SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Department of
Economics, Management and Statistics

PhD program: **Economics and Statistics** Cycle: **XXXVII**
Curriculum: **Statistics**

BAYESIAN TRANSFER LEARNING APPROACHES FOR LARGE-SCALE SPATIOTEMPORAL PROBLEMS

Surname: **PRESICCE**
Name: **LUCA**
Registration number: **862302**

Supervisor: Prof. **TOMMASO RIGON**
Co-Supervisor: Prof. **SUDIPTO BANERJEE**

Coordinator: Prof. **MATTEO MANERA**

Academic Year: 2024/2025

A me medesimo meco,
compagno chiassoso de' miei solitari affanni,
interlocutore paziente de' miei dubbi,
spettatore giocondo de' miei passi incerti.

Vitam inpendere vero.

Decimo Giunio Giovenale

Acknowledgements

Honestly, I have always struggled when faced with acknowledgements. Let alone the intrinsic difficulty of finding the exact list of people who supported you over a (let's even say *sometimes maybe everlasting, sometimes maybe transient*) time span of 4 years, enriched by innumerable sparkling encounters worldwide. The eternal review process, feeling eternally guilty for missing someone. That said, I will do (again) my best to accomplish this (to me) "arduous" task.

No matter how much was taken for granted, I feel that starting with my family ought to be a good start. Thank you to all my family, whose lineage was the farthest from academic institutions. Notwithstanding the complete absence of knowledge about my research area, they always helped me whenever a doubt arose in my life. Continuously asking for explanations, continuously understanding nothing, but continuously present for me.

No less important are my friends, all my friends. Everyone gives, in their own way, the right support at the right moment. Thank you, even to you, whose I am (almost surely) missing.

Another important pillar of my PhD is the academic connections I have made along this journey, which I was fortunate to establish. Starting from the XXXVII cycle, going further to supporting figure I met, and concluding with many of the other (senior) hopes' spotlights. A special, warm, and heartfelt thank you goes to Professor Banerjee. He inspired me to high-quality, high-value research, without which I would not have been able to push cutting-edge research forward.

I would also like to sincerely thank the two reviewers of this thesis, Pierfrancesco and Michele. Their careful reading, insightful comments, and constructive criticism significantly contributed to improving the clarity, rigor, and overall quality of this work. I am grateful for the time and intellectual effort they devoted to engaging with my research and for their valuable perspective during the final stage of this (long, but pleasing) journey.

I would now like to turn to the person who rightfully deserves the greatest recognition, (at the risk of sounding sentimental) even if incapable of expressing my true feelings. I still remember how we began this journey together: driving around Milan, considering the pros and cons of either starting the PhD journey or accepting one of the many excellent job offers received. I am not ashamed to consider this achievement partly yours, as none of this would have been possible without your constant presence in my life. It was not easy, we know. Despite all the difficult moments we have faced, related or not to this journey, we are here, together. I am almost incredulous how those discussions about my endless passion for statistics, giving up a (economically) promising career in large companies, have brought us this far, already here. Long story short, thank you, *Valeria*, thank you for sincerely believing in me, and for always (naively) doing so unconditionally.

Abstract

The increasing availability of large-scale geospatial and spatiotemporal data presents new opportunities and challenges for statistical modeling in environmental, technological, medical, and other complex areas, which increasingly rely on massive multivariate spatiotemporal datasets. Yet, Bayesian learning for such problems remains severely limited by computational bottlenecks and the lack of flexible modeling tools. Modern applications require methods that are adaptive and effective, but still computationally efficient, scalable to massive datasets, and capable of delivering reliable automated inference with principled uncertainty quantification and (possibly) minimal experienced human intervention. Classical Bayesian approaches, although theoretically appealing and offering rich inferential frameworks, often become computationally infeasible in data-rich environments, especially when confronted with massive datasets or dynamic, high-dimensional dependence structures. Existing approaches often fail to scale, leaving a gap between the theoretical richness of Bayesian inference and its practical deployment in data-rich applications.

This thesis develops Bayesian transfer learning methodologies to address these challenges, enabling efficient information propagation and scalable inference across large spatial and spatiotemporal domains, providing a unified framework that merges distributional theory for matrix-variate models with computational innovations in Bayesian predictive stacking. Through extensive simulation experiments and data applications to global and satellite monitoring of vegetation indices, sea surface temperature, and land-atmospheric climate composition, the thesis also demonstrates the potential of Bayesian transfer learning to redefine spatial and spatiotemporal multivariate modeling, providing flexible, computationally efficient solutions that open the way for scalable, automated, and truly modern tools for geospatial learning in data-rich environments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Geospatial artificial intelligence (GeoAI) is a rapidly evolving discipline at the intersection of spatial data science and machine learning, aiming to leverage the analytical capabilities of artificial intelligence to analyze massive volumes of geographic and spatiotemporal data. This remarkable growth has been driven largely by the unprecedented explosion in the availability of georeferenced and spatiotemporal data that the past decade has witnessed. Advances in remote sensing, sensor networks, satellite imaging, and digital infrastructure have generated massive and complex datasets that describe environmental processes, human mobility, epidemiological trends, and climate dynamics. At the same time, computing power and data storage capabilities have expanded to levels that make global-scale analyses feasible. GeoAI arises as a response to these twin developments: the need to analyze massive geospatial data, and the opportunity to use advanced computational methods to do so. In broad terms, GeoAI encompasses any integration of artificial intelligence methods, such as deep learning, reinforcement learning, or automated reasoning with geospatial data. However, in its current form, the field remains skewed toward machine learning approaches that emphasize predictive accuracy at the expense of probabilistic inference, uncertainty quantification, and interpretability.

This imbalance creates a unique role for statisticians. While still an emerging research frontier, GeoAI craves the inferential zeal of statistical theory, presenting fertile ground to design data-analytic tools that preserve the rigor of probabilistic models while remaining scalable to modern data sizes. Statistical science is built on modeling uncertainty, assessing evidence, and making predictions that explicitly account for randomness. The challenge lies in balancing the inferential richness of hierarchical statistical models with the computational demands of high-dimensional spatial and spatiotemporal datasets. For GeoAI to become a mature scientific discipline, it must integrate these principles into its methodological core. Otherwise, it risks becoming a collection of ad hoc predictive tools that are powerful but unclear, accurate but breakable, and scalable but inconsistent in the presence of uncertainty.

The central motivation of this thesis is to explore how formal statistical inference, particularly Bayesian transfer learning, can be reconciled with the demands of massive spatial and spatiotemporal data. The methodological challenge is substantial: traditional spatial statistical methods are too computationally intensive for modern data, while most machine learning

methods ignore uncertainty quantification. Bridging this gap is not only of intellectual interest but also practically necessary, as applications such as disease surveillance, environmental monitoring, and disaster management demand both scalability and reliability.

The study of spatial and spatiotemporal random fields is deeply rooted in probability theory and statistics. Traditional spatial and spatiotemporal modeling relies heavily on Gaussian processes (GPs), as developed in foundational works (Cressie, 1993; Stein, 1999; Gelfand et al., 2010; Cressie and Wikle, 2011; Banerjee et al., 2015). The related covariance-based models and hierarchical formulations (Banerjee et al., 2015) provide flexible representations and allow formal probabilistic inference within both classical and Bayesian paradigms. These models provide a principled way to represent spatial correlation, interpolate values at unobserved locations while enabling full inference. All unknowns, including parameters, latent effects, and predictive random variables, are estimated: uncertainty is propagated to predictions.

In the Bayesian hierarchical modeling framework (Gelfand et al., 2010; Banerjee et al., 2015), spatial inference typically follows a rigorous structure $[data \mid process] \times [process \mid parameters] \times [parameters]$. The data model links observed data to latent processes and parameters. Modeling complex dependencies is the purpose of the process model (often specified as a Gaussian process or conditional autoregressive structure), while the parameter model assigns prior distributions to hyperparameters (such as variance, range, or smoothness). This three-level hierarchy enables joint posterior inference on all unknowns, allowing for full uncertainty quantification. Importantly, it also provides flexibility: one can accommodate irregularly spaced data, complex boundaries, or heterogeneous measurement error. Dynamic linear models (DLMs) and their multivariate and spatiotemporal extensions (DSTMs) (West and Harrison, 1997; Cressie and Wikle, 2011; Schmidt and Lopes, 2019; Wikle et al., 2019) have become a cornerstone of modern spatial and spatiotemporal modeling. They offer a flexible state-space modeling framework for capturing evolving temporal dependencies while accommodating spatial variation. Dynamic linear models make use of observational equations to link data to latent states, while specifying system equations to regulate latent states' evolution. Their computational appeal stems from the Forward Filtering Backward Sampling (FFBS) algorithm (Carter and Kohn, 1994), which leverages conjugate structures for efficient sequential updating and smoothing, making it possible to perform sequential updating of posterior distributions. This is particularly attractive for online inference and forecasting. The literature offers a comprehensive discussion on spatiotemporal applications of DLMs (see e.g. Nobre et al., 2005; Gamerman et al., 2008; Mahmoudian and Mohammadzadeh, 2014; Hefley et al., 2017, for an overview).

In hierarchical models, inference proceeds from spatial or spatiotemporal processes that scale massive data sets. However, the covariance structures associated with those stochastic processes become rapidly computationally prohibitive. This happens since moderately large-scale problems, particularly when data are irregularly distributed in space or collected at high temporal frequency. Markov Chain Monte Carlo (MCMC) or related iterative algorithms, while theoretically appealing, are often infeasible in practice for massive datasets. Indeed, Gaussian process models suffer from well-known computational challenges. The associated covariance matrix for n spatial locations is of size $n \times n$, and likelihood evaluation (which needs matrix

inversion) requires $O(n^3)$ operations, due to the Cholesky decomposition algorithm. This is denoted in the literature as the "big-n" problem. Moreover, many spatial covariance parameters, especially spatial variance and range parameter (controlling correlation decay), are weakly identifiable (Zhang, 2004). This makes posterior sampling slow and unstable, particularly in high dimensions. Spatiotemporal modeling compounds the challenges of spatial statistics by introducing temporal dependence. These challenges are especially acute in spatiotemporal modeling, where the dimensionality grows rapidly as both spatial and temporal dependencies must be represented.

The need for scalable inference has motivated a vast literature on approximations for large-scale spatial and spatiotemporal problems. Even a cursory review reveals a significant literature on methods for massive spatial and spatiotemporal datasets, which is too vast to be summarized here (see, e.g., Banerjee, 2017; Heaton et al., 2017). To address these limitations, we can discern that the literature has pursued two broad strategies for scalability: approximation methods, whose aim is to reduce computational cost by "simplifying" the covariance structure or likelihood; and distribution-theoretic methodologies, which leverage conjugate distributions to derive exact analytical results where possible (see e.g., Banerjee, 2017, 2020).

Examples of approximation approaches range from reduced-rank or subsets of regression approaches (see, e.g., Quiñonero-Candela and Rasmussen, 2005; Sansó et al., 2008; Cressie and Johannesson, 2008; Banerjee et al., 2008; Lemos et al., 2009; Wikle, 2010), spectral decomposition or multi-resolution approaches (Mezić, 2005; Nychka et al., 2015; Katzfuss, 2017; Zhang et al., 2024a), warped processes (King and Kowal, 2021), variational methods (Ren et al., 2011; Wu et al., 2022), stochastic partial differential equation (SPDE) approaches (Rue et al., 2009; Wikle and Hooten, 2010; Lindgren et al., 2011; Rue et al., 2017), and sparse approximation or graph-based models (Vecchia, 1988; Datta et al., 2016; Katzfuss and Guinness, 2021; Dey et al., 2022; Sauer et al., 2023). Despite several advances, difficulties remain. Full inference typically requires Markov chain Monte Carlo (Finley et al., 2019), variational approximations (Ren et al., 2011; Wu et al., 2022; Cao et al., 2023), expectation-maximization (EM) algorithms (Xu and Wikle, 2007), and a significant body of literature focuses on Gaussian Markov random field approximations (Rue et al., 2009; Lindgren et al., 2011) in conjunction with integrated nested Laplace approximations (INLA) for computing the marginal distributions of the process at given locations. All of which become rapidly computationally infeasible as data volumes scale. Moreover, model selection, parameter initialization, and convergence assessment demand extensive human expertise. This makes such models ill-suited for the automated, real-time analysis envisioned in GeoAI. On top of that, in dynamic contexts, data arrive sequentially in time and are often high-dimensional in space, creating the need for dynamic models that can update quickly as new data arrive. Therefore, while effective, these methods often introduce bias and require careful tuning.

The distribution-theory class of methods has been less common but offers unique advantages: it avoids iterative computation, reduces over-parameterization, and allows direct inference conditional on fixed hyperparameters. Recent contributions have sought to integrate statistical distribution theory with scalable machine learning methods, thereby reducing computational burdens while retaining essential inferential capabilities (Zammit-Mangion et al., 2022;

Zhang et al., 2024b; Presicce and Banerjee, 2024; Pan et al., 2025). Within distribution-theoretic approaches, extensions to spatial and multivariate contexts have been widely developed. A promising direction for scaling spatial and spatiotemporal analyses is the use of matrix-variate distributions (Quintana and West, 1987; West and Harrison, 1997; Banerjee, 2017, 2020; Schmidt and Lopes, 2019), which exploit separable spatial and temporal structures to reduce complexity and simplify inference. This allows the simultaneous modeling of multiple separable spatial fields, with modern context applications (e.g., Jiménez and Pereira, 2021; Banerjee et al., 2025). However, the reduction of complexity is not just computational but methodological: these approaches are limited to modeling data that exhibit separable structures.

The key observation is that spatial and spatiotemporal inference, in its classical form, is both rich and fragile: rich because it provides deep probabilistic insight, fragile because it collapses under modern data sizes. The methods mentioned above focus on the richness of statistical inference, but almost invariably involve a significant amount of human intervention to analyze spatial data: nontrivial implementation choices, parameter tuning, and diagnostic checks. This is precisely the kinds of human intervention that limit automation. Even the simplest geostatistical data require exploratory data analysis to learn about aspects of the underlying process that are weakly identified by the data (Zhang, 2004; Tang et al., 2021). Nonetheless, spatiotemporal models are often over-parameterized (Wikle and Hooten, 2010), especially in high-dimensional multivariate settings. Building a GeoAI system, on the other hand, will require minimizing human intervention in offering a robust framework for spatial data analysis.

This presents enough challenges that preclude a comprehensive solution in its entirety within the scope of a single dissertation. Nevertheless, we devise spatial and spatiotemporal data analytic frameworks that hold significant promise for GeoAI. The premise of this approaches relies upon two basic tenets: (i) model-based statistical inference for underlying spatial or spatiotemporal processes (including multivariate processes) in a robust and largely automated manner with minimal human input; and (ii) achieving such inference for truly massive amounts of data without resorting to iterative algorithms that may require significant human intervention to diagnose convergence (such as in MCMC).

Bayesian transfer learning plays a fundamental role as a possible tool for both mitigating the impact of modern data sizes and weakly identifiable parameters. The literature presents many possible definitions of transfer learning, especially when working with the Bayesian paradigm (see Suder et al., 2023, for a comprehensive review). Bayesian transfer learning can support either sharing complex dependency structures in complex models or the development of scalable spatial and spatiotemporal GeoAI systems. Actually, it provides solutions to a major obstacle that remains: how to combine inference across data partitions or over time in a way that preserves predictive validity while remaining computationally efficient.

In the wide Bayesian transfer learning field, a promising solution lies in Bayesian predictive synthesis (McAlinn and West, 2019; McAlinn et al., 2020). It generalizes the well-known concept of stacking coming from data-driven machine learning and statistical learning literature. Originally developed in the context of ensemble learning (Wolpert, 1992; Breiman, 1996), stacking combines multiple predictive models by weighting them to minimize suitable predic-

tive metrics. Bayesian predictive synthesis combines posterior distributions directly operating within hierarchical model frameworks. Notwithstanding the multiple methodological innovations introduced by Bayesian predictive synthesis (see e.g, Tallman and West, 2023; Cabel et al., 2025), its rich and flexible formulation typically relies on MCMC algorithms for posterior inference. As a consequence, these advances incur substantial computational costs, particularly in high-dimensional or large-scale settings. More generally, providing full Bayesian inference over all unknowns often proves prohibitively expensive to develop scalable, large-scale spatial and spatiotemporal models. Bayesian predictive stacking (BPS), as a particular case of Bayesian predictive synthesis (see Discussion in Yao et al., 2018), offers an alternative. It provides a principled way to assimilate predictive distributions by weighting multiple predictive models to minimize Kullback-Leibler divergence from the true predictive distribution, ascribable as a convex optimization problem. As Bayesian predictive synthesis, BPS operates directly on posterior predictive distributions, ensuring probabilistic rigor, but without the necessity of either simulation-based or iterative algorithms. Therefore, Bayesian predictive stacking presents a principled approach to perform distributed inference without compromising Bayesian coherence.

Examples of use of BPS in spatial analyses can be combined inference across data partitions: each subset yields a posterior predictive distribution, and stacking weights are chosen to maximize predictive performance (see, e.g., Zhang et al., 2024b; Presicce and Banerjee, 2024; Pan et al., 2025). In the spatiotemporal context, with small expedients, predictive stacking can be made dynamic: at each time step, predictive distributions are generated, and stacking weights evolve sequentially. This allows predictions to adapt over time while respecting the temporal structure of the data. The philosophy here is distinct from either traditional likelihood-based inference or simulative and iterative algorithms. Rather than insisting on recovering a "true" model or all the unknown parameters, Bayesian predictive stacking explicitly acknowledges model misspecification and focuses on optimizing predictive validity. When combined with distribution-theoretic modeling strategies, such as conjugate formulations with fixed hyperparameters, this approach enables efficient construction and aggregation of possibly misspecified yet tractable models. This is especially appropriate in high-dimensional geospatial settings, where no individual model can be expected to be correct, but where accurate and scalable predictions are of primary practical importance.

This thesis develops a unified framework for scalable spatial and spatiotemporal inference that is both probabilistically grounded and computationally feasible at unprecedented data scales. The main contributions lie in the development of Bayesian transfer learning approaches for spatial and spatiotemporal large-scale problems. For large-scale spatial problems, in Chapter 2, Bayesian inference is transferred across data partitions and aggregated via a double Bayesian predictive stacking (DBPS) procedure, enabling efficient posterior propagation in massive spatial datasets. For large-scale spatiotemporal problems, in Chapter 3, predictive posterior inference under matrix-variate DLMs is integrated over time in a proposal for information propagation, which relies on two major developments: first, we establish a fully conjugate predictive framework for sequential inference by using variational approximations; second, we introduce Dynamic Bayesian Predictive Stacking (DYNBPS) for spatiotemporal processes. All

this defines a novel predictive-synthesis mechanism that transfers information over time by extending Bayesian predictive stacking with a Markovian structure and dynamically evolving weights.

The research undertaken here advances the methodological foundations of statistical learning based GeoAI by developing distribution-theoretic frameworks for scalable spatial and spatiotemporal inference. Together, these contributions aim to reconcile the rigor of hierarchical statistical modeling with the scalability and automation required for modern GeoAI applications. By minimizing human intervention, exploiting analytical distribution theory, and leveraging Bayesian predictive stacking, the proposed Bayesian transfer learning frameworks pave the way for robust, adaptive, and automated analysis of massive geospatial and spatiotemporal datasets. In this same framework, temporal prediction, spatial interpolation, and uncertainty quantification are delivered without reliance on MCMC or other computationally intensive algorithms.

In the sections that follow, we develop the theoretical apparatus that forms the backbone of this thesis. We then position the specific contributions, highlighting how they extend existing paradigms and address the challenges outlined in the introduction.

1.2 Matrix-variate Distributions

In this dissertation, we made large use of closed-form distribution theory. By adopting matrix-Gaussian likelihood and Matrix-Gaussian-Inverse-Wishart distribution family as prior distribution in spatial and spatiotemporal models, we capture separable structures while avoiding the over-parameterization typical of multivariate modeling frameworks. To this end, we dedicate some points to useful content which may help the reader better understand model formulations, parametrizations, and some of the main results achieved later on.

Matrix-variate distributions extend classical multivariate probability distributions to a class of distributions defined on matrix-valued random variables rather than vector-valued ones (see Gupta and Nagar, 2000, for comprehensive details). However, vectorizing matrices, as is standard in multivariate modeling, generally inflates covariance dimension and leads to costly computations. Matrix-variate distributions, such as the matrix-Gaussian, imply a separable covariance structure between rows and columns, enabling parsimonious modeling parametrizations and efficient algebra. However, separability also introduces limitations of the method. Indeed, all outcomes are forced to respect the same row-covariance structure. Despite restrictions, working with separable structures is especially advantageous for spatial and spatiotemporal processes, longitudinal data, image analysis, and Bayesian hierarchical models, where dependence often arises within both rows and columns.

We adopt the following notation: let $Z \in \mathbb{R}^{m \times k}$ denote a random matrix with m rows and k columns. For matrices A, B, the Kronecker product is written $A \otimes B$, and $\text{vec}(X)$ denotes the column-wise vectorization of Z. The Frobenius norm is denoted $\|Z\|_F = \text{tr}(Z^\top Z)^{\frac{1}{2}}$. We now introduce three fundamental matrix-variate distributions that play a central role in Bayesian statistics and spatiotemporal modeling: the matrix-Gaussian distribution, the inverse Wishart distribution, and the matrix-variate Student's t-distribution.

1.2.1 Matrix Gaussian distribution

The matrix-Gaussian distribution (also called the matrix normal distribution) generalizes the multivariate Gaussian distribution to random matrices. Let $M \in \mathbb{R}^{m\times k}$ be a mean matrix, $V \in \mathbb{R}^{m\times m}$ a positive-definite row covariance matrix, and $U \in \mathbb{R}^{k\times k}$ a positive-definite column covariance matrix. A random matrix $Z \in \mathbb{R}^{m\times k}$ is endowed with a probability law from the matrix-normal distribution, denoted $Z \sim \mathrm{MN}_{m,k}(M, V, U)$, if and only if $\mathrm{vec}(Z) \sim \mathrm{N}_{mk}\left(\mathrm{vec}(M),\, U \otimes V\right)$. Hence, between matrix-variate and multivariate Gaussian distributions, there is a strict link, as the latter can be equivalently written as the vectorization of the former. The density of a matrix-Gaussian with respect to the Lebesgue measure on $\mathbb{R}^{m\times k}$ is

$$p(Z \mid M, V, U) = \frac{\exp\left[-\frac{1}{2}\,\mathrm{tr}\left\{U^{-1}(Z-M)^{\mathrm{T}}V^{-1}(Z-M)\right\}\right]}{(2\pi)^{\frac{mk}{2}} \mid U \mid^{\frac{m}{2}} \mid V \mid^{\frac{k}{2}}}. \tag{1.1}$$

Matrix-Gaussian distributions possess several desirable properties; here, we present only those relevant to our interest, as a comprehensive treatment is beyond the scope of this Section.

Matrix normal random matrices must obey the marginalization property. The marginalization property of the matrix normal distribution states that, given two ordered subsets of rows and columns $I \subset \{1, \ldots, m\}$ and $J \subset \{1, \ldots, k\}$, the corresponding subset matrix identified by I and J is also matrix normal distributed as $Z_{I,J} \sim \mathrm{MN}(M_{I,J}, V_{I,I}, U_{J,J})$. It actually implies that: (i) each row $i \in \{1, \ldots, n\}$ of Z, i.e., $Z_{i,\cdot}$, follows $Z_{i,\cdot} \sim \mathrm{N}(M_{i,\cdot}, v_{i,i}\times U)$ (where $v_{i,i}$ is the i-th element of V's diagonal); (ii) each column $j \in \{1, \ldots, k\}$ of Z, i.e., $Z_{\cdot,j}$, follows $Z_{\cdot,j} \sim \mathrm{N}(M_{\cdot,j}, u_{j,j} \times V)$ (where $u_{j,j}$ is the j-th element of U's diagonal); and (iii) each entry i, j, s.t. $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, k\}$ of Z, i.e., $z_{i,j}$, follows $z_{i,j} \sim \mathrm{N}(M_{i,j}, v_{i,i} \times u_{j,j})$.

The transpose transformation of a matrix normal random variable Z, i.e., Z^{T}, retain the same distribution, but with transpose mean matrix with the row and column covariance matrices swapped, such that $Z^{\mathrm{T}} \sim \mathrm{MN}_{k,m}(M^{\mathrm{T}},\ U,\ V)$.

The matrix-Gaussian family still possesses the property of linearity. Let D $(r \times m)$ be of full rank matrix $(r \leq m)$, and C $(k\times s)$ be also of full rank $s \leq p$, if we pre-multiply, and post-multiply Z by D, and C respectively, we obtain that: $DZC \sim \mathrm{MN}_{r,s}(DMC,\ DVD^{\mathrm{T}},\ C^{\mathrm{T}}VC)$.

Lastly, we conclude by noticing that the Kronecker covariance structure of the matrix-Gaussian distribution reduces the number of covariance parameters from $O\left((mk)^2\right)$ to $O(m^2 + k^2)$, which may result in big computational advantages. Furthermore, we highlight that the inverse of a Kronecker product is the Kronecker product of the inverses. This implies that the covariance matrix inversion can be attained by inverting two smaller matrices, with substantial computational advantages.

Matrix-Gaussians are natural for modeling multivariate spatial data or spatiotemporal data, where m indexes spatial locations and k may index either different dependent variables or time points (or vice versa).

1.2.2 Inverse Wishart distribution

The inverse Wishart distribution is the inverse of a Wishart distribution, which is the matrix-valued generalization of the Gamma distribution. The inverse Wishart often serves as a conju-

gate prior for covariance matrices in multivariate Gaussian models, and it is largely employed in Bayesian statistics, especially in Bayesian hierarchical modeling.

Let $S \in \mathbb{R}^{k\times k}$ be a positive-definite scale matrix and $\nu > k-1$ a positive scalar representing the degrees of freedom. A random positive-definite matrix $\Sigma \in \mathbb{R}^{k\times k}$ follows an inverse Wishart distribution, denoted $\Sigma \sim \mathrm{IW}_p(\nu, S)$, if and only if its density is

$$p(\Sigma \mid \nu, S) = \frac{|S|^{\nu/2}}{2^{\nu k/2}\,\Gamma_k\left(\frac{\nu}{2}\right)} |\Sigma|^{-(\nu+k+1)/2} \exp\left\{-\tfrac{1}{2}\operatorname{tr}(S\Sigma^{-1})\right\}, \tag{1.2}$$

where $\Gamma_k(\cdot)$ is the multivariate gamma function. Its expected value is $\mathbb{E}[\Sigma] = S/(\nu-k-1)$, only defined for $\nu > k+1$. Another useful property inherited from the Gamma distribution is the following: if $\Sigma \sim \mathrm{IW}_k(\nu, S)$, then $\Sigma^{-1} \sim \mathrm{W}_k(\nu, S^{-1})$, where W_k denotes the Wishart distribution of dimension k.

As mentioned before, the inverse Wishart is conjugate for the covariance matrix with respect to both the multivariate and the matrix normal likelihood. In the matrix-Gaussian setting, the Inverse Wishart is conjugate for both the column-covariance matrix and the row-covariance matrix. However, for the purposes of the present contributions, we will focus exclusively on the column-covariance conjugacy, since this is the only conjugacy that we actually require.

1.2.3 Matrix Student's t distribution

The matrix-variate Student's t distribution extends the multivariate t-distribution to random matrices, combining heavy tails and robustness to outliers with matrix-structured covariance. In Bayesian multivariate regression, based on matrix normal distribution with unknown column covariance matrix, matrix-variate Student's t distributions typically play a twofold fundamental role: (i) as posterior predictive distribution; and (ii) as prior predictive, contributing to model evidence derivation.

Let $M \in \mathbb{R}^{m\times k}$ be a location matrix, $V \in \mathbb{R}^{m\times m}$ and $U \in \mathbb{R}^{k\times k}$ positive-definite scale matrices, and $\nu > 0$ a scalar representing the degrees of freedom. A random matrix $Y \in \mathbb{R}^{m\times k}$ follows a matrix-variate Student's t distribution, denoted $Y \sim \mathrm{T}_{m,k}(\nu, M, U, V)$. The probability density function is

$$p(Y \mid \nu, M, V, U) = \frac{\Gamma_k\left(\frac{\nu+m+k-1}{2}\right)}{(\pi)^{\frac{mk}{2}}\Gamma_k\left(\frac{\nu+k-1}{2}\right)} |\Omega|^{-\frac{m}{2}} |\Sigma|^{-\frac{k}{2}} \times \left|\mathbb{I}_m + \Sigma^{-1}(X-M)\Omega^{-1}(X-M)^{\mathrm{T}}\right|^{-\frac{\nu+m+k-1}{2}}. \tag{1.3}$$

The matrix t distribution has comparable properties to those of the matrix normal distribution. It shares the same relationship with the multivariate t distribution that the matrix normal distribution shares with the multivariate normal distribution. If the matrix has only one row, or only one column, the distributions become equivalent to the corresponding vector distribution, then marginals and conditionals retain t distributional structure.

The transpose transformation of a matrix Student's t random variable Z, i.e., Z^{T}, retain the same distribution, but with transpose location matrix, and swapped scale matrices, such that $Z^{\mathrm{T}} \sim \mathrm{T}_{k,m}(\nu,\ M^{\mathrm{T}},\ U,\ V)$. In addition, the matrix t family possesses the linearity property: letting D $(r \times m)$ be of full rank matrix $(r \le m)$, and C $(k \times s)$ be also of full rank $s \le p$, then

if we pre-multiply, and post-multiply Z by D, and C respectively, we obtain that: $DZC \sim \mathrm{T}_{r,s}(\nu,\ DMC,\ DVD^{\mathrm{T}},\ C^{\mathrm{T}}VC)$.

The matrix t distribution is heavy-tailed compared to the matrix normal, providing enhanced robustness to outliers and model misspecification; but its law converges in-distribution to the matrix-Gaussian as $\nu \to \infty$. Similar to how a one-dimensional t-distribution approaches a normal distribution with increasing degrees of freedom. Intuitively, the matrix t distribution can be viewed as a compound distribution formed by an infinite scale-mixture of matrix normal distributions, where the mixing distribution is an inverse-Wishart over the scale (covariance) matrices. As the parameters of this mixture are adjusted (specifically, as degrees of freedom increase), the distribution leans towards the simpler matrix normal form.

The matrix t-distribution is used in robust multivariate and spatiotemporal modeling, especially in applications subject to anomalous observations (e.g., climate extremes, financial returns). It naturally arises in Bayesian inference when integrating over an inverse Wishart prior for covariance in a Gaussian model.

1.2.4 Matrix Normal Regression Model

Matrix-Gaussian models provide a natural and flexible framework for multivariate spatial and spatiotemporal regression, allowing multiple dependent variables or temporal replicates to be modeled jointly while preserving a parsimonious covariance structure. We introduce a conjugate matrix normal regression model that will serve as a foundational building block throughout the thesis. The key feature of this formulation is the availability of closed-form Bayesian inference for both matrix-valued regression coefficients and covariance parameters, achieved through carefully chosen conjugate priors.

Let us consider the following matrix-normal regression model

$$Y \mid B, \Sigma \sim \mathrm{MN}_{m,k}(XB,\ \mathbb{I}_m,\ \Sigma), \tag{1.4}$$

where $Y \in \mathbb{R}^{m\times k}$ is the observed response matrix, and $X \in \mathbb{R}^{m\times p}$ is the known design matrix. $B \in \mathbb{R}^{p\times k}$ are the matrix-valued regression coefficients, $\Sigma \in \mathbb{R}^{k\times k}$ is the column (response) covariance matrix, and the rows of Y are assumed to be independent (row covariance structure is known, and defined as $\mathbb{I}_m$). It is worth recalling that the model in (1.4), can be equivalently written as $\mathrm{vec}(Y) \mid \mathrm{vec}(B), \Sigma \sim \mathrm{N}_{mk}(\mathrm{vec}(XB),\ \Sigma\otimes\mathbb{I}_m)$, which highlights the separable covariance structure implied by the matrix normal distribution.

Assigning an inverse-Wishart prior distribution to Σ, then the posterior distribution of Σ conditional on response Y still belongs to the inverse Wishart family. As a matter of fact, a convenient conjugate joint prior for (B, Σ) in the matrix-normal regression model (1.4) is the Matrix-Normal Inverse Wishart (MNIW) prior, defined hierarchically with conditional dependence structure among B, Σ. It can be equivalent to explicit MNIW as a whole of the product between a conditional matrix normal and an inverse Wishart distribution. Then, assuming $B \mid \Sigma \sim \mathrm{MN}_{p,k}(M_0,\ V_0,\ \Sigma)$, and $\Sigma \sim \mathrm{IW}_k(\nu_0, S_0)$, it holds the following equivalence relation:

$$\mathrm{MN}_{p,k}(M_0,\ V_0,\ \Sigma) \times \mathrm{IW}_k(\nu_0, S_0) = \mathrm{MNIW}(M_0, V_0, S_0, \nu_0), \tag{1.5}$$

where $M_0 \in \mathbb{R}^{p\times k}$ is the prior mean for B, $V_0 \in \mathbb{R}^{p\times p}$ is a positive-definite row-scale matrix (covariance among rows of B), while ν_0, S_0 are the inverse-Wishart hyperparameters.

Under the Matrix-normal inverse Wishart prior in Equation (1.5), and the likelihood in Equation (1.4), the posterior is again a MNIW, s.t. $p(B, \Sigma \mid Y) \sim \text{MNIW}(M_m, V_m, S_m, \nu_m)$, with the given closed-form posterior updates: $M_m = V_m(X^\intercal \mathbb{I}_m^{-1} Y + V_0^{-1} M_0)$, $V_m = (X^\intercal \mathbb{I}_m^{-1} X + V_0^{-1})^{-1}$, $S_m = S_0 + Y^\intercal \mathbb{I}_m^{-1} Y + M_0^\intercal V_0^{-1} M_0 - M_m^\intercal V_m^{-1} M_m$, and $\nu_m = \nu_0 + m$. Thus, the marginal posteriors may also be written as $B \mid \Sigma, Y \sim \text{MN}_{p,k}(M_m, V_m, \Sigma)$, and $\Sigma \mid Y \sim \text{IW}_k(\nu_m, S_m)$.

The Matrix-Gaussian-inverse-Wishart prior is ubiquitous in Bayesian multivariate analysis and hierarchical modeling, appearing in multivariate regression, multivariate time series analysis, spatiotemporal hierarchical models, and, more generally, in high-dimensional multivariate settings. Its appeal lies not only in mathematical convenience but also in its ability to deliver exact Bayesian updates, closed-form marginal likelihoods, and computationally efficient inference without resorting to iterative simulation-based methods. Under suitable conditions, multivariate spatial regression models equipped with such conjugate structures admit closed-form posterior predictive distributions for both the latent spatial process and the observed outcomes, which take the form of a matrix Student's t distribution (see Presicce and Banerjee, 2024, for derivation). These properties are particularly valuable in high-dimensional spatial and spatiotemporal contexts, where scalability and analytical tractability are essential.

1.3 Dynamic Linear Models

Many approaches have emerged in the spatiotemporal modeling literature, bringing together concepts from statistical modeling, physics, signal processing, and engineering. Starting from population dynamics modeling (Czaran and Bartha, 1992), passing to ecological models (Chen et al., 2011), most of the contributions consider dynamical system theory as a basis. More recent works also integrate modern machine learning procedures within statistical frameworks (e.g. Ivanovic and Pavone, 2019; Zammit-Mangion and Wikle, 2020). Nevertheless, many successful methods rely upon dynamic linear models (DLMS).

Dynamic linear modeling bases its roots on dynamical systems theory. In particular, Dynamic linear models (DLMS) corresponds to a first-order affine discrete dynamical systems (Sandefur, 1990), which provide an adequate framework for modeling Markovian dependence structures. Since their first appearance, dynamic linear models have offered a versatile approach to analyzing time series. They model observations as functions of both time-varying parameters and underlying state processes (West and Harrison, 1997). These models are particularly effective for capturing temporal dependencies and accommodating structural changes in the data over time. Dynamic linear models, as state-space models, envelop in their simple structure many modeling frameworks (such as Ali, 1979), with a broad range of applications (e.g. Nobre et al., 2005; Gamerman et al., 2008; Pherwani et al., 2024; Idjigbèrou et al., 2025).

The DLM framework also naturally extends to spatiotemporal data (Cressie and Wikle, 2011; Schmidt and Lopes, 2019), which can handle observations at continuously varying locations across a region of interest, with spatial interpolation as a primary focus. This class is denoted in literature as dynamic spatiotemporal models (DSTMS) (details can be found in Cressie and

Wikle, 2011, Chapter 7). In this section, we introduce the DLMs and the accommodating slice of literature that serves later on, starting from matrix formulation. Besides, a comprehensive discussion on DSTM is beyond the scope of this Section.

1.3.1 Matrix-variate DLM

Matrix-variate dynamic linear models were introduced decades ago (see Quintana and West, 1987), but recently modern perspectives have attracted renewed interest, with the same velocity of multidimensional data availability (Schmidt and Lopes, 2019). Setting up matrix-variate DLM may not have univocal solutions. As seen in Section 1.2.1, random matrices endowed with the Gaussian law can be equivalently written either as matrix normal or as a multivariate normal distribution. Similarly to what happens for multivariate linear models, even for DLMs, the customary choice falls on matrix vectorization and the use of the multivariate Gaussian distribution. This led to dimension explosions, especially for the covariance matrix, not to mention the overparameterization for the consequent model. Conversely, we focus and introduce the reader to the matrix formulation of DLMs, using matrix normal distributions instead.

Let us consider the temporal dependent variables Y_t, such that each entry $Y_{i,j}^{(t)}$, for $i \in \{1, \dots, n\}$, $j \in \{1, \dots, q\}$, and $t \in \mathcal{T}$, is the observed value at a specific location i, for the variable j, at time t. The set of $\mathcal{T}$ observation $Y = \{Y_t\}_{t=1,\dots,\mathcal{T}}$ is thought as a $(n \times q)$ matrix time series, where each Y_t correspond to a matrix observation at discrete time points $t \in \mathcal{T} \subset \mathbb{N}$. Then, our observational sample Y will take the form of a $(n \times q \times T)$ three-dimensional tensor. By convention, we define $\mathscr{D}_t$ either as the information set or dataset available at time t, which includes the current observation Y_t, any covariates observed at t, and all the relevant past information $\mathscr{D}_{t-1}$ (due to Markovian dependence structure). For the sake of completion, we introduce $\mathscr{D}_0 \subset \mathscr{D}_1$ as the prior information, while $\mathscr{D}_1$ represent the first observed dataset.

A dynamic linear model for matrix time series decomposes each observation matrix Y_t into two time-dependent elements: a trend component, $F_t\Theta_t$, and an error component, Υ_t, which follows a matrix-variate normal distribution with row covariance V_t and column covariance Σ. The (latent) $(p \times q)$ matrix Θ_t, known as the "state" matrix at time t, evolves over time following a transition equation only depending on the previous state matrix Θ_{t-1}. Specifically, we formulate a matrix-variate DLM by the following set of equations:

$$\begin{aligned} Y_t &= F_t\Theta_t + \Upsilon_t, & \Upsilon_t &\sim \mathrm{MN}(0, V_t, \Sigma) \\ \Theta_t &= G_t\Theta_{t-1} + \Xi_t, & \Xi_t &\sim \mathrm{MN}(0, W_t, \Sigma), \end{aligned} \tag{1.6}$$

Where 0 denotes a matrix of zeros, Θ_t is the $(p \times q)$ state matrix at time t, and Υ_t and Ξ_t are zero-mean matrix-Gaussian error matrices with row and column covariances as specified in Equation (1.6). Here, F_t is a $(n \times p)$ matrix of covariates, G_t is a $p \times p$ evolution matrix, and W_t is a $(p \times p)$ row covariance matrix defining the dependence structure across the rows of Θ_t. Equations (1.6) are commonly called the "observation" and "state" equations. The model (1.6), assumes Σ, i.e., the column-covariance matrix, to be shared between Υ_t and Ξ_t across time. It is also assumed that Υ_t and Ξ_t are mutually independent, and independent from Θ_0, Σ. Additionally, given Θ_t, the variable Y_t is conditionally independent from past observations.

A DLM is fully characterized by the quadruple $\{F_t, G_t, V_t, W_t\}$. If we assume that $V_t = V$ and $W_t = W$ for all $t \in \mathcal{T}$, the model is referred to as the constant model. This type includes the local level and local trend models, the trigonometric seasonal model, and the time-varying parameter auto-regressive model. Often, the matrices in the quadruple can be decomposed into block-diagonal forms to represent conditional independence assumptions. The Markovian structure of Θ_t, paired with Gaussian assumptions, facilitates efficient inference and prediction through Kalman filtering. Consequently, DLMs are particularly well-suited for online learning applications, enabling real-time updates to estimates as new data is observed. Sequential updating, forecasting, and retrospective smoothing naturally follow from derivations presented in West and Harrison (1997); Schmidt and Lopes (2019) for univariate and multivariate models.

Bayesian inference for dynamic linear models, as well as dynamic spatiotemporal subclasses, typically happens using the Forward Filtering Backward Sampling (FFBS) algorithm (Carter and Kohn, 1994). This algorithm allows conjugate updates within a well-defined Gibbs sampler. We now present its adaptation using matrix-variate distributions presented in Section 1.2. The FFBS algorithm begins by "forward filtering", which serves to propagate information across time. It pushes the information forward, sequentially as available, from the (first) previous state to the (last) consecutive one. This defines recursive distributions, starting from the filtered prior, and the one-step ahead predictive, depicted as:

$$\Theta_t \mid \mathscr{D}_{t-1}, \Sigma \sim \text{MN}(A_t, R_t, \Sigma) \tag{1.7}$$

$$Y_t \mid \mathscr{D}_{t-1}, \Sigma \sim \text{MN}(q_t, Q_t, \Sigma), \tag{1.8}$$

where $A_t = G_t m_{t-1}$, $R_t = G_t C_{t-1} G_t^{\mathrm{T}} + W_t$, $q_t = F_t A_t$, and $Q_t = F_t R_t F_t^{\mathrm{T}} + V_t$. Passing then to the filtered posteriors for the current state Θ_t, and the common column covariance matrix Σ

$$\Theta_t \mid \mathscr{D}_t, \Sigma \sim \text{MN}(m_t, C_t, \Sigma) \tag{1.9}$$

$$\Sigma \mid \mathscr{D}_t \sim \text{IW}(\nu_t, \Psi_t), \tag{1.10}$$

with $m_t = C_t[R_t^{-1} A_t + F_t^{\mathrm{T}} V_t^{-1} Y_t]$, $C_t = [R_t^{-1} + F_t^{\mathrm{T}} V_t^{-1} F_t)]^{-1}$, $\nu_t = \nu_{t-1} + \frac{n}{2}$, and $\Psi_t = \Psi_{t-1} + \frac{1}{2}(Y_t - q_t)^{\mathrm{T}} Q_t^{-1}(Y_t - q_t)$. The procedure start considering $t = 0$ as prior information, where $\Theta_0 \mid \Sigma \sim \text{MN}(m_0, C_0, \Sigma)$ represents the initial information on state matrix, with m_0 and C_0 as the known $(p \times q)$ mean matrix and $(p \times p)$ row covariance matrix, respectively. While prior information for the common column covariance matrix is depicted by $\Sigma \sim \text{IW}(\nu_0, \Psi_0)$, with scalar parameter ν_0, and scale matrix Ψ_0. Then forward filtering recursively uses the filtered posterior of time $t-1$ as a prior for time t, under conjugacy of Θ_t and Σ.

Once the filtered posterior $p(\Theta_t \mid \mathscr{D}_t, \Sigma)$ is recovered for all $t = 1, \ldots, T$, backward sampling is then applied to smooth out posterior distributions using future knowledge, until the last observed point T. This allows information from all observed time points to be incorporated into the posterior distribution and propagated to the posterior samples. Backward smoothed posterior takes the form:

$$\Theta_t \mid \Theta_{t+1}, \mathscr{D}_T, \Sigma \sim \text{MN}(h_t, H_t, \Sigma) \tag{1.11}$$

where "backward" smoothed parameters are $h_t = H_t[C_t^{-1} m_t + G_{t+1}^{\mathrm{T}} W_{t+1}^{-1} \Theta_{t+1}]$, and $H_t = [C_t^{-1} +$

$G_{t+1}^{\mathrm{T}} W_{t+1}^{-1} G_{t+1}]^{-1}$. Wrapping up, standard posterior inference for DLMs start with FF by the application of Equations from (1.7) to Equation (1.10), from $t = 1, \ldots, T$. After Θ_T is sampled from $\mathrm{MN}(m_T, C_T, \Sigma)$, smoothed posterior samples are recovered by BS in (1.11), and each Θ_t is subsequently draw from $\mathrm{MN}(h_t, H_t, \Sigma)$. This achieves posterior sampling of $\Theta_{0:T}$ from the joint posterior distribution $p(\Theta_0, \ldots, \Theta_T \mid \mathcal{D}_T, \Sigma)$. This process must be repeated for the desired number of posterior samples.

Often, experts and researchers, when analyzing time series data, are primarily in temporal forecasting of outcomes and parameters, for a certain fixed prediction horizon k. We conclude this section by recalling the prediction procedure for dynamic linear models in the matrix-variate case. Here, we represent the joint predictive distribution for Θ_{T+k}, Y_{T+k}, an arbitrary future time point at k instants after the last observation T:

$$\begin{bmatrix} \Theta_{T+k} \\ Y_{T+k} \end{bmatrix} \Bigg| \mathcal{D}_T, \Sigma \sim \mathrm{MN}\left(\underbrace{\begin{bmatrix} A_T(k) \\ q_T(k) \end{bmatrix}}_{A^\star(k)}, \underbrace{\begin{bmatrix} R_T(k) & F_{T+k} R_T(k) \\ R_T(k) F_{T+k}^{\mathrm{T}} & Q_T(k) \end{bmatrix}}_{R^\star(k)}, \Sigma \right). \tag{1.12}$$

The predictive parameters are $A_T(k) = G_{T+k} A_T(k-1)$, $R_T(k) = G_{T+k} R_T(k-1) G_{T+k}^{\mathrm{T}} + W_{T+k}$, $q_T(k) = F_{T+k} A_T(k)$, and $Q_T(k) = F_{T+k} R_T(k) F_{T+k}^{\mathrm{T}} + V_{T+k}$, with $A_T(0) = m_T$, $R_T(0) = C_T$. As perceivable from Equation (1.12), the set of parameter updating follows a recursion. To obtain the prediction k steps ahead, the parameters for the previous $k-1$ steps must be computed.

1.3.2 Spatiotemporal matrix-variate DLM

Spatiotemporal modeling provides a natural framework for analyzing processes observed across spatial locations and time, where both spatial and temporal dependence structures play a central role. For univariate spatiotemporal processes, matrix-variate distribution may be introduced to incorporate joint spatial and temporal dependence structure into the column, or row (depending on chosen problem formulation), covariance matrix. This allows for lighter model parametrization and contemporaneous temporal structure learning. In multivariate spatiotemporal settings, each observation is naturally represented as a random matrix. Matrix-variate dynamic spatiotemporal models exploit this structure by employing matrix-variate distributions, enabling computational scalability while jointly modeling correlated spatiotemporal processes. By leveraging the correlation structure among dependent variables, these models enhance inferential efficiency and allow direct inference on the underlying cross-variable dependence.

Based upon the matrix-variate DLMs presented in Section 1.3.1, the introduction to the spatiotemporal framework starts by defining the spatial domain. Let $\mathcal{S} = \{s_1, \ldots, s_n\}$ be a set of n fixed georeferenced locations over the region of interest $\mathcal{D}$, where typically $\mathcal{D} \subset \mathbb{R}^2$ or $\mathbb{R}^3$. These locations constitute the spatial sample and are assumed to remain fixed over time. Observations collected at these locations form the rows of the response matrices Y_t, for $t = 1, \ldots, T$. For each of these, we collect information on q correlated dependent variables, over a discrete equispaced grid of time points of length T.

There may be different possibilities to adapt the standard matrix-variate dynamic linear

formulation to allow accounting for spatiotemporal dependencies. We describe our proposal, without assuming it as the unique possible construction. We manage the model in Equations (1.6), such that $F_t = [\, X_t \;\; \mathbb{I}_n \,]$ has dimension $n \times (p+n)$, with p the number of the available temporal predictors in X_t. The definition of Θ_t becomes crucial: $\Theta_t = [B_t^{\mathrm{T}} : \Omega_t^{\mathrm{T}}]^{\mathrm{T}}$, with B_t matrix of dynamic regression coefficients, and Ω_t a latent multivariate spatial process. The block-matrix representation of Θ_t has dimension of $(p+n) \times q$. To model temporal evolution, We further assume $B_t \perp\!\!\!\perp \Omega_t \; \forall t \in \mathcal{T}$. In particular, we assume that B_t depends only on B_{t-1} and Ω_t depends only on Ω_{t-1}, with no cross-dependence across blocks. This conditional independence assumption implies a block-diagonal evolution matrix $G_t = \begin{bmatrix} \mathbb{I}_p & 0_{p\times n} \\ 0_{n\times p} & \mathbb{I}_n \end{bmatrix}$. Instantaneous dependence at time t is instead governed by the innovation row-covariance matrix W_t, whose block-diagonal structure reflects the assumed independence between the innovations driving B_t and Ω_t.

under these assumptions the matrix-variate DLM formulation in Equation (1.6), reduces to the following spatiotemporal model:

$$
\begin{aligned}
Y_t &= \underbrace{\begin{bmatrix} X_t & \mathbb{I}_n \end{bmatrix}}_{F_t} \underbrace{\begin{bmatrix} B_t \\ \Omega_t \end{bmatrix}}_{\Theta_t} + \Upsilon_t, \qquad && \Upsilon_t \sim \mathrm{MN}(0, V_t(\alpha), \Sigma) \\
\underbrace{\begin{bmatrix} B_t \\ \Omega_t \end{bmatrix}}_{\Theta_t} &= \underbrace{\begin{bmatrix} \mathbb{I}_p & 0_{p\times n} \\ 0_{n\times p} & \mathbb{I}_n \end{bmatrix}}_{G_t} \underbrace{\begin{bmatrix} B_{t-1} \\ \Omega_{t-1} \end{bmatrix}}_{\Theta_{t-1}} + \Xi_t, \qquad && \Xi_t \sim \mathrm{MN}(0, W_t(\phi), \Sigma),
\end{aligned}
\tag{1.13}
$$

where the column-covariance matrix Σ is shared across observation and state equations and is assumed constant over time. The row-covariance matrices are specified as $V_t(\alpha) = (\frac{1-\alpha}{\alpha})\mathbb{I}_n$, and $W_t(\phi) = \begin{bmatrix} W_t^B & 0_{p\times n} \\ 0_{n\times p} & \mathcal{R}_t(\mathcal{S},\mathcal{S};\phi) \end{bmatrix}$, where W_t^B is a generic positive-definite $(p \times p)$ covariance matrix governing the innovations of the dynamic regression coefficients, and $\mathcal{R}_t(\mathcal{S},\mathcal{S};\phi)$ is a spatial correlation matrix generated by a valid spatial correlation function, e.g., exponential or Matérn, parametrized by the set of parameters ϕ.

The specification of $V_t(\alpha)$ allows the introduction of a discontinuity. It corresponds to a "nugget"-effect formulation, commonly used in spatial statistics to model micro-scale variability or measurement error. Under this parametrization, $\alpha \in (0,1)$ controls the relative contribution of the spatially structured component versus the unstructured noise, with smaller values of α increasing the nugget variance relative to the spatial signal. Conditioning on fixed hyperparameters $\{\alpha, \phi\}$, the model in Equation (1.13) is conjugate with respect to Θ_t, and Σ, yielding closed-form filtering, smoothing, and predictive distributions as described in Equations (1.7)-(1.11).

Expanding the matrix form of the model in Equation (1.13), the observation equation yields $Y_t = X_t B_t + \Omega_t + \Upsilon_t$, which corresponds, at each time point, to a multivariate spatial regression model. This implies that at each time, we are fitting a standard linear spatial model with well-known properties and well-known interpolation properties, for which distribution theory can help us. Similarly, the state equation decomposes into two equations: $B_t = B_{t-1} + \Xi_t^B$, and $\Omega_t = \Omega_{t-1} + \Xi_t^\Omega$, corresponding to random-walk evolutions for both the regression coefficients

and the latent spatial process. Alternative temporal dynamics can be accommodated through different specifications of G_t. Within the block structure of G_t, the temporal evolution of B_t and Ω_t can still be modulated independently by adjusting the corresponding sub-blocks of G_t depending on knowledge, expertise, or subjective prior beliefs.

Different from customary dynamic linear models, which give the main focus on temporal forecasting, in spatiotemporal data analysis, we are equally interested in both temporal forecasting and spatial prediction, i.e., interpolation of the multivariate spatial process at unobserved spatial points. The aim is to interpolate the distribution of the latent spatial process for unobserved points, to obtain critical insights into the spatial distribution of the underlying phenomena, and to create interpolated maps to help the decision-making process. This will produce an interpolation of the evolution of the multivariate spatial process, which allows understanding complex dynamics over space and time. Let $\mathcal{U} = \{u_1, \ldots, u_m\} \subset \mathcal{D}$ denote a set of m unobserved spatial locations at time $t \in \mathcal{T}$. For these locations, we assume covariate information is available, collected in the design matrix $\tilde{F}_t = [\, \tilde{X}_t \;\; \mathbb{I}_m \,]$, where $\tilde{X}_t$ contains the corresponding predictors. Let $\tilde{Y}_t$ and $\tilde{\Omega}_t$ denote, respectively, the outcomes and latent spatial process evaluated at the locations in $\mathcal{U}$.

We study the joint posterior predictive distribution of the stacked vector $(\tilde{Y}_t^{\mathrm{T}}, \tilde{\Omega}_t^{\mathrm{T}})^{\mathrm{T}}$, which is given by the integral

$$p(\tilde{Y}_t, \tilde{\Omega}_t \mid \mathscr{D}_t) = \int \left[\int p(\tilde{Y}_t, \tilde{\Omega}_t \mid \mathscr{D}_t, \Theta_t, \Sigma) p(\Theta_t \mid \mathscr{D}_t, \Sigma) \, \mathrm{d}\Theta_t \right] p(\Sigma \mid \mathscr{D}_t) \, \mathrm{d}\Sigma. \tag{1.14}$$

For the model in Equation (1.13), the integral in (1.14) can be computed analytically (using similar arguments used in A.1.1), yielding a matrix-variate Student's t distribution,

$$p(\tilde{Y}_t, \tilde{\Omega}_t \mid \mathscr{D}_t) = \mathrm{T}_{2\nu_t}(\mu_t, E_t, \Psi_t, \nu_t), \tag{1.15}$$

where $\mu_t = \chi_t m_t$, $E_t = \chi_t C_t \chi_t^{\mathrm{T}} + N_t$, with $\chi_t = \begin{bmatrix} \tilde{X}_t & \tilde{M}_t \\ 0 & \tilde{M}_t \end{bmatrix}$, $N_t = \begin{bmatrix} \tilde{V}_t(\alpha) + \tilde{W}_t(\phi) & \tilde{W}_t(\phi) \\ \tilde{W}_t(\phi) & \tilde{W}_t(\phi) \end{bmatrix}$. Here $\tilde{V}_t(\alpha) = (\frac{1-\alpha}{\alpha})\mathbb{I}_m$ is the row-covariance matrix for the new outcomes matrix $\tilde{Y}_t$, while $\tilde{M}_t = \mathcal{R}_t(\mathcal{U}, \mathcal{S}; \phi)\mathcal{R}_t^{-1}(\mathcal{S}, \mathcal{S}; \phi)$, $\tilde{W}_t(\phi) = \mathcal{R}_t(\mathcal{U}, \mathcal{U}; \phi) - \mathcal{R}_t(\mathcal{U}, \mathcal{S}; \phi)\mathcal{R}_t^{-1}(\mathcal{S}, \mathcal{S}; \phi)\mathcal{R}_t(\mathcal{S}, \mathcal{U}; \phi)$ are the mean, and row-covariance matrices for the posterior distribution of the spatial process, i.e. $\Omega_t \mid \mathscr{D}_t$.

All the contents illustrated in the following Sections can be adapted to multivariate or univariate time series modeling straightforwardly (see West and Harrison, 1997; Schmidt and Lopes, 2019, and references therein), even though it is beyond the scope of this Section. This concludes the discussion on the matrix-variate formulations for both DLMs and DSTMs, along with their foundational characteristics. In Chapter 3, we focus our modeling efforts on making extensive use of the spatiotemporal model in Equation (1.13).

1.4 Bayesian Predictive Stacking

To provide a better understanding of the contents in the following Sections, it is worthwhile to spend some words introducing Bayesian predictive stacking (BPS). The Bayesian predictive

stacking brings common statistical learning concepts into the Bayesian framework.

Model stacking, also known as stacked generalization (Wolpert, 1992), is a machine learning ensemble technique. Stacked generalization uses predictive metrics to evaluate the predictive performances of competitive models, then combines the models' predictions as weighted averages. In different formulations, a meta-model may be used to combine predictions of competitive models. The scope of stacked generalization is to improve generalization accuracy. Distinctively, Bayesian predictive stacking (primarily introduced by Yao et al., 2018) falls into the class of Bayesian model averaging (BMA), which selects the model combination weights by maximizing a score of the log posterior densities. Bayesian Predictive Stacking differs from Bayesian model averaging techniques, as directly focused on predictive accuracy. It optimizes a score function of the posterior predictive distributions, instead of relying on posterior model probabilities. More general approaches were introduced later on (see, e.g., McAlinn and West, 2019; McAlinn et al., 2020) to incorporate weights uncertainty, within a full Bayesian hierarchical setting.

