rank}(M)}\|M\|_* \le \frac{1}{\mathrm{rank}^{1/2}(M)}\|M\|_F \le \|M\|_2 \le \|M\|_F \le \|M\|_*$$

for any matrix $M$. We also use a special combined submultiplicative property

$$\|MR\|_F \le \|M\|_2\|R\|_F$$

for matrices $M$ and $R$ of suitable dimensions. Next we prove three basic matrix algebra lemmas, and one probability lemma, both of which will be used in the proofs to follow.

**Lemma 5** *Suppose $M$ is an $n\times n$ symmetric matrix, and $X, Y$ are $n\times d$ matrices. Then*

$$\langle MX, X-Y\rangle = \frac{1}{2}\langle M, XX^\intercal - YY^\intercal\rangle + \frac{1}{2}\langle M, (X-Y)(X-Y)^\intercal\rangle\,.$$

**Proof** [Lemma 5]

$$\begin{aligned}
\langle MX, X-Y\rangle &= \langle M, XX^\intercal - YX^\intercal\rangle \\
&= \left\langle M, \frac{1}{2}(XX^\intercal - YY^\intercal) + \frac{1}{2}(XX^\intercal + YY^\intercal) - YX^\intercal\right\rangle \\
&= \frac{1}{2}\langle M, XX^\intercal - YY^\intercal\rangle + \frac{1}{2}\langle M, XX^\intercal + YY^\intercal - YX^\intercal - XY^\intercal\rangle \\
&= \frac{1}{2}\langle M, XX^\intercal - YY^\intercal\rangle + \frac{1}{2}\langle M, (X-Y)(X-Y)^\intercal\rangle\,,
\end{aligned}$$

where the second to last equality uses the symmetry of $M$. ■

**Lemma 6** *Suppose an $n\times q$ matrix $W$ satisfies $\|WB(x_k)\|_2 \le \gamma$ for all $k = 1,\ldots,m$, where $B(x)$ and $\boldsymbol{B}$ are defined as in Section 4. Then under Assumption 1, the $q\times nm$ block matrix*

$$M = \begin{pmatrix} B(x_1)B(x_1)^\intercal W^\intercal & \cdots & B(x_m)B(x_m)^\intercal W^\intercal\end{pmatrix}$$

*satisfies $\|M\|_2 \le \gamma(C_B m/q)^{1/2}$.*

**Proof** [Lemma 6] Rewrite

$$M = \boldsymbol{B}^\intercal \begin{pmatrix} B(x_1)^\intercal W^\intercal & 0 & \\ 0 & \ddots & 0 \\ & 0 & B(x_m)^\intercal W^\intercal \end{pmatrix},$$

where the second factor is an $m \times nm$ block diagonal matrix. Then by Assumption 1, part (B), $\|\boldsymbol{B}\|_2 \leq (C_B m/q)^{1/2}$, and $\|WB(x_k)\|_2 \leq \gamma$ by assumption. ■

**Lemma 7** *Define vectors $x, y, z \in \mathbb{R}^n$. Suppose (without loss of generality) $x^\intercal z > 0$. If*

$$\min_{s \in \{-1,1\}} \|sx - y\|_2 + \|y - z\|_2 < \|x\|_2,$$

*then $x^\intercal y > 0$.*

**Proof** Define $s^* = \operatorname{argmin}_{s\in\{-1,1\}} \|sx - y\|_2$.

Write $c_1 = \|s^*x - y\|_2/\|x\|_2$ and $c_2 = \|y - z\|_2/\|x\|_2$. $c_1, c_2 \in (0, 1)$ and satisfy $c_1 + c_2 < 1$. Then

$$|x^\intercal y| = |x^\intercal(y - s^*x + s^*x)| \geq (1 - c_1)\|x\|_2^2.$$

Suppose, with the goal of obtaining a contradiction, that $x^\intercal y \leq 0$. Since $x^\intercal z > 0$ this implies

$$x^\intercal z - x^\intercal y > (1 - c_1)\|x\|_2^2.$$

On the other hand, by assumption

$$x^\intercal z - x^\intercal y \leq c_2\|x\|_2^2 < (1 - c_1)\|x\|_2^2,$$

which is a contradiction. ■

**Lemma 8** *Suppose $\{A_k\}_{k=1}^m$ are generated from a latent process network model, with independent sub-Gaussian edges with parameter at most $\sigma$, $B(x)$ satisfies Assumption 1, and $n$, $q$ are such that $nq \log q \geq nq \log 5 + (n + q) \log 9$. Define the set*

$$\mathcal{B} = \{W : \|WB(x_k)\|_2 \leq \gamma \quad \forall k = 1, \ldots, m\} \subseteq \mathbb{R}^{n\times q}$$

*for some $\gamma > 0$. Then there is a constant $c_{\text{prob}}$ such that the event*

$$\bigcap_{W\in\mathcal{B}} \left\{ \left\| \frac{1}{m} \sum_{k=1}^m \left\{ A_k - \sum_{r=1}^d Z_r(x_k) Z_r(x_k)^\intercal \right\} WB(x_k)B(x_k)^\intercal \right\|_2 \leq c_{\text{prob}}\gamma \left( \frac{\sigma^2 q^2 n \log q}{m} \right)^{1/2} \right\}$$

*denoted by $\mathcal{E}$, satisfies $\mathbb{P}(\mathcal{E}) \geq 1 - 2\exp(-n/2)$.*

**Proof** [Lemma 8] We will prove a high probability bound for

$$\sup_{W\in\mathcal{B}} \|M(W)\|_2,$$

where $M(W) = \sum_{k=1}^m \{A_k - \sum_{r=1}^d Z_r(x_k)Z_r(x_k)^\intercal\} WB(x_k)B(x_k)^\intercal$. Define

$$\mathcal{B}^+ = \{W : \|W\|_F \leq \frac{\sqrt{C_B q}}{c_B}\gamma\}$$

Note that $\mathcal{B}^+$ is equivalent to a closed Euclidean ball in $\mathbb{R}^{nq}$ of radius $\sqrt{C_B q}\gamma/c_B$. Moreover, for any $W \in \mathcal{B}$,

$$\begin{aligned}
\|W\|_F^2 &= \|W(\boldsymbol{B}^\intercal\boldsymbol{B})(\boldsymbol{B}^\intercal\boldsymbol{B})^{-1}\|_F^2 \\
&\leq \|W\boldsymbol{B}^\intercal\|_F^2 \|\boldsymbol{B}(\boldsymbol{B}^\intercal\boldsymbol{B})^{-1}\|_2^2 \\
&\leq \left(\sum_{k=1}^m \|WB(x_k)\|_2^2\right)\left(\frac{C_B m}{q}\right)\left(\frac{c_B m}{q}\right)^{-2} \\
&\leq \frac{C_B}{c_B^2}\gamma^2 q.
\end{aligned}$$

so that $\mathcal{B} \subseteq \mathcal{B}^+$.

By standard covering results (Vershynin, 2018, Proposition 4.2.12), we can find a $(C_B^{1/2}\gamma/2c_B)$-net for $\mathcal{B}^+$ (under Frobenius metric), denoted by $\mathcal{L}$, satisfying

$$|\mathcal{L}| \leq (4q^{1/2}+1)^{nq} \leq \left(5q^{1/2}\right)^{nq}.$$

Every element $W$ of $\mathcal{B}$ can be written as $W' + E$, where $W' \in \mathcal{L}$, and $\|E\|_2 \leq C_B^{1/2}\gamma/2c_B$. Fix $W \in \mathcal{B}$. Then

$$\begin{aligned}
\|M(W)\|_2 &\leq \|M(W')\|_2 + \|M(E)\|_2 \\
&\leq \max_{W'\in\mathcal{L}} \|M(W')\|_2 + \frac{1}{2}\sup_{W\in\mathcal{B}}\|M(W)\|_2. \\
&\leq \max_{W'\in\mathcal{L}} \left\{2 \max_{x\in\mathcal{N}, y\in\mathcal{M}} x^\intercal M(W')y\right\} + \frac{1}{2}\sup_{W\in\mathcal{B}}\|M(W)\|_2,
\end{aligned}$$

where $\mathcal{N}$ and $\mathcal{M}$ are (1/4)-nets for $\mathcal{S}^{n-1}$ and $\mathcal{S}^{q-1}$ of cardinalities $9^n$ and $9^q$, respectively (Vershynin, 2018, Theorem 4.4.5). Taking a supremum on the left hand side and rearranging,

$$\sup_{W\in\mathcal{B}}\|M(W)\|_2 \leq 4 \max_{W'\in\mathcal{L}, x\in\mathcal{N}, y\in\mathcal{M}} x^\intercal M(W')y. \tag{15}$$

For fixed $x$, $y$, and $W'$, concentration follows directly from Theorem 4.4.5 in Vershynin (2018), and the fact that $\|B(x_k)\|_2 \leq 1$ for all $k$, resulting in the sub-Gaussian tail bound

$$\mathbb{P}\left\{x^\intercal M(W')y \geq t\right\} \leq 2\exp\left(\frac{-c'_{\text{prob}}t^2}{\sigma^2 m q \gamma^2}\right)$$

for a constant $c'_{\text{prob}}$ (which depends on $C_B$ and $c_B$). We now take a union bound over the elements in the net, obtaining

$$\mathbb{P}\left[\max_{W'\in\mathcal{L},x\in\mathcal{N},y\in\mathcal{M}}\left\{x^\intercal M(W')y\right\}\geq t\right]\leq 2\cdot\left(5q^{1/2}\right)^{nq}9^{n+q}\exp\left(\frac{-c'_{\text{prob}}t^2}{\sigma^2 mq\gamma^2}\right).$$

By assumption, $n$ and $q$ satisfy $nq\log q\geq nq\log 5+(n+q)\log 9$. Set

$$c_{\text{prob}}=4\left(\frac{2}{c'_{\text{prob}}}\right)^{1/2},$$

and $t^*=c_{\text{prob}}\sigma\gamma(nmq^2\log q)^{1/2}$. Then

$$\mathbb{P}\left[\max_{W'\in\mathcal{L},x\in\mathcal{N},y\in\mathcal{M}}\left\{x^\intercal M(W')y\right\}\geq\frac{t^*}{4}\right]\leq 2\exp\left(-\frac{n}{2}\right).$$

In combination with (15), this completes the proof. ■

### A.3 Proof of Theorem 2

In this section, we prove Theorem 2. Recall $c_B$, $C_B$, $\gamma_Z$, $\kappa$ defined in the main body of the paper, and define

$$\begin{aligned}
c_{\text{SNR}}&=\frac{\gamma_Z^2}{\sigma}\left(\frac{m}{q^5 n\log q}\right)^{1/2},\\
c_{\text{init}}&=\frac{\|\widehat{\mathcal{W}}^0-\mathcal{W}^{*,0}\|_F^2}{\gamma_Z^2},\\
c_{\text{approx},2}&=\frac{q}{\gamma_Z^2}\sup_{h\geq 0}\varepsilon^{(h)}_{\text{approx},2}\\
c_{\text{approx},\infty}&=\frac{1}{\gamma_Z^2}\sup_{h\geq 0}\varepsilon^{(h)}_{\text{approx},\infty}.
\end{aligned}$$

We start from two necessary lemmas.

**Lemma 9** *Suppose the assumptions of Theorem 2 hold. Fix $h\geq 0$ and let*

$$c_{\text{prev}}=\frac{1}{\gamma_Z^2}\|\widehat{\mathcal{W}}^h-\mathcal{W}^{*,h}\|_F^2.$$

*Then*

$$\|\widehat{\boldsymbol{W}}_r^h B(x_k)\|_2\leq c_W\gamma_Z$$

*uniformly over $r=1,\ldots,d$ and $k=1,\ldots,m$ for a constant $c_W=(c_{\text{prev}}^{1/2}+c_{\text{approx},\infty}^{1/2}+\kappa^{1/2})$.*

**Proof** [Lemma 9]

$$\begin{aligned}\|\widehat{\boldsymbol{W}}_r^h B(x_k)\|_2\|_2 &\leq \|(\widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h})B(x_k)\|_2 + \|\boldsymbol{W}_r^{*,h}B(x_k) - Z(x_k)Q_k^{*,h}\|_2 + \|Z(x_k)Q_k^{*,h}\|_2 \\ &\leq c_{\text{prev}}^{1/2}\gamma_Z + c_{\text{approx},\infty}^{1/2}\gamma_Z + \kappa^{1/2}\gamma_Z.\end{aligned}$$

■

**Lemma 10** *Suppose the assumptions of Theorem 2 hold, and that $\mathcal{E}$ occurs. Fix $h \geq 0$, define $c_{\text{prev}}$ as in Lemma 9, and suppose*

$$c_{\text{prev}} \leq \frac{c_B}{16C_B}. \tag{16}$$

*Then for positive constants*

$$\rho = c_B/8, \quad c_{\text{step}} = \max\left\{c_{\text{prob}}^2 c_W\left(\frac{2d}{c_B} + \frac{1}{16}\right), 4c_{\text{approx},2} + 6\kappa\right\}, \tag{17}$$

*we have*

$$\|\widehat{\mathcal{W}}^{h+1} - \mathcal{W}^{*,h+1}\|_F^2 \leq \left(1 - \eta'\rho\right)\|\widehat{\mathcal{W}}^h - \mathcal{W}^{*,h}\|_F^2 + c_{\text{step}}\eta'\left(\frac{\sigma^2 q^5 n \log q}{\gamma_Z^2 m} + q\varepsilon_{\text{approx},2}^{(h)}\right).$$

