Bayesian Analysis (2024) **TBA**, Number TBA, pp. 1–29

Scalable Spatiotemporally Varying Coefficient Modeling with Bayesian Kernelized Tensor Regression

Mengying Lei[*], Aurélie Labbe[†], and Lijun Sun[*,§]

Abstract. As a regression technique in spatial statistics, the spatiotemporally varying coefficient model (STVC) is an important tool for discovering nonstationary and interpretable response-covariate associations over both space and time. However, it is difficult to apply STVC for large-scale spatiotemporal analyses due to its high computational cost. To address this challenge, we summarize the spatiotemporally varying coefficients using a third-order tensor structure and propose to reformulate the spatiotemporally varying coefficient model as a special low-rank tensor regression problem. The low-rank decomposition can effectively model the global patterns of large data sets with a substantially reduced number of parameters. To further incorporate the local spatiotemporal dependencies, we use Gaussian process (GP) priors on the spatial and temporal factor matrices. We refer to the overall framework as Bayesian Kernelized Tensor Regression (BKTR), and kernelized tensor factorization can be considered a new and scalable approach to modeling multivariate spatiotemporal processes with a low-rank covariance structure. For model inference, we develop an efficient Markov chain Monte Carlo (MCMC) algorithm, which uses Gibbs sampling to update factor matrices and slice sampling to update kernel hyperparameters. We conduct extensive experiments on both synthetic and real-world data sets, and our results confirm the superior performance and efficiency of BKTR for model estimation and parameter inference.

Keywords: Gaussian process, tensor regression, Bayesian framework, multivariate spatiotemporal processes, spatiotemporal modeling.

1 Introduction

Local spatial regression aims to characterize the nonstationary and heterogeneous associations between the response variable and the corresponding covariates observed in a spatial domain (Banerjee et al., 2014; Cressie and Wikle, 2015). This is achieved by assuming that the regression coefficients vary locally over space. Local spatial regression offers enhanced interpretability of complex relationships and has become an important technique in many fields, such as geography, ecology, economics, environment, public health and climate science, to name but a few. In general, a local spatial regression

[*] Department of Civil Engineering, McGill University, Montreal, Quebec, H3A 0C3, Canada, mengying.lei@mail.mcgill.ca

[†] Department of Decision Sciences, HEC Montréal, Montreal, Quebec, H3T 2A7, Canada, aurelie.labbe@hec.ca

[*] Department of Civil Engineering, McGill University, Montreal, Quebec, H3A 0C3, Canada, lijun.sun@mcgill.ca

[§] Corresponding author.

https://doi.org/10.1214/24-BA1428

model for a scalar response y can be written as:

$$y(\boldsymbol{s}) = \boldsymbol{x}(\boldsymbol{s})^\top \boldsymbol{\beta}(\boldsymbol{s}) + \epsilon(\boldsymbol{s}), \tag{1.1}$$

where $\boldsymbol{s}$ is the index (e.g., longitude and latitude) for a spatial location, $\boldsymbol{x}(\boldsymbol{s}) \in \mathbb{R}^P$ and $\boldsymbol{\beta}(\boldsymbol{s}) \in \mathbb{R}^P$ are the covariate vector and the regression coefficients at location $\boldsymbol{s}$, respectively, and $\epsilon(\boldsymbol{s}) \sim \text{i.i.d.}\, \mathcal{N}(0, \tau^{-1})$ is a white noise process with precision τ.

There are two common methods for local spatial regression analysis—the Bayesian spatially varying coefficient model (SVC) (Gelfand et al., 2003) and the geographically weighted regression (GWR) (Fotheringham et al., 2003). SVC is a Bayesian hierarchical model in which the regression coefficients are modeled using Gaussian processes (GP) with a kernel function to be learned (Rasmussen and Williams, 2006). For a collection of M observed locations, the original SVC developed by Gelfand et al. (2003) imposes a prior such that $\text{vec}(\boldsymbol{\beta}_{\text{mat}}^\top) \sim \mathcal{N}(\boldsymbol{1}_{M\times 1} \otimes \boldsymbol{\mu}_\beta, \boldsymbol{K}_s \otimes \boldsymbol{\Lambda}^{-1})$, where $\boldsymbol{\beta}_{\text{mat}}$ is a $M \times P$ matrix of all coefficients, $\text{vec}(\boldsymbol{X})$ denotes vectorization by stacking all columns in $\boldsymbol{X}$ as a vector, $\boldsymbol{\mu}_\beta$ represents the overall regression coefficient vector used to construct the mean, $\boldsymbol{K}_s$ is a $M \times M$ spatial correlation matrix, $\boldsymbol{\Lambda}$ is a $P \times P$ precision matrix for covariates, and $\otimes$ denotes the Kronecker product. In this paper, for simplicity, we use a zero-mean GP to specify $\boldsymbol{\beta}$, and the global effect of the covariates can be learned (or removed) through a linear regression term as in Gelfand et al. (2003). In addition, setting $\boldsymbol{K}_s$ as a correlation matrix simplifies the covariance specification since the variance can be captured by scaling $\boldsymbol{\Lambda}^{-1}$. This formulation is equivalent to having a matrix normal distribution $\boldsymbol{\beta}_{\text{mat}} \sim \mathcal{MN}_{M\times P}\left(\boldsymbol{0}, \boldsymbol{K}_s, \boldsymbol{\Lambda}^{-1}\right)$. GWR was developed independently using a local weighted regression approach, in which a bandwidth parameter is used to calculate the weights (based on a weight function) for different observations, with closer observations carrying larger weights. In practice, the bandwidth parameter is either prespecified based on domain knowledge or tuned through cross-validation. However, it has been shown in the literature that the estimation results of GWR are highly sensitive to the selection of the bandwidth parameter (e.g., Finley, 2011). Compared with GWR, the Bayesian hierarchical framework of SVC provides more robust results and allows us to learn the hyperparameters of the spatial kernel $\boldsymbol{K}_s$, e.g., the length-scale, which is critical to understanding the underlying characteristics of the spatial processes. In addition, by using Markov chain Monte Carlo (MCMC), we can not only explore the posterior distribution of the kernel hyperparameters and regression coefficients, but also perform out-of-sample prediction with uncertainty quantification.

The formulation in Eq. (1.1) can be easily extended to local spatiotemporal regression to further characterize the temporal variation of the coefficients. For a response matrix $\boldsymbol{Y} \in \mathbb{R}^{M\times N}$ observed from a set of locations $S = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M\}$ over a set of time points $T = \{t_1, \ldots, t_N\}$, the local spatiotemporal regression model defined on the Cartesian product $S \times T = \{(\boldsymbol{s}_m, t_n) : m = 1, \ldots, M,\ n = 1, \ldots, N\}$ can be formulated as:

$$y\left(\boldsymbol{s}_m, t_n\right) = \boldsymbol{x}\left(\boldsymbol{s}_m, t_n\right)^\top \boldsymbol{\beta}\left(\boldsymbol{s}_m, t_n\right) + \epsilon\left(\boldsymbol{s}_m, t_n\right), \tag{1.2}$$

where we use $m = 1, \ldots, M$ and $n = 1, \ldots, N$ to index rows (i.e., location) and columns (i.e., time point), respectively, $y\left(\boldsymbol{s}_m, t_n\right)$ is the (m, n)th element in $\boldsymbol{Y}$, and $\boldsymbol{x}\left(\boldsymbol{s}_m, t_n\right)$ and $\boldsymbol{\beta}\left(\boldsymbol{s}_m, t_n\right)$ are the covariate vector and coefficient vector at location $\boldsymbol{s}_m$

and time t_n, respectively. Based on this formulation, Huang et al. (2010) extended GWR to geographically and temporally weighted regression (GTWR) by introducing more parameters to quantify spatiotemporal weights in the locally weighted regression. For SVC, Gelfand et al. (2003) suggested using a separable kernel structure to build a spatiotemporally varying coefficient model (STVC), which assumes that $[\boldsymbol{\beta}(\boldsymbol{s}_1,t_1);\ldots;\boldsymbol{\beta}(\boldsymbol{s}_M,t_1);\boldsymbol{\beta}(\boldsymbol{s}_1,t_2);\ldots;\boldsymbol{\beta}(\boldsymbol{s}_M,t_N)] \sim \mathcal{N}(\mathbf{0},\boldsymbol{K}_t\otimes\boldsymbol{K}_s\otimes\boldsymbol{\Lambda}^{-1})$, where $\boldsymbol{K}_t$ is a $N\times N$ kernel matrix defining the correlation structure for the N time points. Note that with this GP formulation, it is not necessary for the N time points to be equally spaced. If we parameterize the regression coefficients in Eq. (1.2) as a third-order tensor $\boldsymbol{\mathcal{B}}\in\mathbb{R}^{M\times N\times P}$ with mode-3 fiber $\boldsymbol{\mathcal{B}}(m,n,:)=\boldsymbol{\beta}(\boldsymbol{s}_m,t_n)$, the above specification is equivalent to having a tensor normal distribution $\boldsymbol{\mathcal{B}}\sim\mathcal{TN}_{M\times N\times P}\left(\mathbf{0},\boldsymbol{K}_s,\boldsymbol{K}_t,\boldsymbol{\Lambda}^{-1}\right)$. However, despite the elegant separable kernel-based formulation in STVC, the model is rarely used in real-world practice mainly due to the high computational cost. For example, for a fully observed matrix $\boldsymbol{Y}$ with corresponding spatiotemporal covariates, updating the coefficients $\boldsymbol{\beta}$ in each MCMC iteration requires time complexity of $\mathcal{O}\left(M^3N^3P^3\right)$. Updating the kernel hyperparameters can be achieved by integrating out $\boldsymbol{\beta}$, but it still requires $\mathcal{O}\left(M^3N^3\right)$ in time.

In this paper, we provide an alternative estimation strategy—Bayesian Kernelized Tensor Regression (BKTR)—to perform Bayesian spatiotemporal regression analysis on large-scale data sets. Inspired by the idea of low-rank regression and tensor regression (see e.g., Izenman, 1975; Cressie and Johannesson, 2008; Banerjee et al., 2008; Zhou et al., 2013; Bahadori et al., 2014; Guhaniyogi et al., 2017), we use low-rank tensor factorization to encode the dependencies among the three dimensions in $\boldsymbol{\mathcal{B}}$ with only a few latent factors. To further incorporate local spatial and temporal dependencies, we use GP priors on the spatial and temporal factor vectors following Lopes et al. (2008) and Luttinen and Ilin (2009), thus translating the default tensor factorization into a kernelized factorization model. With a specified tensor rank R, the time complexity becomes $\mathcal{O}\left(R^3\left(M^3+N^3+P^3\right)\right)$, which is substantially reduced compared with the default STVC formulation. In addition to the spatial and temporal framework, we also consider the case where a proportion of the response matrix $\boldsymbol{Y}$ can be unobserved or corrupted, given observed values of the covariates $\boldsymbol{X}$. Such a scenario is very common in many real-world applications, such as traffic state data collected from emerging crowdsourcing and moving sensing systems (e.g., Google Waze) for example, where observations are inherently sparse in space and time. We show that the underlying Bayesian tensor decomposition structure allows us to effectively estimate both the model coefficients and the unobserved outcomes even when the missing rate of $\boldsymbol{Y}$ is high. We conduct numerical experiments on both synthetic and real-world data sets, and our results confirm the promising performance of BKTR.

2 Related work

The key computational challenge in SVC/STVC is how to efficiently and effectively learn a multivariate spatial/spatiotemporal process (i.e., $\boldsymbol{\beta}_{\text{mat}}$ in Eq. (1.1) and the third-order spatiotemporal tensor $\boldsymbol{\mathcal{B}}$ in Eq. (1.2)). For a general multivariate spatial process that is fully observed on the Cartesian product with white noise, a popular approach is to use separable covariance specification on which one can leverage the property of Kronecker products to substantially reduce the computational cost (Saatçi, 2012; Wilson et al., 2014). However, for SVC, we cannot benefit directly from the Kronecker property since the data $\boldsymbol{y}$ is obtained through a linear transformation of $\boldsymbol{\beta}_{\text{mat}}$. In this case, computing the inverse of an $MP \times MP$ matrix becomes inevitable when sampling these spatially varying coefficients. Existing frameworks for SVC essentially adopt a two-step approach (Gelfand et al., 2003; Finley and Banerjee, 2020): (1) update only kernel hyperparameters and $\boldsymbol{\mu}$ by marginalizing $\boldsymbol{\beta}$ with cost $\mathcal{O}\left(M^3\right)$; and (2) after burn-in, use composition sampling on the obtained MCMC samples to generate samples for $\boldsymbol{\beta}$ with cost $\mathcal{O}\left(M^3P^3\right)$; see Finley and Banerjee (2020) for a detailed implementation of Bayesian SVC. For STVC, the corresponding costs in the two steps are $\mathcal{O}\left(M^3N^3\right)$ and $\mathcal{O}\left(M^3N^3P^3\right)$, respectively. The high computational cost in step (2) is the primary issue that limits the application of SVC/STVC in practice.