Let Y be a real-valued random variable defined on a measurable space $(\mathcal{E}, \mathcal{A})$, and let $\mathcal{P}$ be a convex class of probability measures on $(\mathcal{E}, \mathcal{A})$, whose elements represent probabilistic forecasts. A scoring rule is a function $S : \mathcal{P} \times \mathcal{E} \to \overline{\mathbb{R}} = [-\infty, +\infty]$ such that, for every probabilistic forecaster $P \in \mathcal{P}$, the function $S(P, \cdot)$ is $\mathcal{P}$-quasi-integrable, i.e., its expectation $\mathbb{E}_P[S(P, Y)] = \int_{\mathcal{E}} S(P, y)\, dP(y)$ exists in the extended real numbers. In the continuous case, each distribution $P \in \mathcal{P}$ is identified with its density function p. For two probabilistic forecasts P and Q, we define $S(P, Q) = \int_{\mathcal{E}} S(P, y)\, dQ(y)$. A proper scoring rule induces a divergence $d : \mathcal{P} \times \mathcal{P} \to (0, \infty)$ defined as $d(P, Q) = S(Q, Q) - S(P, Q)$, and satisfying $d(P, Q) \geq 0$ with equality if and only if $P = Q$. For further background on proper scoring rules, we refer the reader to Gneiting and Raftery (2007); Yao et al. (2018) and references therein.

In the literature, there exist many proper scoring rules. In Yao et al. (2018), the authors refer to three commonly used score rule functions for continuous variables: (i) quadratic score rule $\mathrm{QS}(p, y) = 2p(y) - \| p \|_2^2$ with $d(p, q) = \| p - q \|_2^2$; (ii) logarithm score rule $\mathrm{LogS}(p, y) = \log(p(y))$ with $d(p, q) = \mathrm{KL}(q, p)$ (the only proper local score rule assuming regularity conditions); (iii) continuous ranked score rule $\mathrm{CRPS}(F, y) = -\int_{\mathbb{R}} (F(x) - \mathbb{1}(x \geq y))^2 \, \mathrm{d}x$ with $d(F, G) = \int_{\mathbb{R}} (F(y) - G(y))^2 \, \mathrm{d}y$ where F, G are corresponding distribution function; and (iv) energy score rule $\mathrm{ES}(P, y) = 1/2 \, \mathbb{E}_P \| Y - X \|_2^{\beta}$ where Y and X are two independent random variables from distribution P (strictly proper when $\beta \in (0, 2)$ but not when $\beta = 2$).

Each (possibly proper) scoring rule could be used to define a Bayesian stacking problem. Indeed, the Bayesian stacking problem concerns a maximization of a scoring rule, or equivalently, the minimization of the induced divergence. There is a preference for proper score rules, as a not strictly proper scoring rule can give rise to identification problems. Among proper scoring rules, the logarithmic score has desirable properties. In particular, Gneiting and Raftery (2007) shows that every proper local scoring rule is proportional to the logarithmic score, in the sense that any proper local scoring rule can be expressed as a strictly increasing transformation of the log score. This implies that using a different proper local scoring rule leads to the same ordering of predictive distributions as the log score. Consequently, the logarithmic score is particularly attractive in practice.

In general, the goal is to find the distribution in the convex hull $\mathbf{C} = \left\{ \sum_{j=1}^{J} w_j p\left(\cdot \mid \mathscr{M}_j\right) : \sum_j w_j = 1,\ w_j \geq 0 \right\}$ that is optimal according to a chosen proper scoring rule (or the divergence induced from it); considering a set of J predictive distributions built from the models $\mathscr{M} = \left(\mathscr{M}_1, \ldots, \mathscr{M}_j\right)$.

Defining the J-dimensional simplex as $\mathcal{S}_1^J = \left\{ w \in [0,1]^J : \sum_{j=1}^{J} w_j = 1 \right\}$, then we can write the stacking problem as the following optimization problem

$$\min_{w \in \mathcal{S}_1^J} d\left(\sum_{j=1}^{J} w_j p\left(\cdot \mid y, \mathscr{M}_j\right), p^\star(\cdot \mid y) \right) \text{ or } \max_{w \in \mathcal{S}_1^J} S\left(\sum_{j=1}^{J} w_j p\left(\cdot \mid y, \mathscr{M}_j\right), p^\star(\cdot \mid y) \right) \tag{1.16}$$

where $p\left(\tilde{y} \mid y, \mathscr{M}_j\right)$ is the predictive density of new data $\tilde{y}$ in model $\mathscr{M}_j$ that has been trained on observed data y, while $p^\star(\tilde{y} \mid y)$ refers to the true predictive distribution.

Generally, $p^\star(\tilde{y} \mid y)$ is not known. However, we can consider the following reasoning to obtain an empirical approximation. First of all, recall that for two probabilistic forecasts, here $\tilde{p} \equiv \sum_{j=1}^{J} w_j p\left(\cdot \mid y, \mathscr{M}_j\right)$ and $p^\star \equiv p^\star(\cdot \mid y)$, we write

$$S(\tilde{p}, p^\star) = \int S(\tilde{p}, \tilde{y}) dp^\star(\tilde{y}) = \mathbb{E}[S(\tilde{p}, \tilde{y})], \quad \tilde{y} \sim p^\star. \tag{1.17}$$

At this point, we consider the following approximation of this expected value

$$\mathbb{E}[S(\tilde{p}, \tilde{y})] \approx \frac{1}{n} \sum_{i=1}^{n} S(\tilde{p}^{-i}, y_i), \tag{1.18}$$

where $\tilde{p}^{-i} = \sum_{j=1}^{J} w_j \hat{p}_j^{-i}$ and $\hat{p}_j^{-i}\left(y_i\right) = \int p\left(y_i \mid \theta_j, \mathscr{M}_j\right) p\left(\theta_j \mid y_{-i}, \mathscr{M}_j\right) d\theta_j$. That corresponds to the leave-one-out (LOO) estimate.

Thus, an empirical approximation can be constructed by replacing the full predictive distribution $p\left(\tilde{y} \mid y, \mathscr{M}_j\right)$ evaluated at a new datapoint $\tilde{y}$ with the corresponding LOO predictive distribution $\hat{p}_j^{-i}\left(y_i\right)$. The corresponding stacking weights are then recovered as the solution to this optimization problem

$$(\hat{w}_1, \ldots, \hat{w}_J)^{\mathrm{T}} = \arg\max_{w \in \mathcal{S}_1^J} \frac{1}{n} \sum_{i=1}^{n} S\left(\sum_{j=1}^{J} w_j \hat{p}_j^{-i}, y_i \right) \tag{1.19}$$

From which the stacked estimate of the predictive density is derived as follows

$$\hat{p}(\tilde{y} \mid y) = \sum_{j=1}^{J} \hat{w}_j p\left(\tilde{y} \mid y, \mathscr{M}_j\right). \tag{1.20}$$

The stacking estimate finds the optimal predictive distribution within the convex set C, that is, the closest to the data-generating process with respect to the chosen scoring rule. This is different from Bayesian model averaging, which asymptotically with probability 1 will select a single model: the one that is closest in KL divergence to the true data-generating process.

Solving for the stacking weights is an M-estimation problem, i.e., the estimates minimizing a loss function computed on the data. Under some mild conditions, the stacking weights converge to the weights that maximizes predictive accuracy, in the sense of yielding the highest expected score. Thus, the stacking weights are the optimal combination of weights asymptotically (Yao et al., 2018).

1.4.1 BPS of predictive densities

In the following, we only consider the Logarithmic score rule, which is defined as $\text{LogS}(p, y) = \log(p(y))$ with $d(p, q) = \text{KL}(q, p)$. As presented, the Logarithmic scoring rule is the one inducing the notable Kullback-Liebler (KL) divergence, here denoted as $\text{KL}(\cdot, \cdot)$. As a matter of fact, it is a crucial scoring rule not only for its relation to the KL divergence, but also because, without further smoothness assumptions, every proper local scoring rule is equivalent to the logarithmic score. In this matter, the aforementioned Logarithmic scoring rule defines a specific class of Bayesian stacking: the stacking of predictive distributions.

Bayesian predictive stacking (BPS) of predictive distributions assimilates models using a weighted distribution in the convex hull, $C = \left\{\sum_{j=1}^{J} w_j p(\cdot \mid \mathscr{D}, \mathscr{M}_j) : \sum_j w_j = 1, w_j \geq 0\right\}$, of individual posterior distributions by maximizing the logarithm score rule (Gneiting and Raftery, 2007; Yao et al., 2018) to fetch

$$(\hat{w}_1, \ldots, \hat{w}_J)^{\mathrm{T}} = \underset{w \in S_1^J}{\arg\max} \frac{1}{n} \sum_{i=1}^{n} \log \sum_{j=1}^{J} w_j p\left(Y_i \mid \mathscr{D}_{-i}, \mathscr{M}_j\right). \tag{1.21}$$

Here, $\mathscr{D}_{-i}$ is the dataset excluding the i-th block (indexed by a row) of observations in Y, and $\mathscr{M} = \left(\mathscr{M}_1, \ldots, \mathscr{M}_J\right)$ are J different models.

Solving (1.21) minimizes the Kullback-Leibler divergence from the true predictive distribution using convex optimization (Grant and Boyd, 2008; CVX Research, 2012). In particular, the Bayesian predictive stacking problem with the logarithm score rule is the $\mathscr{M}^\star$-optimal projection of the information in the actual belief model $\mathscr{M}^\star$ to $\hat{w}$. The explicit specification of $\mathscr{M}^\star$ is avoided by re-using data as a proxy for the predictive distribution of $\mathscr{M}^\star$, and $\{w_1, \ldots, w_J\}$ are the free parameters.

As explained in Section 1.4, while the true predictive distribution is unknown, we use a leave-one-out (LOO) estimate of the expected value of the score (see. e.g., Yao et al., 2018, for details), which requires fitting the model n times as we exclude one row of the data $\mathscr{D}$ at a time. We generally prefer K-fold cross-validation in large-scale applications, as a more cost-effective method for generating predictions (Breiman, 1996).

The maximization in Equation (1.21) is a convex optimization problem (see Grant, 2005; Grant and Boyd, 2008, and reference therein) that can be solved with standard optimization libraries present in the most popular statistical computing software. In this thesis, while building the programs and the associated `R` packages `spBPS` and `spFFBS`, we make extensive use of the `CVXR` package freely available on `CRAN` for the `R` environment CVX Research (2012). It applies signed disciplined convex programming (DCP) to verify the problem's convexity, converting the problem into standard form using graph implementations and passing it into

a quadratic solver such as OSQP, or a cone solver such as ECOS or SCS (Grant, 2005; Grant and Boyd, 2008; CVX Research, 2012).

1.4.2 Dynamic BPS of predictive densities

Bayesian predictive stacking of predictive distributions, detailed in Section 1.4.1, offers an extremely flexible tool for assimilating models. However, it shows some limitations regarding the intrinsic dependence structure of data. In general, while evaluating models the BPS procedure assumes data to be i.i.d., which may be problematic when applying it to multivariate correlated variables, and related heterogeneity across models. Within Bayesian predictive synthesis, some new contributions go toward accounting for either spatial or complex dependence structures (Tallman and West, 2023; Cabel et al., 2025). Besides, Bayesian stacking is still attractive. As a matter of fact, it provides a feasible and promising solution to deliver computationally friendly posterior prediction inferences and uncertainty quantification (Zhang et al., 2024b; Presicce and Banerjee, 2024; Pan et al., 2025).

To the best of our knowledge, there is no contribution merging the flexibility of standard Bayesian predictive stacking and accounting for dependence in data. In this Section, we aim to provide a novel contribution toward this research path. Specifically, we offer an updated version of Bayesian predictive stacking that accounts for temporal dependence in data. The proposed formulation can account for Markovian dynamics, posing the first milestone in modeling complex dependences by using Bayesian predictive stacking.

As presented in Yao et al. (2018) Section 3.1, and resumed in Section 1.4, the Bayesian predictive stacking problem is defined as the optimization problem in Equation (1.16), aiming to either maximize a score rule function or minimize the induced divergence between the true predictive distribution and the posterior predictive. The empirical approximation to Equation (1.16) proposed in the original manuscript of Yao et al. (2018) relies upon the replacement of the posterior predictive density evaluated at a new data point with its corresponding leave-one-out (LOO) cross-validation predictive distribution. In doing so, they avoid over-optimistic estimates, see Equation (1.18).

Here, we aim to manage Markovian dependence by defining a renewed stacking problem, ad hoc for dynamic contexts. Specifically made for scenarios where time-dependence has relevance, and a one-step ahead posterior predictive distribution is more informative than the standard posterior predictive. Typically, in these situations, customary LOO approximations fail to provide reliable validations (Snijders, 1988; Paul-Christian Bürkner and Vehtari, 2020). Introducing temporal dynamic in the BPS problem should then imply that the proper scoring rule must be assessed with respect to the true one-step ahead predictive distribution, at a generic time instant t, i.e. $p^\star(Y_t \mid \mathscr{D}_{t-1})$, instead of posterior predictive as seen in Equation (1.16). Thus, we define the "dynamic" stacking problem as

$$\min_{w_t \in \mathcal{S}_1^J} d\left(\sum_{j=1}^{J} w_{t,j} p(\cdot \mid \mathscr{D}_{t-1}, \mathscr{M}_j),\ p^\star(\cdot \mid \mathscr{D}_{t-1})\right) \quad \text{or} \quad \max_{w_t \in \mathcal{S}_1^J} S\left(\sum_{j=1}^{J} w_{t,j} p(\cdot \mid \mathscr{D}_{t-1}, \mathscr{M}_j),\ p^\star(\cdot \mid \mathscr{D}_{t-1})\right). \tag{1.22}$$

Since also the one-step-ahead posterior predictive distribution is generally unknown, an ap-

proximation is required. In this situation, as mentioned before, leave-one-out cross-validation does not provide an accurate approximation, as it does not account for the time ordering naturally present in time series data (Snijders, 1988).

Leave-future-out (LFO) arises in literature to solve precisely this issue. We then pursue an approximation to the expectation of the proper scoring rule by using leave-future-out cross-validation. Indeed, proper scoring rules for two probabilistic forecasts can also be written as the following expectation (Gneiting and Raftery, 2007; Yao et al., 2018, for further details)

$$S\left(\sum_{j=1}^{J} w_{t,j} p(\cdot \mid \mathscr{D}_{t-1}, \mathscr{M}_j), p^{\star}(\cdot \mid \mathscr{D}_{t-1})\right) = \mathbb{E}_{p^{\star}}\left[S\left(\sum_{j=1}^{J} w_{t,j} p(Y_t \mid \mathscr{D}_{t-1}, \mathscr{M}_j), Y_t\right)\right]. \tag{1.23}$$

Implementing the leave-future-out cross-validation we derive the subsequent approximation to the expectation in (1.23), which takes into account temporal dependence with Markovian dynamics

$$(\hat{w}_{t,1}, \ldots, \hat{w}_{t,J})^{\mathrm{T}} = \underset{w_t \in \mathcal{S}_1^J}{\arg\max} \frac{1}{t-1} \sum_{\tau=1}^{t-1} S\left(\sum_{j=1}^{J} w_{t,j} p(Y_{\tau+1} \mid \mathscr{D}_{\tau}, \mathscr{M}_j), Y_{\tau+1}\right). \tag{1.24}$$

We also opt for the logarithm score, since its desirable properties as a locally proper score rule. Again solving (1.24) then identifies, at each time point t, the predictive distribution within the convex hull $C_t = \left\{\sum_{j=1}^{J} w_{t,j} p(\cdot \mid \mathscr{D}_{t-1}, \mathscr{M}_j) : \sum_j w_{t,j} = 1, w_{t,j} \geq 0\right\}$ that is closest to the data-generating process with respect to the chosen scoring rule, equivalently minimizing the Kullback-Leibler divergence from the true predictive distribution. The maximization problem decomposes into t independent problems, yielding optimal stacking weights separately at each time point.

In addition, similarly to standard Bayesian predictive stacking of predictive distribution problem, dynamic Bayesian predictive stacking can be easily executed using convex optimization (Grant, 2005; CVX Research, 2012), leading to the optimization problem reported in Equation (1.24). Likewise, we call this novel approach dynamic Bayesian predictive stacking (DYNBPS) of predictive distributions.

1.5 Bayesian transfer learning

This dissertation proposes methodologies that specifically belong to metalearning, or "learning to learn". In Section 2.1, Suder et al. (2023) state: "Vanschoren et al. (2019) defines metalearning as methods aiming to improve the configuration of the model for the target task by training on metadata; referring to information obtained from models trained individually with different configurations; for example one may vary different aspects of the model and measure its performance via cross validation", and subsequently "While some researchers consider metalearning as distinct from transfer learning Hospedales et al. (2022), following Definition 1.5.1 we consider it to be a special case of transfer learning. In this case, the information from the source domain is utilized by training models with various configurations on these domains

and then using the metadata generated from them in improving the model for the target task." As a specific case, this thesis must be considered as a contribution within the context of transfer learning.

Transfer learning (TL) broadly refers to propagating knowledge from one task to accomplish a different, but related, task (see Suder et al., 2023, and references therein). A "task" denotes a specific problem that a model is trained to solve, such as classifying images or detecting spam emails. Typically, a model is first trained on a "source" task, often with a large dataset, to learn general features. These learned features are then leveraged to improve performance on a new "target" task, often smaller or related, via fine-tuning or adaptation. This process, referred to as "knowledge transfer", accelerates learning and enhances predictive accuracy on the target task. In conventional machine learning (ML), transfer learning specifically involves re-using a trained model for a new but related task (e.g., adapting a classifier for cars to recognize trucks without retraining from scratch). The approach presented in Chapter 2, which focuses on splitting large spatial datasets and recombining posterior inference, does not trivially align with this classical usage. Following Suder et al. (2023), we formalize transfer learning as follows:

Definition 1.5.1. *Consider the source domains $\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_K$ with respective associated source tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K$, as well as the target domain $\mathscr{D}_0$ with the associated target task $\mathcal{T}_0 = \{\mathcal{Y}_0, \mathrm{f}_0\}$, where an approximation to f_0 can be learned based on the available data (X_0, Y_0) with $X_0 \in \mathcal{X}_0, Y_0 \in \mathcal{Y}_0$. Suppose that $\mathscr{D}_k \neq \mathscr{D}_0$ or $\mathcal{T}_k \neq \mathcal{T}_0$ for any $k = 1, \ldots, K$. Transfer learning refers to algorithms which aim at improving the approximation of f_0 by incorporating the knowledge from $\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_K$ and $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K$.*

This definition provides a foundation for the Bayesian transfer learning framework developed in Chapter 2, where knowledge from multiple data partitions is combined to improve inference on a target spatial process.

In Chapter 2, we consider transferring inference from one subset to the next in a stream of subsets to analyze the entire dataset. This resembles "divide and conquer" methods: we divide a computationally unfeasible problem into tractable sub-problems. Ideally, we wish to reproduce the inference where we are able to analyze the entire dataset with a desired model. However, this is achieved only in special cases (see Section 2.1). Chapter 3 offers a customary definition of transfer learning, where information is propagated forward in time.

Note that our focus remains on statistical inference with uncertainty quantification. More specifically, in Section 2.4.3, we show how predictive stacking can be used to support amortized inference within deep learning frameworks. In probabilistic modeling and Bayesian statistics, amortized inference refers to shifting the computational burden of inference to an offline training phase, during which a neural network learns a reusable mapping from observed data to posterior distributions of interest, or to their summaries. Once this training phase is completed, inference for new datasets can be obtained almost instantaneously, without rerunning computationally intensive algorithms, and yielding posterior summaries for spatial random fields at negligible additional cost.

This perspective aligns naturally with the notion of transfer learning commonly adopted in the machine learning literature, where a model trained across related tasks enables rapid inference on new ones. While amortized inference is not the primary focus of this contribution,

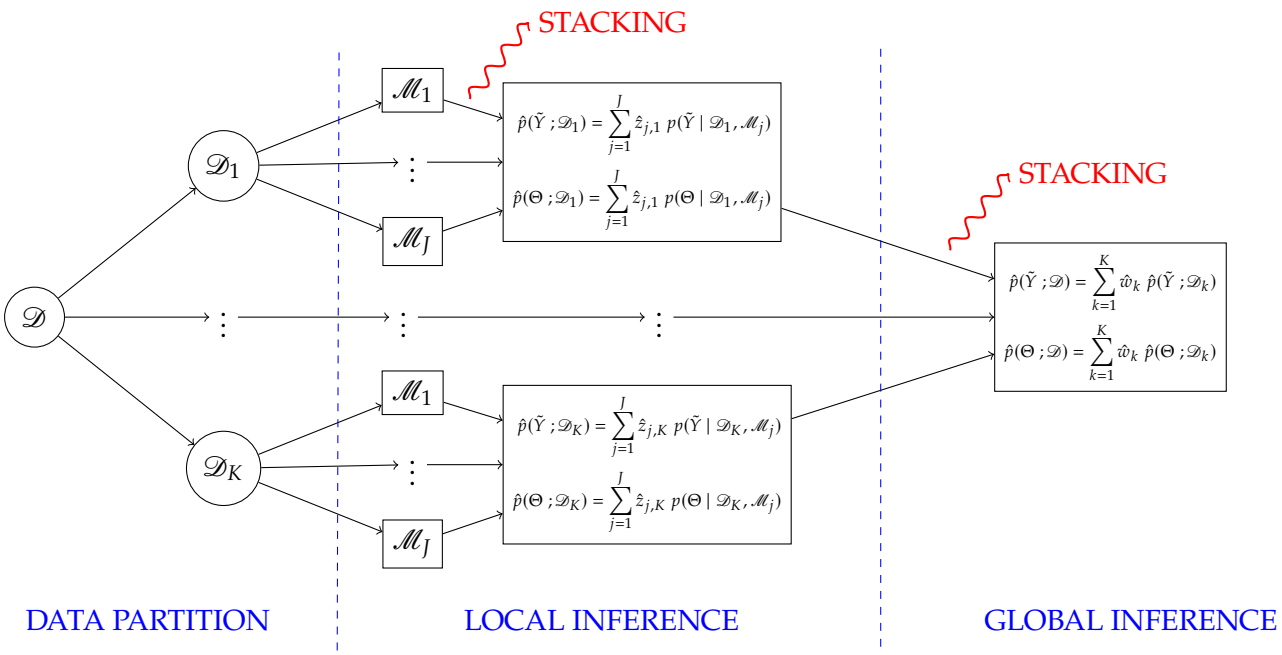

Figure 1.1: Double Bayesian predictive stacking approach representation

these results highlight the practical value of predictive stacking, which provides accurate predictive targets for network training without relying on expensive iterative sampling-based procedures.

1.6 Thesis contributions

The research presented in this dissertation takes a distribution-theoretic perspective, exploiting the structure of matrix-variate probability distributions to enable conjugate analytic updates. It builds on Bayesian predictive stacking principles to achieve coherent Bayesian transfer learning in both static and dynamic environments. The contributions unfold along two main directions, corresponding to Chapters 2 and 3, which are complementary yet unified under the same distributional foundations.

The first major contribution is the development of a Bayesian transfer learning framework designed for large-scale spatial analyses. At its core, this framework exploits conjugacy properties of matrix-Gaussian models to design a geostatistical model inference strategy that is computationally feasible even under millions of georeferenced locations. The methodology proceeds by partitioning the spatial domain and conducting local inference. Each partition is analyzed using multiple conjugate matrix-Gaussian models, yielding closed-form posterior distributions. This local step preserves exact Bayesian posterior information while remaining analytically tractable. The resulting local posteriors are then synthesized through a novel double Bayesian predictive stacking procedure: the first stacking occurs within data partitions, where information from multiple competitive models $\mathscr{M}$ is coherently assimilated; the second stacking takes place across partitions, allowing information to be transferred from local analyses to a global spatial perspective. Figure 1.1 shows a schematic representation of the DOUBLE BPS approach. As mentioned in Section 1.5 following Suder et al. (2023), this two-stage approach

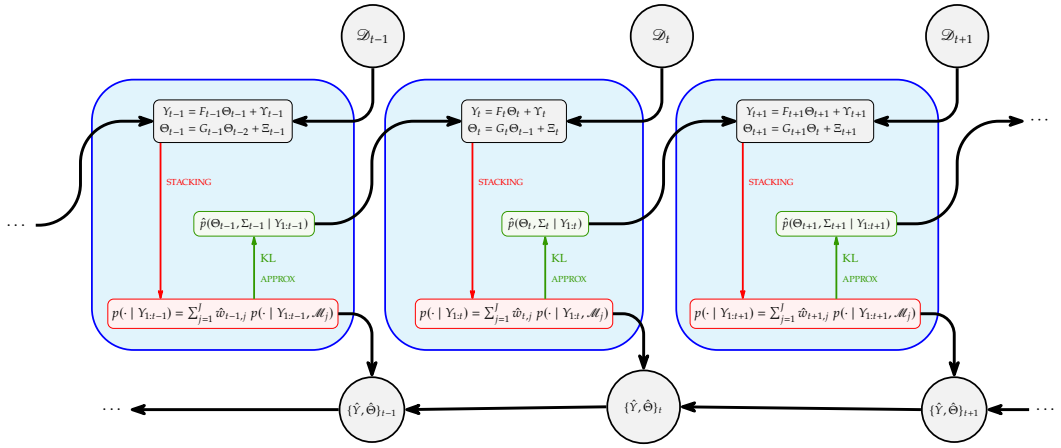

Figure 1.2: Data shards dynamics dependences representation

effectively constitutes a form of Bayesian transfer learning, as knowledge gleaned from one part of the spatial domain informs inference in other parts, and local analyses are integrated into a coherent global predictive distribution. Importantly, we achieve the information transfer entirely through analytic distributional updates, without the need for high-dimensional and expansive iterative or simulation-based algorithms, such as customary MCMC. The framework enables both scalable and feasible posterior inference and predictive uncertainty quantification for datasets that would be computationally inaccessible to conventional Gaussian process methods or hierarchical Bayesian models. This advantage is particularly evident when working with scarce computational resources. Chapter 2 develops this framework in detail, presenting both the theoretical underpinnings rooted in matrix-Gaussian distribution theory and empirical demonstrations on large-scale spatial datasets.

In the second major contribution, we extend Bayesian transfer learning by embedding it within the structure of matrix-variate dynamic linear models (DLMS), thereby adapting Bayesian predictive stacking to sequential online settings, i.e., where information arrives over time. We bring ideas from static spatial data detailed in Chapter 2 into dynamic spatiotemporal settings, through a twofold parallel direction, working with dynamic linear models. Firstly, we propose a variational adaptation to integrate dynamic spatiotemporal models with Bayesian predictive stacking. Secondly, we focus on a dynamic adaptation for the Bayesian predictive stacking methodology. Indeed, BPS of predictive distribution, as presented in Section 1.4.1, provides a mechanism for combining multiple predictive distributions into a single coherent forecast, but without accounting for dependencies in data. The resulting dynamic predictive stacking framework enables the weights used in predictive combination to evolve adaptively, responding to temporal heterogeneities in the data. Figures 1.2,1.3 offer a representation of the time data shards' dynamic dependence, and how it is controlled either by variational adaptation in Section 3.1 or parallel flows propagation in Section 3.2, respectively. Both methodologies aim to handle temporal propagation similarly: at each time step, exact posterior inference is achieved by exploiting the tractability of matrix-Gaussian and Student's t distributions, while ensuring that sequential updates can be carried out analytically and without iterative simulation. The method depicted in Section 3.2 dynamically integrates information across models and across time, producing posterior distributions and predictive densities that remain coherent

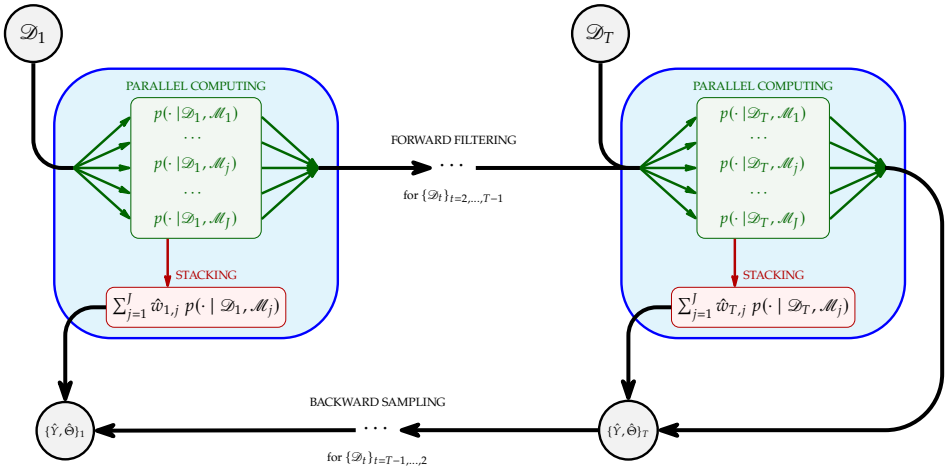

Figure 1.3: Data shards dynamics dependences representation

and calibrated throughout the sequence. Conversely, the contents in Section 3.1 introduce a variational approximation which allows for performing online learning, notwithstanding BPS induces non-conjugate mixtures of distributions. The methodologies provide an analytically tractable alternative to sequential Monte Carlo methods, which are typically required for such problems but become computationally burdensome in large-scale settings. Chapter 3 develops this contribution in detail, presenting the theoretical distributional results, the sequential and parallel updating algorithms, and applications to climate science forecasting and interpolation tasks.

Taken together, these contributions establish a coherent probabilistic framework that addresses the dual challenges of scalability and automation in modern GeoAI. By grounding inference procedures in distribution theory and minimizing reliance on iterative computation, the thesis develops robust and flexible Bayesian transfer learning tools for large geospatial and spatiotemporal systems. Central to this framework is the use of Bayesian predictive stacking as a unifying principle, which enables the principled combination of multiple probabilistic models while maintaining computational efficiency at scale. Within this perspective, customary ideas of conjugacy, extended to matrix-variate distributions, are repurposed to accommodate modern large-scale problems, yielding analytically tractable updates and scalable inference procedures. Predictive stacking then complements this structure by extending beyond static model averaging to fully dynamic and sequential settings, allowing model weights and predictive distributions to adapt over time as new information becomes available. The thesis outlines a research agenda in which distributional conjugacy and predictive stacking jointly form the foundation for next-generation Bayesian inference at scale. This perspective enables the analysis of datasets that are orders of magnitude larger and more complex than those traditionally addressed by Bayesian spatiotemporal models, while preserving interpretability, uncertainty quantification, and statistical rigor.

The remainder of the thesis is organized as follows. Chapter 2 develops the Bayesian transfer learning framework for massive spatial data, introducing partitioned conjugate inference using matrix-Gaussian models and the double-stacking procedure that enables transfer of information across partitions. Chapter 3 presents the dynamic Bayesian predictive stacking framework

for spatiotemporal models, also extending predictive stacking to sequential inference with variational approximations under matrix-variate dynamic linear models and demonstrating applications to forecasting and interpolation. Finally, Chapter 4 offers extensions, discussion, and future research directions, reflecting on the methodological implications of the work, its limitations, and opportunities for advancing scalable Bayesian inference in geospatial science and beyond.

Supporting materials for the methods, algorithms, and proofs presented in Chapter 2 are provided in Appendix A. Appendix B contains supplementary material for the dynamic predictive stacking framework of Chapter 3, including derivations, implementation details, and additional experimental results.

Chapter 2

Bayesian transfer learning for Spatial Large-Scale Problems

The first contribution devises transfer learning frameworks for deployment in artificially intelligent systems. Here, a massive dataset is split into smaller datasets that stream into the analytical framework to propagate learning and assimilate inference for the entire dataset. Specifically, we introduce Bayesian predictive stacking for multivariate spatial data and demonstrate the rapid and automated analysis of massive datasets. Furthermore, inference is delivered without either human intervention or excessively demanding hardware settings. We illustrate the effectiveness of our approach through extensive simulation experiments. We produce inference from massive datasets on sea surface temperatures and vegetation index that are indistinguishable from traditional (and more expensive) statistical approaches.

2.1 Divide-and-Conquer multivariate Bayesian inference

We consider the matrix-variate Bayesian linear regression model

$$Y = X\beta + E_Y \ , \quad E_Y \mid \Sigma \sim \text{MN}(O,\ V,\ \Sigma) \tag{2.1}$$

where Y is the $n \times q$ response matrix, X is the $n \times p$ explanatory variables matrix, and β is the $p \times q$ matrix of regression coefficients. The matrix E_Y is a zero-centered random error, with O denoting a conformable matrix of zeros, and V an $n \times n$ row-covariance matrix characterizing dependence across observations. We assign a matrix-Gaussian $\beta \mid \Sigma \sim \text{MN}(M_0 m_0,\ M_0,\ \Sigma)$, where M_0 is the $p \times p$ prior row-covariance matrix, encoding prior dependence and scaling across regression coefficients. The $p \times q$ matrix m_0 represents prior information on β. In particular, the parameterization specifying the prior mean as $M_0 m_0$ simplifies posterior updating formulas, reduces redundant matrix operations, and leads to more efficient closed-form expressions for the posterior parameters under conjugacy. The likelihood and the prior share the same $q \times q$ column-covariance matrix Σ, governing the dependence across responses. We assign an inverse-Wishart distribution $\Sigma \sim \text{IW}(\Psi_0, \nu_0)$ and denote the resulting joint prior distribution of β and Σ by $\text{MNIW}(M_0 m_0, M_0, \Psi_0, \nu_0)$, which yields a closed-form posterior for $\{\beta, \Sigma\}$ within the same family.

Let $\mathscr{D} = \{Y, X\}$ be the entire dataset, which is too massive to even be accessed, let alone be analyzed using (1.1), within the GeoAI system. Therefore, we envisage K disjoint and exhaustive subsets $\mathscr{D} = \{\mathscr{D}_1, \ldots, \mathscr{D}_K\}$ streaming into GeoAI as a sequence. Each $\mathscr{D}_k = \{Y_k, X_k\}$ consists of n_k rows of Y and X, where $n = \sum_{k=1}^{K} n_k$, Y_k is $n_k \times q$ and X_k is $n_k \times p$ for each $k = 1, \ldots, K$. We now fit (2.1) to each subset using $Y_k = X_k\beta + E_k$, $E_k \sim \text{MN}(O, V_k, \Sigma)$, where V_k is the $n_k \times n_k$ row-covariance matrix corresponding to the rows in Y_k; the specification for $\{\beta, \Sigma\}$ remains as in (2.1). Starting with $\text{MNIW}(\beta, \Sigma \mid M_k m_k, M_k, \Psi_k, \nu_k)$ at $k = 0$ (the prior), we use Bayesian updating $p(\beta, \Sigma \mid \mathscr{D}_{1:k+1}) \propto p(\beta, \Sigma \mid \mathscr{D}_{1:k}) \times p(Y_{k+1} \mid X_{k+1}\beta, V_{k+1}, \Sigma)$ to obtain $\beta, \Sigma \mid \mathscr{D}_{1:k+1} \sim \text{MNIW}(M_{k+1}m_{k+1}, M_{k+1}, \Psi_{k+1}, \nu_{k+1})$ with $M_{k+1}^{-1} = M_k^{-1} + X_{k+1}^{\top} V_{k+1}^{-1} X_{k+1}$, $m_{k+1} = m_k + X_{k+1}^{\top} V_{k+1}^{-1} Y_{k+1}$, $\nu_{k+1} = \nu_k + n_{k+1}$, and $\Psi_{k+1} = \Psi_k + Y_{k+1}^{\top} V_{k+1}^{-1} Y_{k+1} + m_k^{\top} M_k m_k - m_{k+1}^{\top} M_{k+1} m_{k+1}$. Upon termination at $k = K$, we exactly recover the posterior distribution $p(\beta, \Sigma \mid \mathscr{D})$. There is no need to interact between the subsets, and computational complexity is determined solely by the dimension of the subsets.

Spatial random fields immediately present a challenge. The above method delivers full inference without loss of information only if Y_k's are independent (or exchangeable) across blocks. If each row of Y corresponds to one of n spatial locations so that V is an $n \times n$ spatial correlation matrix, then each V_k is the spatial correlation matrix constructed from n_k spatial locations in $\mathscr{D}_k$. Independence among the K blocks may yield reasonable inference if we can design the blocks such that the spatial correlation across blocks do not impact overall inference. However, designing such blocks will require significant human intervention or a separate AI system. We seek to avoid this in GeoAI.

Instead, we devise a method for assimilating the learning from each of these blocks using predictive stacking. We exploit the fact that V is indexed by a small number of parameters in a spatial correlation kernel. Fixing these parameters fixes V, and hence V_k for each k, yielding closed-form posterior inference on β and Σ. Stacking combines these analytically accessible distributions using an optimal set of weights that are computed using a convex optimization algorithm. These weights are then used to reconstruct the posterior and predictive distributions for the spatial random field. It is worth remarking here that existing "meta-kriging" and related "divide and conquer" approaches (e.g., Guhaniyogi and Banerjee, 2018; Scott et al., 2016, and other references on "divide and conquer" methods provided in Section 1) analyze subsets of data using MCMC, which is expensive and not fully automated. A key distinction of the current manuscript is that we abandon all iterative estimation algorithms, let alone expensive MCMC, and focus on assimilating inference using closed forms of distributions.

What remains to be resolved is the issue of fixing the parameters in V. These parameters govern the strength of association across the spatial random field and possibly the smoothness of the field. Unfortunately, these parameters are weakly identified by the data, and posterior learning struggles due to slower convergence of iterative algorithms. Exploratory spatial data science tools for gleaning information about these parameters will also be less helpful, as they may suggest different values for each of the variables indexed by the columns, while we insist on retaining a single parameter to exploit conjugate distribution theory. Hence, we collect the closed-form posterior distributions obtained for a collection of fixed values of the spatial parameters and, subsequently, combine these posterior distributions.

2.2 Accelerated learning for multivariate spatial random fields

Let $\mathcal{S} = \{s_1, \ldots, s_n\} \subset \mathcal{D}$ be a set of n locations yielding observations on q possibly correlated outcomes collected into a $q \times 1$ vector $y(s) = (y_1(s), \ldots, y_q(s))^{\mathrm{T}}$ for each $s \in \mathcal{S}$. We collect these measurements into the $n \times q$ matrix $Y = \left[\, y_j(s_i) \,\right]$ for $i = 1, \ldots, n$ and $j = 1, \ldots, q$. Let $X = [x(s_i)^{\mathrm{T}}]$ be $n \times p$ with rows $x(s_i)^{\mathrm{T}}$ consisting of $p < n$ explanatory variables at location $s_i \in \mathcal{S}$; we assume X has rank p. We introduce latent spatial processes, $\omega_j(s)$, for each outcome $y_j(s)$ to capture spatial dependence and a $q \times q$ covariance matrix, Σ, to capture non-spatial dependence among the elements of $y(s)$ within s. This matrix is typically adjusted by a scale factor $\left(\alpha^{-1} - 1\right)$ to accommodate additional variation at local scales. For example, setting $\alpha = \sigma^2/\left(\sigma^2 + \tau^2\right)$ where σ^2 and τ^2 denote variances for the spatial process and measurement error ("nugget") implies α is the ratio of the spatial variance (partial sill) to the total variance (sill) gleaned from a variogram in classical geostatistics.

We cast this into (2.1), explicitly introduce a latent $q \times 1$ spatial process $\omega(s)$ as

$$\begin{aligned} Y \mid \beta, \Omega, \Sigma &\sim \mathrm{MN}(X\beta + \Omega,\ (\alpha^{-1} - 1)\mathbb{I}_n,\ \Sigma) \\ \beta \mid \Sigma &\sim \mathrm{MN}(M_0 m_0,\ M_0,\ \Sigma) \\ \Omega \mid \Sigma &\sim \mathrm{MN}(O,\ V,\ \Sigma) \\ \Sigma &\sim \mathrm{IW}(\Psi_0,\ \nu_0) \end{aligned} \tag{2.2}$$

Where $\Omega = [\omega(s_i)^{\mathrm{T}}]$ is $n \times q$ with rows $\omega(s_i)^{\mathrm{T}}$. To capture spatial dependence, V is an $n \times n$ spatial correlation matrix with (i, j)-th element equaling the value of a positive definite spatial correlation function $\rho(s_i, s_j; \phi)$ indexed by parameter(s) ϕ. To account for measurement errors in observations, as is customary in geostatistics, we introduce a discontinuity in the spatial correlation function and modify the elements to $\rho(s_i, s_j; \phi) + (\alpha^{-1} - 1)\mathbb{1}_{s_i = s_j}$, where $\alpha \in [0, 1]$ represents the proportion of total variability attributed to the spatial process.

Let $\gamma^{\mathrm{T}} = \left[\, \beta^{\mathrm{T}}\ \Omega^{\mathrm{T}} \,\right]$ be $q \times (p + n)$, we assume $\{\gamma, \Sigma\} \sim \mathrm{MNIW}\left(\mu_\gamma, V_\gamma, \Psi_0, \nu_0\right)$, where

$$\mathrm{MNIW}\left(\gamma, \Sigma \mid \mu_\gamma, V_\gamma, \Psi_0, \nu_0\right) = \mathrm{IW}(\Sigma \mid \Psi_0, \nu_0) \times \mathrm{MN}_{p,q}(\gamma \mid \mu_\gamma V_\gamma, \Sigma), \tag{2.3}$$

with $\mu_\gamma^{\mathrm{T}} = \left[m_0^{\mathrm{T}} M_0 \ \ 0_{q \times n}\right]$ and $V_\gamma = \text{blockdiag}\left\{M_0, \rho_\phi(\mathcal{S}, \mathcal{S})\right\}$. The MNIW prior is conjugate with respect to the matrix-normal likelihood. Thus, for any fixed $\{\alpha, \phi\}$ and hyperparameters in the prior density, we obtain a MNIW posterior density for $\{\gamma, \Sigma\}$,

$$p(\gamma, \Sigma \mid \mathscr{D}) = \mathrm{MNIW}\left(\gamma, \Sigma \mid \mu_\gamma^\star, V_\gamma^\star, \Psi^\star, \nu^\star\right), \tag{2.4}$$

where $\mu_\gamma^\star = V_\gamma^\star \begin{bmatrix} \frac{\alpha}{1-\alpha} X^{\mathrm{T}} Y + m_0 \\ \frac{\alpha}{1-\alpha} Y \end{bmatrix}$, and $V_\gamma^\star = \begin{bmatrix} \frac{\alpha}{1-\alpha} X^{\mathrm{T}} X + M_0^{-1} & \frac{\alpha}{1-\alpha} X^{\mathrm{T}} \\ \frac{\alpha}{1-\alpha} X & \rho_\phi^{-1}(\mathcal{S}, \mathcal{S}) + \frac{\alpha}{1-\alpha} \mathbb{I}_n \end{bmatrix}^{-1}$, $\Psi^\star = \Psi_0 + \frac{\alpha}{1-\alpha} Y^{\mathrm{T}} Y + m_0^{\mathrm{T}} M_0 m_0 - \mu_\gamma^{*\mathrm{T}} V_\gamma^{\star -1} \mu_\gamma^\star$ and $\nu^\star = \nu_0 + n$.

The framework in (2.2) is equivalent to (2.1) with $V = R_\phi + (\alpha^{-1} - 1)\mathbb{I}_n$ with $R_\phi = [\rho(s_i, s_j; \phi)]$. We recover posterior samples of Ω by drawing a value of Ω from $p(\Omega \mid \mathscr{D}, \beta, \Sigma, \mathscr{M}_j)$ for every posterior draw of $\{\beta, \Sigma\}$. This renders itself seamlessly to the Bayesian transfer learning framework described in Section 2.1 provided that V, or $\{\alpha, \phi\}$, is fixed. For GeoAI, we seek

to minimize human intervention. Rather than fixing them at one particular value, perhaps gleaned from a spatial variogram that requires human inspection, we use a set of J candidate values $\{\alpha_j, \phi_j\}$ specifying model $\mathscr{M}_j$ for $j = 1, \ldots, J$. We now obtain analytical closed forms for $p(\beta, \Sigma \mid \mathscr{D}, \mathscr{M}_j)$ for each j, as described in Section 2.1, and use Bayesian predictive stacking to evaluate the stacked posterior distribution.

Turning to prediction, let $\mathcal{U} = \{u_1, \ldots, u_{n'}\}$ be a finite set of locations where we seek to predict or impute the value of Y based upon an observed $n' \times p$ design matrix $X_{\mathcal{U}}$ associated with the locations in $\mathcal{U}$. The joint posterior predictive for $Y_{\mathcal{U}}$ and the unobserved latent process $\Omega_{\mathcal{U}} = [\omega(u_i)^{\mathrm{T}}]$ for $i = 1, \ldots, n'$, can be recast by integrating out $\{\gamma, \Sigma\}$ from the conditional posterior predictive distribution to yield

$$\begin{aligned} p(Y_{\mathcal{U}}, \Omega_{\mathcal{U}} \mid \mathscr{D}) = \int & \mathrm{MN}_{n',q}\left(Y_{\mathcal{U}} \mid X_{\mathcal{U}}\beta + \Omega_{\mathcal{U}}, \left(\alpha^{-1} - 1\right)\mathbb{I}_{n'}, \Sigma\right) \\ & \times \mathrm{MN}_{n',q}\left(\Omega_{\mathcal{U}} \mid M_{\mathcal{U}}\Omega, V_{\Omega_{\mathcal{U}}}, \Sigma\right) \times \mathrm{MNIW}\left(\gamma, \Sigma \mid \mu^{\star}_{\gamma}, V^{\star}_{\gamma}, \Psi^{\star}, \nu^{\star}\right) d\gamma d\Sigma, \end{aligned} \tag{2.5}$$

where $M_{\mathcal{U}} = \rho_{\phi}(\mathcal{U}, \mathcal{S})\rho^{-1}_{\phi}(\mathcal{S}, \mathcal{S})$ and $V_{\Omega_{\mathcal{U}}} = \rho_{\phi}(\mathcal{U}, \mathcal{U}) - \rho_{\phi}(\mathcal{U}, \mathcal{S})\rho^{-1}_{\phi}(\mathcal{S}, \mathcal{S})\rho_{\phi}(\mathcal{S}, \mathcal{U})$. This is a matrix-variate Student's t $\mathrm{T}_{2n',q}(\nu^{\star}, \mu^{\star}, V^{\star}, \Psi^{\star})$, with degrees of freedom $\nu^{\star}$, location matrix $\mu^{\star} = M\mu^{\star}_{\gamma}$, row-scale matrix $V^{\star}$, and column-scale matrix $\Psi^{\star}$, where $M = \begin{bmatrix} 0 & M_{\mathcal{U}} \\ X_{\mathcal{U}} & M_{\mathcal{U}} \end{bmatrix}$ and $V^{\star} = MV^{\star}_{\gamma}M^{\mathrm{T}} + V_E$ with $V_E = \begin{bmatrix} V_{\Omega_{\mathcal{U}}} & V_{\Omega_{\mathcal{U}}} \\ V_{\Omega_{\mathcal{U}}} & V_{\Omega_{\mathcal{U}}} + (\alpha^{-1} - 1)\mathbb{I}_{n'} \end{bmatrix}$. See Section A.1.1 for details. The conditional posterior predictive distributions take the following forms $p(\Omega_{\mathcal{U}} \mid \mathscr{D}, \gamma, \Sigma) = \mathrm{MN}_{n',q}\left(\Omega_{\mathcal{U}} \mid M_{\mathcal{U}}\Omega, V_{\Omega_{\mathcal{U}}}, \Sigma\right)$, and $p(Y_{\mathcal{U}} \mid \mathscr{D}, \Omega_{\mathcal{U}}, \gamma, \Sigma) = \mathrm{MN}_{n',q}(Y_{\mathcal{U}} \mid X_{\mathcal{U}}\beta + \Omega_{\mathcal{U}}, \left(\alpha^{-1} - 1\right)\mathbb{I}_{n'}, \Sigma)$. Hence, we can proceed with posterior predictive inference by sampling from the closed-form joint predictive distribution or sampling from the conditional distributions. We draw one instance of $\Omega_{\mathcal{U}} \sim p(\Omega_{\mathcal{U}} \mid \mathscr{D}, \gamma, \Sigma)$ for each posterior draw of $\{\gamma, \Sigma\}$ and then draw one value of $Y_{\mathcal{U}} \sim p(Y_{\mathcal{U}} \mid \mathscr{D}, \Omega_{\mathcal{U}}, \gamma, \Sigma)$ for each drawn $\{\Omega_{\mathcal{U}}, \gamma, \Sigma\}$. The resulting samples are exactly drawn from the posterior predictive distribution $p(Y_{\mathcal{U}} \mid \mathscr{D})$.

This tractability is possible if the spatial correlation parameter(s) ϕ and α are fixed. While data can inform about these parameters, they are inconsistently estimable and lead to poor convergence (Zhang, 2004). Finley et al. (2019) explored K-fold cross-validation, but inference is limited to only one set of values for the parameters. Instead, we pursue exact inference using (2.4) and (2.5) by stacking over different fixed values of $\{\alpha, \phi\}$ using BPS of predictive densities as described in Section 1.4.1. This drastically reduces human intervention and enables automation.

For each subset of the data, we compute the stacking weights $\hat{z}_k = \{z_{k,j}\}_{j=1,\ldots,J}$ as

$$\max_{z_k \in \mathcal{S}^J_1} \frac{1}{n_k} \sum_{i=1}^{n_k} \log \sum_{j=1}^{J} z_{k,j} p\left(Y_{k,i} \mid \mathscr{D}_{k,[-l]}, \mathscr{M}_j\right), \tag{2.6}$$

where $Y_{k,i}$ denotes the i-th of the n_k rows of $Y_{k,[l]} \in \mathscr{D}_{k,[l]}$, the l-th fold within the k-th dataset. The training set $\mathscr{D}_{k,[-l]}$ is obtained by removing the l-th fold from the k-th dataset, with $l = 1, \ldots, L$, and L denoting the number of folds used to form K-fold cross-validation estimates of the expected score (see Section 1.4.1). The posterior predictive density, $p\left(Y_{k,i} \mid \mathscr{D}_{k,[-l]}, \mathscr{M}_j\right)$,

is available in closed form as a matrix t distribution, which enables efficient computation. As a result, Equation (2.6) can be written equivalently as

$$\max_{z_k \in \mathcal{S}_1^J} \frac{1}{n_k} \sum_{i=1}^{n_k} \log \sum_{j=1}^{J} z_{k,j}\, \mathrm{T}_{1,q}(Y_{k,i} \mid \nu^{\star}_{[-l]}, \mu^{\star}_i, V^{\star}_i, \Psi^{\star}_{[-l]}), \tag{2.7}$$

where $\nu^{\star}_{[-l]} = \nu_0 + n_{k,[-l]}$, $n_{k,[-l]}$ is the cardinality of $\mathcal{S}_{k,[-l]}$ (which is the set of locations in $\mathscr{D}_{k,[-l]}$), and $\Psi^{\star}_{[-l]} = \Psi_0 + (\alpha_j^{-1} - 1) Y^{\mathrm{T}}_{k,[-l]} Y_{k,[-l]} + m_0^{\mathrm{T}} M_0 m_0 - \mu^{\mathrm{T}\star}_{\gamma,[-l]} V^{-1\star}_{[-l]} \mu^{\star}_{\gamma,[-l]}$. The remaining parameters are given by $V^{\star}_i = M_{y,i} V^{\star}_{\gamma} M^{\mathrm{T}}_{y,i} + V_{\Omega_i} + (\alpha_j^{-1} - 1)$, $\mu^{\star}_i = M_{y,i} \mu^{\star}_{\gamma,[-l]}$, which are defined by computing the following auxiliary quantities: $V^{-1\star}_{\gamma,[-l]} = \begin{bmatrix} \frac{\alpha_j}{1-\alpha_j} X^{\mathrm{T}}_{k,[-l]} X_{k,[-l]} + M_0^{-1} & \frac{\alpha_j}{1-\alpha_j} X^{\mathrm{T}}_{k,[-l]} \\ \frac{\alpha_j}{1-\alpha_j} X_{k,[-l]} & \rho^{-1}_{\phi_j}(\mathcal{S}_{k,[-l]}, \mathcal{S}_{k,[-l]}) + \frac{\alpha_j}{1-\alpha_j} \mathbb{I}_{n_{k,[-l]}} \end{bmatrix}$, $\mu^{\star}_{\gamma,[-l]} = V^{\star}_{\gamma,[-l]} \begin{bmatrix} \frac{\alpha_j}{1-\alpha_j} X^{\mathrm{T}}_{k,[-l]} Y_{k,[-l]} + m_0 \\ \frac{\alpha_j}{1-\alpha_j} Y_{k,[-l]} \end{bmatrix}$, $M_{y,i} = \begin{bmatrix} X_{k,i} & \rho_{\phi_j}(\mathcal{S}_{k,i}, \mathcal{S}_{k,[-l]}) \rho^{-1}_{\phi_j}(\mathcal{S}_{k,[-l]}, \mathcal{S}_{k,[-l]}) \end{bmatrix}$, and $V_{\Omega_i} = \rho_{\phi_j}(\mathcal{S}_{k,i}, \mathcal{S}_{k,i}) - \rho_{\phi_j}(\mathcal{S}_{k,i}, \mathcal{S}_{k,[-l]}) \rho^{-1}_{\phi_j}(\mathcal{S}_{k,[-l]}, \mathcal{S}_{k,[-l]}) \rho_{\phi_j}(\mathcal{S}_{k,[-l]}, \mathcal{S}_{k,i})$. Note that $\nu^{\star}_{[-l]}$ is scalar, $\mu^{\star}_i$ is $1 \times q$ row vector, $V^{\star}_i$ is a scalar, and $\Psi^{\star}_{[-l]}$ is $q \times q$ matrix. Further details, including derivations and implementation, are provided in Section A.1.1 and Algorithm 1.

For each dataset $\mathscr{D}_k$, BPS produces: (i) an estimate of the posterior predictive $\hat{p}(\,\cdot\,; \mathscr{D}_k) = \sum_{j=1}^{J} \hat{z}_{k,j}\; p\left(\cdot \mid \mathscr{D}_k, \mathscr{M}_j\right)$ for $k = 1, \ldots, K$ and $j = 1, \ldots, J$; and (ii) the corresponding set of stacking weights $\hat{z}_k = \{\hat{z}_{k,j}\}_{j=1,\ldots,J}$.

Although the models considered here admit closed-form posterior and predictive distributions, directly applying a single stacking procedure to the full dataset (e.g. Zhang et al., 2024b) may still be computationally prohibitive in large-scale settings. The computational bottleneck arises from a combination of memory constraints associated with large covariance matrices, repeated matrix inversions required to evaluate predictive densities, and the optimization problem defining stacking weights when both the number of observations and candidate models are large. To address these challenges, we adopt a divide-and-conquer strategy based on data partitioning, which motivates the introduction of a double Bayesian predictive stacking (DBPS) procedure.

Once the subset predictive distributions and the associated weights are available, we then apply BPS a second time to combine the $\hat{p}(\,\cdot\,; \mathscr{D}_k)$ over the k subsets. This second stacking step yields a global combination of locally stacked predictive distributions. Specifically, the DOUBLE BPS (DBPS) of predictive densities seeks weights $w = \{w_k\}_{k=1,\ldots,K}$ such that $\hat{w} = \max_{w \in \mathcal{S}_1^K} \frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k \hat{p}\,(Y_i \mid \mathscr{D}_k)$. Each $\hat{p}(Y_i \mid \mathscr{D}_k)$ is evaluated using internally validated predictive distributions. Although $Y_i \in \mathscr{D}_k$, the internal cross-validation structure underlying the construction of $\hat{p}(\,\cdot\,; \mathscr{D}_k)$ mitigates over-optimism (see Section 2.3.1).

The stacked predictive distribution based on double stacking is therefore given by

$$\hat{p}(\,\cdot\,; \mathscr{D}) = \sum_{k=1}^{K} \hat{w}_k \sum_{j=1}^{J} \hat{z}_{k,j}\; p\left(\cdot \mid \mathscr{D}_k, \mathscr{M}_j\right). \tag{2.8}$$

Once double stacking weights $\hat{w} = \{\hat{w}_k\}_{k=1,\ldots,K}$ are computed using Algorithm 2, the resulting approximation defines a mixture of finite mixtures. This structure enables straightforward

sampling from (2.8) and the evaluation of both posterior and posterior predictive distributions, with a substantial simplification compared to other distributed approaches (see e.g., Guhaniyogi and Banerjee, 2018, 2019). The DBPS does not require empirical approximations of posterior or predictive distributions.

In particular, posterior sampling follows directly from the mixture representation. First, the set of stacking weights $\hat{z}_k = \{\hat{z}_{k,j}\}_{j=1,\dots,J}$ obtained using BPS within each subset of the data $\mathscr{D}_k$, is used to approximate the corresponding subset posterior distribution as

$$\hat{p}(\Theta\,;\mathscr{D}_k) = \sum_{j=1}^{J} \hat{z}_{k,j} p\left(\Theta \mid \mathscr{D}_k, \mathscr{M}_j\right), \tag{2.9}$$

for $k = 1, \dots, K$. By introducing the second set of stacking weights $\hat{w} = \{\hat{w_k}\}_{k=1,\dots,K}$, the stacked posterior distribution based on the full dataset is

$$\hat{p}(\Theta\,;\mathscr{D}) = \sum_{k=1}^{K} \hat{w}_k\ \hat{p}(\Theta\,;\mathscr{D}_k) = \sum_{k=1}^{K} \hat{w}_k\ \sum_{j=1}^{J} \hat{z}_{k,j}\ p(\Theta \mid \mathscr{D}_k, \mathscr{M}_j), \tag{2.10}$$

where $\Theta = \{\gamma, \Sigma\}$. The same construction applies to posterior predictive distributions. The predictive distribution of the random variable $Y_{\mathcal{U}}$ is recovered from (2.8) as

$$\hat{p}(Y_{\mathcal{U}}\,;\mathscr{D}) = \sum_{k=1}^{K} \hat{w}_k\ \sum_{j=1}^{J} \hat{z}_{k,j}\ p\left(Y_{\mathcal{U}} \mid \mathscr{D}_k, \mathscr{M}_j\right). \tag{2.11}$$

Inferential interest often resides on the posterior predictive surface for the latent spatial process Ω. Accordingly, we estimate $\Omega_{\mathcal{U}}$ via

$$\hat{p}(\Omega_{\mathcal{U}}\,;\mathscr{D}) = \sum_{k=1}^{K} \hat{w}_k\ \hat{p}(\Omega_{\mathcal{U}}\,;\mathscr{D}_k), \tag{2.12}$$

where $\hat{p}(\Omega_{\mathcal{U}}\,;\mathscr{D}_k) = \sum_{j=1}^{J} \hat{z}_{k,j}\ p\left(\Omega_{\mathcal{U}} \mid \mathscr{D}_k, \mathscr{M}_j\right)$.