**Proof** [Lemma 10] We define the following terms which we will see later in the proof:

$$T_{\text{mean}} = \sum_{k=1}^{m} \left\| \left\{ \sum_{r=1}^{d} \widehat{\boldsymbol{W}}_r^h B(x_k) B(x_k)^\intercal (\widehat{\boldsymbol{W}}_r^h)^\intercal \right\} - Z(x_k) Z(x_k)^\intercal \right\|_F^2,$$

$$T_{\text{cross}} = \left| \sum_{r=1}^{d} \sum_{k=1}^{m} \left\langle \left\{ \sum_{r'=1}^{d} \widehat{\boldsymbol{W}}_{r'}^h B(x_k) B(x_k)^\intercal (\widehat{\boldsymbol{W}}_{r'}^h)^\intercal \right\} - Z(x_k) Z(x_k)^\intercal, \right.$$
$$\left. (\widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h}) B(x_k) B(x_k)^\intercal (\widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h})^\intercal \right\rangle \Bigg|,$$

$$T_{\text{op}} = 2 \left| \sum_{r=1}^{d} \left\langle \sum_{k=1}^{m} \{A_k - Z(x_k) Z(x_k)^\intercal\} \widehat{\boldsymbol{W}}_r^h B(x_k) B(x_k)^\intercal, \widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h} \right\rangle \right|,$$

$$T_{\text{approx}} = 2 \Bigg| \sum_{k=1}^{m} \Bigg\langle \left\{ \sum_{r=1}^{d} \widehat{\boldsymbol{W}}_r^h B(x_k) B(x_k)^\intercal (\widehat{\boldsymbol{W}}_r^h)^\intercal \right\} - Z(x_k) Z(x_k)^\intercal,$$
$$\left\{ \sum_{r'=1}^{d} \boldsymbol{W}_{r'}^{*,h} B(x_k) B(x_k)^\intercal (\boldsymbol{W}_{r'}^{*,h})^\intercal \right\} - Z(x_k) Z(x_k)^\intercal \Bigg\rangle \Bigg|,$$

$$T_{\text{quad.mean}}^2 = 2 \sum_{r=1}^{d} \left\| \sum_{k=1}^{m} \left[ \left\{ \sum_{r'=1}^{d} \widehat{\boldsymbol{W}}_{r'}^h B(x_k) B(x_k)^\intercal (\widehat{\boldsymbol{W}}_{r'}^h)^\intercal \right\} - Z(x_k) Z(x_k)^\intercal \right] \widehat{\boldsymbol{W}}_r^h B(x_k) B(x_k)^\intercal \right\|_F^2,$$

$$T_{\text{quad.op}}^2 = 2 \sum_{r=1}^{d} \left\| \sum_{k=1}^{m} \{A_k - Z(x_k) Z(x_k)^\intercal\} \widehat{\boldsymbol{W}}_r^h B(x_k) B(x_k)^\intercal \right\|_F^2.$$

Then we have

$$
\begin{aligned}
&\|\widehat{\mathcal{W}}^{h+1} - \mathcal{W}^{*,h+1}\|_F^2 \\
=&\sum_{r=1}^{d} \|\widehat{\boldsymbol{W}}_r^{h+1} - \boldsymbol{W}_r^{*,h+1}\|_F^2 \\
\leq&\sum_{r=1}^{d} \|\widehat{\boldsymbol{W}}_r^{h+1} - \boldsymbol{W}_r^{*,h}\|_F^2 \\
=&\sum_{r=1}^{d} \|\widehat{\boldsymbol{W}}_r^{h} - \boldsymbol{W}_r^{*,h} + \eta_h \sum_{k=1}^{m} \left( A_k - \sum_{r'=1}^{d} \left\{ \widehat{\boldsymbol{W}}_{r'}^h B(x_k) B(x_k)^\intercal \left[ \widehat{\boldsymbol{W}}_{r'}^h \right]^\intercal \right\} \right) \widehat{\boldsymbol{W}}_r^h B(x_k) B(x_k)^\intercal \|_F^2 \\
\leq&\sum_{r=1}^{d} \|\widehat{\boldsymbol{W}}_r^{h} - \boldsymbol{W}_r^{*,h}\|_F^2 + \eta_h^2 T_{\text{quad.mean}}^2 + \eta_h^2 T_{\text{quad.op}}^2 \\
&+ 2\eta_h \sum_{r=1}^{d} \left\langle \sum_{k=1}^{m} \left( A_k - \sum_{r'=1}^{d} \left\{ \widehat{\boldsymbol{W}}_{r'}^h B(x_k) B(x_k)^\intercal \left[ \widehat{\boldsymbol{W}}_{r'}^h \right]^\intercal \right\} \right) \widehat{\boldsymbol{W}}_r^h B(x_k) B(x_k)^\intercal, \widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h} \right\rangle \\
\leq&\|\widehat{\mathcal{W}}^{h+1} - \mathcal{W}^{*,h+1}\|_F^2 + \eta_h^2 T_{\text{quad.mean}}^2 + \eta_h^2 T_{\text{quad.op}}^2 + \eta_h T_{\text{op}} \\
&- 2\eta_h \sum_{r=1}^{d} \left\langle \sum_{k=1}^{m} \sum_{r'=1}^{d} \left\{ \left( \widehat{\boldsymbol{W}}_{r'}^h B(x_k) B(x_k)^\intercal \left[ \widehat{\boldsymbol{W}}_{r'}^h \right]^\intercal \right\} - Z(x_k) Z(x_k)^\intercal \right) \widehat{\boldsymbol{W}}_r^h B(x_k) B(x_k)^\intercal, \widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h} \right\rangle \\
\leq&\|\widehat{\mathcal{W}}^{h} - \mathcal{W}^{*,h}\|_F^2 + \eta_h^2 T_{\text{quad.mean}}^2 + \eta_h^2 T_{\text{quad.op}}^2 + + \eta_h T_{\text{op}} + \eta_h T_{\text{approx}} + \eta_h T_{\text{cross}} - \eta_h T_{\text{mean}},
\end{aligned}
\tag{18}
$$

where the first inequality follows from the choice of $\mathcal{W}^{*,h+1}$, and the final inequality uses Lemma 5. We will next bound each of these terms.

For $T_{\text{quad.mean}}$, we fix $r \in \{1, \ldots, d\}$ and bound each term by

$$
\begin{aligned}
T_{\text{quad.mean}} &= \sqrt{2} \left\| \begin{pmatrix} \widehat{\boldsymbol{W}}_r^h B(x_1) B(x_1)^\intercal (\widehat{\boldsymbol{W}}_r^h)^\intercal - Z_r(x_1) Z_r(x_1)^\intercal \\ \vdots \\ \widehat{\boldsymbol{W}}_r^h B(x_m) B(x_m)^\intercal (\widehat{\boldsymbol{W}}_r^h)^\intercal - Z_r(x_m) Z_r(x_m)^\intercal \end{pmatrix}^\intercal \begin{pmatrix} \widehat{\boldsymbol{W}}_r^h B(x_1) B(x_1)^\intercal \\ \vdots \\ \widehat{\boldsymbol{W}}_r^h B(x_m) B(x_m)^\intercal \end{pmatrix} \right\|_F \\
&\leq \sqrt{2} \left\| \begin{pmatrix} \widehat{\boldsymbol{W}}_r^h B(x_1) B(x_1)^\intercal (\widehat{\boldsymbol{W}}_r^h)^\intercal - Z_r(x_1) Z_r(x_1)^\intercal \\ \vdots \\ \widehat{\boldsymbol{W}}_r^h B(x_m) B(x_m)^\intercal (\widehat{\boldsymbol{W}}_r^h)^\intercal - Z_r(x_m) Z_r(x_m)^\intercal \end{pmatrix}^\intercal \right\|_F \left\| \begin{pmatrix} \widehat{\boldsymbol{W}}_r^h B(x_1) B(x_1)^\intercal \\ \vdots \\ \widehat{\boldsymbol{W}}_r^h B(x_m) B(x_m)^\intercal \end{pmatrix} \right\|_2 \\
&\leq T_{\text{mean}}^{1/2} \gamma_Z \left( \frac{2 c_W m}{q} \right)^{1/2},
\end{aligned}
$$

to conclude

$$
T_{\text{quad.mean}}^2 \leq 2 c_W d \frac{m \gamma_Z^2}{q} T_{\text{mean}}.
$$

For $T_{\text{quad.op}}$, we fix $r \in \{1, \ldots, d\}$ and bound each term by

$$\begin{aligned} T_{\text{quad.op}} &\leq \sqrt{2}m \| \frac{1}{m} \sum_{k=1}^{m} \{A_k - Z_r(x_k) Z_r(x_k)^\intercal\} \widehat{\boldsymbol{W}}_r^h B(x_k) B(x_k)^\intercal \|_F \\ &\leq 2mq^{1/2} \| \frac{1}{m} \sum_{k=1}^{m} \{A_k - Z_r(x_k) Z_r(x_k)^\intercal\} \widehat{\boldsymbol{W}}_r^h B(x_k) B(x_k)^\intercal \|_2 \\ &\leq \sqrt{2}mq^{1/2} c_{\text{prob}} c_W \gamma_1 \left( \frac{\sigma^2 q^2 n \log q}{m} \right)^{1/2}, \end{aligned}$$

where the final inequality uses Lemma 8, to conclude

$$T_{\text{quad.op}}^2 \leq 2c_{\text{prob}}^2 c_W^2 d \sigma^2 q^3 m \gamma_Z^2 n \log q.$$

For $T_{\text{cross}}$, we bound

$$\begin{aligned} T_{\text{cross}} &\leq \sum_{r=1}^{d} \left\{ T_{\text{mean}}^{1/2} \left\| \begin{pmatrix} (\widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h}) B(x_1) B(x_1)^\intercal (\widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h})^\intercal \\ \vdots \\ (\widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h}) B(x_m) B(x_m)^\intercal (\widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h})^\intercal \end{pmatrix} \right\|_F \right\} \\ &\leq \sum_{r=1}^{d} \left\{ T_{\text{mean}}^{1/2} \| \widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h} \|_F \left\| \begin{pmatrix} (\widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h}) B(x_1) B(x_1)^\intercal \\ \vdots \\ (\widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h}) B(x_m) B(x_m)^\intercal \end{pmatrix} \right\|_2 \right\} \\ &\leq \sum_{r=1}^{d} \left\{ \left( \frac{C_B m T_{\text{mean}}}{q} \right)^{1/2} \| \widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h} \|_F^2 \right\} \\ &= \left( \frac{C_B m T_{\text{mean}}}{q} \right)^{1/2} \| \widehat{\mathcal{W}}^h - \mathcal{W}^{*,h} \|_F^2 \\ &\leq c_{\text{cross}} T_{\text{mean}} + \frac{c_{\text{prev}} C_B m \gamma_Z^2}{4 c_{\text{cross}} q} \| \widehat{\mathcal{W}}^h - \mathcal{W}^{*,h} \|_F^2 \end{aligned}$$

for any positive constant $c_{\text{cross}}$. Specifying $c_{\text{cross}} = 1/8$, we conclude

$$T_{\text{cross}} \leq \frac{1}{8} T_{\text{mean}} + \frac{2 c_{\text{prev}} C_B m \gamma_Z^2}{4q} \| \widehat{\mathcal{W}}^h - \mathcal{W}^{*,h} \|_F^2.$$

For $T_{\text{op}}$, applying Lemma 8,

$$\begin{aligned}
T_{\text{op}} &\leq \sum_{r=1}^{d} 2m \| \frac{1}{m} \sum_{k=1}^{m} (A_k - Z_r(x_k) Z_r(x_k)^\intercal) \widehat{\boldsymbol{W}}_r^h B(x_k) B(x_k)^\intercal \|_2 \| \widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h} \|_* \\
&\leq \sum_{r=1}^{d} m q^{1/2} \left\{ c_{\text{prob}} c_W \gamma_1 \left( \frac{\sigma^2 q^2 n \log q}{m} \right)^{1/2} \right\} \| \widehat{\boldsymbol{W}}_r^h - \boldsymbol{W}_r^{*,h} \|_F \\
&\leq \frac{c_{\text{prob}}^2 c_W^2 d \sigma^2 q^4 n \log q}{4 c_{\text{op}}} + \frac{c_{\text{op}} m \gamma_1^2}{q} \| \widehat{\mathcal{W}}^h - \mathcal{W}^{*,h} \|_F^2,
\end{aligned}$$

where $\|\cdot\|_*$ denotes the matrix nuclear norm. Specifying $c_{\text{op}} = c_B/8$, we conclude

$$T_{\text{op}} \leq \frac{2 c_{\text{prob}}^2 c_W^2 d \sigma^2 q^4 n \log q}{c_B} + \frac{c_B m \gamma_1^2}{8q} \| \widehat{\mathcal{W}}^h - \mathcal{W}^{*,h} \|_F^2.$$

For $T_{\text{approx}}$, we require one auxiliary result:

$$\sum_{k=1}^{m} \| \left\{ \sum_{r=1}^{d} \boldsymbol{W}_r^{*,h} B(x_k) B(x_k)^\intercal (\boldsymbol{W}_r^{*,h})^\intercal \right\} - Z(x_k) Z(x_k)^\intercal \|_F^2 \leq (4 c_{\text{approx},2} + 6\kappa) m \gamma_Z^2 \varepsilon_{\text{approx},2}^{(h)}. \tag{19}$$

Then

$$\begin{aligned}
T_{\text{approx}} &= 2 \Bigg| \sum_{k=1}^{m} \Bigg\langle \left\{ \sum_{r=1}^{d} \widehat{\boldsymbol{W}}_r^h B(x_k) B(x_k)^\intercal (\widehat{\boldsymbol{W}}_r^h)^\intercal \right\} - Z(x_k) Z(x_k)^\intercal, \\
&\qquad \left\{ \sum_{r'=1}^{d} \boldsymbol{W}_{r'}^{*,h} B(x_k) B(x_k)^\intercal (\boldsymbol{W}_{r'}^{*,h})^\intercal \right\} - Z(x_k) Z(x_k)^\intercal \Bigg\rangle \Bigg| \\
&\leq T_{\text{mean}}^{1/2} \left\{ \sum_{k=1}^{m} \| \left\{ \sum_{r'=1}^{d} \boldsymbol{W}_{r'}^{*,h} B(x_k) B(x_k)^\intercal (\boldsymbol{W}_{r'}^{*,h})^\intercal \right\} - Z(x_k) Z(x_k)^\intercal \|_F^2 \right\}^{1/2} \\
&\leq \{ (4 c_{\text{approx},2} + 6\kappa) m \}^{1/2} \gamma_Z T_{\text{mean}}^{1/2} (\varepsilon_{\text{approx},2}^{(h)})^{1/2} \\
&\leq c'_{\text{approx}} T_{\text{mean}} + \frac{(4 c_{\text{approx},2} + 6\kappa) m \gamma_Z^2}{4 c'_{\text{approx}}} \varepsilon_{\text{approx},2}^{(h)},
\end{aligned}$$

for any positive constant $c'_{\text{approx}}$. The second inequality uses (19).

