Statistical Models for Dependent Data

by

Simon Fontaine

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in the University of Michigan
2024

Doctoral Committee:

Professor Jian Kang, Co-Chair
Professor Ji Zhu, Co-Chair
Professor Elizaveta Levina
Associate Professor Gen Li

Simon Fontaine
simfont@umich.edu
ORCID iD: 0000-0003-1835-1231

ACKNOWLEDGEMENTS

I wish to express my profound gratitude to my advisor, Ji Zhu, for his exceptional mentorship throughout this academic journey and beyond. I am wholeheartedly grateful for the confidence he has invested in me, granting me the liberty to delve into diverse subjects and flourish as a researcher. Additionally, I am immensely appreciative of the myriad opportunities that have unfolded before me as a direct result of his guidance. First, I want to thank Liza Levina along with everyone in the Levina-Zhu research group for the invaluable comments and interesting discussions throughout the years. Second, I want to thank Joel H. Rubenstein and Akbar Waljee for the incredible opportunity of working on the HOSEA project (Rubenstein et al., 2023), which taught me a lot about big data, risk modeling and applied research. Third, I want to thank Gen Li for making me discover the fascinating area of microbiome data analysis, which I will surely continue studying in the near future. I am also thankful towards all members of the MDAWG research group for the engaging discussions. Fourth, I would like to thank Jian Kang for introducing me to the captivating topic of brain-computer interfaces and for his deep insights into Bayesian modeling.

I wish to extend my heartfelt appreciation to the entire Department of Statistics for fostering a welcoming environment and for offering an abundance of assistance, support, and opportunities to all students, with a special emphasis on the needs of international students. The prospect of a five-year academic venture far from home brought with it inherent challenges and uncertainties, yet my experience at Michigan has been seamless and rewarding, thanks in large part to this community. My gratitude goes out to the staff members whose tireless efforts have been instrumental in ensuring our well-being throughout our time here. In particular, I must express my sincere thanks to Judy McDonald for her warm and efficient support, especially during the organization of the Michigan Student Symposium for Interdisciplinary Statistical Sciences. I am also indebted to all the faculty members of the department, and especially to my professors, whose excellent teaching has been pivotal in preparing us to pursue careers as independent researchers. Furthermore, I owe a debt of gratitude to my fellow students; their efforts have created a nurturing space that has been vital for both personal and academic growth. I wish to single out the altruistic contributions of the preceding cohorts, who have left a legacy of camaraderie and hospitality

that greeted me from my very first day in Michigan. A special acknowledgment is due to Dan Kessler for his thoughtful and thorough support, which has played a significant part in enhancing our departmental life.

I would also like to thank everyone who contributed to my academic path leading up to my time in Michigan. There are too many inspiring teachers to list here, but I wish to give a special acknowledgement to Archer Yang for mentoring me through my first research experience in Statistics and to Mylène Bédard for her kind and thoughtful guidance through my Master's and for her encouragement to pursue a PhD.

My time in Ann Arbor has been made immeasurably richer by the friendships I've forged throughout these five years. I am eternally thankful to Brian, Moritz, and Roman for the incredible bond that began on visit day back in 2019. Our camaraderie provided me with cherished memories and a vital escape from the rigors of research. I would also like to extend my heartfelt thanks to Luke, Declan, Derek, Seamus, and Gabe, along with many others, for the joyous moments we've shared during flag-football intramural games, cheering at Michigan sporting events, and partaking in our ritualistic Thursday bogo wings. As we each set out on diverse paths across the globe, my hope is that the distances will not hinder the continuity of these treasured relationships post-graduation. It is my sincere wish that we nurture these bonds and that they endure well beyond our time here.

I owe a profound debt of gratitude to my parents, Manon and Normand, whose unwavering support has been the cornerstone of my achievements. Their infinite understanding and encouragement have not only paved the way to my time in Michigan but have also provided an enduring source of comfort and motivation from afar throughout these five years.

Finally, I extend my deepest thanks to my wife, Jessica, who has graciously navigated the complexities of a long-distance relationship with strength and patience. If there is one silver lining to the Covid restrictions, it is that they created the circumstances for our connection and for the adoption of our beloved dog, Stella. Jessica, your presence has been an endless source of happiness and optimism during my academic pursuits, and I eagerly anticipate building an even brighter future together in the next chapter of our lives.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF APPENDICES

ABSTRACT

Dependency among observations can arise from a multitude of sources, including spatial or temporal correlation, grouped, clustered or repeated measurements, hierarchical structures, and dyadic interactions. Neglecting these interdependencies in statistical analyses may result in incorrect inferences or loss of statistical power. Conversely, adequately modeling these dependencies not only enhances the validity of our statistical inferences but also deepens our comprehension of the intricate dynamics that generate the data. In this dissertation, we propose innovative methodologies tailored to three cases, each exemplifying unique challenges of dependent data.

First, we consider an imputation task where dependency among subjects is captured by a network structure. The dual nature of the data features pairwise binary relationships between subjects and subject-specific attributes that are partially observed and need to be imputed. While it is possible to conduct imputation leveraging solely subject-wise information, the relational data embedded in the network could offer auxiliary information that enhance the imputation accuracy. To capitalize on this, we explore a joint latent space model that employs subject-specific latent variables to bridge the two data modes, simultaneously addressing the interdependencies among both subjects and attributes. By adopting a Bayesian framework, we achieve an optimal integration of all available information, leading to more accurate imputed values. We ensure the practical viability of our model through the use of variational approximations.

Second, we study a differential analysis task within the context of longitudinal data, driven by a study on microbial abundance data. To accommodate the temporal dependencies and the continuous nature of the biological data, we opt for a varying-coefficient mixed model with kernel smoothing. Through a sparsity-inducing penalty, we identify time periods of differences between experimental conditions. Our novel estimation method is the first of its kind to simultaneously account for longitudinal effects while obtaining smooth, locally-sparse estimates and permitting any sampling design, such as irregularly-sample time points or missing data. Crucially, accounting for time dependence is shown to improve estimation accuracy as well as support recovery, in comparison to an equivalent approach with independence assumptions. This provides greater confidence in the discoveries obtained when

applied to the motivating data, which revealed novel scientific insights that eluded detection by cross-sectional or independent approaches.

Third, we proceed with a joint inference and prediction task emerging from a brain-computer interface setting. Specifically, we aim to construct a model of the brain's electrical activity, as captured by multiple electroencephalogram (EEG) channels over time, in relation to the brain's response to visual stimuli presented under an oddball paradigm, in which only certain stimuli are pertinent. To that end, we propose a Bayesian model explaining the measured response to target and nontarget stimuli, with particular interest in the difference in responses. This model not only explains the observed data but also generates predictive probabilities, thus enabling the classification of unlabeled responses. Our model innovates in three ways over existing approaches. We aggregate EEG channels in a collection of latent factors, which abstracts the EEG system design and naturally capture spatial correlation. Additionally, our model allows temporal variations in spatial covariance, thus supporting dynamic modeling of the covariance structure. Lastly, our approach not only characterizes mean differences in response based on stimulus type but also allows for the dynamic spatial covariance to be modulated by the type of stimulus being presented. Experimental validation confirms that our framework yields prediction accuracy on par with that of discriminative methods. However, the flexibility inherent in our generative model provides deeper understanding of the underlying brain functions responsible for the responses.

CHAPTER 1

Introduction

In many real-world settings, data points cannot be observed independently of each other. To produce convincing analyses, statisticians are tasked with identifying and addressing these various sources of dependencies. In some cases, inter-dependence between observations is a *bug* that needs to be accounted for in order to reach robust and meaningful conclusions. In other cases, dependency is rather a *feature* of the data that can be leveraged to improve performance or to be studied for its own sake. Either scenario calls for modeling approaches that explicitly incorporate dependencies among samples. In this dissertation, we study three seemingly-unrelated combinations of problems and solutions; yet, the unifying theme is that they all exemplify the necessity and potential of modeling dependencies.

In non-mathematical terms, *statistical dependence* can be broadly defined as "a type of relation between any two features of units under study" (Wermuth and Cox, 2005). Dependence may arise for various reasons, depending on the particular attributes and units involved. For example, in longitudinal studies, multiple measurements of a variable of interest are recorded over time from a group of subjects. In such cases, the repeated observations for each subject are likely to be correlated with one another, and this dependency is an important consideration that cannot be overlooked (Chapters 3 and 4).

More generally, we can encompass such *temporal* dependency within a broader class of *within-unit* dependencies, where units are presumed to be independent, while the multiple measurements within each unit are interdependent. With temporal dependency, the primary concern lies with correlations along the time axis; however, dependencies can also exist among various measured variables. A closely related concept is that of *spatial* dependency, where a variable is measured across multiple locations for each unit, as explored in Chapter 4. In these cases, it is often crucial to understand the patterns of connectivity or to quantify the influence range. When different variables are measured for a particular subject, it is typically expected that these variables will not be entirely independent of one another. Indeed, exploring inter-variable relationships is fundamental to regression and association analyses, which seek to understand these relationships even when there is no specific "outcome" variable (Chapter 2).

Between-unit dependency occurs when there exists relationships between units. This often arises through hierarchical organization, with the classical example of students-teachers-schools dynamics. If an analyst is examining a particular outcome of interest at the student level, across various classrooms and schools, it is critical to account for the group effects exerted by both the teacher and the school, as these factors are likely to influence individual outcomes uniformly within each group. Beyond hierarchical structures, units can also be linked through pairwise relationships, as discussed in Chapter 2, where the measurement of one unit is influenced by those it is connected to (Chapter 2).

Of course, the classes and examples of dependencies previously mentioned are neither exhaustive nor mutually exclusive. For instance, when multiple variables are measured over time, there can be both temporal dependency and inter-variable dependency, as is the case in the context of Chapter 4. Similarly, it is possible to observe both within-unit dependency and between-unit dependency simultaneously, such as in Chapter 2. Notably, the definition of "units" in a study can be somewhat subjective, which means that the distinction between within-unit and between-unit dependency can be ambiguous. Nevertheless, these categories are useful for identifying common patterns across data sets.

In Chapter 2, we examine an imputation task where pairwise relationships among units complement the partially observed variables. Our goal is to capitalize on both the relationships between subjects and the dependencies among variables to improve the prediction of missing values. We demonstrate that even in the absence of observed variables for a particular unit, the relational data enables us to provide non-trivial imputation, which is impossible without network information. This is accomplished by using latent variables to form a bridge between two modes of data: dyadic and unit-wise observations. Specifically, we embed the network in a latent space model and the attributes in a generalized latent factor model, with both modalities sharing a common latent space. Beyond imputation, this joint latent space model also offers insights into the network's structure in relation to the attributes. The proposed approach is a novel Bayesian treatment of the joint latent space model of Ma et al. (2022) estimated using variational inference: the Bayesian framework, with its natural uncertainty quantification, provides an optimal weighing of all available information and the variational inference ensures the practical efficiency of the method. Through extensive simulations and analyses of both social and citation network data, we demonstrate our methodology's superiority in comparison to the frequentist estimation of the same model, particularly in scenarios with sparse information. Moreover, it outperforms standalone generalized latent factor models, confirming the critical role of network information for precise imputation. Additionally, when supervision is limited, our approach outperforms specialized semi-supervised node classification methods, indicating a more effective use of network

structure. The content of this chapter is based on Fontaine et al. (2024c), a joint work with Ji Zhu and Jian Kang.

In Chapter 3, we consider a longitudinal regression problem where temporal dependency is inherent due to the sequential, repeated measurements. Specifically, we explore *varying-coefficients mixed models* (VCMM) which yield parameter estimate that are smooth over time while simultaneously accounting for longitudinal effects and permitting flexible regression designs. We introduce a novel estimation method producing *locally sparse* varying coefficients, based on penalized kernel smoothing, which enables identification of differential time interval. Our method stands out in its generality; it is the first, to our knowledge, capable of yielding smooth and locally-sparse estimates that also consider serial correlation and are adaptable to any regression and sampling design. Notably, this allows us to move beyond simple group comparisons or regular sampling schemes. The experimental results underscore that the inclusion of longitudinal effects significantly enhances the efficiency of the estimates compared to those obtained under the assumption of independence. The materials presented in this chapter are adapted from Fontaine et al. (2024a), a joint work with Ji Zhu and Gen Li. This research is driven by a longitudinal microbial differential analysis, particularly focusing on data provided by Nisha J. D'Silva, Marcell Costa de Meideros, and Grace Y. Chen. They aim to identify microbial taxa that are associated with disease progression and with a tumor suppressor gene's presence. The complexity of the analysis necessitated a regression design that incorporated interaction terms and adjustments for covariates, which traditional group comparison methods could not accommodate. Moreover, due to experimental constraints, the resulting incomplete data records over time precluded the application of existing methods. Our approach successfully identified several taxa correlated with the interaction between genotype and diagnosis, indicating a potential mediating role of the microbiome.

In Chapter 4, we delve into the world of neuroscience by modeling brain activity, as measured by electroencephalography (EEG). We are particularly interested in the brain-computer interface (BCI) setting, where we study the brain's response to stimuli presented to the user. Specifically, by analyzing the electrical activity following the onset of visual stimuli, a BCI system can infer the subject's intent, allowing direct communication from the brain. This technology thus bypasses verbal or typed communication, which requires motor control that is unavailable to patients suffering from the "locked-in" syndrome. Research has shown that the brain's response to target stimuli share some similarities across people. For example, the P300 event-related potential (ERP) is characterized by an increase in electrical activity in the fronto-central region of the brain roughly 300ms after a target stimulus' onset. However, this ERP does not materialize at the same latency and location across users,

specifically with age, condition and fatigue. Additionally, several other signals have been identified to be predictive of stimulus type (target or non-target). Hence, there is strong interest from researchers to better understand the underlying brain processes responsible for information processing and decision-making. To this end, we proposed a latent factor model able to identify important subcomponents of the brain activity in relation to stimulus type and time since onset. In particular, the latent factors decompose the brain activity across electrodes and across time into a few components defined by electrode weights and temporal variations. An important aspect of EEG is the spatial dependency across electrodes, sometimes referred to as *functional connectivity*. The latent factor structure naturally captures the spatial correlation through a low-rank model, which abstracts the BCI system design. Additionally, changes in spatial dependency may contain important information about the brain processes by capturing synchronicity and communication patterns. Thus, we propose a dynamic spatial covariance model where it is allowed to vary both with time since stimulus onset and with stimulus type. Applied to real BCI sessions provided by Jane E. Huggins, our model successfully a handful of signals associated with stimulus response, some of which matches with known ERPs and some identifying new patterns. Additionally, the changes in spatial covariance provides novel insights into the functional connectivity associated with ERPs. The contents of this chapter are adapted from Fontaine et al. (2024b), a collaboration with Ji Zhu and Jian Kang.

CHAPTER 2

Missing Value Imputation in Relational Data using Variational Inference

Abstract. In real-world networks, node attributes are often only partially observed, necessitating imputation to support analysis or enable downstream tasks. However, most existing imputation methods overlook the rich information contained within the connectivity among nodes. This research is inspired by the premise that leveraging all available information should yield improved imputation, provided sufficient association between attributes and edges. Consequently, we introduce a joint latent space model that produces a low-dimensional representation of the data and simultaneously captures the edge and node attribute information. This model relies on the pooling of information induced by shared latent variables, thereby enhancing the prediction of node attributes and providing a more effective attribute imputation method. Our approach uses variational inference to approximate posterior distributions for these latent variables, resulting in predictive distributions for missing values. Through numerical experiments, conducted on both simulated data and real-world networks, we demonstrate that our proposed method successfully harnesses the joint structure information and significantly improves the imputation of missing attributes, specifically when the observed information is weak.

2.1 Introduction

Missing values occur naturally in a wide variety of real-world applications and pose challenges for statistical analysis and machine learning. While there has been extensive work done on imputing missing values in standard $N \times p$ data matrices (see Little and Rubin, 2019; Lin and Tsai, 2020, for recent reviews), less common data structures may benefit—or even require—methods tailored to their particularities. In this paper, we consider network data with node

attributes, which consists of two data components: the graph, composed of a collection of N nodes and links among them, and a set of p attributes for each node. In many applications, the graph is fully observed, that is, whether two nodes are connected is always known, but the node attributes are only partially observed. As an example, relationships in a social network are generally fully accessible while the user profiles may be incomplete. Various downstream tasks may necessitate complete data, in which case imputing the missing entries becomes a crucial step of the analysis pipeline.

A naïve solution would be to forget about the graph and use a vanilla imputation method on the $N \times p$ attribute matrix. However, meaningful information may thus be discarded, leading to poorer imputations. Indeed, the node attributes can contain relevant information on link formation and vice versa. Another simple solution would be to use network smoothing ideas. For example, one could predict missing attributes by averaging (observed) neighboring attributes. Unfortunately, this method also suffers from not fully utilizing all available information: there often is information in the absence of links and this method cannot share information between the p attributes. It is also sensitive to the density of the network and the proportion of missing values. This suggests developing methodology that considers edges and all attributes simultaneously.

Research focusing on this particular missing node attribute imputation problem is scarce, and none of them convincingly utilize all available information or are general enough. For example, before imputing missing link data in longitudinal social surveys, Ouzienko and Obradovic (2014) use simple imputation methods disregarding completely the graph information. Koskinen et al. (2013) also encounter a similar issue with partially observed graphs, but only discuss their method's deficiencies at dealing with missing node attributes. Chakrabarti et al. (2017) propose a prediction method for categorical node attributes assuming a fixed network. For a broader review of imputation in network data, we refer the reader to Huisman and Krause (2017).

A special case of node attribute imputation in network data is transductive semi-supervised node classification, where only a few nodes are labelled, but other node attributes may be available. Various approaches to solve such problems have been proposed, such as label propagation (Zhu and Ghahramani, 2002) and graph convolutional neural networks (Kipf and Welling, 2017), among many extensions and alternatives (see Song et al., 2022, for a recent review). Our work focuses on a more general setting, where missing values can occur for multiple attributes. Furthermore, we do not assume any structure in the missingness as we do not require that any node has fully-observed attributes.

There have been multiple proposals to include side information in commonly-used network models, such as in *latent space models* (Hoff, 2009; Miller et al., 2009) and in *stochastic*

blockmodels (Kim et al., 2012; Sweet, 2015). While these methods are powerful on their own for link predictions, most treat the side information as fixed covariates and therefore cannot proceed to imputation. There exist also various methods for network embeddings that allow side information (see Cai et al., 2018, for a review), but all treat node attributes as input, which makes it hard to deal with incomplete data. One common thread of the previous examples is that no model is specified for side information, which are then required to be fully observed. We thus propose a *joint latent space model* approach where we learn a low-dimensional representation of the nodes, known as their *latent positions*, that simultaneously explains the network connectivity and the node attributes. Sharing the latent variables across the edge model and the attribute models enables the propagation of information across the two data modes. By construction, the latent position of a node will be predictive of its attributes, allowing prediction and thus imputation.

Our proposed approach builds on the joint latent space model using inner products of Zhang et al. (2022), which utilizes latent variables to model the joint distribution of the graph and node attributes. Joint latent space models are agnostic to the direction of effects between network and attributes: we make no assumption that either data mode is downstream of the other. A Bayesian treatment tracks uncertainty around all quantities involved, leading to a more efficient utilization of all available information towards imputations. In particular, we find the largest improvements under weak or imbalance information regimes, such as small or sparse networks and few or rarely observed node attributes. Furthermore, the Bayesian approach automatically adapts to the predictive power of the network on node attributes, in contrast to frequentist approaches (Zhang et al., 2022) requiring tuning of the relative importance between edges and node attributes. Leveraging variational inference, we produce an imputation method that is computationally efficient and requires minimal tuning. In fact, our approach is robust against overfitting, so tuning is of lesser importance.

Fosdick and Hoff (2015) proposed the first joint model for edges and node attributes, restricted to continuous edges and attributes and focusing on the dependency between network effects and attributes. There are also some work on network mediation analysis (Liu et al., 2021a; Che et al., 2021) which utilizes joint latent space models, but typically focus on a single node attribute. Item-response models, targeting binary node attributes, have also been extended to include network dependency through joint latent space models (Wang et al., 2023) based off the latent space model of Jin and Jeon (2019). Liu et al. (2018) propose a joint model for continuous attributes only and Gu and Yu (2022) for ordinal attributes only. Our proposed joint latent space model allows for mixed attribute types, unifying the aforementioned models, though we consider an inner product structure rather than a Euclidean distance one. Finally, we can see our proposed model as a special case of the dynamic

joint latent factor model of Guhaniyogi and Rodriguez (2020), which is also based on the inner product model and allows mixed attributes. In addition, none of these methods directly provide predictions for missing node attributes, as the objective is the low-dimensional representation itself or link prediction.

Variational inference is becoming increasingly popular among statisticians for its improved computational efficiency compared to *Markov chain Monte Carlo* (MCMC) to perform approximate Bayesian inference in complex models. There have been various recent developments in terms of algorithms, application and theory; see Blei et al. (2017) and Zhang et al. (2019) for recent reviews. In the context of network models, variational inference has been applied to a variety of models including the stochastic blockmodel (Daudin et al., 2008) and extensions thereof (Airoldi et al., 2008; Mariadassou et al., 2010; Xing et al., 2010; Ho et al., 2011; Yang et al., 2011; Matias and Miele, 2017), as well as a variety of latent space models (Salter-Townshend and Murphy, 2013; Gollini and Murphy, 2016; Sewell and Chen, 2017; Lee et al., 2021; Liu and Chen, 2021), including a joint latent space model (Wang et al., 2023). Our work differs from Guhaniyogi and Rodriguez (2020) in that our static setting greatly simplifies the structure, allowing very efficient variational inference compared to their MCMC approach. Also, our variational inference algorithm, based off *variational message passing* (Winn and Bishop, 2005) is more efficient, stable and scalable than the variational inference utilized in Wang et al. (2023) based off Taylor approximations; this is enabled by using an inner product model whose functional dependency on the latent variable is linear rather than quadratic in the Euclidean distance model.

In Section 2.2, we introduce the joint latent space model, which allows information sharing between the two data components through common latent variables. In Section 2.3, we describe our estimation procedures using variational inference as well as the imputation method through posterior predictive distribution. In Section 2.4, we use numerical experiments on both synthetic and real-world data to compare the performance of our proposed `NAIVI` methods, short for *Node Attribute Imputation using Variational Inference*, with some benchmark methods. We provide a Python implementation available at `https://www.github.com/fontaine618/NAIVI`. Calculations, extensions, and additional numerical results can be found in the Supplementary Material.

2.2 Joint latent space model

We first introduce the notation. We use $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ to denote a graph consisting of a set of N nodes $\mathcal{V} = [N] := \{1, \dots, N\}$ and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We summarize $\mathcal{G}$ by its adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$, where the (u,v)-entry a_{uv} is equal to 1 if $(u,v) \in \mathcal{E}$ and 0

otherwise. We consider *undirected* edges, that is, $a_{uv} = a_{vu}$. For each node $u \in \mathcal{V}$, we have a vector of p attributes $\mathbf{x}_u^\top = (x_{u1}, \ldots, x_{up})$ with entries in $\mathbb{R}$; we aggregate all N attribute vectors in the attribute matrix $\mathbf{X} \in \mathbb{R}^{N \times p}$ with rows $\mathbf{x}_1^\top, \ldots, \mathbf{x}_n^\top$.

To each node $u \in \mathcal{V}$, we associate two latent variables. First, the *latent position* of node u is a vector $\mathbf{z}_u \in \mathbb{R}^K$, for some common latent dimension $K \in \mathbb{N}$, which will be used to model both the adjacency matrix $\mathbf{A}$ and the attribute matrix $\mathbf{X}$. Aggregated in the latent position matrix $\mathbf{Z} \in \mathbb{R}^{N \times K}$, these latent positions allow the sharing of information between the two parts of the model: the adjacency matrix and node attributes are assumed conditionally independent given $\mathbf{Z}$. Second, the latent *degree heterogeneity* $\alpha_u \in \mathbb{R}$ of a node $u \in \mathcal{V}$ is an additional latent variable modeling the propensity of a node of forming links with other nodes. Define $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)^\top \in \mathbb{R}^N$ as the vector of latent degree heterogeneities.

We assume all latent variables to be *random*, rendering our model Bayesian, in contrast to the approaches in Zhang et al. (2022) which consider similar models but treat latent variables as fixed model parameters. To this end, we put independent Gaussian priors on both latent positions and latent heterogeneity:

$$p_0(\mathbf{Z}, \boldsymbol{\alpha}) = \prod_{u \in \mathcal{V}} \left[\varphi\left(\alpha_u \mid \mu_\alpha, \sigma_\alpha^2\right) \varphi_K\left(\mathbf{z}_u \mid \mu_Z \mathbf{1}, \sigma_Z^2 \mathbf{I}\right) \right],$$

where $\varphi(\cdot \mid \mu, \sigma^2)$ denotes the Gaussian density with mean μ and variance σ^2, $\varphi_K(\cdot \mid \boldsymbol{\mu}, \Sigma)$ denotes the K-variate Gaussian density with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ, and where $\mu_\alpha, \mu_Z \in \mathbb{R}$, $\sigma_\alpha^2, \sigma_Z^2 > 0$ are hyper-parameters to be specified. While default values for these hyper-parameters such as mean 0 and variance 1 might seem intuitive, other values may be more appropriate (see Section 2.2.4 for a discussion). From the conditional independence between the two parts of the model and the independence of the attributes on the latent heterogeneity, the joint distribution of the model is given by

$$P(\mathbf{Z}, \boldsymbol{\alpha}, \mathbf{A}, \mathbf{X}) = p_0(\mathbf{Z}, \boldsymbol{\alpha}) P(\mathbf{A} \mid \mathbf{Z}, \boldsymbol{\alpha}) P(\mathbf{X} \mid \mathbf{Z}),$$

where $P(\mathbf{A} \mid \mathbf{Z}, \boldsymbol{\alpha})$ is the adjacency likelihood and $P(\mathbf{X} \mid \mathbf{Z})$ is the attribute likelihood, both to be defined. For ease of notation, we overload the P function whose specific instance can be recovered from the arguments.

2.2.1 Attribute models

The attribute part of the model defines the likelihood $P(\mathbf{X} \mid \mathbf{Z})$, that is, the generative model of the attributes conditional on the latent positions. For each attribute $j \in [p]$, we consider a different *generalized linear model* (GLM) conditional on $\mathbf{Z}$, in the spirit of generalized factor

models (Liu et al., 2021b). In particular, we model the expected value of X_{uj} as an affine transformation of Z_u passed through a link function g_j^{-1}:

$$\Theta_{uj}^X = b_{0j} + \mathbf{b}_j^\top \mathbf{z}_u, \qquad \mathbb{E}(x_{uj} \mid \mathbf{z}_u) = g_j^{-1}(\Theta_{uj}^X), \tag{2.1}$$

where $b_{0j} \in \mathbb{R}$ and $\mathbf{b}_j \in \mathbb{R}^K$ are model parameters corresponding respectively to the intercept (bias) and regression coefficients (weights).

We consider two cases, but our method can be extended to other GLMs, provided inference is feasible (see Section 2.5 for a discussion). First, if x_{uj} is continuous, we consider a Gaussian model with identity link:

$$x_{uj} \mid \mathbf{z}_u \sim \mathcal{N}\left(\Theta_{uj}^X, \sigma_j^2\right), \tag{2.2}$$

where $\sigma_j^2 > 0$ is the scale parameter. Second, if x_{uj} is a binary attribute, we consider a logistic model:

$$x_{uj} \mid \mathbf{z}_u \sim \text{Bernoulli}\left\{\sigma\left(\Theta_{uj}^X\right)\right\},$$

where $\sigma(x) = \{1 + \exp(-x)\}^{-1}$, i.e., the sigmoid transformation. Hence, the attribute likelihood, assuming conditional independence given latent positions, is given by

$$P\left(\mathbf{X} \mid \mathbf{Z}\right) = \prod_{u \in \mathcal{V}} \prod_{j \in [p]} P_j\left(x_{uj} \mid \mathbf{z}_u\right), \tag{2.3}$$

where $P_j(x_{uj} \mid \mathbf{z}_u) = \varphi(x_{uj} \mid \Theta_{uj}^X, \sigma_j^2)$ when x_{uj} is continuous and $\{\sigma(\Theta_{uj}^X)\}^{x_{uj}}\{1 - \sigma(\Theta_{uj}^X)\}^{1-x_{uj}}$ when x_{uj} is binary. The attribute model thus depends on several model parameters: the noise parameters σ_j^2 for all continuous attributes, the bias vector $\mathbf{b}_0 = (b_{01}, \ldots, b_{0p})^\top \in \mathbb{R}^p$ and the weight matrix $\mathbf{B} = (\mathbf{b}_1; \ldots; \mathbf{b}_p) \in \mathbb{R}^{K \times p}$, often called the *loading matrix* in the latent factor model context.

2.2.2 Adjacency model

The adjacency part of the joint model describes the assumed relationship between the latent variables $(\mathbf{Z}, \boldsymbol{\alpha})$ and the links $\mathbf{A}$, i.e., $P(\mathbf{A} \mid \mathbf{Z}, \boldsymbol{\alpha})$. Conditional on the latent variables, we assume the links are independent and follow a Bernoulli distribution with probability defined through an *inner product space model* (Hoff, 2003; Ma et al., 2020):

$$a_{uv} \mid \Theta_{uv}^A \sim \text{Bernoulli}\left\{\sigma\left(\Theta_{uv}^A\right)\right\}, \qquad \Theta_{uv}^A = \alpha_u + \alpha_v + \mathbf{z}_u^\top \mathbf{z}_v, \tag{2.4}$$

where $\Theta^A_{uv} \in \mathbb{R}$ defines the logit of the link probability. An edge between nodes u and v is thus modeled to be more likely whenever Θ^A_{uv} is larger, which occurs when either nodes have larger node heterogeneity α_u, α_v or when the two latent positions point in similar directions, i.e., when $\mathbf{z}_u^\top \mathbf{z}_v$ is large and positive. By conditional independence, the likelihood is given by

$$P(\mathbf{A} \mid \mathbf{Z}, \boldsymbol{\alpha}) = \prod_{u<v} \left\{\sigma\left(\Theta^A_{uv}\right)\right\}^{a_{uv}} \left\{1 - \sigma\left(\Theta^A_{uv}\right)\right\}^{1-a_{uv}}.$$

We choose an inner product model, in contrast to a distance model, to match the topology of both models. Indeed, we can view attribute j as an extra node of the network, where the weight vector $\mathbf{b}_j$ is its latent position and the bias b_{0j} is its latent heterogeneity. Then, the "edge", now understood as a *weighted edge*, between a node u and an attribute j is the corresponding node attribute x_{uj}, with linear predictor also given by an inner product strucutre: $\Theta^X_{uj} = b_{0j} + \mathbf{b}_j^\top \mathbf{z}_u$. Additionally, the inner product structure simplifies the estimation since the dependence on the $\mathbf{z}_u$'s is linear—rather than quadratic in a distance model—, which provides enough conjugacy for local updates.

2.2.3 Missingness model

To allow unobserved attributes in the model, we simply exclude the missing entries of $\mathbf{X}$ from the attribute likelihood (2.3). This simplification can be justified via various missing value models and the conditional independence between different attributes given the latent variables. In particular, a main assumption about our model is that the latent variables contain *all* the information about the nodes, including the missingness.