2.3 Computational perspectives

Key computational aspects of the proposed method involve two points: comparing its theoretical complexity to state-of-the-art approaches and addressing memory constraints. The complexity comparison evaluates the method's time and space efficiency, particularly its scalability with larger datasets. This includes discernment from the explicit objective function, highlighting its computational impact. Memory constraints are equally critical, as limitations can hinder performance despite powerful processors. The proposed method addresses these challenges, ensuring both scalability and efficient resource use.

In summary, this section will examine both the theoretical complexity, including the explicit derivation of the objective function for the optimization problems detailed in Equations (2.6) and (2.13), as well as the memory management strategies, offering a comprehensive view of

Algorithm 1 Computing stacking weights within subsets using BPS

Input: Y ($n \times q$ matrix of outcomes), X ($n \times p$ design matrix), $\mathcal{S}$ (coordinates of n locations); $\{m_0, M_0, \Psi_0, \nu_0\}$: Prior parameters; $G_\alpha \times G_\phi$: Grids of $\{\alpha, \phi\}$; n (no. of locations), q (no. of outcomes), p (no. of predictors); K (no. of subsets), J (no. of models), L (no. of folds).
Output: $\hat{z} = \{\hat{z}_k = \{\hat{z}_{k,j}\} : k = 1, \ldots, K, j = 1, \ldots, J\}$: Stacking weights within subsets; $\{pd_{k,j,i} : k = 1, \ldots, K, j = 1, \ldots, J, i = 1, \ldots, n\}$: point-wise predictive density of Y; G_{all}: Grid of dimension J, spanned by G_α, G_ϕ

Partition $Y, X, \mathcal{S}$ into $\mathscr{D}_k = \{Y_k, X_k, \mathcal{S}_k\}$, $k = 1, \ldots, K$
Store n_k, as cardinality of $\mathcal{S}_k$; Compute G_{all} by expanding G_α, G_ϕ
for $k = 1, \ldots, K$ **do Parallel**
  for $j = 1, \ldots, J$ **do**
    Extract $\{\alpha_j, \phi_j\}$ from j-th row of G_{all}
    Form L folds: $\mathscr{D}_{k,[l]} = \{Y_{k,[l]}, X_{k,[l]}, \mathcal{S}_{k,[l]}\}$ and $\mathscr{D}_{k,[-l]} = \{Y_{k,[-l]}, X_{k,[-l]}, \mathcal{S}_{k,[-l]}\}$
    Store $n_{k,[-l]}$, as cardinality of $\mathcal{S}_{k,[-l]}$
    for $l = 1, \ldots, L$ **do**
      Compute $R_{\phi_j}([-l]) = \rho_{\phi_j}(\mathcal{S}_{k,[-l]}, \mathcal{S}_{k,[-l]})$, $R^{-1}_{\phi_j}([-l])$ and M_0 for M_0^{-1}
      Construct $V^{-1\star}_{\gamma,[-l]} = \begin{bmatrix} \frac{\alpha_j}{1-\alpha_j} X^{\mathrm{T}}_{k,[-l]} X_{k,[-l]} + M_0^{-1} & \frac{\alpha_j}{1-\alpha_j} X^{\mathrm{T}}_{k,[-l]} \\ \frac{\alpha_j}{1-\alpha_j} X_{k,[-l]} & R^{-1}_{\phi_j}([-l]) + \frac{\alpha_j}{1-\alpha_j} \mathbb{I}_{n_{k,[-l]}} \end{bmatrix}$
      Solve for $\mu^\star_{\gamma,[-l]}$: $V^{-1\star}_{\gamma,[-l]} \mu^\star_{\gamma,[-l]} = \begin{bmatrix} \frac{\alpha_j}{1-\alpha_j} X^{\mathrm{T}}_{k,[-l]} Y_{k,[-l]} + m_0 \\ \frac{\alpha_j}{1-\alpha_j} Y_{k,[-l]} \end{bmatrix}$
      Calculate $\Psi^\star_{[-l]} = \Psi_0 + (\alpha_j^{-1} - 1) Y^{\mathrm{T}}_{k,[-l]} Y_{k,[-l]} + m_0^{\mathrm{T}} M_0 m_0 - \mu^{\mathrm{T}\star}_{\gamma,[-l]} V^{-1\star}_{[-l]} \mu^\star_{\gamma,[-l]}$
      Calculate $\nu^\star_{[-l]} = \nu_0 + n_{k,[-l]}$
      for $i \in [l]$ **do**
        Compute $R_{\phi_j}(i) = \rho_{\phi_j}(\mathcal{S}_{k,i}, \mathcal{S}_{k,i})$ and $R_{\phi_j}(i, [-l]) = \rho_{\phi_j}(\mathcal{S}_{k,i}, \mathcal{S}_{k,[-l]})$
        Calculate $M_i = R_{\phi_j}(i, [-l]) R^{-1}_{\phi_j}([-l])$ and form $M_{y,i} = \begin{bmatrix} X_{k,i} & M_i \end{bmatrix}$
        Calculate $\mu^\star_i = M_{y,i} \mu^\star_{\gamma,[-l]}$ and $V_{\Omega_i} = R_{\phi_j}(i) - M_i R_{\phi_j}([-l], i)$
        Construct $V_{e,i} = V_{\Omega_i} + (\alpha_j^{-1} - 1)$ and $V^\star_i = M_{y,i} V^\star_{\gamma,[-l]} M^{\mathrm{T}}_{y,i} + V_{e,i}$
        Compute $pd_{k,j,i} = \mathrm{T}_{1,q}(Y_{k,i} \mid \nu^\star_{[-l]}, \mu^\star_i, V^\star_i, \Psi^\star_{[-l]})$.
      end for
    end for
  end for
  Solve: $\max_{z_k \in \mathcal{S}_1^J} \frac{1}{n_k} \sum_{i=1}^{n_k} \log \sum_{j=1}^{J} z_{k,j} pd_{k,j,i}$ such that $z_k \in [0,1]^J$ and $\sum_{j=1}^{J} z_{k,j} = 1$
end Parallel for
return $\{\hat{z}, \{pd_{k,j,i}\}, G_{all}\}$

the computational feasibility in practical applications.

2.3.1 Objective function for double Bayesian predictive stacking

We expand the derivation of double Bayesian predictive stacking of Section 2.2. The optimization problem used to compute the stacking weights in Equation (2.8) is formally defined as:

$$\max_{w \in \mathcal{S}_1^K} \frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k \hat{p}\left(Y_i \,; \mathscr{D}_k\right) = \max_{w \in \mathcal{S}_1^K} \frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k \sum_{j=1}^{J} \hat{z}_{k,j} p\left(Y_{k,i} \mid \mathscr{D}_{k,[-l]}, \mathscr{M}_j\right), \tag{2.13}$$

Algorithm 2 Calculating stacking weights between subsets using BPS

Input: $\hat{z} = \{\hat{z}_k = \{\hat{z}_{k,j}\} : k \in \{1,\ldots,K\}, j \in \{1,\ldots,J\}\}$: Stacking weights within subsets; $\{pd_{k,j,i} : k = 1,\ldots,K, j = 1,\ldots,J, i = 1,\ldots,n\}$: point-wise predictive density of Y; n, q, p: Number of rows, number of outcomes, and number of predictors; $K, \{n_k : k \in \{1,\ldots,K\}\}, J$: Number of subsets, dimension of each subset, and number of competitive models in each subset.

Output: $\hat{w} = \{\hat{w}_k : k = 1,\ldots,K\}$: Stacking weights between subsets.

1: Construct $pd = [pd_1^{\mathrm{T}} : \cdots : pd_K^{\mathrm{T}}]^{\mathrm{T}}$ of dimension $(n \times J)$

where $pd_k = \begin{bmatrix} pd_{k,1,1} & \ldots & pd_{k,J,1} \\ \vdots & pd_{k,j,i} & \vdots \\ pd_{k,1,n_k} & \ldots & pd_{k,J,n_k} \end{bmatrix}$ of dimension $(n_k \times J)$

2: **for** $k = 1,\ldots,K$ **do**

3: Compute $epd_k = pd\ \hat{z}_k$ of dimension $(n \times 1)$

4: **end for**

5: Solve convex optimization problem:

$\max_{w \in \mathcal{S}_1^K} \frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k epd_{k,i} = \max_{w \in \mathcal{S}_1^K} \frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k \sum_{j=1}^{J} \hat{z}_{k,j} pd_{k,j,i}$

where $pd_{k,j,i} = p\left(Y_{k,i} \mid \mathscr{D}_{k,[-l]}, \mathscr{M}_j\right) = \mathrm{T}_{1,q}(Y_{k,i} \mid \nu^{\star}_{[-l]}, \mu_i^{\star}, V_i^{\star}, \Psi^{\star}_{[-l]})$

for $\forall i \notin [-l], l \in \{1,\ldots,L\}$ and $\mathcal{S}_1^K = \{w \in [0,1]^K : \sum_{k=1}^{K} w_k = 1\}$

6: **return** $\hat{w} = \{\hat{w}_k : k \in \{1,\ldots,K\}\}$

as $\hat{p}\left(Y_i\ ; \mathscr{D}_k\right) = \sum_{j=1}^{J} \hat{z}_{k,j} p\left(Y_{k,i} \mid \mathscr{D}_{k,[-l]}, \mathscr{M}_j\right)$. In this DOUBLE BPS framework, we focus exclusively on $\mathscr{D}_k$, which is treated equivalently to $\mathscr{M}_j$ in the first step. It is crucial to discriminate the predictive performance induced by each $\mathscr{D}_k$. We must utilize a common set of Y across all $\mathscr{D}_k$ – namely, Y itself. This stems from the construction of DOUBLE BPS. Specifically, for DOUBLE BPS to be effective, it necessitates predictive assessments over a common set of points for each model in the competition. Otherwise, the predictive performances cannot be directly compared, and the weights cannot be optimized to distinguish predictive capabilities across models, as different points would be used for different models. To illustrate, consider the first stacking step. Here, we compute $p\left(Y_{k,i} \mid \mathscr{D}_k, \mathscr{M}_j\right)$ for each subset, where $i = 1,\ldots,n_k$ and $j = 1,\ldots,J$. This allows us to evaluate Y_k with respect to the predictive density of all J models under comparison. Similarly, in DOUBLE BPS, the goal is to evaluate Y_i with respect to the predictive density across all K subsets (acting as competing models) for comparison. The weights $\{\hat{z}_{k,j}\}$, which are derived from the optimization problem specified in Equation (2.6) also appear in Equation (2.13). However, comparing the right-hand sides of Equations (2.13) and (2.6), we observe that the objective functions are almost identical, with the only difference being the second convex linear combination governed by the weights $\{w_k\}$. Therefore, the predictive distributions in both optimization problems refer to the same quantity. To summarize, the objective function in Equation (2.6) can be derived by substituting each $\{z_{k,j}\}$ with its optimized counterpart $\{\hat{z}_{k,j}\}$ and incorporating the weights $\{w_k\}$. This leads to the maximization objective in Equation (2.13).

Next, we consider the optimization problem in Equation (2.6) with objective function,

$$\max_{z_k \in \mathcal{S}_1^J} \frac{1}{n_k} \sum_{i=1}^{n_k} \log \sum_{j=1}^{J} z_{k,j} p\left(Y_{k,i} \mid \mathcal{D}_{k,[-l]}, \mathscr{M}_j\right) = \max_{z_k \in \mathcal{S}_1^J} f\left(z_k\right), \tag{2.14}$$

where $f\left(z_k\right) = \frac{1}{n_k} \sum_{i=1}^{n_k} \log \sum_{j=1}^{J} z_{k,j} p\left(Y_{k,i} \mid \mathcal{D}_{k,[-l]}, \mathscr{M}_j\right)$. An explicit form of $f(z_k)$ is

$$\begin{aligned} f(z_k) = f\left(z_{k,1}, \ldots, z_{k,J}\right) &= \frac{1}{n_k} \sum_{i=1}^{n_k} \log \sum_{j=1}^{J} z_{k,j} p\left(Y_{k,i} \mid \mathcal{D}_{k,[-l]}, \mathscr{M}_j\right) \\ &= \frac{1}{n_k} \sum_{i=1}^{n_k} \log \sum_{j=1}^{J} z_{k,j}\, \mathrm{T}_{1,q}(Y_{k,i} \mid \nu_{[-l]}^{\star}, \mu_i^{\star}, V_i^{\star}, \Psi_{[-l]}^{\star}) \\ &= \frac{1}{n_k} \sum_{i=1}^{n_k} \log \sum_{j=1}^{J} z_{k,j} K(Y_{k,i}) \left|1 + V_i^{\star -1}(Y_{k,i} - \mu_i^{\star})\Psi_{[-l]}^{-1\star}(Y_{k,i} - \mu_i^{\star})^{\mathrm{T}}\right|^{-\frac{\nu_{[-l]}^{\star}+1}{2}}, \end{aligned} \tag{2.15}$$

where $K(Y_{k,i}) = \dfrac{\left|\Psi_{[-l]}^{\star}\right|^{-\frac{1}{2}} \left|V_i^{\star}\right|^{-\frac{q}{2}} \Gamma_q\left(\frac{\nu_{[-l]}^{\star}+1}{2}\right)}{(\pi)^{\frac{1q}{2}} \Gamma_q\left(\frac{\nu_{[-l]}^{\star}}{2}\right)}$.

The logarithm of a linear combination precludes further accessibility, but $f(z_k)$ is computed easily by evaluating the matrix-T density. This is a standard convex optimization (Yao et al., 2018); see Section 2.3.5 for further details.

The objective function in Equation (2.13) is related to Equation (2.6) as

$$\max_{w \in \mathcal{S}_1^K} \frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k \sum_{j=1}^{J} \hat{z}_{k,j} p\left(Y_{k,i} \mid \mathcal{D}_{k,[-l]}, \mathscr{M}_j\right) = \max_{w \in \mathcal{S}_1^K} g(w), \tag{2.16}$$

where $w = (w_1, \ldots, w_K)^{\mathrm{T}}$ and

$$\begin{aligned} g(w) &= \frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k \sum_{j=1}^{J} \hat{z}_{k,j} p\left(Y_{k,i} \mid \mathcal{D}_{k,[-l]}, \mathscr{M}_j\right) \\ &= \frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k \sum_{j=1}^{J} \hat{z}_{k,j}\, \mathrm{T}_{1,q}(Y_{k,i} \mid \nu_{[-l]}^{\star}, \mu_i^{\star}, V_i^{\star}, \Psi_{[-l]}^{\star}) \\ &= \frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k \sum_{j=1}^{J} \hat{z}_{k,j} K(Y_{k,i}) \left|1 + V_i^{\star -1}(Y_{k,i} - \mu_i^{\star})\Psi_{[-l]}^{-1\star}(Y_{k,i} - \mu_i^{\star})^{\mathrm{T}}\right|^{-\frac{\nu_{[-l]}^{\star}+1}{2}}. \end{aligned} \tag{2.17}$$

In practical applications, we address Equation (2.13) separately from Equation (2.6), which is defined for each k^{th} subset. Although it must be solved for each subset, we perform K separate maximizations of Equation (2.6), one for each subset. Once we obtain all K sets of $\{\hat{z}_{k,j}\}$, we can recover the weights $\{\hat{w}_k\}$ across subsets by solving the convex optimization problem in Equation (2.13).

Overlooking the possibility of computing the subset stacking weights $\{z_{k,j}\}$ in parallel for

the time being, this method offers a significant computational benefit: no extra quantities need to be computed to solve the problem in Equation (2.13). All necessary information is already available from independent computations performed within each subset. For clarity, all the terms in Equations (2.6) and (2.13) are known and coincide except for the weights w, which still need to be optimized. Consequently, there is no need to recompute the cross-validated predictive distributions or $\{\hat{z}_{k,j}\}$ as all these components have been computed.

2.3.2 Theoretical complexity

In terms of theoretical computational complexity, we provide a comparison between spatial meta-kriging (SMK Guhaniyogi and Banerjee, 2018) and the DOUBLE BPS. So far, given a dataset $\mathscr{D}_n$, n denotes the total number of observations and K the number of subsets. Let M be the number of target posterior samples. From Guhaniyogi and Banerjee (2018), it is known that each posterior sample, considering a parallel implementation over K cores of SMK requires $O\left((\frac{n}{K})^3\right)$. Thus, taking into account M draws from each of the subset posteriors yields theoretical complexity of $O\left(M(\frac{n}{K})^3\right)$, for each computational core. The cost of computing the Geometric median also has to be added. As stated in Minsker et al. (2017), Weiszfeld's algorithm has a complexity of $O(M^2)$ for each step, and it needs at most $O(1/\epsilon)$ steps to approximate the full posterior to a degree ϵ of accuracy. Especially in large-scale applications, having a computational core for each subset is not always possible. Let us consider the number of available cores hereafter as m, generally $m << K$. Leading the total complexity of SMK, for K partitions in parallel over m cores to $O\left(\frac{K}{m}\left[M(\frac{n}{K})^3\right] + \frac{(KM)^2}{\epsilon}\right)$.

For theoretical complexity of double Bayesian predictive stacking, we have to specify J as the number of competitive models, and L as the number of folds used for cross-validation. Equivalently to SMK, model fitting within subsets is dominated by Cholesky decompositions implying costs in the order of $O\left((\frac{n}{K})^3\right)$. Nevertheless, in DOUBLE BPS, we perform J Cholesky decompositions, and for each of them, we refit the model L times. Hence, the theoretical complexity boils down to $O\left(\frac{K}{m}JL(\frac{n}{K})^3\right)$. In addition, we use the package CVXR Fu et al. (2020) in the R statistical computing environment by applying disciplined convex programming Grant (2005); CVX Research (2012) to find the stacking weights in polynomial time using an interior-point algorithm. We used the solvers SCS (O'Donoghue et al., 2016) and ECOSOLVER (Fu et al., 2023) to obtain the stacking weights. This introduces the discipline convex problems into the theoretical complexity, turning out to be $O\left(\frac{K}{m}[JL(\frac{n}{K})^3 + J^p] + K^p\right)$, for K subsets over m cores, and a polynomial degree p (which depends on the chosen interior-point algorithm). The portion in square brackets pertains to model fitting within each subset, consisting of a term related to cross-validation and the polynomial cost of DOUBLE BPS across J models. Finally, we account for the complexity introduced by the second stacking process across the K subsets.

Next, we compare the computational complexities of the two approaches. We will separately examine the terms associated with subset modeling and global inference combination. Thus,

for SMK and DOUBLE BPS, respectively, the computational complexities are as follows:

$$O\left(\frac{K}{m}\underbrace{\left[M\left(\frac{n}{K}\right)^3\right]}_{\text{subset modeling}} + \underbrace{\frac{(KM)^2}{\epsilon}}_{\text{combination}}\right), \qquad O\left(\frac{K}{m}\underbrace{\left[JL\left(\frac{n}{K}\right)^3 + J^p\right]}_{\text{subset modeling}} + \underbrace{K^p}_{\text{combination}}\right) \tag{2.18}$$

Focusing on the subset modeling component, as highlighted in Equation (2.18), two key specifications stand out. First, consider the difference in magnitude between M and the product JL. In this context, DOUBLE BPS offers a theoretical advantage when $JL < M$, a quite common condition in practice. This is because M represents the number of posterior samples required for convergence across all the Markov chains involved, and it typically needs to be at least on the order of 10^3. In contrast, the product JL consists of relatively small terms, making it highly likely that this inequality will hold. Second, due to the significant difference in scale , the term J^p is absorbed by $(n/K)^3$

When comparing the combination phase, the analysis reduces to a comparison between the geometric median approximation and Bayesian predictive stacking. Since a substantial number of posterior samples M is required by SMK for each of the K partitions, we generally find that $K^p < (KM)^2/\epsilon$. Thus, while empirical computational times are significantly lower for double Bayesian predictive stacking compared to SMK (see Section 2.4.4), there are some modest theoretical differences between the two methods. The major advantage lies in avoiding simulation-based methods, such as MCMC, while achieving local inferences through exact approaches.

Like Weiszfeld's algorithm, modern disciplined convex programming encounters computational challenges in high-dimensional contexts, particularly in managing random memory allocation. In Section 2.3.3, we present a feasible strategy for approximating the DOUBLE BPS weights, tailored for very large-scale memory problems.

2.3.3 Memory management and Pseudo-BMA

When modeling GEOAI systems, as the number of locations exceeds the order of millions, managing storage space becomes crucial. Timing issues may arise depending on the available optimizer. While open-source solvers theoretically offer faster solutions compared to iterative algorithms, e.g. geometric median, they often face practical challenges when the problem size considerably exceeds dimensions of 10^2. In contrast, commercial optimizers behave slightly better, even if these approaches are not exempted from random allocation memory constraints. We emphasize working with portable approaches, i.e. with open-source solvers, that can effectively handle large-scale problems.

We present a computationally cheaper alternative that facilitates better management of available RAM. The subsequent contents, including Algorithm 3, were implemented in data analyses involving millions (10^6) of locations of Sections 2.5. When addressing optimization problems of significant dimensions, AIC-based alternatives could be considered.

To facilitate model stacking, various methodologies exist within the Bayesian model aver-

Algorithm 3 Calculating stacking weights between subsets using pseudo-BMA

Input: $\hat{z} = \{\hat{z}_k = \{\hat{z}_{k,j}\} : k \in \{1,\dots,K\}, j \in \{1,\dots,J\}\}$: Stacking weights within subsets; $\{pd_{k,j,i} = \mathrm{T}_{1,q}(Y_{k,i} \mid \nu^{\star}_{[-l]}, \mu^{\star}_i, V^{\star}_i, \Psi^{\star}_{[-l]}) : k = 1,\dots,K, j = 1,\dots,J, i = 1,\dots,n\}$: point-wise predictive density of Y; n, q, p: Number of rows, number of outcomes, and number of predictors; $K, \{n_k : k \in \{1,\dots,K\}\}, J$: Number of subsets, dimension of each subset, and number of competitive models in each subset.
Output: $\hat{w} = \{\hat{w}_k : k = 1,\dots,K\}$: Stacking weights between subsets.

1: Construct $pd = \underbrace{[pd_1^{\mathrm{T}} : \cdots : pd_K^{\mathrm{T}}]^{\mathrm{T}}}_{n \times J}$, $pd_k = \underbrace{\begin{bmatrix} pd_{k,1,1} & \dots & pd_{k,J,1} \\ \vdots & pd_{k,j,i} & \vdots \\ pd_{k,1,n_k} & \dots & pd_{k,J,n_k} \end{bmatrix}}_{n_k \times J}$
2: **for** $k = 1,\dots,K$ **do**
3: Compute $\widehat{elpd}^{\,k} = \sum_{i=1}^{n} \log\left(pd\ \hat{z}_k\right)$
4: **end for**
5: **for** $k = 1,\dots,K$ **do**
6: Compute $\hat{w}_k = \exp\left(\widehat{elpd}^{\,k}\right) / \sum_{k=1}^{K} \exp\left(\widehat{elpd}^{\,k}\right)$
7: **end for**
8: **return** $\hat{w} = \{\hat{w}_k : k \in \{1,\dots,K\}\}$

aging (BMA) framework. In particular, we present an approach based on information criteria, and formerly introduced in Yao et al. (2018). To ensure comparability between datasets and enhance interpretability, we estimate the expected log point-wise predictive density (as done in DOUBLE BPS). The expected log pointwise predictive density ($elpd$) for each partition k is defined as

$$\widehat{\mathrm{elpd}}^{\,k} = \sum_{i=1}^{n} \widehat{\mathrm{elpd}}_i^{\,k} = \sum_{i=1}^{n} \log\ p(Y_{k,i} \mid \mathscr{D}_{k,[-l]}) \tag{2.19}$$

Importantly, each $elpd$ term need not be computed individually, as these values are generated during the first BPS procedure within each subset for all model configurations, significantly reducing memory storage requirements and the total computational burden. Given the set $\{\widehat{\mathrm{elpd}}^{\,k}\}_{k=1,\dots,K}$, the pseudo Bayesian model averaging (pseudo-BMA) weights are computed as

$$\hat{w}_k = \frac{\exp\left(\widehat{\mathrm{elpd}}^{\,k}\right)}{\sum_{k=1}^{K} \exp\left(\widehat{\mathrm{elpd}}^{\,k}\right)}. \tag{2.20}$$

This formulation, early introduced by Yao et al. (2018), simplifies the computation of the stacking weights, significantly reducing computational costs in terms of complexity and storage while maintaining the Bayesian predictive stacking framework. Thus, it serves as a viable alternative to BPS for predictive densities in challenging scenarios. When dealing with datasets comprising millions of instances and a substantial number of partitions, optimization solvers may fail or produce errors due to memory constraints, and the iterative processes involving large matrices can lead to increased procedure times.

Based on empirical experience, we primarily utilize pseudo-BMA as a Bayesian predictive stacking approach when datasets require excessive memory storage, particularly when

$n >> 10^5$ and $K >> 10^2$. When feasible we generally prefer convex optimization using BPS of predictive densities without reservations. Simulations highlighting potential differences in posterior predictive and posterior inference performances between these two model stacking approaches can be found in Yao et al. (2018), where several alternatives to Bayesian stacking approaches were discussed. We opted for pseudo-BMA due to its simpler analytical formulation, which enables matrix algebra to mitigate the computational burden of both random allocation memory and runtime.

2.3.4 Memory-efficient posterior sampling

In the matrix-variate conjugate Bayesian linear regression model presented in Equation (2.1), the Bayesian updating process becomes costly as the number of data shards, or even more, as the dimensions increase. Computational problems are often related to the available RAM, especially when working with considerable datasets. We present a memory-efficient posterior sampling for the regression coefficient β, helping in such intricate contexts. We consider the model in Equation (2.1) of Section 2.1,

$$\begin{aligned} &Y = X\beta + E \;, \quad E \mid \Sigma \sim \mathrm{MN}(O, V, \Sigma) \;; \\ &\beta = M_0 m_0 + E_\beta \;, \quad E_\beta \mid \Sigma \sim \mathrm{MN}(O, M_0, \Sigma) \;; \quad \Sigma \sim \mathrm{IW}(\Psi_0, \nu_0) \;, \end{aligned} \tag{2.21}$$

where Σ is assumed as known hereafter. Then, by matrix normal distribution theory, we know the exact form of the posterior distribution

$$\beta \mid \mathscr{D}, \Sigma \sim \mathrm{MN}(M_n m_n, M_n, \Sigma), \tag{2.22}$$

where $M_n^{-1} = M_0^{-1} + X^\top V^{-1} X$, $m_n = m_0 + X^\top V^{-1} Y$. We provide a memory-efficient way to sample from this distribution and reduce its computational burden. We define random variables $Y_{rep} \sim \mathrm{MN}(Y, V, \Sigma)$ and $Z \sim \mathrm{MN}(M_0 m_0, M_0, \Sigma)$. Expressing the relation between Y_{rep}, Z, and B as

$$M_n^{-1} B = A_1 Z + A_2 Y_{rep}. \tag{2.23}$$

We seek matrices A_1, A_2 such that $B \overset{d}{=} \beta \mid \mathscr{D}, \Sigma$. Since $\beta \mid \mathscr{D}, \Sigma$ is distributed as a matrix-Gaussian random variable, it is fully characterized by its mean and variance (in such a case, with both row and column covariance matrices). Then, all we need are A_1 and A_2 so that the first two moments of B matches with $\beta \mid \mathscr{D}, \Sigma$.

For $G \sim \mathrm{MN}(m, v, s)$, we have $DGC \sim \mathrm{MN}(DmC, DvD^\top, CsC^\top)$. Moreover, if G is $n \times q$, the row-variance matrix is defined as $v = \mathbb{V}_{row}(G) = \mathbb{E}[(G-m)(G-m)^\top]\,\mathrm{tr}(s)^{-1}$ of dimension $n \times n$ and its elements are defined as the variance computed on each row, while the $q \times q$ column covariance matrix is depicted by $s = \mathbb{V}_{col}(G) = \mathbb{E}[(G-m)^\top(G-m)]\,\mathrm{tr}(v)^{-1}$ (see, e.g., Gupta and Nagar, 2000, for further details). Without loss of generality, we compute the row covariance matrix for B since the column covariance matrix Σ is given. Note that

$$M_n^{-1} \mathbb{V}_{row}(B) M_n^{-1} = A_1 \mathbb{V}_{row}(Z) A_1^\top + A_2 \mathbb{V}_{row}(Y_{rep}) A_2^\top = A_1 M_0 A_1^\top + A_2 V A_2^\top \tag{2.24}$$

Setting these matrices as $A_1 = M_0^{-1}$, and $A_2 = X^\top V^{-1}$, we have

$$M_n^{-1}\mathbb{V}_{row}(B)M_n^{-1} = M_0^{-1}M_0M_0^{-1} + X^\top V^{-1}VV^{-1}X = M_0^{-1} + X^\top V^{-1}X = M_n^{-1}. \tag{2.25}$$

This implies $\mathbb{V}_{row}(B) = M_n$. The mean follows from

$$M_n^{-1}\mathbb{E}[B] = A_1\mathbb{E}[Z] + A_2\mathbb{E}[Y_{rep}] = M_0^{-1}M_0m_0 + X^\top V^{-1}Y = m_n, \tag{2.26}$$

and we obtain $\mathbb{E}[B] = M_n m_n$. Therefore, we can derive the next equality in distribution between B and the posterior distribution of the regression coefficient β as

$$B \overset{d}{=} \beta \mid \mathscr{D}, \Sigma \sim \text{MN}(M_n m_n, M_n, \Sigma). \tag{2.27}$$

This implies we can sample from $\beta \mid \mathscr{D}, \Sigma$ by solving a linear system. Specifically, by simply drawing samples from Z and Y_{rep}, we obtain a sample from $\beta \mid \mathscr{D}, \Sigma$ by solving the system $(M_0^{-1} + X^\top V^{-1}X)B = (M_0^{-1}Z + X^\top V^{-1}Y_{\text{rep}})$ for B. This approach is particularly advantageous for Bayesian transfer learning, as it avoids storing several large matrices when computing the posterior of $\beta \mid \mathscr{D}, \Sigma$. Instead, only the prior precision matrix for β, M_0^{-1}, and the product matrix $X^\top V^{-1}$ need to be stored, significantly reducing the memory footprint.

2.3.5 Computer programs and resources

All our subsequent analyses are implemented in native `R` and `c++` deploying the `spBPS` package. All programs required to reproduce the analysis are publicly accessible from the GitHub repository lucapresicce/Bayesian-Transfer-Learning-for-GeoAI that links the `Rcpp`-based `spBPS` package. The reported results are from a standard laptop running an Intel Core I7-8750H CPU with 5 cores for parallel computation and 16 GB of RAM.

We fit linear model of coregionalization (LMC, Banerjee et al., 2015), nearest-neighbor Gaussian process model (NNGP, Datta et al., 2016), and multivariate seemingly unrelated Bayesian additive regression trees (multivariate BART, Esser et al., 2025) using `spBayes`, `spNNGP` and `suBART` packages, respectively. We also compare with machine learning methods and AI systems using a scalable platform for parallelized supervised and unsupervised machine learning algorithms offered by `h2o` (Fryda et al., 2024). We specifically fit distributed random forest (DRF), gradient boosting (GBM), deep neural network (DNN), and a fully automatic machine learning algorithm (AUTOML).

For parallel implementations of DBPS, we employ `R` packages `doParallel`, and `foreach` (Microsoft and Weston, 2022; Microsoft and Weston, 2022). We map the interpolated spatial surfaces using `MBA` (Finley et al., 2011), while sampling from the matrix-variate normal and t distributions is achieved using `mvnfast` (Fasiolo, 2014). Section 2.3 specifies computational considerations and sensitivity to the number of data shards, K, for spatial "BIG" data analysis.

We build a Bayesian transfer learning engine to conduct amortized Bayesian inference (Zammit-Mangion et al., 2024) using DBPS. We implement a residual neural network (ResNet) (He et al., 2015) using the `R` interfaces supplied by `tensorflow` (Allaire et al., 2024) and `keras` (Kalinowski et al., 2024) for native `Python`.

2.4 Simulation experiments

We evaluate computational and inferential performances of DOUBLE BPS, while underscoring comparisons with multiple alternative methodologies.

2.4.1 Transfer learning in $\mathscr{M}$-closed & $\mathscr{M}$-open settings

We evaluate inferential and predictive performance under different settings. We explore how DOUBLE BPS behaves in the $\mathscr{M}$-closed and $\mathscr{M}$-open settings and compare with an exact transfer learning framework we devise in Section 2.1. We perform the experiment using 50 replications. Each replicate consists of values of $n \times q$ outcome Y generated from (2.2) with $n = 5,000$, $q = 3$, and $p = 2$, X includes an intercept and a predictor generated from a standard uniform distribution over $[0, 1]$, $\beta = \begin{bmatrix} -0.75 & 1.05 & -0.35 \\ 2.20 & -1.10 & 0.45 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 2.00 & 0.80 & 0.20 \\ 0.80 & 2.00 & -0.45 \\ 0.20 & -0.45 & 2.00 \end{bmatrix}$. The $n \times n$ spatial correlation matrix V is specified using an exponential correlation function with $\phi = 4$ and $\alpha = 0.8$.

For exact transfer learning, model specification is characterized by different values of α and ϕ representing (i) well-specified (WS) setting with the data generating values $\{\alpha = 0.8, \phi = 4\}$; (ii) moderately misspecified (MS) setting with $\{\alpha = 0.45, \phi = 6.63\}$; and (iii) a highly misspecified (HMS) setting with $\{\alpha = 0.25, \phi = 50\}$.

Then, the DOUBLE BPS was tested over $\mathscr{M}$-closed and $\mathscr{M}$-open settings. The former considers situations where the true model exists, and it is identified within a finite set of considered models. Here, the "true" model is the one such that $\{\alpha = 0.8, \phi = 4\}$. Then, for DOUBLE BPS under closed setting (BPS-C) we specify $J = 9$ competitive models with $\alpha \in \{0.75, 0.80, 0.85\}$ and $\phi \in \{2, 4, 6\}$ that yields effective spatial ranges in the percentage of maximum point inter-distance of 105%, 53%, 35% respectively, including the true model as one of the possible candidates. Conversely, in the $\mathscr{M}$-open setting, even though the true model exists, it cannot be fully specified. Thus, for DOUBLE BPS under the open setting (BPS-O), we randomly define $J = 9$ candidate models. In particular, we uniformly sample 3 values for $\alpha \in (0, 1)$ and 3 values for $\phi \in (0, 50)$.

Figure 2.1 presents (i) mean square prediction error (MSPE); (ii) predictive interval width; (iii) absolute bias; and (iv) variance. We present boxplots for the distribution of each metric over the 50 replicates. This is made for each response and for each setting. In terms of predictive MSPE, absolute bias, and variance, the settings BPS-C, BPS-O depict slightly better performance. However, this seems to be compromising predictive interval width as the uncertainty of prediction results is much higher. By approaches that estimate $\{\alpha, \phi\}$, rather than fix them, more uncertainty is somehow expected. In addition, we find no evidence of any difference between BPS-C, and BPS-O for any metric. This is surprising, as it suggests the reliability of DBPS even under the transfer learning setting we devise in Section 2.2. Finally, irrespective of $\mathscr{M}$-closed or $\mathscr{M}$-open settings, it is more convenient to specify a set of candidate models using DOUBLE BPS instead of attempting to fix $\{\alpha, \phi\}$.

Figure 2.2 presents posterior inference for (i) average empirical bias; (ii) average coverage; and (iii) average standard deviation, where the average is taken over the 50 replications. As expected, misspecification induces empirical bias in posterior estimates. The top panel reports

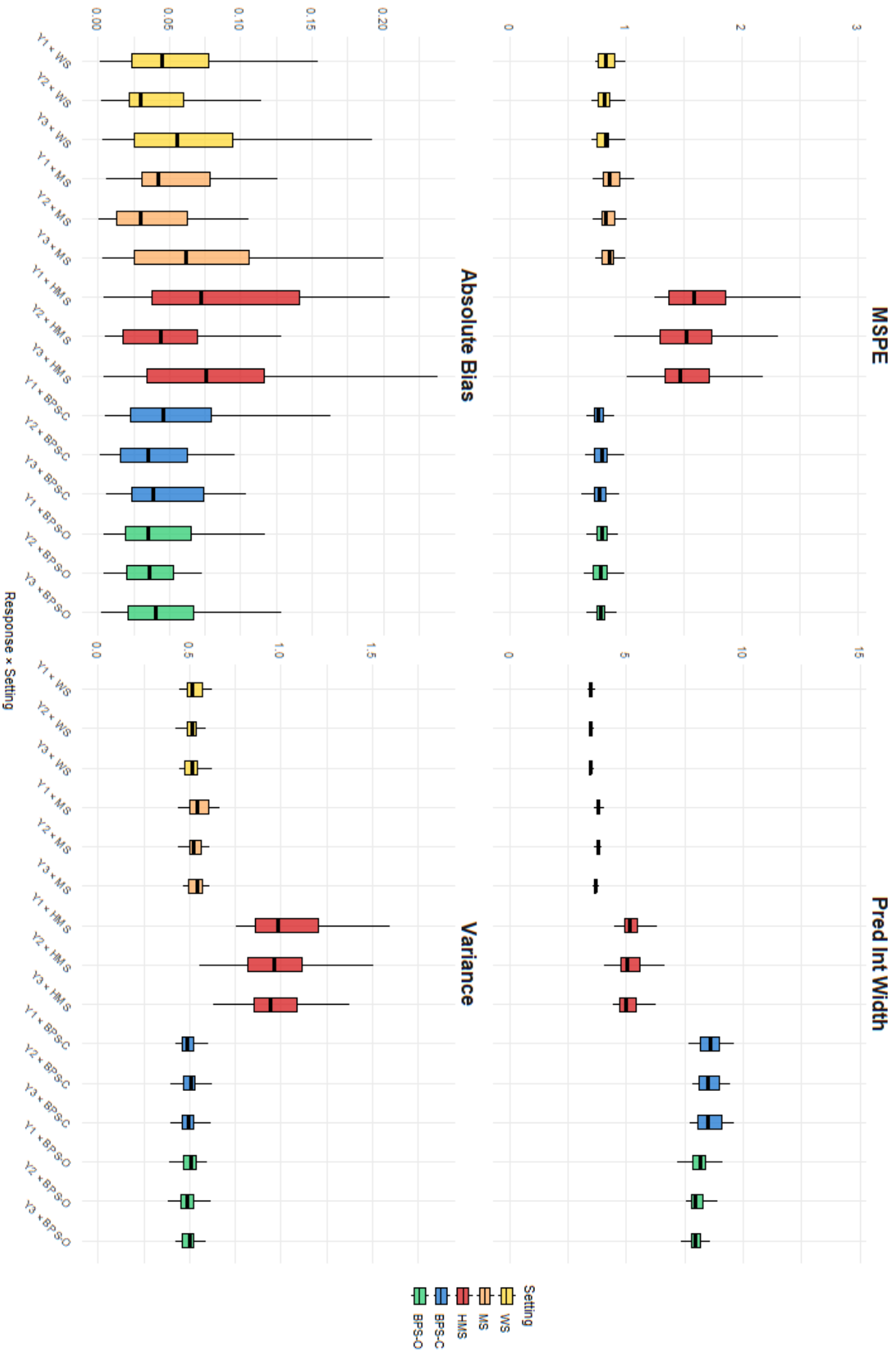

Figure 2.1: Predictive MSPE, interval width, absolute bias, and variance boxplot across responses and settings from 50 replications.

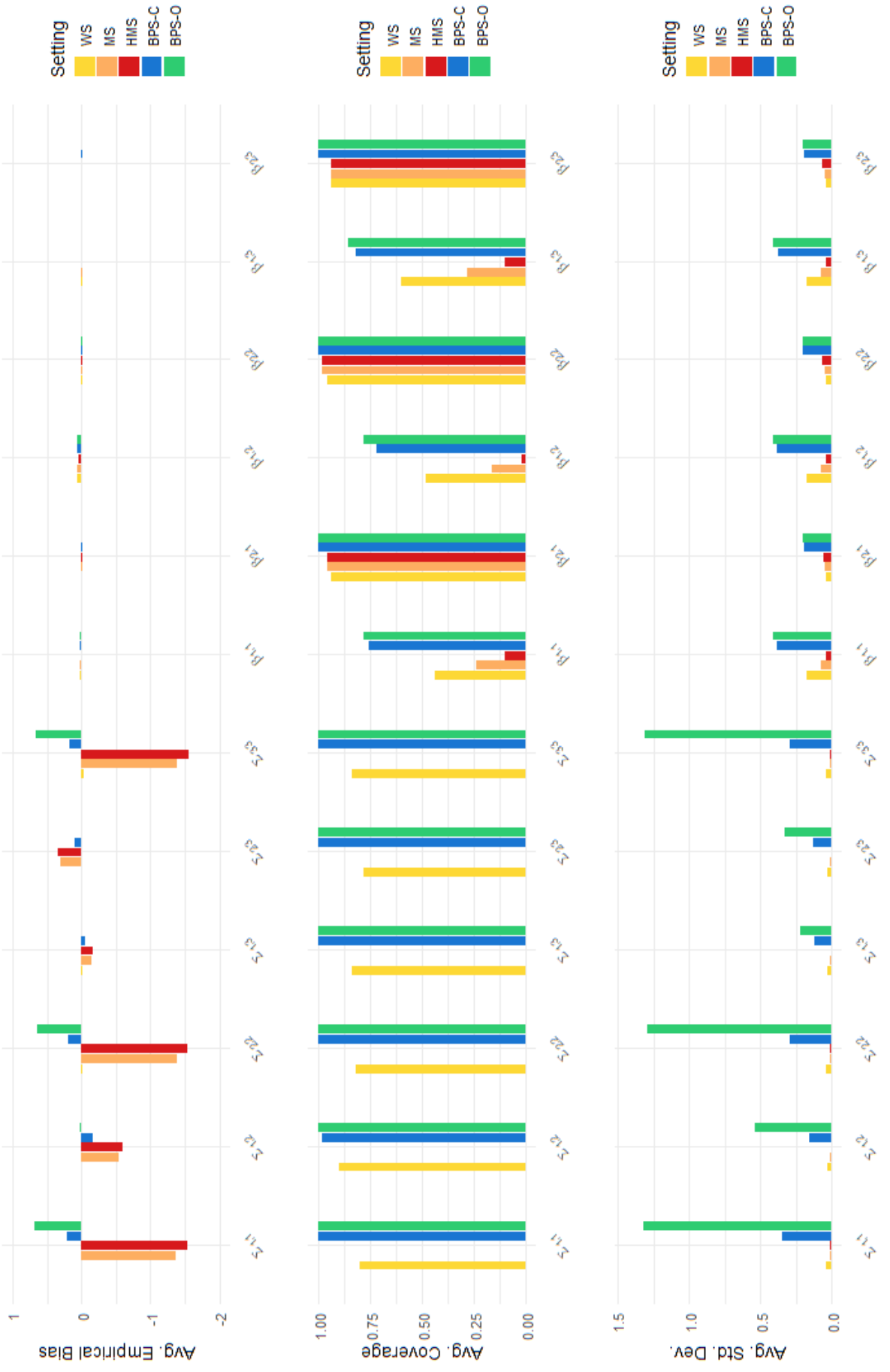

Figure 2.2: Average posterior bias, coverage, and standard deviation across parameters and settings from 50 replications.

how greater levels of bias are associated with models that are farther away from the truth. Here, DOUBLE BPS is placed in the middle for both $\mathscr{M}$ settings. The middle plot in Figure 2.2 shows that Σ is the most affected parameter by misspecifications. Its elements are well captured only by DOUBLE BPS, with coverage close to nominal, followed by WS specification, which performs worse. A Similar pattern holds for β, where only DOUBLE BPS ensures adequate coverage. This reflects greater posterior variability in stacking approaches, as shown in the bottom panel reporting posterior standard deviations across settings. Unsurprisingly, inferential performance is weakest for the spatial variance, which is not identifiable from the realized data (Zhang, 2004). DOUBLE BPS behaves very similarly among $\mathscr{M}$-closed and $\mathscr{M}$-open.

While DBPS outperforms transfer learning in $\mathscr{M}$-closed and $\mathscr{M}$-open settings, transfer learning improves predictive performance over a well-specified model with misspecified data than with well-specified data. This, while somewhat unexpected, is attributed to the estimated response surfaces being practically indistinguishable. Furthermore, transfer learning in misspecified settings perform competitively. These findings seem to be consistent with theoretical insights offered by Stein (1988) and Stein and Handcock (1989), who theoretically established the ability of Gaussian processes to deliver good predictive performance even for misspecified covariance functions in fixed domains (in-fill asymptotics).

2.4.2 Predictive coverage performance

We evaluate the predictive coverage performance and computational efficiency of our proposed framework using a synthetic spatial dataset comprising $2,250$: $n = 2,000$ locations used for training, and $u = 250$ held out for predictive evaluations. We also include a design matrix X with $p = 2$ comprising an intercept and a single predictor whose values were sampled independently from a uniform distribution over $[0, 1]$, and a univariate ($q = 1$) response Y. Spatial coordinates are sampled uniformly over the unit square $[0, 1]^2$, and the $n \times n$ spatial correlation matrix over these coordinates using $\rho_\phi(s_i, s_j) = \exp(-\phi\|s_i - s_j\|)$ with $\phi = 4$. The response is generated according to a Gaussian process model with regression coefficients $\beta = (1.0, 0.5)^\top$, spatial variance $\sigma^2 = 1$, and nugget variance $\tau^2 = 0.25$, which corresponds to $\alpha = 0.8$.

While our methodology is inherently multivariate, we restrict this study to a univariate response and a moderate sample size. This design allows direct comparison with established gold-standard approaches: full Gaussian process models (Full GP) and nearest-neighbor Gaussian process (NNGP) models. Larger or multivariate datasets would render repeated full GP analyses infeasible, yet these settings are sufficient to evaluate predictive coverage, MSPE, and interval width across competing methods.

We assess distributed learning via DBPS, varying the number of subsets $K \in 5, 10, 20$, with candidate models defined over a grid of hyperparameters $\alpha \in 0.7, 0.8, 0.9$ and $\phi \in 3, 4, 5$. For benchmarking, we fit NNGP models with neighbor numbers ranging in $m \in 5, 10, 20$ and a full GP model as a benchmark. For both simulation-based methods, we specify the same non-informative default priors and $2,000$ MCMC samples. Each method is replicated $B = 50$ times to capture variability in predictive performance, and then the results are averaged over replications.

Model	Time (sec)	Pred. Int. Width	MSPE	Emp. Coverage
FULL GP	1654	**2.00**	**0.264**	0.945
NNGP ($m = 5$)	17	2.01	0.264	0.948
NNGP ($m = 10$)	39	2.01	0.264	**0.949**
NNGP ($m = 20$)	120	2.01	0.264	0.947
DBPS ($K = 5$)	24	2.04	0.265	0.946
DBPS ($K = 10$)	13	2.11	0.268	0.956
DBPS ($K = 20$)	**12**	2.19	0.271	0.961

Table 2.1: Average predictive interval width, MSPE, empirical coverage at 95%, and computation time (in seconds) for different specifications of NNGP, DBPS, and full Gaussian process models. Results are averaged over 50 replications.

Table 2.1 summarizes the results in terms of average predictive interval width, mean squared prediction error (MSPE), empirical coverage, and computational time in seconds. Distributed DBPS achieves predictive accuracy and coverage comparable to NNGP and full GP models while dramatically reducing computation time with respect the latter. Moreover, computation time decreases as the number of subsets increases, highlighting the scalability of the framework without compromising inferential quality. These findings demonstrate that a distributed approach can efficiently replicate the predictive performance of full Gaussian process models, making it a practical alternative for large spatial datasets. These results demonstrate that double BPS achieves substantial computational savings relative to full GP models while maintaining comparable predictive accuracy and coverage. This simulation experiment illustrates the scalability of the framework without compromising inference quality, even with respect to Vecchia-style approximations as nearest-neighbor Gaussian process models.

The prediction oversmoothing, which typically happens in distributed approaches (e.g., Guhaniyogi and Banerjee, 2018), seems not to have a strong effect for double BPS. Despite this, it is evident that increasing the number of subsets induces some extra variability and extra coverage. However, simulations confirm the effect is negligible. The origin of this extra width of the predictive interval may arise from a disagreement term, often associated with linear pooling forecast (Knüppel and Krüger, 2022). We reserve the study of disagreement tempering techniques to remove possible variance inflation as future work.

2.4.3 Amortized Bayesian inference

We conduct transfer learning by supervising a neural network using the outputs of DOUBLE BPS to deliver amortized Bayesian inference. We generate 100 instances of Y from (2.2) using a fixed realization of Ω for $q = 2$ correlated outcomes, $n = 500$ spatial locations that remain fixed across the datasets, and a fixed design matrix X with $p = 2$ comprising an intercept and a single predictor whose values were sampled independently from a uniform distribution over $[0, 1]$. The true regression coefficients are fixed at $\beta = \begin{bmatrix} -0.75 & 1.85 \\ 0.9 & -1.10 \end{bmatrix}$, with $\Sigma = \begin{bmatrix} 1 & -0.3 \\ -0.3 & 1 \end{bmatrix}$, $\alpha = 0.8$, and $\rho_\phi(s_i, s_j) = \exp(-\phi\|s_i - s_j\|)$ with $\phi = 4$.

We train the neural network using $R = 250$ posterior samples by applying DOUBLE BPS to each generated dataset with $K = 5$ subsets, $\alpha \in \{0.7, 0.8, 0.9\}$ and $\phi \in \{3, 4, 5\}$. These yield 100 instances of $\{Z, \Theta\}$, where $Z = [Y : X] \in \mathbb{R}^{n\times(q+p)}$ and $\Theta \in \mathbb{R}^{[(qp)+(q(q+1)/2)+(nq)]\times 3}$ comprises the $\{2.5, 50, 97.5\}$ posterior quantiles for the distinct elements of $\{\beta, \Sigma, \Omega\}$.

We use a deep neural network comprising 3 hidden layers with 128, 256, and 512 nodes, with

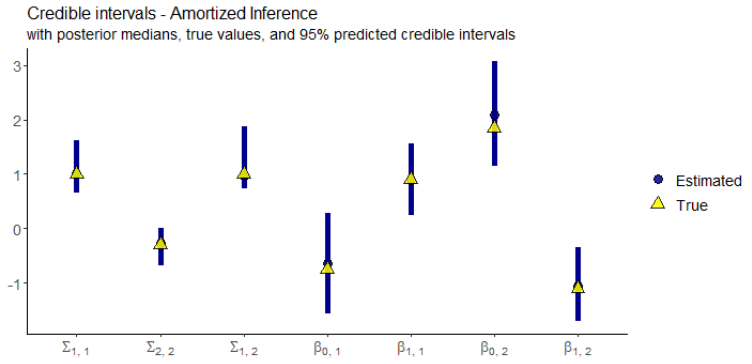

Figure 2.3: Amortized posterior credible intervals for parameters. True parameters in yellow.

ReLU activations. The residual network is trained over 50 epochs with 24 batches per epoch. For evaluation, we apply the trained model to unseen datasets with the same dimensions. Figure 2.3 displays predicted posterior intervals (blue bars) for $\{\beta, \Sigma\}$, alongside true parameter values (yellow triangles). The predicted medians (50%) align closely with the ground truth, and all true values are captured by the estimated 95% credible intervals, highlighting the deep network's strong predictive accuracy in recovering posterior summaries.

Figure 2.4 displays the predicted posteriors for Ω quantiles. We compare the results from Amortized inference, with the true Ω, and the DOUBLE BPS prediction for 50th quantile, presented in the first and the second column of Figure 2.4, respectively.

This experiment aims to show the strengths of amortized inference and transfer learning working together. Once trained, the deep network provides instant posterior quantile estimates for new datasets, without requiring re-running DOUBLE BPS, thus amortizing the computational cost across future tasks. Additionally, the model generalizes across a range of data-generating conditions, effectively enabling posterior transfer to new but structurally similar problems. This makes the approach especially useful in large-scale or resource-constrained applications where repeated full Bayesian inference would be prohibitively expensive.

2.4.4 Computational performance

We investigate running times of our framework using two synthetic datasets with common structures but distinct sizes. Both datasets consist of $p = 2$ predictors and $q = 2$ response variables, but different numbers of spatial locations, $n = 5,000$ and $n = 10,000$, respectively. We generate and fix spatial coordinates from a uniform distribution on the unit square ($[0,1]^2$). We build the $n \times n$ spatial correlation matrix V over these coordinates using $\rho_\phi(s_i, s_j) = \exp(-\phi\|s_i - s_j\|)$ with $\phi = 4$ and specify $\Sigma = \mathbb{I}_q$. From these specifications we generate the $n \times q$ matrix Y from the first equation of (2.2) with fixed $p \times q$ matrix $\beta = \left[\begin{smallmatrix} -0.75 & 1.85 \\ 0.90 & -1.10 \end{smallmatrix}\right]$, a fixed $n \times p$ matrix X with a first column of ones, representing the intercept, and $p - 1$ columns of values randomly simulated from a uniform distribution on $[0, 1]$ (emulating standardized predictors), and the proportion of spatial variability $\alpha = 0.8$. We compare with multivariate meta-kriging (Guhaniyogi and Banerjee, 2019), linear model of coregionalization LMC, and seemingly unrelated Bayesian additive regression tree (SUBART) (see Section 2.3).

For distributed learning approaches, i.e., DOUBLE BPS and MSMK, we perform the analyses

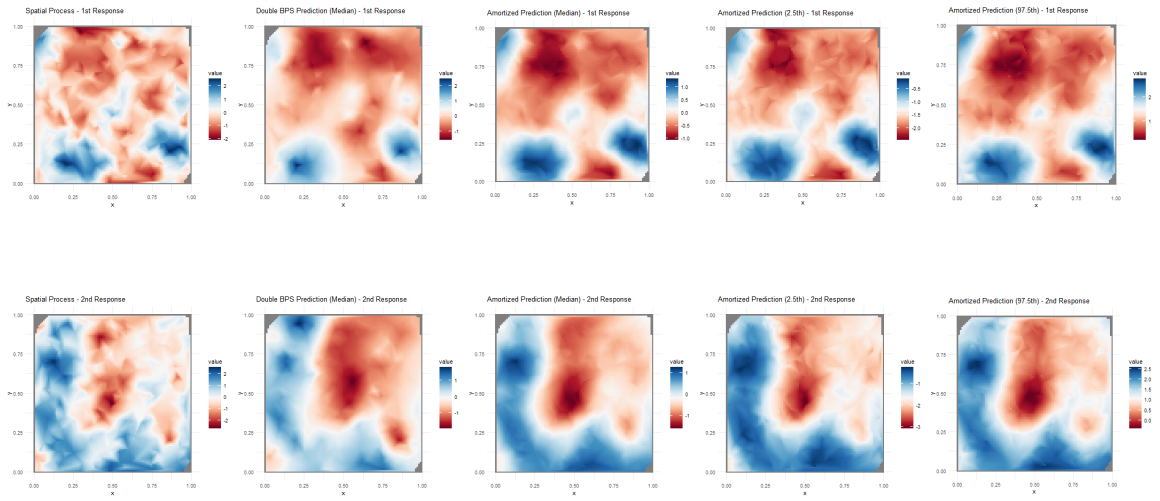

Figure 2.4: Surface interpolations for true spatial process, BPS prediction (50 quantile), and Amortized prediction of $\{50, 2.5, 97.5\}$ quantiles. Each row corresponds to an outcome.

in two settings: (i) $K = 10$ and $n = 5,000$; (ii) $K = 10$ and $n = 10,000$; (iii) $K = 5$ and $n = 5,000$; and (iv) $K = 20$ and $n = 10,000$. These settings produce subsets of size $n/K \in \{500, 1000\}$. We implement DOUBLE BPS using $J = 9$ candidate models $\mathscr{M}_j$, $j = 1, \ldots, J$, where each model is specified by a set of candidate values for the hyperparameters α_j, ϕ_j in (2.2). These hyperparameters represent the proportion of spatial variability and the parameter(s) of the spatial correlation function, respectively. The set of candidate models is constructed as the set of all possible combinations of values for these hyperparameters. In the subsequent experiments, the grid of models was built using $\alpha \in \{0.70, 0.80, 0.90\}$ and $\phi \in \{3, 4, 5\}$. These values resemble an effective spatial range of $\{0.99, 0.75, 0.60\}$ units, corresponding to 70%, 53%, 42% of the maximum inter-site distance inside the unit square, beyond which the spatial correlation drops below 0.05. Equation (2.3) follows choices in Zhang et al. (2021). Specifically, we set $m_0 = 0_{p\times q}$, $M_0 = 10\mathbb{I}_p$, $\Psi_0 = \mathbb{I}_q$, and $\nu_0 = 3$ in the MNIW joint prior for $\{\gamma, \Sigma\}$ in 2.3. For data analysis, we use an exponential spatial correlation function for $\rho_\phi(\cdot, \cdot)$, which is completely defined conditionally on $\mathscr{M}_j$, since specifies a value for ϕ. Again, in the conjugate framework, we draw $R = 250$ posterior samples used for inference.

We also apply MSMK to the two simulated datasets with the same combinations of n and K as DOUBLE BPS. Unlike DOUBLE BPS, where we stack analytically tractable posteriors over a range of fixed values of spatial covariance kernel parameters, the MSMK implementation attempts full Bayesian inference using prior distributions on spatial covariance kernel parameters. We fit the linear model of coregionalization described in Finley et al. (2015) for each subset of the multivariate spatial data using MCMC. The posterior samples from the K subsets are combined using Weiszfeld's iterative algorithm (Minsker et al., 2017) to produce an estimate of the geometric median of the posterior distributions. For both experimental settings, we then fit LMC, and SUBART on the full dataset, where they were endowed with default prior settings, following Finley et al. (2015), and Esser et al. (2025), respectively.