Our work follows a different approach. Instead of modeling $\boldsymbol{\beta}$ directly using a GP, we parameterize the third-order tensor $\boldsymbol{\mathcal{B}}$ for STVC using a low-rank tensor decomposition (Kolda and Bader, 2009). The idea is inspired by recent studies on low-rank tensor regression/learning (see e.g., Lopes et al., 2008; Zhou et al., 2013; Bahadori et al., 2014; Rabusseau and Kadri, 2016; Yu and Liu, 2016; Guhaniyogi et al., 2017; Yu et al., 2018). The low-rank assumption not only preserves the global patterns and higher-order dependencies in the variable, but also greatly reduces the number of parameters. In fact, without considering spatiotemporal indices, we can formulate Eq. (1.2) as a scalar-tensor regression problem (Guhaniyogi et al., 2017) by reconstructing each $\boldsymbol{x}(\boldsymbol{s}_m, t_n)$ as a sparse covariate tensor of the same size as $\boldsymbol{\mathcal{B}}$. However, for spatiotemporal data, the low-rank assumption alone cannot fully characterize the strong local spatial and temporal consistency. To better encode local spatial and temporal consistency, existing studies in tensor regression have introduced graph Laplacian regularization in defining the loss function (e.g., Bahadori et al., 2014; Rao et al., 2015) in an optimization framework. Nevertheless, this approach also introduces more parameters (e.g., those used to define the distance/similarity function and weights in the loss function) and without a Bayesian hierarchical specification it has limited power in modeling complex spatial and temporal processes. The most relevant work is a Gaussian process factor analysis model (Luttinen and Ilin, 2009) developed for a completely different problem—completing a spatiotemporal matrix observed from M locations over N time points, in which different GP priors are assumed on the spatial and temporal factors, and the whole model is learned through variational Bayesian inference. Similarly, Lopes et al. (2008) developed a spatial dynamic factor model in which spatial factors are assumed to have GP priors and temporal factors follow a dynamic linear model. Lei et al. (2022) presents an MCMC scheme for this model, in which slice sampling is used to update kernel hyperparameters and Gibbs sampling is used to update factor matrices. We follow a similar idea as in Luttinen and Ilin (2009) and Lei et al. (2022) to parameterize the coefficients $\boldsymbol{\beta}$ and develop MCMC algorithms for model inference. In a recent work, Zhang and Banerjee (2022) also proposed to use a Bayesian Linear Model of Coregionalization (LMC) factor model to model high-dimensional multivariate spatial processes involving both M and P. To solve the large M issue, the Nearest Neighbor Gaussian Process (NNGP) (Datta et al., 2016; Finley et al., 2019) is used to model spatial factors. In the literature, there also exist other parameterization methods to model spatial processes/coefficients with multidimensional structures. For instance, Martinez-Beneito et al. (2017) used Kronecker decomposition to model a large coefficient matrix for tensor-variate data; Guhaniyogi et al. (2023) proposed to model spatially varying coefficients using basis-function models (with pre-defined basis functions and random basis coefficients) and used sketching to reduce data dimensionality to achieve scalable inference for large M. BKTR can also be considered to use tensor factorization as a special basis function method to model $\boldsymbol{\mathcal{B}}$, where the basis functions are also random variables (see Lei and Sun (2023) and Cressie et al. (2022)).

3 Bayesian Kernelized tensor regression

3.1 Preliminaries

Notations Throughout this paper, we use lowercase letters to denote scalars, e.g., x, boldface lowercase letters to denote vectors, e.g., $\boldsymbol{x} \in \mathbb{R}^M$, and boldface uppercase letters to denote matrices, e.g., $\boldsymbol{X} \in \mathbb{R}^{M \times N}$. The ℓ_2-norm of $\boldsymbol{x}$ is defined as $\|\boldsymbol{x}\|_2 = \sqrt{\sum_m x_m^2}$. For a vector $\boldsymbol{x}$, we denote its mth entry by $\boldsymbol{x}(m)$. For a matrix $\boldsymbol{X} \in \mathbb{R}^{M \times N}$, we denote its (m,n)th entry by $x_{m,n}$ or $\boldsymbol{X}(m,n)$. We use $\boldsymbol{I}_N$ to denote an identity matrix of size $N \times N$. Given two matrices $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ and $\boldsymbol{B} \in \mathbb{R}^{P \times Q}$, the Kronecker product is defined as $\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{1,1}\boldsymbol{B} & \cdots & a_{1,N}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{M,1}\boldsymbol{B} & \cdots & a_{M,N}\boldsymbol{B} \end{bmatrix} \in \mathbb{R}^{MP \times NQ}$. If $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N]$ and $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_Q]$ have the same number of columns, i.e., $N = Q$, then the Khatri-Rao product is defined as the column-wise Kronecker product $\boldsymbol{A} \odot \boldsymbol{B} = [\boldsymbol{a}_1 \otimes \boldsymbol{b}_1, \ldots, \boldsymbol{a}_N \otimes \boldsymbol{b}_N] \in \mathbb{R}^{MP \times N}$. The vectorization $\mathrm{vec}(\boldsymbol{X})$ stacks all column vectors in $\boldsymbol{X}$ as a single vector. Following the tensor notation in Kolda and Bader (2009), we denote a third-order tensor by $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M \times N \times P}$ and its mode-k $(k = 1, 2, 3)$ unfolding by $\boldsymbol{X}_{(k)}$, which maps a tensor into a matrix. The mode-3 fibers and frontal slices of $\boldsymbol{\mathcal{X}}$ are denoted by $\boldsymbol{\mathcal{X}}(m, n, :) \in \mathbb{R}^p$ and $\boldsymbol{\mathcal{X}}(:, :, p) \in \mathbb{R}^{M \times N}$, respectively. Lastly, we use $\mathrm{ones}(M, N)$ and $\boldsymbol{1}_M \in \mathbb{R}^M$ to represent a $M \times N$ matrix and a length M column vector of ones, respectively.

Tensor CP decomposition For a third-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{M \times N \times P}$, the CANDECOMP/PARAFAC (CP) decomposition factorizes $\boldsymbol{\mathcal{A}}$ into a sum of rank-one tensors (Kolda and Bader, 2009):

$$\boldsymbol{\mathcal{A}} = \sum_{r=1}^{R} \boldsymbol{u}_r \circ \boldsymbol{v}_r \circ \boldsymbol{w}_r, \tag{3.1}$$

where R is the CP rank, $\circ$ represents the outer product, $\boldsymbol{u}_r \in \mathbb{R}^M$, $\boldsymbol{v}_r \in \mathbb{R}^N$, and $\boldsymbol{w}_r \in \mathbb{R}^P$ for $r = 1, \ldots, R$. The factor matrices that combine the vectors from the rank-one components are denoted by $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_R] \in \mathbb{R}^{M \times R}$, $\boldsymbol{V} \in \mathbb{R}^{N \times R}$, and $\boldsymbol{W} \in \mathbb{R}^{P \times R}$, respectively. We can write Eq. (3.1) in the following matricized form:

$$\boldsymbol{A}_{(1)} = \boldsymbol{U}\left(\boldsymbol{W} \odot \boldsymbol{V}\right)^\top, \ \boldsymbol{A}_{(2)} = \boldsymbol{V}\left(\boldsymbol{W} \odot \boldsymbol{U}\right)^\top, \ \boldsymbol{A}_{(3)} = \boldsymbol{W}\left(\boldsymbol{V} \odot \boldsymbol{U}\right)^\top, \tag{3.2}$$

where $\odot$ is the Khatri-Rao product. Eq. (3.2) relates the mode-k unfolding of a tensor to its polyadic decomposition.

3.2 Model specification

Let $\boldsymbol{\mathcal{X}}$ be an $M \times N \times P$ tensor, of which the (m,n)th mode-3 fiber is the covariate vector at location $\boldsymbol{s}_m$ and time t_n, i.e., $\boldsymbol{\mathcal{X}}(m,n,:) = \boldsymbol{x}\left(\boldsymbol{s}_m, t_n\right)$. For example, in the application of spatiotemporal modeling on bike-sharing demand that we illustrate later (see Section 5), the response matrix $\boldsymbol{Y} \in \mathbb{R}^{M\times N}$ is a matrix of daily departure trips for M bike stations over N days, and the tensor variable $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M\times N\times P}$ represents a set of P spatiotemporal covariates for the corresponding locations and time. Using $\boldsymbol{y} \in \mathbb{R}^{MN}$ to denote $\operatorname{vec}(\boldsymbol{Y})$, Eq. (1.2) can be formulated as:

$$\boldsymbol{y} = \left(\boldsymbol{I}_{MN} \odot \boldsymbol{X}_{(3)}\right)^{\top} \operatorname{vec}\left(\boldsymbol{B}_{(3)}\right) + \boldsymbol{\epsilon}, \tag{3.3}$$

where $\boldsymbol{X}_{(3)}$ and $\boldsymbol{B}_{(3)}$ are the unfoldings of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{B}}$, respectively, the Khatri-Rao product $\left(\boldsymbol{I}_{MN} \odot \boldsymbol{X}_{(3)}\right)^{\top}$ is a $MN \times MNP$ block diagonal matrix, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \tau^{-1}\boldsymbol{I}_{MN})$. Assuming that

$$\boldsymbol{\mathcal{B}} = \sum_{r=1}^{R} \boldsymbol{u}_r \circ \boldsymbol{v}_r \circ \boldsymbol{w}_r \tag{3.4}$$

admits a CP decomposition with rank $R \ll \min\{M, N\}$, we can rewrite Eq. (3.3) as:

$$\boldsymbol{y} = \tilde{\boldsymbol{X}} \operatorname{vec}\left(\boldsymbol{W}(\boldsymbol{V} \odot \boldsymbol{U})^{\top}\right) + \boldsymbol{\epsilon}, \tag{3.5}$$

where $\tilde{\boldsymbol{X}} = \left(\boldsymbol{I}_{MN} \odot \boldsymbol{X}_{(3)}\right)^{\top}$ denotes the expanded covariate matrix. The number of parameters in (3.5) is $R(M+N+P)$, which is substantially less than MNP in (3.3).

Local spatial and temporal processes are critical to the modeling of spatiotemporal data. However, as mentioned above, the low-rank assumption alone cannot encode such local dependencies. To address this issue, we assume specific GP priors on $\boldsymbol{U}$ and $\boldsymbol{V}$ following the GP factor analysis strategy (Luttinen and Ilin, 2009), use a conjugate normal prior on $\boldsymbol{W}$, and then develop a fully Bayesian approach to estimate the model in Eq. (3.5). Figure 1 illustrates the proposed framework, which is referred to as *Bayesian Kernelized Tensor Regression* (BKTR) in the remainder of this paper. The graphical model of BKTR is shown in Figure 2.

As mentioned in the introduction, in real-world applications the dependent data is often partially observed on a set Ω of observation indices, with $|\Omega| < MN$. This means that we only observe a subset of entries $y_{m,n}$, for $\forall(s_m, t_m) \in \Omega$. We denote by $\boldsymbol{D} \in \mathbb{R}^{M\times N}$ a binary indicator matrix with $d_{m,n} = 1$ if $(m,n) \in \Omega$ and $d_{m,n} = 0$ otherwise, and by $\boldsymbol{O}$ a binary matrix of $|\Omega| \times MN$ formed by removing the rows corresponding to the zero values in $\operatorname{vec}(\boldsymbol{D})$ from a $MN \times MN$ identity matrix. The vector of observed data can be obtained by $\boldsymbol{y}_{\Omega} = \boldsymbol{O}\boldsymbol{y}$. Therefore, we have:

$$\boldsymbol{y}_{\Omega} \sim \mathcal{N}\left(\boldsymbol{O}\left(\tilde{\boldsymbol{X}} \operatorname{vec}\left(\boldsymbol{W}(\boldsymbol{V} \odot \boldsymbol{U})^{\top}\right)\right), \tau^{-1}\boldsymbol{I}_{|\Omega|}\right). \tag{3.6}$$

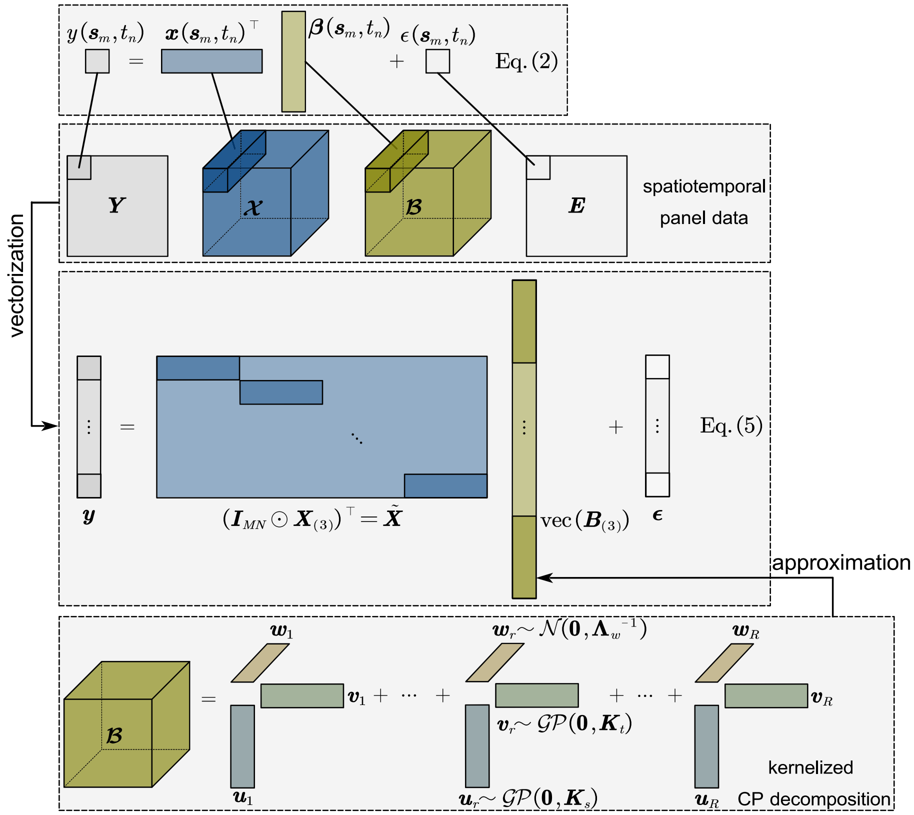

$y(\boldsymbol{s}_m,t_n)$
$\boldsymbol{x}(\boldsymbol{s}_m,t_n)^\top$
$\boldsymbol{\beta}(\boldsymbol{s}_m,t_n)$
$\epsilon(\boldsymbol{s}_m,t_n)$
=
+
Eq. (2)
$\boldsymbol{Y}$
$\boldsymbol{\mathcal{X}}$
$\boldsymbol{\mathcal{B}}$
$\boldsymbol{E}$
spatiotemporal panel data
vectorization
=
+
Eq. (5)
$\boldsymbol{y}$
$(\boldsymbol{I}_{MN}\odot\boldsymbol{X}_{(3)})^\top=\tilde{\boldsymbol{X}}$
$\operatorname{vec}(\boldsymbol{B}_{(3)})$
$\boldsymbol{\epsilon}$
approximation
$\boldsymbol{\mathcal{B}}$
=
$\boldsymbol{w}_1$
$\boldsymbol{v}_1$ + ⋯ +
$\boldsymbol{u}_1$
$\boldsymbol{w}_r\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{\Lambda}_w^{-1})$
$\boldsymbol{v}_r\sim\mathcal{GP}(\boldsymbol{0},\boldsymbol{K}_t)$
$\boldsymbol{u}_r\sim\mathcal{GP}(\boldsymbol{0},\boldsymbol{K}_s)$
+ ⋯ +
$\boldsymbol{w}_R$
$\boldsymbol{v}_R$
$\boldsymbol{u}_R$
kernelized CP decomposition

Figure 1: Illustration of the proposed BKTR framework.