Let $\mathcal{O} \subseteq [N] \times [p]$ denote the set of indices corresponding to observed attributes and $\mathcal{M} = [N] \times [p] \setminus \mathcal{O}$, the missing value indices. Then, $\mathbf{X}_\mathcal{O}$ is the set of observed attributes and $\mathbf{X}_\mathcal{M}$ denotes the set of missing values to be imputed. Define the missingness matrix $\mathbf{M} \in \{0,1\}^{N \times p}$ whose entries are missingness indicators $m_{uj} = \mathbb{1}[(u,j) \in \mathcal{M}]$ and define the actual observations as $x^*_{uj} = x_{uj}$ if observed ($m_{uj} = 0$) and $x^*_{uj} = \texttt{NA}$ if missing ($m_{uj} = 1$). Let $\mathbf{X}^*$ denote the $N \times p$ matrix with entries $x^*_{uj} \in \mathbb{R} \cup \{\texttt{NA}\}$. Since the observed attributes $\mathbf{X}^*_\mathcal{O} = \mathbf{X}_\mathcal{O}$ and the missingness matrix $\mathbf{M}$ are sufficient to reconstruct $\mathbf{X}$, the observations come from the generating model $P(\mathbf{X}^* \mid \mathbf{Z}) = P(\mathbf{X}_\mathcal{O} \mid \mathbf{M}, \mathbf{Z}) P(\mathbf{M} \mid \mathbf{Z})$. To be able to omit the missing values from the attribute likelihood (2.3), we can require $\mathbf{M}$ to be uninformative about $\mathbf{Z}$, i.e., $P(\mathbf{M} \mid \mathbf{Z}) = P(\mathbf{M})$, and that $\mathbf{X}_\mathcal{O}$ does not depend on $\mathbf{M}$ given $\mathbf{Z}$. In this case, we find $P(\mathbf{X}^* \mid \mathbf{Z}) = P(\mathbf{X}_\mathcal{O} \mid \mathbf{Z}) P(\mathbf{M})$, which imply we can restrict the joint likelihood to $\mathbf{X}_\mathcal{O} = \mathbf{X}^*_\mathcal{O}$ only.

The *missing completely at random* (MCAR, Rubin, 1976) assumption is sufficient for these

simplifications to hold, but weaker assumptions are possible. For example, in our context, a safe assumption would be that the missingness is independent between nodes (conditionally on latent positions), i.e.,

$$P\left(\mathbf{X}^* \mid \mathbf{Z}\right) = \prod_u P\left(\mathbf{x}_{\mathcal{O}_u}, \mathbf{m}_u \mid \mathbf{z}_u\right)$$

Then, further assumptions required to ignore missing values must only apply within nodes. Now, for a given node u, $\mathbf{x}_u$ may be partially observed, that is, $\mathbf{m}_u$ is non-constant. Since the GLM model is such that all of $\mathbf{z}_u$ is associated to all of $\mathbf{x}_u^*$, stronger assumptions must be made. Among the various strengthenings of the usual *missing at random* (MAR) assumption for graphical models described in Tian (2015), we note the proposed graphical-MAR assumption: a missing data model is *graphical-MAR* if no latent (or unobserved) variable are parents of missingness indicators. In practice, this extends the MCAR assumption by allowing missingness indicators to depend on other *observed* attributes. In Section B of the Supplementary Material, we discuss a possible extension of this joint latent space model where we directly model the missingness indicators as additional binary attributes, also function of the latent positions $\mathbf{z}_u$, which would move beyond MCAR or graphical-MAR.

2.2.4 Model properties

The joint model is completely determined by the following hyper-parameters: the prior parameters $\mu_Z, \sigma_Z^2, \mu_\alpha, \sigma_\alpha^2$, the bias vector $\mathbf{b}_0$ and weight matrix $\mathbf{B}$ and the continuous attributes' noise variance $\boldsymbol{\sigma}^2 = \{\sigma_j^2 \mid \text{attribute } j \text{ is continuous}\}$.

For the attribute model, the interaction between all these parameters leads to identifiability issues. Indeed, we have the marginals $\Theta_u^X \sim \mathcal{N}(\mathbf{b}_0 + \mu_Z \mathbf{B}^\top \mathbf{1}, \sigma_Z^2 \mathbf{B}^\top \mathbf{B})$, and, moreover, for continuous attributes, $\mathbf{x}_u \sim \mathcal{N}(\mathbf{b}_0 + \mu_Z \mathbf{B}^\top \mathbf{1}, \sigma_Z^2 \mathbf{B}^\top \mathbf{B} + \operatorname{diag} \boldsymbol{\sigma}^2)$, which implies that the model is identifiable only up to $\mathbf{b}_0 + \mu_Z \mathbf{B}^\top \mathbf{1}$ and up to $\sigma_Z^2 \mathbf{B}^\top \mathbf{B}$ or $\sigma_Z^2 \mathbf{B}^\top \mathbf{B} + \operatorname{diag} \boldsymbol{\sigma}^2$. For convenience, we thus choose to set $\mu_Z = 0$ leading to the identifiability of the biases $\mathbf{b}_0$ as well as to better interpretability. This choice is also reasonable for the adjacency model, as the logit link probability will then be centered around the node heterogeneity (in particular, we could transfer any off-centering to the heterogeneity). Similarly, we also choose to fix σ_Z^2 to a fixed value (typically 1) to improve the identifiability of the regression weights $\mathbf{B}_j$.

The adjacency model is completely determined by the prior parameters. By fixing the prior parameters for the latent positions, μ_Z, σ_Z^2, we fix the centering and scale of the inner products, $\mathbb{E}(\mathbf{z}_u^\top \mathbf{z}_v) = K\mu_Z^2$, and $\operatorname{Var}(\mathbf{z}_u^\top \mathbf{z}_v) = K(2\mu_Z^2\sigma_Z^2 + \sigma_Z^4)$. For example, if we choose $\mu_Z = 0$ and $\sigma_Z^2 = 1$, we find that the inner products are centered at 0 with variance K. Adding the node heterogeneity, we find $\mathbb{E}\left\{\Theta_{uv}^A\right\} = 2\mu_\alpha$, and $\operatorname{Var}\left(\Theta_{uv}^A\right) = 2\sigma_\alpha^2 + K$. The network density will be dictated by the average link probability so that an appropriate value

for μ_α can be easily obtained. In particular, we suggest an empirical Bayes estimate of $\mu_\alpha, \sigma_\alpha^2$ obtained using the mean and variance of the empirical distribution of logit degrees.

Similarly to latent factor models, the joint latent space model is invariant under rotations. Indeed, we can rotate all positions $\mathbf{Z}$ and apply the inverse rotation to the loading matrix $\mathbf{B}$ and obtain the exact same joint likelihood. Since the prior for $\mathbf{Z}$ is spherical, the prior is also invariant under such rotations, so the full model is invariant. However, this unidentifiability does not cause any issue for inference since initialization will fix a rotation and subsequent update of $\mathbf{Z}$ and $\mathbf{B}$ holds the other quantity fixed.

Conceptually, any submodels $P(\mathbf{X} \mid \mathbf{Z})$ and $P(\mathbf{A} \mid \mathbf{Z})$ can be utilized to produce accurate predictions. However, the specific choices we make provide relevant properties. First, choosing the same inner product structure for either parts, i.e, $\mathbf{z}_u^\top \mathbf{z}_v$ and $\mathbf{z}_u^\top \mathbf{b}_j$, enables improved interpretability of the joint latent space, which allows exploratory analyses and visualizations. In fact, we can reinterpret $\mathbf{b}_j$ as the latent position of attribute j. For example, given a binary attribute j, we can plot the latent positions $\{\mathbf{z}_u\}_u$ together with the linear boundary induced by $\mathbf{b}_j$ and b_{0j}. Second, that same inner product structure greatly simplifies the upcoming inference as the dependency on each quantity is linear, in opposition to distance models, such as the ones used in Wang et al. (2023). Third, the range of values of inner products being unrestricted allows easier extensions to other distributions, whereas distance models suffer from the strict bound on linear predictors.

2.3 Inference

2.3.1 Imputation via the posterior predictive distribution

The joint model proposed in Section 2.2, together with observed links $\mathbf{A}$ and partially observed attributes $\mathbf{X}_\mathcal{O}$, allows us to produce imputations for the missing attributes $\mathbf{X}_\mathcal{M}$. Under the Bayesian framework, this is possible through the posterior distribution of the missing attributes given the data $\mathbf{A}$ and $\mathbf{X}_\mathcal{O}$ known as the *posterior predictive distribution*, here given by

$$P(\mathbf{X}_\mathcal{M} \mid \mathbf{A}, \mathbf{X}_\mathcal{O}) = \int P\left(\mathbf{X}_\mathcal{M} \mid \mathbf{Z}\right) p\left(\mathbf{Z} \mid \mathbf{A}, \mathbf{X}_\mathcal{O}\right) \, \mathrm{d}\mathbf{Z}, \tag{2.5}$$

where $p(\mathbf{Z} \mid \mathbf{A}, \mathbf{X}_\mathcal{O})$ is the posterior distribution of the latent variables given the data. Since $P(\mathbf{X}_\mathcal{M} \mid \mathbf{Z})$ is completely defined by the attribute model in Section 2.2.1 (up to regression parameters $\mathbf{b}_0, \mathbf{B}$ that need to be estimated), we only need to compute the latent variable posterior distribution $p(\mathbf{Z} \mid \mathbf{A}, \mathbf{X}_\mathcal{O})$. We relegate the calculations of (2.5) under either

attribute models to Section A of the Supplementary Material.

2.3.2 Variational expectation-maximization

The posterior distribution of the latent variables $p(\mathbf{Z} \mid \mathbf{A}, \mathbf{X}_{\mathcal{O}})$ is fixed given model parameters $\boldsymbol{\phi} = \{\boldsymbol{\sigma}^2, \mathbf{b}_0, \mathbf{B}\}$; different values will thus lead to different posterior distributions, producing predictions of varying quality. A common criterion to select model parameter values is to maximize the *model evidence* (or *marginal likelihood*) given by integrating out the latent variables in the joint model, i.e.,

$$P_{\boldsymbol{\phi}}\left(\mathbf{A}, \mathbf{X}_{\mathcal{O}}\right) = \int P_{\boldsymbol{\phi}}\left(\mathbf{A}, \mathbf{X}_{\mathcal{O}}, \mathbf{Z}, \boldsymbol{\alpha}\right) \, \mathrm{d}\mathbf{Z} \, \mathrm{d}\boldsymbol{\alpha}, \tag{2.6}$$

where we now make the dependence on the model parameters $\boldsymbol{\phi}$ explicit.

The global parameters $\boldsymbol{\phi}$ are here treated as fixed quantities while the local variables $\{\mathbf{Z}, \boldsymbol{\alpha}\}$ are random: this suggests using an *expectation-maximization* (EM) approach to optimize the global parameters and get posterior distributions for local variables. In particular, we optimize global parameters by solving

$$\underset{\boldsymbol{\phi}}{\text{maximize}} \quad \log P_{\boldsymbol{\phi}}(\mathbf{A}, \mathbf{X}_{\mathcal{O}}).$$

The usual E-step consists of taking the expectation of the log complete likelihood, namely,

$$Q(\boldsymbol{\phi} \mid \boldsymbol{\phi}^*) = \mathbb{E}_{\boldsymbol{\phi}^*}\left\{\log P_{\boldsymbol{\phi}}(\mathbf{A}, \mathbf{X}_{\mathcal{O}}, \mathbf{Z}, \boldsymbol{\alpha})\right\}$$

where the expectation is taken with respect to the posterior distribution of the local variables $\{\mathbf{Z}, \boldsymbol{\alpha}\}$ given current values of the global parameters $\boldsymbol{\phi}^*$, that is, $p_{\boldsymbol{\phi}^*}(\mathbf{Z}, \boldsymbol{\alpha} \mid \mathbf{A}, \mathbf{X}_{\mathcal{O}})$. However, this expectation is intractable, so we instead resort to a variational E-step (VE-step) where the expectation is rather taken with respect to a simpler distribution $q(\mathbf{Z}, \boldsymbol{\alpha})$ of the local variables. This leads to a *variational EM* (VEM) algorithm.

In particular, the VE step consists of choosing the best distribution q among some family $\mathcal{Q}$ according to the criteria

$$\underset{q \in \mathcal{Q}}{\text{maximize}} \quad \mathrm{ELBO}_{\boldsymbol{\phi}^*}(q) := \mathbb{E}_q\left\{\log P_{\boldsymbol{\phi}^*}(\mathbf{A}, \mathbf{X}_{\mathcal{O}}, \mathbf{Z}, \boldsymbol{\alpha})\right\} + \mathcal{H}(q)$$

where ELBO stands for *evidence lower bound*, and where $\mathcal{H}(q) = -\mathbb{E}_q\{\log q(\mathbf{Z}, \boldsymbol{\alpha})\}$ is the entropy of q. As its name indicates, this quantity lower bounds the model evidence: Jensen's

inequality implies that

$$\mathrm{ELBO}_{\boldsymbol{\phi}^*}(q) := \mathbb{E}_q \left\{ \log P_{\boldsymbol{\phi}^*}(\mathbf{A}, \mathbf{X}_{\mathcal{O}}, \mathbf{Z}, \boldsymbol{\alpha}) \right\} + \mathcal{H}(q) \leqslant \log P_{\boldsymbol{\phi}^*}(\mathbf{A}, \mathbf{X}_{\mathcal{O}}).$$

After rearrangement, the gap can be explicitly expressed as the Kullback-Leibler (KL) divergence between the q distribution and the true posterior,

$$\log P_{\boldsymbol{\phi}^*}(\mathbf{A}, \mathbf{X}_{\mathcal{O}}) - \mathrm{ELBO}_{\boldsymbol{\phi}^*}(q) = \mathrm{KL}\left\{ q(\cdot) \| p_{\boldsymbol{\phi}^*}(\cdot \mid \mathbf{A}, \mathbf{X}_{\mathcal{O}}) \right\} = \mathbb{E}_q \left\{ \log \frac{q(\mathbf{Z}, \boldsymbol{\alpha})}{p_{\boldsymbol{\phi}^*}(\mathbf{Z}, \boldsymbol{\alpha} \mid \mathbf{A}, \mathbf{X}_{\mathcal{O}})} \right\}.$$

While not formally a distance, the KL divergence is a measure of discrepancy between distributions, so the gap will be smaller when q is a better approximation to the true posterior. The (backward) KL divergence is an *exclusive* divergence, whose properties imply that the optimal variational approximation q only adapts to one mode of the true posterior (Minka, 2005), which is desirable in our case. Choosing q to be the exact posterior leads to a null KL divergence, and the VE step coincides with the standard E step. When $\mathcal{Q}$ does not include the true posterior, we find a non-zero gap.

The M-step then consists of optimizing the surrogate function

$$\underset{\boldsymbol{\phi}}{\text{maximize}} \quad \widetilde{Q}(\boldsymbol{\phi} \mid q^*) := \mathbb{E}_{q^*} \left\{ \log P_{\boldsymbol{\phi}}(\mathbf{A}, \mathbf{X}_{\mathcal{O}}, \mathbf{Z}, \boldsymbol{\alpha}) \right\}$$

where the expectation is taken with respect to q^*, which was optimized in the VE step using the current model parameters. Since $\mathcal{H}(q^*)$ remains unchanged, this is equivalent to maximizing the ELBO with respect to model parameters for fixed approximate posterior.

2.3.3 VE step using variational message passing

Our choice of approximating family $\mathcal{Q}$ is motivated by balancing flexibility and ease of optimization in both VE and M steps. First, by conditional independence, we find that the ELBO splits into multiple likelihood fragments that depend only on a few local variables:

$$\begin{aligned} \mathrm{ELBO}_{\boldsymbol{\phi}}(q) = &\sum_{(u,j) \in \mathcal{O}} \mathbb{E}_q \left\{ \log P_{\boldsymbol{\phi}}(x_{uj} \mid \mathbf{z}_u) \right\} + \sum_{u<v} \mathbb{E}_q \left\{ \log P_{\boldsymbol{\phi}}(a_{uv} \mid \mathbf{z}_u, \mathbf{z}_v, \alpha_u, \alpha_v) \right\} \\ &+ \sum_u \mathbb{E}_q \left\{ \log p_0(\mathbf{z}_u) \right\} + \sum_u \mathbb{E}_q \left\{ \log p_0(\alpha_u) \right\} + \mathcal{H}(q). \end{aligned}$$

Hence, a natural choice of assumption on $\mathcal{Q}$ is that it factorizes over all local variables so that the marginal distributions are readily available. This is sometimes referred to as *mean-field variational inference*: $q(\mathbf{Z}, \boldsymbol{\alpha}) = \prod_u q(\mathbf{z}_u) q(\alpha_u)$. This assumption also splits the

entropy terms across all local variables so that the prior and entropy terms aggregate in KL divergence terms for each variable:

$$\sum_u \mathbb{E}_q \left\{\log p_0(\mathbf{z}_u)\right\} + \sum_u \mathbb{E}_q \left\{\log p_0(\alpha_u)\right\} + \mathcal{H}(q) = \\ - \sum_u \mathrm{KL}(q(\mathbf{z}_u) \| p_0(\mathbf{z}_u)) - \sum_u \mathrm{KL}(q(\alpha_u) \| p_0(\alpha_u)).$$

Then, we choose Gaussian distributions for each term in q, which will greatly simplify the calculations required in both VE and M steps, that is,

$$q(\mathbf{z}_u) = \varphi_K(\mathbf{z}_u \mid \mu_{\mathbf{z}_u}, \Sigma_{\mathbf{z}_u}), \qquad q(\alpha_u) = \varphi(\alpha_u \mid \mu_{\alpha_u}, \sigma^2_{\alpha_u}).$$

Then, the VE step boils down to finding the values of the variational parameters $\{\mu_{\mathbf{z}_u}, \Sigma_{\mathbf{z}_u}, \mu_{\alpha_u}, \sigma^2_{\alpha_u}\}$ that maximize the ELBO. We note a significant difference with the variational approximation proposed by Wang et al. (2023), where the covariance is shared across nodes, i.e., $\Sigma_{\mathbf{z}_u} \equiv \Sigma_{\mathbf{Z}}$ for all u. This prevents adapting to the varying amount of information across nodes: nodes with higher degree or with fewer missing attributes should have a smaller posterior variance.

To this end, we employ *variational message passing* (VMP, Winn and Bishop, 2005; Minka, 2005). While VMP was originally designed for fully conjugate models, our model contains non-conjugate fragments. In particular, Gaussian distributions are not conjugate to the logistic fragments present in both the adjacency model and the binary attribute models. There have been multiple proposals to extend VMP to non-conjugate models, especially for logistic fragments (Saul and Jordan, 1998; Jaakkola and Jordan, 2000; Knowles and Minka, 2011; Nolan and Wand, 2017). After experimentation, we settled on the quadratic bound of Jaakkola and Jordan (2000) for stability and efficiency. To compute expectations of Guassian-logistic integrals for prediction, we utilize the normal mixture approximation proposed by Monahan and Stefanski (1989). Complete details on the VMP scheme are presented in Section A of the Supplementary Material.

2.3.4 M step

The optimization of the global parameters given a posterior approximation q obtained from VMP can be done analytically using intermediary representations emerging from the VMP procedure. In particular, we can understand the VEM algorithm using VMP in the VE step as a broader VMP algorithm on all variables (Dauwels et al., 2009), including both the local variables $\mathbf{Z}, \boldsymbol{\alpha}$ and the global parameters $\boldsymbol{\phi}$. The difference is that the variational family for

the global parameters is chosen to be point masses at the parameter value. The message passing then provides natural update rules for the global parameters (see Section A of the Supplementary Material for details). This approximating family also departs from the one in Wang et al. (2023), where a multivariate Gaussian distribution is assigned to each attribute latent position $\mathbf{b}_j$ (again with shared covariance).

2.4 Numerical experiments

Competing methods We compare our proposed method, labelled `NAIVI`, to the same model estimated by the maximum a posterior (`MAP`) as well as the projected MLE (`PMLE`) proposed by Zhang et al. (2022). We also include a version of `NAIVI` where we omit the edges (`GLFM`), that is, a generalized latent factor model estimated using VMP. We do not include the closely-related method of Wang et al. (2023) as we found the implementation to be too numerically unstable for practical use. We also propose a simple method to impute missing attributes in a network by iteratively imputing the mean of a node's neighbors until convergence (`Smooth`); this is closely related to label propagation, and, particularly, to a semi-supervised extension proposed by Liu et al. (2014). For one experiment, we also compare our method to the semi-supervised graph convolutional neural network approach of Kipf and Welling (2017, `GCN`). We also compare our method to common imputation methods that do not utilize network information: `MICE` (Van Buuren et al., 1999), K-nearest-neighbors imputation (Troyanskaya et al., 2001, `KNN`) as well as mean imputation (`Mean`). When data is simulated from our model, we also include metrics using the true generating values (`Oracle`). Implementation details for each method can be found in Section C of the Supplementary Material. All code for running experiments and processing results can be found at https://www.github.com/fontaine618/NAIVI. In the various plots that follow, we will display methods that use network information with shades of blue (`NAIVI`, `MAP`, `PMLE`, `Smooth` and `GCN`) and methods that do not in shades of green (`Mean`, `MICE`, `KNN` and `GLFM`.)

Missingness mechanisms We consider three ways to generate missing values from complete data. First, the `Uniform` mechanism simply samples entries in $\mathbf{X}$ at random with some pre-determined rate. Second, the `Row deletion` approach samples nodes uniformly at random and then sets their entire rows in $\mathbf{X}$ to be missing: we then get two sets of nodes, one with all attributes observed and one with all attributes missing. Third, the `Triangle` mechanism samples entries uniformly in each column of $\mathbf{X}$ but with varying rate such that the proportion of missing values matches a pre-determined value across all columns: in this method, some attributes will be very sparsely observed, while others will be rarely missing.

Comparing `Row deletion` to either `Uniform` or `Triangle` with all else held fixed allows to study how much we can learn from the other attributes. Indeed, prediction in the `Row deletion` can only be based on the network information.

2.4.1 Synthetic data

For this first set of experiments, we generate data from the true generative model (sample from the prior, compute $\mathbf{\Theta}^X$ and $\mathbf{\Theta}^A$, sample $\mathbf{X}$ and $\mathbf{A}$). In particular, we generate latent positions in $K = 5$ dimensions with prior $\mathcal{N}_5(\mathbf{0}, \mathbf{I})$ and latent heterogeneities with prior $\mathcal{N}(\mu_\alpha, 1)$ where μ_α is chosen to induce a target edge density (for example, $\mu_\alpha = -2$ produces an edge density around 10%). The bias vector $\mathbf{b}_0$ and weight matrix $\mathbf{B}$ are both generated from iid $\mathcal{N}(0, 1)$ entries.

We consider four settings, where we alternatively vary network size (A), number of binary attributes (B), edge density (C) and missing value rate (D) while holding all others fixed ($N = 200$ nodes, $p = 100$ binary attributes, 12% edge density and 50% missing rate). We measure imputation performance using the area under the receiver operator characteristic (AuROC) between the predicted probability and the missing binary values, aggregated over all p attributes. Experiments are repeated for both the `Uniform` and `Row deletion` missingness mechanisms; medians across 30 repetitions are reported. Results can be found in Figure 2.1.

In the `Row deletion` scenario, we note that methods using only attribute information can at best predict a constant value, since no information is available for attributes to be predicted. Hence, `GLFM`, `KNN` and `MICE` all perform exactly as `Mean` imputation.

In setting A, increasing the number of nodes has two effects. First, we have more information to estimate the latent positions since we get additional pairwise relationships with each new node. Second, increasing the number of rows in $\mathbf{X}$ improves the estimation of each attribute model. `GLFM` only benefits from the second effect, so its prediction performance is capped by the uncertainty in estimating positions. We also note that as N increases, we run into posterior contraction for latent variables so that point estimate methods (`MAP`, `PMLE`) attain the same performance as `NAIVI`; all three methods approach oracle performance for large N. For smaller network size, we find that `NAIVI` significantly outperforms its point estimate counterparts. Setting D essentially disentangles the two effects in setting A as increasing the missing rate worsens the estimation of the attribute models while keeping the network information fixed.

Setting B, where the number of attributes increases, shows a similar comparison. For small p, most of the information is contained in the network, so methods relying only on

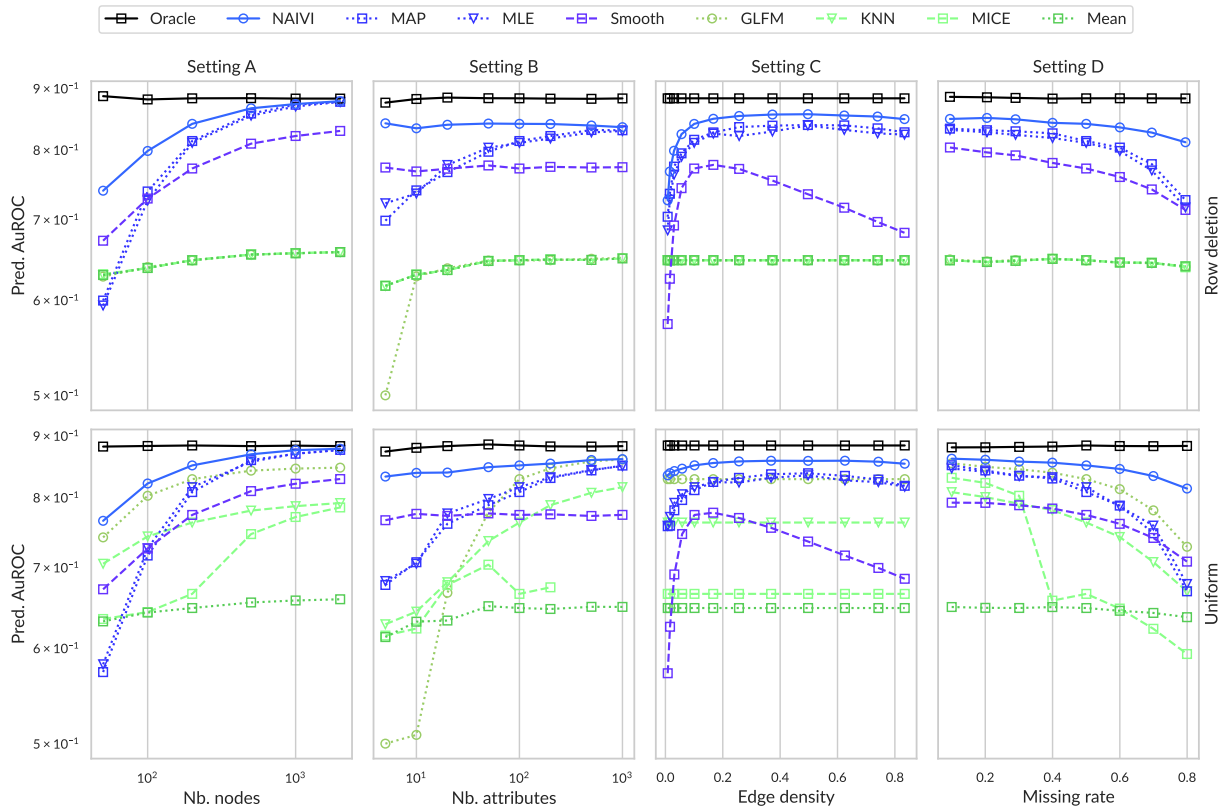

Figure 2.1: Area under the ROC curve for the predicted missing values averaged over all attributes for simulation settings A–D (columns) and for two missing mechanisms (rows). Median across 30 replications is displayed. `MICE` imputation for large p is omitted due to large computational cost.

attributes do much worse. For large p, most of the information is now contained in the attribute matrix: we find that `GLFM` achieves similar performance as `NAIVI` in the `Uniform` scenario. Similarly to setting A, we observe a significant difference between `NAIVI` and the two point estimate methods for smaller p, where the total information is small (recall that $N = 200$ in setting B). The `Smooth` method is essentially independent across attributes and therefore performs the same no matter p. `GLFM` has convergence issues for small p: the approximate posterior for the latent positions $\mathbf{z}_u$ is based on average on only $p \times 0.5$ binary observations, so covariance matrices become singular quite often, especially for $K = 5$. `NAIVI` does not suffer from this issue since the network provides sufficient information.

Setting C also varies the strength of the information contained in the network. When the edge density is very small, there is not much entropy among edges, which prevents learning latent positions well. Similarly to setting B, `GLFM` achieves similar performance as `NAIVI` when the network information vanishes. As expected, the three latent space models

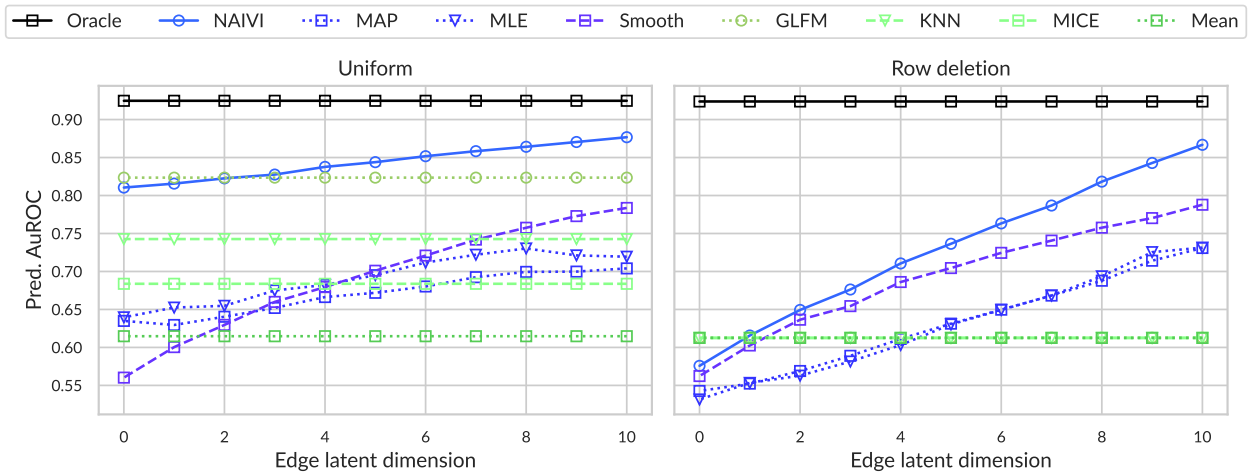

Figure 2.2: Area under the ROC curve for the predicted missing values averaged over all attributes under simulation setting E. "Edge latent dimension" refers to the number of latent dimensions K_e, out of $K = 10$, used to generate link probabilities. Increasing values of K_e corresponds to increased predictive power of the network on the node attributes.

achieve peak performance when the edge density is around 50%, corresponding to maximum network information. Interestingly, the `Smooth` imputation peaks around 10-15%: if nodes have very few neighbors, then the average will be very noisy, leading to poor predictions; when the degree increases, we take the mean over increasingly-many nodes and lose locality, so predictions approach the overall mean. In fact, this shows the power of modeling the edges along with the attributes, compared to treating the network as fixed: `Smooth` cannot leverage the information contained in the absence of an edge, as can be seen from the non-symmetric curve around 0.5.

We also consider a fifth synthetic experiment (setting E) to study the importance of the association between network information and attributes. We generate data from the model itself as previously with $N = 200$ nodes, $p = 100$ binary attributes, edge density around 10% and introduce missing values at a 50% rate in both `Row deletion` and `Uniform` scenarios. Latent positions are generated from a 10-dimensional spherical Gaussian, but not all dimensions are used to generate link probabilities. In particular, the inner products $Z_u^\top Z_v$ are replaced by the inner product of the last K_e dimensions only. For example, when $K_e = 3$, then 3 dimensions are used to generate both edges and attributes and the other 7 dimensions only generate attributes. When all 10 dimensions are used to generate edges, then `NAIVI` is correctly specified; when no dimensions are used, then `GLFM` is correctly specified. Results, displayed in Figure 2.2, show that, as long as there is *some* network information, `NAIVI` can leverage it and will outperform `GLFM`. Notably, when network information vanishes, some

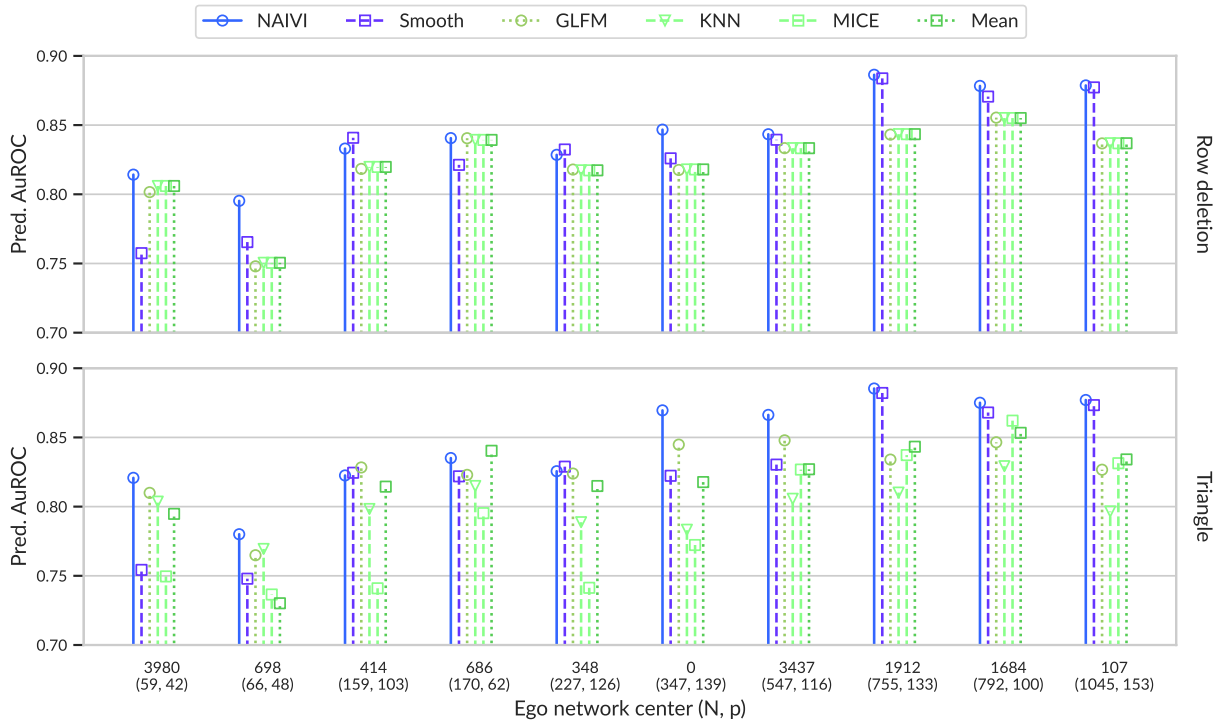

Figure 2.3: Median predictive AuROC across 30 missing value samplings for the Facebook™ ego networks (Leskovec and Mcauley, 2012).

methods can perform worse than `Mean` imputation because of overfitting; `NAIVI` is less prone to this behavior. This suggests that `NAIVI` is robust to the predictive power the network has on attributes; when there is essentially no such predictive power, `NAIVI` does not do considerably worse than the correctly-specified model.