Table 2.2, which compares the computational speed of DOUBLE BPS with other considered

Setting	Model	Time (min)	Relative to DBPS	Visual
$n = 5000, K = 10$	DBPS	1.38	1.0×	
	MSMK	51.41	37.3×	
$n = 5000, K = 5$	DBPS	7.22	1.0×	
	MSMK	237.23	32.9×	
$n = 5000$	SUBART	215.67	29 – 156×	–
	LMC	8975.31	>1000×	
$n = 10000, K = 20$	DBPS	2.24	1.0×	
	MSMK	103.36	46.1×	
$n = 10000, K = 10$	DBPS	10.58	1.0×	
	MSMK	446.01	42.2×	
$n = 10000$	SUBART	557.60	53 – 249×	–
	LMC	–	–	–

Table 2.2: Running times (in minutes), relative to DOUBLE BPS. Bars give a visual impression of time cost (where applicable).

approaches reveals massive computational gains accrued from DOUBLE BPS. The computational advantage evinced from the relative ratio becomes more pronounced as the size of the data becomes larger, despite the larger subsets. This is explained by the fact that fitting the Gaussian process regression dominates the computation relative to the assimilation of inference from the subsets. If the number of locations explode, then the geometric median of posteriors required by MSMK is computationally unfeasible. While MSMK offers Bayesian estimates using MCMC for each subset, DOUBLE BPS avoids MCMC and, hence, issues of convergence. Similar arguments follow for LMC and SUBART. As expected, the linear model of coregionalization, when fitted on the entire set of locations, gives a disastrous performance, taking almost a week of computation in the lighter simulation settings, and makes it infeasible to record results for $n = 10,000$. Notwithstanding the scalability offered by this multivariate extension of the Bayesian additive regression model, the SUBART does not have any chance to provide inference in a comparable time.

Figure 2.5, which depicts estimated response surfaces using DOUBLE BPS and MSMK corresponding to $n = 5000$ and $K = 10$, shows that inferences are practically indistinguishable, but oversmoothed with respect to the true surfaces. This behavior is somehow expected, as typical of distributed approaches (Guhaniyogi and Banerjee, 2018). The root mean squared prediction error (RMSPE), reported in Figure 2.5, denotes the average squared differences between the generated and estimated values of the response and reveals minor discrepancies between DOUBLE BPS and MSMK.

Figure 2.6 reports 95% posterior predictive intervals for the response. Again, the empirical coverage is impressive. While we see slightly wider intervals from DOUBLE BPS, this is less pronounced than the underestimation of variability seen with MSMK. Moreover, Figure 2.6 reveals superior MAP estimates for the DOUBLE BPS. Finally, Figure 2.7 presents the recovery of parameter estimates. As seen in predictive inference, the posterior credible intervals for parameters also deliver practically indistinguishable inference for the two modeling frameworks. In particular, both methods recover the true values for β and Σ, while DOUBLE BPS reconstructs a better point estimate for range parameters ϕ using $\sum_{k=1}^{K} \hat{w}_k \sum_{j=1}^{J} \hat{z}_j \phi_j$. The presented results show that DOUBLE BPS achieves spatial interpolation performance comparable to MSMK, while requiring

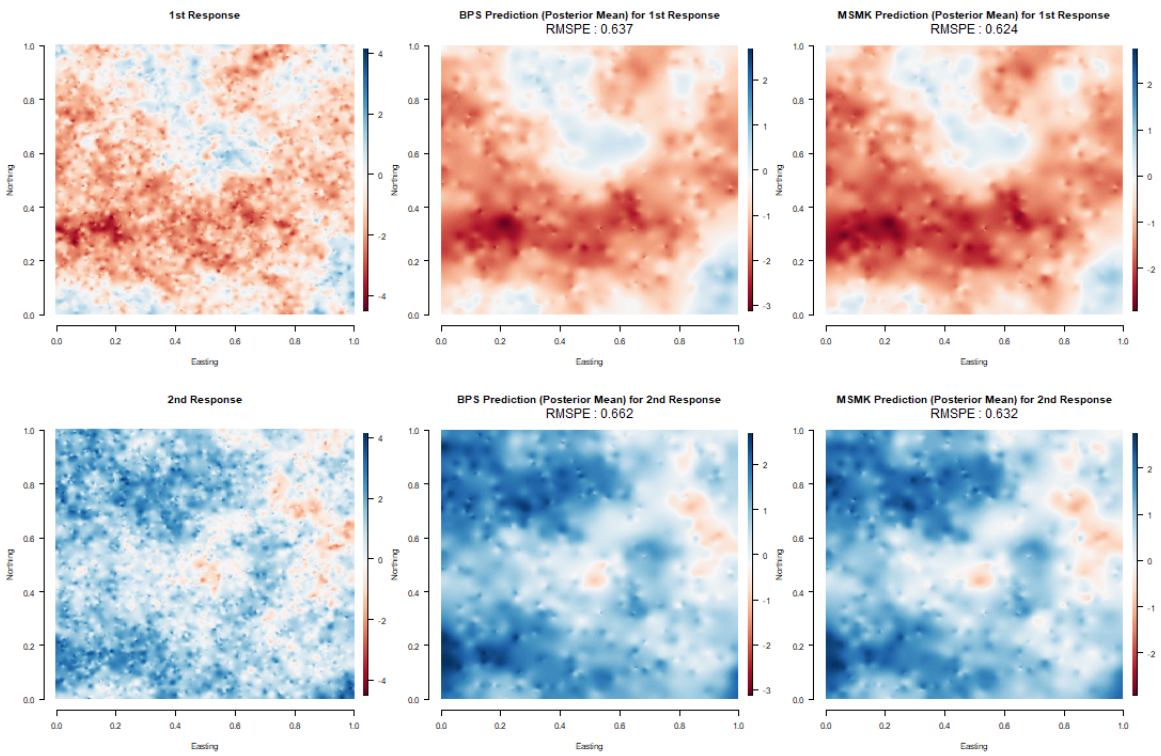

Figure 2.5: from left to right: comparison between the true generated response surfaces, the surfaces predicted from DOUBLE BPS and MSMK (posterior mean), with RMSPE. For $n = 5000$, $K = 10$.

only a fraction of the computational cost.

2.4.5 Subset size sensitivity

The methodological novelty introduced in Section 2.2 can be summarized in three main steps, as illustrated in Figure 1.1. First, we partition the original, often massive, dataset into K smaller subsets. The number of locations in each partition is a critical decision, seriously impacting inferential, predictive, and computational outcomes. Accordingly, a trade-off arises between computational resources and performance.

To address this, we conduct a simulation analysis to assess the sensitivity of the results to subset size. This section aims to investigate how predictive performance (in terms of RMSPE), and runtime (in seconds) change as the number of locations within each partition grows. Intuitively and theoretically, as the dimension of the subsets grows, we expect predictive performance to improve, while runtime increases polynomially with n. To enhance the comparability of the results, we apply min-max normalization to each variable, defined as $\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)}$. This normalization scales all variables to the interval $[0, 1]$, facilitating a more direct graphical comparison.

We utilize a multivariate synthetic dataset comprising $n = 5,000$ locations, $q = 2$ simulated responses, and $p = 2$ predictors to explore the sensitivity to subset size. This dataset is generated from the model in Equation (2.2), with parameters $\beta = \begin{bmatrix} -0.75 & 1.85 \\ 0.90 & -1.10 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 1.00 & -0.30 \\ -0.30 & 1.00 \end{bmatrix}$. The predictor matrix X includes an intercept and $p - 1$ columns generated from a standard uniform distribution over $[0, 1]$. The range parameter for the exponential spatial covariance function,

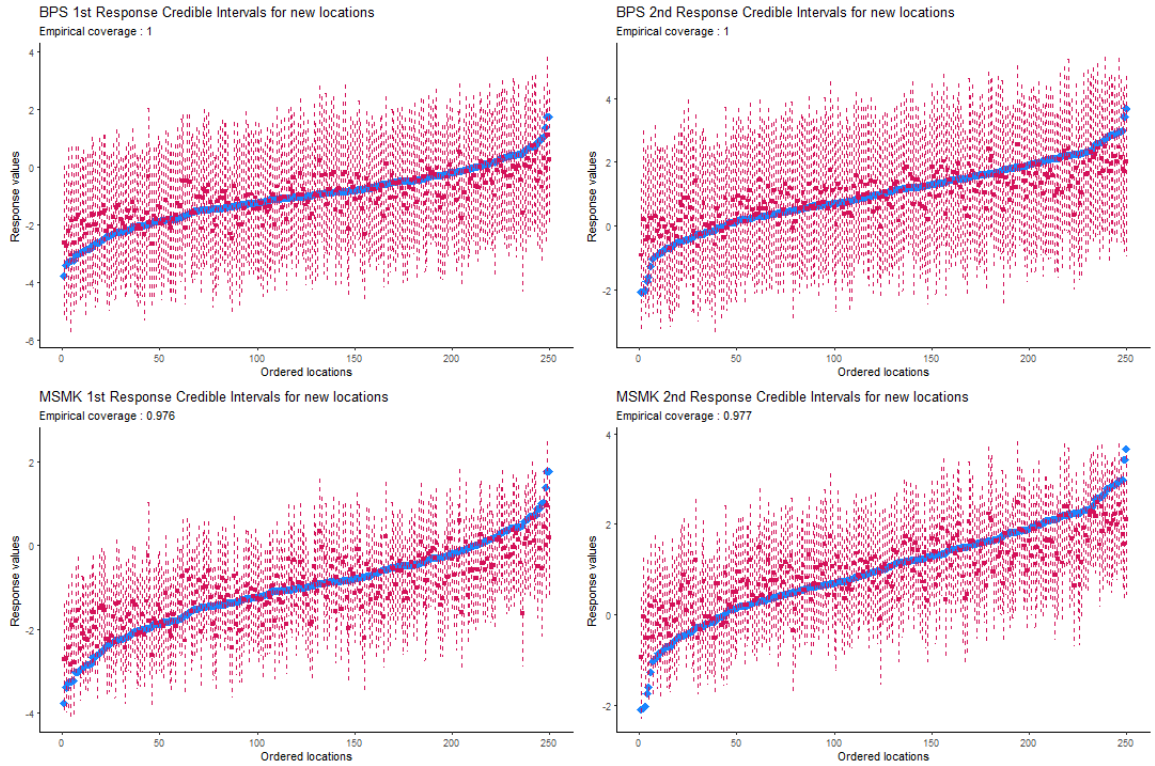

Figure 2.6: from top to bottom: comparison between posterior predictive intervals for the predicted response from DOUBLE BPS and MSMK, with empirical coverage. For $n = 5000$, $K = 10$.

and the proportion of spatial variability, are fixed at $\phi = 4$ and $\alpha = 0.8$ respectively. We set prior information as follows: $m_0 = 0_{p\times q}$, $M_0 = 10\mathbb{I}_p$, $\Psi_0 = \mathbb{I}_q$, and $\nu_0 = 3$. These specifications and prior information remain constant, allowing only the number of locations in each subset to vary. In performing DOUBLE BPS detailed in Section 2.2, we consider $J = 9$ competitive models characterized by $\alpha \in \{0.70, 0.80, 0.90\}$ and $\phi \in \{3, 4, 5\}$.

We focus our sensitivity analysis by selecting the following set of partition sizes: {25, 50, 100, 250, 500, 1000, 1250}, which correspond to the number of partitions $K \in$ {200, 100, 50, 20, 10, 5, 4}. Figure 2.8 illustrates the two curves resulting from this sensitivity analysis.

As anticipated, the behavior of the two curves aligns with theoretical expectations across most scenarios. Specifically, the total time required to fit the model increases monotonically, exhibiting more than linear growth in the number of locations within each partition, as the dashed line exhibits in Figure 2.8. Conversely, the root mean square prediction error experiences an unexpected fluctuation, likely due to the extremes of very low or high numbers of locations/subsets (50/100). RMSPE decreases until it stabilizes at a "plateau" for partition sizes of 500 units on. The trends observed in Figure 2.8 reveal a compromise between predictive performance and computational effort close to a subset size of 500 units.

However, Figure 2.8 also raises an important question: how much predictive error is acceptable? The two quantities, although normalized for comparison, differ significantly in their scales. More precisely, the trade-off is asymmetric: doubling the number of locations per partition yields a moderate reduction in RMSPE, while the runtime can increase dramatically, rising at least quadratically with n. For all these reasons, we generally opt for a subset size of 500 locations in both our simulation studies and data applications. Nonetheless, we should not

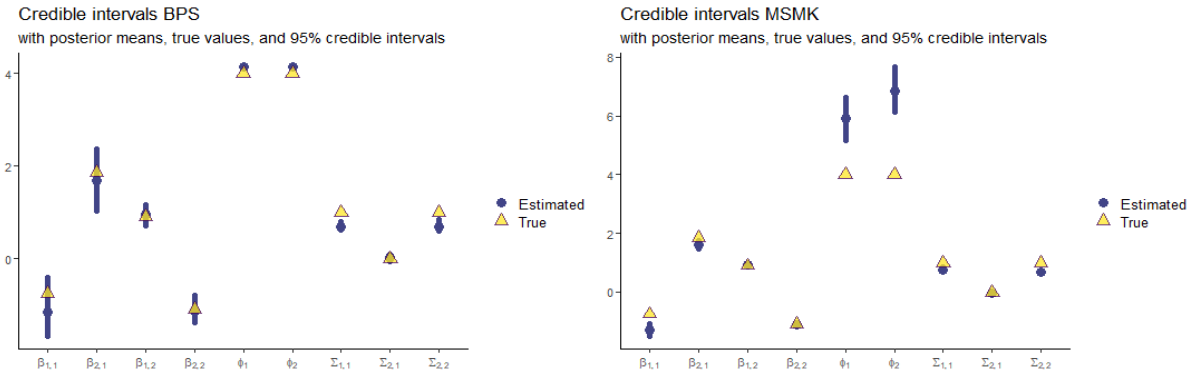

Figure 2.7: from left to right: comparison between posterior credible intervals for the parameters recovered from DOUBLE BPS and MSMK. For $n = 5000$, $K = 10$.

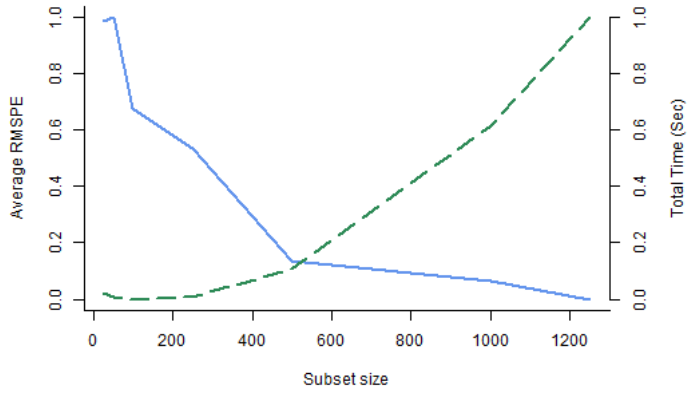

Figure 2.8: Comparison between average RMSPE (solid line) and model fitting time (dashed line) across various subset dimensions (both min-max normalized).

overlook the opportunity to reduce this size, accepting a trade-off in predictive performance to achieve even faster global Bayesian inference for exceptionally large GEOAI applications.

2.5 Application to MODIS data

Global warming and its critical consequences (see Fisher, 1958; Nicholls, 1989; Friehe et al., 1991; O'Carroll et al., 2019) are increasingly being investigated using machine learning and AI tools that assist in developing predictive global models and formulating data-driven policies. trying to produce faster solutions. The key aspects of global warming are related to global temperature, atmospheric and land surface compositions. We analyze data extracted from NASA's Moderate Resolution Imaging Spectroradiometer (MODIS) platform.

Specifically, we consider as response variables two vegetation indices present in VI data: the normalized difference vegetation index (NDVI), and the red reflectance (RR). Both measure the activity of vegetation on land surfaces and are commonly investigated to help understand biophysical and structural properties of vegetation at global scales (Walther et al., 2018; Jiang et al., 2020; Ai et al., 2020). We apply the multivariate model developed in Section 2.2, with NDVI and RR comprising the $q = 2$ columns of Y. The predictors comprise an intercept and the

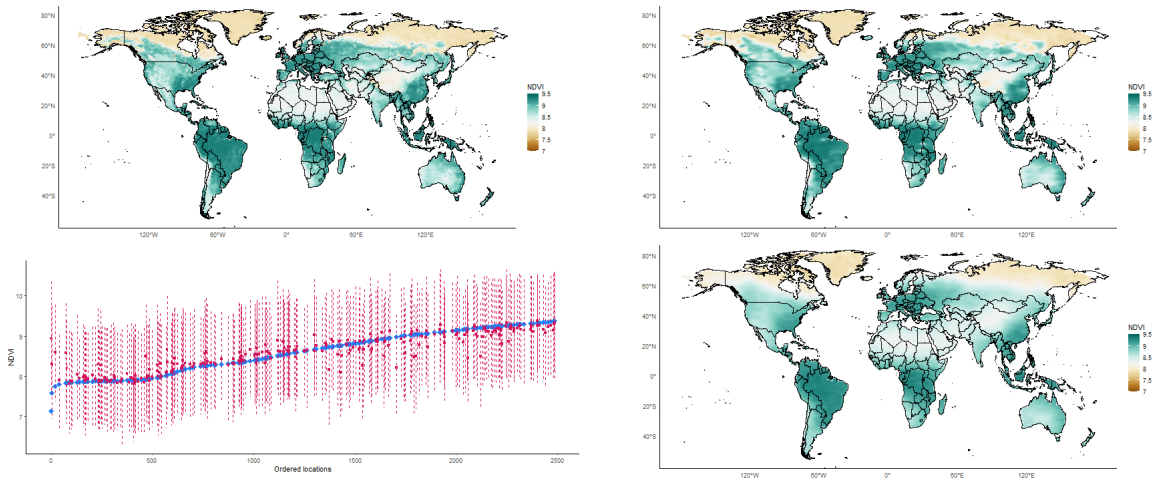

Figure 2.9: Left to right: Maps for training data (top left), test data (top right) and predicted surface (bottom right) for NDVI. Empirical coverage for held-out values are in the bottom left. Results correspond to $K = 2,000$.

solar zenith angle for that location ($p = 2$). The latter depicts the strength of solar irradiation. All the data were collected over a 16-day window in May 2024 and then averaged.

Following other investigations (Zhang and Banerjee, 2022), we report our analysis on all spatially dependent variables in the logarithmic scale, and we opt for an exponential spatial covariance function. Machine generated exploratory data analysis (EDA) is described in Section A.5 to guide grid settings for α, ϕ. Based on Section A.5, we set $\alpha \in \{0.825, 0.909\}$, and $\phi \in \{0.049, 0.067\}$ respectively. We specify $\{\gamma, \Sigma\}$ in (2.3) using $m_0 = 0_{p\times q}$, $M_0 = 10\mathbb{I}_p$, $\Psi_0 = \mathbb{I}_q$, $\nu_0 = 3$. The analyzed data consists of $1,002,500$ observed spatial locations across the world, of which $n = 1,000,000$ locations are used as training data and the rest are withheld for evaluating predictive performance. Following insights from Section 2.4.5, we fix the subset size at $n_k \in \{250, 500\}$, leading to a number of subsets $K \in \{4,000\ ,\ 2,000\}$ respectively. We use a random scheme to form the partition of the dataset and present results when $K = 2,000$ and in Table 2.3 for $K = 4,000$.

Figures 2.9 and 2.10 illustrate maps corresponding to NDVI and RR, respectively, using $K = 2,000$ subsets. The top left panel in each figure presents the spatially interpolated map of the training data for the respective responses. The interpolated maps for the test data are displayed in the top right panel of the respective figures. These maps reveal pronounced spatial variation, where the darker shades of green in NDVI represent higher values of detected biomass, while lighter shades of brown represent low biomass. Conversely for RR, warmer colors in the red spectrum represent higher reflectance.

Figures 2.9 and 2.10 show results from DOUBLE BPS. Maps of interpolated posterior means and predicted responses are displayed in the bottom right of the respective figures. The DBPS interpolated maps are almost indistinguishable from the true surface (top right). This suggests that the automated DBPS effectively, perhaps even strikingly, recovers the spatial patterns in spite of the modeling simplifications over more elaborate statistical models (Banerjee, 2020; Zhang and Banerjee, 2022). The predictive 95% credible intervals when $K = 2,000$ for both responses are shown in the bottom left of Figures 2.9 and 2.10, respectively.

We attempted comparisons with other Bayesian models, which represent the benchmark for

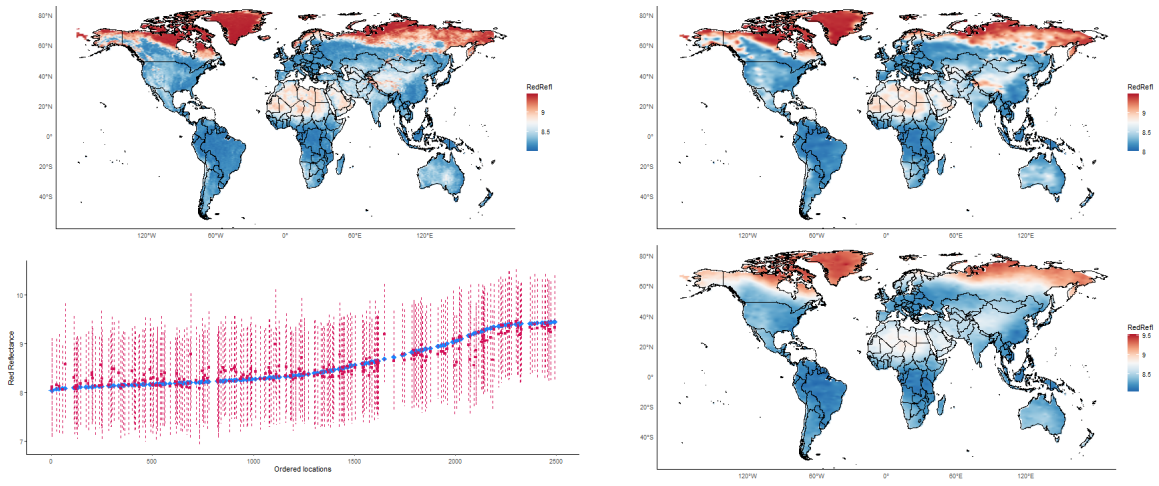

Figure 2.10: Left to right: Maps for training data (top left), test data (top right) and predicted surface (bottom right) for RR. Empirical coverage for held-out values of outcomes (bottom left). Results correspond to $K = 2,000$.

Parameter	Conjugate Linear model	DBPS ($K = 4,000$)	DBPS ($K = 2,000$)
$\beta_{0,\text{NDVI}}$	34.495 (34.392, 34.601)	1.670 (-0.570, 4.159)	1.814 (-0.787, 4.312)
$\beta_{1,\text{NDVI}}$	-2.708 (-2.719, -2.697)	0.716 (0.453, 0.955)	0.700 (0.431, 0.979)
$\beta_{0,\text{RR}}$	-16.924 (-17.014, -16.837)	-0.586 (-2.512, 1.498)	-0.633 (-2.408, 1.407)
$\beta_{1,\text{RR}}$	2.664 (2.655, 2.674)	0.962 (0.739, 1.162)	0.966 (0.749, 1.156)
Σ_{NDVI}	0.221 (0.220, 0.221)	0.182 (0.133, 0.245)	0.147 (0.114, 0.211)
$\Sigma_{\text{NDVI,RR}}$	-0.167 (-0.168, -0.166)	-0.116 (-0.167, -0.081)	-0.093 (-0.132, -0.072)
Σ_{RR}	0.155 (0.154, 0.155)	0.118 (0.082, 0.165)	0.093 (0.073, 0.134)

Table 2.3: Vegetation Index data analysis parameter estimates for candidate models. Parameter posterior summary 50 (2.5, 97.5) percentiles.

spatial data analysis: the LMC, the NNGP, and BART models. We were not able to fit any of them to our multivariate data because they exceeded memory. Notwithstanding, the NNGP is the gold standard in large-scale analysis, the restricted computational resources we use to fit the dataset do not allow us to allocate the output. This highlights a main advantage of using DBPS when working on limited computational frameworks. Table 2.3 compares model parameters posterior estimates for DBPS with $K \in \{4,000\ ,\ 2,000\}$, and the Bayesian conjugate linear model, which does not take into account spatial variability. The notably higher magnitudes of the intercepts in the non-spatial linear model are unsurprising as the spatial random effects absorb much of the variation in the intercepts. The solar zenith angle is positively associated with RR and negatively associated with NDVI in the non-spatial model. This, too, is expected since higher levels of solar irradiation are associated with higher red reflectance and with more arid regions with less vegetation. However, the spatial models reveal that the slope for NDVI is significantly positive with solar irradiation after the spatial effects have absorbed previously unaccounted latent or lurking factors that might have contributed to the negative slopes in the non-spatial model.

In addition to DOUBLE BPS (DBPS), LMC, NNGP, and Bayesian multivariate linear regression, we expanded the analysis by including four competitive algorithms: distributed random forest (DRF), gradient boosting (GBM), deep neural network (DNN), and a fully automatic machine learning algorithm (AUTOML). As detailed in Section 2.3, we implement and perform the analysis using the `h2o` R package (Fryda et al., 2024). As no method for multivariate models is available

Model	Time (min)	RMSPE	$\varrho_{\text{NDVI,RR}}$
DBPS (K = 2,000)	81	[0.208, 0.165, 0.187]	-0.915 (-0.929, -0.894)
DBPS (K = 4,000)	21	[0.230, 0.186, 0.208]	-0.910 (-0.931, -0.872)
NNGP (m = 5)	–	[–, –, –]	– (–, –)
LMC	–	[–, –, –]	– (–, –)
BART	–	[–, –, –]	– (–, –)
GBM	2	[0.419, 0.352, 0.385]	– (–,–)
DRF	10	[0.420, 0.353, 0.386]	– (–,–)
DNN	26	[0.422, 0.354, 0.388]	– (–,–)
AUTOML	44	[0.419, 0.352, 0.385]	– (–,–)
Conjugate Linear Model	12	[0.474, 0.404, 0.439]	-0.921 (-0.928, -0.915)

Table 2.4: Vegetation Index data analysis computing time in minutes, RMSPE, and empirical correlation (ϱ) for candidate models. Root mean square prediction error(s) presentation [NDVI, red reflectance, average].

within the `h2o` framework, we fit these algorithms separately for NDVI, and RR, considering the same explanatory variables used for DBPS and other Bayesian models. Then, we compute the empirical correlation among the predictions. In Table 2.4, we offer a comparison of computational costs, predictive performances, and empirical correlation. The DOUBLE BPS dominates both running time and predictive performances. It achieves an average RMSPE of 0.187 when $K = 2,000$, and 0.208 with $K = 4,000$ subsets for NDVI and RR. This is more than 100% lower than the standard Bayesian conjugate multivariate linear model, which produces average RMSPE of 0.439 for the two vegetation indexes. Actually, the DOUBLE BPS outperformed all the other competitive algorithms in terms of RMSPE. This is attained at a fraction of the computational time for other methods, let alone DOUBLE BPS models multivariate distributions jointly, and offer full explainability. Estimates of ϱ_{NDVI} and ϱ_{RR} reveal a well-documented negative association between the two responses. Indeed, the spatial patterns in these indices are almost the reverse of each other as revealed in Figures 2.9 and 2.10. The conjugate Bayesian linear model estimates a higher mode for negative correlation with respect to DBPS, even though the 95% credible intervals remain comparable. However, DBPS tends to lower negative correlation strength as n_k decreases.

Recalling that all the analyses are produced on a personal laptop (with just 5 physical cores) with minimal human intervention, the total run time of only 21 minutes with $K = 4,000$ and 81 minutes with $K = 2,000$ for DOUBLE BPS are impressive and confirm the quadratic dependence of the partition size discussed in Section 2.4.5. Moreover, the strong dependence on the number of J competitive models assessed is worth noting. Here $J = 4$, and this suggests a marginal computational burden for each competitive model.

Chapter 3

Bayesian Transfer Learning for Spatiotemporal Large-Scale Problems

The second contribution focuses on dynamic spatiotemporal systems, where online updating of both spatial and temporal dependence is essential. Building on matrix-variate Gaussian distribution theory, dynamic linear models, and Bayesian predictive stacking, the proposed approach facilitates efficient sharing of information across temporal shards while preserving spatial coherence. A Markovian dependence framework supports scalable modeling of complex multivariate dynamics, and exact inference is retained through predictive stacking combined with variational, sequential, and parallel updating schemes. Methodologically, the work advances Bayesian transfer learning by introducing dynamic Bayesian predictive stacking (DYNBPS), able to assimilate models accounting for inherited temporal dependence. This allows learning temporal and spatial dependence among long-massive datasets to be propagated and combined without reliance on expensive hardware or manual intervention. Almost-automated processing of shards provides conjugate inference at scale, transforming traditionally intractable problems into feasible analyses without reliance on high-performance computing.

3.1 Variational propagation

Bayesian predictive stacking (Yao et al., 2018) assimilates models using weighted distribution in the convex hull of individual posteriors, maximizing a score rule between the true predictive distribution and the weighted posterior (Gneiting and Raftery, 2007; Yao et al., 2018). Some applications already show great results of BPS for spatial data analysis (Zhang and Banerjee, 2022; Presicce and Banerjee, 2024; Pan et al., 2025, and Chapter 2). Here, we aim to assimilate models for different couples $\{\alpha, \phi\}$ using predictive stacking, to retain the full conjugacy of the model in Equation (1.13) at any time point. We specify a set of J competitive models $\mathscr{M}_j$ for $j = 1, \ldots, J$, characterized by couples $\{\alpha_j, \phi_j\}$. For each $t \in \mathcal{T}$, we compute the stacking

Algorithm 4 Computing stacking weights at time t using dynamic-BPS

Input: n, q, p: Number of observed rows, number of prediction points, number of outcomes, and number of predictors; J: number of competitive models in each subset, and number of predictive samples; $\{\tilde{m}_{t-1}, \tilde{C}_{t-1}, \tilde{\Psi}_{t-1}, \tilde{\nu}_{t-1}\}$: KL-minimizer posterior parameters; $\{\{\alpha^{(j)}, \phi^{(j)}\} : j = 1, \ldots, J\}$: Parameter associated with J models.
Output: $\hat{w}_t = \{\hat{w}_{t,j} : j \in \{1, \ldots, J\}\}$: Stacking weights within time shard.

for $j = 1, \ldots, J$ **do**
  for $i = 1, \ldots, n$ **do**
    Compute $q_{t,i}^{(j)}, Q_{t,i}^{(j)}$ using $\{\tilde{m}_{t-1}, \tilde{C}_{t-1}, \tilde{\Psi}_{t-1}, \tilde{\nu}_{t-1}, \alpha^{(j)}, \phi^{(j)}\}$, and $F_{t,i}$ as the i-th row of F_t
    Compute $pd_{t,j}^{(i)} = \mathrm{T}_{\tilde{\nu}_t}(Y_{t,i} \mid q_{t,i}^{(j)}, Q_{t,i}^{(j)}, \tilde{\Psi}_{t-1}, \tilde{\nu}_{t-1})$
  end for
end for
Solve $\hat{w}_t = \underset{w_t \in \mathcal{S}_1^J}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} \log \sum_{j=1}^{J} w_t \; pd_{t,j}^{(i)}$
with $\mathcal{S}_1^J$ simplex of dimension J
return $\hat{w}_t = \{\hat{w}_{t,j} : j \in \{1, \ldots, J\}\}$

weights $\hat{w}_t$ as

$$\hat{w}_t = (\hat{w}_{t,1}, \ldots, \hat{w}_{t,J})^{\mathrm{T}} = \underset{w_t \in \mathcal{S}_1^J}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} \log \sum_{j=1}^{J} w_{t,j} \; p(Y_{t,i} \mid Y_{1:t-1}, \mathscr{M}_j), \tag{3.1}$$

where $Y_{t,i}$ is the i-th row of Y_t, $p(\cdot \mid Y_{1:t-1}, \mathscr{M}_j)$ is the one-step ahead predictive distribution given on $\mathscr{M}_j$, and $\mathcal{S}_1^J$ is the simplex of dimension J. The marginal predictive density is available in closed form as a matrix t distribution, which makes the computation efficient. The derivation follows contents outlined in Section A.1.1, starting from $p(\cdot \mid Y_{1:t-1}, \Sigma, \mathscr{M}_j)$, available as a matrix-variate distribution from the forward filtering procedure (Carter and Kohn, 1994; West and Harrison, 1997), integrating out $p(\Sigma \mid Y_{1:t-1}, \mathscr{M}_j)$, also accessible through FFBS. Once we derive the set of dynamic weights $\hat{w}_t$, posterior inference at any $t \in \mathcal{T}$ follows straightforwardly using the stacked posterior distributions:

$$\hat{p}(\cdot \mid Y_{1:t}) = \sum_{j=1}^{J} \hat{w}_{t,j} \; \hat{p}(\cdot \mid Y_{1:t}, \mathscr{M}_j). \tag{3.2}$$

A substantial problem arises when attempting to borrow information throughout FFBS dynamics between data shards. The resulting posterior distributions come out as finite mixtures when using Bayesian predictive stacking. This will break the FFBS machinery. Unluckily, stacked posteriors in Equation (3.2) do not allow conjugate posterior-to-prior update, breaking online learning machinery. Intuitively, a solution relies on approximating the stacked posterior with a distribution that retains the conjugacy of the FFBS algorithm. We opt for a common approach, widely used when the posterior distribution appears to be intractable. We propose a variational approach: choose a new tractable posterior that minimizes the Kullback-Leibler (KL) divergence w.r.t. the stacked posterior. Here, intractable posterior distributions are finite

Algorithm 5 Compute KL-approximate posterior distribution parameters

Input: $\hat{w}_t = \{\hat{w}_{t,j}\} : j \in \{1, \ldots, J\}\}$: Stacking weights within time shard; n, q, p: Number of observed rows, number of outcomes, and number of predictors; J: number of competitive models in each subset; $\{\{m_t^{(j)}, C_t^{(j)}, \Psi_t^{(j)}, \nu_t^{(j)}, \alpha^{(j)}, \phi^{(j)}\} : t = 1, \ldots, T; j = 1, \ldots, J\}$: Set of parameters for each time point, and model.
Output: $\{\tilde{m}_t, \tilde{C}_t, \tilde{\nu}_t, \tilde{\Psi}_t\}$: KL-minimizer posterior parameters.

1: **for** $j = 1, \ldots, J$ **do**
2: Compute $\hat{\Sigma}^{(j)} = \mathbb{E}[\Sigma \mid Y_{1:t}, \mathscr{M}_j] = \Psi_t^{(j)}(\nu_t^{(j)} - q - 1)^{-1}$
3: **end for**
4: Compute $\tilde{m}_t = \sum_{j=1}^{J} \hat{w}_{t,j}\, m_t^{(j)}$
5: Compute $\tilde{C}_t = \sum_{j=1}^{J} \hat{w}_{t,j}\, [C_t^{(j)} + (m_t^{(j)} - \tilde{m}_t)^{\mathrm{T}} \hat{\Sigma}^{(j)} (m_t^{(j)} - \tilde{m}_t)]$
6: Compute $\tilde{\nu}_t = \sum_{j=1}^{J} \hat{w}_{t,j}\, \nu_t^{(j)}$
7: Compute $\tilde{\Psi}_t = \tilde{\nu}_t [\sum_{j=1}^{J} \hat{w}_{t,j}\, \nu_t^{(j)} \Psi_t^{-1(j)}]^{-1}$
8: **return** $\{\tilde{m}_t, \tilde{C}_t, \tilde{\nu}_t, \tilde{\Psi}_t\}$

mixtures of matrix-normal inverse-Wishart densities, while matrix-normal inverse-Wishart act as tractable counterparts.

In general, for two real-valued measures P, Q, the KL divergence is formulated through the density functions (assuming their existence), respectively $p(\cdot), q(\cdot)$

$$\mathrm{KL}(P \mid\mid Q) = \int_{\mathbb{R}} \log\left(\frac{p(x)}{q(x)}\right) p(x) dx. \tag{3.3}$$

We proceed by separating the computation between the two parameters Θ_t, Σ, and starting on the conditional posterior distribution for Θ_t. This is possible since the joint distribution of Θ_t and Σ admits a closed-form factorization as $p(\Theta_t, \Sigma) = p(\Theta_t \mid \Sigma)\, p(\Sigma)$, with both components being analytically available. Let us consider $P(\Theta_t) = \sum_{j=1}^{J} \hat{w}_j \,\mathrm{MN}(\Theta_t \mid m_t^{(j)}, C_t^{(j)}, \Sigma)$ to be the stacked posterior we want to approximate, which is a finite mixture of matrix-variate Gaussians, and $Q(\Theta_t) = \mathrm{MN}(\Theta_t \mid \tilde{m}_t, \tilde{C}_t, \Sigma)$ to be a single matrix-variate normal measure (representing the matrix normal distribution which borrows information between time instants). Later on, we fix the time instant at t (without loss of generality for any $t = 1, \ldots, T$), and we consider Σ known for $j = 1, \ldots, J$. In Section B.1.1, we compute the $\mathrm{KL}(P(\Theta_t) \mid\mid Q(\Theta_t))$ by its definition in Equation (3.3). Now consider minimizing the KL divergence obtained in Equation (B.10) in B.1.1 with regard to the parameters of $Q(\Theta_t)$, i.e. $\tilde{m}_t, \tilde{C}_t$ (since Σ is known). We seek the matrix-variate Gaussian distribution that minimizes KL divergence with $P(\Theta_t)$, which is fully determined by matching the first two moments

$$\underset{\tilde{m}_t \in \mathbb{R}^{n \times q}, \tilde{C}_t > 0 \in \mathbb{R}^{n \times n}}{\arg\min} \; \mathrm{KL}\Big(\sum_{j=1}^{J} \hat{w}_j \,\mathrm{MN}(\Theta_t \mid m_t^{(j)}, C_t^{(j)}, \Sigma) \Big\| \mathrm{MN}(\Theta_t \mid \tilde{m}_t, \tilde{C}_t, \Sigma) \Big). \tag{3.4}$$

In Section B.1.2, we obtain a closed-form solution for the optimization problem stated in Equation 3.4. Hence, from results in Equations (B.13),(B.16) of Section B.1.2, the matrix normal distribution which minimizes the KL divergence from a finite mixture of matrix normal

distributions is the one, fully characterized, by the following optimal parameters:

$$\tilde{m}_t = \sum_{j=1}^{J} \hat{w}_j m_t^{(j)}, \quad \tilde{C}_t = \sum_{j=1}^{J} \hat{w}_j \left[C_t^{(j)} + (m_t^{(j)} - \tilde{m}_t)^\intercal \Sigma^{-1} (m_t^{(j)} - \tilde{m}_t) \right], \tag{3.5}$$

conditionally to a known common column covariance matrix Σ, for $j = 1, \dots, J$. This leads to the following approximated posterior distribution $\hat{p}_{KL}(\Theta_t \mid Y_{1:t}, \Sigma) = \text{MN}(\Theta_t \mid \tilde{m}_t, \tilde{C}_t, \Sigma)$ with $\tilde{m}_t$, and $\tilde{C}_t$ defined in (3.5).

Then, we present an approximation for the posterior distribution of Σ, based on an equivalent reasoning made for Θ_t. Where $P(\Sigma) = \sum_{j=1}^{J} \hat{w}_j \, \text{IW}(\Sigma \mid \nu_t^{(j)}, \Psi_t^{(j)})$, and $Q(\Sigma) = \text{IW}(\Sigma \mid \tilde{\nu}_t, \tilde{\Psi}_t)$. In Section B.1.3, we derive a formulation for $\text{KL}(P(\Sigma) \mid\mid Q(\Sigma))$. Let us now consider optimizing the following problem, looking to characterize the inverse Wishart distribution which minimizes $\text{KL}(P(\Sigma) \mid\mid Q(\Sigma))$. We are now looking for

$$\underset{\tilde{\nu}_t \in \mathbb{R}^+, \tilde{\Psi}_t > 0 \in \mathbb{R}^{m \times m}}{\arg\min} \; \text{KL}\Big(\sum_{j=1}^{J} \hat{w}_j \, \text{IW}(\Sigma \mid \nu_t^{(j)}, \Psi_t^{(j)}) \Big\| \, \text{IW}(\Sigma \mid \tilde{\nu}_t, \tilde{\Psi}_t) \Big). \tag{3.6}$$

We obtain a closed-form for the optimal scale matrix Ψ_t in Section B.1.4. However, when we pass to ν_t, no closed-form solutions are available due to the presence of the parameter within the argument of multiple digamma functions. This makes a direct computation of the optimal parameter unavailable, but implementing a numerical optimizer or a root finder for the derivative (which exists analytically and is presented in Section B.1.4) is still feasible. Nevertheless, alternative paths were investigated. In particular, commonly used approximations of the involved special function can be adopted, leading to simpler formulations for the optimization problem.

Actually, the degree of freedom across models must be the same, since the Bayesian updating for each $\nu_t^{(j)}$ depends only on the number of observations, and models $\mathscr{M}_j$ have no influence. Therefore, a logical choice is to set $\tilde{\nu}_t = \sum_j^J \hat{w}_j \nu_t^{(j)}$, which removes the need for a numerical optimization procedure to recover $\hat{\nu}_t$. Since $\nu_t^{(j)} = \nu_{t-1} + n/2$ for all $j = 1, \dots, J$. This reduce the approximated posterior distribution for Σ to $\hat{p}_{KL}(\Sigma \mid Y_{1:t}) = \text{IW}(\Sigma \mid \tilde{\nu}_t, \tilde{\Psi}_t)$, with $\tilde{\nu}_t = \sum_j^J \hat{w}_j \nu_t^{(j)}$, and $\tilde{\Psi}_t = \tilde{\nu}_t \left[\sum_{j=1}^{J} \hat{w}_j \nu_t^{(j)} \Psi_t^{-1(j)} \right]^{-1}$ (see Section B.1.4 for further details).

In Algorithm (5), we provide a practical procedure to compute $\tilde{m}_t, \tilde{C}_t, \tilde{\Psi}_t, \tilde{\nu}_t$ for $t = 1, \dots, T$. Thus, we found a solution that allows us to average out nuisance spatial parameters, and this was obtained by merging instruments from BPS framework with concepts from variational inference. It is worth noting that the KL approximations of the posterior distribution at time t do not affect inference within the same data shard at time t, but serve solely as a prior for subsequent data shards. In practice, approximated posterior distributions are only used to allow the information propagation across time, but inferences are pursued by a stacked posterior. This permits us to take advantage of the finite mixture formulation of stacked posterior gained from BPS, making it possible to fit more complex distributions, for prediction and uncertainty quantification. Figure 1.2 provides a graphical representation of the entire procedure within each "cell".

3.1.1 Computational details

Section 3.1 proposed to improve the dynamic spatiotemporal propagation of Bayesian updating in dynamic linear models by using variational approximations. Here, we present the computational details regarding the algorithms we developed to provide online spatiotemporal learning for massive georeferenced historical data with variational propagation.

Standards Bayesian inference for DLMs proceed with the well-known forward filter backward sampling, where possible. Indeed, conjugacy is generally not available for all parameters. In spatial and spatiotemporal settings, spatial range parameter(s) introduce complications. Circumnavigating this problem requires the implementation of simulation-based approaches, e.g. MCMC, to perform sampling from posterior distributions, making conjugacy only exploitable as full conditional analysis in Metropolis-within-Gibbs frameworks. In section 3.1, we propose a mixed strategy based on Bayesian predictive stacking of predictive distributions. Notwithstanding several advantages of including Bayesian predictive stacking, it also introduces drawbacks. The main limitations come from the mixture-form structure of stacked posterior distributions. Even though it can be seen as a benefit generally, it is quite "destructive" for DLMs since the stacked posterior breaks down posterior-to-prior update, which allows borrowing of information through time instants of the forward filter backward sampling algorithm. To be precise, the stacked posteriors when used as priors for the next time point, as a mixture distribution, do not preserve conjugacy for DLMs. To this matter, we find a solution that permits the introduction of BPS without renouncing to full conjugacy. This was detailed in Section 3.1, while now we present how to apply the solution in practice. A series of algorithms replacing standard procedures for DLMs will be introduced and motivated in this Section, having special consideration for the spatiotemporal modeling.

Algorithm (8) presents the modification to the forward filtering (FF) procedure within FFBS algorithm for each time instant $t = 1, \dots, T$. In particular, it incorporates Algorithm (4), and Algorithm (5), making its construction different from standard FF, but retaining most of the original structure. The first step in Algorithm (8) uses Algorithm (5) to approximate the information from preceding stacked posterior exploiting KL divergences computed in Appendix B.1. Once an approximated prior information from time $t-1$ is obtained under a suitable form, the Algorithm (8) proceeds as usual for DLMs. Indeed, we derive filtered prior, filtered posterior, and one-step ahead predictive distributions from Equations (1.7)-(1.11), using the new approximated parameters $\{\tilde{m}_t, \tilde{C}_t, \tilde{\nu}_t, \tilde{\Psi}_t\}$. This is repeated for each competing model $\mathscr{M}_j$, with $j = 1, \dots, J$. Then, we recover the set of dynamic weights at time t by Algorithm (4), computing expected logarithm predictive density under each model $\mathscr{M}_j$. Thus, posterior inference occurs using staked posterior distributions defined in Equation (3.10). To achieve forward filtering, the procedure in Algorithm (8) is repeated for $t = 1, \dots, T$. At starting point $t = 1$, we fix $\hat{w}_{0,j} = 1/J \quad \forall j = 1, \dots, J$ implying that $\{\tilde{m}_t, \tilde{C}_t, \tilde{\nu}_t, \tilde{\Psi}_t\} = \{m_0, C_0, \nu_0, \Psi_0\}$, i.e. prior information are equally used across models $\mathscr{M}_j$. To fully incorporate the new piece of information, as the latest dataset occurs at $T+1$, it suffices to apply Algorithm (8) only, followed by Algorithm (10). This allows us to gain an advantage from the Markovian dependence assumption across datasets, and reduce the computational effort without the necessity to reconsider the full collection of T datasets. In the FFBS algorithm, after the forward filtering step, the backward

sampling (BS) step is applied to smooth posterior distributions by propagating information from the final observed time point T backward to each state $t = T-1, \ldots, 1$. This procedure incorporates future observations into earlier latent states, yielding smoothed posterior samples and improving inferential accuracy beyond filtering alone. Here, to achieve BS into the framework we are building, we propose the Algorithm (10). For each desired posterior draw $r = 1, \ldots, R$, we sample one of the competitive model $\mathcal{M}_j$, with $j = 1, \ldots, J$, from which draw the posterior sample from correspondent conditional posterior for Θ_T, i.e. $p(\Theta_T \mid Y_{1:T}, \mathcal{M}_j)$, by using weights $\hat{w}_T$. In doing so, we obtain a sample from $\hat{p}(\Theta_T \mid Y_{1:T}) = \sum_{j=1}^{J} \hat{w}_{T,j}\, p(\Theta_T \mid Y_{1:T}, \mathcal{M}_j)$. Then, still for $r = 1, \ldots, R$, we use BS algorithm letting vary $t = T-1, \ldots, 1$, and repeating the model "sampling", and using the correspondent weights $\hat{w}_t$, computed for all t in Algorithm (8). At the end of the procedure depicted in Algorithm (10), we obtain a smoothed posterior sample of size R. In particular, with Algorithm (10), we generate samples from the stacked joint posterior, whose density admits the factorization

$$\hat{p}(\Theta_0, \ldots, \Theta_T \mid Y_{1:T}) = \hat{p}(\Theta_0 \mid \Theta_1, Y_{1:T})\hat{p}(\Theta_1 \mid \Theta_2, Y_{1:T}) \cdots \hat{p}(\Theta_{T-1} \mid \Theta_T, Y_{1:T})\hat{p}(\Theta_T \mid Y_{1:T}), \quad (3.7)$$

so that backward sampling yields draws distributed according to (3.7). Each conditional distribution is given by $\hat{p}(\Theta_t \mid \Theta_{t+1}, Y_{1:T}) = \sum_{j=1}^{J} \hat{w}_{t,j}\ \hat{p}(\Theta_t \mid \Theta_{t+1}, Y_{1:t}, \mathcal{M}_j)$. Together with Algorithm (8), this concludes the Bayesian estimation for DLMS, from which posterior inference descends. However, in spatiotemporal settings, two crucial targets still remain to be investigated. Indeed, after we conclude the posterior inference on parameters, forecasting for the outcomes is usually imperative, and here is a two-way problem, since we can perform temporal forecasting and spatial predictions separately. Let us start by managing temporal forecasting for the time series involved.

When working with time series, we often aim to forecast the future value of the outcome. This is a well-defined problem and, especially for DLMS, is quite trivial. In Section 3.1, we derive the posterior predictive distribution k-step ahead, valid for a generic forecasting horizon k. Thus, based on all the information available, i.e., up to T, we recursively obtain the distribution of the future outcome after k time instants, leading to an out-of-sample posterior inference. The same can be made in-sample, meaning one computes the posterior predictive for each instant t observed, for $t = 1, \ldots, T$. As belonging to the state-space models class, forecasting for DLMS is carried out similarly to ARIMA models. Algorithm (11) casts the k-step ahead posterior predictive distribution detailed in Section 3.1 into the novel framework we introduce in this manuscript. In particular, Algorithm (11) introduces the Bayesian predictive stacking in the recursive predictive procedure. Following the same logic as the beforehand algorithms, for each posterior predictive sample $r = 1, \ldots, R$ we sample a model $\mathcal{M}_j$ based upon $\hat{w}_T$, i.e., the last set of stacking weights. Then, standard recursive steps for prediction for Bayesian state-space models occur. Concluding the Algorithm (11), we obtain a stacked posterior predictive sample from $\hat{p}(\tilde{\Theta}_{T+k}, \tilde{Y}_{T+k} \mid Y_{1:T}) = \sum_{j=1}^{J} \hat{w}_{t,j}\ p(\tilde{\Theta}_{T+k}, \tilde{Y}_{T+k} \mid Y_{1:T}, \mathcal{M}_j)$ of size R. The recursive part in Algorithm (11) resides in the inner loop. As the one-step ahead predictive depends upon the last observed time, the k-step ahead predictive depends upon the $k-1$-step ahead leading to the need to compute all the previous $k-1$ predictive distributions, to obtain the k-step ahead uncertainty quantification. The computational burden grows with the length of

the forecast horizon k. Moreover, the prediction will worsen, leading to an increasing diffusion for the posterior predictive distribution. This retraces the prediction behavior for standard ARIMA, where the posterior mean move forward to the global mean (known as mean-reverting property), and the variance linearly depend on the length of the forecast horizon.

Along with temporal forecasting, spatial interpolation helps improve decision-making and problem-understanding when spatial components play a fundamental role, e.g., environmental and climate sciences, medical screenings, etc. Spatial interpolation, also called spatial prediction, is a crucial step in spatial and spatio-temporal settings. It allows us to infer spatial process characteristics and its spatial distribution over unobserved locations, and possibly at any point within the region of interest. Letting $\mathcal{U} = \{u_1, \ldots, u_m\}$ be the set of m unmonitored locations as specified in Section 3.1. In Equation (1.15) we derive the joint posterior predictive for the response $\tilde{Y}_t$, and the spatial process $\tilde{\Omega}_t$ for these points, at any generic time point $t = 1, \ldots, T$. The Algorithm (12) provides a practical procedure that incorporates Bayesian predictive stacking into spatial prediction to draw posterior predictive samples from the stacked predictive distribution $\hat{p}(\tilde{Y}_t, \tilde{\Omega}_t \mid Y_{1:t}) = \sum_{j=1}^{J} \hat{w}_{t,j}\ p(\tilde{Y}_t, \tilde{\Omega}_t \mid Y_{1:t}, \mathcal{M}_j)$, at any $t \in \mathcal{T}$. Specifically, in Algorithm (12) we start sampling one across the competitive models $\mathcal{M}_j$ using weights $\hat{w}_t$. Then we compute the parameters of the corresponding conditional predictive posterior distribution $p(\tilde{Y}_t, \tilde{\Omega}_t \mid Y_{1:t}, \mathcal{M}_j)$, and draw a predictive sample. This must be repeated for the number R of desired predictive samples, as with Algorithm (11). After completing the sampling procedure, we can perform uncertainty quantification for the outcomes and the spatial processes across the entire region of interest, perhaps providing useful graphical maps of their spatial distributions.

3.2 Parallel propagation

Section 1.3 described standard matrix-variate dynamic linear models to incorporate spatiotemporal problems. Similar approaches were already investigated, especially for complex data analysis (see e.g. Jiménez and Pereira, 2021). However, to the best of our knowledge, no previous contribution avoids simulation-based methodology in estimating dynamic spatiotemporal models with FFBS, while retaining full conjugacy, including spatial parameters. This pretends to be achieved thanks to the pillar of this novelty: a dynamic adaptation of Bayesian predictive stacking (BPS), supported by computational stratagems.

Since it was introduced, Bayesian predictive stacking (Yao et al., 2018) provides a widely applicable and effective alternative to Bayesian model averaging focused on predictive criteria. We aim to apply BPS for spatiotemporal problems. In doing so, we have to implement a "dynamic" variation for predictive stacking, to take advantage of the predictive structure which DLMS intrinsically offer. We introduce predictive stacking to retain the full conjugacy of the model in Equation (1.13). Indeed, fixing the couple $\{\alpha, \phi\}$, the model in (1.13) could be easily estimated through the FFBS algorithm, without accessing simulation-based approaches (e.g. MCMC). How can we effectively fix them? Bayesian stacking is perfectly tailored to address this task. The methodology is remarkably straightforward: specify a set of J competitive models, each defined by the pair $\{\alpha_j, \phi_j\}$, and let the procedure do the heavy lifting by assigning relative weights to determine the most effective models. Then provide stacked

posterior inference over these models. Unfortunately, BPS was not originally thought to face temporal dependencies. Gathering ideas from Paul-Christian Bürkner and Vehtari (2020); Vehtari et al. (2016, 2017), we introduce leave-future-out (LFO), instead of vanilla leave-one-out cross-validation, into Bayesian predictive stacking (we refer the reader to Ruiz Maraggi et al., 2021; Cooper et al., 2023; Kennedy et al., 2024, for examples of leave-future-out and similar approaches that account for dependence in validations). This modification enables us to more accurately account for temporal dependencies within observations. In addition, let us recall the structure of model (1.6), or (1.13) equivalently, and its estimation. In particular, at each time instant t, one-step ahead predictive distributions are computed by default within the FFBS procedure, see Equation (1.8). We notice these distributions exactly match the one-step ahead predictive distribution needed for LFO cross-validation. Indeed, leave-future-out cross-validation focuses on the predictive evaluation of the next future observation given the past. Figure 3.1 gives a sketch of its operating principle.

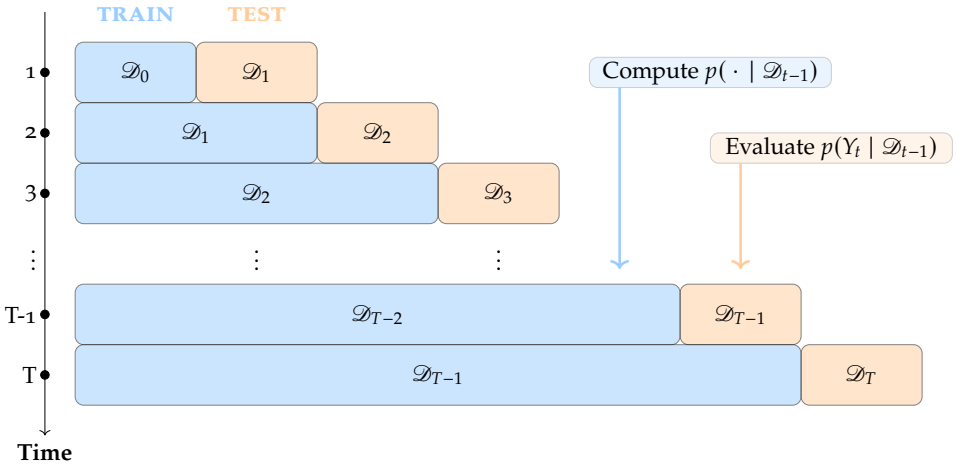

Figure 3.1: Leave-future-out cross-validation

Bayesian predictive stacking attempts to find the distribution in the convex hull $\mathcal{C} = \{\sum_{j=1}^{J} w_j p(\cdot \mid \mathscr{M}_j) : \sum_{j=1}^{J} w_j = 1, w_j \geq 0\}$ of individual posterior distributions which maximize a proper scoring rule from the true predictive distribution (Gneiting and Raftery, 2007; Yao et al., 2018). While the true predictive distribution is unknown, they use a leave-one-out estimate of the expected value of the score (see Yao et al., 2018, Section 3.1). Transposing the same reasoning, we replace the predictive distribution with the one-step ahead predictive distribution, as we consider it more functional to assess model predictive capabilities in time-dependent contexts. In Section 1.4.2, we provide further details on deriving the optimization problem for dynamic predictive stacking. Dynamic stacking weights at time τ are then computed by solving this optimization problem

$$\hat{w}_i^{(\tau)} = \underset{w_i^{(\tau)} \in \mathcal{S}_1^J}{\operatorname{argmax}} \frac{1}{\tau - 1} \sum_{t=1}^{\tau-1} \log \sum_{j=1}^{J} w_{i,j}^{(\tau)} \, p(Y_{t+1,i} \mid \mathscr{D}_t, \mathscr{M}_j), \tag{3.8}$$

where $Y_{t,i}$ is the i-th row of Y_t, $p(Y_{t,i} \mid \mathscr{D}_{t-1}, \mathscr{M}_j)$ corresponds to the marginalization of one-step ahead predictive distribution in Equation (1.8) based on $\mathscr{M}_j$ fixed parameters, and $\mathcal{S}_1^J$ is the

simplex of dimension J. The marginal predictive density $p(Y_{t,i} \mid \mathscr{D}_{t-1}, \mathscr{M}_j)$, is available in closed form as a matrix t distribution, which makes the computation efficient. Leading Equation (3.8) to

$$\hat{w}_i^{(\tau)} = \underset{w_i^{(\tau)} \in \mathcal{S}_1^J}{\operatorname{argmax}} \frac{1}{\tau - 1} \sum_{t=1}^{\tau-1} \log \sum_{j=1}^{J} w_{i,j}^{(\tau)} \; \mathrm{T}_{\nu_t^{(j)}}(Y_{t+1,i} \mid q_{t+1,i}^{(j)}, Q_{t+1,i}^{(j)}, \Psi_t^{(j)}, \nu_t^{(j)}), \tag{3.9}$$

where $q_{t+1,i}^{(j)} = F_{t+1,i} A_{t+1}^{(j)}$, and $Q_{t+1,i} = F_{t+1,i} R_{t+1}^{(j)} F_{t+1,i} + V_{t+1}(\alpha_j)$; with $A_{t+1}^{(j)} = G_{t+1} m_t^{(j)}$, and $R_{t+1}^{(j)} = G_{t+1} C_t^{(j)} G_t^{\mathrm{T}} + W_t(\phi_j)$. As noticed, for each step t the parameters $\{m_{t-1}^{(j)}, C_{t-1}^{(j)}, \Psi_{t-1}^{(j)}, \nu_{t-1}^{(j)}\}$ must be inherited from the previous step $t-1$, for each model $\mathscr{M}_j$.