Specifying $c'_{\text{approx}} = 1/4$, we conclude

$$T_{\text{approx}} \leq \frac{1}{4} T_{\text{mean}} + (4 c_{\text{approx},2} + 6\kappa) m \gamma_Z^2 \varepsilon_{\text{approx},2}^{(h)}.$$

For $T_{\text{mean}}$, define an $m$-tuple of orthogonal transformations, $\mathcal{Q}^{\text{Proc},h}$ such that for each $k = 1, \ldots, m$,

$$Q_k^{\text{Proc},h} = \text{argmin}_{Q \in \mathcal{O}_d} \| \widehat{\mathcal{W}}^h \times_2 B(x_k) - Z(x_k) Q \|_F^2$$

which has a closed form expression (Cape et al., 2019). Define an operator

$$\mathcal{P}_{\boldsymbol{B}}(Z) = Z \times_2 (\boldsymbol{B}^\intercal \boldsymbol{B})^{-1} \boldsymbol{B}^\intercal,$$

and suppose that for an arbitrary $n \times m \times d$ tensor $\mathcal{Z}$ and an $m$-tuple of orthogonal transformations $\mathcal{Q}$, $\mathcal{Z}\mathcal{Q}$ gives the $n \times m \times d$ tensor which right multiplies each component of $\mathcal{Q}$ by the corresponding $n \times d$ slice of $\mathcal{Z}$. Then,

$$\begin{aligned}
\|\widehat{\mathcal{W}}^h - \mathcal{W}^{*,h}\|_F^2 &\leq \|\widehat{\mathcal{W}}^h - \mathcal{P}_{\boldsymbol{B}}(\mathcal{Z}\mathcal{Q}^{\mathrm{Proc},h})\|_F^2 \\
&= \|\left\{\widehat{\mathcal{W}}^h - \mathcal{P}_{\boldsymbol{B}}(\mathcal{Z}\mathcal{Q}^{\mathrm{Proc},h})\right\} \times_2 \left\{(\boldsymbol{B}^\intercal \boldsymbol{B})^{-1} \boldsymbol{B}^\intercal\right\} \boldsymbol{B}\|_F^2 \\
&\leq \frac{q}{c_B m} \|\widehat{\mathcal{W}}^h \times_2 \boldsymbol{B} - \mathcal{P}_{\boldsymbol{B}}(\mathcal{Z}\mathcal{Q}^{\mathrm{Proc},h}) \times_2 \boldsymbol{B}\|_F^2 \\
&\leq \frac{q}{c_B m} \sum_{k=1}^{m} \|\widehat{\mathcal{W}}^h \bar{\times}_2 B(x_k) - Z(x_k) Q_k^{\mathrm{Proc},h}\|_2^2 \\
&\leq \frac{q}{c_B m 2(2^{1/2}-1)\gamma_Z^2} T_{\mathrm{mean}}. \qquad (20)
\end{aligned}$$

The first inequality follows from the choice of $\mathcal{W}^{*,h}$, the second inequality follows from Assumption 1, the third inequality follows from a projection argument, and the final inequality follows from Ma et al. (2020), Lemma 28.

This display implies that

$$\frac{1}{2} T_{\mathrm{mean}} \geq \frac{c_B(\sqrt{2}-1) m \gamma_Z^2}{q} \|\widehat{\mathcal{W}}^h - \mathcal{W}^{*,h}\|_F^2.$$

Substituting all these inequalities into (18), we have that

$$\begin{aligned}
&\|\widehat{\mathcal{W}}^{h+1} - \mathcal{W}^{*,h+1}\|_F^2 \\
\leq &\|\widehat{\mathcal{W}}^h - \mathcal{W}^{*,h}\|_F^2 + \eta_h^2 T_{\mathrm{quad.mean}}^2 + \eta_h^2 T_{\mathrm{quad.op}}^2 + \eta_h T_{\mathrm{op}} \\
&+ \eta_h T_{\mathrm{approx}} + \eta_h T_{\mathrm{cross}} - \eta_h T_{\mathrm{mean}} \\
\leq &\left(1 + \frac{\eta_h c_B m \gamma_Z^2}{8q} + \frac{2\eta_h c_{\mathrm{prev}} C_B m \gamma_Z^2}{q} - \frac{\eta_h c_B (2^{1/2}-1) m \gamma_Z^2}{q}\right) \|\widehat{\mathcal{W}}^h - \mathcal{W}^{*,h}\|_F^2 \\
&+ \left(\frac{\eta_h}{8} + \frac{\eta_h}{4} + 2\eta_h^2 c_W d \frac{m\gamma_Z^2}{q} - \frac{1}{2}\eta_h\right) T_{\mathrm{mean}} \\
&+ \frac{2\eta_h c_{\mathrm{prob}}^2 c_W^2 d \sigma^2 q^4 n \log q}{c_B} + 2\eta_h^2 c_{\mathrm{prob}}^2 c_W^2 d \sigma^2 q^3 m \gamma_Z^2 n \log q \\
&+ \eta_h (4 c_{\mathrm{approx},2} + 6\kappa) m \gamma_Z^2 \varepsilon_{\mathrm{approx},2}^{(h)}. \qquad (21)
\end{aligned}$$

We consider the coefficients of the first two terms of (21) separately. For the second term, expanding $\eta_h \equiv \eta' q/m\gamma_Z^2$, for a constant

$$\eta' = \left(32d\left\{\left(\frac{c_B}{16C_B}\right)^{1/2} + c_{\text{approx},\infty}^{1/2} + \kappa^{1/2}\right\}\right)^{-1},$$

we have coefficient

$$\left(\frac{\eta' q}{\gamma_Z^2 m}\right)\left(\frac{1}{8} + 14 + 2\eta' c_W d - \frac{1}{2}\right) < 0.$$

For the first term, again expanding the definition of $\eta_h$, we have coefficient

$$\left(1 - \eta'\left[c_B(2^{1/2}-1) - \frac{c_B}{8} - 2c_{\text{prev}}C_B\right]\right).$$

Since $c_{\text{prev}} \leq c_B/(16C_B)$ by assumption, we lower bound the quantity inside the square brackets by $\rho$. Then, expanding the definition of $\eta_h$ in the final three terms of (21), and choosing a constant $c_{\text{step}}$ as in (17), we complete the proof. ■

To complete the proof of Theorem 2, we begin by showing that (16) and $\mathcal{E}$ (defined in Lemma 8) hold with high probability for all $h \geq 0$, and thus we can repeatedly apply Lemma 10. Suppose (16) holds for all $0 \leq h' \leq h$. Then by repeated application of Lemma 10,

$$\begin{aligned}\|\widehat{\mathcal{W}}^{h+1} - \mathcal{W}^{*,h+1}\|_F^2 &\leq \|\widehat{\mathcal{W}}_1^0 - \mathcal{W}^{*,0}\|_F^2 + \frac{c_{\text{step}}}{\rho}\left(\frac{\sigma^2 q^5 n \log q}{\gamma_Z^2 m} + q \cdot \max_{0\leq h'\leq h} \varepsilon_{\text{approx},2}^{(h')}\right)\\ &\leq \left(c_{\text{init}} + \frac{c_{\text{step}}}{\rho c_{\text{SNR}}^2} + \frac{c_{\text{step}} c_{\text{approx},2}}{\rho}\right)\gamma_Z^2,\end{aligned}$$

which implies that both the inductive step and the base case hold as long as $c_{\text{init}}$ and $c_{\text{approx}}$ are sufficiently small, and $c_{\text{SNR}}$ is sufficiently large.

In particular, we require that

$$c_{\text{init}} + \frac{c_{\text{step}}}{\rho c_{\text{SNR}}^2} + \frac{c_{\text{step}} c_{\text{approx},2}}{\rho} \leq \frac{c_B}{16C_B}. \tag{22}$$

Recall that $\rho = c_B/8$ and by assumption,

$$c_{\text{step}} \leq \max\left\{c_{\text{prob}}^2\left\{\left(\frac{c_B}{16C_B}\right)^{1/2} + c_{\text{approx},\infty}^{1/2} + \kappa^{1/2}\right\}\left(\frac{2d}{c_B} + \frac{1}{16}\right), 4c_{\text{approx},2} + 6\kappa\right\}.$$

Thus, it is easy to see that there exist positive constants $\nu_1$, $\nu_2$, $\nu_3$, and $\nu_4$ (written in terms of $c_B$, $C_B$, $\kappa$, $d$, and $c_{\text{prob}}$ but free of $n$, $m$, $q$, $\sigma$ and $\gamma_Z$) such that

$$c_{\text{SNR}} \geq \nu_1, \quad c_{\text{init}} \leq \nu_2, \quad c_{\text{approx},2} \leq \nu_3, \quad c_{\text{approx},\infty} \leq \nu_4 \tag{23}$$

is sufficient for (22) to hold.

By Assumptions 2-4 and Lemma 8, choose $n$ sufficiently large so that $nq\log q \geq nq\log 5 + (n+q)\log 9$, and both $\mathcal{E}$ and (23) hold with probability at least $1-\xi$.

Thus, we can repeatedly apply Lemma 10 to conclude that for any $h \geq 0$,

$$\|\widehat{\mathcal{W}}^h - \mathcal{W}^{*,h}\|_F^2 \leq \left(1-\eta'\rho\right)^h \|\widehat{\mathcal{W}}^0 - \mathcal{W}^{*,0}\|_F^2 + c_{\text{step}}\eta' \frac{\sigma^2 q^5 n \log q}{\gamma_Z^2 m} \sum_{j=0}^{h} \left(1-\eta'\rho\right)^j + c_{\text{step}}\eta' q \sum_{j=0}^{h} \varepsilon_{\text{approx},2}^{(h-j)} \left(1-\eta'\rho\right)^j.$$

The first two terms have limits in $h$, and for even $h$, the final term satisfies

$$\begin{aligned}
\sum_{j=0}^{h} \varepsilon_{\text{approx},2}^{(h-j)} \left(1-\eta'\rho\right)^j &\leq \frac{c_{\text{approx},2}\gamma_Z^2}{q}(1-\eta'\rho)^{h/2} \sum_{j=0}^{h/2}(1-\eta'\rho)^j + \left(\sup_{j'>h/2} \varepsilon_{\text{approx},2}^{(j')}\right) \sum_{j=0}^{h/2}(1-\eta'\rho)^j \\
&\leq \frac{c_{\text{approx},2}\gamma_Z^2}{q}(1-\eta'\rho)^{h/2} \sum_{j=0}^{h/2}(1-\eta'\rho)^j + \left(\sup_{j'>h/2} \varepsilon_{\text{approx},2}^{(j')}\right) \sum_{j=0}^{h/2}(1-\eta'\rho)^j \\
&\leq \frac{c_{\text{approx},2}\gamma_Z^2}{\eta'\rho q}(1-\eta'\rho)^{h/2} + \frac{1}{\eta'\rho} \sup_{j'>h/2} \varepsilon_{\text{approx},2}^{(j')}.
\end{aligned}$$

Thus for a constant $C_2' = c_{\text{step}}/\rho + 2$,

$$\limsup_{h\to\infty} \|\widehat{\mathcal{W}}^h - \mathcal{W}^{*,h}\|_F^2 \leq C_2' \left(\frac{\sigma^2 q^5 n \log q}{\gamma_Z^2 m} + q \limsup_{h\to\infty} \varepsilon_{\text{approx},2}^{(h)}\right).$$

Then to complete the proof of (11),

$$\begin{aligned}
&\limsup_{h\to\infty} \frac{1}{m}\sum_{k=1}^{m} \|\widehat{Z}^h(x_k) - Z(x_k)Q_k^{*,h}\|_F^2 \\
=&\limsup_{h\to\infty} \frac{1}{m}\sum_{k=1}^{m} \|\widehat{\mathcal{W}}^h \bar{\times}_2 B(x_k) - \mathcal{W}^{*,h} \bar{\times}_2 B(x_k) + \mathcal{W}^{*,h} \bar{\times}_2 B(x_k) - Z(x_k)Q_k^{*,h}\|_F^2 \\
\leq&\limsup_{h\to\infty} \left\{ \frac{2}{m} \|(\widehat{\mathcal{W}}^h - \mathcal{W}^{*,h}) \times_2 \boldsymbol{B}\|_F^2 + 2\varepsilon_{\text{approx},2}^{(h)} \right\} \\
\leq& C_2 \left( \frac{\sigma^2 q^4 n \log q}{\gamma_Z^2 m} + \limsup_{h\to\infty} \varepsilon_{\text{approx},2}^{(h)} \right)
\end{aligned}$$

for a constant $C_2 = 2C_B C_2' + 2$, as desired.

Recall that for arbitrary $\xi > 0$, this inequality of random variables holds with probability at least $1-\xi$ for any $n \geq N(\xi)$. By assumption,

$$\limsup_{h\to\infty} \varepsilon_{\text{approx},2}^{(h)} = O_p(\alpha_n),$$

and thus, by definition we conclude that

$$\limsup_{h\to\infty} \frac{1}{m}\sum_{k=1}^{m} \|\widehat{Z}^h(x_k) - Z(x_k)Q_k^{*,h}\|_F^2 = O_p\left(\frac{\sigma^2 q^4 n \log q}{\gamma_Z^2 m} + \alpha_n\right),$$

completing the proof.

### A.4 Proof Sketch of Corollary 3

The proof of Corollary 3 proceeds almost identically to Theorem 2, specifying $d = 1$.