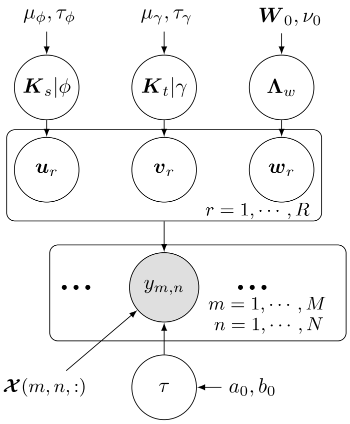


Figure 2: Graphical model of BKTR.

For spatial and temporal factor matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, we use identical GP priors on the component vectors:

$$\begin{aligned} \boldsymbol{u}_r &\sim \mathcal{GP}\left(\mathbf{0}, \boldsymbol{K}_s\right), \ r=1,\ldots,R, \\ \boldsymbol{v}_r &\sim \mathcal{GP}\left(\mathbf{0}, \boldsymbol{K}_t\right), \ r=1,\ldots,R, \end{aligned} \tag{3.7}$$

where $\boldsymbol{K}_s \in \mathbb{R}^{M\times M}$ and $\boldsymbol{K}_t \in \mathbb{R}^{N\times N}$ are the spatial and temporal covariance matrices built from two valid kernel functions $k_s(\boldsymbol{s}_m, \boldsymbol{s}_{m'}; \phi)$ and $k_t(t_n, t_{n'}; \gamma)$, respectively, with ϕ and γ being kernel length-scale hyperparameters. Note that we capture the variance through $\boldsymbol{W}$ and thus restrict $\boldsymbol{K}_s$ and $\boldsymbol{K}_t$ to being correlation matrices by setting the variance to one. We reparameterize the kernel hyperparameters as log-transformed variables to ensure their positivity and assume normal priors on them, i.e., $\log(\phi) \sim \mathcal{N}(\mu_\phi, \tau_\phi^{-1})$, $\log(\gamma) \sim \mathcal{N}(\mu_\gamma, \tau_\gamma^{-1})$. For the factor matrix $\boldsymbol{W}$, we assume all columns follow an identical zero-mean Gaussian distribution with a conjugate Wishart prior on the precision matrix:

$$\begin{aligned} \boldsymbol{w}_r &\sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Lambda}_w^{-1}\right), \ r=1,\ldots,R, \\ \boldsymbol{\Lambda}_w &\sim \mathcal{W}\left(\boldsymbol{\Psi}_0, \nu_0\right), \end{aligned} \tag{3.8}$$

where $\boldsymbol{\Psi}_0$ is a $P\times P$ positive-definite scale matrix and $\nu_0 > P-1$ denotes the degrees of freedom. Finally, we use a conjugate Gamma prior $\tau \sim \text{Gamma}(a_0, b_0)$ on the noise precision τ defined in Eq. (3.6).

Based on the assumed priors and hyperpriors, we can write the covariance function of the coefficients $\boldsymbol{\mathcal{B}}$ modeled with BKTR. Specifically, consider a data pair in the input space with the indices (m,n,p) and (m',n',p'), the covariance between the two entries is

$$\begin{aligned} &\operatorname{Cov}\left(\boldsymbol{\mathcal{B}}\left(m,n,p\right), \boldsymbol{\mathcal{B}}\left(m',n',p'\right)\right) \\ &\quad = \operatorname{Cov}\left(\sum_{r=1}^{R} \boldsymbol{u}_r\left(m\right)\boldsymbol{v}_r\left(n\right)\boldsymbol{w}_r\left(p\right), \sum_{r'=1}^{R} \boldsymbol{u}_{r'}\left(m'\right)\boldsymbol{v}_{r'}\left(n'\right)\boldsymbol{w}_{r'}\left(p'\right)\right). \end{aligned} \tag{3.9}$$

Given the prior on $\boldsymbol{w}_r$ (see Eq. (3.8)), we have

$$\begin{cases} \operatorname{Cov}\left(\boldsymbol{w}_r\left(p\right), \boldsymbol{w}_{r'}\left(p'\right)\right) = 0, \ \forall r \neq r', \ \forall p, p' \ \text{(columns are independent)} \\ \operatorname{Cov}\left(\boldsymbol{w}_r\left(p\right), \boldsymbol{w}_r\left(p'\right)\right) = \boldsymbol{\Lambda}_w^{-1}\left(p,p'\right). \end{cases} \tag{3.10}$$

Integrating out $\boldsymbol{w}_r$ in Eq. (3.4) using Eqs. (3.9) and (3.10), we have

$$\operatorname{vec}\left(\boldsymbol{B}_{(3)}\right) \mid \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Lambda}_w \sim \mathcal{N}\left(\mathbf{0}, \left(\sum_{r=1}^{R}\left(\boldsymbol{v}_r\boldsymbol{v}_r^\top\right)\otimes\left(\boldsymbol{u}_r\boldsymbol{u}_r^\top\right)\right)\otimes\boldsymbol{\Lambda}_w^{-1}\right). \tag{3.11}$$

Similarly, if we marginalize $\boldsymbol{U}$ in Eq. (3.4), we can derive

$$\operatorname{vec}\left(\boldsymbol{B}_{(1)}\right)\big| \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{K}_s \sim \mathcal{N}\left(\mathbf{0}, \left(\sum_{r=1}^{R}\left(\boldsymbol{w}_r\boldsymbol{w}_r^\top\right)\otimes\left(\boldsymbol{v}_r\boldsymbol{v}_r^\top\right)\right)\otimes\boldsymbol{K}_s\right). \tag{3.12}$$

With the analysis of the variance-covariance matrix in Eqs. (3.11) and (3.12), we can interpret the kernelized tensor CP factorization as a higher-order extension of the intrinsic model of coregionalization (Bonilla et al., 2007; Banerjee et al., 2014; Álvarez et al., 2012), where the coregionalization matrix also has a low-rank specification. However, for model inference we do not use the covariance structure; instead, we directly use tensor factorization to learn the latent factor matrices for fast inference.

3.3 Model inference

We use Gibbs sampling to estimate the model parameters, including coefficient factors $\{\boldsymbol{U},\boldsymbol{V},\boldsymbol{W}\}$, the precision τ, and the precision matrix $\boldsymbol{\Lambda}_w$. For the kernel hyperparameters $\{\phi,\gamma\}$ whose conditional distributions are not easy to sample from, we use the slice sampler.

Sampling the coefficient factor matrices $\{\boldsymbol{U},\boldsymbol{V},\boldsymbol{W}\}$

Sampling the factor matrices can be considered as a Bayesian linear regression problem. Taking $\boldsymbol{W}$ as an example, we can rewrite Eq. (3.6) as:

$$\boldsymbol{y}_\Omega \sim \mathcal{N}\left(\boldsymbol{O}\left(\tilde{\boldsymbol{X}}\left((\boldsymbol{V}\odot\boldsymbol{U})\otimes\boldsymbol{I}_P\right)\operatorname{vec}(\boldsymbol{W})\right),\tau^{-1}\boldsymbol{I}_{|\Omega|}\right), \tag{3.13}$$

where $\boldsymbol{U},\boldsymbol{V}$ are known and $\operatorname{vec}(\boldsymbol{W})$ is the coefficient to estimate. Considering that the priors of each component vector $\boldsymbol{w}_r$ are independent and identical, the prior distribution over the whole vectorized $\boldsymbol{W}$ becomes $\operatorname{vec}(\boldsymbol{W})\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_R\otimes\boldsymbol{\Lambda}_w^{-1})$. Since both likelihood and prior of $\operatorname{vec}(\boldsymbol{W})$ follow Gaussian distributions, its posterior is also Gaussian with mean $\boldsymbol{\mu}_W^*$ and precision $\boldsymbol{\Lambda}_W^*$, such as:

$$\boldsymbol{\Lambda}_W^* = \tau\boldsymbol{H}_W^\top\boldsymbol{H}_W + \boldsymbol{I}_R\otimes\boldsymbol{\Lambda}_w,\ \boldsymbol{\mu}_W^* = \tau(\boldsymbol{\Lambda}_W^*)^{-1}\left(\boldsymbol{H}_W^\top\boldsymbol{y}_\Omega\right), \tag{3.14}$$

where $\boldsymbol{H}_W = \boldsymbol{O}\left(\tilde{\boldsymbol{X}}\left((\boldsymbol{V}\odot\boldsymbol{U})\otimes\boldsymbol{I}_P\right)\right)$ with size $|\Omega|\times RP$. Sampling from $\mathcal{N}\left(\boldsymbol{\mu}_W^*,\boldsymbol{\Lambda}_W^*\right)$ is mainly dominated by the Cholesky decomposition of $\boldsymbol{\Lambda}_W^*$, and the procedure requires $\mathcal{O}(R^2P^2)$ in storage and $\mathcal{O}(R^3P^3)$ in time. The posterior distributions of $\boldsymbol{U}$ and $\boldsymbol{V}$ can be obtained similarly using different tensor unfoldings. In order to sample $\boldsymbol{U}$, we use the mode-1 unfolding in Eq. (3.2) and reconstruct the regression model with $\operatorname{vec}(\boldsymbol{U})$ as coefficients:

$$\boldsymbol{y}_\Omega = \boldsymbol{O}\left(\tilde{\boldsymbol{X}}_U\left((\boldsymbol{W}\odot\boldsymbol{V})\otimes\boldsymbol{I}_M\right)\operatorname{vec}(\boldsymbol{U})\right)+\boldsymbol{\epsilon}_\Omega, \tag{3.15}$$

where $\tilde{\boldsymbol{X}}_U = \left(\boldsymbol{X}_{(3)}\odot\boldsymbol{I}_{MN}\right)^\top\in\mathbb{R}^{MN\times MNP}$ and $\boldsymbol{\epsilon}_\Omega\sim\mathcal{N}\left(\boldsymbol{0},\tau^{-1}\boldsymbol{I}_{|\Omega|}\right)$. Thus, the posterior of $\operatorname{vec}(\boldsymbol{U})$ has a closed form—a Gaussian distribution with mean $\boldsymbol{\mu}_U^*$ and precision $\boldsymbol{\Lambda}_U^*$, where

$$\boldsymbol{\Lambda}_U^* = \tau\boldsymbol{H}_U^\top\boldsymbol{H}_U + \boldsymbol{K}_U^{-1},\ \boldsymbol{\mu}_U^* = \tau(\boldsymbol{\Lambda}_U^*)^{-1}\left(\boldsymbol{H}_U^\top\boldsymbol{y}_\Omega\right), \tag{3.16}$$

with $\boldsymbol{K}_U = \boldsymbol{I}_R\otimes\boldsymbol{K}_s$ and $\boldsymbol{H}_U = \boldsymbol{O}\left(\tilde{\boldsymbol{X}}_U\left((\boldsymbol{W}\odot\boldsymbol{V})\otimes\boldsymbol{I}_M\right)\right)\in\mathbb{R}^{|\Omega|\times MR}$. The posterior for $\operatorname{vec}(\boldsymbol{V})$ can be obtained by applying the mode-2 tensor unfolding. It should be noted that the above derivation provides the posterior for the whole factor matrix, i.e., $\{\text{vec}(\boldsymbol{U}), \text{vec}(\boldsymbol{V}), \text{vec}(\boldsymbol{W})\}$, so the time complexity is $\mathcal{O}(R^3(M^3 + N^3 + P^3))$. We can further reduce the computational cost by sampling $\{\boldsymbol{u}_r, \boldsymbol{v}_r, \boldsymbol{w}_r\}$ one by one for $r = 1, \ldots, R$ as in Luttinen and Ilin (2009). In this case, the time complexity in learning these factor matrices can be further reduced to $\mathcal{O}(R(M^3 + N^3 + P^3))$ at the cost of slow/poor mixing.

Sampling kernel hyperparameters $\{\phi, \gamma\}$

As shown in Figure 2, sampling kernel hyperparameters conditional on the factor matrices should be straightforward through the Metropolis-Hastings algorithm. However, in practice, conditioning on the latent variables $\{\boldsymbol{U}, \boldsymbol{V}\}$ in such hierarchical GP models usually induces sharply peaked conditional posteriors over $\{\phi, \gamma\}$, making the Markov chains mix slowly and resulting in poor updates (Murray and Adams, 2010). To address this issue, we integrate out the latent factors from the model to get the marginal likelihood, and sample ϕ and γ from their marginal posterior distributions based on the slice sampling approach (Neal, 2003); i.e., we integrate out $\boldsymbol{U}$ when deriving the marginal posterior distribution $p\left(\phi \mid \boldsymbol{y}_\Omega, \boldsymbol{V}, \boldsymbol{W}, \tau, \boldsymbol{\mathcal{X}}\right)$, and likewise we build the marginal posterior $p\left(\gamma \mid \boldsymbol{y}_\Omega, \boldsymbol{U}, \boldsymbol{W}, \tau, \boldsymbol{\mathcal{X}}\right)$ by integrating out $\boldsymbol{V}$. Compared with recent related studies that sample hyperparameters conditioned on latent factors, e.g., Zhang and Banerjee (2022), the marginalization of latent factors can obtain tractable posteriors and avoid the possible issues in MCMC convergence. Note that in the work of Murray and Adams (2010), the proposed auxiliary variable strategy can also be used to handle outcomes with distributions beyond Gaussian. In addition, for kernel functions that contain more than one hyperparameter, one can apply a modified rectangle slice sampler (Neal, 2003).