It is worth noting that we derive stacking weights for each location i in Equation (3.8), which corresponds to a single time series, and for each time $t = 2, \ldots, T$. This stems primarily from natural dynamic stacking weights in (3.8). We compare predictive performances across time instants and not across observations as vanilla Bayesian predictive stacking does. This originates from the difference existing between leave-one-out cross-validation and leave-future-out cross-validation (schematized in Figure 3.1). On the other hand, even if it is theoretically possible to design this way, the simultaneous evaluation across time and locations is infeasible. There are several reasons for this, starting from the curse of dimensionality. As the number of points increases, the density lies in a higher-dimensional space, where the notion of point rapidly loses sense, and its probability goes to zero even faster. Then, when the number of locations grows, the evaluation of the predictive density for the whole matrix Y_t results in numerical instabilities. Nevertheless, individual evaluations allow us to reach a finer granularity of stacking weights and possibly provide individual temporal inferences for each location. Individual stacking weights are useful instruments to extract insight into model competition in several areas of interest, as shown in Section 3.3 using simulated data.

However, performing posterior inference on the parameters using Algorithm (10) requires a unique set of stacking weights. Moreover, posterior inference on parameters should not depend on single locations, even though it is possible to do so. Let us make the reader aware that performing individual posterior inferences may turn into an excessively demanding computational burden, along with we use of global weights for this scope. Parallel computing may be helpful in this matter but requires high computational resources to lighten individual inferences over n locations, especially in big-n settings.

To retain computational admissibility over the locations, we propose two approaches to achieve stacking weights. We may opt for the average weights across all locations, i.e. $\{\hat{w}_{G,1}^{(\tau)}, \ldots, \hat{w}_{G,J}^{(\tau)}\} = \hat{w}_G^{(\tau)} = \frac{1}{n}\sum_{i=1}^n \hat{w}_i^{(\tau)} = \{\frac{1}{n}\sum_{i=1}^n \hat{w}_{i,1}^{(\tau)}, \ldots, \frac{1}{n}\sum_{i=1}^n \hat{w}_{i,J}^{(\tau)}\}$, referring to them as "global" weights, and indexed with the subscript G. Or we may consider the consensus across all locations, i.e. $\{\hat{w}_{C,1}^{(\tau)}, \ldots, \hat{w}_{C,J}^{(\tau)}\} = \hat{w}_C^{(\tau)} = \left\{\#\{i : \max\{\hat{w}_{i,1}^{\tau}, \ldots, \hat{w}_{i,J}^{\tau}\} = \hat{w}_{i,j}^{\tau}\} : \; j = 1, \ldots, J\right\}$, referring to them as "consensus" weights, and indexed with the subscript C. We provide an empirical investigation on the impact of this choice in Section 3.3.

After we derive the set of dynamic weights at time τ, $\hat{w}_\tau$, posterior inference follow straight-

forwardly using the stacked posterior distributions:

$$\hat{p}(\cdot \mid \mathscr{D}_\tau) = \sum_{j=1}^{J} \hat{w}_j^{(\tau)} \, p(\cdot \mid \mathscr{D}_\tau, \mathscr{M}_j). \tag{3.10}$$

The computational procedure needed to compute the individual stacking weights within each time shard is depicted in Algorithm (6). We take advantage of parallel computing over the competitive model evaluation, as the number J of competitive models in most applications allows a light implementation even in modest computational environments.

As mentioned in Section 3.1, a substantial problem arises when borrowing temporal posterior-to-prior information throughout FFBS using the content presented so far. Indeed, also dynamic predictive stacking results in posterior distributions defined as finite mixtures, see Equation (3.10). This could break the FFBS machinery, i.e., its conjugacy across previous instant posteriors acting as priors for the next one; a solution to mutual information must be found. Possibly looking at something different from variational approximations presented in Section 3.1, as we aim to retain dynamics introduced with DYNBPS. Intuitively, a solution relies on assimilating a parallel learning flow for each model. So that stacked distribution retains reliability for inference in each generic dataset $\mathscr{D}_\tau$, but information to the incoming dataset $\mathscr{D}_{\tau+1}$ only passes across J parallel learning paths. Then, posterior-to-prior distribution retains the conjugacy of FFBS algorithm, as we do not use stacked posterior $\hat{p}(\cdot \mid \mathscr{D}_\tau)$ but conditional posterior instead, i.e. $p(\cdot \mid \mathscr{D}_\tau, \mathscr{M}_j)$ for $j = 1, \ldots, J$. Hence, these computational adaptations allow us to preserve the conjugacy in the FFBS algorithm while borrowing information between time instants.

Thus, we found a solution that allows us to average out nuisance spatial parameters, and this was obtained by adapting instruments from BPS into a dynamical environment. It is worth specifying that the conditional posterior distributions do not affect the inference within each data shard at a specific time instant. In practice, conditional posterior distributions are only used to allow the information propagation across time, but inferences are pursued by stacked posteriors. This permits us to take advantage of the finite mixture formulation of stacked posterior gained from BPS, making it possible to fit more complex distributions, for prediction and uncertainty quantification. Figure 1.3 provides a graphical representation of the entire procedure within each "cell".

3.2.1 Computational details

Section 3.2 introduced the dynamic Bayesian predictive stacking (DYNBPS) framework, which provides scalable advancements for Bayesian estimations of dynamic spatiotemporal models, improving Markovian propagation for large-scale problems. We have developed an efficient and effective approach to managing online spatiotemporal learning for massive georeferenced historical data.

In this Section, we aim to describe strategies that avoid simulation or iterative-based methodologies, looking for a full conjugate analysis for DSTMS. Vanilla implementations of Bayesian predictive stacking, let alone the advantages, also introduce drawbacks. The BPS posterior

Algorithm 6 Computing individual stacking weights at time τ using dynamic-BPS

Input: n, q, p: Number of observed rows, number of prediction points, number of outcomes, and number of predictors; J: number of competitive models in each subset, and number of predictive samples; $\{m_{\tau-1}^{(j)}, C_{\tau-1}^{(j)}, \Psi_{\tau-1}^{(j)}, \nu_{\tau-1}^{(j)} : j = 1, \ldots, J\}$: previous parameters for each model; $\{\{\alpha^{(j)}, \phi^{(j)}\} : j = 1, \ldots, J\}$: Parameter characterizing the J models; $\{pd_{i,j}^{(t)} : i = 1, \ldots, n\,;\; j = 1, \ldots, J\,;\; t = 1, \ldots, \tau - 1\}$: Previous predictive density evaluations.

Output: $\hat{w}_i^{(\tau)} = \{\hat{w}_{i,j}^{(\tau)} : j = 1, \ldots, J\}$: Individual stacking weights for time shard τ ($i = 1, \ldots, n$).

for $j = 1, \ldots, J$ **do Parallel**
  for $i = 1, \ldots, n$ **do**
    Compute $q_{\tau,i}^{(j)}, Q_{\tau,i}^{(j)}$ using $\{m_{\tau-1}, C_{\tau-1}, \Psi_{\tau-1}, \nu_{\tau-1}, \alpha^{(j)}, \phi^{(j)}\}$,
    Compute $pd_{i,j}^{(\tau)} = \mathrm{T}_{\nu_{\tau-1}}(Y_{\tau,i} \mid q_{\tau,i}^{(j)}, Q_{\tau,i}^{(j)}, \Psi_{\tau-1}, \nu_{\tau-1})$
  end for
end Parallel for
for $i = 1, \ldots, n$ **do**
  Solve $\hat{w}_i^{(\tau)} = \underset{w_i^{(\tau)} \in \mathcal{S}_1^J}{\operatorname{argmax}} \frac{1}{\tau-1} \sum_{t=1}^{\tau-1} \log \sum_{j=1}^{J} w_{i,j}^{(\tau)}\, pd_{i,j}^{(t+1)}$
end for

distributions are finite mixtures, then it breaks down posterior-to-prior updating, which allows borrowing of information through time instants in the forward filter backward sampling algorithm. The same happens for more sophisticated BPS-based modeling approaches (e.g., Presicce and Banerjee, 2024). A feasible solution, to retain FFBS algorithm structure available, should include temporal dynamics within the Bayesian predictive stacking while preserving conjugate posterior-to-prior updates. In section 3.2, we then propose a dynamic strategy based on Bayesian predictive stacking of predictive distributions. Algorithm (6) shows its implementation.

We now present the computational adjustment to the algorithms characterizing the FFBS procedure (Carter and Kohn, 1994). We tailored these having special consideration for spatiotemporal modeling, referring to Model (1.13).

The parallel propagation strategy we define in Section 3.1 shares most of the computational modifications we introduce in Section 3.1.1, and the algorithms detailed in Appendix B, specifically in Section B.2. Only the algorithm associated with the forward filtering procedure presented in Section 3.1.1, i.e., Algorithms 8, must be changed when passing from variational to parallel propagation.

Algorithm (9) presents the specific modification to forward filtering (FF) procedure within FFBS algorithm for each time instant $t = 1, \ldots, T$ when working with parallel learning flows. In particular, it incorporates Algorithm (6), instead of Algorithms (4), and (5), making its structure different from standard FF and Algorithm (8), actually retaining the founding structure. It exploits parallel computations, establishing J parallel forward filter flows. This means that the Algorithm (9) implements parallel filtering procedures, where each of these conditions to model $\mathscr{M}_j$, i.e., uses the couple $\{\alpha_j, \phi_j\}$, for $j = 1, \ldots, J$. Then, there is no need to compute KL approximation using Algorithm (8), as the J parallel paths push information forward exactly.

Once each j-th flow provides filtered posterior distributions, especially the one-step-ahead predictive distribution, we computed DYNBPS weights using Algorithm (6). Prior information is equally used across models $\mathscr{M}_j$. Then forward propagated for all the observed times. To fully incorporate the new piece of information, as the latest dataset occurs at $T+1$, it suffices to apply Algorithm (9) once, followed by Algorithm (10) to update posterior sampling with the newest info. This gains advantage from the Markovian dependence assumption between datasets.

The same reasoning we state in Section 3.1 holds true: we still conclude smoothing inference with weighted backward sampling Algorithm (10), where only the weights are derived differently. Then, similarly as in Section 3.1.1, to obtain each posterior draw $r = 1, \ldots, R$, we first sample a model $\mathscr{M}_j$, with $j = 1, \ldots, J$, by giving models weights $\hat{w}_T$ obtained at last observed time. Then, we draw the posterior sample from the corresponding conditional posterior for Θ_T, i.e. $p(\Theta_T \mid \mathscr{D}_T, \mathscr{M}_j)$. In doing so, we obtain a sample from the approximated stacked posterior $\hat{p}(\Theta_T \mid \mathscr{D}_T) = \sum_{j=1}^{J} \hat{w}_{T,j}\, p(\Theta_T \mid \mathscr{D}_T, \mathscr{M}_j)$. We then make use of standard BS algorithm, sampling the models $\mathscr{M}$ with weights $\hat{w}_t$, for $r = 1, \ldots, R$. We obtain a smoothed posterior distribution sample of size R after executing Algorithm (10). As mentioned in Section 3.1.1, we smooth out the samples from the approximated stacked joint posterior, corresponding again to

$$\hat{p}(\Theta_0, \ldots, \Theta_T \mid \mathscr{D}_T) = \hat{p}(\Theta_0 \mid \Theta_1, \mathscr{D}_T)\hat{p}(\Theta_1 \mid \Theta_2, \mathscr{D}_T) \cdots \hat{p}(\Theta_{T-1} \mid \Theta_T, \mathscr{D}_T)\hat{p}(\Theta_T \mid \mathscr{D}_T). \tag{3.11}$$

Each $\hat{p}(\Theta_t \mid \Theta_{t+1}, \mathscr{D}_T) = \sum_{j=1}^{J} \hat{w}_{t,j}\ \hat{p}(\Theta_t \mid \Theta_{t+1}, \mathscr{D}_T, \mathscr{M}_j)$. Together with Algorithm (9), this concludes the Bayesian computation needed to achieve posterior inference and uncertainty quantification for Model (1.13) by using parallel information propagation.

Concerning the forecast of future values, and providing uncertainty quantification for the outcomes and the spatial processes across the entire region of interest, the argument remains the same as at the end of Section 3.1.1. Then forecasting can still be achieved following Algorithm (11), while spatial interpolation can still be computed using Algorithm (12), implemented following the same procedure. Thus, we consider the variational and parallel propagation approaches extremely similar; they mostly differ in the derivation of the weighting. Even though there is a sensible difference, as DYNBPS include temporal dynamics.

3.3 Simulations experiment

We highlight the potential of dynamic Bayesian predictive stacking under different simulation experiments, while posing challenges and exploring features of multivariate spatiotemporal analysis. We also provide a comparison among variational and parallel information propagation strategies, including a discussion on the effect of dynamics when included in stacking weights. On this matter, we focus our simulation and data application on the parallel propagation approach, which brings in more premises on efficiency when working with large-scale problems, notwithstanding its better performance. The simulations and analysis were implemented using native R and C++ programming languages. We obtain the results by using just a standard laptop running an Intel Core i7-8750H CPU with 5 cores for parallel computation, with 16 GB of RAM; representing the efficiency of DYNBPS within scarce computational resources frameworks. Additional details on programs and Algorithms are explored in Section 3.2.1.

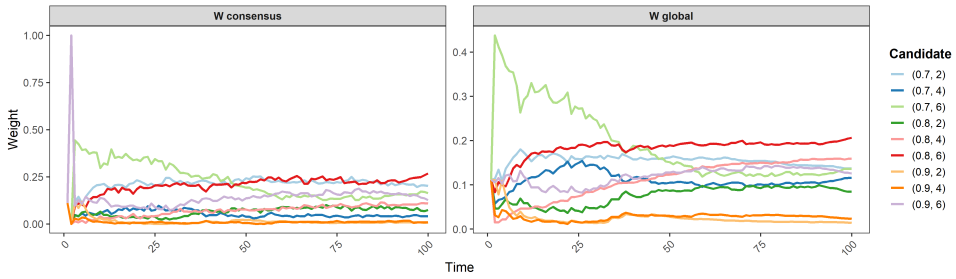

Figure 3.2: DYNBPS weights dynamic comparison

3.3.1 Space-time weights dynamics

In this simulation experiment, we investigated the different behavior for dynamic Bayesian predictive stacking weights. In Section 3.2, we introduce the concept of "global" weights (G), and "consensus" weights (C). Descending from the need to concentrate the local information on weights, which actually DYNBPS provides (see Section 3.2 for motivation and further technical details). Either global or consensus weight sets work as central tendency extrapolation from their spatial distributions, allowing FFBS algorithm implementation, as detailed in Section 3.2.1. In particular, they reflect information on the average weight and the "consensus" among locations, to each of the J candidate models, at any time point. Although they mainly serve to retain computational admissibility, there are many interesting features we may obtain by investigating local weights and their dynamics across time. Thus, we report here a simulation experiment that gives more insight into these matters, allowing us to obtain model preferences "locally".

We start our simulation experiment by generating a set of spatiotemporal synthetic data. We mainly follow the same simulation framework used in Section 3.3.3, without replications, as we focus on weights communalities. Then, we have values on $n \times q \times t$ outcome Y generated from (1.13) with $n = 500$, $q = 3$, $t = 100$ and $p = 2$, X includes two predictors generated from a standard uniform distribution over $[0, 1]$. While the $n \times n$ spatial correlation matrix V is specified using an exponential correlation function with $\phi = 4$ and $\alpha = 0.8$. We specify $J = 9$ models $\mathscr{M}_j$ letting range $\alpha \in \{0.7, 0.8, 0.9\}$, and $\phi \in \{2, 4, 6\}$, for all time points. See Section 3.3.3 for further details.

We start our discussion with the examination of Figure 3.2. We report here two panels, from left to right: "consensus" weights ($\hat{w}_C^{(t)}$), and "global" weights ($\hat{w}_G^{(t)}$) dynamics over time-points. Firstly, worth highlighting that, by construction of dynamic BPS, as the time instant grows, more information is available to determine the weights. This leads to possibly interpreting panels in Figure 3.2 as trace plots, i.e., reflecting weights convergence. That said, we notice decreasing variability of the weights' magnitude as time passes. Both approaches completely stabilize the weights' central tendency after about 30 time-points, even if $\hat{w}_G^{(t)}$ appear to have smoother and faster transitions. Let alone the transitions, we also noticed a greater separation in magnitude for $\hat{w}_C^{(t)}$, while spatially averaged weights ($\hat{w}_G^{(t)}$) tend to concentrate. Figure 3.2 also shows a model-ranking coherence among the two approaches, which means they give the same predictive "relevance" to the same model. Therefore, we can state there are no sensible differences using either one method or the other, as we obtain comparable results.

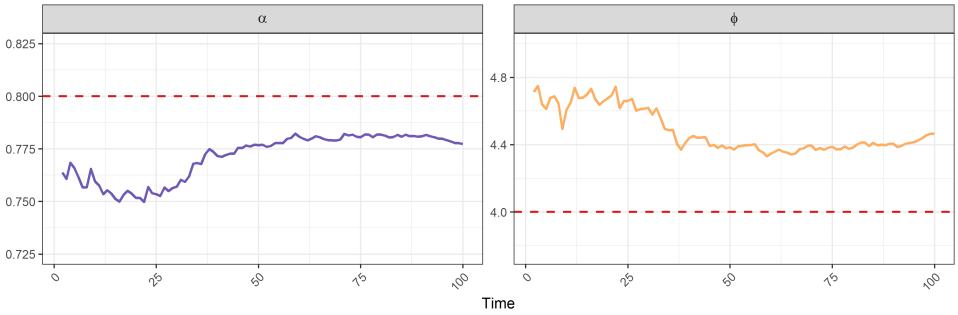

Figure 3.3: Parameter estimates dynamic - with true value (red dashed line)

Hereafter, we then consider global weights $\hat{w}_G^{(t)}$ for DYNBPS. Figure 3.3 represent the empirical estimates for $\{\alpha, \phi\}$ at different time point. As Figure 3.2, we could interpret these as trace plots. Let us notice that the estimates are given as follows: $\hat{\alpha}_t = \sum_{j=1}^{J} \hat{w}_G^{(t)} \alpha_j$, and $\hat{\phi}_t = \sum_{j=1}^{J} \hat{w}_G^{(t)} \phi_j$. In Figure 3.3, leftmost panel show the trace plot for α, while rightmost for ϕ. Both are in some sense attracted by the true values (dashed lines), but for α there is an understimation, in opposition to ϕ, which is overestimated. The latter, which can be identified as "oversmoothing" is typical when working with a distributed learning approach (see e.g., Guhaniyogi and Banerjee, 2018; Presicce and Banerjee, 2024). The understimation for α, may actually descend from the oversmoothing of ϕ. Indeed, the model could underestimate the proportion of spatial variance, i.e., α, when a bigger ϕ is preferred, as it suggests a stronger spatial dependence, which, ceteris paribus, should absorb less variability from the "total", still retaining the same spatial dependence. It remains of interest the investigation into the model balancing behaviour of the dynamic stacking weights, but we will reserve it for future work. At the moment, we can only conclude observing a good attitude of the DYNBPS, at least when working in $\mathscr{M}$-closed setting, as the experiment was.

Lastly, we extrapolate insight from location-wise weights, which are shown in Figure 3.4. This panel is twofold: the average weight of models, and the spatial distribution of weights coloured per model. The latter makes use of the Voronoi tassellation in order to help interpretation. To better understand the leftmost subplot in Figure 3.4, just consider every coloured bar as the frequency of location that "prefers" that model. Where preference is expressed as the biggest weight. The same reasoning applies to the right subplot: it shows for each location, which is the model to which the greater weight was assigned, i.e., the preferred model. Figure 3.4 clearly reports that most of the locations prefer the model $\{\alpha = 0.8, \phi = 6\}$, but the second preferred model is the true one, i.e., $\{\alpha = 0.8, \phi = 4\}$. Notwithstanding many locations do not prefer the true model, as in the data-generating process, the conjunction between spatial variability and non-spatial association among variables leads to sample georeferenced points with individual characteristics for which the posterior predictive distribution pushes towards different values. We actually average out locally dependent micro-characteristics as justified by the backward sampling, which aims to identify a common set of parameters. However, the methodological contribution allows for further investigation of many other perspectives, giving users a flexible level of interpretability for the analyses. From Figure 3.4, we can bring home that every location has a personal set of weights, based on the local evaluation of posterior

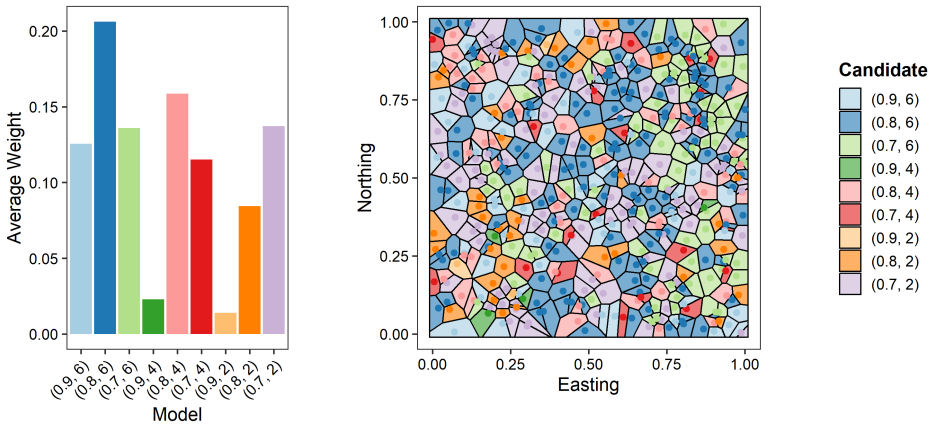

Figure 3.4: Location-wise weights spatial preference - over Voronoi tassellation

predictive distributions. Even though it may be computationally cumbersome to start a local analysis for all the points, there is still the chance to individually investigate each point, which corresponds to a multivariate time series by itself. This will imply that from the joint procedure, you can also decide to further study a subset of points, which may be of interest for a specific research question. Once the forward filtering, along with weights, the backward sampling can be done for just the interested point, for which any of those can be used as an individual set of weights, reflecting local model preferences.

3.3.2 Propagation comparison along $\mathscr{M}$-closed & $\mathscr{M}$-open settings

We establish and compare the effectiveness of contributions in Sections 3.1,3.2 under 50 replications, using synthetically generated data from the model in (1.13), with $\alpha = 0.8$ and $\phi = 4$, under an exponential spatial correlation function. At each round, we uniformly sample $n = 150$ fixed locations over a unit square, for $T = 20$ observed time instants, yielding 3000 observations. We opt to generate data for $h = 5$ future time points, helping us with temporal forecast evaluation. The response Y consists of $q = 3$ correlated spatial outcomes, while X includes $p = 2$ predictors, simulated from a uniform distribution fixed over time. The matrices G_t and F_t are also held constant over time, as specified in Section 3.1. We define $\Sigma = \begin{bmatrix} 1 & -0.3 & 0.6 \\ -0.3 & 1.2 & 0.4 \\ 0.6 & 0.4 & 1 \end{bmatrix}$, while the true regression coefficients are initialized at $\Theta_0 = 0_{(p+n)\times q}$. Then, data evolved following Model (1.13) dynamics.

For BPS-based procedures, we define two scenarios: $\mathscr{M}$-closed, and $\mathscr{M}$-open settings. This allows for a fair comparison between the two alternatives, whether in simple or complex contexts. We refer to BPS-C and BPS-O for the variational propagation approach described in Section 3.1, and to DYNBPS-C and DYNBPS-O for the parallel propagation method introduced in Section 3.2, corresponding to closed and open settings, respectively. Within closed environments, we specify $J = 9$ models $\mathscr{M}_j$ letting range $\alpha \in \{0.65, 0.8, 0.95\}$, and $\phi \in \{2, 4, 6\}$, for all time points. For open settings, we randomly sample three values for α from a uniform distribution over $[0.2, 1]$ and ϕ from a uniform distribution over $[1, 20]$, for each replication. We complete the model specification putting on Θ, Σ a matrix-normal inverse Wishart prior, where parameters were

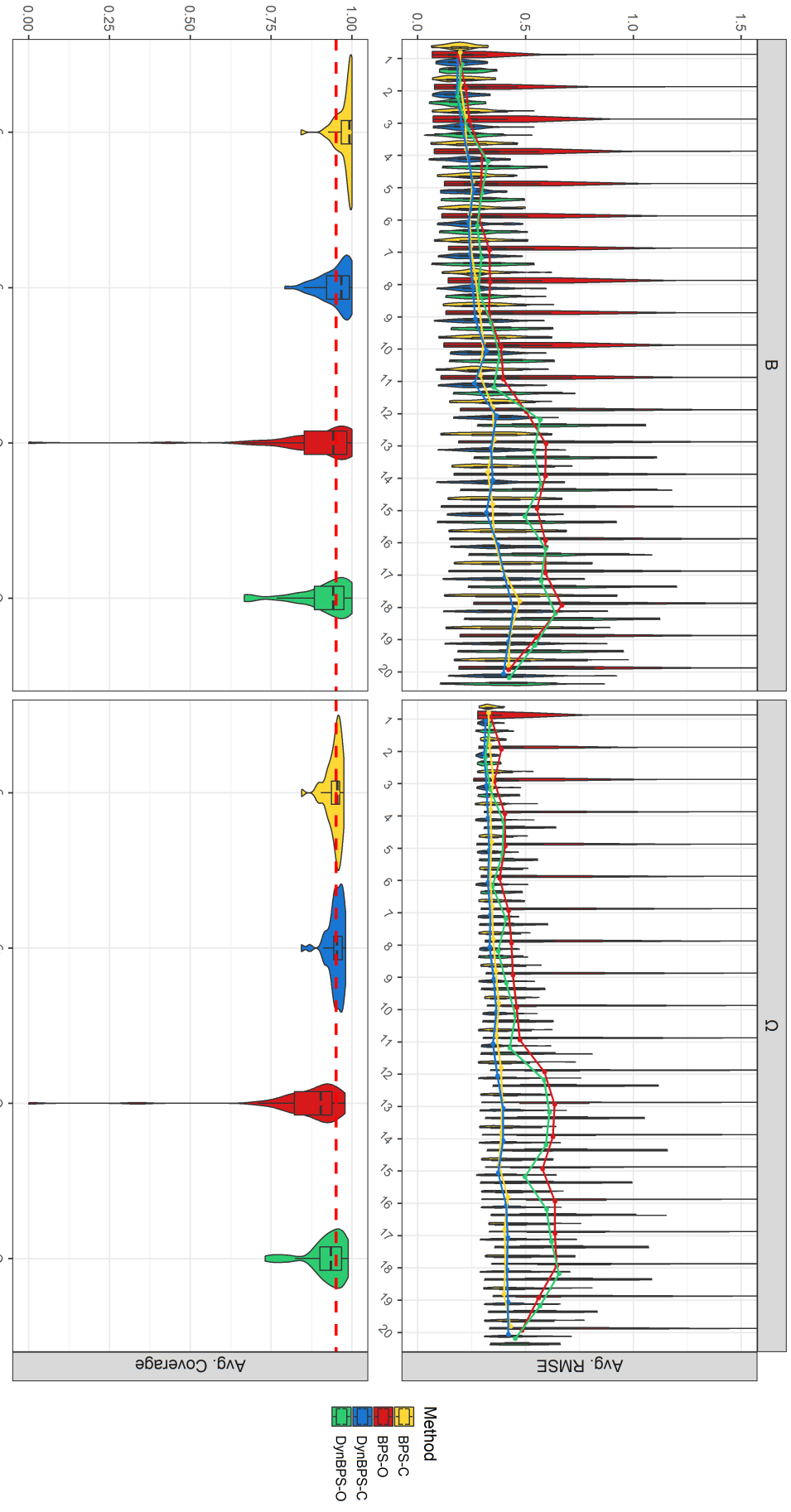

Figure 3.5: Average RMSE, and coverage, for regression coefficients B, i.e. $\Theta_{1:p,1:q}$ - over 50 replications

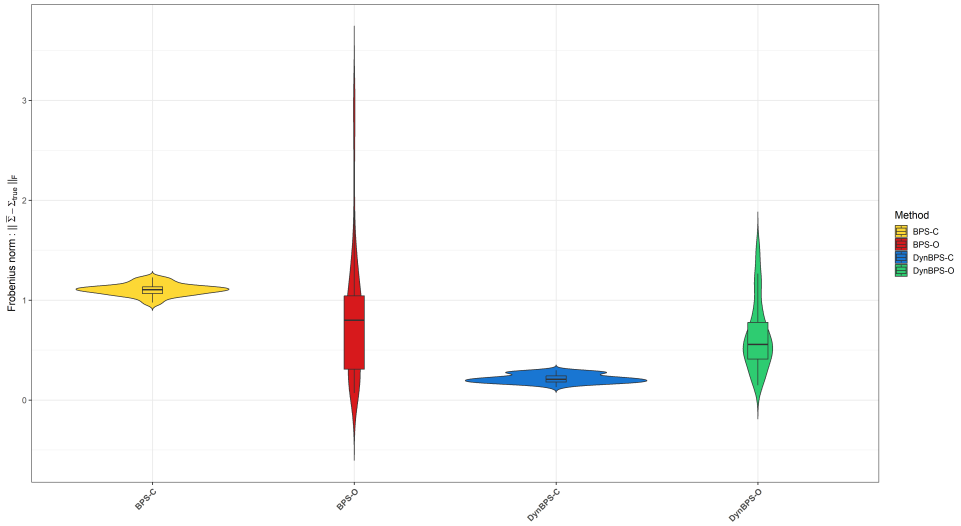

Figure 3.6: Frobenius norm for Σ - over 50 replications

fixed as: $m_0 = 0_{(p+n)\times q}, C_0 = \begin{bmatrix} \mathbb{I}_p & 0_{p\times n} \\ 0_{n\times p} & \mathcal{R}_0(\mathcal{S},\mathcal{S};\phi{=}1) \end{bmatrix}$, opting for an exponential spatial correlation function, and $\nu_0 = q, \Psi_0 = \mathbb{I}_q$. We withheld out 50 locations for each time point, comprising $\mathcal{U}$, to check spatial interpolation performances.

We start presenting parameter replication results, presented in Figure 3.5, and Figure 3.6, for dynamic parameters Θ_t and column covariance matrix Σ, respectively. Figure 3.5 shows four panels: divided by metrics on rows, and submatrices of Θ by column. Actually, the first column refers to the dynamic regression coefficients B_t, while the second refers to the multivariate spatial process Ω_t. The first row reports replication results on the average root mean square error at any observed time point; the second row shows replication results on the average coverage, with respect the nominal 95% level. Any panel of Figure 3.5 points out a neat separation between BPS, and DYNBPS, especially in terms of variability across replications. This also suggests an intrinsic instability of the estimates. For both dynamic regression coefficients and the multivariate spatial process, we observe how dynamic BPS offers more reliable results, almost unaffected by the setting. In particular, the coverage tends to be concentrated on the nominal level. Conversely, for variational propagation, which uses standard BPS, the coverage is overstimated in closed settings, while understimated in open settings instead. Worth noticing, how open settings are destructive for variational propagation, which suffer and produce a total defeat in terms of any metrics for some replications. Even though median tendency across time (represented as coloured line in the top subpanels) for variational propagation and parallel propagation in open settings appears the be close, actually it is just an effect of the robustness of the median to outliers, as for some replications we observe some values totally out of scale. This can also be extrapolated from the extremely low level of coverage BPS in the open setting.

Figure 3.6 shows the distribution of the Frobenius norm for Σ, where we consider the estimation achieved at the last observed time, with all the information available. As the lower the better, DYNBPS obtain the more promising results, offering even more stable results with respect to variational propagation approaches. Figure 3.6 illustrates the same insight regarding the variability of the estimates, showing that BPS with variational propagation yields unstable

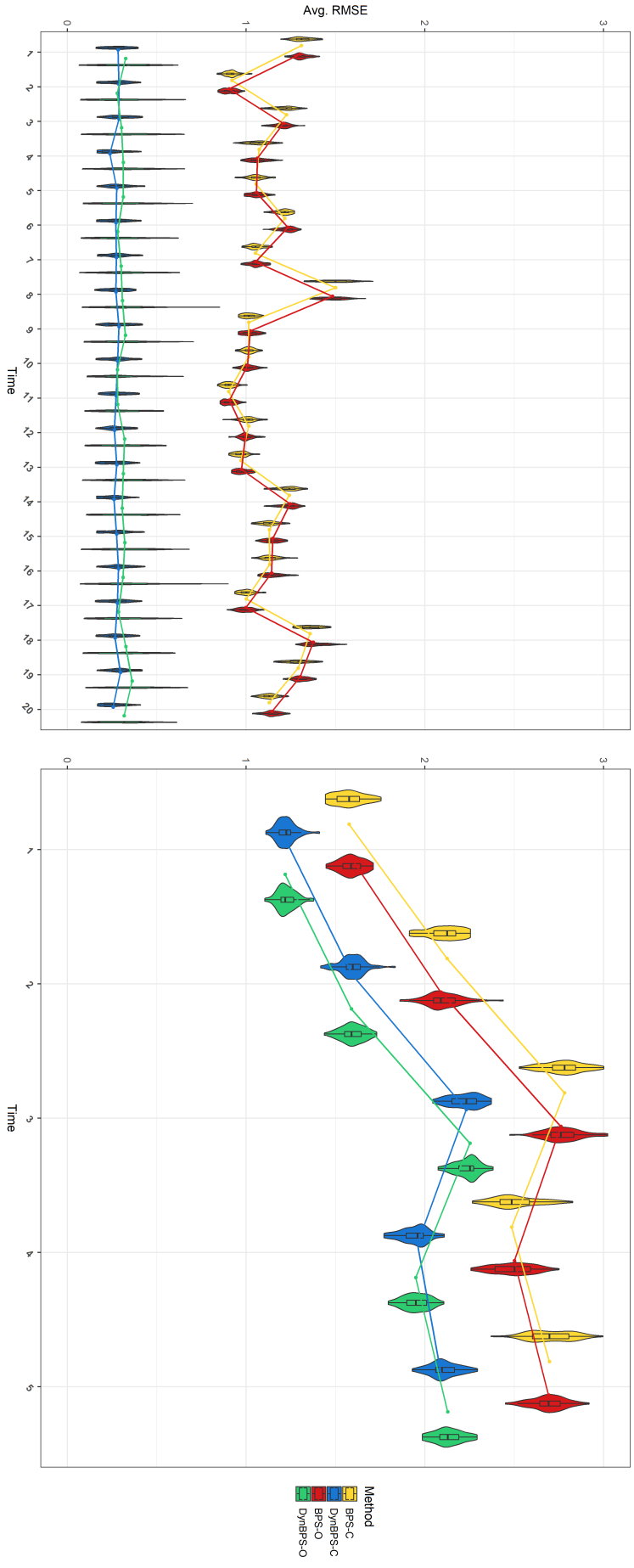

Figure 3.7: Average 1 step-ahead prediction RMSPE for the multivariate response matrix Y, at any time point - over 50 replications

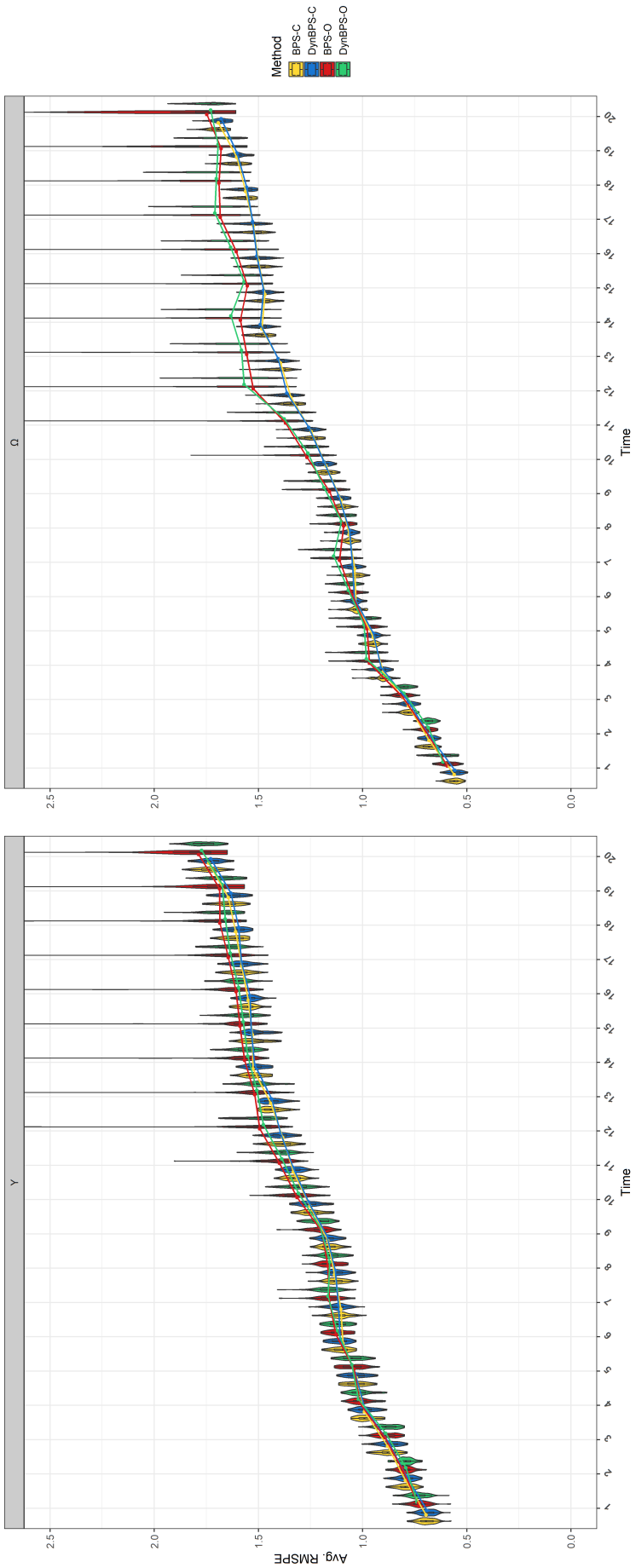

Figure 3.8: Average spatial interpolation RMSPE for the multivariate spatial process Ω, at any time point - over 50 replications

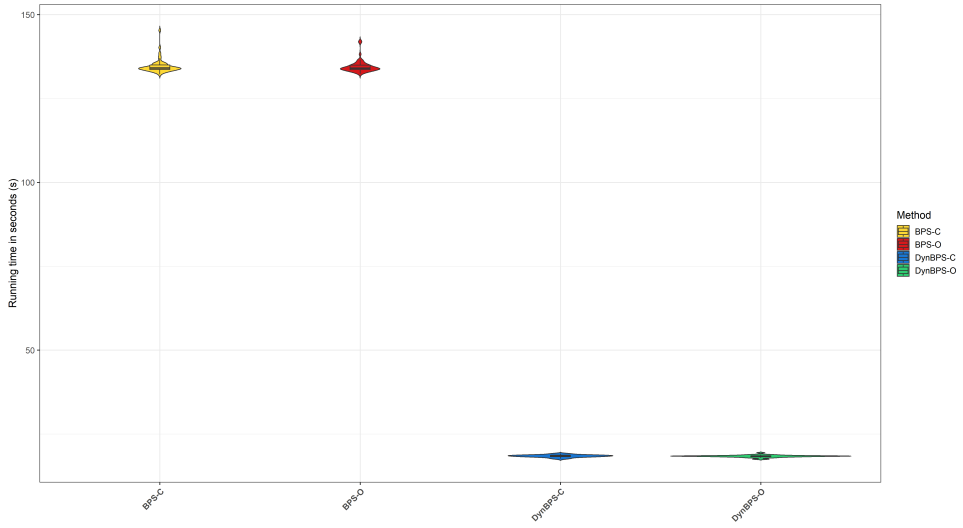

Figure 3.9: Running time distributions for methods and setting - over 50 replications

results. Besides the worst performances, we observe that dynamic Bayesian predictive stacking offers maximum a posteriori estimates for Σ, which are significantly better than those of non-dynamic competitors. This is strengthened by noticing that even in open settings, DYNBPS better estimates that BPS in closed settings.

We also compute temporal forecasts and spatial predictions over replications. We compose Figure 3.7 with two panels. The leftmost shows in-sample average RMSPE for one-step-ahead predictions, the rightmost shows the out-of-sample results instead. Figure 3.7 clearly states the predictive superiority of DYNBPS, for both observed and future times. Actually, parallel propagation performs out-of-sample almost equivalent to variational propagation in-sample. In terms of future predictions, both the propagation approaches show comparable achievement between $\mathscr{M}$-closed and $\mathscr{M}$-settings, suggesting that the misspecification setting does not sensibly affect one-step-ahead predictive sampling.

In Figure 3.8, we report the average spatial prediction RMSPE for the outcome matrix Y_t (left panel) and for the multivariate spatial process Ω_t at both observed and future time points (right panel). Something different happens for spatial interpolation with respect to the temporal forecast. Here, we notice smoother transitions for metric performances between observed time and horizon points. Moreover, Figure 3.8 suggests only slight differences in predictive performance between DYNBPS and BPS. This apparent similarity is largely due to the median "amortization" of extreme values. As shown in the left panel, BPS-O exhibits highly unstable predictive performance, consistent with the variability observed for parameter estimates in Figure 3.5. Lastly, we notice the difference in spatial prediction between closed and open $\mathscr{M}$-settings to be more evident for spatial process interpolation. Despite the extremely poor prediction of BPS-O, the variational and parallel propagation approaches perform reasonably similarly. For both methods, spatial interpolation accuracy worsens over time, consistent with the accumulation of uncertainty as predictions are propagated forward without additional data. This effect is less pronounced for DYNBPS, which benefits from its dynamic updating mechanism.

We conclude this simulation experiment with some computational remarks. We compare the running times for the two approaches under the two settings. Figure 3.9 reports violin plots of running time distributions over the 50 replications. As expected, the $\mathscr{M}$-setting does not impact computational structure, as the discriminant resides in the number of models. The difference in running time between dynamic parallel propagation and variational propagation is striking: in this (moderate) setting BPS takes six times longer than DYNBPS to achieve inferences. Not to mention that DYNBPS also performs better. At the current state, we then suggest to prefer DYNBPS with parallel propagation introduced in Section 3.2, over variational propagation approach presented in Section 3.1. Besides the elegant derivations presented in Appendix B, there is no clear advantage in using variational propagation over parallel propagation. For this reason, we will focus our next simulation experiments and data applications on the contribution introduced in Section 3.2.

3.3.3 Dynamic BPS in $\mathscr{M}$-closed & $\mathscr{M}$-open settings

We explore how dynamic BPS behaves in the $\mathscr{M}$-closed and $\mathscr{M}$-open settings. Generally, these two classes of model misspecifications relate to opposite scenarios. The $\mathscr{M}$-closed refers to the case when the true model can be identified within a finite set of candidate models. $\mathscr{M}$-open class admits the existence of the true models, but this can not be directly specified within candidate models.

Hereafter, model specification is characterized by values of α and ϕ, where the data generating values $\{\alpha = 0.8, \phi = 4\}$ represent the true model (TRUE). The dynamic BPS was tested over $\mathscr{M}$-closed and $\mathscr{M}$-open settings. For DYNBPS under closed setting (DYNBPS-C) we specify $J = 9$ competitive models with $\alpha \in \{0.7, 0.8, 0.9\}$ and $\phi \in \{2, 4, 6\}$ that yields effective spatial ranges in the percentage of maximum point inter-distance of 105%, 53%, 35% respectively, including the true model as one of the possible candidates. Conversely, in the $\mathscr{M}$-open setting, even though the true model exists, it cannot be fully specified. Thus, for dynamic BPS under the open setting (DYNBPS-O), we randomly define $J = 9$ candidate models. In particular, we uniformly sample 3 values for $\alpha \in (0.5, 1)$ and 3 values for $\phi \in [1, 50]$. We perform the experiment using 25 replications.

Each replicate consists of values of $n \times q \times t$ outcome Y generated from (1.13) with $n = 100$, $q = 3$, $t = 20$ and $p = 2$, X includes two predictors generated from a standard uniform distribution over $[0, 1]$, $\Sigma = \begin{bmatrix} 1 & -0.3 & 0.6 \\ -0.3 & 1.2 & 0.4 \\ 0.60 & 0.4 & 1 \end{bmatrix}$. $\Theta = [B^T : \Omega^T]^T$ was defined letting evolving the system in (1.13), starting with a realization from the marginal prior distribution. We equally defined for all model $\mathscr{M}_j$ the joint prior distribution for Θ, Σ from the matrix-normal inverse Wishart family, with parameters $m_0 = 0_{p+n\times q}$, $C_0 = \begin{bmatrix} 0.05\mathbb{I}_p & 0 \\ 0 & \mathcal{R}(\mathcal{S},\mathcal{S};\phi=1) \end{bmatrix}$, $\Psi_0 = 10\mathbb{I}_q$, and $\nu_0 = q + 1$.

The $n \times n$ spatial correlation matrix V is specified using an exponential correlation function with $\phi = 4$ and $\alpha = 0.8$.

Every column in Figure 3.10 shows predictive sampling metrics for each simulated outcome. Rows show (i) absolute bias; (ii) mean square prediction error (MSPE); (iii) predictive interval width; and (iv) predictive variance. We report boxplots for the distribution of each metric across the 50 replicates at any instant in the horizon, i.e., future time points. In terms of predictive MSPE and absolute bias, we notice equivalent results among the settings, indicating excellent

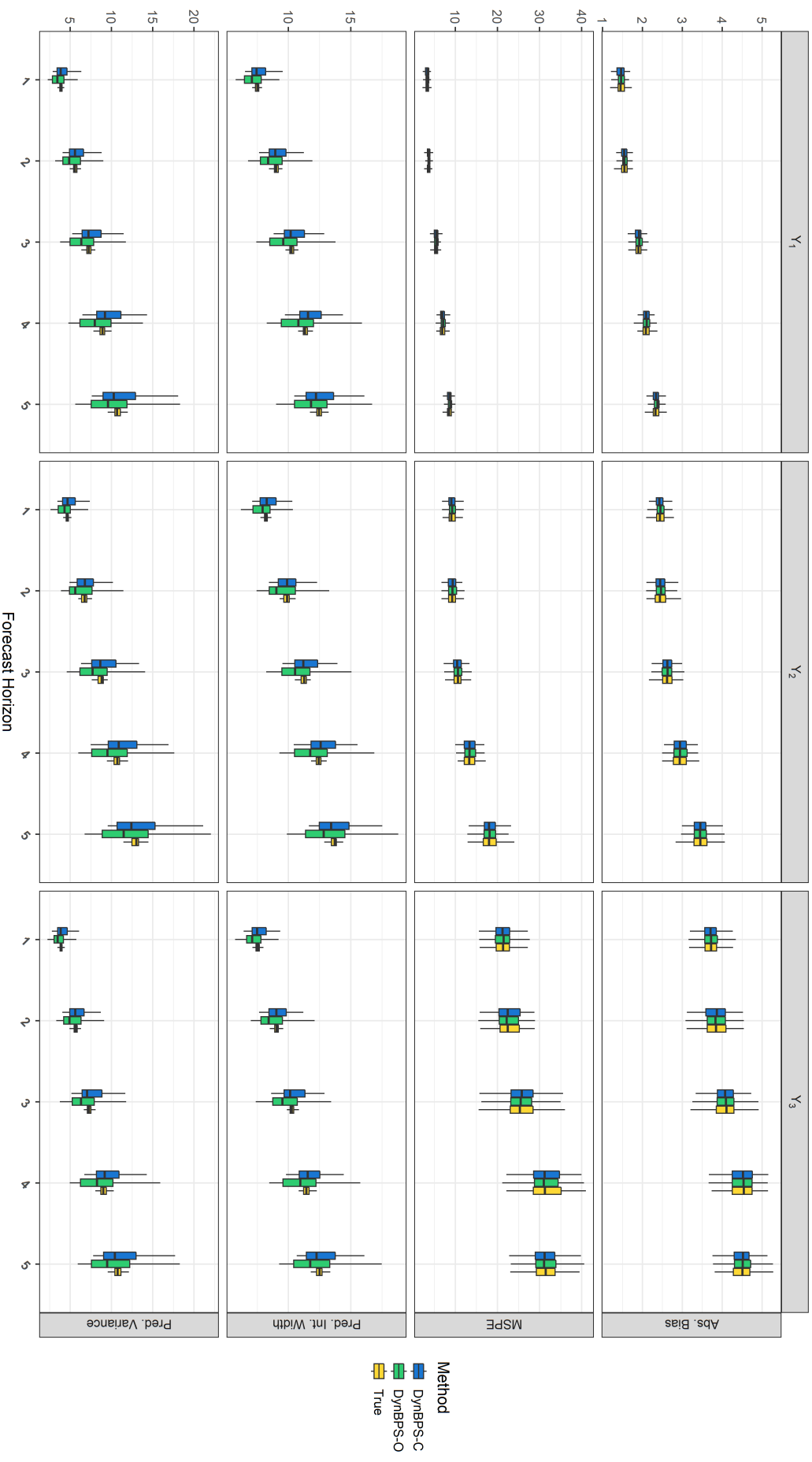

Figure 3.10: Predictive metrics for each response variable - over 50 replications

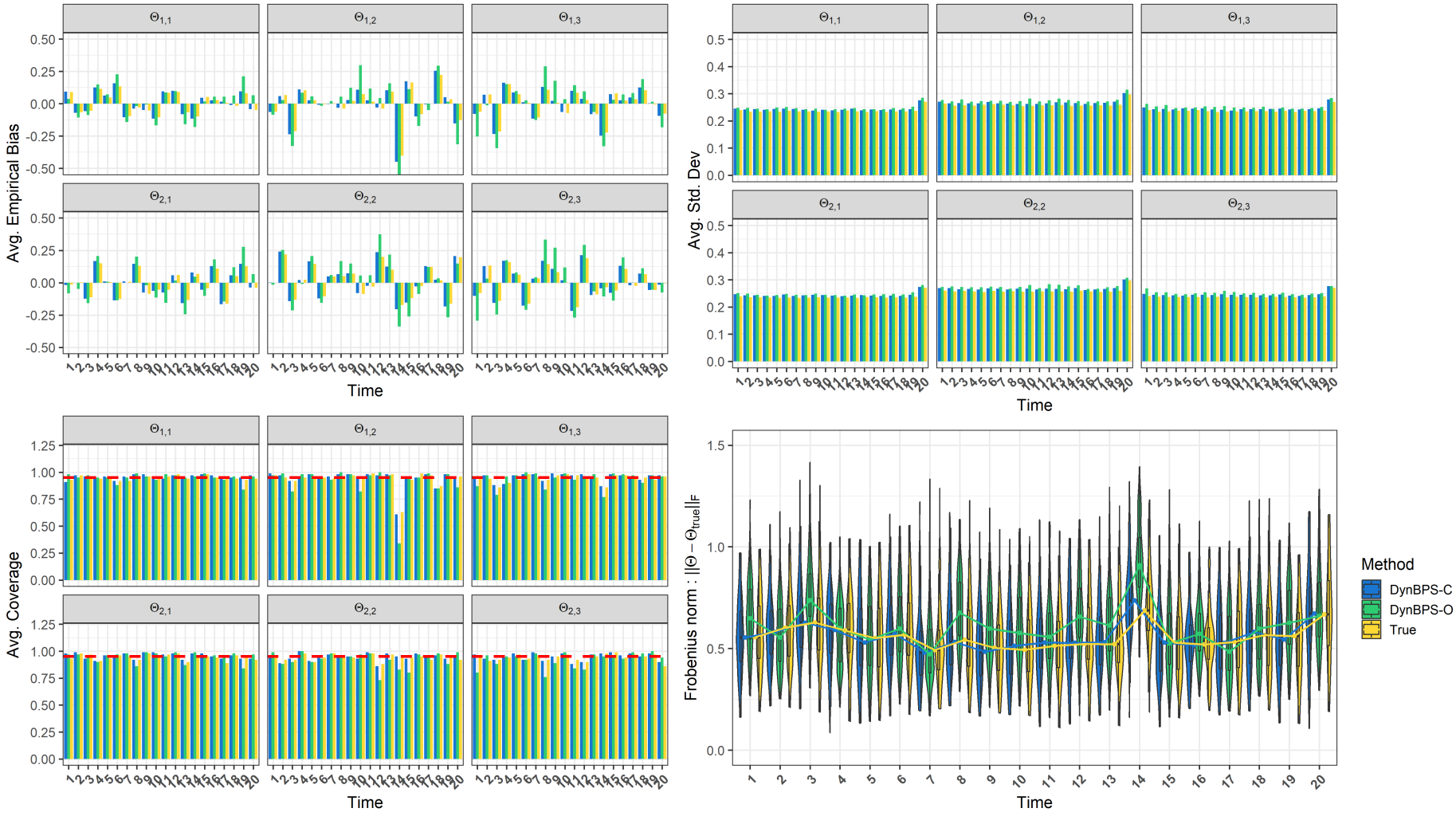

Figure 3.11: Posterior metrics for regression coefficients B, i.e. $\Theta_{1:p,1:q}$ - over 50 replications

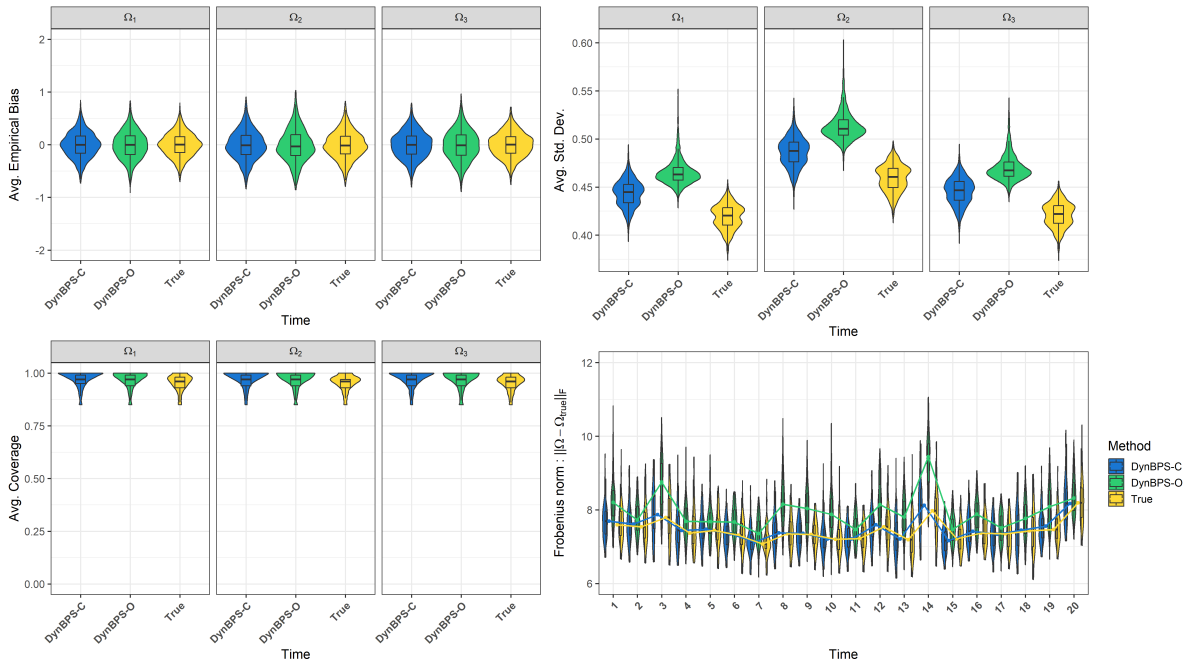

Figure 3.12: Posterior metrics for each component of the multivariate spatial process Ω - over 50 replications

predictive aptitude of dynamic BPS, no matter whether closed or open case. Soft differences emerge with respect to predictive interval width and variance. Expectedly, $\mathscr{M}$-open setting induces bigger variability of predictive distributions, even though it is largely comparable with $\mathscr{M}$-closed. Actually, predictive variability inflates when we introduce uncertainty in $\{\alpha, \phi\}$.

The series of Figures 3.11,3.12,3.13 aim to provide insight of posterior inference achieved by DYNBPS when comparing $\mathscr{M}$-closed and $\mathscr{M}$-open settings. Every figure displays four panels: (i) average empirical bias; (ii) average coverage; (iii) average posterior standard deviation; and (iv) Frobenius norm distribution, across the 50 replicates.