2.4.2 User information imputation in social networks

In this second experimental setting, we consider real-world data in order to study the performance of our proposed method applied to data that were not generated exactly from the underlying model. In particular, we analyze ego networks emerging from the Facebook™ social network compiled by Leskovec and Mcauley (2012).[1] The *ego network* of a given user is defined as the social network with nodes corresponding to all other users connected to that user and all edges among those users within the original network. The data set in Leskovec and Mcauley (2012) contains 10 such ego networks.

The size of each ego network varies from 59 to 1045 users with edge density between

[1]The data was accessed through the Stanford Large Network Dataset Collection (Leskovec and Krevl, 2014).

3.2 and 13.5%. Along with the friendships, we are provided with binary node attributes describing various elements such as education, work, languages, etc. Each ego network has between 42 and 154 binary attributes; attributes with variance below 0.01 were discarded.

The data in Leskovec and Mcauley (2012) contains no missing values, so we artificially mask 50% of the attribute matrix using the `Row deletion` and `Triangle` schemes. Model selection, presented in Section D of the Supplementary Materials, finds that the best latent dimension ranges from $K = 2$ for the smaller networks to $K = 5$ for the largest network.

The predictive AuROC across 30 replications can be found in Figure 2.3. We find that `NAIVI` generally outperforms all other imputation methods considered across all ego networks and missing value scenarios, though some cases show multiple methods achieving similar performance. In the `Row deletion` scenario, only `NAIVI` and `Smooth` can produce non-trivial predictions. We find that joint modeling (`NAIVI`) improves significantly on the per-attribute approach (`Smooth`) for the smaller ego networks. In the `Triangle` scenario, where attribute-only methods can now provide individualized predictions, we find that large performance gap between `NAIVI` and the rest for networks 0 and 3437.

The network for ego center 686 seems to have no predictive power on the attributes, as we observe a similar pattern in Figure 2.2. Indeed, in the `Triangle` case, methods using network information do worse than `Mean` imputation and achieve similar performance in the `Row deletion` scenario; only `NAIVI` remains on par with `Mean`, further suggesting our method is robust against overfitting. All other ego networks contain some predictive power of the attributes.

Comparing the `Row deletion` and `Triangle` scenarios shows that only a few networks benefit from learning from other covariates: only centers 0 and 3437 show a noticeable increase in prediction performance for `NAIVI` between the two scenarios. For these two networks, we note that `NAIVI` improves on `GLFM`, indicating that the edges and the other attributes contain complementary information useful for prediction.

2.4.3 Network-guided semi-supervised classification

Another use-case for node attribute imputation in networks is node label prediction, where only a few labels are given. This setting falls under the transductive semi-supervised learning task category. We consider two commonly-used datasets of networks with ground-truth communities, one of which also contains additional fully-observed attributes we can leverage.

First, the *email-EU-core* dataset (Yin et al., 2017, accessed from Leskovec and Krevl, 2014) records email communications within a European research institute. We construct an adjacency matrix by symmetrizing the existence of an email between two of the 1005 people.

Each person is also assigned to exactly one of 42 departments.

Second, the *Cora* dataset (Sen et al., 2008) is a citation network of 2708 publications classified in exactly one of seven categories. Again, we construct the symmetrized adjacency matrix, where links correspond to citations in either direction. The dataset also contains the presence/absence of 1433 unique words.

For our class prediction experiment, we sample a fixed number of *seeds* (or *context*) within each class (department or topic) whose labels will be known for training; all remaining labels will be used for evaluating predictions. For the Cora dataset, we keep the word presence/absence matrix for all nodes.

We evaluate the prediction performance of `NAIVI` and other competing methods along a sequence of increasing number of known labels per class, indicating an increasing level of supervision. We report the F1 score weighted by class frequency and the overall accuracy, aggregated in the median over 30 replications. Results can be found in Figure 2.4.

As expected, the performance of all methods increases as the amount of supervision increases. `NAIVI` retains a lot of predictive power even when a single label per class is available; this is particularly apparent in the Email dataset, where no other attributes are available. This shows that the network is by itself good at separating the classes, and we only need a few labels to align the predictions. Similarly, in the Cora dataset, we find that `Smooth` performs almost as well as `NAIVI`, indicating that it is mostly the citations that contribute to the prediction; in comparison, `MICE` produces a linear classifier trained on the word composition and cannot achieve the same predictive performance as the method using the network. `GCN`, a method specifically designed for such a task, requires more supervision to be able to slightly outperform `NAIVI`, showing that our general-purpose method is still competitive in settings where more specialized approaches should shine. In particular, `NAIVI` isn't designed to estimate latent positions that are primarily predictive of the topics; it produces latent positions that are predictive of everything.

2.5 Discussion

In this paper, we propose a Bayesian treatment of a joint latent space model for edges and node attributes using variational inference. The Bayesian setting features several advantages compared to the MLE approach of Zhang et al. (2022). First, tracking the uncertainty of all quantities involved provides a natural weighting of the two submodels: indeed, we do not require a tuning parameter controlling the information contribution coming from edges and attributes. Second, the ELBO is a cheap and reliable criterion for determining the number of latent dimensions required; in particular, no cross-validation is required, which would

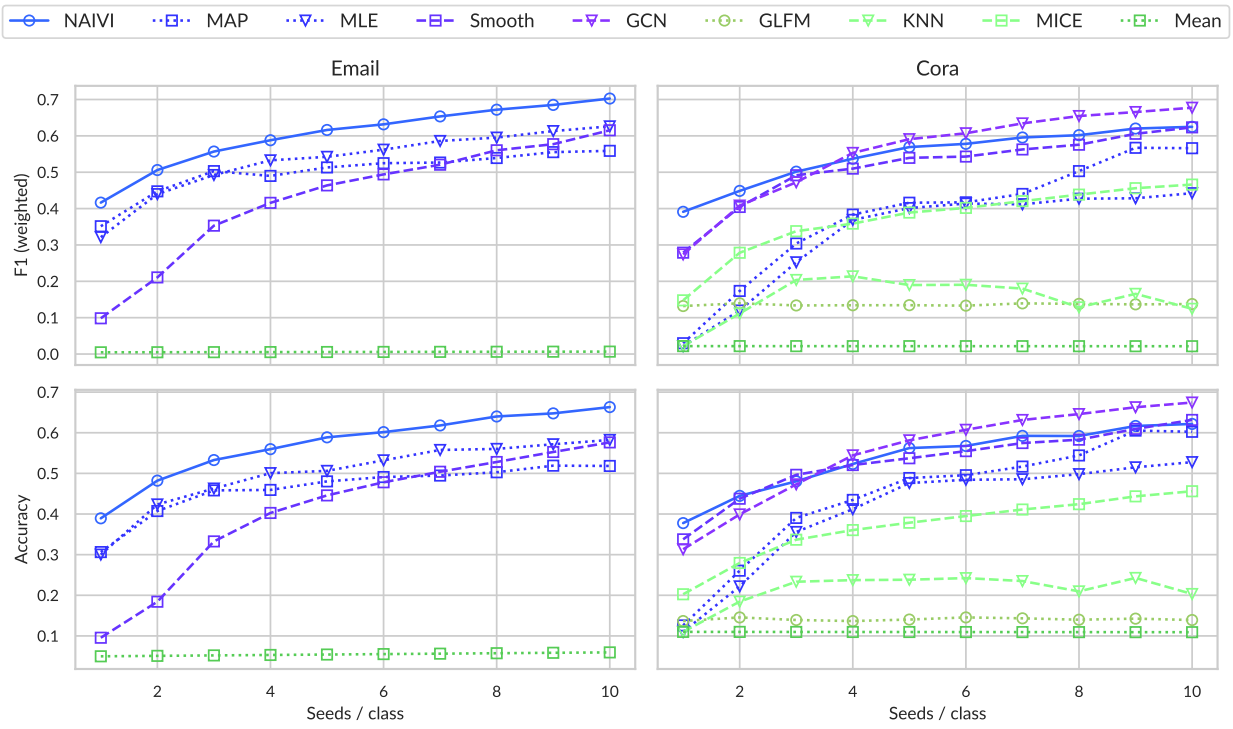

Figure 2.4: F1 score weighted by class proportion and overall classification accuracy for the task of predicting missing node labels in the Email (Yin et al., 2017) and Cora (Sen et al., 2008) datasets. "Seeds / class" refers to the number of labels per class available during training. `GCN` does not apply to the Email dataset, since there are no node attributes other than the department label.

be tricky given the two modes of data. Third, the posterior predictive distribution allows uncertainty quantification around the imputed values, which can be useful for numerical attributes.

Through multiple synthetic and real-world data examples, we showed that `NAIVI` effectively leverages all available information to produce the most accurate imputation. In particular, `NAIVI` shines when the total information is limited such as small sample size or high missing data rate (equivalently, weak supervision). Even when `NAIVI` is mis-specified or specialized methods are more appropriate, we find that our imputation method remains competitive.

The joint latent space model used to share information between the two data components defines a very flexible framework. Indeed, its modular form allows for extensions and modifications to specific cases. Weighted or signed networks can be accommodated by changing the network model. Disassortative dimensions (Rubin-Delanchy et al., 2022) can be added to capture inverse relationships between attributes and connectivity. Similarly,

other types of attributes can be easily handled by choosing different GLM models, with the main restriction being the capacity to compute the VMP messages. There are two important cases already available, but not yet implemented: categorical data using a soft-max multinomial model (Knowles and Minka, 2011) and count data using a Poisson or negative binomial model (Luts and Wand, 2015; Wand, 2017; McLean and Wand, 2019). Notably, the Email and Cora datasets studied in Section 2.4.3 are multiclass problems: we performed one-versus-rest classification, but using a multinomial approach could further improve imputation accuracy. Ordinal attributes can potentially be accommodated by modifying the multinomial model or by an approach similar to Gu and Yu (2022).

Additionally, while we focused on imputing missing node attributes, our approach works out-of-the-box for missing link prediction by simply dropping the corresponding terms from the likelihood and extracting the corresponding posterior link probability.

In Section B of the Supplementary Material, we discuss an extension to weakened missingness assumptions. In particular, we propose a simple approach where we jointly model the missingness indicators as additional binary covariates. Then, since an indicator m_{uj} and its associated attribute x_{uj} would both be dependent on the latent position $\mathbf{z}_u$, we would capture the dependency between an attribute and its status. This model relies on a specific MNAR assumption and can potentially improve prediction, as well as providing insight on the missingness patterns.

While network data are inherently computationally prohibitive because of the $\mathcal{O}(N^2)$ scaling with edges, our method still scales better than other MCMC-based methods such as those of Liu et al. (2018), Gu and Yu (2022) and Wang et al. (2023). Indeed, we can process datasets at least an order of magnitude larger (thousands rather than tens or hundreds of nodes) on a single machine without issues. As discussed in Section C of the Supplementary Material, the main bottleneck is memory consumption required by storing large amounts of messages. However, as sample size increases, numerical experiments indicate that point estimates for latent positions (MAP or MLE) perform just as well, suggesting that a less memory-intensive method, such as that of Zhang et al. (2022), is sufficient. For very large networks (N in the hundred of thousands or more), amortized variational inference might be more appropriate (Liu and Zhu, 2019).

CHAPTER 3

Locally Sparse Varying Coefficient Mixed Model with Application to Longitudinal Microbiome Differential Abundance

Abstract. Differential abundance (DA) analysis in microbiome studies has recently been used to uncover a plethora of associations between microbial composition and various health conditions. While current approaches to DA typically apply only to cross-sectional data, many studies feature a longitudinal design to better understand the underlying microbial dynamics. To study DA on longitudinal microbial studies, we introduce a novel varying coefficient mixed-effects model with local sparsity. The proposed method can identify time intervals of significant group differences while accounting for temporal dependence. Specifically, we exploit a penalized kernel smoothing approach for parameter estimation and include a random effect to account for serial correlation. In particular, it operates effectively regardless of whether sampling times are shared across subjects, accommodating irregular sampling or potentially missing observations. Simulation studies demonstrate the necessity of modelling dependence for precise estimation and support recovery. Our method's application to a longitudinal study of mice oral microbiome during cancer development revealed significant scientific insights that were otherwise not discernible through cross-sectional analyses. An `R` implementation is available at github.com/fontaine618/LSVCMM.

3.1 Introduction

3.1.1 Longitudinal differential analysis in omics data

The modern science and healthcare sectors have seen profound advancements with the emergence of omics data, such as (meta-)genomics, transcriptomics, and proteomics, among others. In particular, many studies collect omics longitudinally, which can provide researchers greater insight into the dynamical complexities underlying a myriad of biological processes. A natural statistical question that emerges from temporal omics is that of longitudinal differential analysis (LDA) where the goal is to identify biomarkers with differences between conditions across time. A naïve approach to LDA would be to perform separate cross-sectional differential analyses at a collection of time points of interests. However, this approach cannot borrow strength over time, as two neighboring time points would be treated independently. Additionally, when samples are collected repeatedly on a set of subject, cross-sectional analysis ignores serial correlation, which can greatly impair statistical properties. Finally, cross-sectional methods often struggle with irregularly-sampled time points as the number of samples at a given time point might be small, thus requiring preprocessing. Recent years have seen multiple methods proposed for LDA to account for some of these challenges (Staicu et al., 2015; Luo et al., 2017; Metwally et al., 2022), though they are limited to simple group comparisons, whereas more complicated designs, such as interactions or confounder adjustment, are often of interest.

Our motivating application involves the field of *microbiomics*, where differential analysis usually takes the name of *differential abundance analysis* (DAA), referring to the *abundance* of various organisms in a system of interest. Specifically, we are interested in the microbial composition in some tissue and its temporal association with a set of conditions. The human microbiota is known to play a key role in a wide array of diseases and health conditions (Gomaa, 2020; Ogunrinola et al., 2020), and identifying taxa that differ in their abundance between conditions can lead to improved diagnosis, prevention and treatment. For the task of longitudinal DAA, few specialized methodologies have been proposed (Paulson et al., 2017; Shields-Cutler et al., 2018; Metwally et al., 2018; Jeganathan et al., 2018) and none satisfactorily addresses the particularities of LDA while being flexible enough for general regression designs.

3.1.2 Oral cancer development mouse study

The tumor suppressor gene DMBT1 (deleted in malignant brain tumors 1) has a critical role in the progression of oral squamous cell carcinoma (SCC) (Singh et al., 2021). DMBT1 is

also present in human saliva, where it has an anti-microbial role (Reichhardt et al., 2017; Ligtenberg et al., 2001). In a recent longitudinal study, Medeiros et al. (2023) observed that DMBT1 is suppressed in saliva from patients with SCC prior to treatment and is upregulated after treatment. Consistent with these findings, mouse saliva showed a decrease in DMBT1 expression after induction of SCC, suggesting that SCC downregulates DMBT1 expression. Additionally, pre- and post-treatment changes in DMBT1 levels in saliva were associated with changes in some bacterial populations. Moreover, there were pre-treatment and post-treatment differences in salivary bacteria in patients who did or did not respond to chemoradiation treatment.

To better understand the interaction between DMBT1, microbial composition and oral SCC development, a mouse study was conducted (Medeiros et al, manuscript in preparation). Seventy-six (76) mice were bred with (*wild type*, WT) and without (*knockout*, KO) the DMBT1 gene before being inoculated with oral SCC. Then, saliva samples were collected across time (0, 4, 8, 12, 16 and 22 weeks after inoculation) and 16S sequencing was performed (16S rRNA, 97% sequence similarity OTU binning). Finally, histopathology of mouse tongues was assessed at week 22 where mice were diagnosed with pre-cancer *epithelial dysplasia* (ED) or *carcinoma in situ* (CIS), or SCC. The results strengthen the original findings of Singh et al. (2021) as 17 of the 34 (50%) knockout mice and only 6 of the 42 (14%) wild type mice developed SCC by week 22, suggesting a *causal* link between DMBT1 and cancer progression.

A potential avenue of action of DMBT1 on cancer progression, as suggested by the findings in Medeiros et al. (2023), is through the microbiota. Therefore, one of the goals of the study is to investigate the longitudinal association of microbial composition with the DMBT1 genotype (WT vs KO) and with diagnosis (dichotomized as pre-cancer, ED/CIS, vs cancer, SCC) as well as the interaction between genotype and diagnosis. In particular, identification of specific OTUs and weeks with differential abundance between any of the sub-groups may uncover how DMBT1 and microbial composition influence cancer progression, potentially leading to improved treatment response prediction and individualized treatments. To this end, we consider the following *varying coefficient model* (VCM) for the *centered log-ratio* (CLR) transformed abundance of each OTU:

$$\text{CLR}(\text{Week}) \sim \text{Genotype}(\text{Week}) \times \text{Diagnosis}(\text{Week}) + \text{Sex}(\text{Week}). \tag{3.1}$$

In particular, we consider five varying coefficient terms: an intercept (corresponding to the reference group of WT, ED/CIS and male), a main effect of the genotype (corresponding to the KO-WT difference), a main effect of diagnosis (corresponding to the SCC-ED/CIS

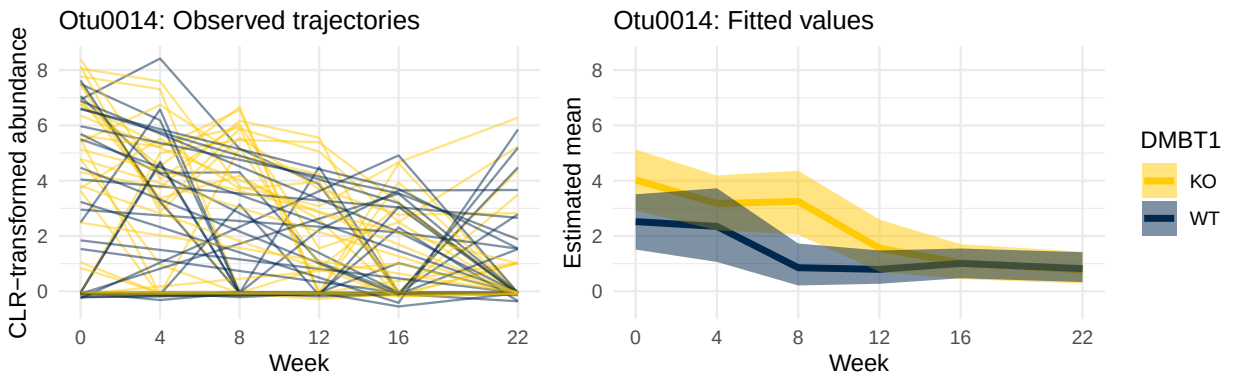

Figure 3.1: Example data and estimates from the DMBT1 OSCC mice study. (Left) CLR-transformed abundance of OTU 0014 (*Staphylococcaceae* family) over time for the 65 mice. (Right) Fitted mean with simultaneous 95% confidence band by genotype obtained from the `LSVCMM` method for the model CLR(Week) $\sim$ Genotype(Week).

difference), an interaction term between genotype and diagnosis (1 if KO and SCC; 0 otherwise), and a main effect of sex (corresponding to the F-M difference). We adjust for sex in the model as it is suspected to be associated with microbial composition, but it is of lesser interest. Figure 3.1 shows an example OTU with differential abundance between genotypes at week 8 obtained from our proposed `LSVCMM` method for the simpler model including the longitudinal effect of genotype only.

An important challenge that emerges from the data collection is a significant amount of missing data. Indeed, not all saliva samples were collected and sequenced. Saliva samples were collected for only 65 mice, and, out of the $6 \times 65 = 390$ potential samples, only 294 (75%) were ultimately sequenced. Figure 3.2 shows the number of samples available per week and per sub-group, along the patterns of missingness. About half (35/65) of the mice were sequenced at all six time points, while most of the remainder (28/65) are missing weeks 4, 8 and 12. Additionally, missingness differs with conditions: specifically, mice with slower progression (ED/CIS) are missing samples more frequently.

3.1.3 Proposed methodology

There are several desired properties for estimating the VCM (3.1). First, the estimated VCs need to be smooth over time as there is no expectation of any discontinuous jumps; suggesting approaches such as splines or kernel smoothing. Second, the longitudinal nature of the data implies serial correlation within mouse that needs to be accounted for. Third, the missing data requires a methodology that does not require all subjects to be measured on a common set of time points. Then, to identify differential abundance between any sub-groups

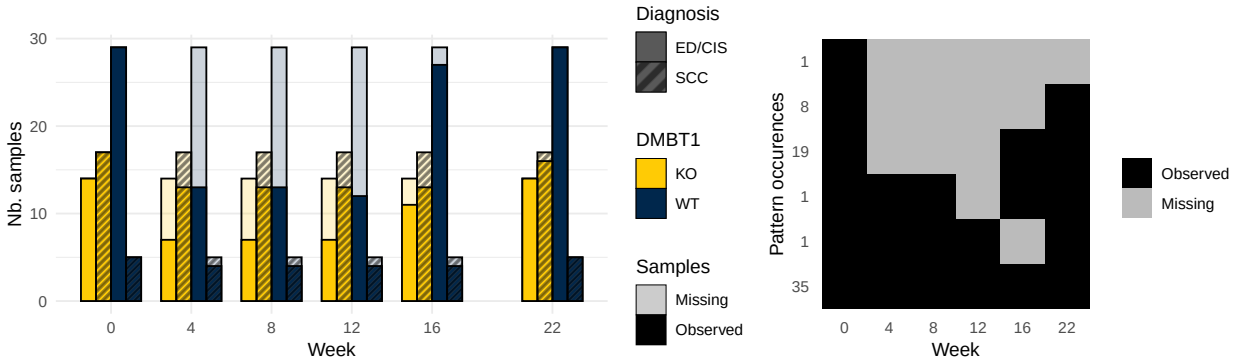

Figure 3.2: Distribution of missing data in the DMBT1 OSCC mice study. (Left) Number of observed and missing samples per week, stratified by genotype and diagnosis. (Right) Patterns of missingness; row labels indicate the frequency of that pattern occurring among the 65 mice.

at any week, we opt for a regularization approach where we need to estimate exact zeroes locally rather than globally (Wang and Kai, 2015); hence, estimation will require a penalty that induces *local sparsity*, and perhaps *global sparsity* to encourage terms in (3.1) to be zero on the whole domain.

We found no methodology that adequately answers all the above characteristics. Many early work for regularized VCM only allow global sparsity with the goal of selecting variables, not time points (Wang et al., 2008; Wang and Xia, 2009; Noh and Park, 2010; Lee and Mammen, 2016; Xue and Qu, 2012; Daye et al., 2012). The idea of local sparsity was first introduced by Wang and Kai (2015) and Kong et al. (2015) using B-splines and kernel smoothing, respectively. When using B-splines, local sparsity can be achieved by penalizing groups of consecutive spline weights, and some sort of overlapping group penalty must be used. In Wang and Kai (2015), a group bridge penalty is used to induce local sparsity in a single varying intercept; Tu et al. (2020) extended the approach to more general VCMs. Zhong et al. (2022) consider instead an application of the functional SCAD penalty of Lin et al. (2017) and extend the methodology beyond least squares and to asynchronous covariates. When using kernel smoothing, local sparsity is simpler to be achieved, as the evaluation of the VCs can be directly penalized. In Kong et al. (2015), which considers a local linear approximation, a SCAD penalty is applied to a combination of the degree 0 and degree 1 parameters of the local linear model. Still, none of the aforementioned methods include dependency in their estimation procedure. Wang et al. (2022) consider a B-spline approach with group bridge penalty on the spline weights and adjust for within-subject dependency using a two-step estimator. However, the estimation of the serial correlation requires all subjects to be sampled on a common set of time points, which is not the case for our data.

From a different angle, there exists a few methods performing longitudinal differential abundance analysis, which is ultimately the goal of the present analysis. A first set of methods use B-splines and an area ratio statistic permutation test to find intervals of group differences across conditions (Paulson et al., 2017; Luo et al., 2017; Metwally et al., 2018, 2022). Shields-Cutler et al. (2018) propose a LOESS approach on the relative abundances, where permutations are used to identify time points of DA across groups. Again, the previous methods do not account for within-subject dependency. Staicu et al. (2015) propose a global test for equality of means across groups, accounting for longitudinal effects and based on Fourier expansions. While we could recast the VCM (3.1), excluding the Sex term, as a four group testing problem, we would lose the structure induced by the main effects.

We thus propose a novel approach for local sparsity in VCMs, accounting for within-subject dependency, called `LSVCMM` (for *locally sparse varying coefficient mixed model*). Specifically, we consider locally-constant kernel smoothing for a VCM with parametric working covariance and obtain local and global sparsity through an (adaptive) sparse group Lasso (Friedman et al., 2010; Simon et al., 2013). Estimation alternates between mean parameter updates and covariance parameter updates. The mean parameter estimates are taken as the solution to penalized score equations leading to proximal operators; the variance parameter estimates optimize a Gaussian quasi-likelihood. An extended Bayesian information criterion is proposed to perform tuning parameter selection, and simultaneous confidence bands are obtained through a bootstrap procedure. An `R` implementation is provided in the `LSMCMM` package available at `github.com/fontaine618/LSVCMM`. In Section 3.3, we conduct extensive simulation studies showing that `LSVCMM` improves estimation accuracy and support recovery when compared to methods lacking longitudinal dependence adjustment or smoothness. Additionally, we compare `LSVCMM` to `SPFDA` (Wang et al., 2022), which requires imputation, and show significant improvements. In Section 3.4, we apply `LSVCMM` to VCM (3.1) discussed in Section 3.1.2. In particular, we identify five candidate OTUs for which their temporal trajectories vary with both the DMBT1 genotype and the SCC diagnosis at week 22. These findings suggest pathways of action between the DMBT1 protein and cancer development through the microbial composition, which are found to be plausible based on related literature.

3.2 Methods

3.2.1 Setting & notation

We consider the following function-on-scalar regression problem. Let $i = 1, \ldots, N$ denote the N sampling units (e.g., subjects). Let t_{in}, $n = 1, \ldots, N_i$, denote the sampling times for subject i and define $\mathbf{t}_i = (t_{i1}, \ldots, t_{iN_i})$; we do not assume any structure on the $\mathbf{t}_i$'s across subjects. The observed response for subject i at time t_{in} is denoted $y_{in} = y_i(t_{in}) \in \mathbb{R}$ and we define $\mathbf{y}_i = y_i(\mathbf{t}_i) = (y_{i1}, \ldots, y_{in_i}) \in \mathbb{R}^{n_i}$ as the vector of responses for subject i. For each subject, we split covariates into two categories: those associated with time-varying effects, $\mathbf{x}_{in} = \mathbf{x}_i(t_{in}) \in \mathbb{R}^{p_x}$, and those with constant effects, $\mathbf{u}_i \in \mathbb{R}^{p_u}$.

In the present exposition, we assume the covariates for the varying coefficient terms to be constant through time, indicated by the absence of time index in $\mathbf{x}_i \equiv \mathbf{x}_i(\cdot)$, but our proposed model and implementation readily works for $\mathbf{x}_{in} \in \mathbb{R}^{p_x}$ varying with time, provided it is observed at the same time points as the responses of subject i. In particular, we do not allow *asynchronous* covariates (see, e.g., Zhong et al., 2022, for a related method).

3.2.2 Varying coefficient mixed model

Our main goal is to study the relationship between covariates $\mathbf{x}_i$ and the functional response $y_i(\cdot)$, while accounting for temporal dependence within subjects and other covariates. In particular, we are interested in identifying *if*, *when* and *how* $y_i(\cdot)$ changes with each entry in $\mathbf{x}_i$. To this end, we consider a *(semi-)varying coefficient mixed model*:

$$\mathbb{E}\{y_{in} \mid \theta_i(t_{in})\} = \boldsymbol{\beta}(t_{in})^\top \mathbf{x}_i + \boldsymbol{\alpha}^\top \mathbf{u}_i + \theta_i(t_{in}) \tag{3.2}$$

$$\mathrm{Var}(y_{in} \mid \theta_i(t_{in})) = \sigma^2 \tag{3.3}$$

with (conditional) independence across i and n, where $\boldsymbol{\beta}(\cdot) : \mathbb{R} \to \mathbb{R}^{p_x}$ is the vector-valued function of time-varying coefficients, $\boldsymbol{\alpha} \in \mathbb{R}^{p_u}$ is the vector of constant linear effects, and where $\theta_i(\cdot)$ is a random process capturing the temporal dependence. In particular, we assume $\mathbb{E}\{\theta_i(t)\} \equiv 0$ with covariance kernel $\mathrm{Cov}(\theta_i(t), \theta_i(t')) = \sigma^2 k_\theta(t, t')$ for some symmetric positive definite kernel k_θ. Define $\mathbf{K}_\theta(\mathbf{t})$ as the (unscaled) covariance matrix for a random process evaluated at the time points in $\mathbf{t}$, that is, $[\mathbf{K}_\theta(\mathbf{t})]_{nn'} = k_\theta(t_n, t_{n'})$. Hence, marginally,

$$\mathbb{E}\{\mathbf{y}_i\} = \mathbf{m}_i := \boldsymbol{\beta}(\mathbf{t}_i)^\top \mathbf{x}_i + \boldsymbol{\alpha}^\top \mathbf{u}_i \mathbf{1}_{N_i}, \qquad \mathrm{Var}(\mathbf{y}_i) = \mathbf{V}_i := \sigma^2 \left(\mathbf{K}_\theta(\mathbf{t}_i) + \mathbf{I}_{N_i}\right), \tag{3.4}$$

where $\boldsymbol{\beta}(\mathbf{t}_i)$ is the $p_x \times N_i$ matrix with columns $\boldsymbol{\beta}(t_{in})$. Given S time points of interest $\mathbf{t} = (t^{(1)}, \ldots, t^{(S)})$, we are interested in the value of $\boldsymbol{\beta}(\cdot)$ at each of those time points. For

example, $\mathbf{t}$ could consists of all observed time points or a regular grid over the observed domain. We define $\mathbf{B}$ as the $p_x \times S$ matrix with entries $b_j^{(s)} = \boldsymbol{\beta}_j(t^{(s)})$, with rows $\mathbf{b}_j = \beta_j(\mathbf{t})$ and with columns $\mathbf{b}^{(s)} = \boldsymbol{\beta}(t^{(s)})$.