Let us consider for example the hyperparameter of $\boldsymbol{K}_s$, i.e., ϕ. As $\text{vec}(\boldsymbol{U}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}_U)$ with $\boldsymbol{K}_U = \boldsymbol{I}_R \otimes \boldsymbol{K}_s$, we integrate out $\text{vec}(\boldsymbol{U})$ in Eq. (3.15) and obtain:

$$\log p\left(\boldsymbol{y}_\Omega \mid \phi, \boldsymbol{V}, \boldsymbol{W}, \tau, \boldsymbol{\mathcal{X}}\right) = -\frac{1}{2}\boldsymbol{y}_\Omega^\top \boldsymbol{K}_{\boldsymbol{y}|\phi}^{-1} \boldsymbol{y}_\Omega - \frac{1}{2}\log\left|\boldsymbol{K}_{\boldsymbol{y}|\phi}\right| - \frac{|\Omega|}{2}\log 2\pi, \tag{3.17}$$

where $\boldsymbol{K}_{\boldsymbol{y}|\phi} = \boldsymbol{H}_U \boldsymbol{K}_U \boldsymbol{H}_U^\top + \tau^{-1}\boldsymbol{I}_{|\Omega|} \in \mathbb{R}^{|\Omega|\times|\Omega|}$ and $\boldsymbol{H}_U$ is the same as the definition used in Eq. (3.16), i.e., $\boldsymbol{H}_U = \boldsymbol{O}\left(\tilde{\boldsymbol{X}}_U\left((\boldsymbol{W} \odot \boldsymbol{V}) \otimes \boldsymbol{I}_M\right)\right)$. The marginal posterior of ϕ becomes:

$$\begin{aligned}\log p\left(\phi \mid \boldsymbol{y}_\Omega, \boldsymbol{V}, \boldsymbol{W}, \tau, \boldsymbol{\mathcal{X}}\right) &\propto \log p(\phi) - \frac{1}{2}\boldsymbol{y}_\Omega^\top \boldsymbol{K}_{\boldsymbol{y}|\phi}^{-1}\boldsymbol{y}_\Omega - \frac{1}{2}\log\left|\boldsymbol{K}_{\boldsymbol{y}|\phi}\right| \\ &\propto \log p(\phi) + \frac{1}{2}\tau^2 \boldsymbol{y}_\Omega^\top \boldsymbol{H}_U\left(\boldsymbol{I}_R \otimes \boldsymbol{K}_s^{-1} + \tau \boldsymbol{H}_U^\top \boldsymbol{H}_U\right)^{-1}\boldsymbol{H}_U^\top \boldsymbol{y}_\Omega \\ &\quad - \frac{1}{2}\log\left|\boldsymbol{I}_R \otimes \boldsymbol{K}_s^{-1} + \tau \boldsymbol{H}_U^\top \boldsymbol{H}_U\right| - \frac{R}{2}\log\left|\boldsymbol{K}_s\right|,\end{aligned} \tag{3.18}$$

where we compute $\boldsymbol{y}_\Omega^\top \boldsymbol{K}_{\boldsymbol{y}|\phi}^{-1}\boldsymbol{y}_\Omega$ based on the Woodbury matrix identity, and use the matrix determinant lemma to compute $\log\left|\boldsymbol{K}_{\boldsymbol{y}|\phi}\right|$. The detailed derivation is given in Section 1 of the supplementary materials (Lei et al., 2024). Computing Eq. (3.18) involves the Cholesky factorization of $\boldsymbol{L} = \texttt{chol}(\boldsymbol{I}_R \otimes \boldsymbol{K}_s^{-1} + \tau \boldsymbol{H}_U^\top \boldsymbol{H}_U)$ of size $MR \times MR$; therefore, the overall time complexity is $\mathcal{O}(M^3R^3)$.

The slice sampling approach is robust to the selection of the sampling scale and easy to implement. Sampling γ can be achieved in a similar way. Note that, as mentioned, the sampling is performed on the log-transformed variables to avoid numerical issues. The detailed sampling process for kernel hyperparameters is provided in Section 2 (Algorithm 1) of the supplementary materials (Lei et al., 2024). We can also introduce different kernel functions (or the same kernel function with different hyperparameters) for each factor vector $\boldsymbol{u}_r$ and $\boldsymbol{v}_r$ as in Luttinen and Ilin (2009). In this case, the marginal posterior of the kernel hyperparameters can be derived in a similar way as in Eq. (3.18).

Sampling $\boldsymbol{\Lambda}_{\boldsymbol{w}}$

Given the conjugate Wishart prior, the posterior distribution of $\boldsymbol{\Lambda}_w$ is $\boldsymbol{\Lambda}_w | \boldsymbol{W}, \boldsymbol{\Psi}_0, \nu_0 \sim \mathcal{W}\left(\boldsymbol{\Psi}^*, \nu^*\right)$, where $[\boldsymbol{\Psi}^*]^{-1} = \boldsymbol{W}\boldsymbol{W}^\top + \boldsymbol{\Psi}_0^{-1}$ and $\nu^* = \nu_0 + R$.

Sampling the precision $\boldsymbol{\tau}$

Since we used a conjugate Gamma prior, the posterior distribution of τ is also a Gamma distribution with shape (a^*) and rate (b^*) being

$$a^* = a_0 + \frac{1}{2}|\Omega|, \; b^* = b_0 + \frac{1}{2}\left\|\boldsymbol{y}_\Omega - \boldsymbol{O}\left(\tilde{\boldsymbol{X}} \operatorname{vec}\left(\boldsymbol{W}(\boldsymbol{V} \odot \boldsymbol{U})^\top\right)\right)\right\|_2^2. \tag{3.19}$$

3.4 Model implementation

We summarize the implementation of BKTR in Algorithm 1. It should be noted that we update the correlated latent factors and the corresponding hyperparameters as one block in the Gibbs sampler to further ensure model convergence (Knorr-Held and Rue, 2002). Specifically, we take $\{\phi, \boldsymbol{U}\}$ as a block and update ϕ from its marginal posterior with a slice sampler followed by sampling $\boldsymbol{U}$ from $p\left(\boldsymbol{U} \mid \phi, -\right)$, and then similarly sample $\{\gamma, \boldsymbol{V}\}$ as another block. For MCMC inference, we run K_1 iterations as burn-in and take the following K_2 samples for estimation.

3.5 Model scalability

Compared to the original STVC approach (Gelfand et al., 2003), which requires $\mathcal{O}\left(|\Omega|^3\right)$ for hyperparameter sampling and $\mathcal{O}\left(|\Omega|^3 P^3\right)$ for coefficient variable learning. BKTR reduces the cost of updating hyperparameters and coefficients to $\mathcal{O}\left(R^3\left(M^3 + N^3 + P^3\right)\right)$; e.g., updating the factor matrix $\boldsymbol{U}$ following Eq. (3.16) requires $\mathcal{O}(M^3R^3)$ in time. For the slice sampling of kernel hyperparameters ϕ, each update inside the slice sampling loop (see Algorithm 1 in the supplementary materials (Lei et al., 2024)) requires $\mathcal{O}(M^3R^3)$ in computing the likelihood in Eq. (3.18). Such substantial gains in computing time allow us to analyze large-scale real-world spatiotemporal data and multi-dimensional relations, where generally the STVC is infeasible.

Given the above analysis, the computation cost BKTR depends on rank R and BKTR can work on large data sets such as $R \times M \approx 10^3 \sim 10^4$ (the same applies

Algorithm 1: MCMC sampling process of BKTR($\boldsymbol{y}_\Omega, \boldsymbol{\mathcal{X}}, R, K_1, K_2$).

```
Initialize $\{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}\}$ as normally distributed random values, $\phi = \gamma = 1$, and
  $\boldsymbol{\Lambda}_w \sim \mathcal{W}(\boldsymbol{I}_P, P)$. Set $\mu_\phi = \mu_\gamma = \log(1)$, $\tau_\phi = \tau_\gamma = 10$, and $a_0 = b_0 = 10^{-4}$.
for $k = 1 : K_1 + K_2$ do
    Sample kernel hyperparameter $\phi$ using Algorithm 1 in Section 2 of the
      supplementary materials;
    Sample factor $\text{vec}(\boldsymbol{U})$ from a Gaussian distribution:
      $\text{vec}(\boldsymbol{U})| - \sim \mathcal{N}\left(\boldsymbol{\mu}_U^*, (\boldsymbol{\Lambda}_U^*)^{-1}\right)$, $\boldsymbol{\mu}_U^* = \tau(\boldsymbol{\Lambda}_U^*)^{-1}\boldsymbol{H}_U^\top \boldsymbol{y}_\Omega$,
      $\boldsymbol{\Lambda}_U^* = \tau \boldsymbol{H}_U^\top \boldsymbol{H}_U + \boldsymbol{I}_R \otimes \boldsymbol{K}_s^{-1}$,
      $\boldsymbol{H}_U = \boldsymbol{O}\left(\left(\boldsymbol{X}_{(3)} \odot \boldsymbol{I}_{MN}\right)^\top ((\boldsymbol{W} \odot \boldsymbol{V}) \otimes \boldsymbol{I}_M)\right)$;
    Sample kernel hyperparameter $\gamma$ using Algorithm 1 in the supplementary
      materials;
    Sample factor $\text{vec}(\boldsymbol{V})$ from a Gaussian distribution:
      $\text{vec}(\boldsymbol{V})| - \sim \mathcal{N}\left(\boldsymbol{\mu}_V^*, (\boldsymbol{\Lambda}_V^*)^{-1}\right)$, $\boldsymbol{\mu}_V^* = \tau(\boldsymbol{\Lambda}_V^*)^{-1}\boldsymbol{H}_V^\top \boldsymbol{y}_\Omega^\top$,
      $\boldsymbol{\Lambda}_V^* = \tau \boldsymbol{H}_V^\top \boldsymbol{H}_V + \boldsymbol{I}_R \otimes \boldsymbol{K}_t^{-1}$,
      $\boldsymbol{H}_V = \boldsymbol{O}'\left(\left(\boldsymbol{X}_{(3)}^\top \odot \boldsymbol{I}_{MN}\right)^\top ((\boldsymbol{W} \odot \boldsymbol{U}) \otimes \boldsymbol{I}_N)\right)$, where $\boldsymbol{O}' \in \mathbb{R}^{|\Omega| \times MN}$ is a
      matrix removing the rows corresponding to the zeros in $\text{vec}(\boldsymbol{D}^\top)$ from
      $\boldsymbol{I}_{MN}$, $\boldsymbol{X}_{(3)}^\top$ is the mode-3 unfolding of $\boldsymbol{\mathcal{X}}^\top \in \mathbb{R}^{N \times M \times P}$, whose frontal slices
      are the transpose matrices of frontal slices of $\boldsymbol{\mathcal{X}}$, and $\boldsymbol{y}_\Omega^\top = \boldsymbol{O}' \, \text{vec}\left(\boldsymbol{Y}^\top\right)$;
    Sample hyperparameters $\boldsymbol{\Lambda}_w$ from a Wishart distribution:
      $\boldsymbol{\Lambda}_w| - \sim \mathcal{W}\left(\left(\boldsymbol{W}\boldsymbol{W}^\top + \boldsymbol{I}_P^{-1}\right)^{-1}, P + R\right)$;
    Sample factor $\text{vec}(\boldsymbol{W})$ from a Gaussian distribution:
      $\text{vec}(\boldsymbol{W})| - \sim \mathcal{N}\left(\boldsymbol{\mu}_W^*, (\boldsymbol{\Lambda}_W^*)^{-1}\right)$, $\boldsymbol{\mu}_W^* = \tau(\boldsymbol{\Lambda}_W^*)^{-1}\boldsymbol{H}_W^\top \boldsymbol{y}_\Omega$,
      $\boldsymbol{\Lambda}_W^* = \tau \boldsymbol{H}_W^\top \boldsymbol{H}_W + \boldsymbol{I}_R \otimes \boldsymbol{\Lambda}_w$,
      $\boldsymbol{H}_W = \boldsymbol{O}\left(\left(\boldsymbol{I}_{MN} \odot \boldsymbol{X}_{(3)}\right)^\top ((\boldsymbol{V} \odot \boldsymbol{U}) \otimes \boldsymbol{I}_P)\right)$;
    Sample precision $\tau$ from a Gamma distribution:
      $\tau| - \sim \text{Gamma}\left(a_0 + \frac{1}{2}|\Omega|, b_0 + \frac{1}{2}\left\|\boldsymbol{y}_\Omega - \boldsymbol{O}\left(\tilde{\boldsymbol{X}}\,\text{vec}\left(\boldsymbol{W}(\boldsymbol{V} \odot \boldsymbol{U})^\top\right)\right)\right\|_2^2\right)$;
    if $k > K_1$ then
        Collect the sample $\boldsymbol{U}^{(k-K_1)} = \boldsymbol{U}, \boldsymbol{V}^{(k-K_1)} = \boldsymbol{V}, \boldsymbol{W}^{(k-K_1)} = \boldsymbol{W}$;
        Compute $\boldsymbol{\mathcal{B}}^{(k-K_1)}$ using $\boldsymbol{\mathcal{B}} = \sum_{r=1}^R \boldsymbol{u}_r \circ \boldsymbol{v}_r \circ \boldsymbol{w}_r$.
return $\{\boldsymbol{\mathcal{B}}^{(k)}\}_{k=1}^{K_2}$ to approximate posterior coefficients and estimate
  unobserved data.
```

to N and P). However, it should be noted that the default BKTR still encounters computational issues when the number of spatial locations M (or time points N) becomes very large (e.g., say $M > 10^4$). We next discuss several solutions when M becomes large. Following Luttinen and Ilin (2009), the most straightforward solution is to update the three factor matrices column by column, which reduces the time cost to $\mathcal{O}\left(R\left(M^3+N^3+P^3\right)\right)$. Through this updating scheme, the Kroneceker products for learning latent factors and hyperparameters can be avoided. Taking $\boldsymbol{u}_r$ (i.e., the r-th column in $\boldsymbol{U}$) as an example, the posterior of $\boldsymbol{u}_r$ when updated independently still follows a Gaussian distribution $\mathcal{N}\left(\boldsymbol{\mu}_{u_r}^*, \left(\boldsymbol{\Lambda}_{u_r}^*\right)^{-1}\right)$, where

$$\boldsymbol{\Lambda}_{u_r}^* = \tau \boldsymbol{H}_{u_r}^\top \boldsymbol{H}_{u_r} + \boldsymbol{K}_s^{-1},\ \boldsymbol{\mu}_{u_r}^* = \tau\left(\boldsymbol{\Lambda}_{u_r}^*\right)^{-1}\left(\boldsymbol{H}_{u_r}^\top \boldsymbol{y}_r\right) \tag{3.20}$$

with $\boldsymbol{H}_{u_r} = \boldsymbol{O}\left(\tilde{\boldsymbol{X}}_U\left(\left(\boldsymbol{w}_r \otimes \boldsymbol{v}_r\right) \otimes \boldsymbol{I}_M\right)\right) \in \mathbb{R}^{|\Omega| \times M}$ and $\boldsymbol{y}_r = \boldsymbol{O}\Big(\boldsymbol{y} - \tilde{\boldsymbol{X}}_U\Big(\left(\boldsymbol{W}_{-r} \odot \boldsymbol{V}_{-r}\right) \otimes \boldsymbol{I}_M\Big) \operatorname{vec}\left(\boldsymbol{U}_{-r}\right)\Big) \in \mathbb{R}^{|\Omega|}$. $\{\boldsymbol{U}_{-r}, \boldsymbol{V}_{-r}, \boldsymbol{W}_{-r}\}$ are the factor matrices without the rth columns and $\boldsymbol{H}_{u_r}^\top \boldsymbol{H}_{u_r} \in \mathbb{R}^{M \times M}$ is a diagonal matrix. Luttinen and Ilin (2009) also suggested to use sparse approximation and predictive processes (Banerjee et al., 2008; Titsias, 2009) to model the each latent factor vector when M becomes large, i.e., using M_0 inducing points where $M_0 \ll M$. This reduces the time complexity to $\mathcal{O}(RM_0^2M)$.