Starting from time-dependent parameters, Figure 3.11 mainly shows which regression coefficient submatrix B_t is well-identified from DYNBPS. The bottom-right sub-panel includes the distributions over replications, at any time point t for any setting, of the Frobenius norm, i.e., $\| \hat{B}_t - B_t \|_F = \sqrt{\sum_{i=1}^{p} \sum_{j=1}^{q} | \hat{\beta}_{i,j,t} - \beta_{i,j,t} |^2}$, where $\hat{B}_t$, $\hat{\beta}_t$ represent the maximum a posteriori (MAP) estimate of B, and its element $\beta_{i,j}$. We support the interpretation using a different coloured line for each setting, representing the median of the Frobenius norm distribution across time points. Looking at the average coverage (top-right) and Frobenius norm (bottom-right) subplots, we record a unique exception at time $t = 14$: all the evaluated models produce the worst estimates. Nevertheless, the replication results are generally balanced among settings for the considered metrics. $\mathscr{M}$-open stand out in terms of empirical bias and Frobenius norm, but the differences are numerically negligible. This is somewhat expected when working with $\mathscr{M}$-open settings, especially given the simulation setting: we sampled a different set of values for $\{\alpha, \phi\}$ across replications. This inevitably inflates variability and shifts into diffusion metrics. Different from Figure 3.11, which reports a panel for each element of B; Figure 3.12 collects data for each of the $q = 3$ components of the multivariate spatial process. Thus, the only subpanel that is time-dependent reflects Frobenius norm distributions (still bottom-right),

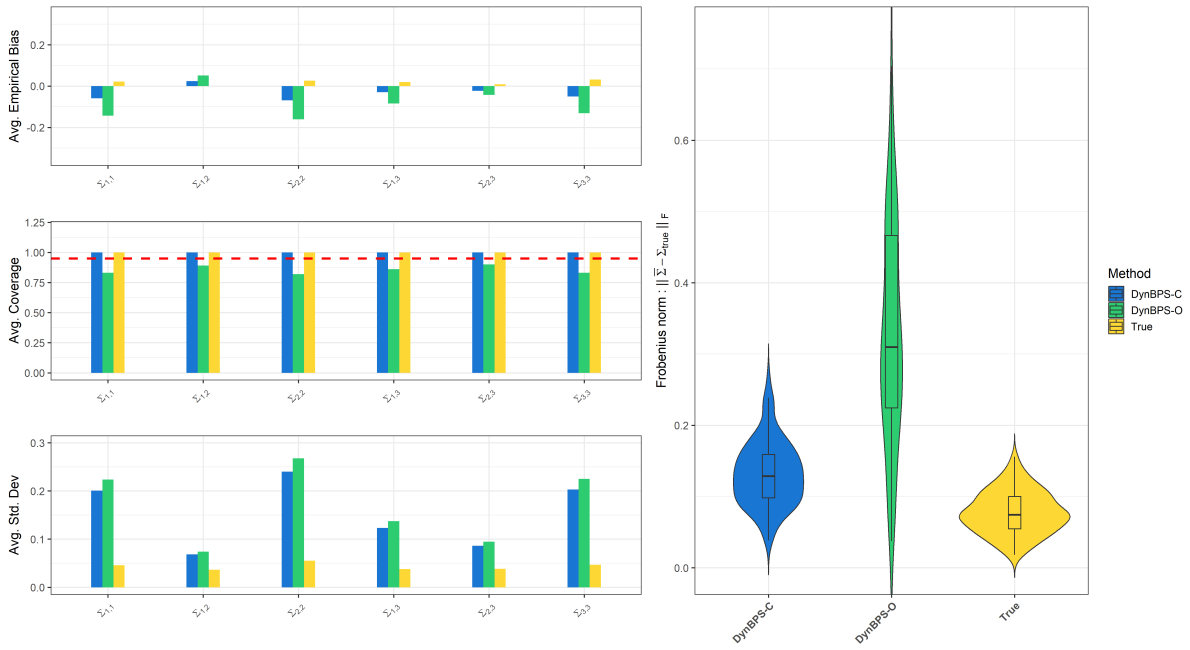

Figure 3.13: Posterior metrics for Σ - over 50 replications

similarly to the bottom-right subplot of Figure 3.11. Here, there is a net separation between $\mathscr{M}$-open and other settings, which perform more stably. Indeed, the median-trace for $\mathscr{M}$-open settings shows sensibly greater values. However, in terms of average empirical bias (top-left) and average coverage (bottom-left), $\mathscr{M}$-close and $\mathscr{M}$-open performances are indistinguishable, ensuring good inferential achievements. While the average standard deviation results (top-right) mirror the behavior of the underlying subpanel: $\mathscr{M}$-close stay "closer" to the true model in terms of estimates' diffusion, "setting" distance from $\mathscr{M}$-open setting. From Figure 3.12, we ascertain the striking capabilities of recovering the true multivariate spatial-dependence between the q outcomes, across time. DYNBPS provides excellent uncertainty quantification for either $\mathscr{M}$-close or $\mathscr{M}$-open setting.

We conclude the simulation experiment, gathering information on the time-independent dense covariance matrix Σ, which exposes the multivariate non-spatial dependence among the q variables. We divided Figure 3.13 into two columns. We composed the leftmost column stacking vertically three metrics: (i) average empirical bias; (ii) average coverage; and (iii) average standard deviation, separately represented by its element, i.e., $\Sigma_{i,j}$ for $i, j = 1, \ldots, q$. In the rightmost column, we report the Forbenius norm distribution across replication, for each setting. Conversely, from Figures 3.11,3.12, which show time-dependent objects, here we only have a distribution by setting, where the posterior distribution at the last observed time is considered to compute the MAP estimates. The latter reflects something already seen for B, and Ω: DYNBPS-O inflates variability on the estimates. However, if for regression coefficients and multivariate spatial process, the greater diffusion of the estimates does not affect the inferential performances, such as the empirical bias and the coverage, here it does. The average coverage (middle-left) suffers from undercoverage, in equal measure for all the parameters. Even though differences are negligible, as it attests to having an average coverage of about 90%, it is worth

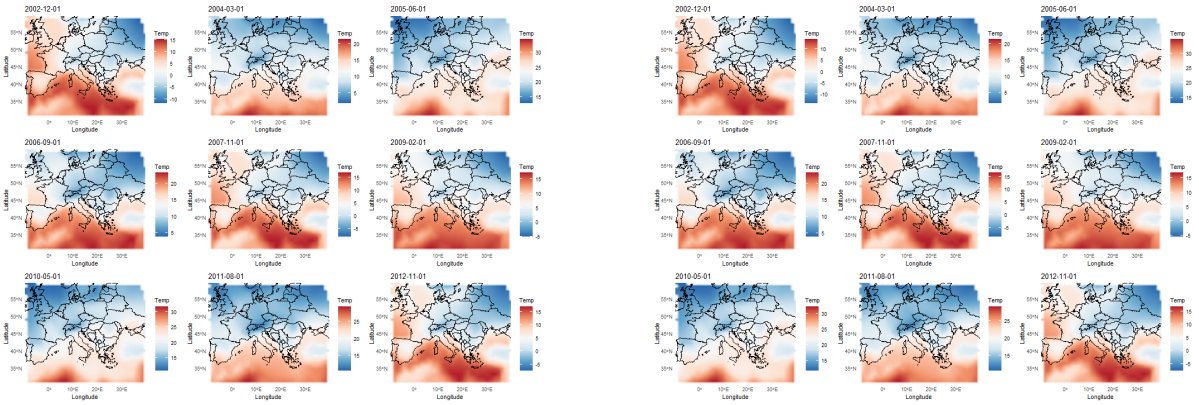

Figure 3.14: 1 step ahead average monthly temperature forecast for selected time points: truth (left panel) and predicted (right panel)

to be noticed. Similar traits are reflected by average empirical bias, which takes distance from that achieved by other settings. Notwithstanding the great achievement of dynamic BPS, we are not surprised that the worst inferential performances are related to the common-covariance matrix Σ. Indeed, when working with georeferenced data is well-known that covariances are not identifiable from data realizations (Zhang, 2004; Tang et al., 2021), even more so when data shows multidimensional dependencies. Worth noting, these findings are consistent with Stein (1988); Stein and Handcock (1989), who theoretically established the ability of Gaussian processes to deliver good predictive performance even for misspecified covariance functions in fixed domains (in-fill asymptotics).

3.4 Application to COPERNICUS data

There is broad consensus that climate change represents one of the most overreaching global challenges, alongside other highly critical and interconnected threats to humanity and life on Earth (Houghton, 2005; Barnard et al., 2021). The consequences of global warming permeate many aspects of human and natural systems, including the spread of infectious diseases (Khasnis and Nettleman, 2005), agriculture and food production (Smith et al., 2008), as well as the climate itself, which is increasingly characterized by extreme events (Pielke et al., 2005). The primary drivers of global warming are associated with changes in global temperatures and atmospheric composition. For this reason, international agencies, e.g., NASA, ESA, and NOAA, developed monitoring systems based on satellite imaging and in-situ monitoring. Thus, different satellite-based collections of global-extended data are available and managed in accessible databases by national administration agencies.

This study utilizes high-resolution climate data from the Copernicus Climate Change Service (C3S) Atlas, a multi-source dataset developed within the European Union's Copernicus

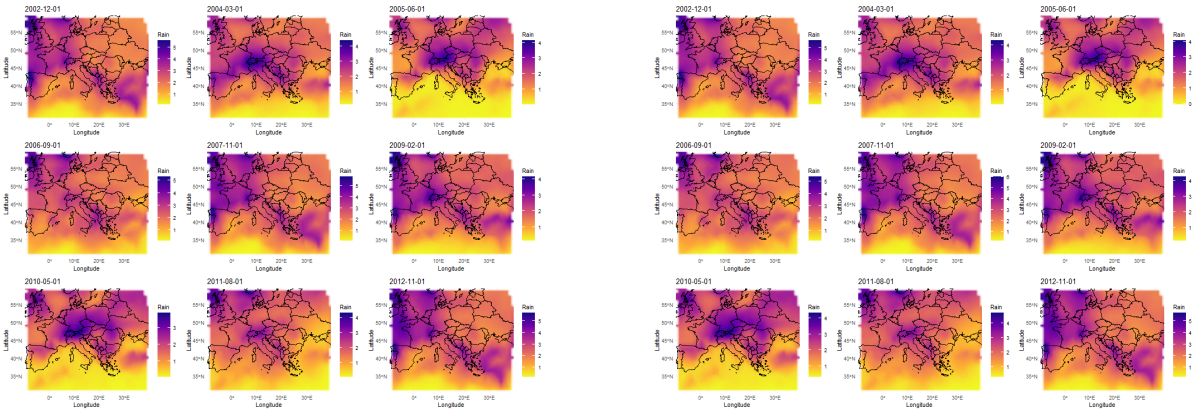

Figure 3.15: 1 step ahead average monthly rain forecast for selected time points: truth (left panel) and predicted (right panel)

Programme. The C3S Atlas compiles harmonized gridded climate variables from different sources, including satellite remote-sensing and in-situ monitoring networks. These datasets are accessible via the Climate Data Store (CDS) and conform to ACDD NETCDF4 metadata conventions, ensuring compatibility, reproducibility across climate analyses, and consistency for long-term climate monitoring.

Temperature is not the only crucial factor for environmental and climate scientists. Actually, many variables are determinants of the atmospheric composition, especially when considering its evolution. Here, we present a multivariate spatiotemporal modeling for four of these important key quantities employed in climate science studies: 2-meter air temperature (Celsius degrees), total monthly precipitation (mm), 10-meter wind speed (ms^{-1}), and monthly evaporation (mm). These variables collectively represent essential surface climate processes with relevance to hydrology, energy balance, and atmospheric dynamics. The scientific research question concentrates on jointly predicting atmospheric influencing factors from massive, spatiotemporal datasets of this scale using GeoAI. We assess whether multivariate statistical models can deliver accurate, scalable, and timely predictions across tens of thousands of spatiotemporal points. These settings are characterized by long-term temporal dependencies and strong associations, while accounting for evolving multidimensional dependence structures among multivariate outcome series. The aforementioned contribution has the scope to enable researchers to perform spatiotemporal analysis, based on a rigorous statistical framework, on large temporal and spatial scales, even with modest computational and memory resources. Public and private research organizations are under increasing pressure to examine and interpret data related to global warming. Naturally, the issue of moving such research to AI platforms has come up due to the sheer size of these datasets and the process-driven models needed for their analysis.

We focused our representative analysis on the European region bounded by latitudes 40°, -60°N and longitudes -10°,-30°E, extracting complete monthly records for the period spanning

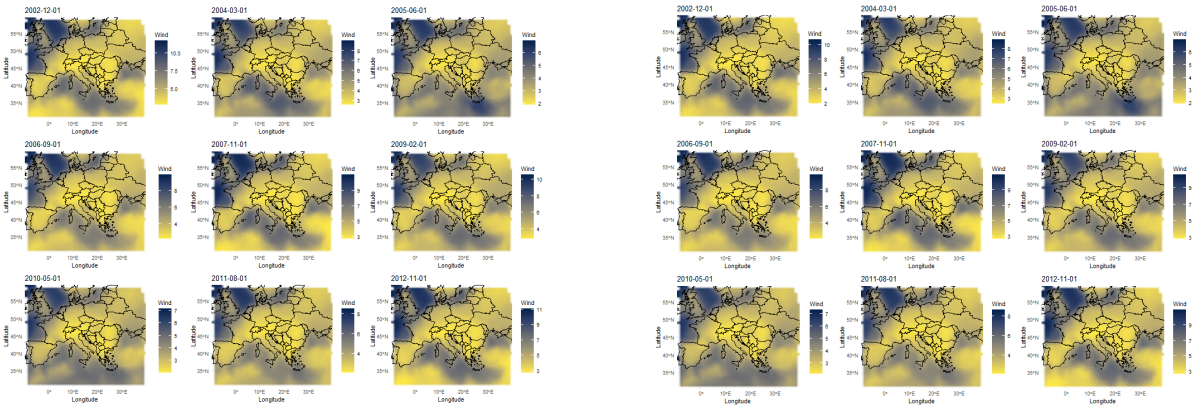

Figure 3.16: 1 step ahead average monthly wind forecast for selected time points: truth (left panel) and predicted (right panel)

the window from December 2002 to December 2014, which is the latest available data point with full variable coverage. This makes the total number of time points $T = 144$, from which we exclude the last $h = 14$ for predictive evaluation, while leaving the first 120 months as the learning set. From this spatiotemporal domain, we extracted data at $n = 600$ regularly distributed locations (100 withheld for spatial interpolation), aggregated for the $q = 4$ outcomes. This led to $60,000$ total number of multivariate space-time points in the dataset, which also corresponds to $240,000$ univariate spatiotemporal observations jointly modeled. The maximum inter-site distance among fixed sites is approximately $6,320$ kilometers. Our statistical modeling approach incorporated a spatiotemporal regression structure that included an intercept term and sinusoidally projected spatial coordinates (an equal-area projection), corresponding to $p = 3$, to capture broad-scale geographic gradients and boundary effects. For the spatial covariance function at any month, we choose to use an exponential kernel, which reflects the spatial process smoothness typically observed over large-scale domains. The temporal structure via monthly indexing enables the model to learn seasonal and interannual dynamics; this setup facilitates both interpolation and prediction tasks within a coherent statistical framework. We implement a fully automated procedure, depicted in Section A.5, to set the grid of values for $\{\alpha, \phi\}$ by using spatiotemporal variograms. The process ends by providing $J = 4$ candidate models: the variograms automatically estimate the absence of nugget for the outcomes, then we fix $\alpha = 0.99$, while letting vary the range parameter as $\phi \in \{0.03, 0.05, 0.15, 0.16\}$. Prior distribution was set following non-informative choices, typically implemented when working with matrix-variate models in spatial analysis (see e.g., Zhang and Banerjee, 2022; Presicce and Banerjee, 2024). We give on Θ, Σ a matrix-normal inverse Wishart prior, where parameters where fixed as: $m_0 = 0_{(p+n)\times q}, C_0 = \begin{bmatrix} \mathbb{I}_p & 0_{p\times n} \\ 0_{n\times p} & \mathcal{R}_0(\mathcal{S},\mathcal{S};\phi=1) \end{bmatrix}$, opting for an exponential spatial correlation function, and $\nu_0 = q, \Psi_0 = \mathbb{I}_q$.

We train the model over the entire observed multivariate time series and then conduct one-

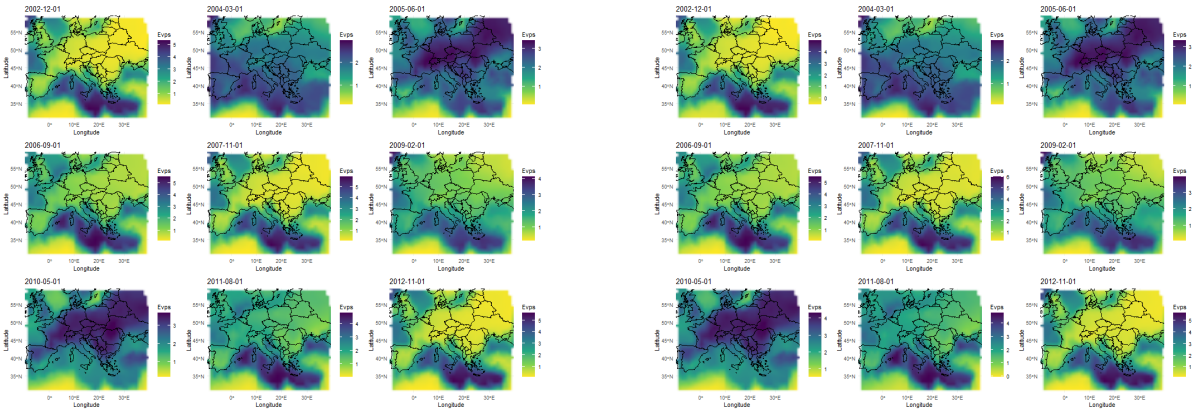

Figure 3.17: 1 step ahead average monthly evaporation forecast for selected time points: truth (left panel) and predicted (right panel)

step-ahead forecasting for the full temporal sequence, including both observed and unobserved time points. The model demonstrated excellent predictive performance, with forecast values nearly indistinguishable from the held-out observations. Forecast performance was evaluated over the full time series for each of the four variables, demonstrating highly accurate predictive behavior with forecast trajectories closely matching ground-truth values. This is illustrated in the included forecast plots, one for each variable: (i) Figure 3.14 for air temperature; (ii) Figure 3.15 for rain precipitation; (iii) Figure 3.16 for wind speed; and (iv) Figure 3.17. The results from these Figures are clear: DYNBPS achieved striking forecast performances, exactly mimicking the observed multivariate spatiotemporal process. In addition to temporal forecasting, we performed spatial interpolation at 100 out-of-sample withheld locations. We conducted spatial prediction at observed time points (in-sample) and at future unobserved times (out-of-sample), providing a robust assessment of the model's ability to generalize spatial interpolation beyond its learning set. Multivariate interpolated maps exhibited smooth gradients and high alignment with known climatological patterns, suggesting that the model accurately captures both temporal variability and spatial structure. Figure 3.18 presents interpolated surfaces for a selected observed time slice (September 2007) for all four variables (right panel), in comparison to observed interpolated fields (left panel). Similarly, Figure 3.19 reports spatial predicted maps for a selected out-of-sample month (September 2013) against observed spatial surfaces (right to left).

Both sets of interpolated fields prove the excellent spatial prediction achieved by dynamic Bayesian predictive stacking. In addition, as we on purpose selected the same month for in-sample and out-of-sample interpolation, i.e., September, we can compare Figure 3.18 with Figure 3.19. Actually, they show extreme similarities, implying how the model can learn spatial patterns without losing seasonal coherence. Together, these results validate the pattern identified by c3s Atlas dataset for large-scale spatiotemporal analyses in climate research and

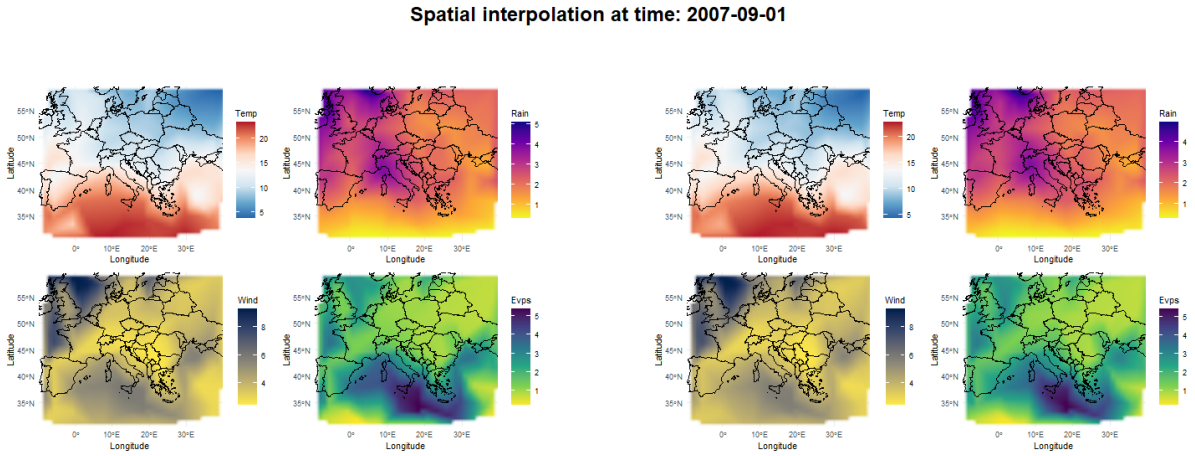

Figure 3.18: Spatial surface interpolation at observed time: truth (left panel) and predicted (right panel)

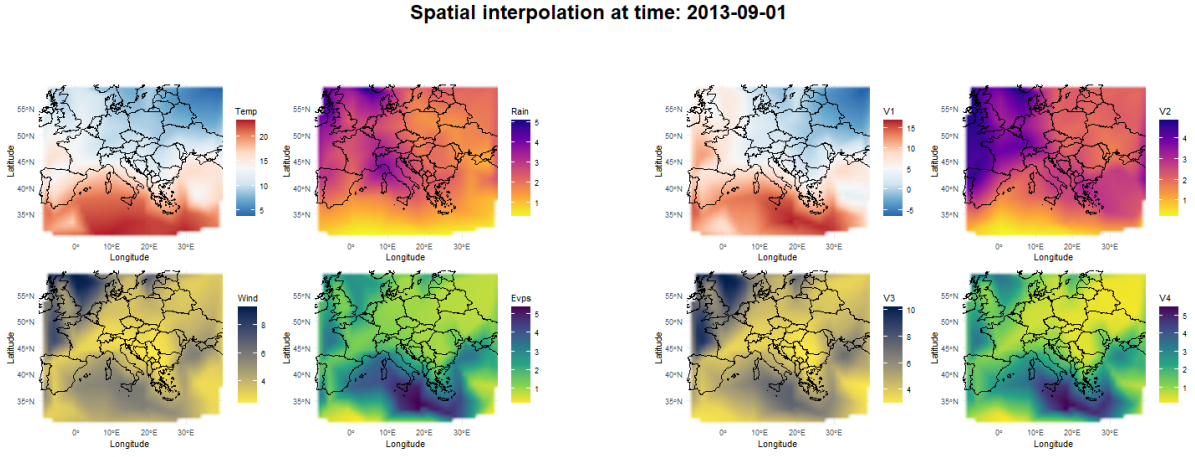

Figure 3.19: Spatial surface interpolation at unobserved time: truth (left panel) and predicted (right panel)

demonstrate the scalability of DYNBPS for joint multivariate forecasting and interpolation tasks across entire Europe. Dynamic BPS bring the feasibility of combining physically interpretable multidimensional factors with data-driven spatiotemporal modeling to a newer level, supporting climate monitoring and decision-making within GEOAI systems. Indeed, the entire Bayesian estimation framework only takes nearly 12 minutes to provide posterior and posterior predictive sampling, along with uncertainty quantification. This happens on a standard laptop, with very limited resources as explained in Section 3.2.1, allowing almost-automated multivariate spatiotemporal Bayesian modeling for large-scale settings to constrained GEOAI systems.

Chapter 4

Conclusions

This thesis develops distribution-theoretic and predictive stacking-based Bayesian transfer learning frameworks for scalable, large-scale spatial and spatiotemporal analysis. The overarching goal is to alleviate key limitations that restrict the practical use of Bayesian methods in geospatial statistics and GeoAI, most notably the scalability of Gaussian-process-based geostatistical models to massive, high-dimensional datasets indexed over tens of thousands to millions of locations, the computational burden of posterior inference and predictive analysis, and the need for substantial manual intervention to devise and monitor analyses in such settings. The proposed contributions combine analytically tractable multivariate and matrix-variate distributions with Bayesian predictive stacking, including dynamic extensions, to deliver rapid inference and uncertainty quantification without reliance on computationally intensive iterative procedures that often require extensive human tuning. These ideas culminate in geospatial inference systems based on double Bayesian predictive stacking and Dynamic Bayesian predictive stacking, which are designed to process large-scale and streaming spatial data efficiently on both limited and high-performance CPU architectures. The resulting framework supports flexible Bayesian transfer learning across spatial and spatiotemporal domains.

The main contributions unfold in two complementary directions. The first focuses on Bayesian transfer learning for spatial data, where in Chapter 2 we introduce a framework for partitioned inference in massive spatial datasets. Each data partition is modeled using a conjugate matrix-variate Gaussian distribution, leading to analytically tractable posterior updates that circumvent the computational demands of simulation-based inference. A novel double Bayesian predictive stacking (DBPS) procedure is then developed, in which predictive distributions are first synthesized within partitions and subsequently across partitions. This double-stacking mechanism yields a coherent global predictive distribution that successfully transfers information across the spatial domain, while preserving the local heterogeneity of different subregions. The approach thus enables posterior inference and predictive uncertainty quantification at a scale and efficiency that cannot be achieved with traditional Gaussian process modeling. Beyond computational efficiency, DBPS offers a principled mechanism for addressing variability across spatial domains, ensuring robustness when data sources are unevenly distributed or when local spatial dynamics differ. This line of contribution positions Bayesian transfer learning as a central statistical component for intelligent geospatial systems capable of operating under massive data regimes.

The second direction embeds Bayesian predictive stacking in spatiotemporal geospatial AI, parallel along two paths. We provide variational approximations of non-conjugate stacked posterior distributions, ensuring forward pushing of temporal information. Then, we extend predictive stacking to dynamical settings, resulting in a methodology for dynamic Bayesian predictive stacking in spatiotemporal models. Dynamic linear models offer a convenient framework, analytically tractable, but still sufficiently flexible to model complex dependencies among space-time for multivariate latent stochastic processes. Chapter 3 focuses on developing a mechanism for sequentially combining predictive distributions across time, with weights that evolve adaptively to structural changes in the data. At each time step, posterior inference remains analytically tractable with closed-form posterior distributions through the combination of conjugate matrix-variate families and Bayesian transfer learning techniques. This is obtained without renouncing full probabilistic coherence, also enabling scalable sequential forecasting, interpolation, and uncertainty quantification across both spatial and temporal domains.

Establishing the rigor of Bayesian analysis within AI geospatial systems means being able to coherently assimilate the foundations of probability theory, along with modern and scalable techniques. In this thesis, we have taken charge of this task. We propose one of the possible paths, reconciling the Bayesian distribution theory with the computational and operational demands of modern GeoAI. We advance the development of reliable tools by grounding multivariate inference in matrix-variate distribution theory, minimizing dependence on iterative algorithms. While the methodological contributions of the thesis have been made explicit in earlier sections, hereafter, we reflect on broader implications, potential extensions, and avenues for future work.

Additional remarks are warranted regarding the versatility and future directions of this work, and several directions naturally emerge. Although the exposition has focused on analytically tractable hierarchical matrix-variate spatial processes, the proposed DBPS framework generalizes naturally to settings where full conjugacy does not hold. The development of fully automated GeoAI systems demands not only scalable inference but also robustness to model misspecification, missing data, and misaligned spatial supports. For instance, in cases of spatially misaligned multivariate data, where missing entries in Y complicate modeling, the distributional theory remains applicable and can be adapted to deliver closed-form posterior summaries (Zhang and Banerjee, 2022).

Likewise, for more complex models with unknown hyperparameters or alternate covariance structures, local inference can be performed via MCMC, variational Bayes, or Laplace approximations, with predictive stacking then synthesizing results across data partitions. Even in such cases, the scaling benefits of predictive stacking are preserved, offering substantial efficiency gains compared with other divide-and-conquer approaches, such as geometric median-based posterior aggregation (used in meta-kriging developed by Guhaniyogi and Banerjee, 2018). The geometric median (GM) generalizes the classical median to a functional space (Minsker et al., 2017). The GM represents the "median" distribution, while BPS seeks the distribution within the convex span of predictive distributions from competitive models that minimizes divergence, induced by the related proper rule score, from the true unknown predictive distributions. As shown in this dissertation, our designed geospatial AI systems require minimal computational

resources (no greater than a standard laptop), making them accessible without the need for extensive computing infrastructure.

A preliminary examination of Chapter 2 and Chapter 3 reveals that posterior inference is conceptualized as a "mixture of mixtures", or mixture in general. This is somehow expected when working with Bayesian predictive stacking. Following the Discussion in Yao et al. (2018), we prefer stacking to fitting a mixture model because the former is numerically robust and requires almost no human tuning, notwithstanding the easier implementation. Moreover, the structure of the mixture-of-mixtures conceptually resembles the "Mixture of Experts" (MoE) that are adopted by AI platforms such as GPT-4 and Mistral. There are broader opportunities to connect predictive stacking with the "mixture of experts" paradigm that underlies many modern artificial intelligence systems. Developing this connection further could allow Bayesian transfer learning procedures to inform and be informed by scalable AI architectures, providing a statistical foundation for systems that integrate geospatial online data streams with large language models.

To this end, future directions will also explore the perceived potential of DBPS, and DYNBPS as feeders for emerging amortized inference methods (Ganguly et al., 2023; Zammit-Mangion et al., 2024; Sainsbury-Dale et al., 2024) to achieve Bayesian inference. Rapid delivery of posterior estimates of the entire spatial process from Bayesian transfer learning approaches devised in Chapters 2 and 3 will amount to more training data for amortized neural learners that can result in accelerated tuning for subsequent Bayesian inference. We already provide some preview of the potential integration between the DOUBLE BPS procedure and artificial intelligence. In Section 2.4.3, we offer an example integrating the computational scalability achieved by the double Bayesian predictive stacking and a deep neural network, where we use the output from the former to train the latter. Dynamic Bayesian predictive stacking also holds promise as a feeder mechanism for amortized inference methods currently emerging in Bayesian computation. By rapidly delivering posterior summaries in massive spatiotemporal datasets, the framework can generate training data for neural amortization techniques, thereby accelerating the calibration of neural Bayesian estimators. The reasoning could be extended to complex convolutional or recurrent deep neural networks, which are particularly data "voracious" models. In this way, we do not view predictive stacking as a competitor to neural amortized inference, but rather as a complementary tool that can provide fast, distributionally principled inputs to machine learning systems. The integration of predictive stacking with emerging computational paradigms in Bayesian inference represents an important opportunity. Amortized inference methods based on neural networks, such as Bayesian deep learning with variational autoencoders or neural posterior estimators, require large volumes of simulated training data. This opens the possibility of hybrid architectures where classical distribution theory provides the data backbone for machine-learned Bayesian engines, combining interpretability with computational acceleration. Such developments will be pursued as future research. We also seek to expand integration of formally defined statistical models with artificial intelligence black-boxes to further expedite and improve geospatial AI systems.

The treatment of heterogeneity across spatial and temporal partitions remains a fertile area for development. However, the versatility of Bayesian predictive stacking also descends from

independence assumptions. While computationally advantageous, it may underrepresent dependencies between neighboring spatial regions or across consecutive time windows. Looking forward, further extensions of Bayesian predictive stacking to explicitly model dependence structures across spatial partitions, or to incorporate graphical Markovian dependence across temporal subsets, represent promising directions for accelerating inference and improving predictive calibration. Indeed, incorporating graphical or Markovian structures into the stacking framework, letting predictive distributions to share information across spatial or temporal boundaries, offers a promising way to balance scalability with statistical fidelity. More broadly, by aligning distribution theory with predictive stacking, the thesis sets the stage for the design of automated, scalable, and robust Bayesian engines that can serve as the statistical backbone of next-generation geospatial artificial intelligence systems. Notwithstanding the introduction of the DYNBPS, which account for Markovian temporal dependence, much can still be done. For example, it could be generalized to further temporal dependency structures, without limiting to Markovian models, again accounting for either spatial or spatiotemporal dependence. We are working on extending our contribution to overcome possible issues and accounting for spatial and temporal dependence within predictive stacking, following the arguments in Cabel et al. (2025). Currently, we still recommend choosing either DBPS or DYNBPS when facing large-scale spatial or spatiotemporal problems, despite the non-i.i.d. nature of the data, as they are capable of delivering approximate Bayesian inference at an unprecedented scale even with scarce computational resources.

Moreover, there is space to broaden the distributional foundations beyond the conjugate matrix-normal inverse-Wishart families emphasized here, particularly by relaxing the implicit separability assumption between row and column covariance structures that underlies matrix-variate formulations. While separability is crucial for analytical tractability and scalability, it represents a substantial modeling restriction that may be inadequate for capturing complex cross-variable and spatial dependence patterns in many applications, and thus deserves further methodological attention. Although these models provide the analytical tractability necessary for scalable inference, many applications call for richer dependence structures, non-stationary processes, or non-Gaussian data distributions. Extending Bayesian transfer learning approaches based on Bayesian predictive stacking to frameworks such as matrix-variate skew-t distributions, spatial copula models, or hierarchical spatial point processes would considerably boost the applicability of the methods. These directions would require relaxing some of the conjugacy assumptions, working on distribution theory for such complex probabilistic distributions, and structuring computationally scalable algorithms, while retaining the spirit of predictive stacking as an analytic synthesis engine.

In summary, while this thesis has established novel methods as a foundation for scalable Bayesian geospatial inference, it also opens several paths for future research. This thesis demonstrates that carefully crafted distribution-theoretic frameworks, combined with Bayesian predictive stacking principles, can reconcile the trade-off between theoretical rigor and computational scalability that has long challenged Bayesian geostatistics. Moving beyond Gaussian assumptions, incorporating richer dependence structures, interfacing with amortized and neural Bayesian inference, and linking predictive stacking to modern AI architectures represent

promising directions for advancing both the statistical and computational frontiers of GeoAI. Pursuing these directions will help consolidate the methodological contributions of this thesis and support the development of a new generation of Bayesian tools suited to modern geospatial data environments characterized by scale, dynamics, and heterogeneity. Beyond the specific models and algorithms introduced here, this work contributes a general framework for principled Bayesian learning and prediction in large spatial and spatiotemporal settings. Ongoing dissemination through open-source software, currently under development in `R`, is expected to significantly boost the adoption of these methods and promote further research at the interface of Bayesian statistics and GeoAI.

Bibliography

Ai, J., Xiao, S., Feng, H., Wang, H., Jia, G., and Hu, Y. (2020). A global terrestrial ecosystem respiration dataset (2001-2010) estimated with MODIS land surface temperature and vegetation indices. *Big Earth Data*, 4(2):142–152. 51

Ali, M. M. (1979). Analysis of stationary spatial-temporal processes: Estimation and prediction. *Biometrika*, 66(3):513–518. 10

Allaire, J. J., Kalinowski, T., Falbel, D., Eddelbuettel, D., Tang [aut, Y., cph, Golding, N., Tutorials), G. I. E., , Posit, and PBC (2024). tensorflow: R Interface to 'TensorFlow'. 40

Banerjee, S. (2017). High-Dimensional Bayesian Geostatistics. *Bayesian Analysis*, 12(2):583–614. 3, 4, 111

Banerjee, S. (2020). Modeling massive spatial datasets using a conjugate Bayesian linear modeling framework. *Spatial Statistics*, 37:100417. 3, 4, 52, 107, 108, 109

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2015). *Hierarchical modeling and analysis for Spatial Data*. Chapman & Hall/CRC. 2, 40

Banerjee, S., Chen, X., Frankenburg, I., and Zhou, D. (2025). Dynamic Bayesian Learning for Spatiotemporal Mechanistic Models. *Journal of Machine Learning Research*, 26(146):1–43. 4

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. 3

Barnard, P., Moomaw, W. R., Fioramonti, L., Laurance, W. F., Mahmoud, M. I., O'Sullivan, J., Rapley, C. G., Rees, W. E., Rhodes, C. J., Ripple, W. J., Semiletov, I. P., Talberth, J., Tucker, C., Wysham, D., and Ziervogel, G. (2021). World scientists' warnings into action, local to global. *Science Progress*, 104(4). 80

Breiman, L. (1996). Stacked Regressions. *Machine Learning*, 24(1):49–64. 4, 18

Cabel, D., Sugasawa, S., Kato, M., Takanashi, K., and McAlinn, K. (2025). Bayesian Spatial Predictive Synthesis. arXiv:2203.05197 [stat]. 5, 19, 90

Cao, J., Kang, M., Jimenez, F., Sang, H., Schaefer, F. T., and Katzfuss, M. (2023). Variational Sparse Inverse Cholesky Approximation for Latent Gaussian Processes via Double Kullback-Leibler Minimization. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and

Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 3559–3576. PMLR. 3

Carter, C. K. and Kohn, R. (1994). On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553. 2, 12, 56, 65

Chen, Q., Han, R., Ye, F., and Li, W. (2011). Spatio-temporal ecological models. *Ecological Informatics*, 6(1):37–43. 10

Cooper, A., Simpson, D., Kennedy, L., Forbes, C., and Vehtari, A. (2023). Cross-validatory model selection for Bayesian autoregressions with exogenous regressors. *arXiv preprint*. arXiv:2301.08276 [stat.ME]. 62

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. 3

Cressie, N. A. C. (1993). *Statistics for Spatial Data*. Wiley Series in Probability and Statistics. Wiley, 1 edition. 2

Cressie, N. A. C. and Wikle, C. K. (2011). *Statistics for spatio-temporal data*. Wiley series in probability and statistics. Wiley, Hoboken, N.J. 2, 10

CVX Research, I. (2012). CVX: Matlab Software for Disciplined Convex Programming, version 2.0. 18, 19, 20, 36

Czaran, T. and Bartha, S. (1992). Spatiotemporal dynamic models of plant populations and communities. *Trends in Ecology & Evolution*, 7(2):38–42. 10

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. 3, 40

Dey, D., Datta, A., and Banerjee, S. (2022). Graphical Gaussian process models for highly multivariate spatial data. *Biometrika*, 109(4):993–1014. 3

Didan, K. (2021). MODIS/Terra Vegetation Indices 16-Day L3 Global 0.05Deg CMG V061 | NASA Earthdata. 113

Esser, J., Maia, M., Parnell, A. C., Bosmans, J., Dongen, H. v., Klausch, T., and Murphy, K. (2025). Seemingly unrelated Bayesian additive regression trees for cost-effectiveness analyses in healthcare. arXiv:2404.02228 [stat]. 40, 47

Fasiolo, M. (2014). *An introduction to mvnfast*. University of Bristol. 40

Finley, A. O., Banerjee, S., and Basso, B. (2011). Improving Crop Model Inference Through Bayesian Melding With Spatially Varying Parameters. *Journal of Agricultural, Biological, and Environmental Statistics*, 16(4):453–474. 40

Finley, A. O., Banerjee, S., and Gelfand, A. E. (2015). spBayes for Large Univariate and Multivariate Point-Referenced Spatio-Temporal Data Models. *Journal of Statistical Software*, 63(13):1–28. 47

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. 3, 30, 107

Fisher, E. L. (1958). Hurricanes and the sea-surface temperature field. *Journal of Atmospheric Sciences*, 15(3):328–333. 51, 110, 113

Friehe, C. A., Shaw, W. J., Rogers, D. P., Davidson, K. L., Large, W. G., Stage, S. A., Crescenti, G. H., Khalsa, S. J. S., Greenhut, G. K., and Li, F. (1991). Air-sea fluxes and surface layer turbulence around a sea surface temperature front. *Journal of Geophysical Research: Oceans*, 96(C5):8593–8609. 51, 110, 113

Fryda, T., LeDell, E., Gill, N., Aiello, S., Fu, A., Candel, A., Click, C., Kraljevic, T., Nykodym, T., Aboyoun, P., Kurka, M., Malohlava, M., Poirier, S., Wong, W., Rehak, L., Eckstrand, E., Hill, B., Vidrio, S., Jadhawani, S., Wang, A., Peck, R., Gorecki, J., Dowle, M., Tang, Y., DiPerna, L., Maurerova, V., Syzon, Y., Valenta, A., Novotny, M., and H2O.ai (2024). h2o: R Interface for the 'H2O' Scalable Machine Learning Platform. 40, 53, 111

Fu, A., Narasimhan, B., and Boyd, S. (2020). CVXR: An R Package for Disciplined Convex Optimization. *Journal of Statistical Software*, 94(14):1–34. 36

Fu, A., Narasimhan, B., Schwendinger, F., and Maechler, M. (2023). ECOSolveR: Embedded Conic Solver in R. 36

Gamerman, D., Lopes, H. F., and Salazar, E. (2008). Spatial dynamic factor analysis. *Bayesian Analysis*, 3(4):759 – 792. 2, 10

Ganguly, A., Jain, S., and Watchareeruetai, U. (2023). Amortized Variational Inference: A Systematic Review. *Journal of Artificial Intelligence Research*, 78:167–215. 89

Gelfand, A. E., Diggle, P., Guttorp, P., and Fuentes, M. (2010). *Handbook of Spatial Statistics*. Taylor & Francis. 2

Gneiting, T. and Raftery, A. E. (2007). Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378. 16, 18, 20, 55, 62

Granström, K. and Orguner, U. (2012). On the reduction of Gaussian inverse Wishart mixtures. *2012 15th International Conference on Information Fusion*, pages 2162–2169. 126

Grant, M. and Boyd, S. (2008). Graph implementations for nonsmooth convex programs. In Blondel, V., Boyd, S., and Kimura, H., editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited. 18, 19

Grant, M. C. (2005). *Disciplined convex programming*. PhD Thesis, Stanford University. 18, 19, 20, 36

Guhaniyogi, R. and Banerjee, S. (2018). Meta-Kriging: Scalable Bayesian Modeling and Inference for Massive Spatial Datasets. *Technometrics*, 60(4):430–444. 28, 32, 36, 45, 48, 68, 88, 111

Guhaniyogi, R. and Banerjee, S. (2019). Multivariate spatial meta kriging. *Statistics & Probability Letters*, 144:3–8. 32, 46

Gupta, A. K. and Nagar, D. K. (2000). *Matrix variate distributions*. Monographs and surveys in pure and applied mathematics. Chapman & Hall/CRC, Boca Raton. 6, 39, 104

Haque, M. A., Reza, M. N., Ali, M., Karim, M. R., Ahmed, S., Lee, K.-D., Khang, Y. H., and Chung, S.-O. (2024). Effects of Environmental Conditions on Vegetation Indices from Multispectral Images: A Review. *Korean Journal of Remote Sensing*, 40(4):319–341. 113

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs]. 40

Heaton, M. J., Datta, A., Finley, A. O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D. M., Katzfuss, M., Lindgren, F., Nychka, D. W., Sun, F., and Zammit-Mangion, A. (2017). A Case Study Competition Among Methods for Analyzing Large Spatial Data. *Journal of Agricultural, Biological, and Environmental Statistics*, 24(3):398–425. 3

Hefley, T. J., Hooten, M. B., Hanks, E. M., Russell, R. E., and Walsh, D. P. (2017). Dynamic spatio-temporal models for spatial data. *Spatial Statistics*, 20:206–220. 2

Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. (2022). Meta-Learning in Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169. 20

Houghton, J. (2005). Global warming. *Reports on Progress in Physics*, 68(6):1343. 80

Idjigbèrou, S. E., Assédé, E. S. P., Biaou, S., Gouwakinnou, G. N., Natta, A. K., and Biaou, S. S. H. (2025). Spatio-temporal dynamics of Isoberlinia-dominated woodlands in disturbance-prone landscapes over 15 years. *Global Ecology and Conservation*, 59:e03512. 10

Iranmanesh, A., Arashi, M., and Tabatabaey, S. M. M. a. (2010). On Conditional Applications of Matrix Variate Normal Distribution. *Iranian Journal of Mathematical Sciences and Informatics*, 5(2):33–43. 104

Ivanovic, B. and Pavone, M. (2019). The Trajectron: Probabilistic Multi-Agent Trajectory Modeling With Dynamic Spatiotemporal Graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 10

Jiang, Y., Zhou, L., and Raghavendra, A. (2020). Observed changes in fire patterns and possible drivers over central africa. *Environmental Research Letters*, 15(9):0940b8. 51

Jiménez, J. C. and Pereira, C. A. d. B. (2021). Assessing dynamic effects on a Bayesian matrix-variate dynamic linear model: An application to task-based fMRI data analysis. *Computational Statistics & Data Analysis*, 163:107297. 4, 61

Justice, C. O., Townshend, J. R. G., Holben, B. N., and Tucker, C. J. (1985). Analysis of the phenology of global vegetation using meteorological satellite data. *International Journal of Remote Sensing*, 6(8):1271–1318. 113

Kalinowski, T., Falbel, D., Allaire, J. J., Chollet, F., RStudio, Google, Tang [ctb, Y., cph, Bijl, W. V. D., Studer, M., and Keydana, S. (2024). keras: R Interface to 'Keras'. 40

Katzfuss, M. (2017). A Multi-Resolution Approximation for Massive Spatial Datasets. *Journal of the American Statistical Association*, 112(517):201–214. 3

Katzfuss, M. and Guinness, J. (2021). A General Framework for Vecchia Approximations of Gaussian Processes. *Statistical Science*, 36(1):124–141. 3

Kennedy, L., Vehtari, A., and Gelman, A. (2024). Model validation for aggregate inferences in out-of-sample prediction. *arXiv preprint*. arXiv:2312.06334 [stat.ME]. 62

Khasnis, A. A. and Nettleman, M. D. (2005). Global Warming and Infectious Disease. *Archives of Medical Research*, 36(6):689–696. 80

King, B. and Kowal, D. R. (2021). Warped Dynamic Linear Models for Time Series of Counts. 3

Knüppel, M. and Krüger, F. (2022). Forecast uncertainty, disagreement, and the linear pool. *Journal of Applied Econometrics*, 37(1):23–41. 45

Lemos, R. T., , and Sansó, B. (2009). A Spatio-Temporal Model for Mean, Anomaly, and Trend Fields of North Atlantic Sea Surface Temperature. *Journal of the American Statistical Association*, 104(485):5–18. 3

Lindgren, F., Rue, H., and Lindström, J. (2011). An Explicit Link between Gaussian Fields and Gaussian Markov Random Fields: The Stochastic Partial Differential Equation Approach. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(4):423–498. 3

Mahmoudian, B. and Mohammadzadeh, M. (2014). A spatio-temporal dynamic regression model for extreme wind speeds. *Extremes*, 17(2):221–245. 2

McAlinn, K., , Knut Are, A., , Jouchi, N., , and West, M. (2020). Multivariate Bayesian Predictive Synthesis in Macroeconomic Forecasting. *Journal of the American Statistical Association*, 115(531):1092–1110. 4, 16

McAlinn, K. and West, M. (2019). Dynamic Bayesian predictive synthesis in time series forecasting. *Journal of Econometrics*, 210(1):155–169. 4, 16

Mezić, I. (2005). Spectral Properties of Dynamical Systems, Model Reduction and Decompositions. *Nonlinear Dynamics*, 41(1):309–325. 3

Microsoft and Weston, S. (2022). *Foreach: Provides foreach looping construct*. Microsoft. 40

Microsoft, C. and Weston, S. (2022). *doParallel: Foreach parallel adaptor for the 'parallel' Package*. Microsoft. 40

Minsker, S., Srivastava, S., Lin, L., and Dunson, D. B. (2017). Robust and Scalable Bayes via a Median of Subset Posterior Measures. *Journal of Machine Learning Research*, 18(124):1–40. 36, 47, 88

Nicholls, N. (1989). Sea surface temperatures and australian winter rainfall. *Journal of Climate*, 2(9):965–973. 51, 110, 113

Nobre, A. A., Schmidt, A. M., and Lopes, H. F. (2005). Spatio-temporal models for mapping the incidence of malaria in Pará. *Environmetrics*, 16(3):291–304. 2, 10

Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., and Sain, S. (2015). A Multiresolution Gaussian Process Model for the Analysis of Large Spatial Datasets. *Journal of Computational and Graphical Statistics*, 24(2):579–599. 3

O'Carroll, A. G., Armstrong, E. M., Beggs, H. M., Bouali, M., Casey, K. S., Corlett, G. K., Dash, P., Donlon, C. J., Gentemann, C. L., Høyer, J. L., Ignatov, A., Kabobah, K., Kachi, M., Kurihara, Y., Karagali, I., Maturi, E., Merchant, C. J., Marullo, S., Minnett, P. J., Pennybacker, M., Ramakrishnan, B., Ramsankaran, R., Santoleri, R., Sunder, S., Saux Picart, S., Vázquez-Cuervo, J., and Wimmer, W. (2019). Observational needs of sea surface temperature. *Frontiers in Marine Science*, 6(420):1–27. 51, 110, 113

O'Donoghue, B., Chu, E., Parikh, N., and Boyd, S. (2016). Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068. 36

Pan, S., Zhang, L., Bradley, J. R., and Banerjee, S. (2025). Bayesian Inference for Spatial-Temporal Non-Gaussian Data Using Predictive Stacking. arXiv:2406.04655 [stat]. 4, 5, 19, 55

Paul-Christian Bürkner, J. G. and Vehtari, A. (2020). Approximate leave-future-out cross-validation for Bayesian time series models. *Journal of Statistical Computation and Simulation*, 90(14):2499–2523. 19, 62

Pherwani, P., Hass, N., and Yanchenko, A. (2024). Spatiotemporal Modeling and Forecasting at Scale with Dynamic Generalized Linear Models. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Geospatial Anomaly Detection*, GeoAnomalies '24, pages 16–27, New York, NY, USA. Association for Computing Machinery. event-place: Atlanta, GA, USA. 10

Pielke, R. A., Landsea, C., Mayfield, M., Layer, J., and Pasch, R. (2005). Hurricanes and Global Warming. *Bulletin of the American Meteorological Society*. 80

Presicce, L. and Banerjee, S. (2024). Bayesian Transfer Learning for Artificially Intelligent Geospatial Systems: A Predictive Stacking Approach. *arXiv preprint*. arXiv:2410.09504 [stat.ME]. 4, 5, 10, 19, 55, 65, 68, 82

Quintana, J. M. and West, M. (1987). An Analysis of International Exchange Rates Using Multivariate DLM's. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 36(2/3):275–281. 4, 11

Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A Unifying View of Sparse Approximate Gaussian Process Regression. *Journal of Machine Learning Research*, 6(65):1939–1959. 3

Ren, Q., Banerjee, S., Finley, A. O., and Hodges, J. S. (2011). Variational Bayesian methods for spatial data analysis. *Computational Statistics & Data Analysis*, 55(12):3197–3217. 3

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian Inference for Latent Gaussian models by using Integrated Nested Laplace Approximations. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 71(2):319–392. 3

Rue, H., Riebler, A., Sørbye, S. H., Illian, J. B., Simpson, D. P., and Lindgren, F. K. (2017). Bayesian Computing with INLA: A Review. *Annual Review of Statistics and Its Application*, 4(Volume 4, 2017):395–421. 3

Ruiz Maraggi, L. M., Lake, L. W., and Walsh, M. P. (2021). Using Bayesian Leave-One-Out and Leave-Future-Out Cross-Validation to Evaluate the Performance of Rate-Time Models to Forecast Production of Tight-Oil Wells. In *Proceedings of the 9th Unconventional Resources Technology Conference*, volume Day 2 Tue, July 27, 2021 of *SPE/AAPG/SEG Unconventional Resources Technology Conference*, page D021S027R003. 62

Sainsbury-Dale, M., Zammit-Mangion, A., Richards, J., and Huser, R. (2024). Neural Bayes Estimators for Irregular Spatial Data using Graph Neural Networks. arXiv:2310.02600. 89

Sandefur, J. (1990). *Discrete Dynamical Systems: Theory and Applications*. Clarendon Press. 10

Sansó, B., Schmidt, A. M., and Nobre, A. A. (2008). Bayesian spatio-temporal models based on discrete convolutions. *Canadian Journal of Statistics*, 36(2):239–258. 3

Sauer, A., Cooper, A., and Gramacy, R. B. (2023). Vecchia-Approximated Deep Gaussian Processes for Computer Experiments. *Journal of Computational and Graphical Statistics*, 32(3):824–837. 3

Schmidt, A. M. and Lopes, H. F. (2019). Dynamic models. In *Handbook of Environmental and Ecological Statistics*, pages 57–80. Chapman & Hall. 2, 4, 10, 11, 12, 15

Scott, S. L., Alexander W. Blocker, F. V. B., Chipman, H. A., George, E. I., and McCulloch, R. E. (2016). Bayes and big data: the consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management*, 11(2):78–88. 28

Sellers, P. J. (1985). Canopy reflectance, photosynthesis and transpiration. *International Journal of Remote Sensing*, 6(8):1335–1372. 113, 116

Smith, P., Fang, C., Dawson, J. J. C., and Moncrieff, J. B. (2008). Impact of Global Warming on Soil Organic Carbon. In *Advances in Agronomy*, volume 97, pages 1–43. Academic Press. 80

Snijders, T. A. B. (1988). On Cross-Validation for Predictor Evaluation in Time Series. In Dijkstra, T. K., editor, *On Model Uncertainty and its Statistical Implications*, pages 56–69, Berlin, Heidelberg. Springer Berlin Heidelberg. 19, 20

Stein, M. L. (1988). Asymptotically Efficient Prediction of a Random Field with a Misspecified Covariance Function. *The Annals of Statistics*, 16(1):55–63. 44, 80

Stein, M. L. (1999). *Interpolation of spatial data*. Springer Series in Statistics. Springer-Verlag, New York. 2

Stein, M. L. and Handcock, M. S. (1989). Some asymptotic properties of kriging when the covariance function is misspecified. *Mathematical Geology*, 21:171–190. 44, 80

Suder, P. M., Xu, J., and Dunson, D. B. (2023). Bayesian Transfer Learning. arXiv:2312.13484 [stat]. 4, 20, 21, 22

Tallman, E. and West, M. (2023). Bayesian predictive decision synthesis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(2):340–363. 5, 19

Tang, W., Zhang, L., and Banerjee, S. (2021). On identifiability and consistency of the nugget in Gaussian spatial process models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 83(5):1044–1070. 4, 80

Tucker, C. J. (1979). Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sensing of Environment*, 8(2):127–150. 113, 116

Vanschoren, J., Kotthoff, L., and Hutter, F., editors (2019). *Automated Machine Learning: Methods, Systems, Challenges*. The Springer Series on Challenges in Machine Learning. Springer International Publishing, Cham. 20

Vecchia, A. V. (1988). Estimation and Model Identification for Continuous Spatial Processes. *Journal of the Royal Statistical society, Series B*, 50:297–312. 3

Vehtari, A., Gelman, A., and Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27(5):1413–1432. 62

Vehtari, A., Mononen, T., Tolvanen, V., Sivula, T., and Winther, O. (2016). Bayesian Leave-One-Out Cross-Validation Approximations for Gaussian Latent Variable Models. *Journal of Machine Learning Research*, 17(103):1–38. 62

Walther, S., Guanter, L., Heim, B., Jung, M., Duveiller, G., Wolanin, A., and Sachs, T. (2018). Assessing the dynamics of vegetation productivity in circumpolar regions with different satellite indicators of greenness and photosynthesis. *Biogeosciences*, 15(20):6221–6256. 51

West, M. and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models (Springer Series in Statistics)*. Springer-Verlag. 2, 4, 10, 12, 15, 56

Wikle, C. K. (2010). Low-Rank Representations for Spatial Processes. *Handbook of Spatial Statistics*, pages 107–118. 3

Wikle, C. K. and Hooten, M. B. (2010). A general science-based framework for dynamical spatio-temporal models. *TEST*, 19(3):417–451. 3, 4

Wikle, C. K., Zammit Mangion, A., and Cressie, N. A. C. (2019). *Spatio-temporal statistics with R*. Chapman & Hall/CRC the R series. CRC Press, Boca Raton London New York. 2

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5(2):241–259. 4, 16

Wu, L., Pleiss, G., and Cunningham, J. P. (2022). Variational nearest neighbor Gaussian process. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 24114–24130. PMLR. 3

Xu, K. and Wikle, C. K. (2007). Estimation of parameterized spatio-temporal dynamic models. *Journal of Statistical Planning and Inference*, 137(2):567–588. 3

Yao, Y., Vehtari, A., Simpson, D., and Gelman, A. (2018). Using Stacking to Average Bayesian Predictive Distributions (with Discussion). *Bayesian Analysis*, 13(3):917–1007. 5, 16, 18, 19, 20, 35, 38, 39, 55, 61, 62, 89

Zammit-Mangion, A., Ng, T. L. J., Vu, Q., and Filippone, M. (2022). Deep Compositional Spatial Models. *Journal of the American Statistical Association*, 117(540):1787–1808. 3

Zammit-Mangion, A., Sainsbury-Dale, M., and Huser, R. (2024). Neural Methods for Amortised Inference. _eprint: 2404.12484. 40, 89

Zammit-Mangion, A. and Wikle, C. K. (2020). Deep integro-difference equation models for spatio-temporal forecasting. *Spatial Statistics*, 37:100408. 10

Zhang, H. (2004). Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *Journal of the American Statistical Association*, 99(465):250–261. 3, 4, 30, 44, 80

Zhang, J., Xue, F., Xu, Q., Lee, J.-A., and Qu, A. (2024a). Individualized Dynamic Model for Multi-resolutional Data with Application to Mobile Health. arXiv:2311.12392 [stat] version: 3. 3

Zhang, L. and Banerjee, S. (2022). Spatial factor modeling: A Bayesian matrix-normal approach for misaligned data. *Biometrics*, 78(2):560–573. 52, 55, 82, 88, 111

Zhang, L., Banerjee, S., and Finley, A. O. (2021). High-dimensional multivariate geostatistics: A Bayesian matrix-normal approach. *Environmetrics*, 32(4):e2675. 47

Zhang, L., Tang, W., and Banerjee, S. (2024b). Bayesian Geostatistics Using Predictive Stacking. 4, 5, 19, 31, 108

Appendix A

Chapter 2 appendix

A.1 Distribution theory

A.1.1 Posterior and predictive matrix-variate t distributions

The joint posterior predictive for $Y_{\mathcal{U}}$ and the unobserved latent process $\Omega_{\mathcal{U}}$, can be recast by integrating out $\{\gamma, \Sigma\}$ from the joint conditional posterior predictive, that is

$$p\left(Y_{\mathcal{U}}, \Omega_{\mathcal{U}} \mid \mathscr{D}\right) = \int \mathrm{MN}_{n',q}\left(Y_{\mathcal{U}} \mid X_{\mathcal{U}}\beta + \Omega_{\mathcal{U}}, \left(\alpha^{-1}-1\right)\mathbb{I}_{n'}, \Sigma\right) \times \mathrm{MN}_{n',q}\left(\Omega_{\mathcal{U}} \mid M_{\mathcal{U}}\Omega, V_{\Omega_{\mathcal{U}}}, \Sigma\right) \times \mathrm{MNIW}\left(\gamma, \Sigma \mid \mu_{\gamma}^{\star}, V_{\gamma}^{\star}, \Psi^{\star}, \nu^{\star}\right) d\gamma d\Sigma,$$

where $M_{\mathcal{U}} = \rho_{\phi}(\mathcal{U}, \mathcal{S})\rho_{\phi}^{-1}(\mathcal{S}, \mathcal{S})$ and $V_{\Omega_{\mathcal{U}}} = \rho_{\phi}(\mathcal{U}, \mathcal{U}) - \rho_{\phi}(\mathcal{U}, \mathcal{S})\rho_{\phi}^{-1}(\mathcal{S}, \mathcal{S})\rho_{\phi}(\mathcal{S}, \mathcal{U})$. We derive $p(\Omega_{\mathcal{U}}, Y_{\mathcal{U}} \mid \Sigma, \mathscr{D})$ by avoiding direct integration with respect to γ using the following augmented linear system

$$\underbrace{\begin{bmatrix}\Omega_{\mathcal{U}} \\ Y_{\mathcal{U}}\end{bmatrix}}_{\Upsilon} = \underbrace{\begin{bmatrix}0_{n'\times q} & M_{\mathcal{U}} \\ X_{\mathcal{U}} & M_{\mathcal{U}}\end{bmatrix}}_{M} \underbrace{\begin{bmatrix}\beta \\ \Omega\end{bmatrix}}_{\gamma} + \underbrace{\begin{bmatrix}E_{Yu} \\ E_{\Omega u}\end{bmatrix}}_{E}, \quad E \sim \mathrm{MN}_{2n',q}\left(0_{2n'\times q}, V_E, \Sigma\right) \ , \tag{A.1}$$

where $V_E = \begin{bmatrix} V_{\Omega_{\mathcal{U}}} & V_{\Omega_{\mathcal{U}}} \\ V_{\Omega_{\mathcal{U}}} & (\alpha^{-1}-1)\mathbb{I}_{n'} + V_{\Omega_{\mathcal{U}}} \end{bmatrix}$. We write the posterior distribution $p(\gamma \mid \mathscr{D})$ in Equation (2.4) as a linear equation, $\gamma = \mu_{\gamma}^{\star} + E_{\gamma}$, with $E_{\gamma} \sim \mathrm{MN}_{n',q}(O, V_{\gamma}^{\star}, \Sigma)$, where E and E_{γ} are independent of each other. Then,

$$\Upsilon = M\mu_{\gamma}^{\star} + ME_{\gamma} + E \sim \mathrm{MN}_{2n',q}\left(M\mu_{\gamma}^{\star}, V^{\star}, \Sigma\right), \tag{A.2}$$

where $V^{\star} = MV_{\gamma}^{\star}M^{\top} + V_E$. This yields $p(\Upsilon \mid \Sigma, \mathscr{D}) = p(\Omega_{\mathcal{U}}, Y_{\mathcal{U}} \mid \Sigma, \mathscr{D})$ as the closed-form joint predictive distribution by integrating out Σ from $p(\Omega_{\mathcal{U}}, Y_{\mathcal{U}} \mid \Sigma, \mathscr{D})p(\Sigma \mid \mathscr{D})$ to get

$$\int \mathrm{MNIW}(\Omega_{\mathcal{U}}, Y_{\mathcal{U}}, \Sigma \mid M\mu_{\gamma}^{\star}, V^{\star}, \Psi^{\star}, \nu^{\star})\, d\Sigma = \mathrm{T}_{2n',q}(\nu^{\star}, M\mu_{\gamma}^{\star}, V^{\star}, \Psi^{\star})$$

which is a matrix-variate Student's t random variable. Defining $\Upsilon = [\Omega_{\mathcal{U}}^{\mathrm{T}}, Y_{\mathcal{U}}^{\mathrm{T}}]^{\mathrm{T}}$, as a matrix of dimension $m \times q$, where $m = 2n'$, the predictive distribution is

$$p(\Upsilon \mid Y) = \int P(\Upsilon, \Sigma \mid Y) d\Sigma \,. \tag{A.3}$$

This matrix-variate integral can be avoided by simply writing

$$p(\Upsilon \mid Y) = \frac{p(\Upsilon, \Sigma \mid Y)}{p(\Sigma \mid \Upsilon, Y)}. \tag{A.4}$$

The density $p(\Upsilon, \Sigma \mid Y)$ comes from Equation (A.2), while the denominator is obtained as

$$\begin{aligned} p(\Sigma \mid \Upsilon, Y) &= \frac{p(\Sigma \mid Y) p(\Upsilon \mid \Sigma, Y)}{p(\Upsilon \mid Y)} \\ &\propto \frac{|\Psi^\star|^{\frac{\nu^\star}{2}} \left|V_\gamma^\star\right|^{-\frac{q}{2}} |\Sigma|^{-\frac{\nu^\star+m+q+1}{2}}}{2^{\frac{(\nu^\star+m)q}{2}} (\pi)^{\frac{mq}{2}} \Gamma_q\left(\frac{\nu^\star}{2}\right)} \exp\left\{-\frac{1}{2} \operatorname{tr}\left[\Sigma^{-1}\left(\Psi^\star + (\Upsilon - \mu^\star)^{\mathrm{T}} V_\gamma^{\star -1} (\Upsilon - \mu^\star)\right)\right]\right\} \\ &\propto |\Sigma|^{-\frac{\nu^\star+m+q+1}{2}} \exp\left\{-\frac{1}{2} \operatorname{tr}\left[\Sigma^{-1}\left(\Psi^\star + (\Upsilon - \mu^\star)^{\mathrm{T}} V_\gamma^{\star -1} (\Upsilon - \mu^\star)\right)\right]\right\}, \end{aligned}$$

where $\mu^\star = M\mu_\gamma^\star$. Hence, $\Sigma \mid \Upsilon, Y \sim \mathrm{IW}(\hat{\Psi}, \hat{\nu})$ with $\hat{\Psi} = \left(\Psi^\star + (\Upsilon - \mu^\star)^{\mathrm{T}} V_\gamma^{\star -1} (\Upsilon - \mu^\star)\right)$, and $\hat{\nu} = \nu^\star + m$. The joint posterior predictive density by the follows from Equation (A.4)

$$\begin{aligned} p(\Upsilon \mid Y) &= \frac{\mathrm{MNIW}(\Upsilon, \Sigma \mid \mu^\star, V_\gamma^\star, \Psi^\star, \nu^\star)}{\mathrm{IW}(\Sigma \mid \hat{\Psi}, \hat{\nu})} \\ &= K(\Upsilon) \left|\Psi^\star\right|^{-\frac{\nu^\star+m}{2}} \left|\mathbb{I}_m + V_\gamma^{\star -1} (\Upsilon - \mu^\star) \Psi^{\star -1} (\Upsilon - \mu^\star)^{\mathrm{T}}\right|^{-\frac{\nu^\star+m}{2}}, \end{aligned}$$

where $K(\Upsilon) = \frac{\Gamma_q\left(\frac{\hat{\nu}}{2}\right) |\Psi^\star|^{\frac{\nu^\star}{2}} \left|V_\gamma^\star\right|^{-\frac{q}{2}}}{\Gamma_q\left(\frac{\nu^\star}{2}\right) (\pi)^{\frac{mq}{2}}} = \frac{\Gamma_q\left(\frac{\nu^\star+m}{2}\right) |\Psi^\star|^{\frac{\nu^\star}{2}} \left|V_\gamma^\star\right|^{-\frac{q}{2}}}{\Gamma_q\left(\frac{\nu^\star}{2}\right) (\pi)^{\frac{mq}{2}}}$, since $\hat{\nu} = \nu^\star + m$.