Denote the precision matrix $\mathbf{P}_i = \mathbf{V}_i^{-1}$ where $\mathbf{V}_i = \sigma^2 \left(\mathbf{K}_\theta(\mathbf{t}_i) + \mathbf{I}_{N_i}\right)$. To obtain the most efficient estimator of the mean parameters, we need $\mathbf{V}_i$ to accurately capture the dependence structure in the residuals $\mathbf{r}_i = \mathbf{y}_i - \mathbf{m}_i$. For a regular design, i.e, $\mathbf{t}_i \equiv \mathbf{t}$ for some $\mathbf{t}$, and given N sufficiently large, we could simply estimate $\mathbf{V} \equiv \mathbf{V}_i \approx \frac{1}{N}\sum_{i=1}^N \mathbf{r}_i\mathbf{r}_i^\top$, perhaps with a prior smoothing step (Wang et al., 2022). For irregular designs, more care is required as empirical covariances are not applicable.

As observed by Fan et al. (2007), efficiency gains can be obtained even though the working covariance does not exactly match the true covariance. In particular, even a rough optimization of the working covariance can lead to near-optimal estimation efficiency. We propose to specify a working parametric model whose covariance function is determined by a few parameters. Some notable examples include the compound symmetry structure, equivalent to a random intercept model, with covariance function $k_\theta(t, s; r_\theta) = r_\theta$, and the AR(1) model, with covariance function $k_\theta(t, s; r_\theta, \rho) = r_\theta \rho^{|t-s|}$, where r_θ denotes the variance ratio with the noise variance σ^2 and where ρ controls the long-range dependency. Given a working covariance model, we denote by $\boldsymbol{\tau}$ the set of all covariance parameters.

3.2.3 Local and global sparsity

A convenient feature of local regression is that the value of $\boldsymbol{\beta}(\cdot)$ at a specific time $t^{(s)}$ is directly parameterized by $\mathbf{b}^{(s)}$. This is in contrast to spline basis expansions, where the value of $\boldsymbol{\beta}(t^{(s)})$ is a linear combination of basis functions active at time $t^{(s)}$. Hence, to find $\beta_j(t^{(s)}) = 0$, we only need $b_j^{(s)} = 0$, compared to requiring a consecutive set of spline weights to be zero. This allows us to use a simple sparsity-inducing penalty on the entries of $\mathbf{B}$, in comparison to overlapping group penalties used in spline methods (Wang and Kai, 2015; Tu et al., 2020; Wang et al., 2022; Zhong et al., 2022).

We thus propose to encourage local sparsity by including a Lasso penalty (Tibshirani, 1996) on $\mathbf{B}$, namely, $\lambda \sum_{j=1}^{p_x}\sum_{s=1}^{S} \omega_j^{(s)} |b_j^{(s)}|$, where $\omega_j^{(s)}$ are weights and where $\lambda \geqslant 0$ is the regularization parameter. For example, in group comparison setting, we would have two covariates: an intercept $x_{i1} = 1$ and a group membership $x_{i2} \in \{0, 1\}$. Then we are only interested in encouraging zeros for the group difference effect $\mathbf{b}_2$; in that case, the intercept $\mathbf{b}_1$ would not be penalized, which can be achieved by setting $\omega_0^{(s)} \equiv 0$.

Furthermore, we may be interested in encouraging $\beta_j(\cdot) \equiv 0$ altogether to identify if the jth covariate has any association with the response. This suggests to add a group lasso

penalty on the whole vector $\mathbf{b}_j$, leading to the sparse group Lasso penalty (Friedman et al., 2010; Simon et al., 2013):

$$\mathcal{P}_{\lambda,\alpha}(\mathbf{B};\Omega) := \lambda \sum_{j=1}^{p_x} \left[(1-\alpha)\sqrt{S}\omega_j \|\mathbf{b}_j\|_2 + \alpha \sum_{s=1}^{S} \omega_j^{(s)} |b_j^{(s)}| \right] \tag{3.5}$$

where $\alpha \in [0,1]$ is a tuning parameter balancing between encouraging global sparsity ($\alpha = 0$) and local sparsity ($\alpha = 1$), where ω_j are weights for the group lasso penalty, and where Ω contains all weights.

The Lasso and group Lasso penalty are famously known for their estimation bias due to the shrinkage applied to non-zero values. To alleviate this issue, we propose to use adaptive penalties (Zou, 2006; Nardi and Rinaldo, 2008; Poignard, 2020), where the weights are set to $\omega_j = \|\hat{\mathbf{b}}_j^{\text{mle}}\|_2^{-\gamma}$ and $\omega_j^{(s)} = |\hat{b}_j^{(s),\text{mle}}|^{-\gamma}$ for some $\gamma > 0$, where $\widehat{\mathbf{B}}^{\text{mle}}$ contains the coefficients estimated without penalty.

3.2.4 Estimation

We alternate between mean parameter ($\mathbf{B}$ and $\boldsymbol{\alpha}$) updates and variance parameter updates by holding the other fixed. We find that there is generally very little change beyond the first cycle and that a single variance update is often sufficient (similar to the two-step estimator of Wang et al., 2022).

3.2.4.1 Estimating equations

Let $\mathbf{r}_i = \mathbf{y}_i - \mathbf{m}_i$ denote the residual vector for subject i, which implicitly depends on the mean parameters. Further denote $\mathbf{X}_i$ the $n_i \times p_x$ matrix with rows $\mathbf{x}_i(t_{ij})$; for fixed covariates, we simply have $\mathbf{X}_i = \mathbf{1}_{n_i}\mathbf{x}_i^\top$. We define the estimating function for $\boldsymbol{\beta}(t)$ by weighing the residuals using a kernel function $k_h(s) = k(s/h)/h$ depending on the distance from a time point of interest t:

$$U_{\boldsymbol{\beta}(t)} := -\sum_{i=1}^{N} \mathbf{X}_i^\top \operatorname{diag}(\mathbf{k}_i(t))\mathbf{P}_i\mathbf{r}_i, \tag{3.6}$$

where $\mathbf{k}_i(t) = [k_h(t - t_{ij})]_{j=1}^{n_i}$, and where $h > 0$ is the kernel scale. We employ kernel smoothing rather than spline smoothing as inducing zeros through penalization (Section 3.2.4.2) is simpler.

To motivate our GEE-based inference methodology, we start by investigating the score

equations for the mean parameters under a Gaussian model with likelihood

$$\ell_i(\boldsymbol{\beta}(\cdot), \boldsymbol{\alpha}) = -\frac{1}{2}\log\det(2\pi\mathbf{V}_i) - \frac{1}{2}[\mathbf{y}_i - \mathbf{m}_i]^\top \mathbf{P}_i[\mathbf{y}_i - \mathbf{m}_i], \tag{3.7}$$

with $\ell(\boldsymbol{\beta}(\cdot), \boldsymbol{\alpha}) = \sum_{i=1}^N \ell_i(\boldsymbol{\beta}(\cdot), \boldsymbol{\alpha}))$. Consider computing the gradient with respect to $\boldsymbol{\beta}(t)$ for some t. Whenever $t_{ij} \neq t$, the mean m_{ij} does not depend on $\boldsymbol{\beta}(t)$, so we find

$$\nabla_{\boldsymbol{\beta}(t)}\ell(\boldsymbol{\beta}(\cdot), \boldsymbol{\alpha}) = -\sum_{i=1}^N \mathbf{X}_i^\top \mathbf{D}_i(t)\mathbf{P}_i\mathbf{r}_i, \tag{3.8}$$

where $\mathbf{D}_i(t) = \mathrm{diag}(\mathbb{1}[t = t_{ij}])$. Now, consider computing the gradient with respect to $\boldsymbol{\beta}$ by assuming that $\boldsymbol{\beta}(\cdot)$ is a constant function parameterized by $\boldsymbol{\beta}$, i.e., $\boldsymbol{\beta}(\cdot) \equiv \boldsymbol{\beta}$. We find

$$\nabla_{\boldsymbol{\beta}}\ell(\boldsymbol{\beta}, \boldsymbol{\alpha}) = -\sum_{i=1}^N \mathbf{X}_i^\top \mathbf{P}_i\mathbf{r}_i = -\sum_{i=1}^N \mathbf{X}_i^\top \mathbf{I}\mathbf{P}_i\mathbf{r}_i. \tag{3.9}$$

Looking at the difference between (3.8) and (3.9), we see that the pointwise gradient weighs the precision-adjusted residuals $\mathbf{P}_i\mathbf{r}_i$ by $\mathbf{D}_i(t)$, while the constant gradient weighs them equally by $\mathbf{I}$. To obtain a nonconstant smooth estimate that borrows signal from neighboring time points, we utilize kernel smoothing, which interpolates between the pointwise estimator and the constant estimator. Specifically, we employ a locally-constant approximation around t (a degree 0 local polynomial approximation), that is, to estimate $\boldsymbol{\beta}(t)$, we use the working model $\boldsymbol{\beta}(t_{ij}) \equiv \boldsymbol{\beta}(t)$ and downweight observations with t_{ij} away from t. Of note, when $h \to 0$, $\mathrm{diag}(\mathbf{k}_i(t))$ behaves as $\mathbf{D}_i(t)$, in which case we recover the pointwise estimator; when $h \to \infty$, $\mathrm{diag}(\mathbf{k}_i(t))$ behaves as $\mathbf{I}$, in which case we recover the constant estimator.

3.2.4.2 Proximal updates

The penalized estimating functions are given by adding the subgradients of the penalty to the unpenalized functions, similarly to Wang et al. (2012) and Johnson et al. (2008) where a linear approximation to the penalty is rather used:

$$U_{\mathbf{b}_j} + \partial_{\mathbf{b}_j} P_{\lambda,\alpha}(\mathbf{B}; \Omega), \qquad j = 1, \ldots, p_x,$$

where $U_{\mathbf{b}_j} = (U_{b_j^{(1)}}, \ldots, U_{b_j^{(S)}}) \in \mathbb{R}^S$ is the estimating function for covariate j, and where $\partial_{\mathbf{b}_j}$ denotes the subgradient with respect to $\mathbf{b}_j$.

Rather than considering a minorization-maximization scheme to solve the penalized estimating equation $\mathbf{0} \in U_{\mathbf{b}_j} + \partial_{\mathbf{b}_j} P_{\lambda,\alpha}(\mathbf{B}; \Omega)$, we utilize the convexity of the sparse group Lasso

penalty to our advantage and proceed to proximal gradient updates. Specifically, we use the estimating function to perform a gradient step before applying the corresponding proximal operator (Parikh et al., 2014):

$$\mathbf{b}_j \leftarrow \text{prox}_{\eta \mathcal{P}_{\lambda,\alpha}(\cdot;\Omega)} \left(\mathbf{b}_j - \eta U_{\mathbf{b}_j} \right), \tag{3.10}$$

for some stepsize η. We refer to Section S1.1 of the Supplementary Material for details on the proximal update.

3.2.4.3 Estimating the covariance parameters

To estimate the variance parameters $\boldsymbol{\tau}$, we maximize the profile likelihood under a Gaussian model (equivalently, a quasi-likelihood approach, Fan and Wu, 2008) while holding the residuals $\mathbf{r}_i$ fixed:

$$\ell(\boldsymbol{\tau}) := -\frac{1}{2} \sum_{i=1}^{N} \log \det(2\pi \mathbf{V}_i) + \mathbf{r}_i^\top \mathbf{P}_i \mathbf{r}_i,$$

where $\mathbf{P}_i$ and $\mathbf{V}_i$ implicitly depend on the variance parameters. Updates for the compound symmetry covariance can be found in Section S1.2 of the Supplementary Material.

3.2.5 Additional details

We refer to the Supplementary Material for additional information and practical guidelines about `LSVCMM`. In particular, we propose an *extended Bayesian information criterion* (EBIC, Chen and Chen, 2008) for selection of the regularization parameter λ and the kernel scale h. Additionally, we utilize bootstrap sup-t simultaneous confidence bands (Montiel Olea and Plagborg-Møller, 2019) for uncertainty quantification.

3.3 Simulation studies

We consider two scenarios where the sampled time points $\mathbf{t}_i$ vary between subjects. In the first case, subjects are sampled on a common set of time points $\mathbf{t}$, but not all time points are observed for each subject. This situation, inspired from our real data application in Section 3.4, occurs in experimental studies with missing data. In the second case, subjects are each sampled at different time points so that none or few time points are shared across subjects. This situation more commonly arises with observational studies, where researchers do not control sampling.

Synthetic data is generated as follows. We consider the task of estimating temporal group differences within $N = 100$ subjects, half of which are assigned to either group. For each subject i, sampled time points $\mathbf{t}_i$ are generated according to either scenario. A random intercept $\theta_i(t) \equiv \theta_i$ is sampled from a normal distribution centered at 0 with variance $\sigma^2 r_\theta$, where r_θ denotes the variance ratio between noise and random effect. The mean function for each subject is computed as $\mu_i(t) = \beta_0(t) + \beta_1(t)x_i$, where x_i is the group indicator variable. Observations are finally generated as $y_{ij} = \mu_i(t_{ij}) + \theta_i + \sigma\varepsilon$ for $\varepsilon \sim \mathcal{N}(0, 1)$. Unless specifically varied in an experiment, default values for variance parameters are $\sigma^2 = 1$ and $r_\theta = 1$, leading to a correlation of 0.5 across time points. The true generating values for the time-varying effects $\beta_1(\cdot)$ will be defined in each sub-experiments such that it is smooth and non-zero only on part of the domain.

For both scenarios, we conduct three sub-experiments. In the first experiment, we vary the signal strength by increasing the variance σ^2 while keeping the signal fixed. In the second experiment, we investigate the effect of missing data by varying the number of sampled time points per subject. In the third experiment, we study the effect of the dependence by varying the variance ratio parameter r_θ.

We report the mean absolute estimation error (MAE) in estimating the functional group difference $\beta_1(\cdot)$ as well as the classification accuracy induced by the sparsity, both of which over a pre-specified grid of time points. Additional performance metrics are included in Section S2.1 of the Supplementary Material.

Four methods are compared. Our proposed method, `LSVCMM`, is fitted with a compound symmetry variance structure, using a Gaussian kernel with a fixed kernel scale ($h = 0.2$) and the regularization parameter is selected using the EBIC described in Section S1.3 of the Supplementary Materials. We also include `LSVCM`, which is the same as `LSVCMM`, except that an independent variance structure is used. `LSVCM` is closely related to the method of Kong et al. (2015) where a local linear approximation and a SCAD penalty is rather utilized. We further consider another simplification where the kernel smoothing is removed (`ALasso`): this amounts to independent sparse estimation at each time point, though estimation of σ^2 and tuning parameter selection is done jointly. Finally, we include `SPFDA` (Wang et al., 2022) which is a direct competitor to `LSVCMM` as it includes smoothing, dependence and local sparsity, but it requires sampling times to be shared. To apply `SPFDA`, we impute the missing samples using functional PCA (fPCA, Goldsmith et al., 2013, using the `fpca.sc` function of the `refund` package, Goldsmith et al., 2023). In some cases, fPCA fails due to limited number of samples per subject; we proceed to mean imputation instead for those instances. The bridge penalty parameter is set to $\alpha = 0.5$ and the regularization parameter is selected using their propsed EBIC.

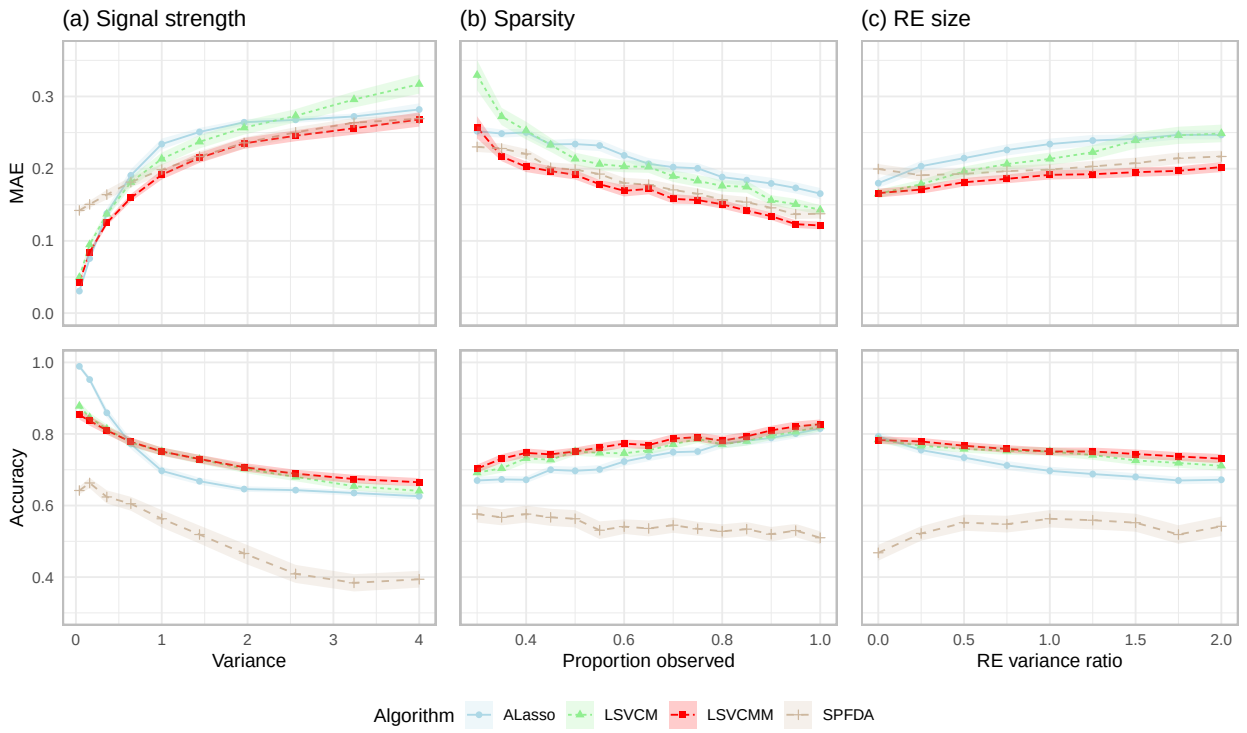

Figure 3.3: Evaluation metrics in the missing data scenario: mean (line) and standard error (band) across 100 replications.

3.3.1 Missing data in regular design

In this scenario, we consider 10 time points regularly-spaced on the unit interval. To introduce missing data, we randomly select 71% of the time points and 71% of the subjects and set the intersection as missing, leading to, on average, 50% of the 10×100 samples to be missing. With this procedure, only 29% of subjects will have 10 samples while the other 71% will only have around three. Symmetrically, three time points will be observed for all 100 subjects, while the remaining seven will only be observed for 29 subjects. The intercept function is chosen to be the zero function and the group difference function is set to be $\sin(2\pi(t - \frac{1}{4})) \vee 0$ so that only the middle four time points are non-zero. `SPFDA` is fitted using 12 spline functions. Evaluation metrics, aggregated over the 10 time points, can be found in Figure 3.3.

Comparing `LSVCMM` to its independent counterpart, `LSVCM`, we find that the largest difference appears in the estimation error, especially for weak signal (large variance or large random effect). The cross-sectional method, `ALasso`, generally does worse in estimation and accuracy, except in very strong signal regimes, where it outperforms all others in accuracy. `SPFDA` performs similarly to `LSVCMM` in terms of estimation, apart from strong signal regimes,

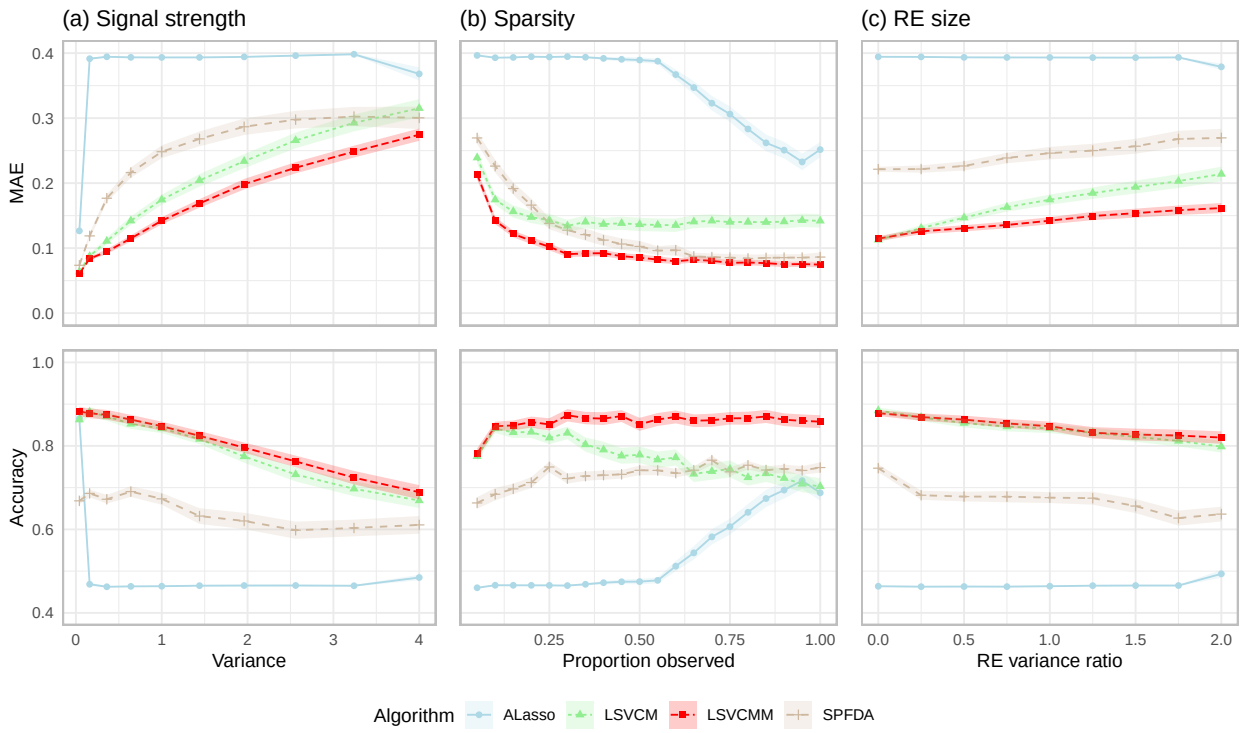

Figure 3.4: Evaluation metrics in the irregular sampling scenario: mean (line) and standard error (band) across 100 replications.

but does much worse in terms of support recovery. Additional metrics, featured in Figure S2 of the Supplementary Material show that `SPFDA` typically selects more time points as differential, leading to largely inflated FDR and slightly better power. In particular, as the proportion of missing data decreases, we would expect `SPFDA` to become comparable to `LSVCMM`, but accuracy actually decreases, interestingly.

3.3.2 Irregular sampling

In this scenario, we mimic uniform sampling by choosing 100 regularly-spaced time points over the unit interval and uniformly draw 10 of those as observed for each subject. Then, each of the 100 time points is observed, on average, only 10 times. The intercept function is again chosen to be the zero function and the group difference function as $\text{sigmoid}(20(0.6-t))\mathbb{1}[t < 0.45]$, such that the first 45 time points are null. `SPFDA` is fitted using 50 splines. Evaluation metrics, aggregated over the 100 time points, can be found in Figure 3.4.

Cross-sectional methods have a much harder time with this setting since there are only a few observations at each time points: indeed, `ALasso` requires over 50 samples per time point before starting to estimate non-zero differences. In terms of dropping longitudinal effects

(LSVCM), we observe the same pattern as in the missing data scenario, where weak signal and strong dependency produce worse performance. Additionally, we notice a much larger drop in accuracy as the sampling time become dense, where longitudinal effects are more noticeable. This setting is also more difficult for SPFDA as it will rely more heavily on the imputed observations: the transition from sparsely-observed data to densely-observed data makes this abundantly clear as the estimation error becomes comparable.

3.4 Application: oral cancer development mouse study

We apply the LSVCMM methodology to VCM (3.1), introduced in Section 3.1.2, regressing microbial abundance (CLR-transformed OTU abundance) on genotype (WT or KO), diagnosis (ED/CIS or SCC) and sex (M or F). After filtering out OTUs below 5% prevalence across the 294 samples, there remains 187 OTUs to be used as the response.

We fit LSVCMM with a compound symmetry working covariance, with a Gaussian kernel with fixed scale $h = 0.2$, with a mixed penalty ($\alpha = 0.5$) to encourage global sparsity in each of the terms as well as local sparsity to identify weeks of differential abundance. The intercept varying coefficient is not penalized since there is no expectation it should be close to 0. The regularization parameter λ is selected using the EBIC proposed in Section S1.3 of the Supplementary Material. We compare LSVCMM to ALasso (i.e., cross-sectional with sparsity), and to SPFDA (with fPCA imputation of the missing entires). We omit LSVCM from the comparison since, as was seen from the simulation studies in Section 3.3, the main difference is in estimation error, not selection accuracy. For LSVCMM and ALasso, we use bootstrap to obtain simultaneous confidence bands for all varying coefficients of interests. SPFDA only provides point-wise standard errors: we produce a simultaneous band using the provided standard error and a Bonferroni adjustment, corresponding to multiplying the standard error by a factor of 2.64, the upper $1 - 0.025/6$ upper quantile of a standard normal distribution. For each of the two main effects of genotype and diagnosis and for their interaction terms, we report the estimated varying coefficients and note which weeks are such that zero was excluded. For brevity, only taxa identified as DA for at least one week by one method are reported, though 187 OTUs were processed.

Figure 3.5 contains the estimated varying coefficients for the main effects of genotype and diagnosis as well as for the interaction term. We generally find strong agreement between LSVCMM and its cross-sectional counter-part ALasso and some agreement between LSVCMM and SPFDA. ALasso tends to select more differences than LSVCMM, but many of the additional discoveries occur for the weeks 4-16 where there are significantly more missing data. This is particularly obvious for the interaction term, where ALasso selects many OTUs at week

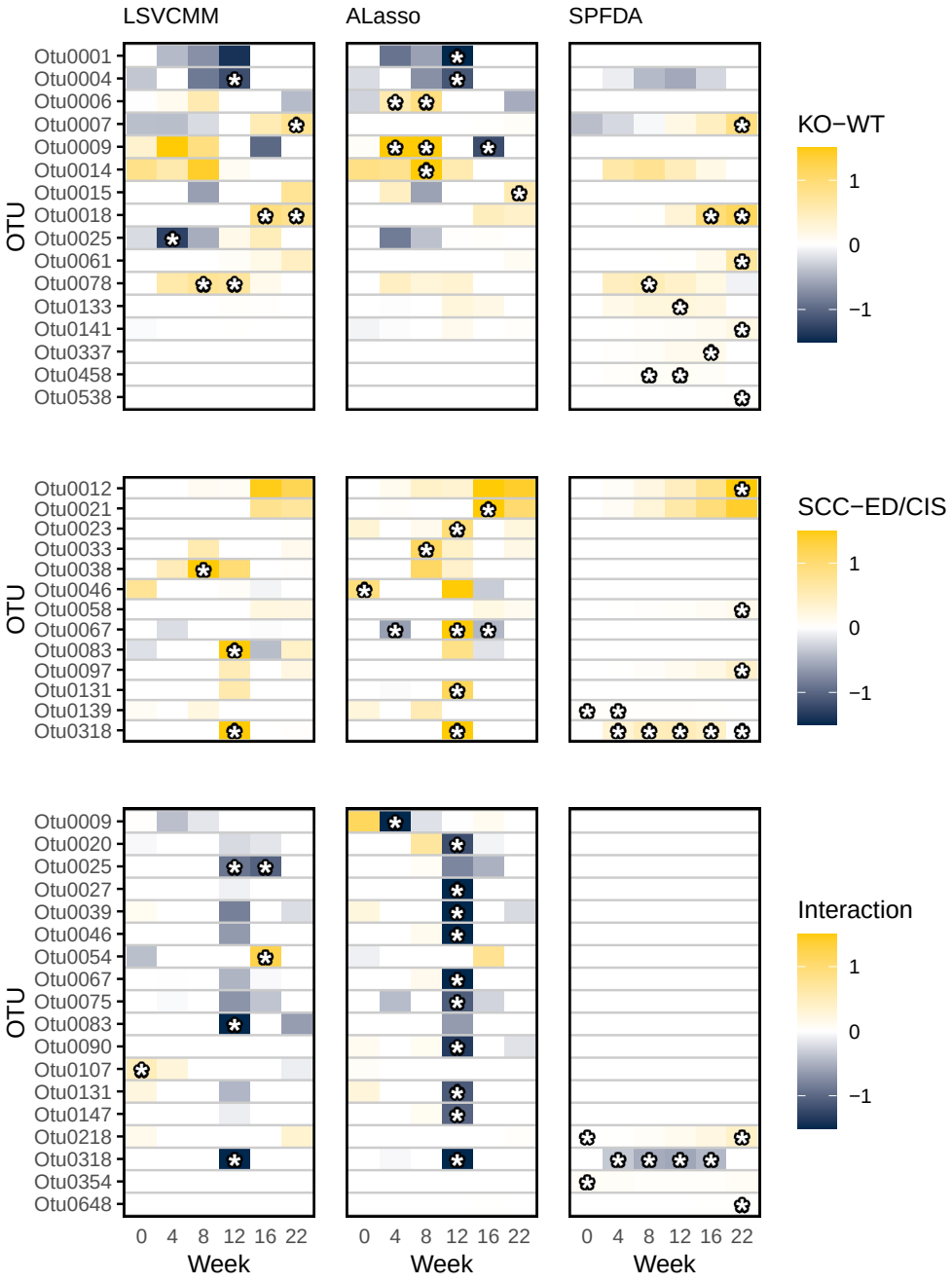

Figure 3.5: Comparison of estimates for the DMBT1 OSCC mice study by various methods: estimated main effects (top: KO less WT; middle: SCC less ED/CIS) and interaction (bottom: coded as 1 if KO and SCC, and 0 otherwise). White cells correspond to a zero estimate, colored cells correspond to a non-zero estimate. Asterisks indicate a time point where the 95% simultaneous confidence band excludes zero. Columns represent estimate emerging from three different methods. Only OTUs with a significant difference for at least one time point and one method are included in each row (out of 187 total OTUs).

12. We note that `SPFDA` finds multiple small group differences among rare taxa (large OTU number) which are not corroborated by `LSVCMM` nor `ALasso`, suggesting that the standard error estimation may be inappropriate for those instances.

Five OTUs are identified by `LSVCMM` as having a non-zero interaction term, suggesting some interplay between the DMBT1 gene, the abundance of those OTUs and cancer development. In particular, OTU 0107 (*Lachnospiraceae* family) identifies a positive difference in the interaction term at week 0 and no differences in the main effects, suggesting that the KO mice with initially higher abundance of that OTU were more likely to develop SCC by week 22. Three OTUs, 0025, 0083 and 0318, have a negative interaction estimate at week 12. OTU 0025 (*Porphyromonadaceae* family) does not have a non-zero main effect estimate, suggesting that KO mice with lower abundance of that OTU between week 12 and 16 were more likely to develop SCC by week 22. OTUs 0083 and 0318 (both *Lachnospiraceae* family) also show a positive difference between diagnosis, indicating that WT mice with higher abundance of those two OTUs at week 12 were more likely to be diagnosed with SCC by week 22. Finally, OTU 0054 (*Sphingomonas* genus) has a positive interaction estimate at week 16 and no non-zero main effect estimates, suggesting that KO mice with higher abundance of that OTU at week 16 were more likely to be diagnosed with SCC.

The OTUs for which an interaction between DMBT1 expression and SCC development is found are plausible candidates. The *Porphyromonadaceae* family contains a notable species, namely *P. gingivalis*, which is a "biomarker for microbe-associated risk of death due to orodigestive cancer" (Ahn et al., 2012). In particular, *P. gingivalis* "relates to [OSCC] even in absence of periodontitis" (Olsen and Yilmaz, 2019) and is associated with lower treatment response rate and lower overall survival in esophageal SCC (Gao et al., 2021). The *Lachnospiraceae* family is also known to intervene in other digestive tract cancers. For example, *Lachnospiraceae* is hypothesized to play a "protective role for certain microbiota types against [colorectal cancer]" (Flemer et al., 2018). Research also suggest that the *Sphingomonas* genus is depleted in breast (Xuan et al., 2014) and gastric (Dong et al., 2019) cancers, but enriched in bladder (Liu et al., 2019) and brain (Higuchi et al., 2021) cancers. In addition to OTUs 0083 and 0318, the main effect estimates find that OTU 0038 (*Bifidobacterium* genus) is enriched at week 8 among mice who developed SCC by week 22. The *Bifidobacterium* genus contains species that are already proposed as probiotic treatment in colorectal cancers (Yoon et al., 2021).