The difference is in the derivation of the upper bound on $T_{\text{mean}}$ in Lemma 10, display (20), where we must prove that the optimal (Procrustes) alignment is given by the identity transformation for each $k = 1, \ldots, m$.

Recall that

$$c_{\text{approx},\infty} = \frac{1}{\gamma_Z^2} \max_{k=1,\ldots,m} \|\boldsymbol{W}_1^* B(x_k) - Z_1(x_k)\|_2^2.$$

Following Cape et al. (2019), the optimal one-dimensional alignment is given by

$$\text{sign}\{Z_1(x_k)^\intercal \widehat{\boldsymbol{W}}_1^h B(x_k)\}.$$

Thus by Lemma 7, it is sufficient to show that

$$2\|\widehat{\boldsymbol{W}}_1^h B(x_k) - Z_1(x_k)\|_2 \le \|Z_1(x_k)\|_2. \tag{24}$$

By definition, $\|Z_1(x_k)\|_2 \ge \gamma_Z$, and by assumption,

$$2\|\widehat{\boldsymbol{W}}_1^h B(x_k) - Z_1(x_k)\|_2 \le 2\gamma_Z(c_{\text{prev}}^{1/2} + c_{\text{approx},\infty}^{1/2}).$$

Recall that by an assumption of Lemma 10, $c_{\text{prev}} \le c_B/16C_B \le 1/16$. Thus a sufficient condition for (24) is $c_{\text{approx},\infty} \le 1/16$.

We can add this condition on $c_{\text{approx},\infty}$ to the statement of Lemma 10, and the remainder of the proof still goes through with possibly stronger condition

$$c_{\text{approx},\infty} \le \nu_4'$$

in (23). Then since (23) still holds asymptotically under Assumption 5, we complete the proof of Corollary 3.

### A.5 Proof of Proposition 4

In this section we prove the main theoretical result stated in Section 4.2 regarding the estimation performance of our local averaged initializer (see Section 3.1).

Recall that we assumed partition sets of equal integer size for local averaging. For $\ell = 1, \ldots, L$, let $k^*(\ell)$ denote the median index in $T_\ell$.

We first state generic perturbation bound on the ASE error up to unknown rotation.

**Lemma 11** *Let $P = VV^\intercal$, where $V \in \mathbb{R}^{n\times d}$, and $E \in \mathbb{R}^{n\times n}$ a symmetric matrix. Then*

$$\min_{Q\in\mathcal{O}_d} \|\mathrm{ASE}_d(P+E) - VQ\|_F^2 \leq \frac{10d\|E\|_2^2}{\lambda_d(P)}.$$

**Proof** By Lemma 5.4 in Tu et al. (2016), we have

$$\min_{Q\in\mathcal{O}_d} \|\mathrm{ASE}_d(P+E) - VQ\|_F^2 \leq \frac{1}{2(\sqrt{2}-1)\lambda_d(P)}\|[P+E]_{(d)} - P\|_F^2$$

where $[\cdot]_{(d)}$ denotes rank truncation.

$$\begin{aligned}
\|[P+E]_{(d)} - P\|_F &\leq \sqrt{2d}\|[P+E]_{(d)} - P\|_2 \\
&\leq \sqrt{2d}\{\|[P+E]_{(d)} - (P+E)\|_2 + \|E\|_2\} \\
&\leq \sqrt{2d}\{\lambda_{d+1}(P+E) + \|E\|_2\} \\
&\leq 2\sqrt{2d}\|E\|_2
\end{aligned}$$

where the first step uses the relationship between the operator and Frobenius norms for low rank matrices, and the final step uses Weyl's inequality. Thus,

$$\min_{Q\in\mathcal{O}_d} \|\mathrm{ASE}_d(P+E) - VQ\|_F^2 \leq \frac{8d}{2(\sqrt{2}-1)\lambda_d(P)}\|E\|_2^2$$

as desired, since $8/\{2(\sqrt{2}-1)\} \leq 10$. ■

The following lemma provides a high probability bound on the errors of the local averages used for initialization.

**Lemma 12** *Under the setting of Proposition 4, with probability at least $1 - 4L\exp(-n)$,*

$$\max_{1\leq\ell\leq L} \left\| \frac{1}{M}\sum_{k\in\tilde{T}_\ell} (A_k - \mathbb{E}A_k) \right\|_2 \leq c_{\mathrm{prob}}\sigma\sqrt{\frac{n}{M}} \tag{25}$$

**Proof** For $\ell = 1, \ldots, L$, let

$$\tilde{E}_\ell = \frac{1}{M}\sum_{k\in\tilde{T}_\ell} (A_k - \mathbb{E}A_k).$$

these matrices are mutually independent over $\ell$, and have independent subgaussian edges with parameter at most $\sigma/\sqrt{M}$.

Fix $\ell$. By Vershynin (2018), Corollary 4.4.8, we have with probability at least $1 - 4\exp(-n)$,

$$\|\tilde{E}_\ell\|_2 \leq c_{\mathrm{prob}}\sigma\sqrt{\frac{n}{M}}$$

for some universal constant $c_{\mathrm{prob}}$.

Then by a union bound, we have that (25) holds with probability at least $1-4L\exp(-n)$, as desired. ■

**Proof** [Proposition 4]

First, we want to bound the error of $\widehat{Z}_\ell^0$ as an estimator of $Z(x_{k^*(\ell)})$ up to an unknown rotation. By the Lemma 11, this requires control of the operator norm error

$$\left\| \frac{1}{M} \sum_{k \in \tilde{T}_\ell} A_k - \Theta(x_{k^*(\ell)}) \right\|_2 . \tag{26}$$

for $\ell = 1, \ldots, L$.

The random part of the operator norm error can be controlled with high probability by Lemma 12. By triangle inequality and the relationship between norms, the deterministic part is bounded above by

$$\leq \max_{k' \in T_\ell} \|\Theta(x_{k'}) - \Theta(x_{k^*(\ell)})\|_2.$$

$$\begin{aligned}
&\|\Theta(x_{k'}) - \Theta(x_{k^*(\ell)})\|_2 \\
=&\|Z(x_{k'})Z(x_{k'})^\intercal - Z(x_{k^*(\ell)})Z(x_{k^*(\ell)})^\intercal\|_2 \\
=&\|Z(x_{k'})Z(x_{k'})^\intercal - Z(x_{k'})Z(x_{k^*(\ell)})^\intercal + Z(x_{k'})Z(x_{k^*(\ell)})^\intercal - Z(x_{k^*(\ell)})Z(x_{k^*(\ell)})^\intercal\|_2 \\
\leq&2\gamma_Z \left( \|Z(x_{k'}) - Z(x_k^*(\ell))\|_F^2 \right)^{1/2} \\
\leq&2\gamma_Z \left( \sum_{i,r} \{z_{i,r}(x_{k'}) - z_{i,r}(x_k^*(\ell))\}^2 \right)^{1/2} \\
\leq&\frac{2K_1\gamma_Z\sqrt{nd}}{L}
\end{aligned}$$

uniformly over $k' \in T_\ell$.

Combining the random and deterministic parts, we get (26) is bounded above by

$$\frac{2K_1\gamma_Z\sqrt{nd}}{L} + \frac{c_{\text{prob}}\sigma\sqrt{nL}}{\sqrt{m}}$$

with high probability over all $\ell = 1, \ldots, L$.

Applying Lemma 11, we get that for each $\ell = 1, \ldots, L$, there exists $\tilde{Q}_\ell$ such that

$$\|\widehat{Z}_\ell^0 - Z(x_{k^*(\ell)})\tilde{Q}_\ell\|_F \leq \frac{2K_1 d\sqrt{10n}}{L} + \frac{c_{\text{prob}}\sigma\sqrt{10dLn}}{\gamma_Z\sqrt{m}}.$$

Now fix an arbitrary $k' \in T_\ell$ and note that by the Lipschitz condition,

$$\|Z(x_{k'})\tilde{Q}_\ell - Z(x_{k^*(\ell)})\tilde{Q}_\ell\|_F \leq \frac{K_1\sqrt{nd}}{L},$$

and thus

$$\max_{k' \in \tilde{T}_\ell} \|\hat{Z}_\ell^0 - Z(x_{k'})\tilde{Q}_\ell\|_F \leq \frac{(2\sqrt{10d}+1)K_1\sqrt{dn}}{L} + \frac{c_{\text{prob}}\sigma\sqrt{10dLn}}{\gamma_Z\sqrt{m}}.$$

Recall that for arbitrary $k \in \{1, \ldots, m\}$, we defined an initializer for the unknown process snapshots as

$$\widehat{Z}^0(x_k) = \{\widehat{Z}_\ell^0 : k \in T_\ell\},$$

which suggests a natural set of initial coordinates

$$\widehat{\mathcal{W}}^0 = \widehat{\mathcal{Z}}^0 \times_2 (\boldsymbol{B}^\intercal \boldsymbol{B})^{-1}\boldsymbol{B}^\intercal \in \mathbb{R}^{n\times q\times d}.$$

Analogously define an initial target process which is aligned to the initializer,

$$\widetilde{\mathcal{Z}}^0 = \{Z(x_k)\tilde{Q}_\ell : k \in T_\ell\}_{k=1}^m.$$

$$\begin{aligned}
\|\widehat{\mathcal{W}}^0 - \mathcal{W}^{*,0}\|_F^2 &\leq \|(\widehat{\mathcal{Z}}^0 - \widetilde{\mathcal{Z}}^0) \times_2 (\boldsymbol{B}^\intercal\boldsymbol{B})^{-1}\boldsymbol{B}^\intercal\|_F^2 \\
&\leq \frac{C_B q}{c_B^2 m}\sum_{\ell=1}^{q}\sum_{k'\in\tilde{T}_\ell}\|\hat{Z}_\ell^0 - Z(x_{k'})\tilde{Q}_\ell\|_F^2 \\
&\leq \left\{\frac{C_B q}{c_B^2}\left(\frac{(2\sqrt{10d}+1)K_1\sqrt{dn}}{\gamma_Z L} + \frac{c_{\text{prob}}\sigma\sqrt{10dLn}}{\gamma_Z^2\sqrt{m}}\right)^2\right\}\gamma_Z^2,
\end{aligned}$$

where the first inequality follows by definition of the target coordinates, since it minimizes the error over rotations of the true processes.

This nonasymptotic bound which holds with probability at least $1 - 4L\exp(-n)$, as desired. ■

## Appendix B. Derivation of NGCV Criterion

In this appendix we will derive the specific form of the NGCV criterion (7). First, recall the standard generalized cross validation (GCV) criterion for linear regression, which is derived based on leave one out cross validation (Golub et al., 1979). Suppose we have univariate responses $y_i$ and $p$-dimensional predictors $\boldsymbol{x}_i$ for $i = 1, \ldots, n$, with $n \times p$ design matrix $\boldsymbol{X}$. The least squares regression coefficients for this problem are $\hat{\beta} = (\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal\boldsymbol{y}$ and the *hat matrix* is given by $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^\intercal\boldsymbol{X})^{-1}\boldsymbol{X}^\intercal$. Then the generalized cross validation criterion (Golub et al., 1979) is

$$\sum_{i=1}^{n}\left(\frac{y_i - \boldsymbol{x}_i^\intercal\hat{\beta}}{1 - [\boldsymbol{H}]_{ii}}\right)^2.$$

A common approximation is to replace each of the diagonal elements of $\boldsymbol{H}$ with their mean, resulting in the criterion

$$\left\{1 - \frac{\mathrm{tr}(\boldsymbol{H})}{n}\right\}^{-2} \left\{\frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^\intercal \hat{\beta})^2\right\}.$$

In least squares, $\mathrm{tr}(\boldsymbol{H}) = p$, so we get the further simplification

$$\left(1 - \frac{p}{n}\right)^{-2} \left\{\frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^\intercal \hat{\beta})^2\right\}.$$

which can be calculated directly from the mean of squared residuals and the dimensions of the linear regression problem.

This simplified GCV criterion will be the building block of the NGCV criterion. Recall from (3) the form of $\ell(\mathcal{W})$, which we will expand in the following way in terms of individual basis coordinates $\boldsymbol{w}_{i,r}$ for $i = 1, \ldots, n$ and $r = 1, \ldots, d$:

$$\ell(\mathcal{W}) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m}\left\{[A_k]_{ij} - \sum_{r=1}^{d}\boldsymbol{w}_{i,r}^\intercal B(x_k)B(x_k)^\intercal \boldsymbol{w}_{j,r}\right\}^2.$$

Note that we can view this as a summation of $2n$ least squares objectives, where the first $n$ fix $i$, the snapshot row, and the second $n$ fix $j$, the snapshot column, and sum over the other two indices However, to avoid double counting matrix entries, we suppose that each $[A_k]_{ij}$ is assigned in a balanced way to either the row $i$ problem or the column $j$ problem. As a result, each problem is a least squares problem with $nm/2$ observations.

In truth, the basis coordinates are shared between these problems and each problem involves the coordinates for all the nodes. Without loss of generality, consider the row 1 problem. For simplicity, suppose the problem only optimizes over the $qd$ total coordinates $\{\boldsymbol{w}_{1,r}\}_{r=1}^{d}$, treating the others as fixed. Thus, plugging in the fitted coordinates $\widehat{W}$ and summing over the $nm/2$ observations for this problem, we can calculate the simplified GCV criterion as

$$\left(1 - \frac{2qd}{nm}\right)^{-2}\left[\frac{2}{nm}\sum_{j}\sum_{k}\left\{[A_k]_{1j} - \sum_{r=1}^{d}\hat{\boldsymbol{w}}_{1,r}^\intercal B(x_k)B(x_k)^\intercal \hat{\boldsymbol{w}}_{j,r}\right\}^2\right].$$

Each of the component squared errors in $\ell(\mathcal{W})$ appears in exactly one of the $2n$ problems. Thus taking a mean of GCV's over these problems we get the overall criterion

$$\left(1 - \frac{2qd}{nm}\right)^{-2}\left\{\frac{1}{mn^2}\ell(\widehat{W})\right\},$$

which is equivalent to the NGCV criterion (7).