This is a matrix-variate T density, which we denote as $\Upsilon \mid Y \sim \mathrm{T}_{m,q}(\nu^\star, \mu^\star, V^\star, \Psi^\star)$. We recover exactly the same result, without needing to integrate out Σ (Iranmanesh et al., 2010; Gupta and Nagar, 2000), but only using Bayes theorem and related distribution theory. Finally, the marginal predictive distributions $\Omega_{\mathcal{U}} \mid \mathscr{D}$, and $Y_{\mathcal{U}} \mid \mathscr{D}$, are also available in analytic form as matrix T distributions for any set of predictive points $\mathcal{U}$.

A.2 Asymptotic behaviors

The DOUBLE BPS extends the Bayesian predictive stacking to a transfer learning framework, as detailed in Section 2.2. Now, we assess the asymptotic behaviour of its approximation in terms of predictive distributions. Specifically, we investigate how the DOUBLE BPS approximations of posterior predictive distributions behave when the number of competitive models (J) and the number of partitions (K) grow towards infinity. We focus on the reversed Kullback-Leibler divergence between the DOUBLE BPS posterior predictive in (2.11) and the true one $p_t(\,\cdot \mid \mathscr{D})$,

defined in Equation (2.5) where $\{\alpha, \phi\}$ are those from the data-generating process.

By applying Jensen's inequality, we derive an upper bound for the reverse KL divergence; further details, including derivations and implementations, are supplied in Section A.2.1 and A.2.2. The resulting upper bound is as follows:

$$D_{KL}\big(\hat{P} \,\big\|\, P_t\big) \leq \log \prod_{k=1}^{K} \left\{ \sum_{k=1}^{K} \hat{w}_k \sum_{j=1}^{J} \hat{z}_{k,j}\, \mathbb{E}_{\hat{p}_{k,j}} \left[\frac{\sum_{j=1}^{J} \hat{p}(y \mid \mathscr{D}_k, \mathscr{M}_j)}{p_t(y \mid \mathscr{D})} \right] \right\}^{\hat{w}_k}, \tag{A.5}$$

where $\hat{P}$ and P_t denote probability measures with probability density $\hat{p}(\,\cdot\,; \mathscr{D})$, and $p_t(\,\cdot \mid \mathscr{D})$, respectively, and $\hat{p}_{k,j} = \hat{p}(\,\cdot \mid \mathscr{D}_k, \mathscr{M}_j)$.

The upper bound in (A.5) can be analyzed both analytically and empirically. In Section A.2.2, we provide a Monte Carlo study that approximates the expected value and explore how the bound varies with the number of subsets (K) and the number of candidate models (J).

A.2.1 Kullback-Leibler divergence from true posterior predictive

The true predictive distribution is defined as: $p_t(y \mid \mathscr{D}) = p(y \mid \mathscr{D}, \alpha_t, \phi_t) = \mathrm{T}(\nu_t, M_t, V_t, \Psi_t)$, while the DOUBLE BPS approximation is:

$$\hat{p}(y\,; \mathscr{D}) = \sum_{k=1}^{K} \hat{w}_k \sum_{j=1}^{J} \hat{z}_{k,j}\; p(y \mid \mathscr{D}_k, \alpha_j, \phi_j) = \sum_{k=1}^{K} \hat{w}_k \sum_{j=1}^{J} \hat{z}_{k,j}\;\; \mathrm{T}(y \mid \nu_{k,j}, M_{k,j}, V_{k,j}, \Psi_{k,j}).$$

Let $\hat{P}$ and P_t be the probability distributions corresponding to the DOUBLE BPS approximate and the true predictive probability distributions, respectively. The reverse KL divergences is given by:

$$D_{\mathrm{KL}}(\,\hat{P} \parallel P_t\,) = \int_y \log \frac{\hat{P}(\,\mathrm{d}y)}{P_t(\,\mathrm{d}y)} \hat{P}(\,\mathrm{d}y) = \int_{y \in \mathbb{R}^{n \times q}} \sum_{k=1}^{K} \hat{w}_k \hat{p}_k(y \mid \mathscr{D}_k) \log \frac{\sum_{k=1}^{K} \hat{w}_k \hat{p}_k(y \mid \mathscr{D}_k)}{p_t(y \mid \mathscr{D})}\, \mathrm{d}y,$$

where $\hat{p}_k(y\,; \mathscr{D}_k) = \sum_{j=1}^{J} \hat{z}_{k,j}\;\; \mathrm{T}(y \mid \nu_{k,j}, M_{k,j}, V_{k,j}, \Psi_{k,j})$. We can reformulate this Equation as the difference of two expectations:

$$\begin{aligned}
&\sum_{k=1}^{K} \hat{w}_k \mathbb{E}_{\hat{p}_k}\left[\log \sum_{k=1}^{K} \hat{w}_k \hat{p}_k(y\,; \mathscr{D}_k) \right] - \sum_{k=1}^{K} \hat{w}_k \mathbb{E}_{\hat{p}_k}\left[\log p_t(y \mid \mathscr{D}) \right] \\
&= \sum_{k=1}^{K} \hat{w}_k \mathbb{E}_{\hat{p}_k}\left[\log \frac{\sum_{k=1}^{K} \hat{w}_k \hat{p}_k(y\,; \mathscr{D}_k)}{p_t(y \mid \mathscr{D})} \right] \\
&\leq \sum_{k=1}^{K} \hat{w}_k \log \mathbb{E}_{\hat{p}_k}\left[\frac{\sum_{k=1}^{K} \hat{w}_k \hat{p}_k(y\,; \mathscr{D}_k)}{p_t(y \mid \mathscr{D})} \right],
\end{aligned}$$

where the inequality follows from Jensen's inequality. Expanding algebraically, this term, we get the more convenient formulation in Equation (A.5).

Algorithm 7 Approximating upper bound for Kullback-Liebler divergence

Input: Y outcomes matrix; X predictors matrix; $\hat{w} = \{\hat{w}_k : k = 1, \dots, K\}$: Stacking weights between subsets; $\hat{z} = \{\hat{z}_k = \{\hat{z}_{k,j}\} : k \in \{1, \dots, K\}, j \in \{1, \dots, J\}\}$: Stacking weights within subsets; $\hat{p}_{k,j}(\cdot)$, $p_t(\cdot)$ approximated and true predictive distributions $\forall j = 1, \dots, J, k = 1, \dots, K$; K: Number of subsets; J: number of competitive models in each subset; n: number of locations; L: number of Monte Carlo samples.
Output: $\widehat{ub}(n, K, J)$: approximated value for the upper bound, for a given set $\{n, K, J\}$.

for $k = 1, \dots, K$ **do**
  for $j = 1, \dots, J$ **do**
    Draw L samples $\{y_l : l = 1, \dots, L\}$ from $\hat{p}(\cdot \mid \mathscr{D}_k, \mathscr{M}_j)$
    for $l = 1, \dots, L$ **do**
      for $j = 1, \dots, J$ **do**
        Evaluate $p_{k,j,l} = \hat{p}(y_l \mid \mathscr{D}_k, \mathscr{M}_j)$
      end for
      Evaluate $p_{t,l} = p(y_l \mid \mathscr{D})$
      Compute $r_l = \frac{\sum_{j=1}^{J} z_{k,j}\, p_{k,j,l}}{p_{t,l}}$
    end for
    Compute $e_{k,j} = \frac{1}{L} \sum_{l=1}^{L} r_l$
  end for
end for
Compute $c_k = \sum_{k=1}^{K} \hat{w}_k \;\; \sum_{j=1}^{J} \hat{z}_{k,j}\; e_{k,j}$
return $\widehat{ub}(n, K, J) = \log \prod_{k=1}^{K} \; c_k^{\hat{w}_k}$

A.2.2 Monte Carlo approximation for upper bound simulations

We perform empirical investigations of the upper bound presented in Section A.2, and detailed in Section A.2.1, for different values of K, and J ceteris paribus.

We approach the problem of approximating the expectation in Equation (A.5) with a Monte Carlo integration. The approximation takes the form

$$\mathbb{E}_{\hat{p}_{k,j}}\left[\frac{\sum_{j=1}^{J} \hat{p}(y \mid \mathscr{D}_k, \mathscr{M}_j)}{p_t(y \mid \mathscr{D})}\right] \approx \frac{1}{L} \sum_{l=1}^{L}\left[\frac{\sum_{j=1}^{J} \hat{p}(y_l \mid \mathscr{D}_k, \mathscr{M}_j)}{p_t(y_l \mid \mathscr{D})}\right],$$

where $y_l \sim \hat{p}(y_l \mid \mathscr{D}_k, \mathscr{M}_j)$ for $l = 1, \dots, L$. We then devise the Algorithm 7 to approximate the upper bound for the KL divergence between the DOUBLE BPS posterior predictive and the true one.

To provide a meaningful interpolation, we consider 20 points for each parameter regulating $\widehat{ub}(n, K, J)$. We let vary $K \in \{5, 100\}$, $J \in \{2, 40\}$, while $n = 1000$ was fixed. Then, we remove data dependency by considering $M = 10$ replications for each evaluation setting. We perform different simulations for any of $\{K, J\}$ ceteris paribus, for the other. The panels in Figure A.1 shows how $D_{KL}(\hat{P} \parallel P_t)$ vary with K, and J, respectively.

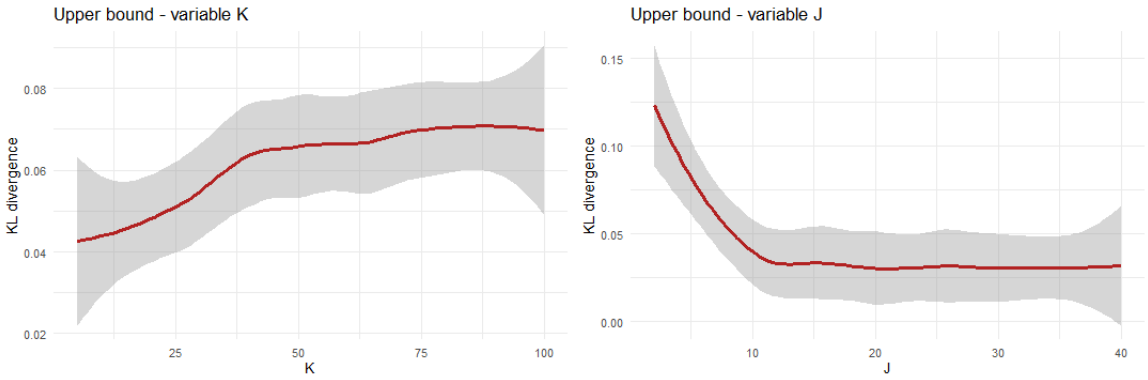

Figure A.1: Upper bound behave for growing values of K, and J

A.3 Accelerated learning for univariate spatial random fields

In Section 2.2, we propose a method to accelerate Bayesian learning for multivariate spatial random fields. While the primary focus is on multivariate geostatistical modeling, we also adapt this approach for simpler frameworks, specifically univariate random fields.

Let $\mathcal{S} = \{s_1, \ldots, s_n\} \subset \mathcal{D}$ denote a set of n locations yielding observations on an outcome $y(s)$ for each $s \in \mathcal{S}$. We collect these measurements into the vector $y = (y(s_i))^{\mathrm{T}}$, where $i = 1, \ldots, n$. Let $X = [x(s_i)^{\mathrm{T}}]$ represent an $n \times p$ matrix, with each row vector $x(s_i)^{\mathrm{T}}$ consisting of recorded measurements on $p < n$ explanatory variables at the location $s_i \in \mathcal{S}$. It is assumed that X has full column rank p. Such data can be modeled spatially using a customary Bayesian hierarchical model.

$$\begin{aligned} &y = X\beta + \omega + e_y\,, \quad e_y \sim \mathrm{N}(0, \delta^2\sigma^2\mathbb{I}_n)\,, \quad \omega \mid \sigma^2 \sim \mathrm{N}\left(0, \sigma^2\rho_\phi(\mathcal{S},\mathcal{S})\right)\,; \\ &\beta = M_0 m_0 + e_\beta\,, \quad e_\beta \sim \mathrm{N}(0, \sigma^2 M_0)\,; \quad \sigma^2 \sim \mathrm{IG}(a_\sigma, b_\sigma)\,, \end{aligned} \tag{A.6}$$

where y is $n \times 1$, X is an $n \times p$ matrix of explanatory variables, β is the $p \times 1$ vector of slopes measuring the trend, $\omega = (\omega(s_1), \ldots, \omega(s_n))^{\mathrm{T}}$ the finite-dimensional realization of the zero-centered spatial Gaussian process on $\mathbb{R}^d$, and let $\rho_\phi(\mathcal{S},\mathcal{S})$ be the $n \times n$ spatial correlation matrix constructed from the correlation function with spatial correlation function $\rho_\phi(\cdot,\cdot)$ depending on spatial range parameter ϕ, and σ^2 is a scale (spatial variance) parameter, and e_y is the zero-centered random vector of errors, with $\tau^2\mathbb{I}$ being the covariance matrix. Actually, Equation (A.6) is parametrized with the noise-to-spatial variance ratio $\delta^2 := \tau^2/\sigma^2$, instead of τ^2. We further model $\beta \mid \Sigma \sim \mathrm{N}(M_0 m_0, \sigma^2 M_0)$ and $\sigma^2 \sim \mathrm{IG}(a_\sigma, b_\sigma)$, where all the hyperparameters, including the spatial correlation parameters ϕ, and the noise-to-spatial variance ratio δ^2 are considered fixed. Fixing the parameters ϕ and δ^2 ensures closed-form conjugate marginal posterior and posterior predictive distributions for this hierarchical specification (Finley et al., 2019; Banerjee, 2020). The joint density of β and σ^2 is denoted by $\mathrm{NIG}(M_0 m_0, M_0, a_\sigma, b_\sigma)$. Harnessing familiar results from conjugate Bayesian linear regression, we cast the spatial model in Equation (A.6)

into the following augmented linear system:

$$\underbrace{\begin{bmatrix} y \\ M_0 m_0 \\ 0 \end{bmatrix}}_{y_\star} = \underbrace{\begin{bmatrix} X & \mathbb{I}_n \\ \mathbb{I}_p & 0 \\ 0 & \mathbb{I}_n \end{bmatrix}}_{X_\star} \underbrace{\begin{bmatrix} \beta \\ \omega \end{bmatrix}}_{\gamma} + \underbrace{\begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix}}_{\eta}, \quad \eta \sim \mathrm{N}(0, \sigma^2 V_\star),$$

where $V_\star = \begin{bmatrix} \delta^2\mathbb{I}_n & 0 & 0 \\ 0 & M_0 & 0 \\ 0 & 0 & \rho_\phi(\mathcal{S},\mathcal{S}) \end{bmatrix}$. For the new linear-system formulation of the hierarchical model presented in Equation (A.6), we derive a conjugate Bayesian model by considering the joint prior on the parameters $\{\gamma, \sigma^2\}$. This prior is denoted as $p\left(\gamma, \sigma^2 \mid M_0 m_0, M_0, a_\sigma, b_\sigma\right) \propto \mathrm{IG}\left(\sigma^2 \mid a_\sigma, b_\sigma\right)$. Given that we have conjugacy for any fixed values of $\{\phi, \delta^2\}$ and hyperparameters in the prior density, we can then obtain the posterior density.

$$p\left(\gamma, \sigma^2 \mid \mathscr{D}\right) = p\left(\sigma^2 \mid \mathscr{D}\right) p\left(\gamma \mid \sigma^2, \mathscr{D}\right) = \mathrm{IG}\left(\sigma^2 \mid a_\sigma^\star, b_\sigma^\star\right) \mathrm{N}\left(\gamma \mid \hat{\gamma}, \sigma^2 M_\star\right), \tag{A.7}$$

where $a_\sigma^\star = a_\sigma + n/2$, $b_\sigma^\star = b_\sigma + 1/2\left(y_\star - X_\star\hat{\gamma}\right)^{\mathrm{T}} V_\star^{-1}\left(y_\star - X_\star\hat{\gamma}\right)$, $M_\star^{-1} = X_\star^{\mathrm{T}} V_\star^{-1} X_\star$, and $\hat{\gamma} = M_\star X_\star^{\mathrm{T}} V_\star^{-1} y_\star$. The marginal posterior distribution $p(\gamma \mid \mathscr{D})$ is a multivariate Student's t with degrees of freedom $2a_\sigma^\star$, location $\hat{\gamma}$ and scale matrix $(b_\sigma^\star/a_\sigma^\star) M_\star$. We refer the reader to Zhang et al. (2024b, Lemma 1) for further details regarding this result.

Let $\mathcal{U} = \{u_1, \ldots, u_{n'}\} \in \mathcal{D}$ denote a set of n' unknown points. We define $\omega_\mathcal{U}$ and $y_\mathcal{U}$ as the $n' \times 1$ vectors with elements $\omega(u_i)$ and $y(u_i)$ for $i = 1, 2, \ldots, n'$. The matrix of predictors is given by $X_\mathcal{U} = (x(u_1) : \ldots : x(u_{n'}))^{\mathrm{T}}$ which is an $n' \times p$ matrix. Additionally, we denote the spatial correlation matrix at $\mathcal{U}$ as $\rho_\phi(\mathcal{U}, \mathcal{U})$. Then, spatial predictive inference follows from the posterior distribution

$$p(\omega_\mathcal{U}, y_\mathcal{U} \mid \mathscr{D}) = \int p\left(y_\mathcal{U} \mid \omega_\mathcal{U}, \beta, \sigma^2\right) p\left(\omega_\mathcal{U} \mid \omega, \sigma^2\right) p\left(\gamma, \sigma^2 \mid \mathscr{D}\right) d\gamma d\sigma^2, \tag{A.8}$$

which is again a multivariate t distribution with degrees of freedom $2a_\sigma^\star$, location $\tilde{\mu}$ and scale matrix $(b_\sigma^\star/a_\sigma^\star)\tilde{M}$ where $\tilde{\mu} = W\hat{\gamma}$, and $\tilde{M} = WM_\star W^{\mathrm{T}} + V_e$. With $V_\omega = \rho_\phi(\mathcal{U}, \mathcal{U}) - \rho_\phi(\mathcal{U}, \mathcal{S})\rho_\phi^{-1}(\mathcal{S}, \mathcal{S})\rho_\phi^{\mathrm{T}}(\mathcal{U}, \mathcal{S})$, and $M_\omega = \rho_\phi(\mathcal{U}, \mathcal{S})\rho_\phi^{-1}(\mathcal{S}, \mathcal{S})$, we can express W, and V_e, as

$$W = \begin{bmatrix} 0 & M_\omega \\ X_\mathcal{U} & M_\omega \end{bmatrix}, \qquad V_e = \begin{bmatrix} V_\omega & V_\omega \\ V_\omega & V_\omega + \delta^2\mathbb{I}_{n'} \end{bmatrix}. \tag{A.9}$$

The predictive distributions $p(\omega(u) \mid \mathscr{D})$ and $p\left(y(u) \mid \mathscr{D}\right)$ are also available in analytic form as non-central t distributions for any single point $u \in \mathcal{D}$. Bayesian inference can proceed from exact posterior samples obtained from Equation (A.7) as follows. We first draw values of $\sigma^2 \mid \mathscr{D} \sim \mathrm{IG}(a_\sigma^\star, b_\sigma^\star)$ followed by a single draw of $\gamma \mid \mathscr{D} \sim \mathrm{N}\left(\hat{\gamma}, \sigma^2 M_\star\right)$ for each drawn value of σ^2. This yields samples $\left\{\gamma, \sigma^2\right\}$ from Equation (A.7). Predictive inference for the latent process $\omega(u)$ and the outcome $y(u)$ is obtained by sampling from Equation (A.8) by drawing a value of $\omega_\mathcal{U} \mid \gamma, \sigma^2 \sim N\left(\mu_\omega(\gamma), \sigma^2 V_\omega\right)$ with $\mu_\omega(\gamma) := M_\omega\omega$, for each value of $\left\{\gamma, \sigma^2\right\}$ drawn above (see Section 3.4 in Banerjee (2020)), then drawing a value of $y_\mathcal{U} \mid \omega_\mathcal{U}, \gamma, \sigma^2 \sim N\left(X_\mathcal{U}\beta + \omega_\mathcal{U}, \sigma^2\delta^2\mathbb{I}_{n'}\right)$

for each drawn value of β (extracted from γ), σ^2 and $\omega_{\mathcal{U}}$.

Analogous to the multivariate case discussed in Section 2.2, we use the DOUBLE BPS model averaging procedure to achieve global inference. This method enables us to leverage conjugate frameworks by integrating out two typically problematic hyperparameters: $\{\phi, \delta^2\}$. For further details, see Banerjee (2020).

To implement Bayesian stacking of predictive densities within subsets, we first partition the full data into K subsets, represented as $\mathscr{D} = \{\mathscr{D}_1, \ldots, \mathscr{D}_K\}$. Next, we select a range of reasonable values for each hyperparameter. We then construct a grid that combines all possible pairs of hyperparameter values, resulting in a set of J competitive models: $\{\mathscr{M}_j\}_{j=1}^{J}$. For each model configuration $\mathscr{M}_j$, we follow the same scheme outlined in Section 2.2. The univariate setting does not alter the dimensions of the working quantities appearing in Algorithms 1 and 2, as the predictive densities are probability density functions, that always yield scalar values greater than zero. We then need to evaluate the predictive distributions $p\left(y_{k,i} \mid \mathscr{D}_{k,[-l]}, \mathscr{M}_j\right)$ for each of the J candidate models, where $i \notin [-l], k = 1, \ldots, K$, and $l = 1, \ldots, L$, with L representing the number of folds. Here $y_{k,i}$ is the i-th element of $y_K \in \mathscr{D}_k$, for $i \in [l]$, and $\mathscr{D}_{k,[-l]}$ is the full dataset excluding the l-th fold of units. As in the multivariate setting detailed in Section 2.2, closed-form conjugate posterior distributions are available. Considering results of Equation (A.8), we have $p\left(y_{k,i} \mid \mathscr{D}_{k,[-l]}, \mathscr{M}_j\right) = \mathrm{t}_{2a^{\star}_{[-l],\sigma}}(\tilde{\mu}_i, [b^{\star}_{[-l],\sigma}/a^{\star}_{[-l],\sigma}]\tilde{M}_i)$. Where $\tilde{\mu}_i = W_{y,i}\hat{\gamma}_{k,-[l]}$, $W_{y,i} = \left[X_{k,i} \quad \rho_{\phi_j}(\mathcal{S}_{k,i}, \mathcal{S}_{k,[-l]})\rho^{-1}_{\phi_j}(\mathcal{S}_{k,[-l]}, \mathcal{S}_{k,[-l]})\right]$ and $\tilde{M}_i = W_{y,i}M_{[-l],\star}W^{\mathrm{T}}_{y,i} + \delta^2_j + V_{\omega,i}$, $V_{\omega,i} = \rho_{\phi_j}(\mathcal{S}_{k,i}, \mathcal{S}_{k,i}) - \rho_{\phi_j}(\mathcal{S}_{k,i}, \mathcal{S}_{k,[-l]})\rho^{-1}_{\phi_j}(\mathcal{S}_{k,[-l]}, \mathcal{S}_{k,[-l]})\rho_{\phi_j}(\mathcal{S}_{k,[-l]}, \mathcal{S}_{k,i})$. $\mathcal{S}_{k,[-l]}, \mathcal{S}_{k,i}$ are the set of locations in k-th partition of data, but not in fold l-th, and the location of $y_{k,i}$, respectively.

When closed-form predictive distributions are unavailable, point-wise predictive densities can be obtained using a sampling scheme that recovers conditional posterior predictive samples, followed by Monte Carlo (MC) averaging for the density. To this end, we can implement the following approximation: $p(y_{k,i} \mid \mathscr{D}_{k,[-l]}, \mathscr{M}_j) \approx \frac{1}{S}\sum_{s=1}^{S} p\left(y_{k,i} \mid \gamma^{(s)}, \sigma^2_{(s)}, \mathscr{M}_j\right)$. Here, $p\left(y_{k,i} \mid \gamma, \sigma^2, \mathscr{M}_j\right) \sim N\left(X^{\mathrm{T}}_{k,i}\beta + \omega_i,\ \sigma^2\delta^2_j\right)$, since $\gamma = [\beta^{\mathrm{T}}\ \omega^{\mathrm{T}}]^{\mathrm{T}}$. The samples $\gamma^{(s)}, \sigma^2_{(s)}$ are drawn from $p\left(\gamma, \sigma^2 \mid \mathscr{D}_{k,[-l]}, \mathscr{M}_j\right)$ (see Equation (A.7)), for $s = 1, \ldots, S$ with S the number of posterior samples used for the MC density approximation. Even when closed-form results are available for the predictive distributions in Equation (A.8), the MC density approximation can significantly reduce computational burden. In fact, recovering the parameters for the marginal posterior predictive distributions may involve heavy matrix computations, rendering such approaches impractical in some contexts. Even when closed-form results are available for the predictive distributions in Equation (A.8), often MC density approximation may result in a computational advantage. Recovering the parameters for the marginal posterior predictive distributions may involve heavy matrix computations, which makes their recovery worthless in some contexts.

We will now consider the same quantities discussed in Section 1.4.1, but adapted for the univariate setting. The first step involves fitting the subset models, specifically performing DOUBLE BPS within the subsets. The within-staking weight solve convex optimization problem takes the form $\max_{z_k \in \mathcal{S}^J_1} \frac{1}{n_k}\sum_{i=1}^{n_k} \log \sum_{j=1}^{J} z_{k,j} p\left(y_{k,i} \mid \mathscr{D}_{k,[-l]}, \mathscr{M}_j\right)$. Accordingly, the between-

stacking weights are derived from the solution to the following problem:

$$\max_{w \in \mathcal{S}_1^K} \frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k \sum_{j=1}^{J} \hat{z}_{k,j} p\left(y_{k,i} \mid \mathscr{D}_{k,[-l]}, \mathscr{M}_j\right). \tag{A.10}$$

We can now derive the desired inferences. Following the multivariate framework in Section 2.2, we obtain the predictive and posterior distributions of $\theta = \{\gamma, \sigma^2\}$ for the complete dataset through the convex combinations of local inferences from subsets, weighted by DOUBLE BPS weights. Specifically $\hat{p}(y_{\mathcal{U}} \,;\mathscr{D}) = \sum_{k=1}^{K} \hat{w}_k \sum_{j=1}^{J} \hat{z}_{k,j}\; p\left(y_{\mathcal{U}} \mid \mathscr{D}_k, \mathscr{M}_j\right)$, and $\hat{p}(\theta \,;\mathscr{D}) = \sum_{k=1}^{K} \hat{w}_k\; \hat{p}(\theta \,;\mathscr{D}_k) = \sum_{k=1}^{K} \hat{w}_k \sum_{j=1}^{J} \hat{z}_{k,j}\; p\left(\theta \mid \mathscr{D}_k, \mathscr{M}_j\right)$, for $k = 1, \ldots, K$, and $j = 1, \ldots, J$.

In conclusion, the main difference between univariate and multivariate settings lies in the introduction of matrix-variate random variables. Although the Bayesian predictive stacking of predictive densities necessitates adaptations to the computations due to the tensors formed by these matrix random variables, the principles of the procedure remain consistent. A comprehensive understanding of the mechanisms within either framework is sufficient for effective implementation.

A.4 Application to NOAA data

One of the most discussed topics of the last decade is global warming and its critical consequences (see Fisher (1958); Nicholls (1989); Friehe et al. (1991); O'Carroll et al. (2019)). The main responsible of global warming are related to global temperature and atmospheric composition. For this reason, NASA and NOAA, developed a monitoring system based on Moderate Resolution Imaging Spectroradiometer (MODIS), and Advanced Very High Resolution Radiometer (AVHRR) platforms. Thus, different satellite-based collections of global-extended data are available and managed in accessible databases by national administration agencies. Sea surface temperature (SST) is crucial for environmental and climate scientists. Since SST is a determinant of the exchanges of vapor, and heat, from ocean to atmosphere. Here, we present an application of the DOUBLE BPS approach for univariate models, fully described in the Section A.3, indeed on sea surface temperature data. The aforementioned development has the scope to enable researchers to perform geostatistical analysis on massive spatial data sets, even with modest computational and memory resources. Thanks to the conjugate framework, together with the DOUBLE BPS to remove the dependence from nuisance hyperparameters δ^2, ϕ, Bayesian model-based inference for geostatistical regression is hastened. This is achieved without compromising the inferences. As an illustration, we analyze a data set composed of $1,002,500$ georeferenced observations of SST, collected within June 2022. To this end, we gather the dataset on the National Oceanic and Atmospheric Administration (NOAA) data server. We consider SST data that extend all over the world, for all the oceans. Within this massive data set, we use $n = 1,000,000$ observations for model fitting while the rest for predictive assessment. As explanatory variables, in order to gather the trend structure across the world, we consider the coordinates after a sinusoidal projection (an equal-area projection) and scaled to $1,000$ kilometers units, plus an intercept,

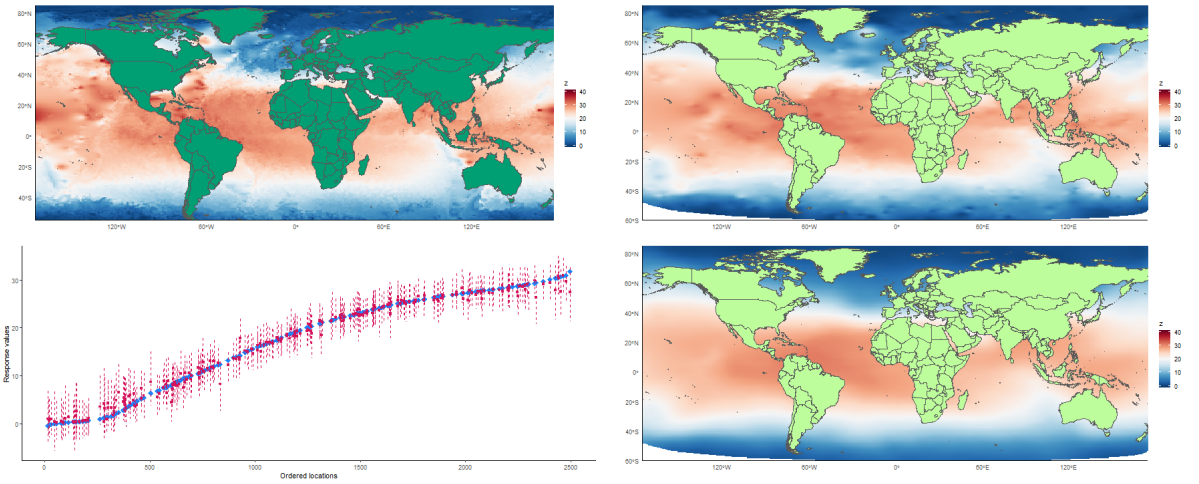

Figure A.2: from left to right: comparison between training (top left panel), test (top right panel), and predicted surface (bottom right panel). In addition, the empirical coverage for the response (bottom left panel). Results for $K = 2,000$.

leading to $p = 3$. For the spatial covariance function, we choose to use an exponential kernel. Prior choices follow standard choices, as specified in (2.3), and as considered in Banerjee (2017); Zhang and Banerjee (2022). As a matter of fact, we consider the NIG joint prior in Section A.3, where we fix as follow its parameters: $m_0 = 0_p$, $M_0 = 10\mathbb{I}_p$, $a_\sigma = 2$, and $b_\sigma = 2$. The set of hyperparameter values was defined by considering explanatory spatial data analysis and evaluating the empirical variogram to find insight about ϕ and δ^2, see Section A.5. Since the variogram suggests the absence of a nugget, we assign a near-zero value to the noise-to-spatial variance ratio, that is $\delta^2 = 10^{-6}$. While, again from EDA, we opt to the following values for range parameter $\phi \in \{0.0163, 0.0244, 0.0326\}$. Regarding the partition strategy, considering the result in Section 2.4.5, we take into account $K \in \{2,000\,, 4,000\}$ as the number of subsets, which correspond to the subset size of $n_k = \{500, 250\}$ units respectively. For both values of K, the subsets were created by random samples of observations all over the global ocean surface, avoiding block or clustering partition, as suggested in Guhaniyogi and Banerjee (2018). As in Section 2.5, we expanded the analysis by including competitive algorithms: nonspatial Bayesian conjugate linear model (BLM), distributed random forest (DRF), gradient boosting (GBM), deep neural network (DNN), and a fully automatic machine learning algorithm (AUTOML), in comparison to DOUBLE BPS (DBPS). We implement and perform the analysis with AI models, using the `h2o` R package (Fryda et al., 2024). These models are benchmarked in terms of predictive accuracy, parameter inference where applicable, and computational efficiency.

Figure A.2 displays a set of interpolated maps for $K = 2,000$. From the top left to the top right corners, we find the interpolation maps of the training and test SST data respectively. In these maps, as in the other, SST is represented in a shade of colors. Warmer colors correspond to higher sea surface temperatures; conversely, colder colors are shown, indicating lower temperatures. From the cartoon, the SST shows a clear spatial pattern. Indeed from the equator to the tropics, one can find high surface temperatures on average, while a drop in temperatures happens by moving toward the poles.

Besides the interpolations of training and test data, Figure A.2 also shows a third interpolation map: the bottom right corner of Figure A.2 shows the predicted posterior means

Parameter	Conjugate Linear model	DBPS ($K = 4,000$)	DBPS ($K = 2,000$)
β_0	34.495 (34.392, 34.601)	1.364 (-1.039, 3.907)	1.767 (-0.422, 3.917)
β_{long}	-2.708 (-2.719, -2.697)	0.744 (0.478, 0.998)	0.707 (0.485, 0.939)
β_{lat}	-2.708 (-2.719, -2.697)	0.744 (0.478, 0.998)	0.707 (0.485, 0.939)
σ^2	0.221 (0.220, 0.221)	0.182 (0.130, 0.257)	0.119 (0.099, 0.146)

Table A.1: Sea Surface Temperature data analysis parameter estimates for candidate models. Parameter posterior summary 50 (2.5, 97.5) percentiles.

interpolation, obtained from the DOUBLE BPS model with $K = 2,000$, for the withheld test locations. From Figure A.2, we claim the capability of the model to recover the true response surface. As commonly observed with meta-kriging approaches, a sort of over-smoothing is still present. Even if it could be seen as a shortcoming, it is attributed to a better generalization ability, avoiding overfitting phenomena. DOUBLE BPS also produces good uncertainty quantification results, despite the number of partitions. The bottom left corner of Figure A.2, shows the ordered withheld locations, and the predictive 95% credible intervals (complete of MAP estimates). Displaying a quite high coverage, corresponding to an empirical coverage of 99.4%. This disproves in this case the underestimation of the variability, often associated with meta-learning approaches. Looking to the right column in Figure A.2, the fitted sea surface temperature interpolated map over the test data, obtained from DOUBLE BPS, is almost indistinguishable from the true one. The performance achieved in the analysis is based on only 250 posterior predictive samples, since the conjugate framework. This was gained without using huge computational resources, but just a laptop, as detailed in Section 2.3.

In Table A.1 we illustrate a comparison between the posterior sample for β, σ^2, obtained by a Bayesian conjugate linear model, and DOUBLE BPS for $K \in \{4,000\ ,\ 2,000\}$. The posterior inferences vary consistently, passing from the non-spatial model to DOUBLE BPS. For the former, we observe inflated distribution for intercept β_0, and the variance σ^2, which show posterior distributions located over high values with respect to the range of response variable. Bayesian predictive stacking modeling provides more reasonable estimates. Moreover, for both values of K, the direction of association with the scaled longitude β_{long} results in an inversion concerning Bayesian conjugate modeling.

Lastly, Table A.2 presents the predictive performance and computation time for all models. The DOUBLE BPS with $K = 2,000$ achieves the best balance, with RMSPE of 1.165 and computation time of 63 minutes on a standard laptop. Notably, increasing the partition count to $K = 4,000$ reduces computation time to 16 minutes, with only minor degradation in accuracy (RMSPE of 1.556). This supports the theoretical scalability of DOUBLE BPS, with computational cost growing quadratically with subset size.

Among the machine learning models, the deep neural network (DNN) performs competitively with DOUBLE BPS, achieving an RMSPE of 1.112 in 18 minutes. Gradient boosting (GBM) also performs well (RMSPE of 1.304), while results for DRF and AUTOML are omitted due to computational limitations/instabilities. By contrast, the non-spatial BLM yields a substantially higher RMSPE of 9.855, confirming that spatial dependence cannot be effectively captured using coordinates alone in a linear model.

Overall, DOUBLE BPS offers an appealing compromise: delivering strong predictive accuracy

Model	Time (min)	RMSPE
DBPS ($K = 2,000$)	63	1.165
DBPS ($K = 4,000$)	16	1.556
Conjugate Linear Model	<1	9.855
GBM	<2	1.304
DRF	–	–
DNN	18	1.112
AUTOML	–	–

Table A.2: Sea Surface Temperature data analysis computing time in minutes, and RMSPE for candidate models.

and reliable uncertainty quantification, while requiring modest computational resources. Its ability to handle massive spatial datasets with limited hardware makes it a valuable tool for practitioners seeking scalable, Bayesian spatial inference.

A.5 Exploratory data analysis

A.5.1 MODIS exploratory data analysis

As already mentioned, in this work we analyse the vegetation index data gathered from the moderate-resolution imaging spectroradiometer (MODIS) data section. The National Aeronautics and Space Administration developed a monitoring system based on MODIS platforms, providing an extensive satellite-based collection of worldwide data. In Section 2.5, we already discussed the criticality of investigating global warming data (see Fisher, 1958; Nicholls, 1989; Friehe et al., 1991; O'Carroll et al., 2019) with the most advanced statistical and artificial intelligence tools assisting data-driven policies. We acquire global-extended data from MODIS' product denominated "MOD13C1.061 - Terra Vegetation Indices 16-Day L3 Global 0.05 Deg Climate Modeling Grid" (Didan, 2021). This product provides a per-pixel vegetation index. The Normalized Difference Vegetation Index (NDVI) is one of the main vegetation layers available, which maintains continuity with the NDVI derived from the National Oceanic and Atmospheric Administration's (NOAA) Advanced Very High Definition Radiometer (AVHRR). The climate modeling grid consists of 3600 rows and 7200 columns of 5600-meter pixels, corresponding to 0.05-degree modeling grid. The global MOD13C1 data are cloud-free spatial composites of the 16-day. MOD13C1 also contains data fields for reflectance data and angular information. Then we consider Red Reflectance (RR) as the second response variable, and solar zenith angle as a common predictor for multivariate outcomes.

Both the NDVI and the red reflectance indices are mainly related to the ability of vegetation to absorb solar radiation, throughout the photosynthetically active radiation, then used as energy source thanks to photosynthesis. Their scientific relevance is foundational in several disciplines as ecology, agriculture, remote sensing, and climate sciences (see Tucker, 1979; Sellers, 1985; Justice et al., 1985; Haque et al., 2024). Solar zenith angle (SZA) is the angle between the sun and the point directly overhead, i.e., the zenith. As the SZA affects how much sunlight reaches the surface, it gives insight into the strength of solar irradiation, which turns out to be extremely relevant for assessing biomass, e.g., vegetation indices.

The original dataset downloaded comprises georeferenced data for $1,500,000$ locations worldwide. However, we randomly selected $1,002,500$ locations, where $2,500$ were held out

Vegetation index	Mean	Std.Dev	Min.	Max.	Histogram	Boxplot
NDVI	8.593	0.517	6.909	9.469		
Red Reflectance	8.563	0.447	8.007	9.472		

Table A.3: Summary statistics and visual representation of response variables.

Σ_{NDVI}	Σ_{RR}	$\varrho_{\text{NDVI,RR}}$
0.2208 (0.2202, 0.2215)	0.1549 (0.1545, 0.1553)	-0.9049 (-0.9052, -0.9046)

Table A.4: Non-spatial association between response variables. 50 (2.5, 97.5) quantile estimates using Bayesian multivariate linear regression.

for evaluating predictive performances. Our spatially dependent outcomes were transformed on a logarithmic scale, such that we labeled log(NDVI + 1) as NDVI and log(RR + 1) as RR. All the data, including SZA, were collected over a 16-day window in May 2024 and then averaged. Table A.3 reports summary statistics and distribution visual representation for the outcomes. Among the observed locations, the maximum inter-site distance corresponds to approximately 42,909 kilometers.

We investigate non-spatial association between NDVI and red reflectance fitting the Bayesian multivariate regression model, defined by Equation (2.1). The model comprises two predictors: an intercept and the solar zenith angle for the million locations in the training set. More details on modeling and prior distribution are provided in Section 2.5, where a comparison of predictive performances is presented. Table A.4 reports quantile estimates for marginal variances, i.e., the diagonal elements of Σ, and the correlation between NDVI and RR. Strong negative values for correlation are estimated, showing an intense inverse relationship between the two spatially dependent outcomes.

Hereafter, we present results from the fully automated explanatory spatial data analyses that complement Sections 2.5. Variograms are employed in both analyses to extract "guidelines" on spatial parameters such as spatial variability proportion α, and spatial range ϕ, which are essential for setting up the DOUBLE BPS framework for GeoAI applications. Variogram fitting, used to gather parameter values required for DOUBLE BPS, is fully automated and requires no human intervention, except for specifying the grid width.

We firstly use independent sample variograms for NDVI and RR, based on 31,875 randomly sampled locations. For NDVI, the empirical variogram estimates the nugget 0.03, a sill of 0.27, and a practical range of approximately 88 based upon automated weighted least squares. This corresponds to significant spatial correlation up to about 10,000 kilometers. The proportion of spatial variability is computed as $\alpha = \sigma^2/(\tau^2 + \sigma^2) = 0.27/(0.03 + 0.27) \approx 0.9$, resulting in 0.909 without rounding. Finally, the spatial range parameter is estimated as $\phi = 0.067$ based upon the distance beyond which the spatial correlation drops to less than 0.05; see the left panel of Figure A.3.

For RR, the variogram parameters include a nugget effect of 0.04, a sill of 0.19, and a practical range of approximately 120, which corresponds to around 13,000 kilometers. The slightly higher nugget effect for RR suggests greater measurement error or micro-scale variability

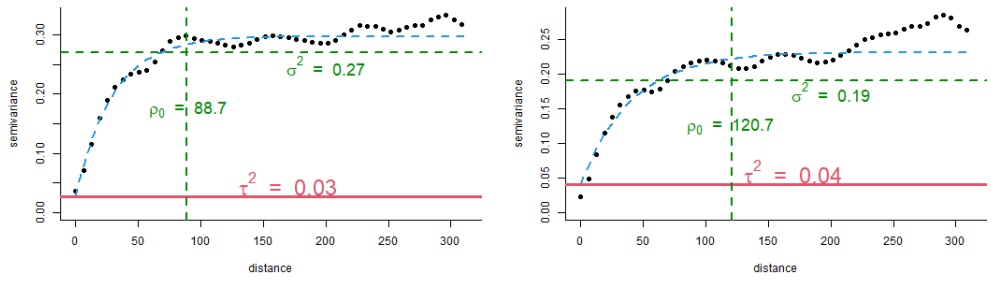

Figure A.3: From left to right: sample variograms of NDVI, and Red Reflectance.

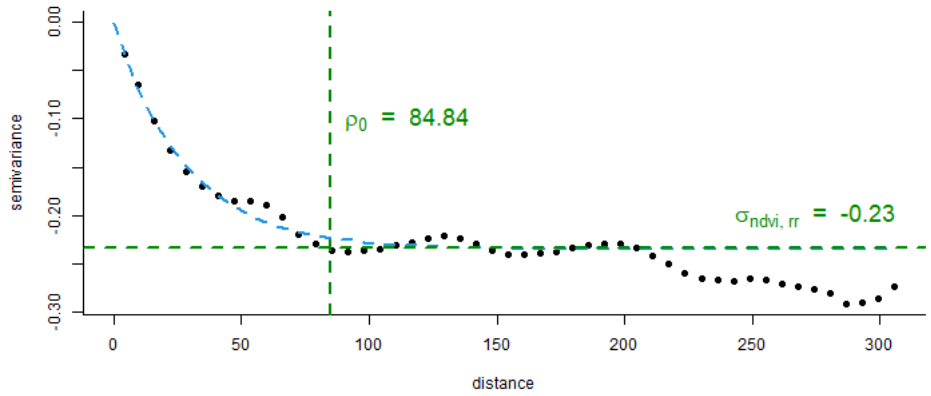

Figure A.4: Sample cross-variogram between NDVI, and Red Reflectance.

compared to NDVI. The proportion of spatial variability is estimated as $\alpha = 0.19/(0.04 + 0.19) \approx 0.825$. The practical range for RR is more extended than that of NDVI, indicating that RR values remain spatially correlated over a greater distance. Concluding the exploratory spatial data analysis, we select a spatial range of $\phi = 0.049$ for RR.

The combined analysis of the variograms for NDVI and RR provides essential information about the spatial variance proportion and range parameters, which are critical for informing artificially intelligent geospatial modeling systems. This analysis results in $\alpha \in \{0.825, 0.909\}$ and $\phi \in \{0.049, 0.067\}$. These findings help improve the accuracy of spatial predictions, enhancing ecological interpretations and increasing computational efficiency by avoiding excessive misspecified model specifications.

The entire explanatory analysis workflow, designed to gather critical insights for improving the DOUBLE BPS methodology, is fully automated. Human input is then minimized; the only required user feed is the number of grid values for each spatial parameter.

Using the same subsample composed of 31,875 locations, we also investigated the cross-variogram; see Figure A.4. This will help us extrapolate insights on spatial cross-dependencies.

The empirical cross-variogram depicts negative values, providing an estimate for the sill of -0.23, and a practical range of approximately 85 based upon automated weighted least squares. Similarly to individual variogram analysis, this shows a significant (negative) spatial correlation that withstands up to several thousand kilometers, suggesting a clear and well-

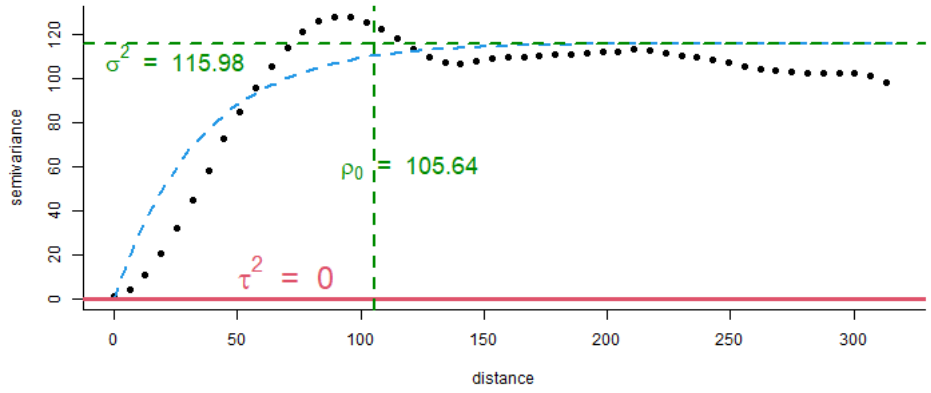

Figure A.5: Sample variogram of SST data.

defined negative spatial correlation structure between NDVI and red reflectance. Negative cross-variogram mirrors the negative correlation found using the non-spatial model (2.1). It is not surprising that the presence of strong (negative) spatial correlation among these indices, as their definition are strictly related, and both are based on spectral reflectance measurements acquired in the visible and near-infrared regions. Intuitively, the negative correlation emerged considering that healthy vegetation, which reflects high levels of biomass (NDVI), has strong chlorophyll absorption abilities, then revealing low red reflectance. Conversely, an increase in red reflectance corresponds to stressed (or low) vegetation, which results in low levels of the normalized difference vegetation index. Then non-spatial, and spatial negative associations occurred are totally coherent with the nature of these indices and, coherent with literature (Tucker, 1979; Sellers, 1985).

A.5.2 NOAA exploratory data analysis

In the second exploratory analysis, associated with Section A.4, we explore the spatial characteristics of the SST data from NOAA using a sample variogram. The variogram provides a deeper understanding of spatial dependence in the SST measurements, crucial for defining a plausible set of values for parameters such as the signal-to-noise ratio δ^2 and spatial range ϕ.

The empirical variogram, depicted in Figure A.5, shows the semi-variance as a function of distance, revealing key spatial statistics. For our SST data, based on 25,000 randomly sampled locations, the variogram indicates no nugget effect, $\tau^2 = 0$, a sill of 115.98, and a practical range of $\rho_0 = 105.64$. This implies that SST measurements become almost spatially uncorrelated beyond a distance of approximately 11,500 kilometers. The practical range, ρ_0, is the distance within which data points exhibit substantial spatial correlation. The absence of a nugget effect suggests no measurement error or micro-scale variability at distances approaching zero. In Section A.4, after testing several values in the interval $(10^{-8}, 10^{-1})$, we fix $\delta^2 = 10^{-6}$. The data exhibit extremely strong spatial dependence, as confirmed by Figure A.5, reinforcing the choice of this parameter. These findings provide a comprehensive view of the spatial structure, which is critical for subsequent geostatistical modeling and spatial predictions.

For the exponential spatial covariance parameter ϕ, we take advantage of empirical variogram estimation. A crucial consideration here is the "over-smoothing" often observed with DOUBLE BPS, which suggests that wider or doubled values of ϕ compared to variogram-based estimates may be more appropriate. In Section A.4, we expand the initial point estimate of ϕ by one-third, considering the set $\phi \in \{0.0163, 0.0244, 0.0326\}$ for further analysis.

Again, the entire explanatory analysis workflow, designed to gather critical insights for improving the DOUBLE BPS methodology, is fully automated. Human input is then minimized; the only required user feed is the number of grid values for each spatial parameter, and potential correction for "singular" values of the hyperparameter (as done for δ^2).