3.5 Discussion

`LSVCMM` addresses two main deficiencies in existing VCM methodologies. First, it includes longitudinal effects in the form of a parametric working covariance model, whereas many methods cannot account for within-subject dependencies (Wang and Kai, 2015; Kong et al., 2015; Tu et al., 2020; Zhong et al., 2022). Simulation experiments in Section 3.3 have shown that omitting the dependency leads to worse estimation accuracy and worse support recovery in the densely sampled regime. Second, our proposed approach applies to any sampling design: in particular, it allows missing data in regular designs as well as irregular designs, where the only method including longitudinal effects (Wang et al., 2022, `SPFDA`) is limited to regular designs. Experiments show that imputation is insufficient in extending `SPFDA` to irregular cases, as support recovery is inferior to `LSVCMM`.

There are multiple avenues of improvement, not only in terms of general enhancements, but also with respect to the longitudinal differential abundance application of Section 3.4. First, the bootstrap simultaneous confidence bands are expensive to compute: indeed, in order to have a granular enough estimate of the quantiles, thousands of samples are required. A natural alternative would be to consider ideas from the *de-biased Lasso* literature (Zhang and Zhang, 2014; Honda, 2021). Second, while computationally convenient, the working covariance model requires some assumption on the dependency structure which could lead to misspecification. In Section S2.2 of the Supplementary Materials, we consider an experiment where the true generating covariance is AR(1), while a compound symmetry structure is fitted. We found that `LSVCMM` still improves on the independent model, even when misspecified. That being said, more severe misspecification could be troublesome and an alternative treatment of the dependency might be more robust. Third, the locally-constant kernel smoothing was similarly chosen for the ease of calculation, but it introduces some bias at the boundary of the domain and in regions or sharp variations: another important extension involves higher-order approximations such as the commonly-used local linear approximation. Fourth, utilizing a least square objective for the log-transformed microbial abundances is generally inappropriate, largely due to the zero-inflation occurring for rarer taxa. For example, 78 of the 187 OTUs considered have more than 90% of zeros across the 265 samples and 173 have more than 50% of zeros. A natural extension of `LSVCMM` would be to allow more general distribution, similarly to GLMs; in particular, a negative binomial or Tweedie compound Poisson-Gamma objective would be of interest for our application. Of note, Zhong et al. (2022) is defined for *generalized* VCM, though it lacks longitudinal effects. Fifth, we processed each OTU independently, and there are two important sources of dependency among them that are thus disregarded. The sequencing procedure introduces

negative dependence between OTUs because of compositionality effects, and OTUs corresponding to related species may have positive dependency which could be captured by the taxonomic tree. Both of these suggest a multivariate extension.

CHAPTER 4

Dynamic Latent Factor Regression for EEG-Based Brain-Computer Interfaces

Abstract. Brain-computer interfaces (BCI) provide direct communication from the brain to an external device by inferring the user's intent from the brain activity. Specifically, current BCI technology often rely on analysing event-related potentials (ERP), such as the P300 signal, as measurable electroencephalography (EEG) responses to stimuli presented to the user. However, crucial information is often overlooked by focusing solely on well-characterized ERPs. In particular, studying the whole EEG measurements following the onset of a stimulus throughout the scalp can improve the performance of BCIs as well as our understanding of the brain functions responsible for information processing and decision-making. We propose a novel Bayesian model based on dynamic latent factor models, which provide two crucial improvements over existing methods. First, we learn latent factors across EEG electrodes and time which allows dimensionality reduction, abstraction of the BCI design and decomposition of complex electrode-wise responses into interpretable sub-components. Second, we allow spatial correlation across electrodes to vary both with time and stimulus type (target or nontarget), which captures changes in functional connectivity following the onset of a stimulus, along with variations associated with intent. The application of our model to real BCI sessions provide valuable insight by recovering and refining some of the known ERPs, discovering new signals and highlighting variations in functional connectivity that were mostly unexplored so far.

4.1 Introduction

Brain-computer interfaces (BCI) aim to enable direct communication between a subject's brain and some external device. This technology is particularly useful for "locked-in" patients (e.g., suffering from amyotrophic lateral sclerosis) whose voluntary muscle are paralyzed, thus preventing speech or typing, but whose brain is fully functioning. BCIs can therefore restore some communication abilities to otherwise isolated patients, potentially improving mental health and care.

An important subset of BCIs are called *non-invasive* in opposition to *invasive* BCIs, which require surgery to implant electrodes under the scalp. The obvious advantages of non-invasive BCIs are met with the downsides of generally poorer brain activity recording and increased subject-specific tuning. Among non-invasive BCIs, a popular method to record brain activity is through *electroencephalography* (EEG), which measures variations in electric potential at multiple electrodes (or *channels*) scattered throughout the scalp. We refer the reader to Abiri et al. (2019) for a recent review of EEG-based BCIs.

A common approach to BCI control is through the use of *oddball stimuli* where a sequence of stimuli (often visual, auditory or motor) is presented to the user, but only some of the stimuli are relevant. These *target* stimuli induce a different brain response than the *non-target* stimuli, which can be captured in the variations in potentials, as measured by EEG. A specific instance of such visual oddball BCI design is that of the *row-and-column paradigm* (Farwell and Donchin, 1988) in which a screen displays a 6×6 grid of keys (the digital keyboard) and where rows and columns flash in a random order. When a specific row or column contain the target key, the corresponding stimulus is deemed target. Hence, to "type" a key, the user has to positively respond to two out of the 12 stimuli and ignore the other 10 stimuli: the intersection of a row and column uniquely identifies the target key.

The study of brain responses to visual oddball stimuli has led to many discoveries of *evoked* or *event-related* potentials (ERPs). In particular, the P300 ERP has seen wide use in visual BCI systems (Abiri et al., 2019). It is defined as the increase (hence the P in P300 for positive) in potential across the midline electrodes (Fz, Cz and Pz in the international 10-20 system) occurring between 200-500ms after the onset of a target stimulus with a peak around 300ms (hence the 300 in P300). The amplitude and latency of the P300 ERP is known to change with subjects, specifically with age and condition (Dinteren et al., 2014). Comerchero and Polich (1999) further argues for a decomposition of the P300 ERP into two components: the P3a ERP is characterized by a peak in fronto-central activity around a latency of 300-450ms attributed to attention mechanisms, and the P3b ERP is characterized by a peak in temporal-parietal activity around a latency of 350-450ms attributed to information processing (Stige

et al., 2007; Polich, 2007). In addition to P300, the N200 ERP has also received some attention in the context of oddball stimuli response Folstein and Van Petten (2008). It is characterized by a decrease in potential (negative, N) in the posterior scalp between 200 and 350 ms after the onset of a stimulus. Folstein and Van Petten (2008) further argues for a decomposition into two subcomponents, one fronto-central and one posterior, in the context of visual stimuli. Beyond P300 and N200, several ERPs have been identified in similar contexts such as N100 in the fronto-central region, P200 throughout the scalp, and N400 in the centro-parietal regions (Rushby et al., 2002), along with visual evoked potentials such as P100 in the occipital lobe.

The recorded brain activity measured by EEG suffers from multiple sources of signal superposition. First, at a given channel, multiple ERPs affect the potential simultaneously, so it is difficult to tell them apart as they can interfere positively or negatively with each other. This led to the plethora of ERPs identified and their further splitting into subcomponents. Second, when the inter-flash interval is smaller than 200-500ms, which is desirable for fast communication, significant overlap between consecutive responses can occur, further obscuring the effect of a single stimulus. We therefore propose a generative model that directly parameterizes the brain activity with these superimposed signals in mind. To this end, we design a dynamic latent factor model in such a way that the latent components can be interpreted as ERPs. We reject inserting prior knowledge of known ERPs into the model to avoid overfitting to perhaps inexact definitions: the learned latent factors may then rediscover these ERPs, clarify their definition and potentially discover new ones.

Most work on EEG-based BCIs focus on the classification aspect of the data by proposing *discriminative* approaches. These include various statistical and machine learning methods along with feature extraction (see Lotte et al., 2018 for a recent review). While many discriminative methods achieve impressive performance in terms of key prediction, they generally provide only limited scientific insight on brain activity. In contrast, few consider modeling the brain activity directly through a *generative model*. Gonzalez-Navarro et al. (2019) propose a channel-wise model taking into account the superposition of stimulus response and the decomposition into multiple ERPs. In particular, they assume that three signals define the response each at a fixed latency: one for the N100/P100 complex assumed to capture initial sensory response, one for the N200/P200 complex assumed to capture decision making and characterization, and one for the P300/N400 complex assumed to post-response evaluation. To combine channels, a naïve Bayes approach is used for prediction; estimation is done independently. Ma et al. (2022), from which our proposed model is inspired, also considers channel-wise stimulus responses taking into account the stimulus superposition, while no decomposition is performed. Multiple channels are fitted together using a compound symmetry

spatial covariance structure.

Our proposed model borrows from both Gonzalez-Navarro et al. (2019) and Ma et al. (2022), but provides more flexibility, interpretability and abstraction. For instance, we similarly model the superposition due to short stimulus-to-stimulus interval. We extend Gonzalez-Navarro et al. (2019) by generalizing the decomposition from three fixed latency signals to an arbitrary number of signals at arbitrary latencies. We model multiple channels more generally than Ma et al. (2022) by considering a low-rank decomposition, which naturally aggregates channels into latent factors. It further allows more flexible spatial correlation cross channels, and that correlation is permitted to vary with time and stimulus type. This dynamic covariance regression can identify changes in the interactions across channels that were beyond the reach of existing methods based on mean differences alone.

On a related note, our dynamic spatial covariance setting is closely related to the study of *functional connectivity* in brain imaging. Brain connectivity, specifically using EEG data, has been studied in a variety of context (see Preti et al., 2017 and Luo et al., 2022, for reviews), but it has rarely been used to improve BCI systems. Kabbara et al. (2016) propose to use phase synchronization to construct additional features fed to a support vector machine classifier and found increased separability between stimulus type. Li et al. (2016) propose a multivariate auto-regressive model to study the dynamic flow of information across the brain in relation to the P300 ERP. Our dynamic covariance model is inspired by both Andersen et al. (2018) and Tsai et al. (2022) in the context of fMRI data. They propose dictionary approaches where the spatial covariance is constructed from a set of common rank-one matrices whose weights are allowed to change with time, though Tsai et al. (2022) model the temporal sample covariance directly, while Andersen et al. (2018) rather model the raw time series.

In Section 4.2, we introduce and discuss the dynamic latent factor regression model starting from the work of Ma et al. (2022). Section 4.3 is reserved for estimation, prediction, model selection and evaluation. In particular, we propose metrics to quantify the importance of latent factors in discriminating between target and nontarget stimuli. The model selection and component evaluation procedure are assessed through simulation studies presented in Section 4.4. Finally, in Section 4.5, our methodology is applied to a real EEG-based BCI session (Thompson et al., 2014) where we compare the estimated latent components to known ERPs.

4.2 Dynamic Latent Factor Regression Model

4.2.1 Notation

During a BCI spelling session, a subject is attempting to type a series of keys $k_\ell \in \mathcal{K}$, $\ell = 1, \ldots, L$, where L is the length of the message and $\mathcal{K}$ is a collection of keys, the *keyboard*. In our specific BCI2000 Speller instance, the keyboard $\mathcal{K}$ contains 36 keys consisting of letters, numbers, punctuation marks and tools, which are arranged in a 6×6 grid. Let $r(k), c(k) \in \{1, 2, 3, 4, 5, 6\}$ denote, respectively, the row index and column index of key k. We encode a key $k \in \mathcal{K}$ as a two-hot vector $\mathbf{y}(k)$ of length 12 where its $r(k)$ and $c(k) + 6$ entries are set to 1; for example,

$$r(k) = 3,\ c(k) = 1 \qquad \Rightarrow \qquad \mathbf{y}(k) = (\underbrace{0, 0, 1, 0, 0, 0}_{\text{row}}, \underbrace{1, 0, 0, 0, 0, 0}_{\text{column}}).$$

Let $\mathcal{Y}$ denote the space of two-hot vectors such that $\mathbf{y} \in \mathcal{Y}$ satisfies $\sum_{j=1}^{6} y_j = \sum_{j=7}^{12} = 1$.

At a given repetition $r = 1, \ldots, R$ for typing character k_ℓ, encoded by $\mathbf{y}_\ell = \mathbf{y}(k_\ell) \in \mathcal{Y}$, the user is presented with 12 stimuli in a random order, each corresponding to a specific row or column flashing on the screen. We record the ordering of the stimuli through $w_{\ell r,j} = j'$, $j = 1, \ldots, 12$, meaning that the jth stimulus (a specific row or column) was presented j'th in the sequence. For example, $w_{\ell r,4} = 8$ means that 8th flash in the sequence was the 4th row. Aggregating the ordering into the length 12 vector $\mathbf{w}_{\ell r}$, we find that $\mathbf{w}_{\ell r}$ is simply a permutation of the first 12 integers. The 12 stimuli are equally-spaced in time, with stimulus-to-stimulus interval Δ ($\Delta = 156.25$ms in our data, but this varies with BCI designs).

Throughout the session, the subject's brain electrical activity is monitored using *electroencephalography* (EEG) with E electrodes. The complete recording is chunked into $L \times R$ sequences, each corresponding to a character-repetition pair, denoted by $\mathbf{x}_{\ell r}(t) \in \mathbb{R}^E$, where $t \in [0, T]$ is such that $t = 0$ denotes the onset of the first stimulus and T is chosen large enough such that the response to the last stimulus is completely captured. In particular, we assume that a stimulus induces a response measurable in the EEG within T_o of the onset. A typical value for T_o is $T_o = 800$ms (recall that the P300 ERP occurs around 300ms), meaning that $T = 11\Delta + T_o \approx 2,500$ms. In practice, the EEG readings at not measured continuously over $[0, T]$, but rather at a regularly-spaced grid: we denote by $t_1, \ldots, t_S \in [0, T]$ the S sampled time points of each sequence. Finally, we aggregate the EEG readings of a sequence in a $E \times S$ matrix $\mathbf{X}_{\ell r}$ with columns $\mathbf{x}_{\ell r}(t_s)$, $s = 1 \ldots, S$.

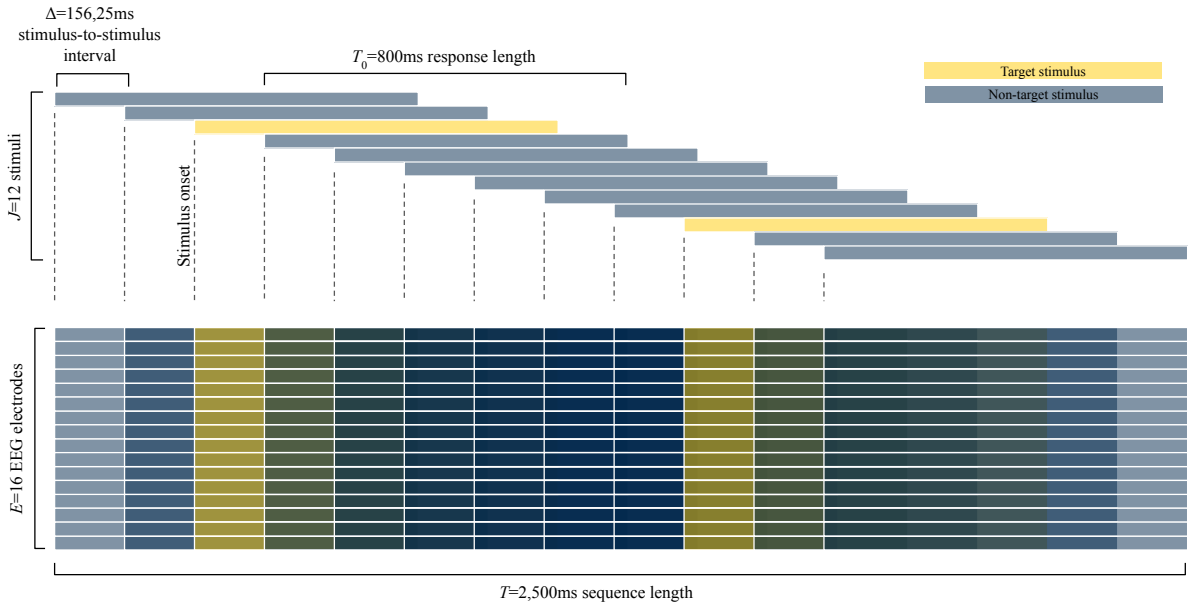

Figure 4.1: Schematic representation of one sequence of stimuli measures across EEG electrodes. Since the expected response is longer than the stimulus-to-stimulus interval, there is a superposition of responses for consecutive stimuli.

4.2.2 Problem description

The main goal of EEG-based BCI spellers is to recover the intended key k_ℓ (equivalently, $\mathbf{y}_\ell$) given the EEG reading $\mathbf{X}_{\ell r}$, possibly aggregating across repetitions. The difficulty in this task emerges from multiple factors. The high-dimensionality of the features $\mathbf{X}_{\ell r}$ ($E \times S = 1,296$ when $E = 16$ and $S = 81$) and the relatively small number of examples ($L \times R = 285$ when $L = 19$ and $R = 15$) requires us to find structure within the data. The random ordering of the stimuli and the changing target implies that whatever signal exists within $\mathbf{X}_{\ell r}$ will occur at different location along the time axis. Additionally, the BCI design we consider is such that signal superposition will occur: indeed, the expected response (on the order of 200-500ms) is longer than the stimulus-to-stimulus interval $\Delta = 156.25$ms (see Figure 4.1). Also, time (across $s = 1, \ldots, S$) and spatial (across $e = 1, \ldots, E$) dependency is expected, and perhaps even interactions between the two could be present.

In addition to the prediction task, we are interested in better understanding the underlying brain functions explaining the variation in electrical activity in response bettween target ($y_{\ell j} = 1$) and non-target ($y_{\ell j} = 0$) stimuli. Notably, we wish to identify interpretable spatio-temporal patterns within $\mathbf{X}_{\ell r}$ associated with the response $\mathbf{y}_\ell$. These could take the form of

a region (an electrode or a group of electrode) changing significantly around a specific delay after a stimulus; the interplay between regions and latencies can provide insight on the brain processes responsible for the response.

4.2.3 Generative prediction model

To simultaneously achieve both goals—prediction and interpretability—, we proceed to a generative model approach. In particular, the effect of a stimulus should be upstream of the observed EEG readings, suggesting specifying a model for the distribution of $\mathbf{X}_{\ell r}$ given the stimulus ordering $\mathbf{w}_{\ell r}$ and the target key $\mathbf{y}_\ell$, namely $p(\mathbf{X}_{\ell r} \mid \mathbf{w}_{\ell r}, \mathbf{y}_\ell)$. Then, Bayes' rule provides us with the predictive distribution of keys given EEG readings:

$$p(\mathbf{y}_\ell \mid \mathbf{X}_{\ell r}, \mathbf{w}_{\ell r}) \;\propto\; p(\mathbf{y}_\ell) p(\mathbf{X}_{\ell r} \mid \mathbf{w}_{\ell r}, \mathbf{y}_\ell).$$

The remainder of this section is therefore devoted to defining an appropriate model $p(\mathbf{X}_{\ell r} \mid \mathbf{w}_{\ell r}, \mathbf{y}_\ell)$ that is sufficiently flexible and robust for accurate predictions, yet with enough structure to recover interpretable relationships between the target stimuli and the observed brain activity.

4.2.4 Static spatial covariance model

Our starting point is the static spatial covariance model proposed by Ma et al. (2022). Spatial and temporal dependence is accounted for using a matrix-normal model:

$$\mathbf{X}_{\ell r} \mid \mathbf{w}_{\ell r}, \mathbf{y}_\ell \sim \mathcal{MN}_{E \times S}\left(\mathbf{M}(\mathbf{w}_{\ell r}, \mathbf{y}_\ell), \Sigma_s, \Sigma_t\right) \tag{4.1}$$

for some mean function $\mathbf{M}(\cdot)$ to be defined, an $E \times E$ spatial covariance matrix Σ_s, and a $S \times S$ temporal covariance matrix Σ_t. The mean function is constructed electrode-wise using a superposition of signals induced by each of the 12 stimuli. Specifically, for each electrode e, we consider two possible responses after a given stimulus: the non-target signal $\beta_{0,e}(\tau)$ and the target signal $\beta_{1,e}(\tau)$. Each $\beta_{y,e}$, $y = 0, 1$, is a real-valued function defined over the response window $\tau \in [0, T_o]$. We define $\tau_{\ell r,j}(t)$ as a time-shift operator mapping time $t \in [0, T]$ to the time τ since the onset of stimulus j:

$$\tau_{\ell r,j}(t) = t - (w_{\ell r,j} - 1)\Delta.$$

Then, the impact of stimulus j on the mean function at time t is given by $\beta_{y_{\ell r,j},e}\left(\tau_{\ell r,j}(t)\right)$, where $y_{\ell r,j}$ selects $\beta_{0,e}$ or $\beta_{0,e}$ depending on whether stimulus j is a target stimulus. Here, we

use the convention that $\beta_{y,e}(\tau) = 0$ whenever $\tau \notin [0, T_o]$, which restricts the effective domain of the response from the onset of stimulus j, $\tau_{\ell r,j}(t) = 0$, to the maximum response range, $\tau_{\ell r,j}(t) = T_o$. Finally, the (e, s) entry of $\mathbf{M}(\mathbf{w}_{\ell r}, \mathbf{y}_\ell)$, corresponding to the mean of $x_{\ell r,e}(t_s)$, is given by summing over the 12 stimuli:

$$m_{\ell r,e}(t_s) = \sum_{j=1}^{12} \beta_{y_{\ell,j},e}\left(\tau_{\ell r,j}(t_s)\right) = \sum_{j=1}^{12} \left[(1 - y_{\ell,j})\beta_{0,e}\left(\tau_{\ell r,j}(t_s)\right) + y_{\ell,j}\beta_{1,e}\left(\tau_{\ell r,j}(t_s)\right)\right]. \qquad (4.2)$$

Hence, at a given time point t_s, the mean response in electrode e is a superposition of multiple responses from the previous stimuli. For stimulus-to-stimulus interval $\Delta = 156.25$ms and response window length $T_o = 800$ms, we find that up to 6 previous stimuli enter the computation of $m_{\ell r,e}(t_s)$. In particular, for an expected response time of 200-500ms, it means that one or more other stimuli occur between the onset of a target stimulus and the actual measured response.

The main object of interest for interpretation is the difference between target and non-target signals, namely, $\beta_{1,e}(\tau) - \beta_{0,e}(\tau)$, which captures the change in mean response at a delay τ after the onset of a target stimulus in electrode e. To induce additional structure, Ma et al. (2022) propose the *split-and-merge Gaussian process* (SMGP) prior, which regularizes the difference $\beta_{1,e} - \beta_{0,e}$, encouraging $\beta_{0,e}$ and $\beta_{1,e}$ to be equal on some subset of the response window $[0, T_o]$. Specifically, the SMGP prior is defined by a non-target process $\alpha_{0,e} : [0, T_o] \to \mathbb{R}$, a target process $\alpha_{1,e} : [0, T_o] \to \mathbb{R}$ and a mixing process $\zeta_e : [0, T_o] \to [0, 1]$ to form

$$\beta_{0,e} = \alpha_{0,e}, \qquad \beta_{1,e} = \zeta_e\alpha_{1,e} + [1 - \zeta_e]\alpha_{0,e}.$$

Note that we can rewrite $\beta_{1,e} - \beta_{0,e} = \zeta_e[\alpha_{1,e} - \alpha_{0,e}]$ as the difference process. Then, whenever the mixing process ζ_e equals 0, the two signals agree. Finally, Gaussian processes priors are applied to the target and non-target processes,

$$\alpha_{y,e} \sim \mathcal{GP}(0, \kappa_\alpha), \qquad y = 0, 1,$$

for some covariance kernel κ_α, and a truncated Gaussian process prior is applied to the mixing process,

$$\zeta_e \sim \mathcal{TGP}_{[0,1]}(0.5, \kappa_\zeta),$$

for some covariance kernel κ_ζ.

Two observations about the model of Ma et al. (2022) led us to consider the present extension. First, upon estimating the electrode-wise target and non-target signals $\beta_{1,e}$ and $\beta_{0,e}$,

they find, perhaps unsurprisingly, that adjacent electrodes feature very similar shapes both in terms of values and split window. This is particularly apparent within the frontal and parietal lobes, each containing three electrodes with essentially identical signal estimates. This hints at a lower-dimensional representation of the signals, which could potentially improve estimation efficiency by reducing model complexity and enhance interpretability. Second, Ma et al. (2022) assume a compound symmetry covariance structure across channels that is constant over time. While this is a decent approximation of the covariance structure, relationships between channels should be more flexible to capture the non-uniform connectivity. Further, correlation between channels is expected to change over time, and with stimulus type, to account for communication between brain regions.

4.2.5 Low-rank dynamic covariance model

The main assumption of our proposed method is that all systematic variability can be captured over a low-dimensional representation of the E electrodes described by a loading matrix $\Theta \in \mathbb{R}^{E\times K}$. In particular, both first- and second-order variability will be along the K directions $\Theta_k \in \mathbb{R}^E$.

To this end, we consider a dynamic factor model (Fox and Dunson, 2015) with extra structure emerging from the response $\mathbf{y}_\ell$ and the stimulus order $\mathbf{w}_{\ell r}$. Specifically, the observation model is univariate Gaussian for each entry of $\mathbf{X}_{\ell r}$, conditionally on some latent processes:

$$\mathbf{x}_{\ell r}(t_s) = \Lambda_{\ell r}(t_s)\mathbf{z}_\ell(t_s) + \varepsilon_{\ell r}(t_s), \qquad \varepsilon_{\ell r}(t_s) \sim \mathcal{N}_E(\mathbf{0}_E, \Sigma), \tag{4.3}$$

where $\Lambda_{\ell r}(\cdot) \in \mathbb{R}^{E\times K}$ is the dynamic *loading process*, where $\mathbf{z}_\ell(\cdot) \in \mathbb{R}^K$ is the dynamic *factor process*, and where $\Sigma = \text{diag}(\sigma_e)_{e=1}^E$ is a diagonal noise variance matrix.

The dynamic loading process factorizes into a static *loading* matrix $\Theta \in \mathbb{R}^{E\times K}$ and a scaling process $\boldsymbol{\xi}_{\ell r}(\cdot) \in \mathbb{R}^K_{>0}$:

$$\Lambda_{\ell r}(\cdot) = \Theta\, \text{diag}[\boldsymbol{\xi}_{\ell r}(\cdot)].$$

Hence, the columns of $\Lambda_{\ell r}(\cdot)$ are simply rescaled versions of the columns of Θ. In particular, the entries of $\boldsymbol{\xi}_{\ell r}(\cdot)$ are expected to be fairly close to 1. Then, we incorporate $\mathbf{y}_\ell$ and $\mathbf{w}_{\ell r}$ using the superposition (4.2) within each of the K latent scaling processes:

$$\log \xi_{\ell r,k}(\cdot) = \sum_{j=1}^{12} \beta^{\xi}_{y_{\ell,j},k}\left(\tau_{\ell r,j}(\cdot)\right), \tag{4.4}$$

where $\beta^{\xi}_{y,k}$, are the non-target ($y = 0$) and target ($y = 1$) *scaling signals*. We note that, given $\mathbf{y}_\ell$ and $\mathbf{w}_{\ell r}$, $\xi_{\ell r,k}(\cdot)$ is a deterministic function of $\beta^{\xi}_{y,k}$, $y = 0, 1$.

The dynamic factor processes are defined as a noisy version of the *mean factor processes*, also defined using the superposition (4.2):

$$z_{\ell r,k}(\cdot) \mid \bar{z}_{\ell r,j}(\cdot) \sim \mathcal{GP}(\bar{z}_{\ell r,j}(\cdot), \kappa_z) \tag{4.5}$$

$$\bar{z}_{\ell r,j}(\cdot) = \sum_{j=1}^{12} \beta^{z}_{y_{\ell,j},k} \left(\tau_{\ell r,j}(\cdot)\right) \tag{4.6}$$

where κ_z is a covariance kernel, and where $\beta^{z}_{y,k}$, are the non-target ($y = 0$) and target ($y = 1$) mean factor signals. In contrast to the scaling processes $\boldsymbol{\xi}_{\ell r}(\cdot)$, the factor processes $\mathbf{z}_{\ell r}(\cdot)$ are stochastic given the shared signals $\beta^{z}_{y,k}$, $y = 0, 1$: this is necessary to enable dynamic spatial covariances.

4.2.6 Model properties

We describe the properties of the model conditionally on the shared quantities, namely, the loading matrix Θ, the scaling signals $\beta^{\xi}_{y,k}$, $y = 0, 1$ and the factor signals $\beta^{z}_{y,k}$, $y = 0, 1$, and the noise variance Σ, collectively denoted by ϕ. Conditionally on ϕ, the dynamic loading processes $\Lambda_{\ell r}(\cdot)$ and the mean factor processes $\bar{\mathbf{z}}_{\ell r}(\cdot)$ are fixed.

The expected value of the EEG readings at a time point t is given by

$$\mathbb{E}\left\{\mathbf{x}_{\ell r}(t) \mid \phi\right\} = \Lambda_{\ell r}(t)\bar{\mathbf{z}}_{\ell r}(t) = \Theta \operatorname{diag}[\boldsymbol{\xi}_{\ell r}(t)]\bar{\mathbf{z}}_{\ell r}(t) = \sum_{k=1}^{K} \bar{z}_{\ell r,k}(t)\xi_{\ell r,k}(t)\Theta_k.$$

We find that the expectation is restricted to the K-dimensional subspace generated by the columns of Θ and regresses on the response $\mathbf{y}_\ell$ through the weighing of each direction Θ_k.

The spatial covariance at a fixed time point t is given by

$$\operatorname{Cov}(\mathbf{x}_{\ell r}(t) \mid \phi) = \kappa_z(t,t)\Lambda_{\ell r}(t)\Lambda_{\ell r}(t)^\top + \Sigma = \kappa_z(t,t)\sum_{k=1}^{K} \xi^2_{\ell r,k}(t)\Theta_k\Theta_k^\top + \Sigma.$$

Here, the spatial covariance takes the form of diagonal plus rank K, where the rank K matrix is only allowed to be a combination of the rank one matrices $\Theta_k\Theta_k^\top$ (sometimes referred to as the *dictionary* approach, Andersen et al., 2018). Variations in $\xi_{\ell r,k}(t)$, which depends on the response $\mathbf{y}_\ell$, only affects the importance of each rank one term $\Theta_k\Theta_k^\top$.

The cross-covariance across two time points $t \neq t'$ is given by

$$\text{Cov}(\mathbf{x}_{\ell r}(t), \mathbf{x}_{\ell r}(t') \mid \phi) = \kappa_z(t,t')\Lambda_{\ell r}(t)\Lambda_{\ell r}(t')^\top = \kappa_z(t,t') \sum_{k=1}^{K} \xi_{\ell r,k}(t)\xi_{\ell r,k}(t')\Theta_k\Theta_k^\top.$$

Provided the scaling processes $\boldsymbol{\xi}_{\ell r}(\cdot)$ stay relatively close to 1, the temporal dependency is primarily determined by the factor process kernel $\kappa_z(t,t')$, as the sum term will feature much less variation in comparison.

4.2.7 Variants

We consider two simplifications to our proposed model. First, we can drop the dependency on the response in the scaling process $\boldsymbol{\xi}_{\ell r}$, while retaining dynamic covariance. The interpretation would be that any stimulus, target or non-target, induces a similar change in spatial correlation. In practice, this amounts to setting $\zeta_k^\xi \equiv 0$ so that the superposition $\xi_{\ell r,k} \equiv \xi_k$ is simply 12 identical signals temporally spaced by Δ. Second, we can drop the dynamic covariance altogether by setting $\boldsymbol{\xi}_{\ell r} \equiv 1$, which essentially amounts to a low-rank version of Ma et al. (2022), though they further assume a compound symmetry spatial covariance structure (see Section C.2 for more details).