## Appendix C. Additional Evaluation on Synthetic Networks

### C.1 Recovery up to a Single Unknown Orthogonal Transformation

In this appendix we evaluate FASE on synthetic functional network data in terms of latent process recovery up to a single unknown orthogonal transformation. As additional post-processing, we perform a sequential Procrustes alignment for a collection of snapshot process estimates, similar to the alignment procedure used in Algorithm 2, and used as post-processing for the FASE estimate in Section 6. We will use this procedure to unambiguously select representatives of the unidentified classes $\mathcal{T}(Z)$ for the ground truth latent processes, and $\mathcal{T}(\widehat{Z})$ for the FASE estimator. Formally, suppose we have latent processes $\widetilde{Z}$, evaluated at indices $\{y_1, \ldots, y_{m'}\} \subset \mathcal{X}$ and stored in an $n \times m' \times d$ tensor. Then the sequential Procrustes alignment procedure $\text{Proc}_{m'}$ sets $\widetilde{O}_1 = I_d$, then for $k = 2, \ldots, m'$, replaces the $k$th $n \times d$ slice $\widetilde{Z}(y_k)$ by $\widetilde{Z}(y_k)\widetilde{O}_k$, where

$$\widetilde{O}_k = \text{argmin}_{O \in \mathcal{O}_d} \|\widetilde{Z}(y_k)O - \widetilde{Z}(y_{k-1})\widetilde{O}_{k-1}\|_F^2.$$

For simplicity, we will set $m' = m$ and compute $\text{Proc}_m$ using the same snapshot times used to generate the data. As the FASE estimator is well-defined for any $x \in \mathcal{X}$, we can evaluate this same sequential Procrustes alignment for arbitrarily fine grids. We stress that this alignment procedure is completely internal to its argument $\widetilde{Z}$, and does not require oracle knowledge of any ground truth $Z$.

We will compare FASE to the same baseline approaches for the same scenarios and settings as in Section 5, but with a new error metric given by

$$\text{Err}_Z^*(\widehat{Z}) = \min_{Q_0 \in \mathcal{O}_d} \left\{ \frac{1}{ndm} \sum_{k=1}^{m} \left\| \text{Proc}_m\{\widehat{Z}\}(x_k) - \text{Proc}_m\{Z\}(x_k)Q_0 \right\|_F^2 \right\}^{1/2}.$$

In contrast to $\text{Err}_Z$ defined in Section 5, this metric only requires optimization over a single orthogonal transformation-valued argument. We also plot $\text{Err}_Z$ for the oracle version of FASE (FASE (ORC, ErrZ)) as an achieveable lower bound for $\text{Err}_Z^*$ for the same oracle FASE estimator.

Many of the overall conclusions from these plots are the same as in Section 5, although we will highlight some key differences in performance seen as a result of switching error metrics.

In Figures 10 and 11, we report results for scenario (i). In all settings, FASE shows a modest difference between $\text{Err}_Z^*$ and the lower bound $\text{Err}_Z$. The difference is most pronounced for small $m$ and $n$, where the larger estimation error is magnified by the sequential Procrustes alignment, and for large $m$, as the domain of the optimization in the definition of $\text{Err}_Z$ grows relative to the analogous domain in the definition of $\text{Err}_Z^*$.

In Figure 12, we show the relationship between the two error metrics for scenario (i) and the setting with $\sigma = 2$, $n = 100$ and $m = 20$. We can see that in the majority of cases for FASE (left panel), both metrics perform well, with comparable error that exceeds the best results of ASE (right panel). Moreover, in terms of $\text{Err}_Z$, FASE outperforms ASE in about 90% of iterations. However, the corresponding $\text{Err}_Z^*$ can be much larger, a phenomenon that occurs only once for ASE. In these cases, the FASE estimate also tends to have slighly

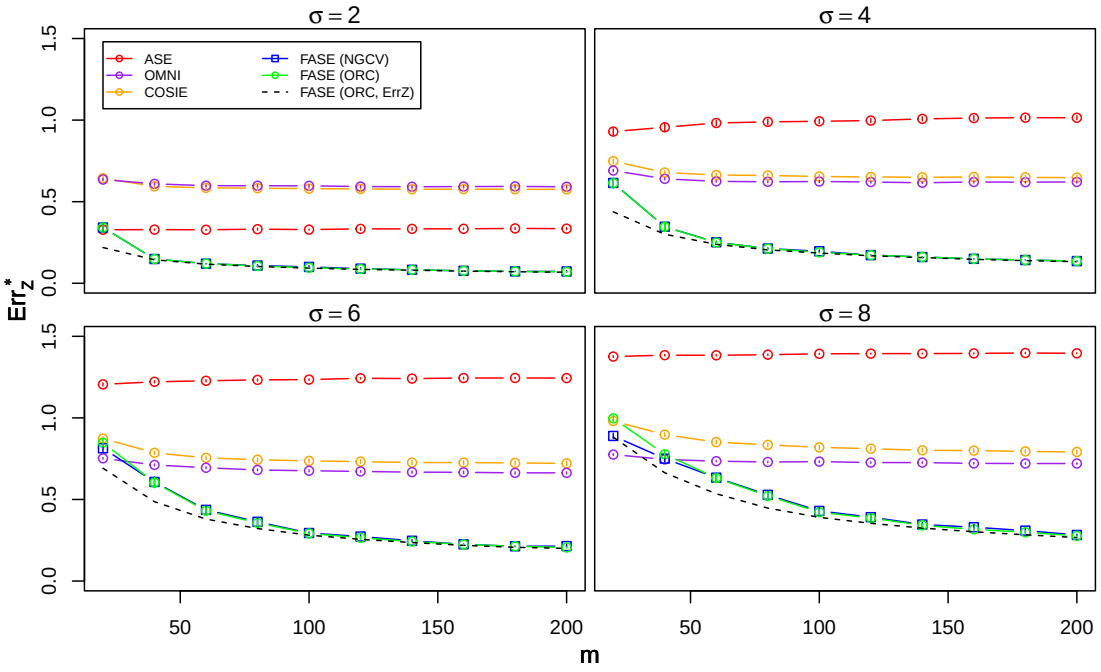


Figure 10: Mean of $\text{Err}_Z^*$, varying $m$, the number of snapshots. Scenario (i), parametric Gaussian networks. Plots are labeled by edge standard deviation $\sigma$.

higher $\text{Err}_Z$, suggesting it has attempted to "smooth out" a discontinuity in the sequence of aligning orthogonal transformations used to evaluate $\text{Err}_Z$, and converged to a local minimum of the objective which cannot take full advantage of the parametric form of the true processes. This phenomenon appears to occur infrequently for sufficiently large values of $m$ or $n$.

In Figures 13 and 14, we report results for scenario (ii). In this nonparametric scenario, switching error metrics has a more substantial effect on the performance of FASE. While we still see a decrease in $\text{Err}_Z^*$ when increasing either $m$ or $n$, there are now high signal settings for $\sigma \leq 4$ in which FASE does not clearly dominate ASE, even for relatively large values of $m$ and $n$.

In Figure 15, we report results for scenario (iii). In this scenario, for sufficiently large $n$, and most values of $m$, there appears to be very little difference in the two error metrics. Especially for functional networks with edge density 1/2, $\text{Err}_Z^*$ can increase as $m$ increases. This phenomenon is considered above for scenario (i), see Figure 12 and the related discussion.

### C.2 Interpolation for Missing Snapshots

In this section we evaluate our FASE estimator against other baselines in the literature as a tool for interpolation to predict edges that were not observe. We generate data similar to scenario (ii) in the manuscript as follows. We set $n = 100$, $d = 2$, and generate latent

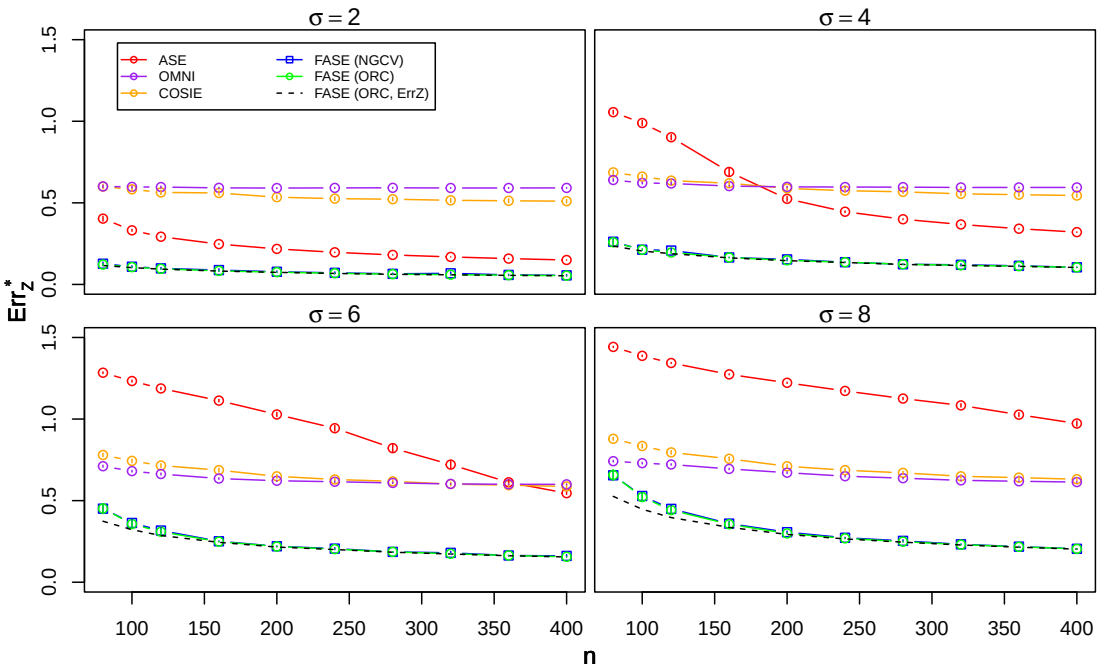


Figure 11: Mean of $\mathrm{Err}_Z^*$, varying $n$, the number of nodes. Scenario (i), parametric Gaussian networks. Plots are labeled by edge standard deviation $\sigma$.

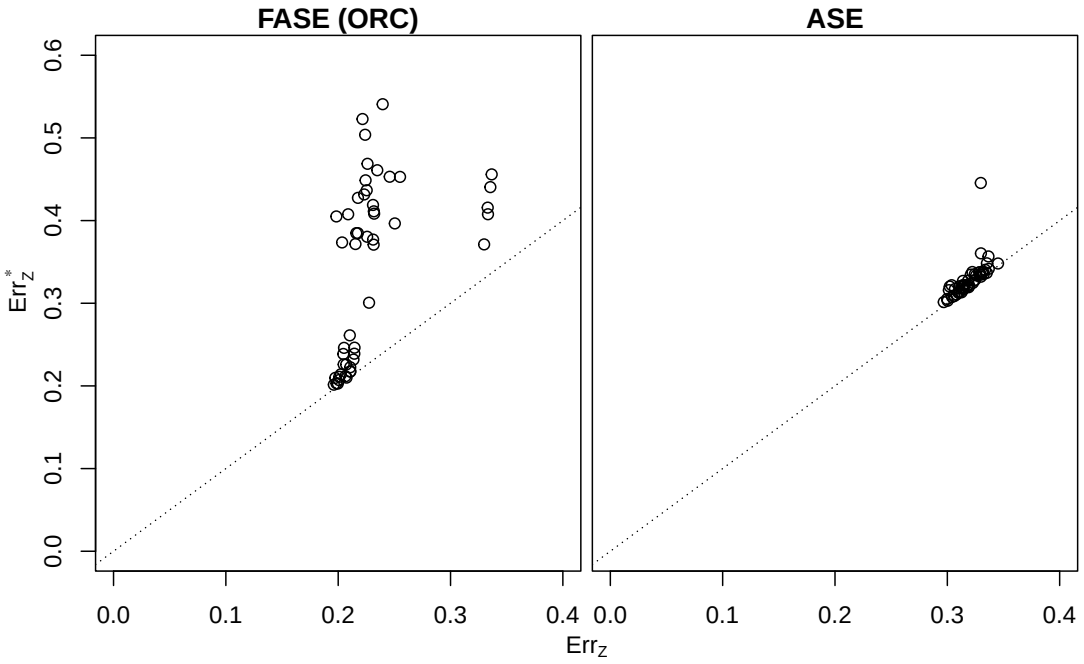


Figure 12: Scatter plot of $\mathrm{Err}_Z$ against $\mathrm{Err}_Z^*$ for FASE (ORC) (left panel) and ASE (right panel). Scenario (i), $\sigma = 2$, $n = 100$, $m = 20$. Dotted lines denote $x = y$.

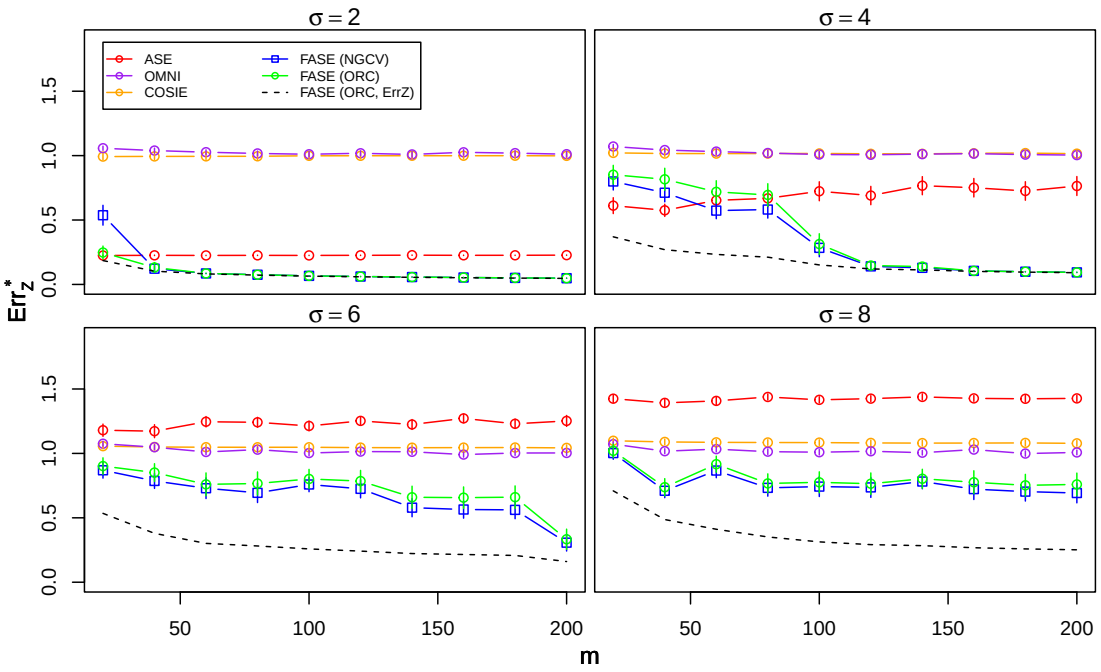


Figure 13: Mean of $\mathrm{Err}^*_Z$, varying $m$, the number of snapshots. Scenario (ii), nonparametric Gaussian networks. Plots are labeled by edge standard deviation $\sigma$.