Given that $\boldsymbol{H}_{u_r}^\top \boldsymbol{H}_{u_r}$ becomes an $M \times M$ diagonal matrix, if we replace the Gaussian prior with a Gaussian Markov random field (GRMF) (Rue and Held, 2005) for which the precision matrix $\boldsymbol{K}_s^{-1}$ becomes a sparse banded matrix, we can then use sparse matrix algorithms (as $\boldsymbol{\Lambda}_{u_r}^*$ also becomes a sparse banded matrix) to model large-scale data sets with $M \approx 10^4 \sim 10^6$. Similarly, NNGP can also be used here to model the spatial latent factor $\boldsymbol{U}$ for scalable inference (Finley et al., 2019). The column-by-column approach also offers the flexibility to learn different kernel hyperparameter ϕ_r (and thus covariance $\boldsymbol{K}_s^r$) for each latent factor vector $\boldsymbol{u}_r$. The marginal likelihood conditioning on ϕ_r in Eq. (3.17) becomes $\log p\left(\boldsymbol{y}_\Omega \mid \phi_r, -\right) \propto -\frac{1}{2}\boldsymbol{y}_r^\top \boldsymbol{K}_{\boldsymbol{y}_r|\phi}^{-1} \boldsymbol{y}_r - \frac{1}{2}\log\left|\boldsymbol{K}_{\boldsymbol{y}_r|\phi}\right|$, where $\boldsymbol{K}_{\boldsymbol{y}_r|\phi} = \boldsymbol{H}_{u_r}\boldsymbol{K}_s^r\boldsymbol{H}_{u_r}^\top + \tau^{-1}\boldsymbol{I}_{|\Omega|} \in \mathbb{R}^{|\Omega|\times|\Omega|}$. Correspondingly, the marginal posterior in Eq. (3.18) becomes:

$$\begin{aligned} &\log p\left(\phi_r \mid \boldsymbol{y}_\Omega, \boldsymbol{U}_{-r}, \boldsymbol{V}, \boldsymbol{W}, \tau, \boldsymbol{\mathcal{X}}\right) \propto \log p\left(\phi_r\right) - \frac{1}{2}\log\left|\boldsymbol{K}_s^r\right| + \\ &\frac{1}{2}\tau^2\boldsymbol{y}_r^\top\boldsymbol{H}_{u_r}\left(\left(\boldsymbol{K}_s^r\right)^{-1} + \tau\boldsymbol{H}_{u_r}^\top\boldsymbol{H}_{u_r}\right)^{-1}\boldsymbol{H}_{u_r}^\top\boldsymbol{y}_r - \frac{1}{2}\log\left|\left(\boldsymbol{K}_s^r\right)^{-1} + \tau\boldsymbol{H}_{u_r}^\top\boldsymbol{H}_{u_r}\right|. \end{aligned} \tag{3.21}$$

Lastly, considering that learning hyperparameters via slice sampling is expensive, another possible solution to further reduce the computational cost is to use cross-validation to specify kernel hyperparameters as in Finley et al. (2019). The choice of kernel hyperparameters is often not that sensitive in regression and thus cross-validation can be performed at a moderately crude resolution. However, it could still be time-consuming to directly implement the cross-validatory approach based on MCMC sampling, since such an iterative process needs to be run several folds/times for each combination of hyperparameters.

4 Simulation study

In this section, we evaluate the performance of BKTR on three simulated data sets. All simulations are performed on a laptop with a 6-core Intel Xenon 2.60 GHz CPU and

32GB RAM. Specifically, we conduct three studies: (1) a low-rank structured association analysis to test the estimation accuracy and statistical properties of BKTR with different rank settings and in scenarios with different observation/missing rates, (2) a small-scale analysis to compare BKTR with STVC and a pure low-rank tensor regression model, and (3) a moderately sized analysis to test the performance of BKTR on more practical STVC modeling.

4.1 Simulation 1: performance and statistical properties of BKTR

Simulation setting

To fully evaluate the properties of BKTR, we first simulate a data set where $\boldsymbol{\mathcal{B}}$ is constructed following an exact CP decomposition. We generate $M = 300$ spatial locations in a $[0, 10] \times [0, 10]$ square and $N = 100$ evenly located time points in $[0, 10]$. The variable $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{300\times100\times5}$ $(P = 5)$ contains an intercept and four spatiotemporal covariates as follows:

$$\begin{aligned}
&\boldsymbol{\mathcal{X}}(:,:,p=1) = \text{ones}(M,N), && \text{(for the intercept)}\\
&\boldsymbol{\mathcal{X}}(:,:,p=2,3) = \mathbf{1}_N^\top \otimes \boldsymbol{x}_s^p, \text{with } \boldsymbol{x}_s^p \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_M), && \text{(covariates only vary with space)}\\
&\boldsymbol{\mathcal{X}}(:,:,p=4,5) = \mathbf{1}_M \otimes (\boldsymbol{x}_t^p)^\top, \text{with } \boldsymbol{x}_t^p \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_N). && \text{(covariates only vary with time)}
\end{aligned} \tag{4.1}$$

We set the true CP rank $R = 10$ and generate the coefficient variable $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{300\times100\times5}$ as $\boldsymbol{\mathcal{B}} = \sum_{r=1}^{R} \boldsymbol{u}_r \circ \boldsymbol{v}_r \circ \boldsymbol{w}_r$, where $\boldsymbol{u}_r \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K}_s)$, $\boldsymbol{v}_r \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K}_t)$, and $\boldsymbol{w}_r \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}_w^{-1})$, for $r = 1, \ldots, R$. Covariance matrices $\boldsymbol{K}_s \in \mathbb{R}^{300\times300}$ and $\boldsymbol{K}_t \in \mathbb{R}^{100\times100}$ are computed from a Matérn 3/2 kernel $k_s(\boldsymbol{s}_m, \boldsymbol{s}_{m'}) = \sigma_s^2 \left(1 + \frac{\sqrt{3}d}{\phi}\right) \exp\left(-\frac{\sqrt{3}d}{\phi}\right)$ (d is the distance between locations $\boldsymbol{s}_m$ and $\boldsymbol{s}_{m'}$) and a squared exponential (SE) kernel $k_t(t_n, t_{n'}) = \sigma_t^2 \exp\left(-\frac{(t_n - t_{n'})^2}{2\gamma^2}\right)$, respectively, with variances $\sigma_s^2 = \sigma_t^2 = 2$, length-scales $\phi = \gamma = 1$, and $\boldsymbol{\Lambda}_w \sim \mathcal{W}(\boldsymbol{I}_P, P)$. Based on $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{B}}$, the spatiotemporal data $\boldsymbol{Y} \in \mathbb{R}^{300\times100}$ is then generated following Eq. (3.3) with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \tau^{-1}\boldsymbol{I}_{MN})$ where $\tau^{-1} = 1$.

We test the performance of BKTR under different settings to evaluate its sensitivity to i) the rank specification defined by the user, and ii) the proportion of observed responses, i.e., $\frac{|\Omega|}{MN}$. For i), we specify $R = \{4, 7, 10, 13, 16, 20, 25, 30, 35, 40\}$ and randomly sample 50% of the data $\boldsymbol{Y}$ as observed values. For ii), we applied the model under 5 different observed response rates, where we randomly sample (90%, 70%, 50%, 30%, 10%) of $\boldsymbol{Y}$ as observed values and evaluate the performance of BKTR with the true rank setting ($R = 10$). In all settings, we assume that the kernel functions are known, i.e., k_s is Matérn 3/2 and k_t is SE. We also include in the analysis a sixth covariate in $\boldsymbol{\mathcal{X}}$ drawn from the standard normal distribution and unrelated to the outcome (the corresponding estimated coefficients should be close to zero). For each setting, we replicate the simulation 40 times and run the MCMC with $K_1 = 1000$ and $K_2 = 500$.

In each setting, we measure the estimation accuracy on $\boldsymbol{\mathcal{B}}$ and the prediction accuracy on $\boldsymbol{y}_{\Omega^c}$ (unobserved data) using mean absolute error (MAE) and root mean square

$\text{MAE}_{\boldsymbol{\mathcal{B}}}/\text{RMSE}_{\boldsymbol{\mathcal{B}}}$	$\text{MAE}_{\boldsymbol{y}_{\Omega^c}}/\text{RMSE}_{\boldsymbol{y}_{\Omega^c}}$	CVG	INT	CRPS
0.21±0.02/0.33±0.04	0.91±0.01/1.15±0.02	94.83%±0.75%	1.04±0.10	0.15±0.01

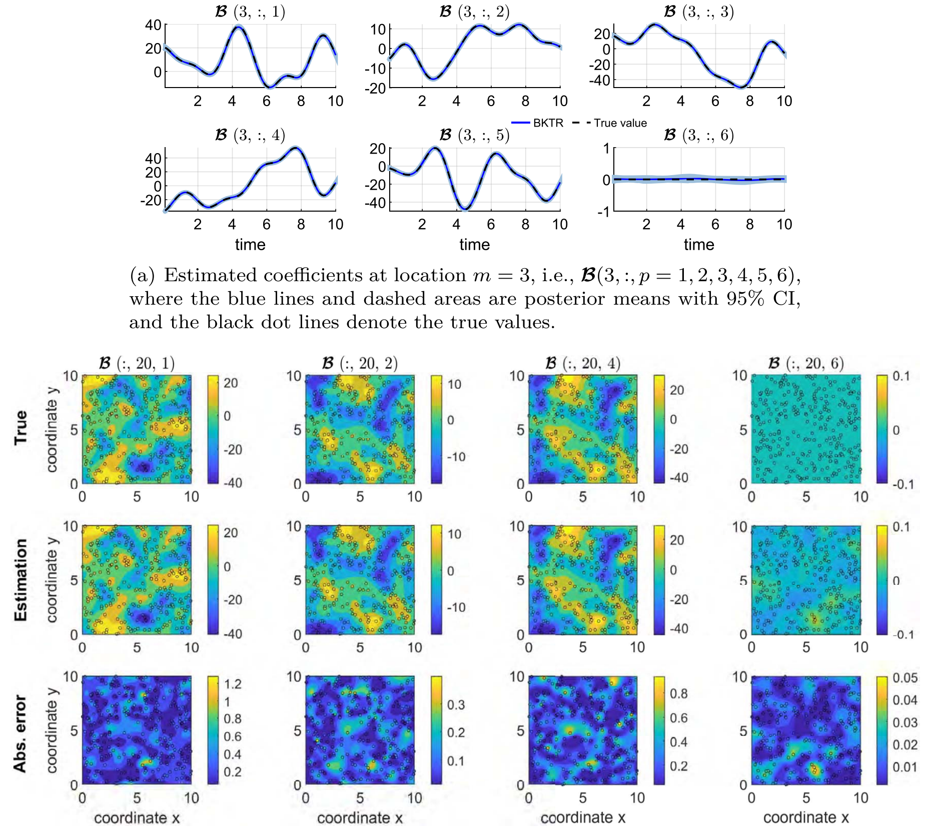
Table 1: Performance of BKTR ($R = 10$) on Simulation 1, with 50% $\boldsymbol{Y}$ missing.

error (RMSE) between the true values and the corresponding posterior mean estimations, where the true $\boldsymbol{\mathcal{B}}$ values for the random non-significant covariate are set to zero. For evaluating the performance on uncertainty estimation, we assess interval coverage (CVG) (Heaton et al., 2019), interval score (INT) of the 95% credible intervals (CI), and continuous ranked probability score (CRPS) (Gneiting and Raftery, 2007) of all $\boldsymbol{\mathcal{B}}$ values. Note that all CIs are obtained based on the K_2 samples after burn-in. Detailed definitions of all these performance metrics are given in Section 3 of the supplementary materials (Lei et al., 2024).

Results

Table 1 shows the performance metrics (mean±std calculated from the 40 replicated simulations) of BKTR in the case where the true rank is specified ($R = 10$) and when 50% of $\boldsymbol{Y}$ is partially observed. We can see that the CVG of 95% CI of $\boldsymbol{\mathcal{B}}$ is around 95% as expected. We also plot some estimation results of one chain/simulation in Figure 3, where panel (a) illustrates the temporal evolution of the estimated coefficients for each covariate at a given location, and panel (b) maps spatial surfaces of $\boldsymbol{\mathcal{B}}$ for each covariate at a given time point. These plots show that BKTR can effectively approximate the spatiotemporally varying relationships between the spatiotemporal data and the corresponding covariates.

The performance of BKTR with different rank specifications and 50% of the data points observed are compared in Figure 4(a). We see that when the rank is overspecified (larger than the true value, i.e., $R > 10$), BKTR can still achieve reliable estimation accuracy for both coefficients and unobserved data, even when R is much larger than 10, e.g., $R = 40$. The CVG for the 95% CI of $\boldsymbol{\mathcal{B}}$ is also maintained around 95% when $R > 10$, the corresponding INT and CRPS maintain a low value. This indicates that BKTR is able to provide accurate estimation for the coefficients along with high-quality uncertainty even when the model includes redundant parameters. In addition, Figure 4(b) illustrates the effect of the observation rate of $\boldsymbol{Y}$, where the results of BKTR (specifying $R = 10$) with different proportions of observed data are compared. We observe that BKTR has a stable low estimation error when the rate of observed data decreases, and it continues to obtain valid CIs despite the increased lengths of CI for lower observation rates (i.e., the INT becomes larger). These results demonstrate the powerful applicability of the proposed Bayesian framework. We further show the temporal estimations of coefficients at a given location obtained in different cases in Figure 5. From Figure 5(a), we see that a higher rank specification generates broader intervals, which is consistent with the INT results in Figure 4(a), i.e., the INT slightly increases with overspecified ranks. Panel (b) shows that BKTR can accurately estimate time-varying coefficients even when only 10% of the response data is observed, demonstrating its high modeling capacity.



(a) Estimated coefficients at location $m = 3$, i.e., $\boldsymbol{\mathcal{B}}(3, :, p = 1, 2, 3, 4, 5, 6)$, where the blue lines and dashed areas are posterior means with 95% CI, and the black dot lines denote the true values.

(b) Interpolated spatial surfaces of true value, estimated value (posterior mean), and absolute estimation error of the coefficients at time point $n = 20$, i.e., $\boldsymbol{\mathcal{B}}(:, 20, p = 1, 2, 4, 6)$. Black circles denote the positions of sampled locations.