We label the full model as `LR-DCR`, for *low-rank dynamic covariance regression*, the model without response in the dynamic covariance as `LR-DC`, for *low-rank dynamic covariance*, and the static covariance model as `LR-SC`, for *low-rank static covariance*. The model of Ma et al. (2022) will be labelled as `FR-CS`, for *full-rank compound symmetry*. A graphical representation of the models can be found in Figure 4.2.

4.3 Inference

4.3.1 Gibbs sampling

Both prediction and interpretation tasks can be resolved using the posterior distribution of the model parameters $p(\phi \mid \mathcal{D})$, where $\mathcal{D} = \{(\{\mathbf{X}_{\ell r}, \mathbf{w}_{\ell r}\}_r, \mathbf{y}_\ell)\}_\ell$ denotes the training data. We obtain approximate samples from the posterior using *Markov chain Monte Carlo* (MCMC). Specifically, we exploit multiple conjugate relationships within the model to perform Gibbs sampling and use Metropolis-within-Gibbs approaches where we lack conjugacy. In particular, the exponential activation between the scaling signals and the scaling processes (4.4) breaks the linear structure required for conjugacy among Gaussian quantities. Hence, we utilize the *Metropolis-adjusted Langevin algorithm* (Roberts and Tweedie, 1996, MALA)

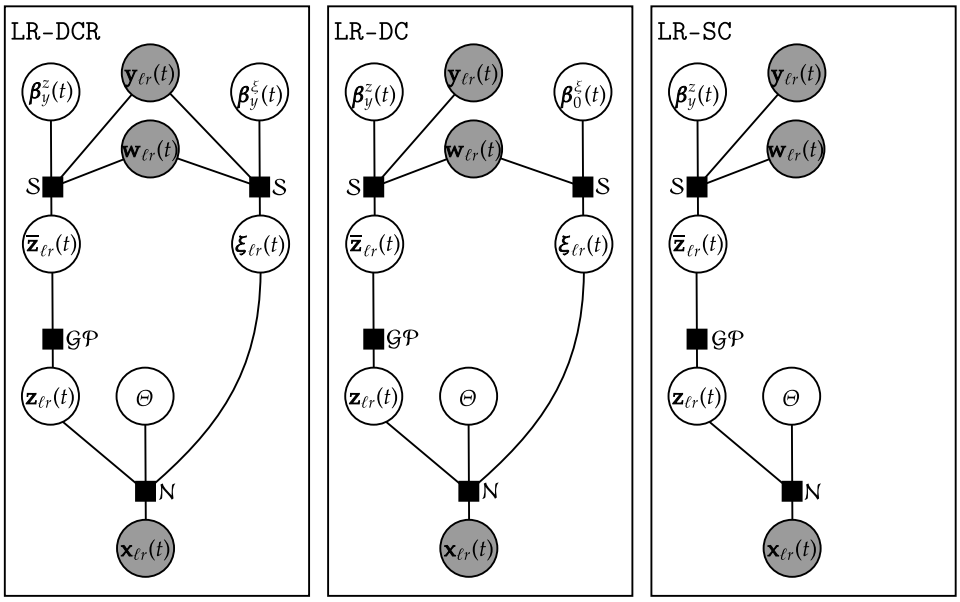

Figure 4.2: Factor graph representation of the proposed model and its variants. Grey and white nodes indicate observed and unobserved quantities, respectively; black squares represent factors, where nodes underneath it are children and nodes above it are parents. The $\mathcal{S}$ factor represent the superposition in (4.6) or (4.4); the $\mathcal{GP}$ factor represents the Gaussian Process term (4.5); the $\mathcal{N}$ factor represents the normal model in (4.3). Note that variant `FR-CS` (Ma et al., 2022) is a special case of `LR-SC`.

to sample from the conditional posteriors of the three components of the scaling signals $\beta^{\xi}_{y,k}$, namely $\alpha^{\xi}_{0,k}, \alpha^{\xi}_{1,k}, \zeta^{\xi}_{k}$. Additionally, even though the conditional posterior for the mixing factor signals ζ^{z}_{k} are truncated Gaussians, we still employ MALA sampling as we found it to be more efficient than direct multivariate truncated Gaussian sampling. We relegate all details to Section C.1 of the Supplementary Materials, including prior specification and initialization.

4.3.2 Prediction

Given R^* test repetitions for an (unobserved) target character $k^* \in \mathcal{K}$, encoded as the two-hot vector $\mathbf{y}^* \in \mathcal{Y}$, with stimuli ordering $\mathbf{w}^*_r$ and EEG recordings $\mathbf{X}^*_r$, $r = 1 \dots, R^*$, we wish to obtain the predictive distribution of the characters, i.e., $p(\mathbf{y} \mid \{\mathbf{X}^*_r, \mathbf{w}^*_r\}_r)$ for all $\mathbf{y} \in \mathcal{Y}$. Using Bayes' rule and a uniform prior over characters, the predictive posterior satisfies

$$p(\mathbf{y} \mid \{\mathbf{X}^*_r, \mathbf{w}^*_r\}_r, \mathcal{D}) \;\propto\; p(\{\mathbf{X}^*_r\}_r \mid \mathbf{y}, \{\mathbf{w}^*_r\}_r, \mathcal{D}) = \int p(\{\mathbf{X}^*_r\}_r \mid \mathbf{y}, \{\mathbf{w}^*_r\}_r, \phi) p(\phi \mid \mathcal{D}) \, \mathrm{d}\phi.$$

Assuming conditional independence across sequences, we find that the joint likelihood factorizes over repetitions:

$$p(\{\mathbf{X}^*_r\}_r \mid \mathbf{y}, \{\mathbf{w}^*_r\}_r, \phi) = \prod_{r=1}^{R^*} p(\mathbf{X}^*_r \mid \mathbf{y}, \mathbf{w}^*_r, \phi).$$

Now, the likelihood $p(\mathbf{X}^*_r \mid \mathbf{y}, \mathbf{w}^*_r, \phi)$ is not available since the model parameters ϕ do not include the local processes $\mathbf{z}^*_r(\cdot)$. Hence, we must marginalize:

$$p(\mathbf{X}^*_r \mid \mathbf{y}, \mathbf{w}^*_r, \phi) = \int p(\mathbf{X}^*_r \mid \mathbf{y}, \mathbf{w}^*_r, \mathbf{Z}^*_r, \phi) p(\mathbf{Z}^*_r \mid \mathbf{y}, \mathbf{w}^*_r, \phi) \, \mathrm{d}\mathbf{Z}^*_r, \tag{4.7}$$

where $\mathbf{Z}^*_r$ is the $K \times S$ matrix with columns $\mathbf{z}^*_r(t^{(s)})$, $s = 1, \dots, S$. Since both terms are Gaussian and $\mathbf{Z}^*_r$ appears linearly in the mean of $\mathbf{X}^*_r$, the integral has an analytical form; details can be found in Section C.1.4 of the Supplementary Materials.

4.3.3 Model selection

The number of components K has to be fixed before estimation. We propose two different approaches for selecting K depending on the desired properties of the fit. In particular, if the main objective is explaining the electrical activity during a BCI session, the likelihood of interest should be that of $\mathbf{x}_{\ell r}(\cdot)$ given $\mathbf{y}_\ell$. Conversely, if the main objective is prediction, or obtaining a simple enough summarization of the brain activity that is sufficient for prediction,

the likelihood of interest should rather be that of $\mathbf{y}_\ell$ given $\mathbf{x}_{\ell r}(\cdot)$. Suppose $\widehat{K}_{x|y}$ is the number of components selected using $p(\mathbf{x}_{\ell r}(\cdot) \mid \mathbf{y}_\ell)$ and $\widehat{K}_{y|x}$ is selected using $p(\mathbf{y}_\ell \mid \mathbf{x}_{\ell r}(\cdot))$. We expect $\widehat{K}_{y|x} \leqslant \widehat{K}_{x|y}$ since adjustment to the whole of $\mathbf{x}_{\ell r}(\cdot)$ should require more flexibility than obtaining accurate posterior probabilities for $\mathbf{y}_\ell$. Then, given a choice of likelihood, selection of K can be performed using commonly-used criteria such as the *widely-applicable information criterion* (Watanabe, 2010, WAIC), *Pareto-smoothed importance sampling leave-one-out cross validation* (Vehtari et al., 2017, 2022, PSIS-LOO-CV), or the Bayes factor (Kass and Raftery, 1995) applied sequentially. Experimentation suggest that the WAIC and PSIS-LOO-CV perform very similarly, whereas the sequential nature of the Bayes factor may lead to undesirable results when considering $p(\mathbf{y}_\ell \mid \mathbf{x}_{\ell r}(\cdot))$. Indeed, suppose that the second component is important in explaining the sequences $\mathbf{x}_{\ell r}(\cdot)$, but does not help in predicting $\mathbf{y}_\ell$, then the Bayes factor decision could accept $K = 1$, even though there might be predictive components beyond the second.

4.3.4 Identification of predictive components

From the initialization scheme (Section C.1.3 of the Supplementary Materials) and the local nature of Gibbs samplers, the estimated components should roughly remain in the same order, i.e., decreasing in size of loading norm. However, similarly to the discussion in Section 4.3.3, that ordering might not match with the *predictive importance* of the components. For example, one might have a large loading $\Theta_{\cdot k}$ while having no difference between target and non-target stimuli. Then, this component captures a large proportion of the variability of the EEG readings, but none of the variability between stimulus type. Identifying which component contribute more to differentiation is highly relevant for understanding the underlying brain functions. To this end, we propose two approaches for ranking components in terms of predictive importance.

The first considers the joint contribution of the loading $\Theta_{\cdot k}$, the scaling signals $\beta^{\xi}_{y,k}$ and the mean signals $\beta^{z}_{y,k}$, $y = 0, 1$. Unfortunately, there is no obvious way to combine all five quantities that matches their effect on the fitted mean because of the superposition. Then, we consider the fictitious case of a single stimulus and compute the difference in fitted means:

$$\Theta \exp\left\{\boldsymbol{\beta}^{\xi}_1(t)\right\} \boldsymbol{\beta}^{z}_1(t) - \Theta \exp\left\{\boldsymbol{\beta}^{\xi}_0(t)\right\} \boldsymbol{\beta}^{z}_0(t) = \sum_{k=1}^{K} \left[\beta^{z}_{1,k}(t) \exp \beta^{\xi}_{1,k}(t) - \beta^{z}_{0,k}(t) \exp \beta^{\xi}_{0,k}(t)\right] \Theta_{\cdot k}.$$

This suggest the following predictive importance of component k:

$$\left\| \left[\beta^z_{1,k}(\cdot) \exp \beta^\xi_{1,k}(\cdot) - \beta^z_{0,k}(\cdot) \exp \beta^\xi_{0,k}(\cdot) \right] \Theta_{\cdot k} \right\|,$$

where $\|\cdot\|$ denotes some norm over vector-valued functions. In practice, we only have access to the functions by their evaluation on a grid over $[0, T_o]$, so we use a Frobenius norm over the matrix of evaluations. Now, that matrix is rank one so the norm simplifies to the product of the norms, i.e.,

$$\left\| \beta^z_{1,k}(\cdot) \exp \beta^\xi_{1,k}(\cdot) - \beta^z_{0,k}(\cdot) \exp \beta^\xi_{0,k}(\cdot) \right\|_2 \left\| \Theta_{\cdot k} \right\|_2 . \tag{4.8}$$

The norm is computed for each posterior sample and the average defines the importance metric. From this expression, we clearly see that for a component to be predictive, it needs both a large difference between stimulus type and a large loading.

The second approach relates to the change in prediction performance. Specifically, to get the predictive importance of component k, we obtain the predicted probabilities while removing the contribution of component k. This is done by setting $\zeta^\xi_k(\cdot) \equiv \zeta^z_k(\cdot) \equiv 0$ in the posterior samples. Then, we compute a given prediction metric, for example, the binary cross entropy, using the *drop-one* probabilities. By comparing the change in the metric across $k = 1, \ldots, K$, we can identify components without whom performance drops the most, which indicates additional predictive information not captured by other components. We similarly define the *add-one* change as the change in a given metric between a null model ($\zeta^\xi_k(\cdot) \equiv \zeta^z_k(\cdot) \equiv 0$ for all k) and a model with only component k active ($\zeta^\xi_j(\cdot) \equiv \zeta^z_j(\cdot) \equiv 0$ for all $j \neq k$), which capture the inherent predictive power of a component, irrespective of other component containing the same information.

4.4 Simulation studies

We generate data according to the `LR-DCR` model with dimensions similar to the real data application in Section 4.5: $L = 19$ characters, $R = 15$ repetitions per character, and $S = 11 \times 5 + 26$ sampled time points. The loading matrix is randomly generated to be in decreasing order with entries picked from $\{-c_j, 0, c_j\}$ in column j for some $c_j > 0$ with increasing sparsity with j. The latent processes are generated such that they start and end at 0 with a damped sine curve such that the main spike occurs between the equivalent of 200-500ms. Figure C.3 of the Supplementary Material contains an example of simulated loadings and processes. We obtain 10,000 posterior samples where all hyper-parameters are set to their generating values. After discarding the first 5,000 samples and thinning by a factor of 10, we have 500

posterior samples to study. For prediction purposes, only 100 posterior samples are used to approximate posterior predictive distributions.

4.4.1 Component importance

In this experiment, we investigate whether the importance metrics proposed in Section 4.3.4 agree with the true generating values. Implicitly, we study the capacity of the model to estimate the true latent components as well as the appropriateness of the importance metrics. Specifically, we set components 1, 3 and 5 to be predictive of stimulus type, while $K = 8$ latent dimensions are used to generate the sequences.

We reorder the components using the cosine similarity between the posterior mean of each loading $\Theta_{\cdot k}$ and the true generating value of $\Theta_{\cdot k}$. We run five independent chains to check the stability of the importance measures. In Figure C.4 of the Supplementary Materials, we include the data likelihood along the five chains as a simple diagnosis: all five chains hover around a similar value, which is slightly above the likelihood computed using the true generating values, as expected.

In Figure 4.3, we display all three measures along with the matrix of cosine similarities. First, we find strong agreement between the three importance measures, suggesting that the cheaper measure based on posterior samples is a good proxy for the more costly ones based on prediction changes. Indeed, while the ordering may change slightly, there remains a clear distinction between relevant and irrelevant components. Comparing the add-one BCE change to the drop-one BCE change, we find that the drop-one variant has improved differentiation. This is particularly apparent for the component that best matches the second true component (estimated component 3 for chains 1 and 5, estimated component 2 for chains 2, 3 and 4), where the add-one BCE assigns some importance to those estimated components, while the drop-one BCE does not. All five chains estimate at least one component exhibiting high similarity with each of the three true predictive components, and they are all found to be important by any of the three measures.

4.4.2 Latent dimension selection

In this simulation study, we investigate the properties of the model selection criteria proposed in Section 4.3.3 with respect to selecting the latent dimension K. We consider four scenarios. For the first two ($K_x = 5$), data is generated such that there are five latent components generating the signal; for the latter two ($K_x = 8$), eight true components generate the data. For each K_x, we consider two scenarios of number of predictive components: in the first ($K_y = 3$), only components 1 and 3 are truly predictive; in the second ($K_y = 5$), components

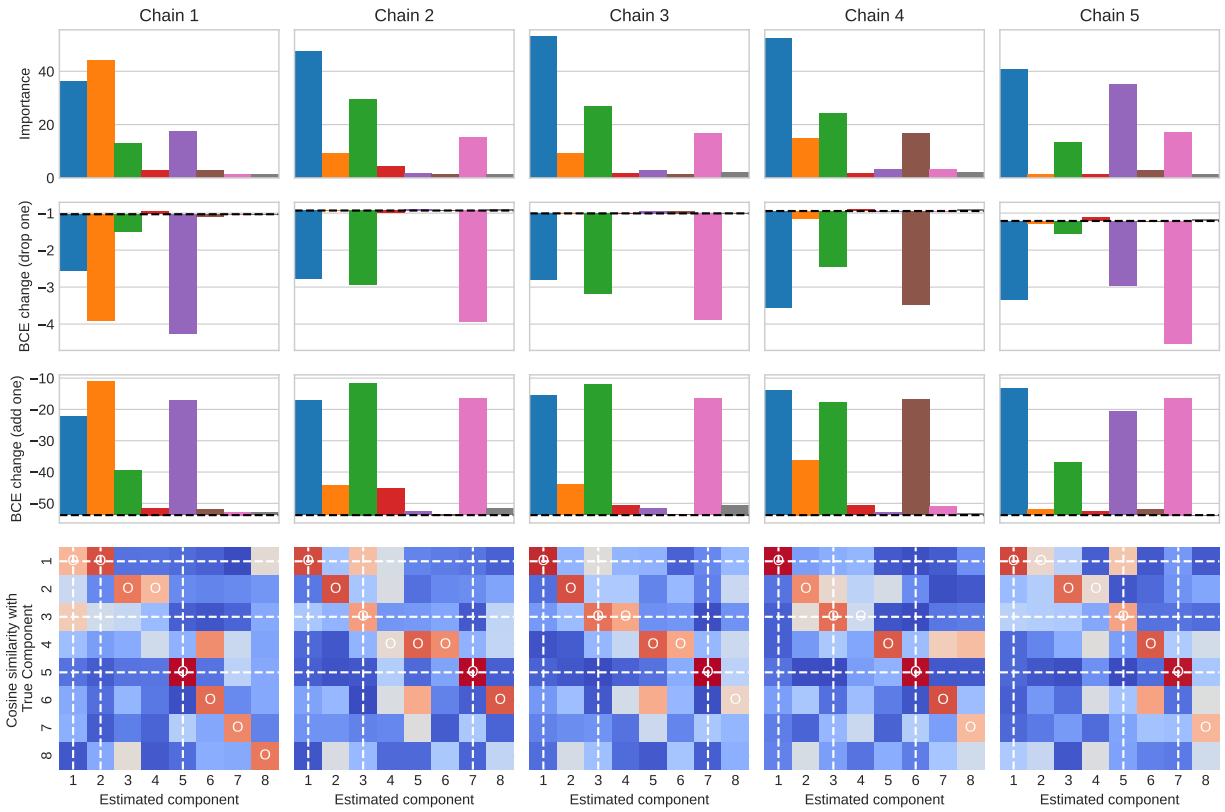

Figure 4.3: Component importance results for the simulation study: (first row) computed using (4.8), (second row) computed as the change in BCE when a specific component is removed from the full model, whose baseline BCE is indicated by the dashed line. (Third row) Component importance computed as the change in BCE when a specific component is added to the null model, whose baseline BCE is indicated by the dashed line. (Fourth row) Cosine similarity between the posterior mean and true values of the loadings (red: high similarity, blue: low similarity); horizontal lines indicate the true predictive component and vertical lines indicate the top three most important components based on the drop-one BCE change.

1, 3 and 5 have differences between stimulus types. We obtain chains for each of latent dimension between $K = 1$ and $K = 10$ and report the PSIS-LOO-CV information criterion using the sequence likelihood, labelled as $p(x \mid y)$, and using the character probabilities, labelled as $p(y \mid x)$. Additionally, we provide the binary cross entropy (BCE) evaluated on an independent test, along with the BCE computed using the true generated values for reference. Results can be found in Figure 4.4

First, the $p(x \mid y)$ criterion accurately finds the true number of components generating the whole data across all four scenarios. Indeed, the PSIS-LOO-CV becomes flat beyond the true K. Second, the $p(y \mid x)$ criterion selects fewer latent dimensions when $K_y < K_x$, as expected. Of note, the $p(y \mid x)$ criterion generally shows some small increase beyond the true K_y until K_x, which can be explained by the fact that the remaining non-predictive components provide better adjustment of the data, leading to more precise predictive probabilities. This can be seen more explicitly with the test BCE computed using the true generating values: in the $K_x = 8$ and $K_y = 3$ scenario, the test BCE still improves slightly beyond $K = 3$. The PSIS-LOO-CV using $p(y \mid x)$ can be interpreted as an in-sample estimate of the out-of-sample BCE: our results show that both the values and the trends generally agrees between the two. This experiment shows that the $p(x \mid y)$ criterion is effective at identifying the true number of components, while the $p(y \mid x)$ criterion is effective at finding sufficiently many components that achieve similar predictive performance.

We refer the reader to Section C.5 of the Supplementary Materials for a simulation study showing that the same model selection criteria can be used for choosing the appropriate model variant.

4.5 EEG-BCI Speller Application

We apply our fitted mode to a participant from Thompson et al. (2014) where a training session consisted of typing the phrase `THE_QUICK_BROWN_FOX` ($L = 19$ characters) for $R = 15$ repetitions per character. A BCI2000 system (Schalk et al., 2004) was used to present the stimuli and record the brain activity. In particular, stimuli correspond to a short flash of a row or column in a 6×6 grid of keys: hence, of the $J = 12$ stimuli presented to type a key, only 2 are target stimuli. The flashes are spaced evenly in time with a stimulus-to-stimulus interval $\Delta = 156.25ms$. The brain activity is recorded with a sampling rate of 256Hz at $E = 16$ EEG electrodes: Figure 4.5 displays their locations and labels.

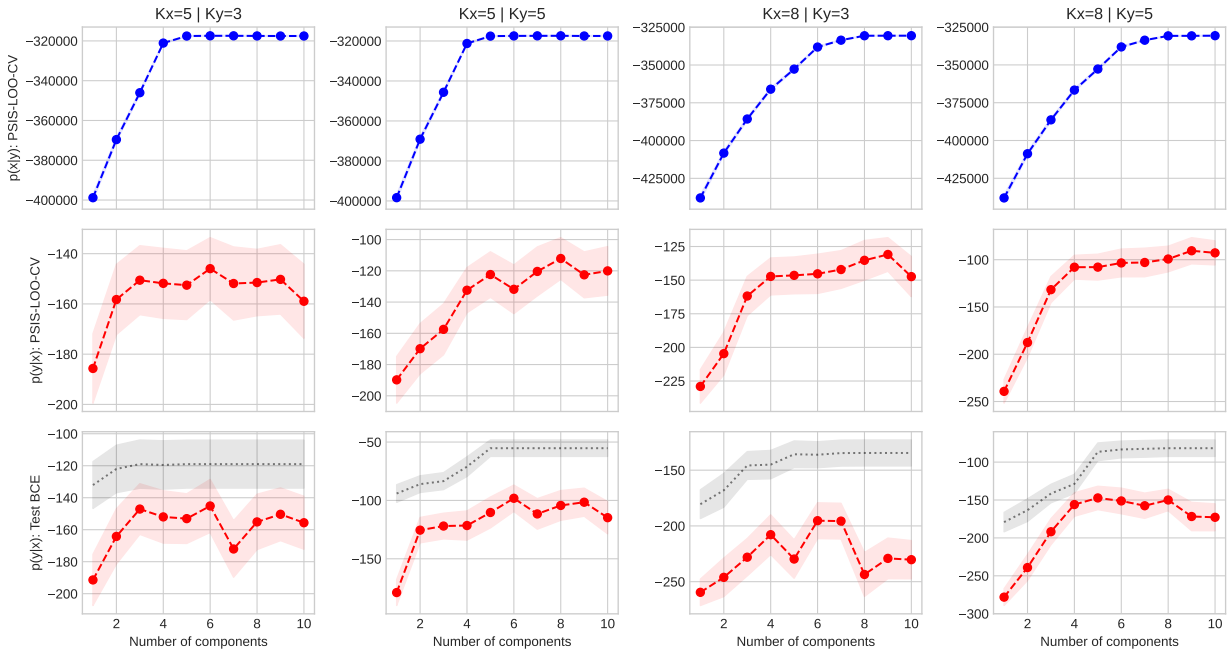

Figure 4.4: Latent dimension selection results for the simulation study: PSIS-LOO-CV estimate (with standard error) for four settings of true generating values. (Top) computed using the likelihood of the sequences and (middle) using the predicted probabilities. (Bottom) BCE evaluated on a test set for the fitted model (red) and using the true generating values (black).

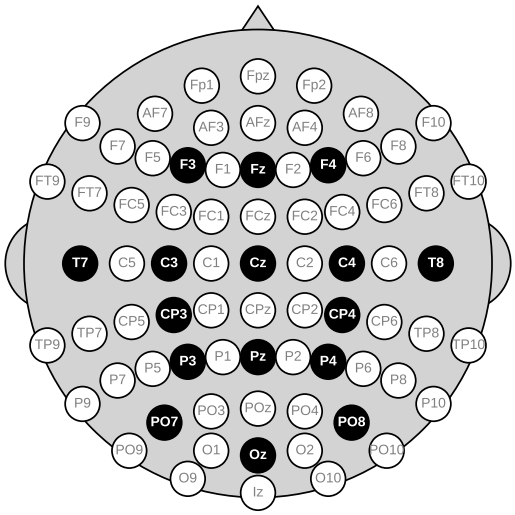

Figure 4.5: Electrodes used in the real BCI application: location and international 10-20 system labels of the 16 EEG electrodes measuring brain activity in Thompson et al. (2014). F: frontal lobe. C: central lobe. P: parietal lobe. O: occipital lobe. T: temporal lobe. z: midline. Even number: right hemisphere. Odd number: left hemisphere.

4.5.1 Data pre-processing and model settings

A bandpass filter between 0.1 and 15Hz was applied to the EEG measurement, which were then chunked into $L \times R = 285$ sequences corresponding to the interval starting from the first flash to $T_o = 800$ms (the response window) after the 12th flash. Hence, each sequence had a length of $(J-1)\Delta + T_o \approx 2,500$ms. Then, sequences were decimated by a factor of eight, which reduces the sampling rate from 256Hz to 32Hz. Thus, each sequence consists of 80 sampling points, the stimulus-to-stimulus interval corresponds to 5 sampling points and the response window corresponds to 25 sampling points.

Unless specifically varied in the experiments, the parameters for the `LR-DCR` model (and its variants) are given as follows. The prior for the observation noise is given by inverse Gamma with $a_\sigma = 1$ and $b_\sigma = 10$. The heterogeneity parameter in the loading prior is set to $\gamma = 3$. The covariance kernels are all set to a squared exponential, parameterized by $\kappa_o \rho^{(i-j)^2}$, where i and j index the equally-spaced time points, for some variance κ_o and one-step correlation $\rho = 0.5$ (equivalent to a Gaussian kernel with scale of $h = -1/\log \rho = 1.44$ or 45ms on the time scale). For mean factor signal and factor process GPs, we set $\kappa_o = 1$; for scaling signals GPs, we set $\kappa_o = 0.1$; for TGPs, we set $\kappa_o = 0.5$. We obtain 20,000 MCMC samples, discard the first half and thin the remaining samples by a factor of 10 for a final set of 1,000 posterior samples. For prediction tasks (including computation of information

criterion during model selection), we thin by a further factor of 10.

4.5.2 Model selection

There are three main parameters to select: the model variant (`LR-DCR`, `LR-DC` and `LR-SC`), the number of components K and the one-step correlation ρ used in the GP priors. We compute the PSIS-LOO-CV (Vehtari et al., 2017) using both the data likelihood $(x \mid y)$ and the predictive key probability $(y \mid x)$ as information criteria for model selection, reported in Figure 4.6.

We find little difference across our model variants for both $x \mid y$ and $y \mid x$ criteria, though the most complicated model, `LR-DCR`, slightly outperforms the other two in terms of key prediction, suggesting there is some slightly predictive information in the correlation. The data likelihood criterion increases with K monotonically, while the key prediction criterion reaches its maximum for $K = 8$, indicating that dimensionality reduction, in the form of low-rank loadings, improves key prediction, but some brain activity remains unexplained by the model. Finally, the one-step correlation parameter ρ controls both the smoothness and strength of time dependency: the data likelihood criterion favors a smaller value around 0.4-0.45, corresponding to a more flexible model, while the key prediction criterion favors a slightly larger value of 0.5-0.55, though both criteria are quite flat in the region. Hence, for the following analysis, we choose the dynamic covariance regression model (`LR-DCR`) with $K = 8$ components and correlation $\rho = 0.5$.

4.5.3 Posterior estimates

Figure 4.7 displays the posterior loadings directions $\Theta_{\cdot k}/\|\Theta_{\cdot k}\|_2$, the posterior difference in mean signal

$$\|\Theta_{\cdot k}\|_2[\beta^z_{1,k}(\cdot)\exp\beta^\xi_{1,k}(\cdot) - \beta^z_{0,k}(\cdot)\exp\beta^\xi_{0,k}(\cdot)] \tag{4.9}$$

and the scaling difference

$$\|\Theta_{\cdot k}\|_2^2\exp\{\beta^\xi_{1,k}(\cdot) - \beta^\xi_{0,k}(\cdot)\} \tag{4.10}$$

for all $K = 8$ components, ordered by increasing binary cross-entropy change when removed (Section 4.3.4). From Section 4.2.6, the difference in mean signal captures the variation in the fitted mean between a target stimulus and a nontarget stimulus after a flash in the direction of the corresponding loading $\Theta_{\cdot k}/\|\Theta_{\cdot k}\|_2$. The scaling difference captures the change in covariance in the direction of the standardized rank-one matrix $\Theta_{\cdot k}\Theta_{\cdot k}^\top/\|\Theta_{\cdot k}\|_2^2$.

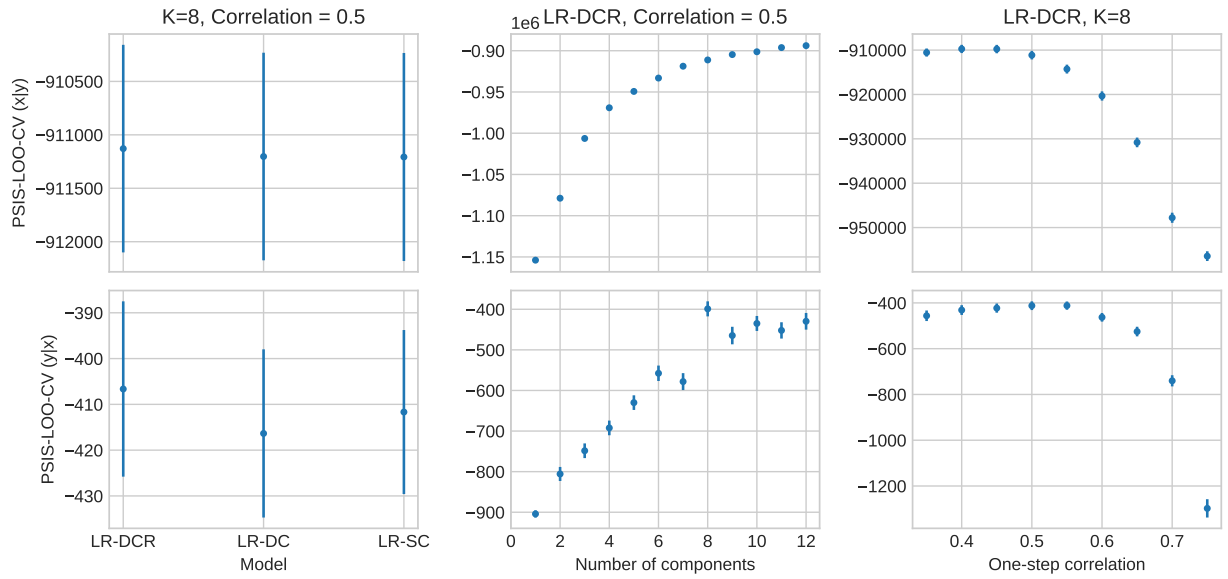

Figure 4.6: Model selection results for the real BCI application: criteria (with standard error) across variants, number of components and correlation hyper-parameter for the 15 training repetitions of a participant in Thompson et al. (2014).