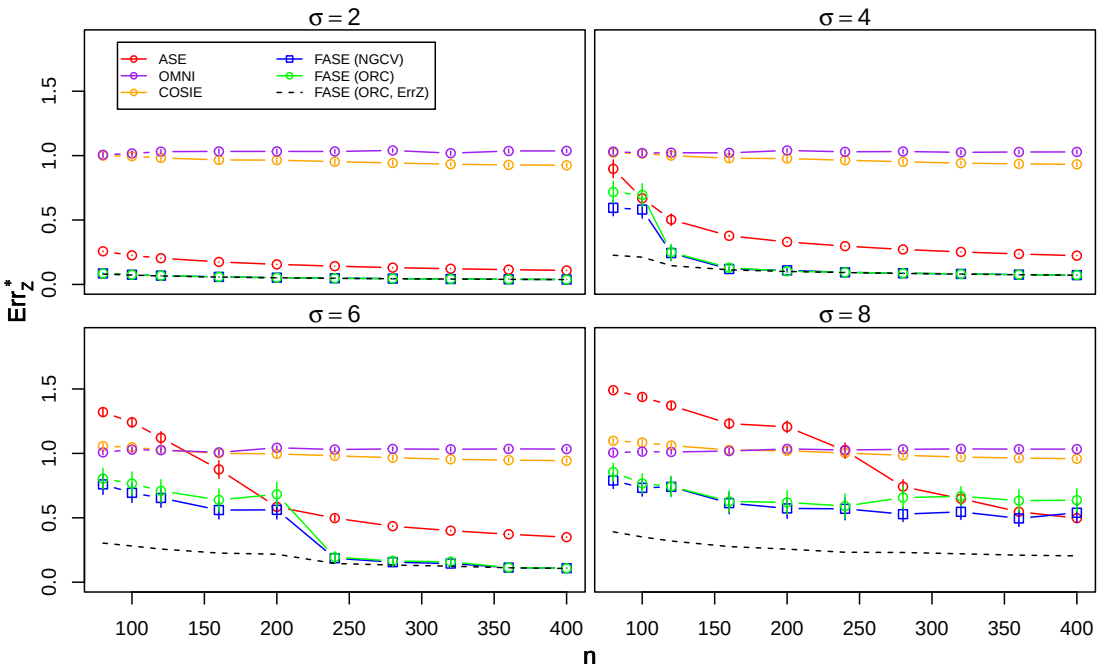


Figure 14: Mean of $\text{Err}_Z^*$, varying $n$, the number of nodes. Scenario (ii), nonparametric Gaussian networks. Plots are labeled by edge standard deviation $\sigma$.

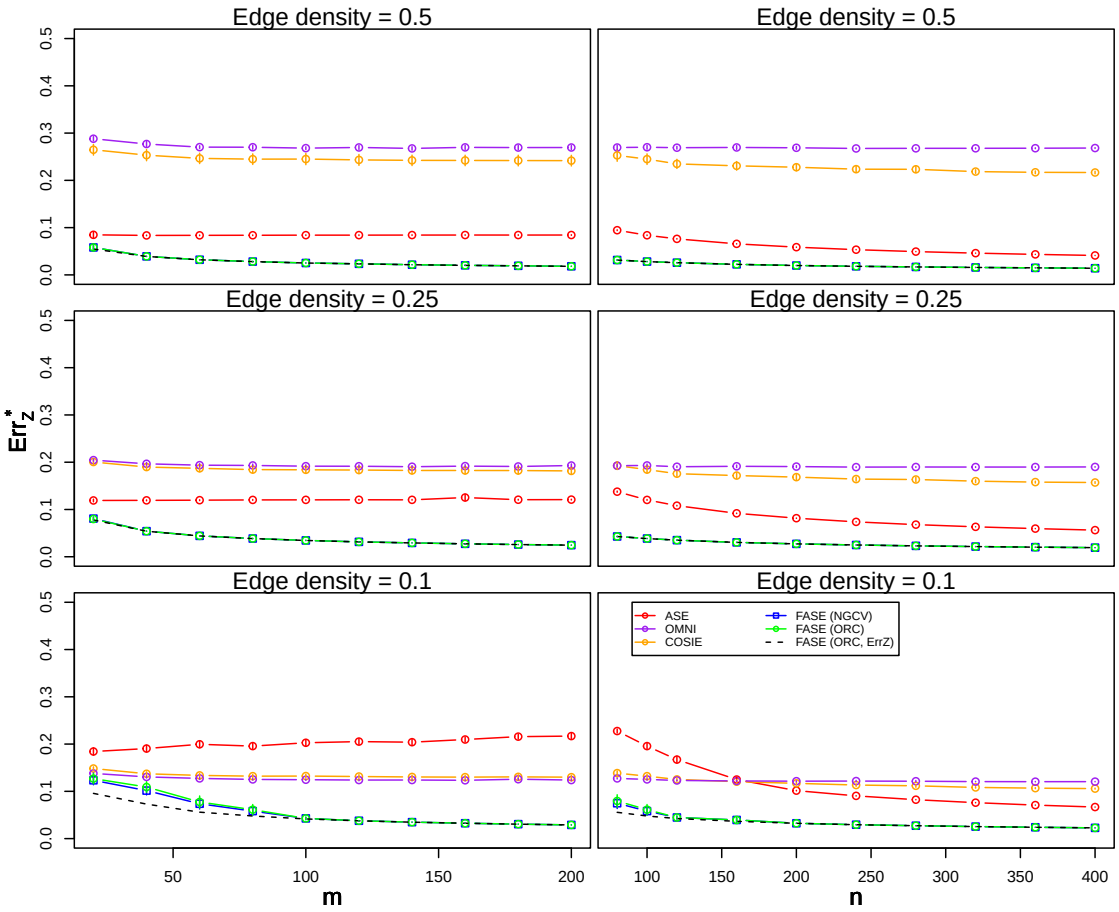


Figure 15: Mean of $\text{Err}_Z^*$, varying $m$, the number of snapshots (left column), and $n$, the number of nodes (right column). Scenario (iii), parametric RDPG networks. Plots are labeled by edge density.

processes according to

$$z_{i,r}(x) = \frac{3\sin[C\pi(2x - U_{i,r})]}{1 + 5[x + B_{i,r}(1 - 2x)]} + G_{i,r}$$

for a constant $C$, where $U_{i,r} \overset{\text{iid}}{\sim} \text{Unif}[0,1]$, $B_{i,r} \overset{\text{iid}}{\sim} \text{Bernoulli}(1/2)$, and $G_{i,r} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/4)$. Setting $C = 2$, the processes go through two complete cycles in the index space $\mathcal{X} = [0,1]$, as in Section 5. We also consider smoother processes which only go through one complete cycle by setting $C = 1$.

We initially generate $m = 100$ equally spaced snapshots on index space $\mathcal{X} = [0,1]$. We then uniformly select a random snapshot index $x_k^* \in [0.25, 0.5]$ (to avoid boundary effects) and remove it along with $M$ snapshots immediately before and after $x_k^*$, for $M = 0, 1, ..., 10$. That is, we treat the $2M+1$ network snapshots closest to the selected snapshot in the index space as missing.

For all of the dynamic network embedding methods we consider, our goal will be estimating the expected adjacency matrix of the central unobserved snapshot, evaluated in terms of the RMSE

$$\text{Err}_{\Theta-\text{mid}}(\widehat{Z}) = \frac{1}{n}\|\widehat{Z}(x_k^*)\widehat{Z}(x_k^*)^\intercal - \Theta(x_k^*)\|_F.$$

For FASE, estimation of $Z$ for an unobserved snapshot index is simple given the basis design and estimated coordinates. To compare to ASE, we consider estimation based on an embedding of the closest observed snapshot, either the next smallest (ASE (below)) or the next largest (ASE (above)). To compare to OMNI and COSIE, which produce embeddings nearly constant in the index space, we average the estimated embeddings for the next smallest and next largest observed snapshots.

Note that in contrast to the examples in the main paper, the snapshot indices for this partially missing data are not equally spaced. Thus to choose the basis for the FASE estimator, rather than specifying equally spaced knots in index space, we place them at equally spaced quantiles of the observed snapshot times. We also consider both oracle and NGCV-selected parameters for FASE, where $d$ is selected form $\{1, 2, 3, 4\}$, and $q$ from $\{6, 8, 10, 12, 14, 16\}$.

The results are presented in Figure 16 for processes which complete one cycle, and Figure 17 for processes which complete two cycles.

We see that in this nonparametric setting, FASE performs well for interpolation of unobserved snapshots relative to competing dynamic network embeddings. The relative performance of the different approaches is similar to the network recovery study in Section 5. Comparing the results in Figures 16 and 17, FASE performs better when the underlying processes are smoother in the index space. In particular, FASE gives the best performance of all methods for all settings in Figure 16. In Figure 17, in almost all settings, FASE outperforms the ASE approaches, especially as $\sigma$ increases. FASE outperforms or is competitive with COSIE and OMNI in most settings. Both OMNI and COSIE produce nearly constant embeddings in this setting, meaning their bias is quite insensitive to the number of missing snapshots, while the bias of FASE is sensitive to the number of missing snapshots, and the smoothness of the underlying latent processes.

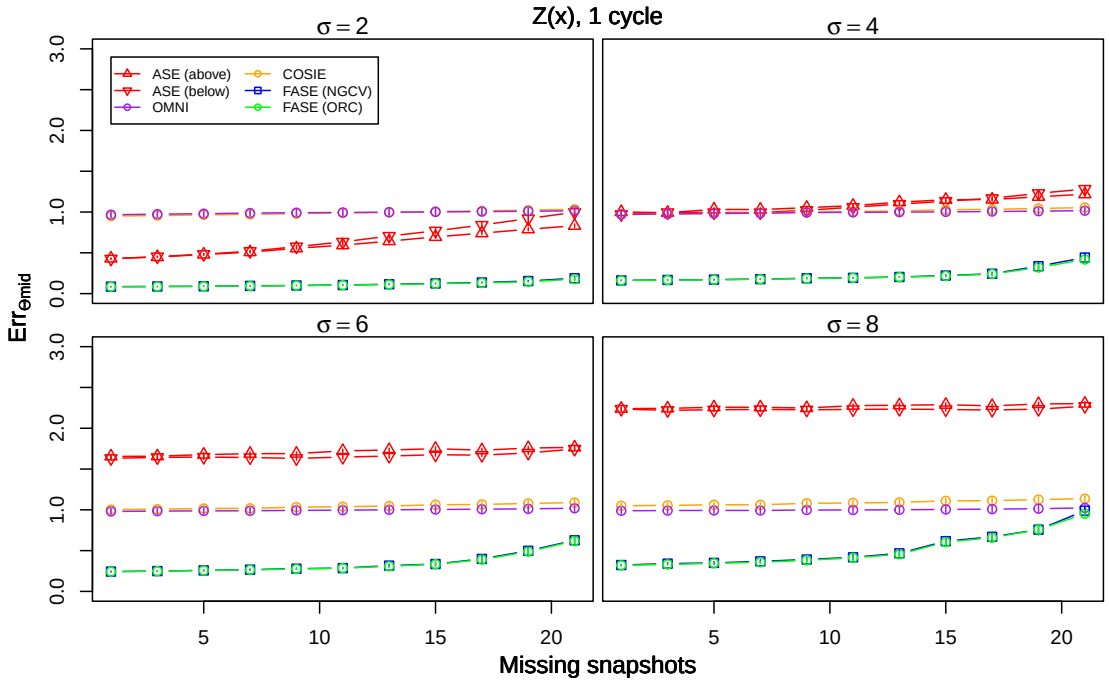


Figure 16: Prediction performance for central unobserved snapshot, latent processes complete one cycle in $[0, 1]$.