Figure 3: BKTR ($R = 10$) estimated coefficients for Simulation 1 (50% of $\boldsymbol{Y}$ is observed).

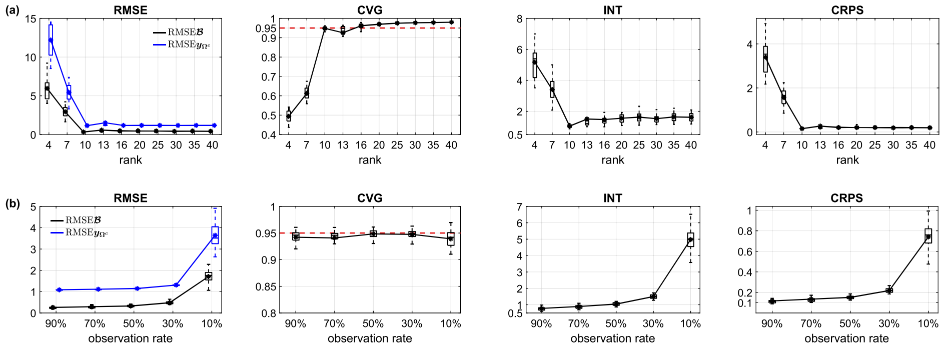
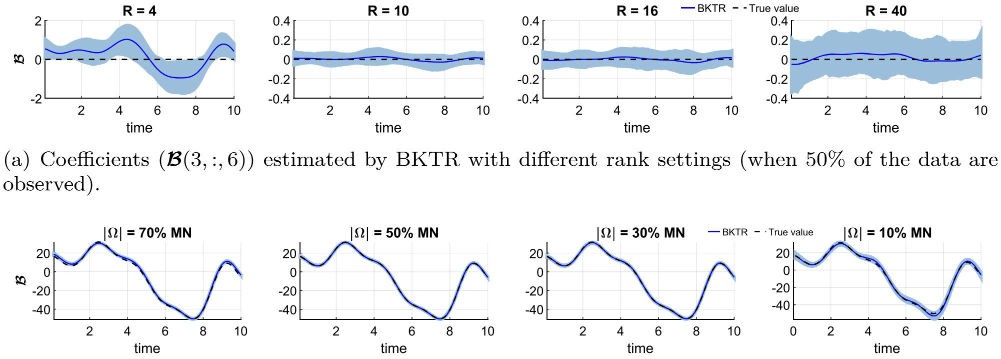


Figure 4: Sensitivity test of BKTR for Simulation 1: (a) Effects of the rank R; (b) Effects of the observation rate $\frac{|\Omega|}{MN}$. For each case, the figure shows the boxplots and mean values of the corresponding metrics calculated from 40 replications.



(a) Coefficients ($\boldsymbol{\mathcal{B}}(3,:,6)$) estimated by BKTR with different rank settings (when 50% of the data are observed).

(b) Coefficients ($\boldsymbol{\mathcal{B}}(3,:,3)$) estimated by BKTR (with $R = 10$) under different observation rates.

Figure 5: Comparison of BKTR in different settings for Simulation 1. (a) and (b) plot the estimated $\boldsymbol{\mathcal{B}}$ (mean with 95% CI) of one simulation at location #3 ($m = 3$) for the 6th and 3rd covariates (i.e., $\boldsymbol{\mathcal{B}}(3,:,6)$ and $\boldsymbol{\mathcal{B}}(3,:,3)$), respectively.

4.2 Simulation 2: model comparison on a small data set

Simulation setting

In this simulation, we generate a small data set in which the true $\boldsymbol{\mathcal{B}}$ is directly generated following a separable covariance matrix. Specifically, we simulate 30 locations ($M = 30$) in a $[0, 10] \times [0, 10]$ square and $N = 30$ evenly distributed time points in $[0, 10]$. We then generate a small synthetic data set $\left\{\boldsymbol{Y} \in \mathbb{R}^{30\times 30}, \boldsymbol{\mathcal{X}} \in \mathbb{R}^{30\times 30\times 3}\right\}$ ($P = 3$) following Eq. (3.3) with:

$$\begin{aligned}
&\boldsymbol{\mathcal{X}}(:,:,1) = \text{ones}(M, N),\ \ \boldsymbol{\mathcal{X}}(:,:,2) = \mathbf{1}_N^\top \otimes \boldsymbol{x}_s,\ \ \boldsymbol{\mathcal{X}}(:,:,3) = \mathbf{1}_M \otimes \boldsymbol{x}_t^\top,\\
&\text{vec}\left(\boldsymbol{B}_{(3)}\right) \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{K}_t \otimes \boldsymbol{K}_s \otimes \boldsymbol{\Lambda}_w^{-1}\right),\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \tau^{-1}\boldsymbol{I}_{MN}),
\end{aligned}$$

where $\boldsymbol{x}_s \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_M)$, $\boldsymbol{x}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_N)$, $\boldsymbol{K}_s$ and $\boldsymbol{K}_t$ are still computed from a Matérn 3/2 kernel and a SE kernel, respectively, and $\boldsymbol{\Lambda}_w \sim \mathcal{W}(\boldsymbol{I}_P, P)$. For model parameters, we set the kernel variance hyperparameters at $\sigma_s^2 = \sigma_t^2 = 2$, and consider combinations of data noise variance $\tau^{-1} \in \{0.2, 1, 5\}$, and kernel length-scale hyperparameters $\phi = \gamma \in \{1, 2, 4\}$. We specify the CP rank $R = 10$ for BKTR estimation and compare BKTR with the original STVC model (Gelfand et al., 2003) and a pure low-rank Bayesian probabilistic tensor regression (BTR) model without imposing any spatiotemporal GP priors by setting $\boldsymbol{K}_s = \boldsymbol{I}_M, \boldsymbol{K}_t = \boldsymbol{I}_N$ as the prior and using the same rank as BKTR. For both BKTR and STVC, we assume that the kernel functions are known, i.e., k_s is Matérn 3/2 and k_t is SE, and use MCMC to estimate the unknown kernel hyperparameters. As in the previous simulations, we add a normally distributed non-significant random covariate in $\boldsymbol{\mathcal{X}}$ when fitting the model. We replicate each simulation 25 times and run the MCMC sampling with $K_1 = 1000$ and $K_2 = 500$.

Settings		BTR	STVC	BKTR
$\tau^{-1}=0.2$	$\phi=\gamma=1$	1.00±0.29 / 1.57±0.49	**0.64**±0.16 / **1.00**±0.29	0.81±0.19 / 1.32±0.32
	$\phi=\gamma=2$	0.94±0.35 / 1.54±0.62	0.63±0.15 / 1.09±0.23	**0.61**±0.18 / **1.06**±0.34
	$\phi=\gamma=4$	0.74±0.29 / 1.24±0.54	0.53±0.11 / 0.88±0.19	**0.40**±0.13 / **0.70**±0.28
$\tau^{-1}=1$	$\phi=\gamma=1$	1.00±0.25 / 1.55±0.44	0.82±0.10 / 1.25±0.24	**0.79**±0.18 / **1.23**±0.33
	$\phi=\gamma=2$	0.85±0.23 / 1.33±0.41	0.63±0.07 / 0.92±0.15	**0.60**±0.12 / **0.90**±0.22
	$\phi=\gamma=4$	0.73±0.22 / 1.18±0.42	0.54±0.09 / 0.74±0.16	**0.45**±0.09 / **0.71**±0.17
$\tau^{-1}=5$	$\phi=\gamma=1$	1.10±0.16 / 1.66±0.27	1.09±0.12 / 1.60±0.19	**0.89**±0.14 / **1.32**±0.23
	$\phi=\gamma=2$	1.01±0.16 / 1.52±0.30	0.92±0.12 / 1.31±0.18	**0.75**±0.11 / **1.11**±0.20
	$\phi=\gamma=4$	0.90±0.15 / 1.37±0.29	0.78±0.08 / 1.08±0.14	**0.61**±0.07 / **0.89**±0.14
Average computing time		≈0.008 sec/iter.	≈15.44 sec/iter.	≈0.011 sec/iter.

Best results are highlighted in bold fonts.

Table 2: Performance comparison for Simulation 2 ($\mathrm{MAE}_{\boldsymbol{\mathcal{B}}}/\mathrm{RMSE}_{\boldsymbol{\mathcal{B}}}$).

Results

Table 2 presents the accuracy (mean±std) of the estimated coefficient tensor $\boldsymbol{\mathcal{B}}$ obtained from the 25 simulations. We also report the average computing time for each MCMC iteration. As we can see, BKTR with rank $R = 10$ achieves similar accuracy to that of STVC, and even exhibits superior performance in the presence of increased data noise or enlarged kernel length-scales. This means that BKTR is more robust for noisy data and the low-rank assumption can fit the spatiotemporal relations better when they vary more smoothly/slowly. Additionally, BKTR also clearly outperforms BTR, which confirms the importance of introducing GPs to encode the spatial and temporal dependencies. These results suggest that the proposed kernelized CP decomposition can approximate the true coefficient tensor relatively well even if the true $\boldsymbol{\mathcal{B}}$ does not follow an exact low-rank specification. In terms of computing time, we see that BKTR is much faster than STVC. Due to the high cost of STVC, it becomes infeasible to analyze a data set of moderate size, e.g., $M, N = 100$, and $P = 10$.

Figure 6 shows the estimation results of the three approaches for one chain/simulation when $\tau^{-1} = 1, \phi = \gamma = \{1, 2, 4\}$. We plot an example for the third covariate at location #8 ($m = 8, p = 3$) to show the temporal variation of the coefficients estimated by the three methods. As can be seen, BKTR and STVC generate similar results and most of the coefficients are contained in the 95% CI for these two models. Although BTR can still capture the general trend, it fails to reproduce the local temporal dependency due to the absence of spatial/temporal priors. To further verify the mixing of the GP hyperparameters, we run the MCMC for BKTR for 5000 iterations and show the trace plots and probability distributions (after burn-in) for ϕ and γ obtained when $\tau^{-1} = 1, \phi = \gamma \in \{1, 2, 4\}$ in Figure 7(a) and (b), respectively. Figure 7(c) shows the effective sample size (ESS) of kernel hyperparameters obtained by BKTR in different settings, where in each case the mean ESS of 25 simulations are given. We can see that the Markov chains for the kernel hyperparameters mix fast and well, and a few hundred iterations should be sufficient for posterior inference.

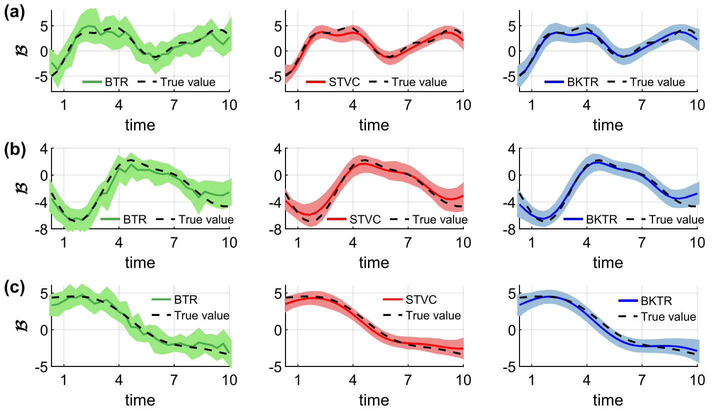


Figure 6: Comparison of the estimated coefficients when (a) $\tau^{-1}=1, \phi=\gamma=1$, (b) $\tau^{-1}=1, \phi=\gamma=2$, (c) $\tau^{-1}=1, \phi=\gamma=4$. We show the approximated coefficients for the third covariate at location #8, with solid curves representing the posterior mean and the shaded areas denoting 95% CI.

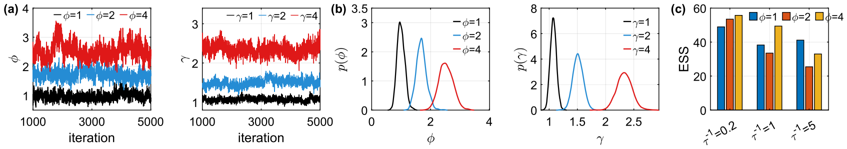


Figure 7: BKTR estimated kernel hyperparameters for Simulation 2 when $\tau^{-1}=1$: (a) trace plots; (b) posterior densities. (c): ESS of kernel hyperparameters learned by BKTR for Simulation 2.

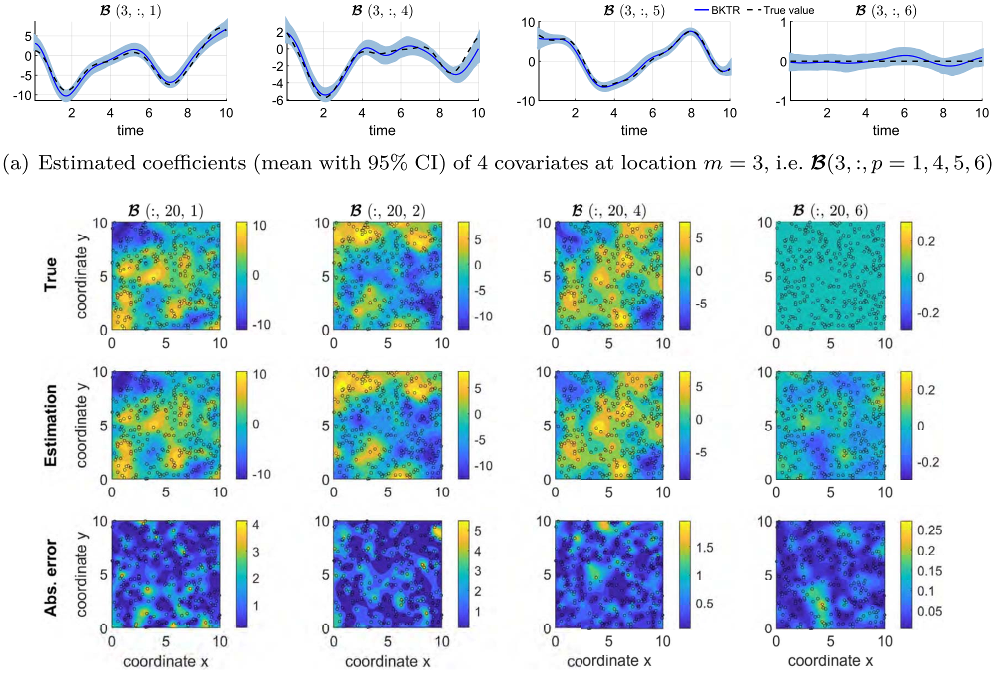
4.3 Simulation 3: STVC modeling on a moderate-sized data set

Simulation setting

To evaluate the performance of BKTR in more realistic settings where the data size is large and the relationships are usually captured without low-rank structure, we further generate a relatively large synthetic data set of the same size as in Simulation 1, i.e., $M=300$ locations and $N=100$ time points. We generate the covariates $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{300\times100\times5}$ $(P=5)$ according to Eq. (4.1). The coefficients are simulated using $\operatorname{vec}\left(\boldsymbol{B}_{(3)}\right) \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{K}_t \otimes \boldsymbol{K}_s \otimes \boldsymbol{\Lambda}_w^{-1}\right)$. The parameters $\{\boldsymbol{K}_s, \boldsymbol{K}_t, \boldsymbol{\Lambda}_w\}$ and $\boldsymbol{\epsilon}$ are specified as in Simulation 1, i.e., $\boldsymbol{K}_s$ and $\boldsymbol{K}_t$ are computed from a Matérn 3/2 and a SE kernel, respectively, with hyperparamaters $\{\sigma_s^2=\sigma_t^2=2, \phi=\gamma=1\}$, $\boldsymbol{\Lambda}_w \sim \mathcal{W}(\boldsymbol{I}_P, P)$, and $\tau^{-1}=1$. We randomly select 50% of the generated data $\boldsymbol{Y}$ as observed responses. To assess the effect of the rank specification, we try different CP rank settings R in $\{5, 10, \ldots, 60\}$ and compute MAE and RMSE for both $\boldsymbol{\mathcal{B}}$ and $\boldsymbol{y}_{\Omega^c}$. We replicate the experiment with each rank setting 15 times and run the MCMC sampling with $K_1=1000$ and $K_2=500$.