Component 6 is the most predictive and corresponds to a decrease in mean around the parietal and occipital channels (notably channel PO8, but also PO7, OZ and P4) and a small increase for the other channels between 150ms and 350ms, with a peak around a latency of 250ms. This result matches with that of Ma et al. (2022), which also identifies PO8, PO7, Oz and P4 as the top four predictive channels, but our model aggregates these four channels into a single component. This component seems to match we the ERP labelled as N2c or posterior N200 (Folstein and Van Petten, 2008). Component 1 features an increase in mean activity concentrated in the fronto-central region with a similar peak as Component 6, though the change lasts until a latency of 500ms, with a second smaller peak around 450ms. This component features similar characteristics to the P3a ERP, also referred to as "novelty P300" (Polich, 2007). Component 7 brings adjustments to Component 1 by distinguishing three regions: decrease in the left parietal lobe, increase in the right parietal lobe and increase in the right frontal and temporal lobes, which suggest a correspondence with the N2b ERP or "novelty N200" (Folstein and Van Petten, 2008). Component 3 generally distinguished between regions adjacent to C4 (increase) and the left parietal lobe (decrease) over a longer time frame (150ms to 550ms) with a peak around 350ms, which share some similarities with the P3b ERP (Polich, 2007). Component 2 features an increase in activity around a latency of 350ms to 600ms away from the frontal lobe and perhaps relates to the N400 ERP, which

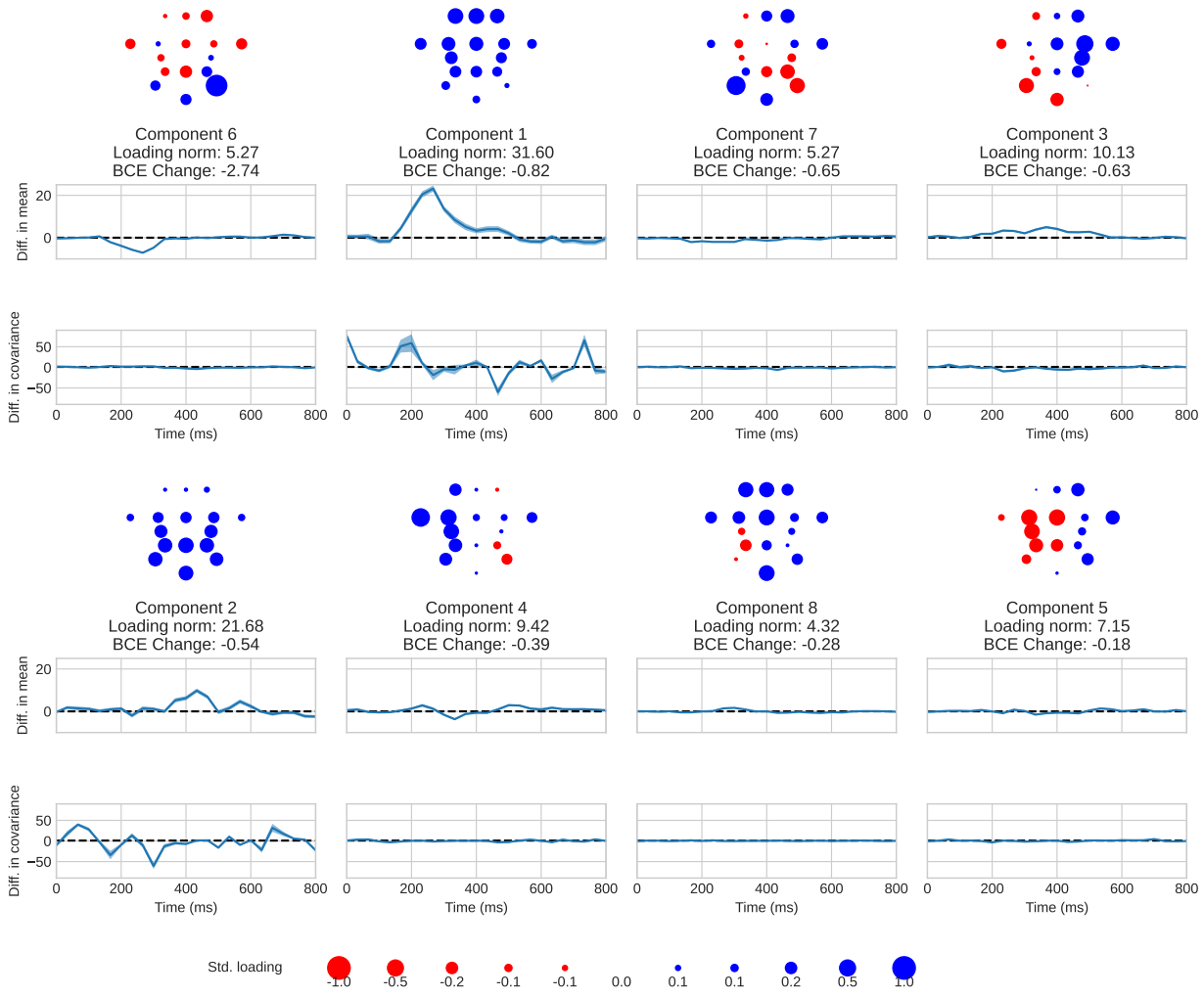

Figure 4.7: Estimated latent factors for the real BCI application: posterior summaries for the $K = 8$ components estimated using all 15 training repetitions of a subject in Thompson et al. (2014). Components are ordered by predictive importance, as measured by the change in binary cross-entropy (BCE) when dropped. For each component: (top) posterior mean of the loading standardized to norm one, (middle) posterior mean and pointwise standard deviation difference in mean (4.9), and (bottom) posterior mean and pointwise standard deviation difference in scaling (4.10). Refer to Figure 4.5 for electrode labels.

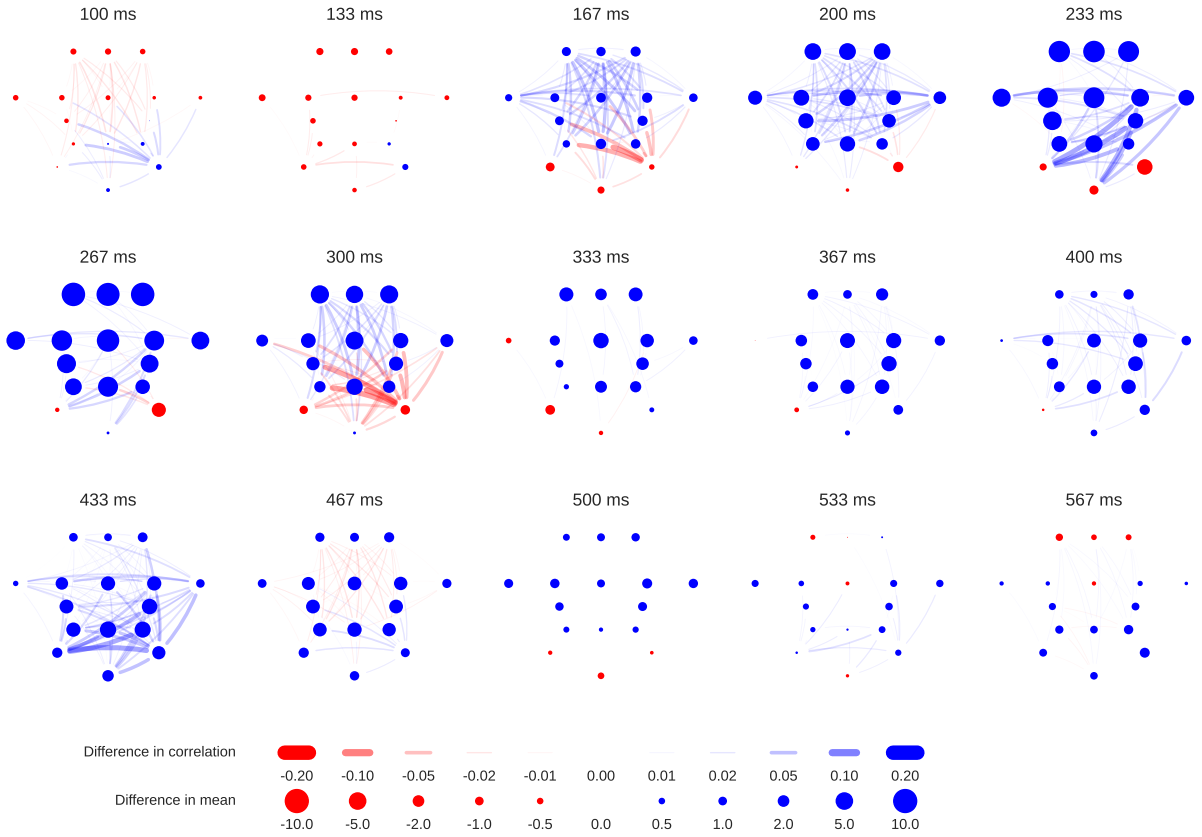

Figure 4.8: Mean and correlation changes for the real BCI application: for each latency after the onset of a stimulus, (nodes) posterior mean difference in fitted mean between target and non-target stimulus at a given channel and (links) posterior mean difference in fitted correlation between target and non-target stimulus for a given pair of channels. Refer to Figure 4.5 for electrode labels.

was found to be of little predictive power in Rushby et al. (2002), as do we. Component 4 features an increase over the left hemisphere around 250ms followed by a decrease around 350ms and a further increase around 500ms.

Mostly two components (1 & 2) capture differences in covariance between target and nontarget signals, though Components 3, 6, and 7 show some small differences as well. Component 1 concentrates more towards the centro-frontal region where it captures increased correlation around latencies of 200ms and 700ms and decreased correlation around 500ms. Interestingly, the first two peaks in correlation changes match with the start and end of the change in mean, with directions matching. Component 2, conversely, concentrates towards the parietal and occipital lobes and features an increase in correlation around 100ms and 700ms and decrease in correlation around 150ms and 300ms.

Figure 4.8 displays the aggregate effect of all 8 components in terms of mean and correlation changes over time. In terms of mean differences, the timeline is as follows. Starting from 150ms and until 300ms, channels PO7, PO8 and Oz decrease in electrical activity while the other channels increase. Then, from 300ms to 450ms, that increase diminishes in the frontal lobe and remains the same in the central and parietal lobes, while the PO7, PO8 and Oz now feature a small increase in activity. In terms of correlation, we find a few events of interests. Around 150ms, there is a general increase in correlation among the frontal and central channels and some decrease in correlation between PO8 and the central and parietal channels. Then, around 250ms, we find an increase in correlation between PO7 and Oz and the right central lobe. Around 300ms, we find a decrease in correlation between PO8 and the parietal and central channels, while the frontal lobe has increased correlation with those same central and parietal channels. Finally, we find an increase in correlation between most of the parietal and occipital channels around 400ms.

4.6 Discussion

Our proposed generative model for brain activity in oddball paradigm BCIs relies on the decomposition of the signal into latent factors. This assumption is quite natural, since the EEG channels only provide discrete measurements of a continuous process in the brain. In particular, there is no expectation that regions of relevant activity exactly map to specific channels, so aggregating channels into latent features allows an abstraction of the system design. Our application (Section 4.5) confirms this as the estimated loadings are generally very smooth even though spatial location was not used to guide estimation.

Additionally, the latent factors enable the decomposition of complex stimulus responses into simpler components, which is not possible with channel-wise mean differences such as in Ma et al. (2022). As discussed in Section 4.5.3, we recover many of the ERPs proposed in the literature, such as the P300 and N200, and their subcomponents, along with other signals to be further studied. In Section C.4.3 of the Supplementary Materials, we repeat the analysis for other subject and find similar latent factors, with some expected variations in amplitude, latency, spatial distribution and relative importance.

The latent factor decomposition also automatically allows spatial dependency to be captured without any assumption beyond the rank. By including the scaling processes, we further enable dynamic covariance, and, by including stimulus type information in these scaling processes, we thus allow dynamic covariance regression. In the participant we studied in Section 4.5, we found some evidence that channel covariance changes over time and with stimulus type. Indeed, we find signals in the two largest components which are such

that the peaks in correlation changes occur either before or after their respective changes in mean, suggesting that the model is capturing some communication or synchronicity patterns. This feature of the model is quite novel and could help understand the dynamics behind the P300 ERP. Our approach rely on the assumption that the variation of the covariance lie in the same directions as the variations of the mean, which may not be sufficient. An alternative covariance model, perhaps decoupled from the mean, could better capture the dynamic covariance. Additionally, our model is not particularly well-suited to model autocorrelation, which could be better accommodated by autoregressive models. Indeed, we mostly capture the spatial covariance over time without lag. Hence, there is no sense of "directionality", which prevents some interpretability.

In terms of prediction power, our approach yields comparable performance as simple classifiers, such as support vector machines, where all achieve perfect or near-perfect accuracy (Section C.4.1 of the Supplementary Materials). However, our method is *generative* rather than *discriminative*, meaning we can extract more information about the underlying brain functions. This comes at a significant computational cost, preventing its practical adoption. Our implementation achieves sampling rates on the order of one second per MCMC sample, meaning that a 10,000-long chain takes a few hours to obtain. We experimented with using only the MAP for prediction to lessen the computational burden, but obtain worse performance, which suggests that uncertainty quantification inherent to Bayesian learning is important for a flexible model such as ours. An alternative estimation procedure based on variational inference (Blei et al., 2017; Andersen et al., 2018) could potentially achieve an optimal tradeoff of computation and performance.

The Bayesian framework allows the incorporation of external knowledge into the model through stronger priors. A notable example is cross-participant estimation. Indeed, while different participants display different response patterns, there remains strong commonalities among them. In the Supplementary Materials (Section C.4.3), we include several examples of posterior estimates for various subjects and find many similarities with those in Figure 4.7. Hence, this suggests that loadings and processes could be guided by a prior informed from other participants. Furthermore, we expect the participant-level parameters to cluster with age and conditions, so a further regression on demographic or clinical variables may improve estimation. This extension could potentially decrease the amount of training data required to obtain accurate estimates.

CHAPTER 5

Conclusion & Future Directions

Through three examples, we have demonstrated the necessity and potential of modeling sources of dependencies in statistical analyses. In Chapter 2, we have shown that accounting for dependencies between subjects organized in a network can significantly improve imputation of subject attributes. In Chapter 3, modeling of serial correlation in longitudinal studies proved to be essential for accurate estimation and support recovery. In Chapter 4, we allow spatial dependency across EEG electrodes to vary with time and stimulus type, which provided valuable insight into the brain processes responsible for information processing and decision-making. Still, some sources of dependencies exist beyond those accounted for, which paves the way for future opportunities for improvement.

In Chapter 2, we have shown that a Bayesian treatment of latent space models for networks can improve the estimation of the latent position along with the prediction of node attributes. Further, the variational inference approach proved to be of similar computational efficiency as frequentist methods. Hence, this suggests that our estimation approach could be extended to related network settings including more general links (such as signed, weighted or counts) along with multiple edge measurements (e.g., multi-plex or temporal) where the latent variables would capture the dependencies across measurements. More closely related to the imputation, we discussed an extension to weaker missing value assumptions by modeling the missingness mechanism directly. This would lead to a model capturing the dependency between an attribute's missingness status with its value, with other attributes' values and missingness statuses and with the edges' presence or absence.

In Chapter 3, an important limitation of the proposed analysis is that we treat taxa as independent of each others. However, there are two main sources of dependencies among them that could lead to improve estimation and interpretability. First, microbial abundance data is *compositional* since the total read count varies across samples and is generally considered to be uninformative. Hence, normalization must be applied for comparisons to be meaningful, which introduces negative correlation across taxa. Omitting this important factor can lead to spurious discoveries. Second, the microorganisms composing a microbiota

exist in a common ecosystem where complex interdependencies between species occur. On the one hand, two species' abundances may interact negatively through parasitism, predation or competition. On the other hand, their abundances may be positively correlated from mutualist or commensal relationships. When studying hundreds if not thousands of communities, these interactions become increasingly complex. Many researchers have proposed methodology to estimate these large interaction network, from either marginal or conditional dependence perspective, but few solutions adequately accounts for all the peculiarities of microbial abundance data, namely, compositionality, zero-inflation and over-dispersion. Still, in the spirit of working covariance models, efficiency gains can be obtained even with slightly misspecified dependency structures. Hence, this suggests that we could extend our proposed methodology to multivariate responses using an appropriately chosen covariance structure and potentially expect improved estimation. Since the multivariate response would be high-dimensional (the number of taxa often exceed the sample size in such studies), it will be necessary to make some relatively strong assumptions, including rank constraints, sparsity of the inverse covariance, or parameterization through taxonomic or phylogenetic tree information.

In Chapter 4, we propose an innovative approach to model dependency across EEG electrodes through time and in relation to stimulus type. While we obtain novel insights in the functional connectivity of the brain, the model is quite limited in multiple ways. First, the dictionary approach used ties the direction of variability of the covariance to the direction of variability of the mean. This facilitates interpretation and estimation, but a more flexible approach could decouple the two. Second, while we model the spatial dependency varying with time, the model does not particularly capture dependencies *through* time. Indeed, a single parameter is responsible to model all the temporal dependency, irrespective of latencies, stimulus type or electrodes. A more informative approach would model the autocorrelation directly by capturing the directed flow of information across the brain. Another type of heterogeneity in the data could be exploited to improve on the current approach. We currently model each participant independently of each other, but we find that relevant signals share some similarities across subjects. Hence, borrowing strength across subjects, perhaps with informed or hierarchical priors, could reduce noise and thus improve estimation. Further, it could align the latent factors across participants, leading to more interpretable and generalizable components. Indeed, the remaining variations between subjects will be easier to analyze.

APPENDIX A

Supplementary Materials to Chapter 2

A.1 Variational message passing

We employ the framework and notation of Minka (2005). Variational message passing (VMP) is a special case of message passing to minimize α-divergences. In particular, VMP utilizes the (exclusive) Kullback-Leibler divergence as the measure to minimize: recall that maximizing the ELBO is equivalent to minimizing the KL between the true posterior and the approximating distribution over the approximating family.

Consider a Bayesian network $p(\mathbf{x}) = \prod_a f_a(\mathbf{x})$, where $f_a(\mathbf{x})$ are known as *factors*, over some variables $\mathbf{x}$, and where each factor typically depends on only a few components of $\mathbf{x}$, say $\mathbf{x}_{n(a)}$, where $n(a)$ denotes the set of neighbors of factor a. Each factor is then approximated by $\tilde{f}_a(\mathbf{x})$ by minimizing the chosen divergence within some family $\mathcal{F}_a$:

$$\tilde{f}_a = \arg\min_{\tilde{f}_a \in \mathcal{F}_a} \mathrm{KL}(\tilde{f}_a q_{\neg a} \| f_a p_{\neg a})$$

where $p_{\neg a} = p/f_a = \prod_{b \neq a} f_b$ and $q_{\neg a} = q/\tilde{f}_a = \prod_{b \neq a} f_b$. We note that $f_a p_{\neg a} = p$ and $\tilde{f}_a q_{\neg a}$ implies that we are simply minimizing the KL between the true posterior and the approximate posterior, by updating a single term of the approximate posterior.

In the fully-factorized case, the approximate factors take the form of products over its neighbors known as *messages*:

$$\tilde{f}_a(\mathbf{x}) = \prod_i m_{a \to i}(x_i) = \prod_{i \in n(a)} m_{a \to i}(x_i).$$

Then, rearranging terms, we get the approximate posterior as the product of messages:

$$q(\mathbf{x}) = \prod_a \tilde{f}_a(\mathbf{x}) = \prod_a \underbrace{\prod_i m_{a\to i}(x_i)}_{\tilde{f}_a(\mathbf{x})} = \prod_i \underbrace{\prod_a m_{a\to i}(x_i)}_{q(x_i)} =: \prod_i q(x_i), \tag{A.1}$$

where $q(x_i) = \prod_a m_{a\to i}(x_i) = \prod_{a\in n(i)} m_{a\to i}(x_i)$. By choosing appropriate families $\mathcal{F}_a$, we can ensure that the product of distributions is tractable. In particular, we will require that all messages are (proportional to a) Gaussian so that their product is also Gaussian. This will yield a fully factorized Gaussian posterior, as desired. We define the reverse messages (node to factors) to satisfy

$$q(x_i) = m_{a\to i}(x_i) m_{i\to a}(x_i),$$

which implies

$$m_{i\to a}(x_i) = \frac{q(x_i)}{m_{a\to i}(x_i)} = \prod_{b\neq a} m_{b\to i}(x_i), \tag{A.2}$$

that is, the aggregation of all messages inbound to i, except the message from a itself.

In VMP, we employ a fixed point algorithm to update all messages in sequence. In particular, for the KL divergence, we update variable-to-factor message using (A.2) and factor-to-variable messages using

$$m_{a\to i}(x_i) \leftarrow \exp\left(\mathbb{E}_{q_{\neg i}}\{\log f_a(\mathbf{x})\}\right) = \exp\left(\int \prod_{j\neq i} m_{a\to j}(x_j) m_{j\to a}(x_j) \log f_a(\mathbf{x}) \, \mathrm{d}\mathbf{x}_{\neg i}\right) \tag{A.3}$$

In our factor model, we add some intermediary representations using deterministic relationship between variables to ease message computation. The message to variable updates (A.3) are however often invalid for equality factors. Instead, we must use the following update:

$$m_{a\to i}(x_i) \leftarrow \mathrm{Proj}\left(\int \prod_{j\neq i} m_{j\to a}(x_j) f_a(\mathbf{x}) \, \mathrm{d}\mathbf{x}_{\neg i}\right) \tag{A.4}$$

where $\mathrm{Proj}(\cdot)$ stands for projection onto the appropriate family. In our case, projection using the KL divergence amounts to moment matching. For Gaussian messages, we thus need to compute the mean and variance of x_i under the distribution that is the argument to the projection operator.

Algorithmically, variational message passing works as follows. First, we initialize all messages. We have found that the initialization does not influence convergence much, except that it is important to break symmetry by randomizing some messages and the loading matrix $\mathbf{B}$. Second, we initialize the posterior of all variables by taking the product of all incoming messages (A.1). Then, we update all messages in sequence until convergence using (A.2), (A.3) or (A.4). For computational efficiency, we compute the message to factors (A.2) using the division formula rather than recomputing the product every time. Similarly, if we update a message to variable x_i, e.g., $m_{a\to i}(x_i)$ to $m^{\text{new}}_{a\to i}(x_i)$, we update the posterior using

$$q^{\text{new}}(x_i) = m^{\text{new}}_{a\to i}(x_i) m_{i\to a}(x_i) = q(x_i)\frac{m^{\text{new}}_{a\to i}(x_i)}{m_{a\to i}(x_i)}. \tag{A.5}$$

This expression avoids re-computing the full product every iteration. Properties of Gaussian densities imply that products and quotient of Gaussian densities are proportional to a Gaussian density, which means the above update remains in the Gaussian family. This property is crucial to the algorithm as leaving the family would lead to exponentially complicated messages. In some cases, the exact message will not be Gaussian so we apply some approximation yielding a Gaussian message.

A.1.1 Useful Gaussian density properties

The multivariate Gaussian density takes the form

$$\varphi_K(\mathbf{x} \mid \boldsymbol{\mu}, \Sigma) = \det(2\pi\Sigma)^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

We also consider the natural parameterization, which enables singular densities since this form allows singular precisions. Rearranging terms, we find

$$\begin{aligned}\varphi_K(\mathbf{x} \mid \boldsymbol{\mu}, \Sigma) &= \exp\left(-\frac{1}{2}\log\det(\Sigma) - \frac{K}{2}\log(2\pi) - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)\\ &= \exp\left(-\frac{1}{2}\log\det(\Sigma) - \frac{K}{2}\log(2\pi) - \frac{1}{2}\mathbf{x}^\top\Sigma^{-1}\mathbf{x} + \mathbf{x}^\top\Sigma^{-2}\boldsymbol{\mu} - \frac{1}{2}\boldsymbol{\mu}^\top\Sigma^{-1}\boldsymbol{\mu}\right)\end{aligned}$$

Define the precision $\Gamma = \Sigma^{-1}$ and the precision-adjusted mean $\boldsymbol{\tau} = \Sigma^{-1}\boldsymbol{\mu}$ to write

$$\begin{aligned}\varphi_K(\mathbf{x} \mid \boldsymbol{\mu}, \Sigma) &= \exp\left(\frac{1}{2}\log\det(\Gamma) - \frac{K}{2}\log(2\pi) - \frac{1}{2}\mathbf{x}^\top\Gamma\mathbf{x} + \mathbf{x}^\top\boldsymbol{\tau} - \frac{1}{2}\boldsymbol{\tau}^\top\Gamma^{-1}\boldsymbol{\tau}\right)\\ &=: \tilde{\varphi}_K(\mathbf{x} \mid \boldsymbol{\tau}, \Gamma)\end{aligned}$$

In particular, we note the integration constant

$$\int \exp\left(-\frac{1}{2}\mathbf{x}^\top \Gamma \mathbf{x} + \mathbf{x}^\top \boldsymbol{\tau}\right)\, \mathrm{d}\mathbf{x} = \exp\left(-\frac{1}{2}\log\det(\Gamma) + \frac{K}{2}\log(2\pi) + \frac{1}{2}\boldsymbol{\tau}^\top \Gamma^{-1}\boldsymbol{\tau}\right)$$

The entropy is given by

$$\mathbb{E}\{-\log\varphi(\mathbf{x} \mid \boldsymbol{\mu}, \Sigma)\} = \frac{1}{2}\left[\log\det(2\pi\Sigma) - K\right]$$

The cross-entropy is given by

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(\boldsymbol{\mu}_1,\Sigma_1)}\left\{-\log\varphi(\mathbf{x} \mid \boldsymbol{\mu}_2, \Sigma_2)\right\} = \\ \frac{1}{2}\left[\log\det(2\pi\Sigma_2) + \mathrm{Tr}(\Sigma_2^{-1}\Sigma_1) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \Sigma_2^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\right]$$

The Kullback-Leiber divergence between two multivariate Gaussians is given by

$$\begin{aligned}\mathrm{KL}(\mathcal{N}_1 \| \mathcal{N}_2) &= \mathbb{E}_{\mathcal{N}_1}\{\log\varphi(\mathbf{x} \mid \boldsymbol{\mu}_1, \Sigma_1) - \log\varphi(\mathbf{x} \mid \boldsymbol{\mu}_2, \Sigma_2)\} \\ &= \frac{1}{2}\left[\log\frac{\det(2\pi\Sigma_2)}{\det(2\pi\Sigma_1)} - K + \mathrm{Tr}(\Sigma_2^{-1}\Sigma_1) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \Sigma_2^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\right]\end{aligned}$$

Products and quotients of Gaussian densities often occurs in message computation. These are easier to compute under the canonical parameterization:

$$\tilde{\varphi}(\mathbf{x} \mid \boldsymbol{\tau}_1, \Gamma_1)\tilde{\varphi}(\mathbf{x} \mid \boldsymbol{\tau}_2, \Gamma_2) \propto \tilde{\varphi}(\mathbf{x} \mid \boldsymbol{\tau}_1 + \boldsymbol{\tau}_2, \Gamma_1 + \Gamma_2).$$

This property extends to products of more than two Gaussians:

$$\prod_{i\in\mathcal{I}} \tilde{\varphi}(\mathbf{x} \mid \boldsymbol{\tau}_i, \Gamma_i) \propto \tilde{\varphi}\left(\mathbf{x} \mid \textstyle\sum_{i\in\mathcal{I}} \boldsymbol{\tau}_i, \sum_{i\in\mathcal{I}} \Gamma_i\right).$$

Similarly, for quotients

$$\frac{\tilde{\varphi}(\mathbf{x} \mid \boldsymbol{\tau}_1, \Gamma_1)}{\tilde{\varphi}(\mathbf{x} \mid \boldsymbol{\tau}_2, \Gamma_2)} \propto \tilde{\varphi}(\mathbf{x} \mid \boldsymbol{\tau}_1 - \boldsymbol{\tau}_2, \Gamma_1 - \Gamma_2).$$

A.1.2 Equality constrained variables

For a deterministic variable c with parent factor $\delta(c - h(\mathbf{p}))$, that is, the equality constraint $c = h(\mathbf{p})$ between a child node c and some parent nodes $\mathbf{p}$, the message to the child will be given by moment matching. In particular, for Gaussian messages, we need to compute $\mathbb{E}_{\to\delta}\{h(\mathbf{p})\}$ and $\mathrm{Var}_{\to\delta}(h(\mathbf{p}))$, where expectations are taken with respect to the distribution

induced by the messages incoming to the factor. Indeed, for $\mathbf{p} = \{p_1, \ldots, p_m\}$, we have to project the following (potentially unscaled) density onto the appropriate family:

$$p(c) = \int \prod_{i=1}^{m} m_{p_i \to \delta}(p_i) \delta(c - h(\mathbf{p})) \, \mathrm{d}\mathbf{p}$$

The KL projection step onto some family $\mathcal{F}$ can be expressed as

$$\mathrm{Proj}_{\mathcal{F}}(p) = \arg\min_{q \in \mathcal{F}} \mathrm{KL}(p \parallel q)$$

When $\mathcal{F}$ is chosen to be an exponential family, we can show that the projection is equivalent to moment matching. Indeed, let

$$q(c \mid \eta) \propto \exp\left(\eta^\top T(c) - A(\eta)\right)$$

which satisfies

$$\nabla_\eta A(\eta) = \mathbb{E}_\eta\{T(c)\}.$$

Then, we write

$$\begin{aligned}
\mathrm{KL}(p \parallel q) &= \mathbb{E}_{c \sim p}\{\log q(c) - \log p(c)\} \\
&= \int [\log q(c) - \log p(c)] \int \prod_{i=1}^{m} m_{p_i \to \delta}(p_i) \delta(c - h(\mathbf{p})) \, \mathrm{d}\mathbf{p} \, \mathrm{d}c \\
&= \int [\log q(h(\mathbf{p})) - \log p(h(\mathbf{p}))] \prod_{i=1}^{m} m_{p_i \to \delta}(p_i) \, \mathrm{d}\mathbf{p} \\
&= \mathbb{E}_{\to \delta}\{\log q(h(\mathbf{p})) - \log p(h(\mathbf{p}))\} \\
&= C_0 + \mathbb{E}_{\to \delta}\{\eta^\top T(h(\mathbf{p})) - \log p(h(\mathbf{p}))\} - A(\eta)
\end{aligned}$$

Setting gradients with respect to η equal to 0 yields the moment matching equation

$$\mathbf{0} = \mathbb{E}_{\to \delta}\{T(h(\mathbf{p}))\} - \nabla_\eta A(\eta),$$

that is,

$$\mathbb{E}_{\to \delta}\{T(h(\mathbf{p}))\} = \mathbb{E}_\eta\{T(c)\}.$$

A.1.3 Message calculations

Figure A.1 shows the augmented factor graph over which message passing is defined. We list all messages required for estimation. We note that messages 5 and 17, from a likelihood fragment to observed quantities, are not required for estimation: we defer their treatment to the posterior predictive calculations in Section A.1.6.

Gaussian prior to latent variable (Messages 1 and 13) The factors take the form $\mathcal{N}_n(x \mid \boldsymbol{\mu}, \Sigma)$. Directly, since we do not have to integrate over anything in (A.3),

$$m_{\mathcal{N}_n \to x}(x) = \varphi_n(x \mid \boldsymbol{\mu}, \Sigma).$$

Stochastic variable to child factor (Messages 2, 10 and 14) We directly apply (A.2) using quotients.

Affine fragment to linear predictor (Message 3) The factor in question is $\delta^X_{u,j}(Z_u, \Theta^X_{uj}) = \delta(\Theta^X_{uj} - b_{0j} - \mathbf{b}_j^\top \mathbf{z}_u)$. This message follows the "deterministic factor to child" structure, so we perform moment matching using the message from $\mathbf{z}_u$:

$$\begin{aligned}
\mathbb{E}_{\to \delta^X_{u,j}}\{\Theta^X_{uj}\} &= b_{0j} + \mathbf{b}_j^\top \boldsymbol{\mu}_{\mathbf{z}_u \to \delta^X_{u,j}} \\
\mathrm{Var}_{\to \delta^X_{u,j}}(\Theta^X_{uj}) &= \mathbf{b}_j^\top \Sigma_{\mathbf{z}_u \to \delta^X_{u,j}} \mathbf{b}_j.
\end{aligned}$$

Deterministic variable to child factor with single parent (Messages 4, 12 and 16) For these messages, we simply copy the incoming message from the variable's parent:

$$\begin{aligned}
m_{\Theta^X_{uj} \to P_j}(\Theta^X_{uj}) &= m_{\delta^X_{uj} \to \Theta^X_{uj}}(\Theta^X_{uj}) \\
m_{\mathrm{ip}_{uv} \to \delta^A_{uv}}(\mathrm{ip}_{uv}) &= m_{\delta^{\mathrm{ip}}_{uv} \to \mathrm{ip}_{uv}}(\mathrm{ip}_{uv}) \\
m_{\Theta^A_{uv} \to P_A}(\Theta^A_{uv}) &= m_{\delta^A_{uv} \to \Theta^A_{uv}}(\Theta^A_{uv}).
\end{aligned}$$

Partially observed variable to parent factor (Messages 6 and 18) There are two cases. If the variable is observed, then the message is the delta function at the observed value, so we simply replace integration with evaluation in subsequent messages. If the variable is missing, we send a unit message. Practically, we store unit messages as Gaussian with 0 precision and delta functions as Gaussian with infinite precision (which neatly matches with the Gaussian limit definition of delta functions.)