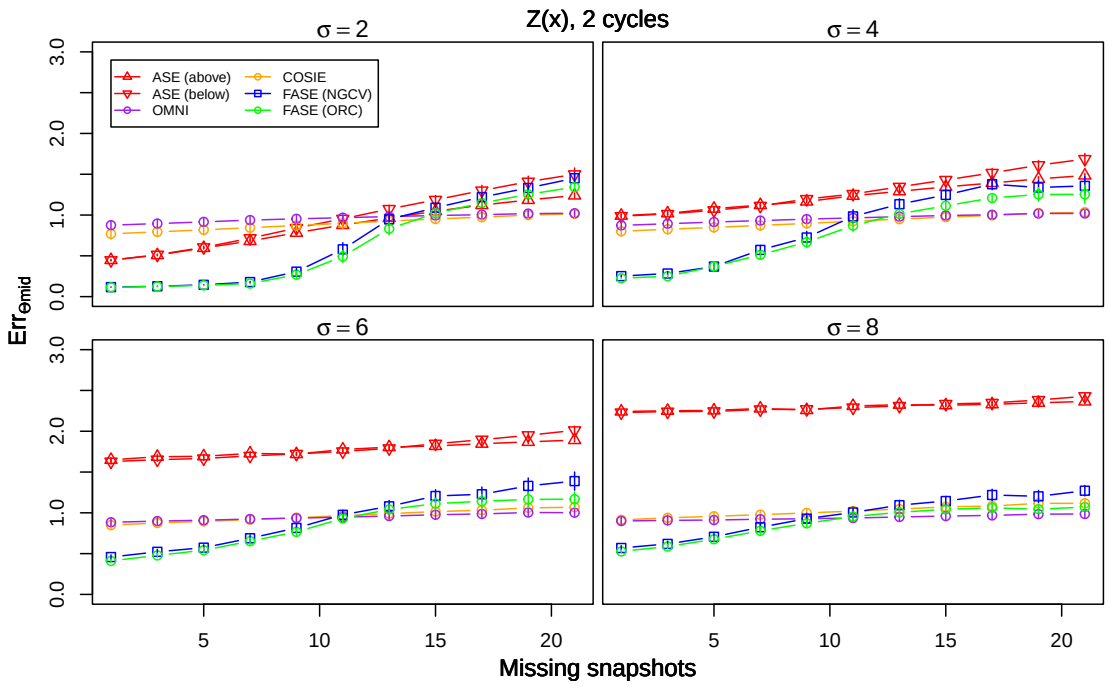


Figure 17: Prediction prediction performance for central unobserved snapshot, latent processes complete two cycles in $[0, 1]$.

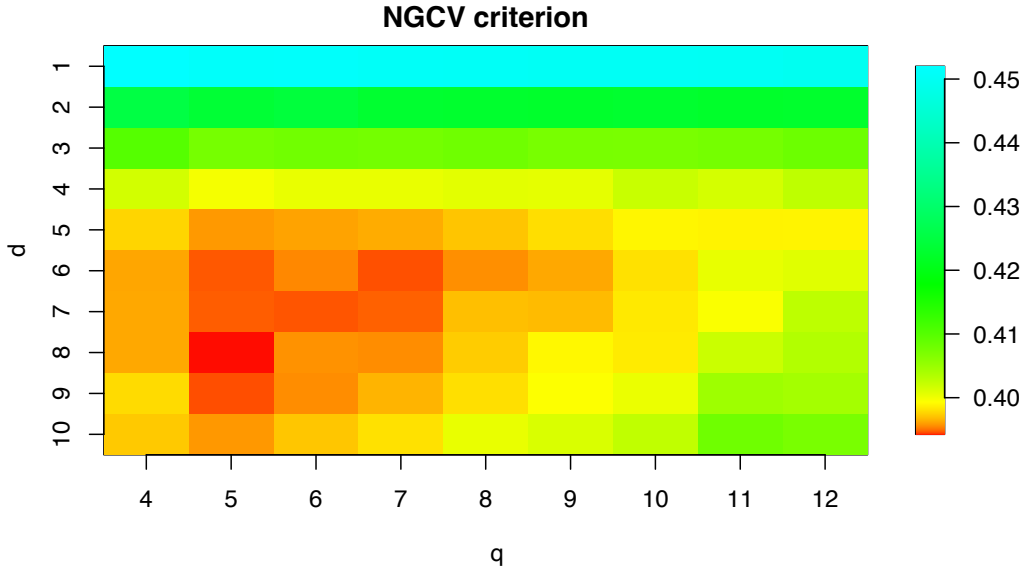


Figure 18: NGCV for FASE estimate, evaluated for each $(d, q)$ pair.

The performance of the NGCV-tuned FASE and the oracle FASE are generally comparable, except in Figure 17 with many missing snapshots and $\sigma \geq 6$. In this case, the oracle approach can choose a smaller $q$ to optimize for imputation, in contrast to the NGCV tuning, which prioritizes model fit on the observed data.

## Appendix D. Additional Analyses of International Relations

In this appendix we include some additional details of the analysis of international political interactions described in Section 6. As described briefly in Section 6, we tune the model parameters $d$ and $q$ by finding a FASE estimate for each pair in a grid, and evaluating NGCV. In particular, we vary $d$ between 1 and 10, incrementing by 1, and vary $q$ between 4 and 12, incrementing by 1. The results are shown in Figure 18. Importantly, we note that the NGCV criterion reaches a minimum on the interior of the grid, supporting the use of a functional embedding on this data, rather than one which is constant over time.

After selecting $\hat{d} = 8$ and $\hat{q} = 5$, we calculate our final estimator $\widehat{Z}$ and apply the sequential Procrustes alignment procedure described in Appendix C. To unambiguously label the latent dimensions from 1 to 8, we evaluate an average magnitude

$$\frac{1}{m}\sum_{k=1}^{m}\sum_{i=1}^{n}\{\hat{z}_{i,r}(x_k)\}^2$$

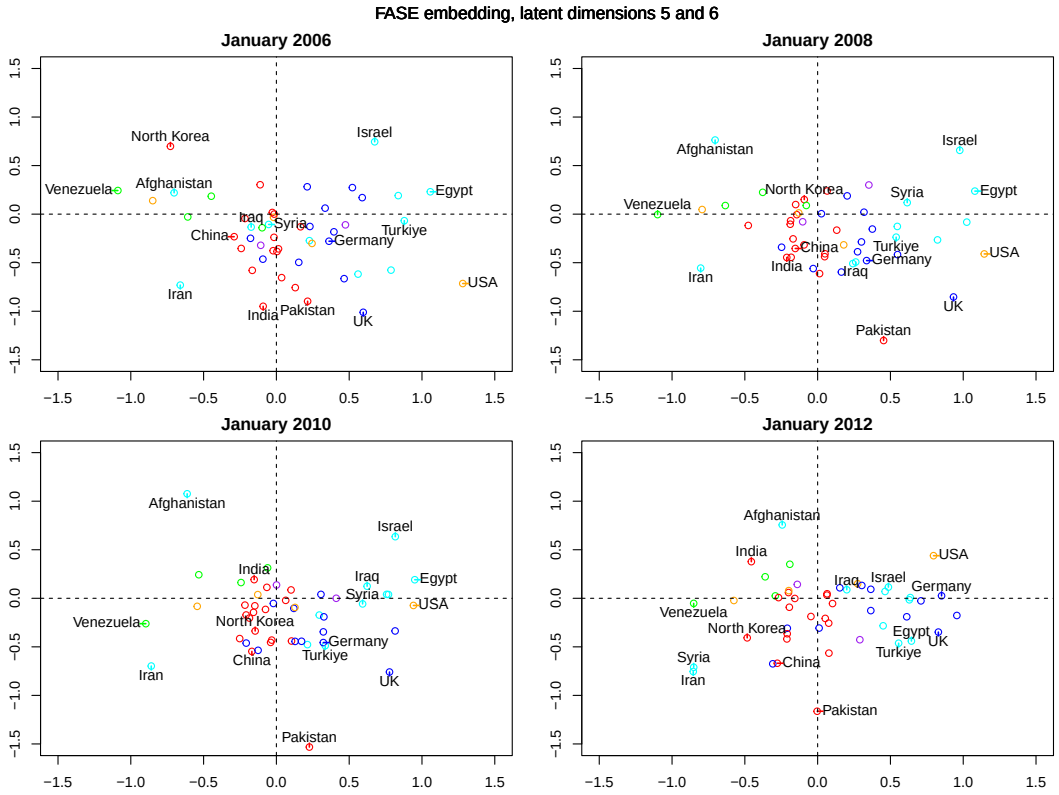


Figure 19: Fifth (horizontal axis) and sixth (vertical axis) dimensions of FASE evaluated at four times: January 2006, January 2008, January 2010, and January 2012. Points are colored by geographical region. Purple: Africa, Red: Asia-Pacific, Blue: Europe, Cyan: Middle East, Orange: North America, Green: South America.

for each $r = 1, \ldots, 8$. The largest average magnitude (dimension 1) is about 46.3, dimensions $2-4$ have smaller average magnitude between 12.9 and 14.8, and the remaining dimensions $5-8$ have average magnitudes between 6.7 and 11.1.

The first four estimated latent dimensions are plotted in Section 6, and we plot the remaining four here, and make some brief remarks on the embeddings. Figure 19 plots the fifth latent dimension against the sixth latent dimension, and Figure 20 plots the seventh latent dimension against the eighth latent dimension. The full evolution of the estimated latent processes can be seen in videos available online at `github.com/peterwmacd/fase/tree/main/videos`.

In Figure 19, the USA and Venezuela are separated at extremes in the fifth dimension, while for much of the time period, the top right and bottom left quadrants separate countries with respect to a conflict between Israel and Iran. As noted in Section 6, the fifth latent coordinate for Syria moves substantially, from the postive to negative half-plane between

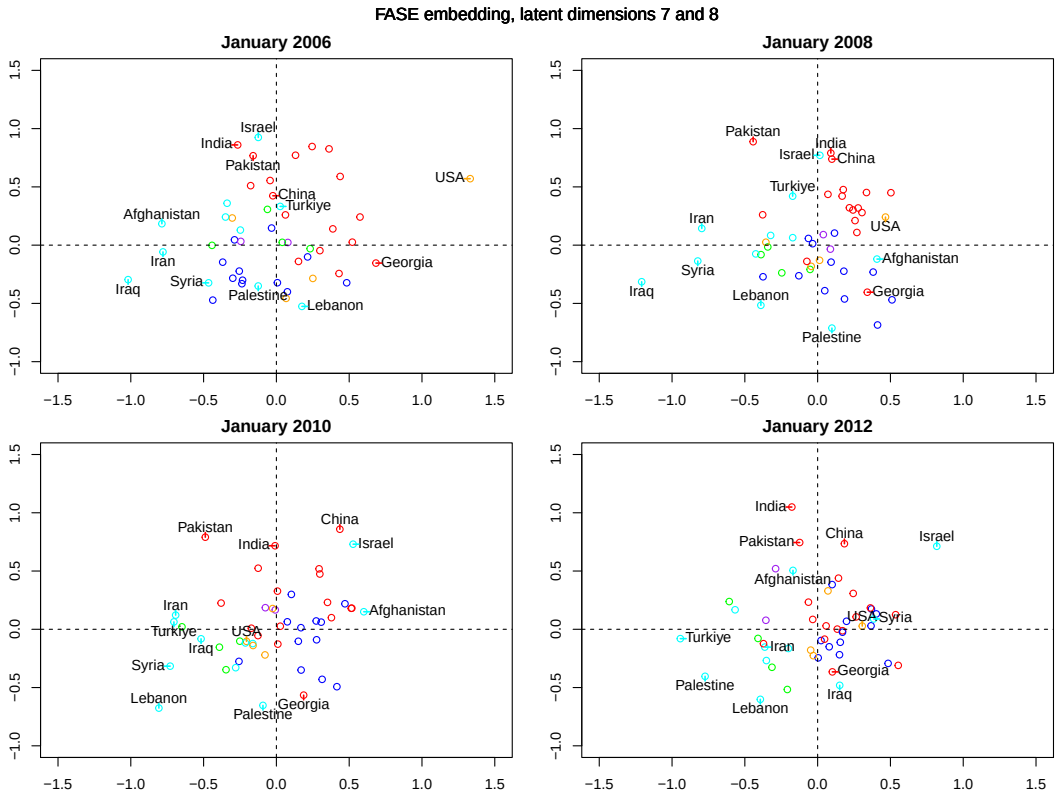


Figure 20: Seventh (horizontal axis) and eighth (vertical axis) dimensions of FASE evaluated at four times: January 2006, January 2008, January 2010, and January 2012. Points are colored by geographical region. Purple: Africa, Red: Asia-Pacific, Blue: Europe, Cyan: Middle East, Orange: North America, Green: South America.

January 2010 and January 2012 (BBC News, 2011). We also see movement, mostly in the sixth latent dimension of Afghanistan and India, reflecting worsening relations with Pakistan during this period (Reuters, 2013).

In Figure 20, we again see a regional clusters formed by countries from Europe and Asia, although these begin to merge by the end of the time period. The seventh dimension separates the USA and Iraq in January 2006 and January 2008, but similar to the conclusion from Figure 8, this conflict appears to have fully dissipated by January 2012, as the two countries have similar seventh latent coordinates, with the same sign.

## Appendix E. FASE with Smoothing Splines

Here we report preliminary results using an extension of the FASE methodology to *smoothing splines*, in which we select a maximal natural spline basis and optimize a penalized objective function.

Briefly, recall that the optimization problem introduced in Section 3 minimizes a loss function $\ell(\mathcal{W})$ over coordinate tensors $\mathcal{W} \in \mathbb{R}^{n\times q\times d}$. The vector-valued function $B(x) \in \mathbb{R}^q$ contains the $B$-spline basis for a $q$-dimensional cubic spline space. We can rewrite this in a functional way if we let $\mathbb{S}_q^{n\times d}$ denote the space of functions from $\mathcal{X}$ to $\mathbb{R}^{n\times d}$ with components in $\text{span}(B)$. Then we can rewrite (3) as

$$\min_{Z\in\mathbb{S}_q^{n\times d}} \left\{ \sum_{k=1}^{m} \|A_k - Z(x_k)Z(x_k)^\intercal\|_F^2 \right\}.$$

Following the usual development for smoothing splines, suppose we instead optimize over $Z$'s with components in the Sobolev space $\text{Sob}_{2,2}^{n\times d}$ of twice-differentiable functions and add a penalty term

$$\text{Pen}(Z) = \sum_{i=1}^{n}\sum_{r=1}^{d} \int_{\mathcal{X}} \left\{ z_{ir}''(x) \right\}^2 dt$$

scaled by a penalty parameter $\lambda \geq 0$. That is we solve the smoothing spline optimization problem

$$\min_{Z\in\text{Sob}_{2,2}^{n\times d}} \left\{ \sum_{k=1}^{m} \|A_k - Z(x_k)Z(x_k)^\intercal\|_F^2 + \lambda\,\text{Pen}(Z) \right\}.$$

Classical results on smoothing splines (Hastie et al., 2001) can be used to justify that this is equivalent to solving

$$\min_{Z\in\mathbb{N}_m^{n\times d}} \left\{ \sum_{k=1}^{m} \|A_k - Z(x_k)Z(x_k)^\intercal\|_F^2 + \lambda\,\text{Pen}(Z) \right\} \tag{27}$$

where $\mathbb{N}_m^{n\times d}$ is the natural cubic spline with knots at the $x_k$'s for $k = 1, \ldots, m$. Hence, we can easily adapt FASE to these settings, solving (27) with gradient descent. The penalty term can be evaluated in terms of integrals of the second derivatives of the natural spline basis functions, and rewritten as a quadratic function of the basis coordinates, hence its gradient is easy to calculate.