Appendix B

Chapter 3 appendix

B.1 Distribution theory

B.1.1 Kullback-leibler divergence between a finite mixture of matrix normal and a matrix normal

Let us consider P to be a finite mixture of matrix-variate Gaussians, and Q to be a single matrix-variate normal measure (representing the matrix normal distribution which borrows information between time instants), such that

$$\begin{aligned} P(X) &= \sum_{j=1}^{J} \pi_j \, \mathrm{MN}(X \mid M_j, V_j, U_j) \\ Q(X) &= \mathrm{MN}(X \mid M_q, V_q, U_q). \end{aligned} \tag{B.1}$$

Later on, we will consider U_j, U_q known for $j = 1, \ldots, J$. Then, we start to compute the $\mathrm{KL}(P \mid\mid Q)$ by substituting the $p(x), q(x)$ in Equation (3.3) with the density function of P, Q.

$$\mathrm{KL}(P \mid\mid Q) = \int \left(\sum_{j=1}^{J} \pi_j \, \mathrm{MN}(X \mid M_j, V_j, U_j) \right) \log \left(\frac{\sum_{j=1}^{J} \pi_j \, \mathrm{MN}(X \mid M_j, V_j, U_j)}{\mathrm{MN}(X \mid M_q, V_q, U_q)} \right) dX. \tag{B.2}$$

Then, proceeding from Equation (B.2), we distribute the integral as

$$\begin{aligned} &= \int \left(\sum_{j=1}^{J} \pi_j \, \mathrm{MN}(X \mid M_j, V_j, U_j) \right) \log \left(\sum_{j=1}^{J} \pi_j \, \mathrm{MN}(X \mid M_j, V_j, U_j) \right) dX - \\ &\qquad \int \left(\sum_{j=1}^{J} \pi_j \, \mathrm{MN}(X \mid M_j, V_j, U_j) \right) \log \left(\mathrm{MN}(X \mid M_q, V_q, U_q) \right) dX \\ &= \mathcal{I}_1 - \mathcal{I}_2. \end{aligned} \tag{B.3}$$

Thus, we now consider separately these two integrals, renamed $\mathcal{I}_1, \mathcal{I}_2$ respectively. Let us start from $\mathcal{I}_1$:

$$\begin{aligned}\mathcal{I}_1 &= \int \left(\sum_{j=1}^{J} \pi_j \,\mathrm{MN}(X \mid M_j, V_j, U_j)\right) \log\left(\sum_{j=1}^{J} \pi_j \,\mathrm{MN}(X \mid M_j, V_j, U_j)\right) dX \\ &= \mathbb{E}_X\left[\log P(X)\right] \qquad \text{with } X \sim \sum_{j=1}^{J} \pi_j \,\mathrm{MN}(X \mid M_j, V_j, U_j) \\ &= -\mathcal{H}(P), \end{aligned} \tag{B.4}$$

where $\mathcal{H}(P)$ represent the entropy associated to measure P. Then, we proceed to compute the second integral:

$$\begin{aligned}\mathcal{I}_2 &= \int \left(\sum_{j=1}^{J} \pi_j \,\mathrm{MN}(X \mid M_j, V_j, U_j)\right) \log\left(\mathrm{MN}(X \mid M_q, V_q, U_q)\right) dX \\ &= \sum_{j=1}^{J} \pi_j \Bigg\{ -\frac{nm}{2}\log 2\pi - \frac{m}{2}\log \mid V_q \mid -\frac{n}{2}\log \mid U_q \mid \\ &- \frac{1}{2}\int \mathrm{MN}(X \mid M_j, V_j, U_j)\,\mathrm{tr}[U_q^{-1}(X - M_q)^{\mathrm{T}} V_q^{-1}(X - M_q)] dX \Bigg\} \\ &= -\frac{nm}{2}\log 2\pi - \frac{m}{2}\log \mid V_q \mid -\frac{n}{2}\log \mid U_q \mid \\ &- \frac{1}{2}\sum_{j=1}^{J} \pi_j \int \mathrm{MN}(X \mid M_j, V_j, U_j)\,\mathrm{tr}[U_q^{-1}(X - M_q)^{\mathrm{T}} V_q^{-1}(X - M_q)] dX \\ &= -\frac{nm}{2}\log 2\pi - \frac{m}{2}\log \mid V_q \mid -\frac{n}{2}\log \mid U_q \mid -\frac{1}{2}\sum_{j=1}^{J} \pi_j \mathbb{E}_X\left[\mathrm{tr}[U_q^{-1}(X - M_q)^{\mathrm{T}} V_q^{-1}(X - M_q)]\right], \end{aligned} \tag{B.5}$$

with $X \sim \mathrm{MN}(X \mid M_j, V_j, U_j)$ for $j = 1, \ldots, J$. Then, we can now rely on the duality between matrix-variate and multivariate distributions. Indeed, we have that $\mathrm{MN}_{n,m}(M, V, U) \overset{d}{=} \mathrm{N}_{nm}(\mathrm{vec}(M), V \otimes U)$. Thus passing to a multivariate parametrization, we define $\mathrm{N}_{nm}(\mu_j, \Sigma_j)$, where $\mu_j := \mathrm{vec}(M_j)$, and $\Sigma_j := V_j \otimes U_j$, for $j = 1, \ldots, J$ and q. Once substituted, $\mathcal{I}_2$ evolves as

$$\begin{aligned} &= \int \left(\sum_{j=1}^{J} \pi_j \,\mathrm{N}(x \mid \mu_j, \Sigma_j)\right) \log\left(\mathrm{N}(x \mid \mu_q, \Sigma_q)\right) dx \\ &= -\frac{nm}{2}\log 2\pi - \frac{1}{2}\log \mid \Sigma_q \mid -\frac{1}{2}\sum_{j=1}^{J} \pi_j \mathbb{E}_x\left[(x - \mu_q)^{\mathrm{T}} \Sigma_q^{-1}(x - \mu_q)\right]. \end{aligned} \tag{B.6}$$

The expected value that appears in (B.6), must be considered with respect to $x \sim \mathrm{N}(\mu_j, \Sigma_j)$ now. Hence, we focus on this expectation:

$$\mathbb{E}_x\left[(x - \mu_q)^{\mathrm{T}} \Sigma_q^{-1}(x - \mu_q)\right] = \mathbb{E}_x\left[(x^{\mathrm{T}} \Sigma_q^{-1} x + \mu_q^{\mathrm{T}} \Sigma_q^{-1} \mu_q - 2\mu_q^{\mathrm{T}} \Sigma_q^{-1} x\right]. \tag{B.7}$$

Exploiting the so-called "trace trick", i.e. $\mu_q^\intercal \Sigma_q^{-1} \mu_q = \text{tr}(\mu_q^\intercal \Sigma_q^{-1} \mu_q) = \text{tr}(\Sigma_q^{-1} \mu_q \mu_q^\intercal)$, and the linearity of trace operator, such that $\mathbb{E}_x[\text{tr}(A)] = \text{tr}(\mathbb{E}_x[A])$, the computations boils down to

$$
\begin{aligned}
&= \text{tr}\left(\Sigma_q^{-1}\Sigma_j + \Sigma_q^{-1}\mu_j\mu_j^\intercal\right) + \mu_q^\intercal\Sigma_q^{-1}\mu_q - 2\mu_q^\intercal\Sigma_q^{-1}\mu_j \\
&= \text{tr}\left(\Sigma_q^{-1}\Sigma_j\right) + \mu_j^\intercal\Sigma_q^{-1}\mu_j + \mu_q^\intercal\Sigma_q^{-1}\mu_q - 2\mu_q^\intercal\Sigma_q^{-1}\mu_j \\
&= \text{tr}\left(\Sigma_q^{-1}\Sigma_j\right) + (\mu_j - \mu_q)^\intercal\Sigma_q^{-1}(\mu_j - \mu_q). \qquad \text{(B.8)}
\end{aligned}
$$

At this point, by inserting in Equation (B.6) the analytic form for the expectation derived in Equation (B.8), together with the definition of $\mu_j, \Sigma_j, \mu_q, \Sigma_q$, boils down to

$$
\begin{aligned}
\mathcal{I}_2 &= -\frac{nm}{2}\log 2\pi - \frac{1}{2}\log|\,\Sigma_q\,| - \frac{1}{2}\sum_{j=1}^{J}\pi_j\mathbb{E}_x\left[(x-\mu_q)^\intercal\Sigma_q^{-1}(x-\mu_q)\right] \\
&= -\frac{nm}{2}\log 2\pi - \frac{1}{2}\log|\,\Sigma_q\,| - \frac{1}{2}\sum_{j=1}^{J}\pi_j\left[\text{tr}\left(\Sigma_q^{-1}\Sigma_j\right) + (\mu_j-\mu_q)^\intercal\Sigma_q^{-1}(\mu_j-\mu_q)\right] \\
&= -\frac{nm}{2}\log 2\pi - \frac{1}{2}\log|\,V_q\otimes U_q\,| \\
&\qquad - \frac{1}{2}\sum_{j=1}^{J}\pi_j\left[\text{tr}\left([V_q\otimes U_q]^{-1}[V_j\otimes U_j]\right) + (\text{vec}(M)_j - \text{vec}(M)_q)^\intercal[V_q\otimes U_q]^{-1}(\text{vec}(M)_j - \text{vec}(M)_q)\right] \\
&= -\frac{nm}{2}\log 2\pi - \frac{1}{2}\log|\,V_q\,|^m|\,U_q\,|^n \\
&\qquad - \frac{1}{2}\sum_{j=1}^{J}\pi_j\left[\text{tr}\left([V_q^{-1}\otimes U_q^{-1}][V_j\otimes U_j]\right) + (\text{vec}(M)_j - \text{vec}(M)_q)^\intercal[V_q^{-1}\otimes U_q^{-1}](\text{vec}(M)_j - \text{vec}(M)_q)\right] \\
&= -\frac{nm}{2}\log 2\pi - \frac{1}{2}\log|\,V_q\,|^m - \frac{1}{2}\log|\,U_q\,|^n \\
&\qquad\qquad - \frac{1}{2}\sum_{j=1}^{J}\pi_j\left[\text{tr}\left(V_q^{-1}V_j\otimes U_q^{-1}U_j\right) + \text{vec}(M_j - M_q)^\intercal\,\text{vec}(V_q^{-1}(M_j-M_q)U_q^{-1})\right] \\
&= -\frac{nm}{2}\log 2\pi - \frac{m}{2}\log|\,V_q\,| - \frac{n}{2}\log|\,U_q\,| \\
&\qquad\qquad - \frac{1}{2}\sum_{j=1}^{J}\pi_j\left[\text{tr}\left(V_q^{-1}V_j\right)\text{tr}\left(U_q^{-1}U_j\right) + \text{tr}\left((M_j-M_q)^\intercal V_q^{-1}(M_j-M_q)U_q^{-1}\right)\right] \\
&= -\frac{nm}{2}\log 2\pi - \frac{m}{2}\log|\,V_q\,| - \frac{n}{2}\log|\,U_q\,| \\
&\qquad\qquad - \frac{1}{2}\sum_{j=1}^{J}\pi_j\left[\text{tr}\left(V_q^{-1}V_j\right)\text{tr}\left(U_q^{-1}U_j\right) + \text{tr}\left(U_q^{-1}(M_j-M_q)^\intercal V_q^{-1}(M_j-M_q)\right)\right].
\end{aligned}
\qquad \text{(B.9)}
$$

Wrapping up, we have that the KL divergence becomes

$$
\text{KL}(P\,||\,Q) = \int\left(\sum_{j=1}^{J}\pi_j\,\text{MN}(X\mid M_j, V_j, U_j)\right)\log\left(\frac{\sum_{j=1}^{J}\pi_j\,\text{MN}(X\mid M_j, V_j, U_j)}{\text{MN}(X\mid M_q, V_q, U_q)}\right)dX
$$

$$= -\mathcal{H}(P) + \frac{nm}{2}\log 2\pi + \frac{m}{2}\log \mid V_q \mid + \frac{n}{2}\log \mid U_q \mid + \frac{1}{2}\sum_{j=1}^{J}\pi_j\left[\operatorname{tr}\left(V_q^{-1}V_j\right)\operatorname{tr}\left(U_q^{-1}U_j\right) + \operatorname{tr}\left(U_q^{-1}(M_j - M_q)^{\mathrm{T}}V_q^{-1}(M_j - M_q)\right)\right]. \tag{B.10}$$

B.1.2 Closed-form optimal minimizer parameters for KL divergence between a finite mixture of matrix normal and a matrix normal

Now consider minimizing the KL divergence in Equation (B.10) with regard to the parameter of Q, i.e. M_q, V_q (since U_q is known), looking to characterize the matrix-variate Gaussian distribution which minimizes Kullback-Leibler divergence with P. Since it seeks into the Gaussian family, it will suffice to minimize Equation B.10 with respect to the unknown parameters. We are now looking for

$$\arg\min_{Q\in\mathbb{Q}} \mathrm{KL}(P \mid\mid Q) = \arg\min_{M_q\in\mathbb{R}^{n\times m}, V_q\in\mathbb{R}^{n\times n}} \mathrm{KL}(\sum_{j=1}^{J}\pi_j \mathrm{MN}(X \mid M_j, V_j, U_j) \mid\mid \mathrm{MN}(X \mid M_q, V_q, U_q)). \tag{B.11}$$

Let us start the minimization from M_q. First of all, we have to compute the partial derivative

$$\begin{aligned}
&\frac{\partial}{\partial M_q}\left\{ -\mathcal{H}(P) + \frac{nm}{2}\log 2\pi + \frac{m}{2}\log \mid V_q \mid + \frac{n}{2}\log \mid U_q \mid \right.\\
&\qquad\left. + \frac{1}{2}\sum_{j=1}^{J}\pi_j\left[\operatorname{tr}\left(V_q^{-1}V_j\right)\operatorname{tr}\left(U_q^{-1}U_j\right) + \operatorname{tr}\left(U_q^{-1}(M_j - M_q)^{\mathrm{T}}V_q^{-1}(M_j - M_q)\right)\right]\right\}\\
&= \frac{1}{2}\sum_{j=1}^{J}\pi_j\,\frac{\partial}{\partial M_q}\left\{\left[\operatorname{tr}\left(V_q^{-1}V_j\right)\operatorname{tr}\left(U_q^{-1}U_j\right) + \operatorname{tr}\left(U_q^{-1}(M_j - M_q)^{\mathrm{T}}V_q^{-1}(M_j - M_q)\right)\right]\right\}\\
&= \frac{1}{2}\sum_{j=1}^{J}\pi_j\,\frac{\partial}{\partial M_q}\left\{\left[\operatorname{tr}\left(U_q^{-1}(M_j - M_q)^{\mathrm{T}}V_q^{-1}(M_j - M_q)\right)\right]\right\}\\
&= \frac{1}{2}\sum_{j=1}^{J}\pi_j\,\frac{\partial}{\partial M_q}\left\{\left[\operatorname{tr}\left(U_q^{-1}M_j^{\mathrm{T}}V_q^{-1}M_j + U_q^{-1}M_q^{\mathrm{T}}V_q^{-1}M_q - 2U_q^{-1}M_j^{\mathrm{T}}V_q^{-1}M_q\right)\right]\right\}\\
&= \frac{1}{2}\sum_{j=1}^{J}\pi_j\,\frac{\partial}{\partial M_q}\left\{\left[\operatorname{tr}(U_q^{-1}M_j^{\mathrm{T}}V_q^{-1}M_j) + \operatorname{tr}(U_q^{-1}M_q^{\mathrm{T}}V_q^{-1}M_q) - 2\operatorname{tr}(U_q^{-1}M_j^{\mathrm{T}}V_q^{-1}M_q)\right]\right\}\\
&= \frac{1}{2}\sum_{j=1}^{J}\pi_j\,\frac{\partial}{\partial M_q}\left\{\left[\operatorname{tr}(U_q^{-1}M_q^{\mathrm{T}}V_q^{-1}M_q) - 2\operatorname{tr}(U_q^{-1}M_j^{\mathrm{T}}V_q^{-1}M_q)\right]\right\}\\
&= \frac{1}{2}\sum_{j=1}^{J}\pi_j\,\left[2U_q^{-1}M_q^{\mathrm{T}}V_q^{-1} - 2U_q^{-1}M_j^{\mathrm{T}}V_q^{-1}\right]\\
&= \sum_{j=1}^{J}\pi_j\,\left[U_q^{-1}(M_q - M_j)^{\mathrm{T}}V_q^{-1}\right]
\end{aligned} \tag{B.12}$$

Setting it equal to zero, and solving for M_q, we obtain

$$\begin{aligned}
\sum_{j=1}^{J} \pi_j \left[U_q^{-1}(M_q - M_j)^\intercal V_q^{-1} \right] &= 0 \\
U_q^{-1} \left(\sum_{j=1}^{J} \pi_j M_q \right) V_q^{-1} &= U_q^{-1} \left(\sum_{j=1}^{J} \pi_j M_j \right) V_q^{-1} \\
M_q \sum_{j=1}^{J} \pi_j &= \sum_{j=1}^{J} \pi_j M_j \\
M_q^\star &= \sum_{j=1}^{J} \pi_j M_j . \qquad \text{(B.13)}
\end{aligned}$$

Go forward for V_q, as before, starting by computing the partial derivative with respect to V_q indeed

$$\begin{aligned}
&\frac{\partial}{\partial V_q} \left\{ -\mathcal{H}(P) + \frac{nm}{2} \log 2\pi + \frac{m}{2} \log \mid V_q \mid + \frac{n}{2} \log \mid U_q \mid \right. \\
&\qquad \left. + \frac{1}{2} \sum_{j=1}^{J} \pi_j \left[\operatorname{tr}\left(V_q^{-1} V_j \right) \operatorname{tr}\left(U_q^{-1} U_j \right) + \operatorname{tr}\left(U_q^{-1}(M_j - M_q)^\intercal V_q^{-1}(M_j - M_q) \right) \right] \right\} \\
&\frac{\partial}{\partial V_q} \left\{ \frac{m}{2} \log \mid V_q \mid + \frac{1}{2} \sum_{j=1}^{J} \pi_j \left[\operatorname{tr}\left(V_q^{-1} V_j \right) \operatorname{tr}\left(U_q^{-1} U_j \right) + \operatorname{tr}\left(U_q^{-1}(M_j - M_q)^\intercal V_q^{-1}(M_j - M_q) \right) \right] \right\}. \qquad \text{(B.14)}
\end{aligned}$$

Hereafter, we consider as known the common column covariance matrix $\Sigma = U_q = U_j$ for $j = 1, \ldots, J$. This also implies that $\operatorname{tr}(U_q^{-1} U_j) = \operatorname{tr}(\Sigma^{-1}\Sigma) = \operatorname{tr}(\mathbb{I}_m) = m$.

$$\begin{aligned}
&\frac{\partial}{\partial V_q} \left\{ \frac{m}{2} \log \mid V_q \mid + \frac{1}{2} \sum_{j=1}^{J} \pi_j \left[\operatorname{tr}\left(V_q^{-1} V_j \right) \operatorname{tr}\left(U_q^{-1} U_j \right) + \operatorname{tr}\left(U_q^{-1}(M_j - M_q)^\intercal V_q^{-1}(M_j - M_q) \right) \right] \right\} \\
&\frac{\partial}{\partial V_q} \left\{ \frac{m}{2} \log \mid V_q \mid + \frac{m}{2} \sum_{j=1}^{J} \pi_j \left[\operatorname{tr}\left(V_q^{-1} V_j \right) + \operatorname{tr}\left(\Sigma^{-1}(M_j - M_q)^\intercal V_q^{-1}(M_j - M_q) \right) \right] \right\} \\
&\frac{m}{2} V_q^{-1} + \frac{m}{2} \sum_{j=1}^{J} \pi_j \left[-V_q^{-1} V_j V_q^{-1} - V_q^{-1}(M_j - M_q)^\intercal \Sigma^{-1}(M_j - M_q) V_q^{-1} \right] \\
&\frac{m}{2} \left\{ V_q^{-1} - V_q^{-1} \left[\sum_{j=1}^{J} \pi_j \left(V_j + (M_j - M_q)^\intercal \Sigma^{-1}(M_j - M_q) \right) \right] V_q^{-1} \right\}. \qquad \text{(B.15)}
\end{aligned}$$

Setting it equal to zero, and solving for M_q, we obtain

$$0 = \frac{m}{2} \left\{ V_q^{-1} - V_q^{-1} \left[\sum_{j=1}^{J} \pi_j \left(V_j + (M_j - M_q)^\intercal \Sigma^{-1}(M_j - M_q) \right) \right] V_q^{-1} \right\}$$

$$V_q^{-1} = V_q^{-1}\left[\sum_{j=1}^{J}\pi_j\left(V_j + (M_j - M_q)^{\intercal}\Sigma^{-1}(M_j - M_q)\right)\right]V_q^{-1}$$

$$V_q^{\star} = \sum_{j=1}^{J}\pi_j\left[V_j + (M_j - M_q)^{\intercal}\Sigma^{-1}(M_j - M_q)\right]. \tag{B.16}$$

Concluding, we obtain a closed-form solution for the optimization problem stated in Equation B.11. Hence, from results in Equations (B.13),(B.16), the matrix normal distribution which minimizes the KL divergence from a finite mixture of matrix normal distributions is the one, fully characterized, by the following optimal parameters:

$$M_q^{\star} = \sum_{j=1}^{J}\pi_j M_j\ , \quad V_q^{\star} = \sum_{j=1}^{J}\pi_j\left[V_j + (M_j - M_q)^{\intercal}\Sigma^{-1}(M_j - M_q)\right], \tag{B.17}$$

conditionally to a known common column covariance matrix $\Sigma = U_q = U_j$, for $j = 1, \ldots, J$.

B.1.3 Kullback-leibler divergence between a finite mixture of inverse Wishart and an inverse Wishart

In Section B.1.1, we consider the case P being a finite mixture of matrix-variate Gaussian distributions. Now, we are going to study the case when P is a finite mixture of inverse Wishart distributions instead. Then, for parallelism Q, a single inverse Wishart measure, such that

$$P(X) = \sum_{j=1}^{J}\pi_j\,\mathrm{IW}(X \mid \nu_j, \Psi_j)$$

$$Q(X) = \mathrm{IW}(X \mid \nu_q, \Psi_q). \tag{B.18}$$

Later on, we will consider U_j, U_q known for $j = 1, \ldots, J$. Then, we start to compute the $\mathrm{KL}(P \,||\, Q)$ by substituting the $p(x), q(x)$ in Equation (3.3) with the density function of P, Q.

$$\mathrm{KL}(P \,||\, Q) = \int\left(\sum_{j=1}^{J}\pi_j\,\mathrm{IW}(X \mid \nu_j, \Psi_j)\right)\log\left(\frac{\sum_{j=1}^{J}\pi_j\,\mathrm{IW}(X \mid \nu_j, \Psi_j)}{\mathrm{IW}(X \mid \nu_q, \Psi_q)}\right)dX. \tag{B.19}$$

Then, proceeding from Equation (B.19), we distribute the integral as

$$= \int\left(\sum_{j=1}^{J}\pi_j\,\mathrm{IW}(X \mid \nu_j, \Psi_j)\right)\log\left(\sum_{j=1}^{J}\pi_j\,\mathrm{IW}(X \mid \nu_j, \Psi_j)\right)dX -$$

$$\int\left(\sum_{j=1}^{J}\pi_j\,\mathrm{IW}(X \mid \nu_j, \Psi_j)\right)\log\left(\mathrm{IW}(X \mid \nu_q, \Psi_q)\right)dX$$

$$= \mathcal{I}_1 - \mathcal{I}_2. \tag{B.20}$$

Thus, we now consider separately these two integrals, renamed $\mathcal{I}_1, \mathcal{I}_2$ respectively. Let us start from $\mathcal{I}_1$:

$$\begin{aligned}\mathcal{I}_1 &= \int \left(\sum_{j=1}^{J} \pi_j \,\mathrm{IW}(X \mid \nu_j, \Psi_j)\right) \log\left(\sum_{j=1}^{J} \pi_j \,\mathrm{IW}(X \mid \nu_j, \Psi_j)\right) dX \\ &= \mathbb{E}_X\left[\log P(X)\right] \qquad \text{with } X \sim \sum_{j=1}^{J} \pi_j \,\mathrm{IW}(X \mid \nu_j, \Psi_j) \\ &= -\mathcal{H}(P), \end{aligned} \tag{B.21}$$

where $\mathcal{H}(P)$ represent the entropy associated to measure P. Then, we proceed to compute the second integral:

$$\begin{aligned}\mathcal{I}_2 &= \int \left(\sum_{j=1}^{J} \pi_j \,\mathrm{IW}(X \mid \nu_j, \Psi_j)\right) \log\left(\mathrm{IW}(X \mid \nu_q, \Psi_q)\right) dX \\ &= \sum_{j=1}^{J} \pi_j \Bigg\{ \int 2^{-\frac{\nu_j m}{2}} \mid \Psi_j \mid^{\frac{\nu_q}{2}} \Gamma_m^{-1}(\frac{\nu_j}{2}) \mid X \mid^{-\frac{\nu_j+m+1}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}(\Psi_j X^{-1})\right\} \\ &\qquad \log\left(2^{-\frac{\nu_q m}{2}} \mid \Psi_q \mid^{\frac{m}{2}} \Gamma_m^{-1}(\frac{\nu_q}{2}) \mid X \mid^{-\frac{\nu_q+m+1}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}(\Psi_q X^{-1})\right\}\right) dX \Bigg\} \\ &= \sum_{j=1}^{J} \pi_j \Bigg\{ \int 2^{-\frac{\nu_j m}{2}} \mid \Psi_j \mid^{\frac{\nu_q}{2}} \Gamma_m^{-1}(\frac{\nu_j}{2}) \mid X \mid^{-\frac{\nu_j+m+1}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}(\Psi_j X^{-1})\right\} \\ &\qquad -\frac{\nu_q m}{2}\log 2 + \frac{\nu_q}{2}\log \mid \Psi_q \mid - \log\Gamma_m(\frac{\nu_q}{2}) - \frac{\nu_q+m+1}{2}\log \mid X \mid -\frac{1}{2}\mathrm{tr}(\Psi_q X^{-1}) dX \Bigg\} \\ &= \sum_{j=1}^{J} \pi_j \Bigg\{ -\frac{\nu_q m}{2}\log 2 + \frac{\nu_q}{2}\log \mid \Psi_q \mid - \log\Gamma_m(\frac{\nu_q}{2}) + \\ &\int 2^{-\frac{\nu_j m}{2}} \mid \Psi_j \mid^{\frac{\nu_q}{2}} \Gamma_m^{-1}(\frac{\nu_j}{2}) \mid X \mid^{-\frac{\nu_j+m+1}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}(\Psi_j X^{-1})\right\} \left[-\frac{\nu_q+m+1}{2}\log \mid X \mid -\frac{1}{2}\mathrm{tr}(\Psi_q X^{-1})\right] dX \Bigg\} \\ &= \sum_{j=1}^{J} \pi_j \Bigg\{ -\frac{\nu_q m}{2}\log 2 + \frac{\nu_q}{2}\log \mid \Psi_q \mid - \log\Gamma_m(\frac{\nu_q}{2}) - \frac{\nu_q+m+1}{2}\mathbb{E}_X\left[\log \mid X \mid\right] - \frac{1}{2}\mathbb{E}_X\left[\mathrm{tr}(\Psi_q X^{-1})\right] \Bigg\} \\ &= -\frac{\nu_q m}{2}\log 2 + \frac{\nu_q}{2}\log \mid \Psi_q \mid - \log\Gamma_m(\frac{\nu_q}{2}) - \frac{\nu_q+m+1}{2}\sum_{j=1}^{J}\pi_j\mathbb{E}_X\left[\log \mid X \mid\right] - \frac{1}{2}\sum_{j=1}^{J}\pi_j\mathbb{E}_X\left[\mathrm{tr}(\Psi_q X^{-1})\right] \\ &= -\frac{\nu_q m}{2}\log 2 + \frac{\nu_q}{2}\log \mid \Psi_q \mid - \log\Gamma_m(\frac{\nu_q}{2}) - \frac{\nu_q+m+1}{2}\sum_{j=1}^{J}\pi_j\mathbb{E}_X\left[\log \mid X \mid\right] - \frac{1}{2}\sum_{j=1}^{J}\pi_j \,\mathrm{tr}\left(\Psi_q\,\mathbb{E}_X\left[X^{-1}\right]\right), \end{aligned} \tag{B.22}$$

with $X \sim \mathrm{IW}(X \mid \nu_j, \Psi_j)$ for $j = 1, \ldots, J$. In order to proceed, we have to find the value of the two expectations in Equation (B.22). Since the existing relationship between the inverse Wishart and the Wishart distributions: $X \sim \mathrm{IW}(\nu_j, \Psi_j) \iff X^{-1} \sim \mathrm{W}(\nu_j, \Psi_j^{-1})$, we can derive

the following expectation straightforwardly

$$\mathbb{E}_X\left[X^{-1}\right] = \mathbb{E}_{X^{-1}}\left[X\right] = \nu_j \Psi_j^{-1}. \tag{B.23}$$

By similar reasoning, we can derive the log expectation in Equation (B.22)

$$\mathbb{E}_X\left[\log \mid X \mid\right] = \mathbb{E}_{X^{-1}}\left[\log \mid X^{-1} \mid\right] = m\log 2 + \log \mid \Psi_j \mid + \sum_{i=1}^{m} \psi\left(\frac{\nu_j - i + 1}{2}\right). \tag{B.24}$$

Alternatively, as in Granström and Orguner (2012), to recover the second expected value, we have to recall that the logarithm of the multivariate Gamma function factorizes as $\log \Gamma_m(a) = \frac{m(m-1)}{4}\log\pi + \sum_{i=1}^{m}\log\Gamma\left(a - \frac{(i-1)}{2}\right)$. Hence, consider now $Y = \log \mid X \mid$ with the related moment generating function (MGF) defined as $\mu_Y(s) = \mathbb{E}_Y\left[e^{sY}\right]$

$$\begin{aligned}
\mu_Y(s) &= \mathbb{E}\left[e^{sY}\right] \\
&= \mathbb{E}\left[e^{s\log|X|}\right] \\
&= \mathbb{E}[\mid X \mid^s] \\
&= \int \mid X \mid^s P(X) dX \\
&= \int \mid X \mid^s \mid X \mid^{-\frac{(\nu_j+m+1)}{2}} \frac{2^{-\frac{\nu_j m}{2}} \mid \Psi_j \mid^{\frac{\nu_j}{2}}}{\Gamma_m\left(\frac{\nu_j}{2}\right)} \exp\left\{-\frac{1}{2}\operatorname{tr}(\Psi_j X^{-1})\right\} dX \\
&= \int \mid X \mid^{-\frac{(\nu_j-2s+m+1)}{2}} \frac{2^{-\frac{\nu_j m}{2}} \mid \Psi_j \mid^{\frac{\nu_j}{2}}}{\Gamma_m\left(\frac{\nu_j}{2}\right)} \exp\left\{-\frac{1}{2}\operatorname{tr}(\Psi_j X^{-1})\right\} dX \\
&= \int \mathrm{IW}(X \mid \nu_j - 2s, \Psi_j)\, dX\, \frac{\Gamma_m\left(\frac{\nu_j - s}{2}\right)}{\Gamma_m\left(\frac{\nu_j}{2}\right)} \left(\frac{\mid \Psi_j \mid}{2^{-m}}\right)^s \\
&= \frac{\Gamma_m\left(\frac{\nu_j - s}{2}\right)}{\Gamma_m\left(\frac{\nu_j}{2}\right)} \left(\frac{\mid \Psi_j \mid}{2^{-m}}\right)^s.
\end{aligned} \tag{B.25}$$

Once obtained an analytical form for the MGF, we must take the first derivative and evaluate it at $s = 0$ to achieve the first moment of $Y = \log \mid X \mid$.

$$\begin{aligned}
&\frac{d}{ds}\left\{\mu_Y(s)\right\}\Big|_{s=0} \\
&= \frac{d}{ds}\left\{\frac{\Gamma_m\left(\frac{\nu_j}{2} - s\right)}{\Gamma_m\left(\frac{\nu_j}{2}\right)} \left(\frac{\mid \Psi_j \mid}{2^{-m}}\right)^s\right\}\Bigg|_{s=0} \\
&= \left\{\frac{d}{ds}\frac{\Gamma_m\left(\frac{\nu_j}{2} - s\right)}{\Gamma_m\left(\frac{\nu_j}{2}\right)} \left(\frac{\mid \Psi_j \mid}{2^{-m}}\right)^s + \frac{\Gamma_m\left(\frac{\nu_j}{2} - s\right)}{\Gamma_m\left(\frac{\nu_j}{2}\right)} \frac{d}{ds}\left(\frac{\mid \Psi_j \mid}{2^{-m}}\right)^s\right\}\Bigg|_{s=0}
\end{aligned}$$

$$
\begin{aligned}
&= \left\{ \frac{d}{ds} \log \Gamma_m \left(\frac{\nu_j}{2} - s \right) \left(\frac{|\, \Psi_j \,|}{2^{-m}} \right)^s + \frac{\Gamma_m \left(\frac{\nu_j}{2} - s \right)}{\Gamma_m \left(\frac{\nu_j}{2} \right)} \left(\frac{|\, \Psi_j \,|}{2^{-m}} \right)^s \log \left(\frac{|\, \Psi_j \,|}{2^{-m}} \right) \right\} \Bigg|_{s=0} \\
&= \left\{ \frac{d}{ds} \left[\frac{m(m-1)}{4} \log \pi + \sum_{i=1}^{m} \log \Gamma \left(\frac{\nu_j - i + 1}{2} - s \right) \right] \left(\frac{|\, \Psi_j \,|}{2^{-m}} \right)^s + \frac{\Gamma_m \left(\frac{\nu_j}{2} - s \right)}{\Gamma_m \left(\frac{\nu_j}{2} \right)} \left(\frac{|\, \Psi_j \,|}{2^{-m}} \right)^s \log \left(\frac{|\, \Psi_j \,|}{2^{-m}} \right) \right\} \Bigg|_{s=0} \\
&= \left\{ \sum_{i=1}^{m} \psi \left(\frac{\nu_j - i + 1}{2} - s \right) \left(\frac{|\, \Psi_j \,|}{2^{-m}} \right)^s + \frac{\Gamma_m \left(\frac{\nu_j}{2} - s \right)}{\Gamma_m \left(\frac{\nu_j}{2} \right)} \left(\frac{|\, \Psi_j \,|}{2^{-m}} \right)^s \log \left(\frac{|\, \Psi_j \,|}{2^{-m}} \right) \right\} \Bigg|_{s=0} \\
&= \sum_{i=1}^{m} \psi \left(\frac{\nu_j - i + 1}{2} \right) + \log \left(\frac{|\, \Psi_j \,|}{2^{-m}} \right) \\
&= \sum_{i=1}^{m} \psi \left(\frac{\nu_j - i + 1}{2} \right) + \log |\, \Psi_j \,| + m \log 2
\end{aligned} \tag{B.26}
$$

Once substituted, the analytical form of the two expectations, $\mathcal{I}_2$ evolves as

$$
\begin{aligned}
&- \frac{\nu_q m}{2} \log 2 + \frac{\nu_q}{2} \log |\, \Psi_q \,| - \log \Gamma_m (\frac{\nu_q}{2}) - \frac{\nu_q + m + 1}{2} \sum_{j=1}^{J} \pi_j \mathbb{E}_X \left[\log |\, X \,| \right] - \frac{1}{2} \sum_{j=1}^{J} \pi_j \operatorname{tr} \left(\Psi_q \, \mathbb{E}_X \left[X^{-1} \right] \right) \\
&= -\frac{\nu_q m}{2} \log 2 + \frac{\nu_q}{2} \log |\, \Psi_q \,| - \log \Gamma_m (\frac{\nu_q}{2}) \\
&\qquad - \frac{\nu_q + m + 1}{2} \sum_{j=1}^{J} \pi_j \left[\sum_{i=1}^{m} \psi \left(\frac{\nu_j - i + 1}{2} \right) + \log |\, \Psi_j \,| + m \log 2 \right] - \frac{1}{2} \sum_{j=1}^{J} \pi_j \operatorname{tr} \left(\nu_j \Psi_j^{-1} \Psi_q \right) \\
&= -\frac{\nu_q m}{2} \log 2 + \frac{\nu_q}{2} \log |\, \Psi_q \,| - \log \Gamma_m (\frac{\nu_q}{2}) - \frac{\nu_q + m + 1}{2} \sum_{j=1}^{J} \pi_j \sum_{i=1}^{m} \psi \left(\frac{\nu_j - i + 1}{2} \right) \\
&\qquad - \frac{\nu_q + m + 1}{2} \sum_{j=1}^{J} \pi_j \log |\, \Psi_j \,| - \frac{\nu_q + m + 1}{2} m \log 2 - \frac{1}{2} \sum_{j=1}^{J} \pi_j \operatorname{tr} \left(\nu_j \Psi_j^{-1} \Psi_q \right)
\end{aligned} \tag{B.27}
$$

Finally, coming back to Equation (B.19), it boils down to

$$
\begin{aligned}
\textsc{kl}(P \,||\, Q) &= \int \left(\sum_{j=1}^{J} \pi_j \operatorname{IW}(X \mid \nu_j, \Psi_j) \right) \log \left(\frac{\sum_{j=1}^{J} \pi_j \operatorname{IW}(X \mid \nu_j, \Psi_j)}{\operatorname{IW}(X \mid \nu_q, \Psi_q)} \right) dX \\
&= -\mathcal{H}(P) + \frac{\nu_q m}{2} \log 2 - \frac{\nu_q}{2} \log |\, \Psi_q \,| + \log \Gamma_m (\frac{\nu_q}{2}) + \frac{\nu_q + m + 1}{2} \sum_{j=1}^{J} \pi_j \sum_{i=1}^{m} \psi \left(\frac{\nu_j - i + 1}{2} \right) \\
&\qquad + \frac{\nu_q + m + 1}{2} \sum_{j=1}^{J} \pi_j \log |\, \Psi_j \,| + \frac{\nu_q + m + 1}{2} m \log 2 + \frac{1}{2} \sum_{j=1}^{J} \pi_j \operatorname{tr} \left(\nu_j \Psi_j^{-1} \Psi_q \right).
\end{aligned} \tag{B.28}
$$

B.1.4 Partially closed-form optimal minimizer parameters for KL divergence between a finite mixture of inverse Wishart and an inverse Wishart

As made for matrix variate Gaussian distribution in Equation (B.11), let us now consider minimizing the KL divergence in Equation (B.28) with regard to the parameter of Q, i.e. ν_q, Ψ_q, looking to characterize the inverse Wishart distribution which minimizes Kullback-Leibler divergence with P. We are now looking for

$$\arg\min_{Q\in\mathbb{Q}} \text{KL}(P \mid\mid Q) = \arg\min_{\nu_q\in\mathbb{R}^+,\Psi_q\succ 0\in\mathbb{R}^{m\times m}} \text{KL}(\sum_{j=1}^{J}\pi_j \text{IW}(X \mid \nu_j,\Psi_j) \mid\mid \text{IW}(X \mid \nu_q,\Psi_q)). \tag{B.29}$$

Starting from Ψ_q

$$\begin{aligned}
&\frac{\partial}{\partial\Psi_q}\{\text{KL}(P \mid\mid Q)\} \\
&= \frac{\partial}{\partial\Psi_q}\Bigg\{ -\mathcal{H}(P) + \frac{\nu_q m}{2}\log 2 - \frac{\nu_q}{2}\log \mid \Psi_q \mid + \log\Gamma_m(\frac{\nu_q}{2}) + \frac{\nu_q+m+1}{2}\sum_{j=1}^{J}\pi_j\sum_{i=1}^{m}\psi\left(\frac{\nu_j-i+1}{2}\right) \\
&\qquad + \frac{\nu_q+m+1}{2}\sum_{j=1}^{J}\pi_j\log\mid\Psi_j\mid + \frac{\nu_q+m+1}{2}m\log 2 + \frac{1}{2}\sum_{j=1}^{J}\pi_j\,\text{tr}\left(\nu_j\Psi_j^{-1}\Psi_q\right)\Bigg\} \\
&= -\frac{\nu_q}{2}\Psi_q^{-1} + \frac{1}{2}\sum_{j=1}^{J}\pi_j\nu_j\Psi_j^{-1}
\end{aligned} \tag{B.30}$$

Setting the derivative equal to zero, we obtain a closed-form optimal parameter formulation

$$\begin{aligned}
-\frac{\nu_q}{2}\Psi_q^{-1} + \frac{1}{2}\sum_{j=1}^{J}\pi_j\nu_j\Psi_j^{-1} &= 0 \\
\frac{\nu_q}{2}\Psi_q^{-1} &= \frac{1}{2}\sum_{j=1}^{J}\pi_j\nu_j\Psi_j^{-1} \\
\Psi_q^{-1} &= \nu_q^{-1}\sum_{j=1}^{J}\pi_j\nu_j\Psi_j^{-1} \\
\Psi_q^{\star} &= \left[\sum_{j=1}^{J}\pi_j\frac{\nu_j}{\nu_q}\Psi_j^{-1}\right]^{-1}
\end{aligned} \tag{B.31}$$

When passing to the parameter associated with the degrees of freedom, i.e., ν_q, no analytical closed-form solutions are available. Mainly, this happens as the parameter ν_q appears as the argument of multiple digamma functions, making a direct derivation of the optimal parameter unavailable.

However, it is instead possible to achieve a numerical solution by implementing a numerical optimizer or a root finder for the derivative, which exists and can be analytically computed In particular, one can directly minimize Equation (B.28) with regard to ν_q, once Ψ_q is substituted

with its the analytic solution found in Equation (B.31), or find the zero of the following derivative function:

$$
\begin{aligned}
&\frac{\partial}{\partial \nu_q}\{\text{KL}(P \mid\mid Q)\} \\
&= \frac{\partial}{\partial \nu_q}\Bigg\{ -\mathcal{H}(P) + \frac{\nu_q m}{2}\log 2 - \frac{\nu_q}{2}\log \mid \Psi_q \mid + \log\Gamma_m(\frac{\nu_q}{2}) + \frac{\nu_q + m + 1}{2}\sum_{j=1}^{J}\pi_j \sum_{i=1}^{m}\psi\left(\frac{\nu_j - i + 1}{2}\right) \\
&\qquad + \frac{\nu_q + m + 1}{2}\sum_{j=1}^{J}\pi_j \log \mid \Psi_j \mid + \frac{\nu_q + m + 1}{2} m \log 2 + \frac{1}{2}\sum_{j=1}^{J}\pi_j \operatorname{tr}\left(\nu_j \Psi_j^{-1}\Psi_q\right)\Bigg\} \\
&= \frac{m}{2}\log 2 - \frac{1}{2}\log \mid \Psi_q \mid + \frac{\partial}{\partial \nu_q}\log\Gamma_m(\frac{\nu_q}{2}) + \frac{1}{2}\sum_{j=1}^{J}\pi_j \sum_{i=1}^{m}\psi\left(\frac{\nu_j - i + 1}{2}\right) - \frac{1}{2}\log \mid \Psi_j \mid + \frac{m}{2}\log 2 \\
&= m\log 2 - \frac{1}{2}\log\left|\nu_q\left[\sum_{j=1}^{J}\pi_j \nu_j \Psi_j^{-1}\right]^{-1}\right| + \frac{\partial}{\partial \nu_q}\log\Gamma_m(\frac{\nu_q}{2}) + \frac{1}{2}\sum_{j=1}^{J}\pi_j \sum_{i=1}^{m}\psi\left(\frac{\nu_j - i + 1}{2}\right) - \frac{1}{2}\log \mid \Psi_j \mid \\
&= m\log 2 - \frac{1}{2}\log \nu_q - \frac{1}{2}\log\left|\left[\sum_{j=1}^{J}\pi_j \nu_j \Psi_j^{-1}\right]^{-1}\right| + \frac{\partial}{\partial \nu_q}\left[m(m-1)\log \pi^{\frac{1}{4}} + \sum_{i=1}^{m}\log\Gamma\left(\frac{\nu_q - i + 1}{2}\right)\right] \\
&\qquad + \frac{1}{2}\sum_{j=1}^{J}\pi_j \sum_{i=1}^{m}\psi\left(\frac{\nu_j - i + 1}{2}\right) - \frac{1}{2}\log \mid \Psi_j \mid \\
&= -\frac{1}{2}\log \nu_q + \sum_{i=1}^{m}\psi\left(\frac{\nu_q - i + 1}{2}\right) \\
&\qquad + m\log 2 - \frac{1}{2}\log\left|\left[\sum_{j=1}^{J}\pi_j \nu_j \Psi_j^{-1}\right]^{-1}\right| + \frac{1}{2}\sum_{j=1}^{J}\pi_j \sum_{i=1}^{m}\psi\left(\frac{\nu_j - i + 1}{2}\right) - \frac{1}{2}\log \mid \Psi_j \mid \\
&= -\frac{1}{2}\log \nu_q + \sum_{i=1}^{m}\psi\left(\frac{\nu_q - i + 1}{2}\right) + C(m, \{\nu_j, \Psi_j\}_{j=1,\dots,J}),
\end{aligned}
\tag{B.32}
$$

where $C(m, \{\nu_j, \Psi_j\}_{j=1,\dots,J})$ is a constant term. Worth noticing that $C(m, \{\nu_j, \Psi_j\}_{j=1,\dots,J})$ only depends on known quantities. Indeed, its argument are m, and the set of different competitive models inverse Wishart parameters $\{\nu_j, \Psi_j\}_{j=1,\dots,J}$. Therefore, we could formally obtain the optimal degrees of freedom parameter $\nu_q^\star$ as the solution to the following equation:

$$
\begin{aligned}
0 &= -\frac{1}{2}\log \nu_q + \sum_{i=1}^{m}\psi\left(\frac{\nu_q - i + 1}{2}\right) + C(m, \{\nu_j, \Psi_j\}_{j=1,\dots,J}) \\
\frac{1}{2}\log \nu_q &= \sum_{i=1}^{m}\psi\left(\frac{\nu_q - i + 1}{2}\right) + C(m, \{\nu_j, \Psi_j\}_{j=1,\dots,J}) \\
\nu_q &= \exp\left\{2\sum_{i=1}^{m}\psi\left(\frac{\nu_q - i + 1}{2}\right) + 2C(m, \{\nu_j, \Psi_j\}_{j=1,\dots,J})\right\}.
\end{aligned}
\tag{B.33}
$$

B.2 Algorithms

Algorithm 8 BPS - forward filter

Input: $\hat{w}_t = \{\hat{w}_{t,j} : j \in \{1, \ldots, J\}\}$: Stacking weights within time shard; n, m, q, p: Number of observed rows, number of outcomes, and number of predictors; J: number of competitive models in each subset, and number of predictive samples; $\{\{m_{t-1}^{(j)}, C_{t-1}^{(j)}, \Psi_{t-1}^{(j)}, \nu_{t-1}^{(j)}, \alpha^{(j)}, \phi^{(j)}\} : j = 1, \ldots, J\}$: Set of parameters for antecedent time point, for J models.

Output: $\{\{m_t^{(j)}, C_t^{(j)}, \Psi_t^{(j)}, \nu_t^{(j)}\} : j = 1, \ldots, J\}$: Set of parameters for current time point; $\hat{w}_t$: Stacking weights at current time.

for $t = 1, \ldots, T$ **do**
  Extract data from previous time shard $\{\{m_{t-1}^{(j)}, C_{t-1}^{(j)}, \Psi_{t-1}^{(j)}, \nu_{t-1}^{(j)}\} : j = 1, \ldots, J\}$
  Obtain $\{\tilde{m}_t, \tilde{C}_t, \tilde{\nu}_t, \tilde{\Psi}_t\}$ using Algorithm 5
  for $j = 1, \ldots, J$ **do**
    Compute $a_t^{(j)}, R_t^{(j)}$ using $\tilde{m}_{t-1}, \tilde{C}_{t-1}$, and $\{\alpha^{(j)}, \phi^{(j)}\}$
    Compute $q_t^{(j)}, Q_t^{(j)}$ using $a_t^{(j)}, R_t^{(j)}$, and $\{\alpha^{(j)}, \phi^{(j)}\}$
    Compute $m_t^{(j)}, C_t^{(j)}$ using $a_t^{(j)}, R_t^{(j)}$, and $\{\alpha^{(j)}, \phi^{(j)}\}$
    Compute $\nu_t^{(j)}, \Psi_t^{(j)}$ using $q_t^{(j)}, Q_t^{(j)}, \tilde{\nu}_{t-1}, \tilde{\Psi}_{t-1}$, and $\{\alpha^{(j)}, \phi^{(j)}\}$
  end for
  Obtain $\hat{w}_t = \{\hat{w}_{t,1}, \ldots, \hat{w}_{t,J}\}$ using Algorithm 6
end for
return $\{\{m_t^{(j)}, C_t^{(j)}, \Psi_t^{(j)}, \nu_t^{(j)}\} : j = 1, \ldots, J\}$, and $\hat{w}_t$

Algorithm 9 BPS - parallel forward filter

Input: $\hat{w}_t = \{\hat{w}_{t,j} : j \in \{1, \ldots, J\}\}$: Stacking weights within time shard; n, m, q, p: Number of observed rows, number of outcomes, and number of predictors; J: number of competitive models in each subset, and number of predictive samples; $\{\{m_{t-1}^{(j)}, C_{t-1}^{(j)}, \Psi_{t-1}^{(j)}, \nu_{t-1}^{(j)}, \alpha^{(j)}, \phi^{(j)}\} : j = 1, \ldots, J\}$: Set of parameters for antecedent time point, for J models.

Output: $\{\{m_t^{(j)}, C_t^{(j)}, \Psi_t^{(j)}, \nu_t^{(j)}\} : j = 1, \ldots, J\}$: Set of parameters for current time point; $\hat{w}^{(t)}$: Stacking weights at the current time.

for $t = 1, \ldots, T$ **do**
  Extract data from previous time shard $\{\{m_{t-1}^{(j)}, C_{t-1}^{(j)}, \Psi_{t-1}^{(j)}, \nu_{t-1}^{(j)}\} : j = 1, \ldots, J\}$
  for $j = 1, \ldots, J$ **do Parallel**
    Compute $a_t^{(j)}, R_t^{(j)}$ using $m_{t-1}^{(j)}, C_{t-1}^{(j)}$, and $\{\alpha^{(j)}, \phi^{(j)}\}$
    Compute $q_t^{(j)}, Q_t^{(j)}$ using $a_t^{(j)}, R_t^{(j)}$, and $\{\alpha^{(j)}, \phi^{(j)}\}$
    Compute $m_t^{(j)}, C_t^{(j)}$ using $a_t^{(j)}, R_t^{(j)}$, and $\{\alpha^{(j)}, \phi^{(j)}\}$
    Compute $\nu_t^{(j)}, \Psi_t^{(j)}$ using $q_t^{(j)}, Q_t^{(j)}, \nu_{t-1}^{(j)}, \Psi_{t-1}^{(j)}$, and $\{\alpha^{(j)}, \phi^{(j)}\}$
  end Parallel for
  Obtain $\hat{w}^{(t)} = \{\hat{w}_1^{(t)}, \ldots, \hat{w}_J^{(t)}\}$ using Algorithm 6
end for
return $\{\{m_t^{(j)}, C_t^{(j)}, \Psi_t^{(j)}, \nu_t^{(j)}\} : j = 1, \ldots, J\}$, and $\hat{w}^{(t)}$

Algorithm 10 BPS - weighted backward sampling

Input: $\hat{w}_t = \{\hat{w}_{t,j}\} : j \in \{1, \ldots, J\}\}$: Stacking weights within time shard; n, q, p: Number of observed rows, number of prediction points, number of outcomes, and number of predictors; J, R: number of competitive models in each subset, and number of posterior samples; $\{\{m_t^{(j)}, C_t^{(j)}, \Psi_t^{(j)}, \nu_t^{(j)}, \alpha^{(j)}, \phi^{(j)}\} : t = 1, \ldots, T; j = 1, \ldots, J\}$: Set of parameters for each time point, and model.
Output: $\{\Theta_{0:T}^{(r)} : r = 1, \ldots, R\}$: Smoothed posterior samples.

for $r = 1, \ldots, R$ **do**
  Sample the model $\mathscr{M}_j$ from $\{1, \ldots, J\}$ with weights $\hat{w}_T$
  Draw Θ_T from $\mathrm{T}_{2\nu_T^{(j)}}(m_T^{(j)}, C_T^{(j)}, \Psi_T^{(j)}, \nu_T^{(j)})$
  for $t = T - 1, \ldots, 1$ **do**
    Sample the model $\mathscr{M}_j$ from $\{1, \ldots, J\}$ with weights $\hat{w}_t$
    Compute $\{h_t, H_t\}$ using $\{m_t^{(j)}, C_t^{(j)}, \alpha^{(j)}, \phi^{(j)}\}$
    Draw Θ_t from $\mathrm{T}_{2\nu_T^{(j)}}(m_T^{(j)}, C_T^{(j)}, \Psi_T^{(j)}, \nu_T^{(j)})$
  end for
end for
return $\{\Theta_{0:T}^{(r)} \sim \hat{p}(\Theta_0, \ldots, \Theta_T \mid \mathscr{D}_t) : r = 1, \ldots, R\}$
with $\hat{p}(\Theta_0, \ldots, \Theta_T \mid \mathscr{D}_T) = \hat{p}(\Theta_0 \mid \Theta_1, \mid \mathscr{D}_T)\hat{p}(\Theta_1 \mid \Theta_2, \mid \mathscr{D}_T) \cdots \hat{p}(\Theta_{T-1} \mid \Theta_T, \mid \mathscr{D}_T)\hat{p}(\Theta_T \mid \mathscr{D}_T)$
and each $\hat{p}(\Theta_t \mid \Theta_{t+1}, \mid \mathscr{D}_T) = \sum_{j=1}^{J} \hat{w}_{t,j}\, \hat{p}(\Theta_t \mid \Theta_{t+1}, \mid \mathscr{D}_T, \mathscr{M}_j)$

Algorithm 11 BPS - temporal forecasting

Input: $\hat{w}_T = \{\hat{w}_{T,j}\} : j \in \{1, \ldots, J\}\}$: Stacking weights within last observed time shard T; n, q, p: Number of observed rows, number of outcomes, and number of predictors; J, R, K: Number of competitive models in each subset, number of predictive samples, and prediction horizon; $\{\{m_t^{(j)}, C_t^{(j)}, \Psi_t^{(j)}, \nu_t^{(j)}, \alpha^{(j)}, \phi^{(j)}\} : t = 1, \ldots, T; j = 1, \ldots, J\}$: Set of parameters for each time point, and model.
Output: $\{\{\tilde{Y}_{T+k}^{(r)}, \tilde{\Omega}_{T+k}^{(r)}\} : k = 1, \ldots, K; r = 1, \ldots, R\}$: Posterior predictive samples.

for $r = 1, \ldots, R$ **do**
  Sample the model $\mathscr{M}_j$ from $\{1, \ldots, J\}$ with weights $\hat{w}_T$
  Compute $\{A^\star(0), R^\star(0)\}$ using $\{m_T^{(j)}, C_T^{(j)}\}$
  for $k = 1, \ldots, K$ **do**
    Compute $\{A^\star(k), R^\star(k)\}$ using $\{A^\star(k-1), R^\star(k-1)\}$
    Draw $\{\tilde{\Theta}_{T+k}^{(r)}, \tilde{Y}_{T+k}^{(r)}\}$ from $\mathrm{T}_{2\nu_T^{(j)}}(A^\star(k), R^\star(k), \Psi_T^{(j)}, \nu_T^{(j)})$
    Store $\{A^\star(k), R^\star(k)\}$ for the next iteration
  end for
end for
return $\{\{\tilde{\Theta}_{T+k}^{(r)}, \tilde{Y}_{T+k}^{(r)}\} \sim \hat{p}(\tilde{\Theta}_{T+k}, \tilde{Y}_{T+k} \mid \mathscr{D}_T) : k = 1, \ldots, K; r = 1, \ldots, R\}$
with $\hat{p}(\tilde{\Theta}_{T+k}, \tilde{Y}_{T+k} \mid \mathscr{D}_T) = \sum_{j=1}^{J} \hat{w}_{t,j}\, p(\tilde{\Theta}_{T+k}, \tilde{Y}_{T+k} \mid \mathscr{D}_T, \mathscr{M}_j)$

Algorithm 12 BPS - spatial prediction

Input: $\hat{w}_t = \{\hat{w}_{t,j}\} : j \in \{1, \ldots, J\}\}$: Stacking weights within time shard; n, m, q, p: Number of observed rows, number of prediction points, number of outcomes, and number of predictors; J, R: Number of competitive models in each subset, and number of predictive samples; $\{\{m_t^{(j)}, C_t^{(j)}, \Psi_t^{(j)}, \nu_t^{(j)}, \alpha^{(j)}, \phi^{(j)}\} : t = 1, \ldots, T; j = 1, \ldots, J\}$: Set of parameters for each time point, and model.

Output: $\{\{\tilde{Y}_t^{(r)}, \tilde{\Omega}_t^{(r)}\} : r = 1, \ldots, R\}$: Posterior predictive samples.

1: **for** $r = 1, \ldots, R$ **do**
2: Sample the model $\mathscr{M}_j$ from $\{1, \ldots, J\}$ with weights $\hat{w}_t$
3: Compute μ_t, E_t using $\{m_t^{(j)}, C_t^{(j)}, \alpha^{(j)}, \phi^{(j)}\}$
4: Draw $\{\tilde{Y}_t^{(r)}, \tilde{\Omega}_t^{(r)}\}$ from $\mathrm{T}_{2\nu_t^{(j)}}(\mu_t, E_t, \Psi_t^{(j)}, \nu_t^{(j)})$
5: **end for**
6: **return** $\{\{\tilde{Y}_t^{(r)}, \tilde{\Omega}_t^{(r)}\} \sim \hat{p}(\tilde{Y}_t, \tilde{\Omega}_t \mid \mathscr{D}_t) : r = 1, \ldots, R\}$

with $\hat{p}(\tilde{Y}_t, \tilde{\Omega}_t \mid \mathscr{D}_t) = \sum_{j=1}^{J} \hat{w}_{t,j}\, p(\tilde{Y}_t, \tilde{\Omega}_t \mid \mathscr{D}_t, \mathscr{M}_j)$