Results

Figure 8 shows the temporal behaviors and spatial patterns of the estimated coefficients of four covariates in one simulation when $R = 40$. As one can see, BKTR can effectively reproduce the true values with acceptable errors. Figure 9 shows the effect of rank. We see that choosing a larger rank R gives better accuracy in terms of both $\text{MAE}_{\boldsymbol{\mathcal{B}}}/\text{RMSE}_{\boldsymbol{\mathcal{B}}}$ and $\text{MAE}_{\boldsymbol{y}_{\Omega^c}}/\text{RMSE}_{\boldsymbol{y}_{\Omega^c}}$. However, the accuracy gain becomes marginal when $R > 30$.



(a) Estimated coefficients (mean with 95% CI) of 4 covariates at location $m = 3$, i.e. $\boldsymbol{\mathcal{B}}(3, :, p = 1, 4, 5, 6)$.

(b) Interpolated spatial surfaces of the estimated coefficients at time point $n = 20$, i.e. $\boldsymbol{\mathcal{B}}(:, 20, p = 1, 2, 4, 6)$, where black circles denote the positions of sampled locations.

Figure 8: BKTR ($R = 40$) estimated coefficients for Simulation 3.

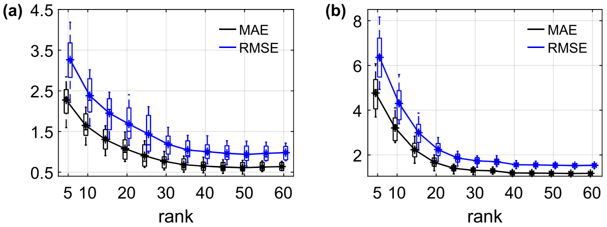


Figure 9: The effect of rank for Simulation 3. The figure shows the boxplot and mean of the estimation error on (a) $\boldsymbol{\mathcal{B}}$ and (b) $\boldsymbol{y}_{\Omega^c}$ with respect to the tensor rank.

spatial	(a)	length of cycle paths; length of major roads; length of minor roads; station capacity;
	(b)	numbers of metro stations, bus stations, and bus routes; numbers of restaurants, universities, other business & enterprises; park area; walkscore; population;
temporal		daily relative humidity; daily maximum temperature; daily mean temperature; total precipitation; dummy variables for holidays.

Table 3: Description summary of independent variables.

This demonstrates that the proposed Bayesian treatment offers a flexible solution in terms of model estimation.

5 Bike-sharing demand modeling

5.1 Data description

In this section, we perform local spatiotemporal regression on a large-scale bike-sharing trip data set collected from BIXI[1]—a docked bike-sharing service in Montreal, Canada. BIXI operates 615 stations across the city and the yearly service time window is from April to November. We collect the number of daily departures for each station over $N = 196$ days (from April 15 to October 27, 2019). We discard the stations with an average daily demand of less than 5, and only consider the remaining $M = 587$ stations. The matrix $\boldsymbol{Y}$ contains 13.0% unobserved/corrupted values in the raw data, and we only keep the rest as the observed set Ω. Given that the collected daily departures can have different variances across the spatial locations, we take the logarithm of the data matrix $\boldsymbol{Y}$ to make the variances more consistent and obtain a nearly Gaussian distributed data. We approximate this transformed dataset using the assumption of Gaussian likelihood with a linear covariance and a homoscedastic residual process.

We follow the analysis in Faghih-Imani et al. (2014) and Wang et al. (2021), and consider two types of spatial covariates: (a) features related to cycling infrastructure, and (b) land-use and built environment features. Particularly, similar to the operations in Wang et al. (2021), the spatial covariates are collected using the intersections of 250-meter circle buffers and Thiessen polygons for obtaining predictors from non-overlapping areas. For temporal covariates, we mainly consider weather conditions and holidays. Table 3 lists the 13 spatial covariates ($\boldsymbol{x}_s^p \in \mathbb{R}^M$) and 5 temporal covariates ($\boldsymbol{x}_t^p \in \mathbb{R}^N$) used in this analysis. The final number of covariates is $P = 13 + 5 + 1 = 19$, including the intercept. In constructing the covariate tensor $\boldsymbol{\mathcal{X}}$, we follow the same approach as in the simulation experiments. Specifically, we set $\boldsymbol{\mathcal{X}}(:,:,1) = \text{ones}(M, N)$, and fill the rest of the tensor slices with $\mathbf{1}_N^T \otimes \boldsymbol{x}_s^p$ $(p = 2 : 14)$ for the spatial covariates and $\mathbf{1}_M \otimes (\boldsymbol{x}_t^p)^\top$ $(p = 15 : 19)$ for the temporal covariates. Given the difference in magnitudes, we normalize all covariates to $[0, 1]$ using a min-max normalization. Since we use a zero-

[1] https://bixi.com.

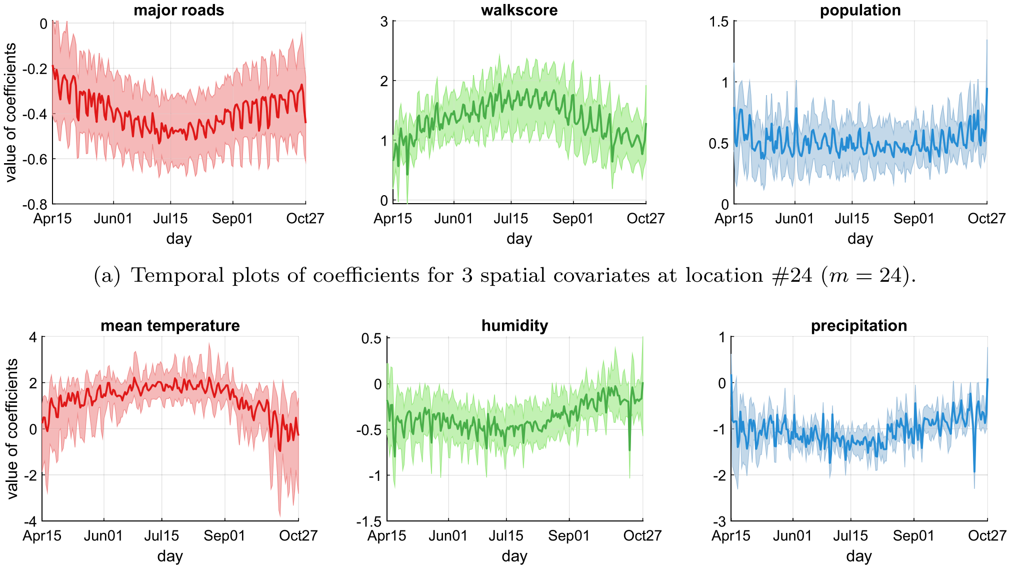
mean GP to parameterize $\boldsymbol{\beta}$, we normalize departure trips by removing the global effect of all covariates through linear regression and consider $\boldsymbol{y}$ to be the unexplained part. The coefficient tensor $\boldsymbol{\mathcal{B}}$ contains more than 2×10^6 entries, preventing us from using STVC directly.

5.2 Experimental setup

For spatial factors, we use a Matérn 3/2 kernel as a universal prior, i.e., $k_s(\boldsymbol{s}_m, \boldsymbol{s}_{m'}) = \left(1 + \frac{\sqrt{3}d}{\phi}\right) \exp\left(-\frac{\sqrt{3}d}{\phi}\right)$, where d is the Euclidean distance between locations $\boldsymbol{s}_m$ and $\boldsymbol{s}_{m'}$, and ϕ is the spatial length-scale. The Matérn class kernel is commonly used as a prior kernel function in spatial modeling. For temporal factors, we use a locally periodic correlation matrix by taking the product of a periodic kernel and a SE kernel, i.e., $k_t(t_n, t_{n'}) = \exp\left(-\frac{2\sin^2(\pi(t_n - t_{n'})/T)}{\gamma_1^2} - \frac{(t_n - t_{n'})^2}{2\gamma_2^2}\right)$, where γ_1 and γ_2 denote the length-scale and decay-time for the periodic component, respectively, and we fix the period as $T = 7$ days. This specification suggests a weekly temporal pattern that can change over time and allows us to characterize both the daily variation and the weekly trend of the coefficients. We set the CP rank $R = 20$, and run MCMC with $K_1 = 1000$ and $K_2 = 500$ iterations.

5.3 Results

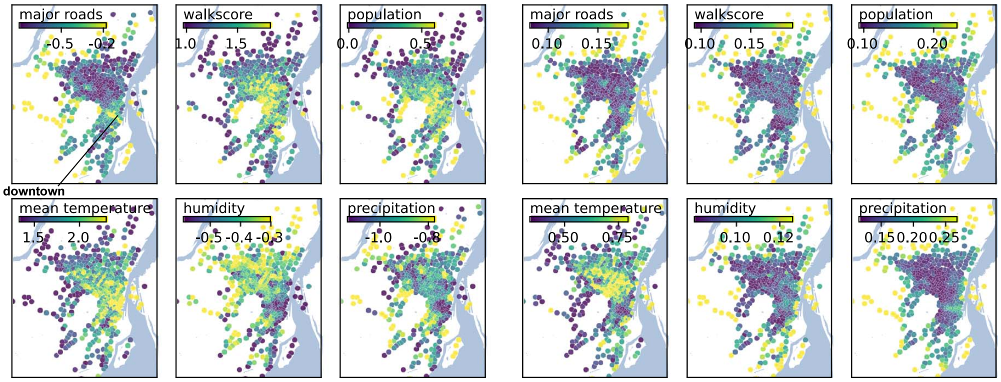
Figure 10 and 11 demonstrate several examples of estimated coefficients, showing temporal plots of $\boldsymbol{\mathcal{B}}(\boldsymbol{s}_m, :, :)$ and spatial maps of $\boldsymbol{\mathcal{B}}(:, t_n, :)$, respectively. As we can see in Figure 10, the temporal variations of the coefficients for both spatial and temporal covariates are interpretable. The coefficients (along with CI describing the uncertainty) allow us to identify the most important factors for each station at each time point. For example, we observe a similar variation over a week and a general long-term trend from April to October. Furthermore, the magnitude of the coefficients can be larger during summer (July / August) compared to the beginning and end of the operation period (e.g., for major roads, walkscore, and mean temperature), which is as expected as the outdoor temperature drops. Overall, for spatial variables, the positive correlation of walkability and the negative impact caused by the length of major roads indicate that bicycle demands tend to be higher in densely populated neighborhoods. For the temporal covariates, the precipitation and humidity variables both relate to negative coefficients, implying that people are less likely to ride bicycles in rainy/humid periods. The spatial distributions of the coefficients in Figure 11 also demonstrate consistent estimations, where the effects of covariates tend to be more obvious in the downtown area, involving coefficients with larger absolute values. Again, one can further explore the credible intervals to evaluate the significance of each covariate at a given location. In addition, the estimated variance τ^{-1} for $\boldsymbol{\epsilon}$ gives a posterior mean of 6.19×10^{-2} with a 95% CI of $[6.15, 6.28] \times 10^{-2}$. This variance is much smaller compared with the variability of the data, validating a good performance of data fitting using the proposed BKTR model. In summary, BKTR produces interpretable results for understanding the effects of different spatial and temporal covariates on bike-sharing demand. These findings



(a) Temporal plots of coefficients for 3 spatial covariates at location #24 ($m = 24$).

(b) Temporal plots of coefficients for 3 temporal covariates at location #7 ($m = 7$).

Figure 10: Temporal illustration of BKTR estimated coefficients for log-transformed BIXI demand data. The plots show the posterior mean with 95% CI.



(a) Spatial maps of coefficients mean.

(b) Spatial maps of coefficients std.

Figure 11: Spatial maps of BKTR estimated coefficients for log-transformed BIXI demand: (a) posterior mean for 6 covariates on August 1, 2019 ($n = 109$); (b): the corresponding posterior standard deviation (std.).

can help planners evaluate the performance of bike-sharing systems and update/design them.

Scenarios	linear regression	SVC	BKTR ($R = 20$)
30% location missing	0.51/ 0.65/ 0.07	0.57/ 0.75/ 0.10	**0.47**/ **0.63**/ **0.26**
30% location & time points missing	0.52/ 0.66/ 0.04	0.44/ 0.62/ 0.14	**0.40**/ **0.57**/ **0.34**

Best results are highlighted in bold fonts.

Table 4: Prediction performance for BIXI bike-sharing demand $\left(\mathrm{MAE}_{\boldsymbol{y}_{\Omega^c}}/\mathrm{RMSE}_{\boldsymbol{y}_{\Omega^c}}/R^2\right)$.