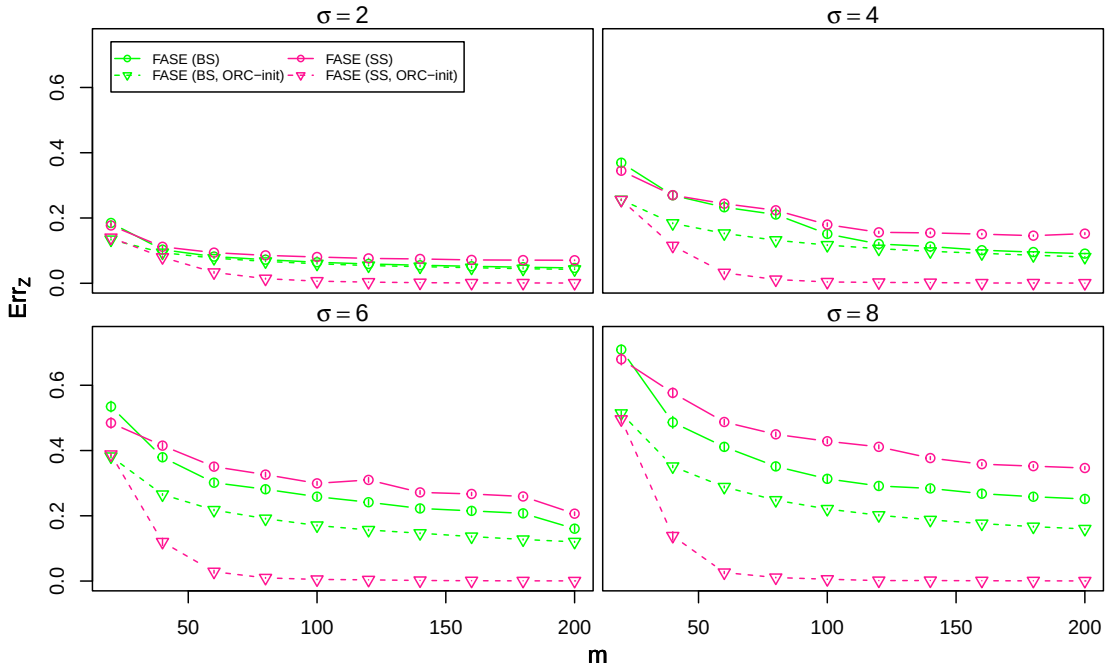


Figure 21: Mean of $\mathrm{Err}_Z$, varying $m$, the number of snapshots. Scenario (ii), nonparametric Gaussian networks. Plots are labeled by edge standard deviation $\sigma$.

We compare this smoothing spline version of FASE (FASE (SS)) to the $B$-spline version developed in the body of the paper (FASE (BS)) through a small simulation study. In particular, we generate functional networks as in scenario (ii) in Section 5, fixing $n = 100$ and varying $m$ from 20 to 200 and $\sigma \in \{2, 4, 6, 8\}$. We perform 50 replications for each setting, and evaluate the mean of $\mathrm{Err}_Z$ (see Section 5). We select the nonparametric scenario (ii) as it should favor the fully nonparametric smoothing spline version of FASE.

For each of FASE (SS) and FASE (BS), we fit an oracle version with $d$ fixed at the ground truth value and $\lambda$ and $q$ respectively selected from a grid to minimize $\mathrm{Err}_Z$. In about 98% of replications, the grids contain a local minimum of $\mathrm{Err}_Z$. We try two initialization routines. First, the usual initializer introduced in Section 3.1. Second, an oracle initialization from the ground truth processes $Z$, or the closest approximation to each component in the $B$-spline space (ORC-init). The results are shown in Figure 21.

In Figure 21, we see that with data-driven initialization, FASE (BS) outperforms FASE (SS) in terms of $\mathrm{Err}_Z$ except in the setting where $m = 20$. This ordering is reversed with oracle initialization. In fact, the performance of FASE (SS) is insensitive to $\sigma$ for large $m$, implying that gradient descent is converging to a local minimum very close to the starting point. This provides evidence that for large $m$ and $n$, as FASE (SS) must optimize far more parameters than FASE (BS), gradient descent becomes unreliable and highly dependent on the starting value. While this does not preclude the existence of an efficient implementation

of FASE (SS) which can overcome these optimization issues, it is not clear that such an implementation would substantially outperform FASE (BS).

## Appendix F. Sequential FASE

In this section, we develop a sequential version of FASE that estimates one latent dimension at a time to overcome some of the identifiability issues. As inputs, the sequential gradient descent algorithm takes a set of initial coordinates

$$\widehat{\mathcal{W}}^0 = \{\widehat{\boldsymbol{W}}_r^0\}_{r=1}^d;$$

step sizes $\eta_{h,r} > 0$, which may depend on the iteration number $h \geq 0$; the latent dimension $d$; and a maximum iteration number $H$. In practice, for both gradient descent schemes we will choose $H$ based on the convergence of the value of $\ell$ to a local minimum.

**Algorithm 3:** Sequential gradient descent algorithm.

Set $\widetilde{\mathcal{W}}^0 = \mathbf{0}_{n\times q\times d}$
For $r = 1$ to $r = d$
  $\widetilde{\boldsymbol{W}}_r^0 \leftarrow \widehat{\boldsymbol{W}}_r^0$
  For $h = 1$ to $h = H$
    $\widetilde{\boldsymbol{W}}_r^h \leftarrow \widetilde{\boldsymbol{W}}_r^{h-1} - \eta_{h-1,r}\frac{\partial \ell}{\partial \boldsymbol{W}_r}(\widetilde{\mathcal{W}}^{h-1})$
  $\widetilde{\boldsymbol{W}}_r^0 \leftarrow \widetilde{\boldsymbol{W}}_r^H$
Output $\widehat{\mathcal{W}}^H = \{\widetilde{\boldsymbol{W}}_r^H\}_{r=1}^d$

The output of Algorithm 3 is an $n \times q \times d$ tensor-valued coordinate estimator $\widehat{\mathcal{W}}^H$. Algorithm 3 computes the coordinate estimator $\widehat{\mathcal{W}}^H$ one $n \times q$ slice at a time by estimating $d$ one-dimensional latent process models sequentially. Once a slice is estimated, it remains fixed, and its contribution is subtracted from the network structure. Slices which have not yet been estimated have all entries fixed at 0.

Empirically, we have found that when the singular values of the true latent processes are separated uniformly (in $x$), this sequential approach can achieve better estimation performance, which we attribute to it essentially reducing the space of unknown orthogonal transformations. Theoretical results can be proven, analogous to Corollary 3, that recover each dimension in sequence with stronger guarantees on the alignment of the estimated and true latent processes. However, extending a result like Corollary 3 to higher dimensions with this sequential estimation scheme requires strong assumptions on the separation of the latent dimensions, in particular that the singular values of each $Z(x)$ are separated uniformly (in $x$), so that the previously estimated or yet to be estimated dimensions do not interfere with the estimation of the current slice. We omit these results, as in general we recommend the concurrent gradient descent estimator for FASE presented in Section 3.

## References

Afghan, Iranian border clash kills Afghan teacher. *Reuters*, 2008.

US ambassador Robert Ford pulled out of Syria. *BBC News*, 2011.

Tensions rise over Afghan, Pakistan border dispute. *Reuters*, 2013.

Makan Arastuie, Subhadeep Paul, and Kevin S. Xu. CHIP: A Hawkes process model for continuous-time networks with scalable and consistent estimation. volume 33, pages 16983–16996, 2020.

Jesus Arroyo and Avanti Athreya. Inference for multiple heterogeneous networks with a common invariant subspace. *The Journal of Machine Learning Research*, 22:49, 2021.

Avanti Athreya, Donniell E. Fishkind, Minh Tang, Carey E. Priebe, Youngser Park, Joshua T. Vogelstein, Keith Levin, Vince Lyzinski, Yichen Qin, and Daniel L. Sussman. Statistical inference on random dot product graphs: a survey. *The Journal of Machine Learning Research*, 18:92, 2018.

Avanti Athreya, Zachary Lubberts, Youngser Park, and Carey E. Priebe. Euclidean mirrors and dynamics in network time series. *Journal of the American Statistical Association*, pages 1–12, 2025.

Sharmodeep Bhattacharyya and Shirshendu Chatterjee. Spectral clustering for multiple sparse networks. *arXiv:1805.10594 [cs, math, stat]*, 2018.

Joshua Cape, Minh Tang, and Carey E. Priebe. The two-to-infinity norm and singular subspace geometry with applications to high-dimensional statistics. *Annals of Statistics*, 47(5):2405–2439, 2019.

Tianyi Chen, Zachary Lubberts, Avanti Athreya, Youngser Park, and Carey E Priebe. Euclidean mirrors and first-order changepoints in network time series. *arXiv:2405.11111 [stat]*, 2024.

Daniele Durante and David B. Dunson. Locally adaptive dynamic networks. *The Annals of Applied Statistics*, 10(4), 2016.

Johan Engvall and Svante E. Cornell. Kazakhstan in Europe: Why Not? Technical report, Institute for Security and Development Policy, 2017.

Jianqing Fan and Qiwei Yao. *Nonlinear Time Series: Nonparametric and Parametric Methods*. Springer, New York, New York, 2003.

Gene H. Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

Trevor Hastie, Jerome Friedman, and Robert Tibshirani. *The Elements of Statistical Learning*. Springer, New York, New York, 2001.

Peter D. Hoff, Adrian E. Raftery, and Mark S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

Bomin Kim, Kevin H. Lee, Lingzhou Xue, and Xiaoyue Niu. A review of dynamic network models with latent variables. *Statistics Surveys*, 12:105–135, 2018.

Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

Alexander Kreiß, Enno Mammen, and Wolfgang Polonik. Nonparametric inference for continuous-time event counting and link-based dynamic network models. *Electronic Journal of Statistics*, 13(2), 2019.

Jennifer Lautenschlager, Steve Shellman, and Michael Ward. ICEWS Event Aggregations, 2015. URL `https://doi.org/10.7910/DVN/28117`.

Nam H. Lee and Carey E. Priebe. A latent process model for time series of attributed random graphs. *Statistical Inference for Stochastic Processes*, 14(3):231–253, 2011.

Keith Levin, Avanti Athreya, Minh Tang, Vince Lyzinski, and Carey E. Priebe. A central limit theorem for an omnibus embedding of multiple random dot product graphs. In *2017 IEEE International Conference on Data Mining Workshops*, pages 964–967, 2017.

Zhuang Ma, Zongming Ma, and Hongsong Yuan. Universal latent space model fitting for large networks with edge covariates. *The Journal of Machine Learning Research*, 21:67, 2020.

Peter W. MacDonald, Elizaveta Levina, and Ji Zhu. Latent space models for multiplex networks with shared structure. *Biometrika*, 109(3):683–706, 2022.

Caterine Matias, Tabea Rebafka, and Fanny Villers. A semiparametric extension of the stochastic block model for longitudinal networks. *Biometrika*, 105(3):665–680, 2018.

Catherine Matias and Vincent Miele. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B*, 79 (4):1119–1141, 2017.

Danielle P. Mersch, Alessandro Crespi, and Laurent Keller. Tracking individuals shows spatial fidelity is a key regulator of ant social organization. *Science*, 340(6136):1090–1093, 2013.

Oscar Hernan Madrid Padilla, Yi Yu, and Carey E Priebe. Change point localization in dependent dynamic nonparametric random dot product graphs. *Journal of Machine Learning Research*, 23(234):1–59, 2022.

Marianna Pensky and Teng Zhang. Spectral clustering in the dynamic stochastic block model. *Electronic Journal of Statistics*, 13(1):678–709, 2019.

Patrick O. Perry and Patrick J. Wolfe. Point process modeling for directed interaction networks. *Journal of the Royal Statistical Society: Series B*, 75(5):821–849, 2013.

Francesco Sanna Passino, Anna S. Bertiger, Joshua C. Neil, and Nicholas A. Heard. Link prediction in dynamic networks using random dot product graphs. volume 35, pages 2168–2199, 2021.

Purnamrita Sarkar and Andrew W. Moore. Dynamic social network analysis using latent space models. *ACM SIGKDD Explorations Newsletter*, 7(2):31–40, 2005.

Larry Schumaker. *Spline Functions: Basic Theory*. Cambridge University Press, Cambridge, UK, 2007.

Daniel K. Sewell and Yuguo Chen. Latent space models for dynamic networks. *Journal of the American Statistical Association*, 110(512):1646–1657, 2015.

Qianhua Shan. *Network Inference with Applications in Neuroimaging*. PhD thesis, University of Michigan, Ann Arbor, Michigan, 2022.

Tom A.B. Snijders. Stochastic actor-oriented models for network dynamics. *Annual Review of Statistics and Its Application*, 4(1):343–363, 2017.

Minh Tang, Daniel L. Sussman, and Carey E. Priebe. Universally consistent vertex classification for latent positions graphs. *The Annals of Statistics*, 41(3), 2013.

Stephen Tu, Ross Boczar, Max Simchowitz, Mahdi Soltanolkotabi, and Ben Recht. Low-rank solutions of linear matrix equations via Procrustes flow. In *International Conference on Machine Learning*, pages 964–973. PMLR, 2016.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, Cambridge, UK, 2018.