5.4 Spatial interpolation

Due to the introduction of GP, BKTR is able to perform spatial prediction, i.e., kriging, for unknown locations, and such results can be used to select sites for new stations. To validate the spatial prediction/interpolation capacity, we randomly select 30% locations of the BIXI data set to be missing, and compute the prediction accuracy for these missing/testing data ($\boldsymbol{y}_{\Omega^c}$). Given that STVC is not applicable for a data set of such size, we compare BKTR (rank $R = 20$) with SVC (spatially varying coefficient processes model) (Gelfand et al., 2003) which omits the modeling in time dimension. A linear regression model with static $\boldsymbol{\beta}$ is also considered as the baseline in order to demonstrate the role of incorporating spatiotemporally varying coefficients in such prediction tasks. We also test these methods in the case where both 30% locations and 30% time periods are missing. The prediction errors are given in Table 4, where we show the MAE, RMSE and R^2 on $\boldsymbol{y}_{\Omega^c}$. R^2 is used to illustrate the fitting performance of the model, and the definition is also provided in the supplementary materials Section 3 (Lei et al., 2024). It is obvious that BKTR offers better performance with lower estimation errors in both missing scenarios, indicating its effectiveness in spatial prediction for real-world spatiotemporal data sets that often contain complicated missing/corruption patterns.

6 Discussion

6.1 Identifiability and convergence of the model

In the regression problem studied in this paper, the kernel/covariance hyperparameters, i.e., $\{\phi, \gamma\}$, interpret/imply the correlation structure of the data, and the coefficients, i.e., $\boldsymbol{\mathcal{B}}$, reveal the spatiotemporal processes of the underlying associations. Thus, we consider the convergence of kernel hyperparameters and the identifiability of the coefficients to be crucial. Note that the identifiability of latent factors decomposed from $\boldsymbol{\mathcal{B}}$ are not that important. For instance, our model is invariant in applying the same column permutation to the three factor matrices. From the results in simulation experiments (see Figure 7), it is clear that the Markov chains for kernel hyperparameters converge fast and well when using the proposed BKTR model.

6.2 The superiority of the BKTR framework

As we can see from the test of different rank settings and observation rates in the simulation experiments (see Figures 4 and 9), BKTR is able to provide high estimation accuracy and valid CIs even with a much larger or over-specified rank, and also effectively estimates the coefficients and the unobserved output values when only 10% of the data is observed. This indicates the advantage of the proposed Bayesian low-rank framework. Since we introduce a fully Bayesian sampling treatment for the kernelized low-rank tensor model which is free from parameter tuning, BKTR can estimate the model parameters and hyperparameters even when only a small number of observations are available. Thus, the model can consistently offer reliable estimation results, implying its effectiveness and usability for real-world complex spatiotemporal data analysis. Another benefit of the proposed framework is the highly improved computing efficiency. As we mentioned in Section 3.5, the computational cost of BKTR is substantially reduced compared to the STVC model. According to the experiments conducted, BKTR is capable of dealing with regression problems containing up to millions of coefficients.

7 Conclusion

This paper introduces an effective solution for large-scale local spatiotemporal regression analysis. We propose parameterizing the model coefficients using low-rank CP decomposition, which greatly reduces the number of parameters from $M \times N \times P$ to $R(M + N + P)$. Contrary to previous studies on tensor regression, the proposed model BKTR goes beyond the low-rank assumption by integrating GP priors to characterize the strong local spatial and temporal dependencies. The framework also learns a low-rank multi-linear kernel which is expressive and able to provide insights for non-stationary and complicated processes. Our numerical experiments on both synthetic data and real-world data suggest that BKTR can reproduce the local spatiotemporal processes efficiently and reliably.

There are several directions for future research. In the current model, the CP rank R needs to be specified in advance. One can make the model more flexible and adaptive by introducing a reasonably large core tensor with a multiplicative gamma process prior such as in Rai et al. (2014). In terms of GP priors, BKTR is flexible and can accommodate different kernels (w.r.t. function form and hyperparameter) for different factors such as in Luttinen and Ilin (2009). The combination of different kernels can also produce richer spatiotemporal dynamics and multiscale properties. In terms of computation, one can further reduce the cost in GP learning (e.g., $\mathcal{O}(M^3)$ for a spatial kernel) by further introducing sparse approximation techniques such as inducing points and predictive processes (Quinonero-Candela and Rasmussen, 2005; Banerjee et al., 2008). Lastly, we can extend the regression model to handle binary- and count-valued data by using proper link functions. In this case, the challenge is that we no longer have analytical posteriors for latent factors; a potential solution is to use elliptical slice sampling (Murray et al., 2010) to sample the latent variables.

Acknowledgments

This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). Mengying Lei would like to thank the Institute for Data Valorization (IVADO) for providing a scholarship to support this study.

Supplementary Material

Supplementary Materials for Scalable Spatiotemporally Varying Coefficient Modeling with Bayesian Kernelized Tensor Regression (DOI: 10.1214/24-BA1428SUPP; .pdf). Supplementary materials present detailed calculation derivations of the marginal likelihood for kernel hyperparameters, the sampling algorithm for kernel hyperparameters, and the evaluation metrics applied in the paper.

References

Álvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012). "Kernels for Vector-Valued Functions: A Review." *Foundations and Trends® in Machine Learning*, 4(3): 195–266. 9

Bahadori, M. T., Yu, Q. R., and Liu, Y. (2014). "Fast Multivariate Spatio-temporal Analysis via Low Rank Tensor Learning." *Advances in Neural Information Processing Systems*, 3491–3499. 3, 4

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2014). *Hierarchical modeling and analysis for spatial data*. CRC press. MR3362184. 1, 9

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008). "Gaussian predictive process models for large spatial data sets." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70(4): 825–848. MR2523906. doi: https://doi.org/10.1111/j.1467-9868.2008.00663.x. 3, 13, 25

Bonilla, E. V., Chai, K. M. A., and Williams, C. K. (2007). "Multi-task Gaussian Process prediction." *Advances in Neural Information Processing Systems*, 153–160. 9

Cressie, N. and Johannesson, G. (2008). "Fixed rank kriging for very large spatial data sets." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70(1): 209–226. MR2412639. doi: https://doi.org/10.1111/j.1467-9868.2007.00633.x. 3

Cressie, N., Sainsbury-Dale, M., and Zammit-Mangion, A. (2022). "Basis-function models in spatial statistics." *Annual Review of Statistics and Its Application*, 9: 373–400. MR4394913. doi: https://doi.org/10.1146/annurev-statistics-040120-020733. 5

Cressie, N. and Wikle, C. K. (2015). *Statistics for spatio-temporal data*. John Wiley & Sons. MR2848400. 1

Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. (2016). "Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets." *Journal of the American Statistical Association*, 111(514): 800–812. MR3538706. doi: https://doi.org/10.1080/01621459.2015.1044091. 4

Faghih-Imani, A., Eluru, N., El-Geneidy, A. M., Rabbat, M., and Haq, U. (2014). "How land-use and urban form impact bicycle flows: evidence from the bicycle-sharing system (BIXI) in Montreal." *Journal of Transport Geography*, 41: 306–314. 21

Finley, A. O. (2011). "Comparing spatially-varying coefficients models for analysis of ecological data with non-stationary and anisotropic residual dependence." *Methods in Ecology and Evolution*, 2(2): 143–154. 2

Finley, A. O. and Banerjee, S. (2020). "Bayesian spatially varying coefficient models in the spBayes R package." *Environmental Modelling & Software*, 125: 104608. 4

Finley, A. O., Datta, A., Cook, B. D., Morton, D. C., Andersen, H. E., and Banerjee, S. (2019). "Efficient algorithms for Bayesian nearest neighbor Gaussian processes." *Journal of Computational and Graphical Statistics*, 28(2): 401–414. MR3974889. doi: https://doi.org/10.1080/10618600.2018.1537924. 4, 13

Fotheringham, A. S., Brunsdon, C., and Charlton, M. (2003). *Geographically weighted regression: the analysis of spatially varying relationships*. John Wiley & Sons. 2

Gelfand, A. E., Kim, H.-J., Sirmans, C., and Banerjee, S. (2003). "Spatial modeling with spatially varying coefficient processes." *Journal of the American Statistical Association*, 98(462): 387–396. MR1995715. doi: https://doi.org/10.1198/016214503000170. 2, 3, 4, 11, 17, 24

Gneiting, T. and Raftery, A. E. (2007). "Strictly proper scoring rules, prediction, and estimation." *Journal of the American statistical Association*, 102(477): 359–378. MR2345548. doi: https://doi.org/10.1198/016214506000001437. 15

Guhaniyogi, R., Laura, B., and Sudipto, B. (2023). "Bayesian Data Sketching for Varying Coefficient Regression Models." Technical report. 5

Guhaniyogi, R., Qamar, S., and Dunson, D. B. (2017). "Bayesian tensor regression." *The Journal of Machine Learning Research*, 18(1): 2733–2763. MR3714242. 3, 4

Heaton, M. J., Datta, A., Finley, A. O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D., Katzfuss, M., et al. (2019). "A case study competition among methods for analyzing large spatial data." *Journal of Agricultural, Biological and Environmental Statistics*, 24(3): 398–425. MR3996451. doi: https://doi.org/10.1007/s13253-018-00348-w. 15

Huang, B., Wu, B., and Barry, M. (2010). "Geographically and temporally weighted regression for modeling spatio-temporal variation in house prices." *International Journal of Geographical Information Science*, 24(3): 383–401. MR3408258. 3

Izenman, A. J. (1975). "Reduced-rank regression for the multivariate linear model." *Journal of Multivariate Analysis*, 5(2): 248–264. MR0373179. doi: https://doi.org/10.1016/0047-259X(75)90042-1. 3

Knorr-Held, L. and Rue, H. (2002). "On block updating in Markov random field models for disease mapping." *Scandinavian Journal of Statistics*, 29(4): 597–614. MR1988414. doi: https://doi.org/10.1111/1467-9469.00308. 11

Kolda, T. G. and Bader, B. W. (2009). "Tensor decompositions and applications." *SIAM Review*, 51(3): 455–500. MR2535056. doi: https://doi.org/10.1137/07070111X. 4, 5

Lei, M., Labbe, A., Wu, Y., and Sun, L. (2022). “Bayesian Kernelized Matrix Factorization for Spatiotemporal Traffic Data Imputation and Kriging.” *IEEE Transactions on Intelligent Transportation Systems*. 4

Lei, M. and Sun, L. (2023). “Bayesian Kernelized Tensor Factorization as Surrogate for Bayesian Optimization.” *arXiv preprint arXiv:2302.14510*. 5

Lei, M., Labbe, A., and Sun, L. (2024). “Supplementary Material for “Scalable Spatiotemporally Varying Coefficient Modeling with Bayesian Kernelized Tensor Regression"." *Bayesian Analysis*. doi: https://doi.org/10.1214/24-BA1428SUPP. 10, 11, 15, 24

Lopes, H. F., Salazar, E., and Gamerman, D. (2008). “Spatial dynamic factor analysis.” *Bayesian Analysis*, 3(4): 759–792. MR2469799. doi: https://doi.org/10.1214/08-BA329. 3, 4

Luttinen, J. and Ilin, A. (2009). “Variational Gaussian-process factor analysis for modeling spatio-temporal data.” *Advances in Neural Information Processing Systems*, 22: 1177–1185. MR2717937. 3, 4, 6, 10, 11, 13, 25

Martinez-Beneito, M. A., Botella-Rocamora, P., and Banerjee, S. (2017). “Towards a multidimensional approach to Bayesian disease mapping.” *Bayesian analysis*, 12(1): 239. MR3597574. doi: https://doi.org/10.1214/16-BA995. 5

Murray, I., Adams, R., and MacKay, D. (2010). “Elliptical slice sampling.” In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 541–548. JMLR Workshop and Conference Proceedings. 25

Murray, I. and Adams, R. P. (2010). “Slice sampling covariance hyperparameters of latent Gaussian models.” *Advances in Neural Information Processing Systems*, 1723–1731. 10

Neal, R. M. (2003). “Slice sampling.” *The annals of statistics*, 31(3): 705–767. MR1994729. doi: https://doi.org/10.1214/aos/1056562461. 10

Quinonero-Candela, J. and Rasmussen, C. E. (2005). “A unifying view of sparse approximate Gaussian process regression.” *The Journal of Machine Learning Research*, 6: 1939–1959. MR2249877. 25

Rabusseau, G. and Kadri, H. (2016). “Low-rank regression with tensor responses.” *Advances in Neural Information Processing Systems*, 29: 1867–1875. 4

Rai, P., Wang, Y., Guo, S., Chen, G., Dunson, D., and Carin, L. (2014). “Scalable Bayesian low-rank decomposition of incomplete multiway tensors.” *International Conference on Machine Learning*, 1800–1808. 25

Rao, N., Yu, H.-F., Ravikumar, P., and Dhillon, I. S. (2015). “Collaborative Filtering with Graph Information: Consistency and Scalable Methods.” *Advances in Neural Information Processing Systems*, 2107–2115. 4

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian Processes for Machine Learning*. MIT Press. MR2514435. 2

Rue, H. and Held, L. (2005). *Gaussian Markov random fields: theory and applications*. CRC press. MR2130347. doi: https://doi.org/10.1201/9780203492024. 13

Saatçi, Y. (2012). "Scalable inference for structured Gaussian process models." Ph.D. thesis, University of Cambridge. 4

Titsias, M. (2009). "Variational learning of inducing variables in sparse Gaussian processes." In *Artificial intelligence and statistics*, 567–574. PMLR. 13

Wang, X., Cheng, Z., Trépanier, M., and Sun, L. (2021). "Modeling bike-sharing demand using a regression model with spatially varying coefficients." *Journal of Transport Geography*, 93: 103059. MR4035553. 21

Wilson, A. G., Gilboa, E., Cunningham, J. P., and Nehorai, A. (2014). "Fast Kernel Learning for Multidimensional Pattern Extrapolation." *Advances in Neural Information Processing Systems*, 3626–3634. 4

Yu, R., Li, G., and Liu, Y. (2018). "Tensor regression meets gaussian processes." *International Conference on Artificial Intelligence and Statistics*, 482–490. 4

Yu, R. and Liu, Y. (2016). "Learning from multiway data: Simple and efficient tensor regression." *International Conference on Machine Learning*, 373–381. MR4209440. 4

Zhang, L. and Banerjee, S. (2022). "Spatial factor modeling: A Bayesian matrix-normal approach for misaligned data." *Biometrics*, 78(2): 560–573. MR4450576. doi: https://doi.org/10.1111/biom.13452. 4, 10

Zhou, H., Li, L., and Zhu, H. (2013). "Tensor regression with applications in neuroimaging data analysis." *Journal of the American Statistical Association*, 108(502): 540–552. MR3174640. doi: https://doi.org/10.1080/01621459.2013.776499. 3, 4