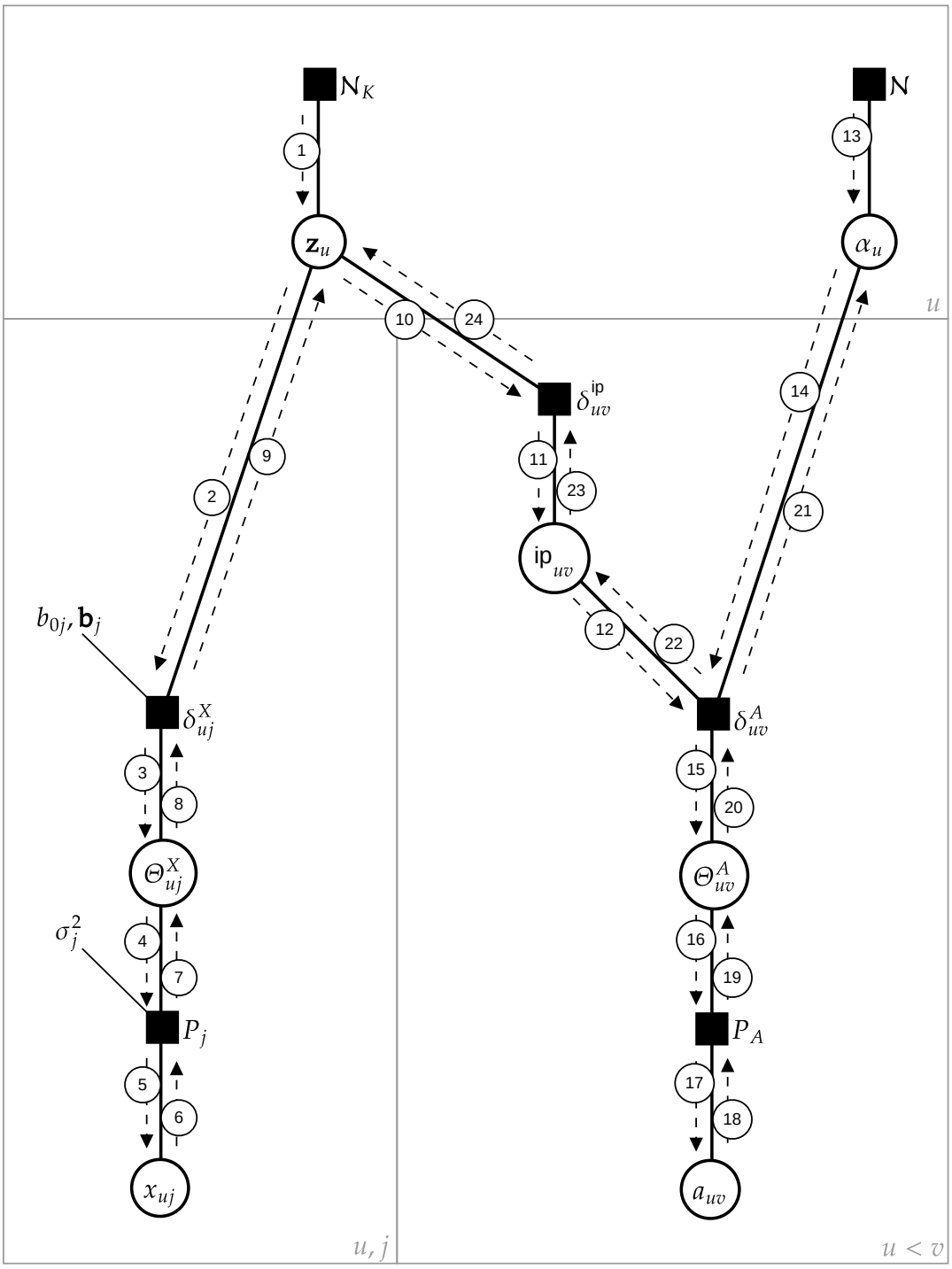

Figure A.1: Factor graph representation of the joint latent space model along with the message passing schedule.

Gaussian likelihood fragment to mean (Message 7) We consider the factor $\mathcal{N}(X \mid \Theta, \sigma^2)$ where X is observed or missing, Θ is random and σ^2 is a model parameter. When X is missing, we send a unit message to Θ in the form of a Gaussian density with precision 0 and precision-adjusted mean 0. This unit message will naturally propagate to the rest of the graph without any issues. When X is observed, we find

$$m_{\mathcal{N}\to\Theta}(\Theta) = \varphi(\Theta \mid X, \sigma^2).$$

Logistic likelihood fragment to logit (Messages 7 and 19) The factor of interest is $P(x \mid \Theta) = \sigma(\Theta)^x[1-\sigma(\Theta)]^{1-x}$ with log factor

$$\begin{aligned}\log P(x \mid \Theta) &= x \log \sigma(\Theta) + [1-x][1-\sigma(\Theta)] \\ &= x\Theta - \log(1+\exp(\Theta))\end{aligned}$$

We refer to Section A.1.4 for a discussion on obtaining Gaussian messages for the logistic-Gaussian fragment. The message we ultimately chose as default is derived from the quadratic bound proposed by Jaakkola and Jordan (2000) and is given by

$$m_{p\to\Theta}(\Theta) \propto \tilde{\varphi}(\Theta \mid x - \tfrac{1}{2}, \lambda(t)).$$

where $\lambda(t) = \frac{1}{t}(\sigma(t) - \frac{1}{2})$ and where the optimal value of t is given by $t = \sqrt{\mathbb{E}_\Theta\{\Theta^2\}}$, where the expectation is computed under the current posterior.

Deterministic variable to parent factor (Messages 8, 20 and 23) For these messages, we simply copy the incoming message from the variable's child:

$$\begin{aligned}m_{\Theta^X_{uj}\to\delta^X_{uj}}(\Theta^X_{uj}) &= m_{P_j\to\Theta^X_{uj}}(\Theta^X_{uj}) \\ m_{\Theta^A_{uv}\to\delta^A_{uv}}(\Theta^A_{uv}) &= m_{P_A\to\Theta^A_{uv}}(\Theta^A_{uv}) \\ m_{\mathrm{ip}_{uv}\to\delta^{\mathrm{ip}}_{uv}}(\mathrm{ip}_{uv}) &= m_{\delta^A_{uv}\to\mathrm{ip}_{uv}}(\mathrm{ip}_{uv})\end{aligned}$$

Affine fragment to latent positions (Message 9) The message from the Affine factor δ^X_{uj} to its parent $\mathbf{z}_u$ can be computed using (A.4):

$$\begin{aligned}\int \prod_{j\neq i} m_{j\to a}(x_j) f_a(\mathbf{x})\, \mathrm{d}\mathbf{x}_{\neg i} &= \int m_{\Theta^X_{uj}\to\delta^X_{uj}}(\Theta^X_{uj})\delta\left(\Theta^X_{uj} - b_{0j} - \mathbf{b}_j^\top \mathbf{z}_u\right)\, \mathrm{d}\Theta^X_{uj} \\ &\propto \varphi\left(b_{0j} + \mathbf{b}_j^\top \mathbf{z}_u \mid \mu_{\Theta^X_{uj}\to\delta^X_{uj}}, \sigma^2_{\Theta^X_{uj}\to\delta^X_{uj}}\right) \\ &\propto \tilde{\varphi}\left(\mathbf{z}_u \mid \gamma_{\Theta^X_{uj}\to\delta^X_{uj}}\mathbf{b}_j\mathbf{b}_j^\top, \gamma_{\Theta^X_{uj}\to\delta^X_{uj}}(\mu_{\Theta^X_{uj}\to\delta^X_{uj}} - b_{0j})\mathbf{b}_j\right)\end{aligned}$$

Since the true message is already Gaussian, there is no need to perform a KL projection. This message exemplifies the utility of the canonical parameterization since the precision matrix is only of rank one.

Inner product fragment to child (Message 11) This deterministic fragment is defined by the factor

$$\delta^{\mathrm{ip}}_{uv}(\mathbf{z}_u, \mathbf{z}_v\, \mathrm{ip}_{uv}) = \delta(\mathrm{ip}_{uv} - \mathbf{z}_u^\top \mathbf{z}_v),$$

so we apply moment matching:

$$\begin{aligned}\mathbb{E}_{\to\delta^{\mathrm{ip}}_{uv}}\{\mathrm{ip}_{uv}\} &= \boldsymbol{\mu}^\top_{\mathbf{z}_u\to\delta^{\mathrm{ip}}_{uv}}\boldsymbol{\mu}_{\mathbf{z}_v\to\delta^{\mathrm{ip}}_{uv}} \\ \mathrm{Var}_{\to\delta^{\mathrm{ip}}_{uv}}(\mathrm{ip}_{uv}) &= \mathrm{Tr}(\Sigma_{\mathbf{z}_u\to\delta^{\mathrm{ip}}_{uv}}\Sigma_{\mathbf{z}_v\to\delta^{\mathrm{ip}}_{uv}}) + \boldsymbol{\mu}^\top_{\mathbf{z}_v\to\delta^{\mathrm{ip}}_{uv}}\Sigma_{\mathbf{z}_u\to\delta^{\mathrm{ip}}_{uv}}\boldsymbol{\mu}_{\mathbf{z}_v\to\delta^{\mathrm{ip}}_{uv}} + \boldsymbol{\mu}^\top_{\mathbf{z}_u\to\delta^{\mathrm{ip}}_{uv}}\Sigma_{\mathbf{z}_v\to\delta^{\mathrm{ip}}_{uv}}\boldsymbol{\mu}_{\mathbf{z}_u\to\delta^{\mathrm{ip}}_{uv}}\end{aligned}$$

Sum fragment to child (Message 15) We consider the fragment

$$\delta^A_{uv}(\Theta^A_{uv}, \mathrm{ip}_{uv}, \alpha_u, \alpha_v) = \delta(\Theta^A_{uv} - \mathrm{ip}_{uv} - \alpha_u - \alpha_v)$$

The message to Θ^A_{uv} is given by moment matching:

$$\begin{aligned}\mathbb{E}_{\to\delta^A_{uv}}\{\Theta^A_{uv}\} &= \mu_{\mathrm{ip}_{uv}\to\delta^A_{uv}} + \mu_{\alpha_u\to\delta^A_{uv}} + \mu_{\alpha_v\to\delta^A_{uv}} \\ \mathrm{Var}_{\to\delta^A_{uv}}(\Theta^A_{uv}) &= \sigma^2_{\mathrm{ip}_{uv}\to\delta^A_{uv}} + \sigma^2_{\alpha_u\to\delta^A_{uv}} + \sigma^2_{\alpha_v\to\delta^A_{uv}}\end{aligned}$$

Sum fragment to parent (Messages 21 and 22) By symmetry, we can interchange the roles of parents and child in a sum fragment. For example, to compute the message to ip_{uv}, write the fragment as $\mathrm{ip}_{uv} = \Theta^A_{uv} - \alpha_u - \alpha_v$, and use a similar expression as in message 15, while adjusting signs in the mean.

Inner product fragment to latent positions (Message 24) This deterministic fragment is defined by the factor

$$\delta^{\text{ip}}_{uv}(\mathbf{z}_u, \mathbf{z}_v \, \text{ip}_{uv}) = \delta(\text{ip}_{uv} - \mathbf{z}_u^\top \mathbf{z}_v)$$

The message to parent $\mathbf{z}_u$ is similar to that from δ^X_{uj} to $\mathbf{z}_u$, except we now integrate $\mathbf{z}_v$ and there is no intercept:

$$\begin{aligned} m_{\delta^{\text{ip}}_{uv} \to \mathbf{z}_u}(\mathbf{z}_u) &= \int m_{\text{ip}_{uv} \to \delta^{\text{ip}}_{uv}}(\text{ip}_{uv}) m_{\mathbf{z}_v \to \delta^{\text{ip}}_{uv}}(\mathbf{z}_v) \delta(\text{ip}_{uv} - \mathbf{z}_u^\top \mathbf{z}_v) \, \mathrm{d}\, \text{ip}_{uv} \, \mathrm{d}\mathbf{z}_v \\ &= \int m_{\text{ip}_{uv} \to \delta^{\text{ip}}_{uv}}(\mathbf{z}_u^\top \mathbf{z}_v) m_{\mathbf{z}_v \to \delta^{\text{ip}}_{uv}}(\mathbf{z}_v) \, \mathrm{d}\mathbf{z}_v. \end{aligned}$$

We use a moment-matching argument to derive the update. If we naively try to compute the expectation of $\mathbf{z}_u$ under this message, we cannot simply interchange the order of integration since the term in $\mathbf{z}_u$ is a singular multivariate Gaussian:

$$m_{\text{ip}_{uv} \to \delta^{\text{ip}}_{uv}}(\mathbf{z}_u^\top \mathbf{z}_v) \propto \tilde{\varphi}\left(\mathbf{z}_u \mid \tau_{\text{ip}_{uv} \to \delta^{\text{ip}}_{uv}} \mathbf{z}_v, \gamma_{\text{ip}_{uv} \to \delta^{\text{ip}}_{uv}} \mathbf{z}_v \mathbf{z}_v^\top\right)$$

We propose a heuristic derivation, but alternative derivations arrive at the same update rules (Winn and Bishop, 2005). We proceed as if we could interchange the integration and utilize a pseudo-inverse:

$$\begin{aligned} \text{Var}(\mathbf{z}_u \mid \mathbf{z}_v) &= \left(\gamma_{\text{ip}_{uv} \to \delta^{\text{ip}}_{uv}} \mathbf{z}_v \mathbf{z}_v^\top\right)^- \\ \mathbb{E}\{\mathbf{z}_u \mid \mathbf{z}_v\} &= \left(\gamma_{\text{ip}_{uv} \to \delta^{\text{ip}}_{uv}} \mathbf{z}_v \mathbf{z}_v^\top\right)^- (\tau_{\text{ip}_{uv} \to \delta^{\text{ip}}_{uv}} \mathbf{z}_v) \end{aligned}$$

Then, we take expectation wrt $\mathbf{z}_v \sim m_{\mathbf{z}_v \to \delta^{\text{ip}}_{uv}}(\mathbf{z}_v)$ to find, using a method of moment argument,

$$\begin{aligned} \Gamma_{\delta^{\text{ip}}_{uv} \to \mathbf{z}_u} &= \gamma_{\text{ip}_{uv} \to \delta^{\text{ip}}_{uv}} \mathbb{E}_{\mathbf{z}_v \to \delta^{\text{ip}}_{uv}}\left\{\mathbf{z}_v \mathbf{z}_v^\top\right\} \\ &= \gamma_{\text{ip}_{uv} \to \delta^{\text{ip}}_{uv}} \left[\Sigma_{\mathbf{z}_v \to \delta^{\text{ip}}_{uv}} + \boldsymbol{\mu}_{\mathbf{z}_v \to \delta^{\text{ip}}_{uv}} \boldsymbol{\mu}^\top_{\mathbf{z}_v \to \delta^{\text{ip}}_{uv}}\right] \\ \boldsymbol{\tau}_{\delta^{\text{ip}}_{uv} \to \mathbf{z}_u} &= \tau_{\text{ip}_{uv} \to \delta^{\text{ip}}_{uv}} \mathbb{E}_{\mathbf{z}_v \to \delta^{\text{ip}}_{uv}}\left\{\mathbf{z}_v\right\} \\ &= \tau_{\text{ip}_{uv} \to \delta^{\text{ip}}_{uv}} \boldsymbol{\mu}_{\mathbf{z}_v \to \delta^{\text{ip}}_{uv}} \end{aligned}$$

A.1.4 Logistic fragment

For both the binary covariate model and the adjacency model, the final step of the model is a logistic model:

$$f(x, \Theta) := P(x \mid \Theta) = \sigma(\Theta)^x [1 - \sigma(\Theta)]^{1-x}$$

where X is the observation and Θ is the logit probability coming either from the affine transformation of the latent position or from the heterogeneous inner product model.

The log fragment is given by

$$\begin{aligned} \log f(x, \Theta) &= x \log \sigma(\Theta) + [1 - x] \log[1 - \sigma(\Theta)] \\ &= \log \sigma(s\Theta), \qquad s = 2x - 1, \\ &= x\Theta - \log(1 + \exp(\Theta)). \end{aligned}$$

The message to Θ can be computed using (A.3): since x is observed, expectation becomes evaluation,

$$m_{p \to \Theta}(\Theta) \propto \exp\left(\mathbb{E}_x \left\{\log p(x, \Theta)\right\}\right) = f(x, \Theta).$$

However, this message is not Gaussian which means the posterior for Θ will not be Gaussian: this non-Gaussianity will then propagate throughout the network.

There has been multiple proposals to circumvent this issue by computing a Gaussian which approximates the true message. We refer to Knowles and Minka (2011) and Nolan and Wand (2017) for reviews and comparisons.

Quadratic bound A first approach, called the *quadratic bound*, originally proposed by Jaakkola and Jordan (2000), replaces the logistic likelihood by a quadratic lower bound. The proposed bound depends on a single parameter that can be analytically optimized to give the tightest bound.

The main inequality is the following point-wise bound on the sigmoid function: for any t,

$$\sigma(x) \geqslant \tilde{\sigma}(x, t) := \sigma(t) \exp\left\{\frac{x - t}{2} - \frac{\lambda(t)}{2}(x^2 - t^2)\right\} \tag{A.6}$$

where $\lambda(t) = \frac{1}{t}(\sigma(t) - \frac{1}{2})$. The relationship with Gaussian is explicit in (A.6): indeed, the lower bound is exponential in a quadratic form of the original argument x.

Then, we use the bound (A.6) to provide a lower bound to the ELBO contribution:

$$\begin{aligned}\mathbb{E}_\Theta\{\log f(x,\Theta)\} &= \mathbb{E}_\Theta\{\log \sigma(s\Theta)\} && \text{(A.7)}\\ &\geqslant \mathbb{E}_\Theta\{\log \tilde{\sigma}(s\Theta,t)\} && \text{(A.8)}\\ &= \log\sigma(t) + \frac{s\mathbb{E}_\Theta\{\Theta\}-t}{2} - \frac{\lambda(t)}{2}(\mathbb{E}_\Theta\{\Theta^2\}-t^2), && \text{(A.9)}\end{aligned}$$

where we note that $s \in \{-1,+1\}$, so $s^2 = 1$. It can be shown that the derivative with respect to t is given by

$$\frac{\partial}{\partial t}\mathbb{E}_\Theta\{\log\tilde{\sigma}(s\Theta,t)\} = -\frac{\lambda'(t)}{2}\left(\mathbb{E}_\Theta\{\Theta^2\}-t^2\right).$$

Since the bound (A.6) is symmetric in t, we can restrict our search to $t \geqslant 0$, where $\lambda'(t) \leqslant 0$, with equality only at $t = 0$. Thus, the sign of the derivative is given by the sign of $\mathbb{E}_\Theta\{\Theta^2\}-t^2$. We thus find that the derivative increases from 0 to $\sqrt{\mathbb{E}_\Theta\{\Theta^2\}}$ and decreases afterwards, which implies that the tightest bound is attained for $t = \sqrt{\mathbb{E}_\Theta\{\Theta^2\}}$.

Now, to compute the message to Θ, we pretend that the true factor is $\tilde{\sigma}(s\Theta,t)$ and apply (A.3) directly:

$$\begin{aligned}m_{p\to\Theta}(\Theta) &\propto \exp\left(\mathbb{E}_X\left\{\log\tilde{\sigma}(s\Theta,t)\right\}\right)\\ &= \tilde{\sigma}(s\Theta,t)\\ &\propto \exp\left\{\frac{s\Theta}{2} - \frac{\lambda(t)}{2}\Theta^2\right\}\\ &\propto \tilde{\varphi}(\Theta \mid s/2, \lambda(t)).\end{aligned}$$

Tilted bound Saul and Jordan (1998) proposed an alternative lower bound to the ELBO fragment: for any a,

$$\mathbb{E}_\Theta\{\log f(x,\Theta)\} \geqslant x\mu - \frac{1}{2}a^2\sigma^2 - \log\left(1+\exp(\mu+(1-2a)\sigma^2/2)\right) \qquad \text{(A.10)}$$

where $\mu = \mathbb{E}_\Theta\{\Theta\}$ and $\sigma^2 = \mathrm{Var}_\Theta(\Theta)$. It can be shown that the tightest lower bound is attained for a satisfying

$$a = \sigma(\mu - (1-2a)\sigma^2/2),$$

which suggests a fixed-point scheme to optimize the bound. Note that a solution will be such that $a \in (0,1)$. In particular, Saul and Jordan (1998) argue that a will be an approximation to $\mathbb{E}_\Theta\{\sigma(\Theta)\}$.

At first glance, this *tilted bound* doesn't seem to improve on the original problem of computing the message to Θ. Knowles and Minka (2011) propose a method to compute the Gaussian message given any ELBO fragment provided some derivative are available. Using the tilted bound, they find

$$m_{p\to\Theta}(\Theta) \propto \tilde{\varphi}(\Theta \mid a(1-a)\mu + x - a, a(1-a)).$$

Direct NCVMP message The method proposed by Knowles and Minka (2011), called *non-conjugate variatonal message passing* (NCVMP), can also be used directly on the true ELBO fragment $\mathbb{E}_\Theta\{\log p(x,\Theta)\}$. The message to Θ is given by

$$m_{p\to\Theta}(\Theta) \propto \tilde{\varphi}(\Theta \mid \tau, \gamma).$$

where

$$\gamma = \frac{\mathbb{E}_\Theta\{\Theta\sigma(\Theta)\} - \mu\mathbb{E}_\Theta\{\sigma(\Theta)\}}{\sigma^2}$$
$$\tau = \mu\gamma + x - \mathbb{E}_\Theta\{\sigma(\Theta)\}.$$

Knowles and Minka (2011) propose to approximate the integrals $\mathbb{E}_\Theta\{\Theta\sigma(\Theta)\}$ and $\mathbb{E}_\Theta\{\sigma(\Theta)\}$ using a Gaussian quadrature; Nolan and Wand (2017) rather propose to use a mixture of Gaussian CDFs to approximate the sigmoid function (Monahan and Stefanski, 1989, *normal scale mixture*). The normal scale mixture approach comes with better guarantees at the cost of computation time: indeed, it requires multiple evaluations of Gaussian CDFs compared to multiple sigmoid evaluation for the quadrature approach. Still, this increased computational cost is negligible compared to other message computation.

Comparison Experimentations will all four messages–quadratic bound, tilted bound with NCVMP, direct NCVMP with quadrature and with normal scale mixture–did not reveal much differences, especially upon convergence. While individual messages are indeed more accurate using the various NCVMP messages, we found very little differences in the ELBO and in prediction metrics after convergence.

We did encounter similar behaviour reported in Nolan and Wand (2017) where the quadratic bound was more stable (Knowles and Minka, 2011 also report that NCVMP can lead to oscillation.) We also experimented with damping, as suggested in Knowles and Minka (2011) and found that it can alleviate convergence issues, but it requires additional tuning of the damping factor.

Hence, for computation cost, convenience and stability concerns, we decided to use the

quadratic bound messages.

A.1.5 M step

The update of model parameters will be given by the mode of the approximate posterior for that parameter. In the message passing scheme, this amounts to sending messages to the parameters, and computing the mode of the product of all incoming messages (Dauwels et al., 2009).

A.1.5.1 Update of Gaussian likelihood variance

To update σ_j^2, we note that the parameter occurs in the likelihood of all observed attribute under a Gaussian model. This corresponds to the factors

$$f(x_{uj}, \Theta_{uj}^X, \sigma_j^2) = \varphi(x_{uj} \mid \Theta_{uj}^X, \sigma_j^2), \qquad u \in \mathcal{O}_j.$$

The message from f to σ_j^2 is given by

$$m_{f \to \sigma_j^2}(\sigma_j^2) \propto \exp \mathbb{E}_q \left\{ \log f(x_{uj}, \Theta_{uj}^X, \sigma_j^2) \right\}$$

For maximization, we only need to maximize the logarithm of the message product, that is, the sum of the log messages:

$$\sigma_j^2 \leftarrow \arg\max_{\sigma^2} \sum_{u \in \mathcal{O}_j} \mathbb{E}_q \left\{ \log f(x_{uj}, \Theta_{uj}^X, \sigma^2) \right\}.$$

The log messages are given by

$$\begin{aligned} \log f(x_{uj}, \Theta_{uj}^X, \sigma^2) &= \log \varphi(x_{uj} \mid \Theta_{uj}^X, \sigma^2) \\ &= -\frac{1}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \left(x_{uj} - \Theta_{uj}^X \right)^2. \end{aligned}$$

We note that the above expression only involves the random quantity Θ_{uj}^X, so expectation can be taken with respect to the current approximate posterior $q(\Theta_{uj}^X) = \varphi(\Theta_{uj}^X \mid \mu_{\Theta_{uj}^X}, \sigma_{\Theta_{uj}^X}^2)$. Hence,

$$\begin{aligned} \mathbb{E}_q \left\{ \log f(x_{uj}, \Theta_{uj}^X, \sigma^2) \right\} &= -\frac{1}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \mathbb{E}_q \left\{ \left(x_{uj} - \Theta_{uj}^X \right)^2 \right\} \\ &= -\frac{1}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \left((x_{uj} - \mu_{\Theta_{uj}^X})^2 + \sigma_{\Theta_{uj}^X}^2 \right) \end{aligned}$$

We thus find

$$\begin{aligned}\sigma_j^2 &\leftarrow \arg\max_{\sigma^2} \sum_{u \in \mathcal{O}_j} \left[-\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left((x_{uj} - \mu_{\Theta_{uj}^X})^2 + \sigma^2_{\Theta_{uj}^X} \right) \right] \\ &= \frac{1}{|\mathcal{O}_j|} \sum_{u \in \mathcal{O}_j} (x_{uj} - \mu_{\Theta_{uj}^X})^2 + \sigma^2_{\Theta_{uj}^X}\end{aligned}$$

We note that we did not really rely on the message-passing framework for this update: all the steps above are exactly the same if we rather directly optimize the ELBO at the current approximate posterior, as would be done in VEM. The update for the affine transformations parameters will benefit from the message passing approach.

A.1.5.2 Update of affine transformations biases

To update B_{0j}, we note that it appears for every observed jth attribute in the deterministic factor

$$\delta_{uj}^X(\mathbf{z}_u, \Theta_{uj}^X, b_{0j}, \mathbf{b}_j) = \delta(\Theta_{uj}^X - b_{0j} - \mathbf{b}_j^\top \mathbf{z}_u), \qquad u \in \mathcal{O}_j$$

where we now make the dependence on the model parameters explicit. With respect to b_{0j}, we can write it as the sum constraint $b_{0j} = \Theta_{uj}^X - \mathrm{ip}_{uj}^X$, where we introduce an intermediate representation $\mathrm{ip}_{uj}^X = \mathbf{b}_j^\top \mathbf{z}_u$, similarly to the heterogeneity latent variables in the inner product model. Hence, we can use the updates from message 15: we find a Gaussian message with moments

$$\begin{aligned}\mu_{\delta_{uj}^X \to b_{0j}} &= \mu_{\Theta_{uj}^X \to \delta_{uj}^X} - b_j^\top \boldsymbol{\mu}_{\mathbf{z}_u \to \delta_{uj}^X} \\ \sigma^2_{\delta_{uj}^X \to b_{0j}} &= \sigma^2_{\Theta_{uj}^X \to \delta_{uj}^X} + \mathbf{b}_j^\top \Sigma_{\delta_{uj}^X \to b_{0j}} \mathbf{b}_j\end{aligned}$$

Then, we aggregate the messages incoming to b_{0j} by summing the natural parameters and extract the mean.

A.1.5.3 Update of affine transformations weights

To update $\mathbf{b}_j$, we need to compute the message from δ_{uj}^X to $\mathbf{b}_j$. We note the close resemblance with message 24 from the inner products to a parent. However, since we did not track the intermediary representation ip_{uv}^X, we cannot use the same update directly. Still, we can write

$$\mathbf{b}_j^\top \mathbf{z}_u = \Theta_{uj}^X - b_{0j} = \mathrm{ip}_{uj}^X$$

and note that the incoming message from ip_{uj}^X is Gaussian with

$$\begin{aligned}\mu_{\text{ip}_{uj}^X \to \delta_{uj}^X} &= \mu_{\Theta_{uj}^X \to \delta_{uj}^X} - b_{0j} \\ \sigma^2_{\text{ip}_{uj}^X \to \delta_{uj}^X} &= \sigma^2_{\Theta_{uj}^X \to \delta_{uj}^X}.\end{aligned}$$

Hence, the message to $\mathbf{b}_j$ is Gaussian with canonical parameters

$$\begin{aligned}\Gamma_{\delta_{uj}^X \to \mathbf{b}_j} &= \gamma_{\Theta_{uj}^X \to \delta_{uj}^X} \left[\Sigma_{\mathbf{z}_u \to \delta_{uj}^X} + \boldsymbol{\mu}_{\mathbf{z}_u \to \delta_{uj}^X} \boldsymbol{\mu}^\top_{\mathbf{z}_u \to \delta_{uj}^X}\right] \\ \boldsymbol{\tau}_{\delta_{uj}^X \to \mathbf{b}_j} &= \tau_{\Theta_{uj}^X \to \delta_{uj}^X} \boldsymbol{\mu}_{\mathbf{z}_u \to \delta_{uj}^X} \\ &= \gamma_{\Theta_{uj}^X \to \delta_{uj}^X} \left[\mu_{\Theta_{uj}^X \to \delta_{uj}^X} - b_{0j}\right] \boldsymbol{\mu}_{Z\mathbf{z}_u \to \delta_{uj}^X}.\end{aligned}$$

The update is then obtained by aggregating all message and extracting the mean.

A.1.6 Posterior predictive distributions

Given an approximate posterior for $\Theta_{uj}^X \sim \mathcal{N}(\mu_{\Theta_{uj}^X}, \sigma^2_{\Theta_{uj}^X})$ and estimated model parameters, we are interested in computing the (approximate) predictive distribution for the missing value x_{uj}. When x_{uj} emerges from a Gaussian model, we simply add the covariate noise σ_j^2 to the posterior,

$$\begin{aligned}q\left(x_{uj} \mid \mathbf{A}, \mathbf{X}_{\mathcal{O}}\right) &= \int \varphi\left(x_{uj} \mid \Theta_{uj}^X, \sigma_j^2\right) \varphi\left(\Theta_{uj}^X \mid \mu_{\Theta_{uj}^X}, \sigma^2_{\Theta_{uj}^X}\right) \, \mathrm{d}\Theta_{uj}^X \\ &= \varphi\left(x_{uj} \mid \mu_{\Theta_{uj}^X}, \sigma^2_{\Theta_{uj}^X} + \sigma_j^2\right).\end{aligned}$$

When x_{uj} is binary, its predictive distribution is given by the (approximate) posterior probability of $x_{uj} = 1$,

$$\begin{aligned}Q(x_{uj} = 1 \mid \mathbf{A}, \mathbf{X}_{\mathcal{O}}) &= \int \sigma\left(\Theta_{uj}^X\right) \varphi\left(\Theta_{uj}^X \mid \mu_{\Theta_{uj}^X}, \sigma^2_{\Theta_{uj}^X}\right) \, \mathrm{d}\Theta_{uj}^X \\ &= \sigma_{\Theta_{uj}^X} \int \sigma\left(\mu_{\Theta_{uj}^X} + \sigma_{\Theta_{uj}^X} z\right) \varphi\left(z\right) \, \mathrm{d}z. \qquad \text{(A.11)}\end{aligned}$$

Unfortunately, there is no closed form solution to (A.11): we resort to the normal-scale mixture utilized by Nolan and Wand (2017) in a similar setting. Specifically, Monahan and Stefanski (1989) approximates the sigmoid function $\sigma(\cdot)$ by a mixture of k Gaussian CDFs, $\sigma(z) \approx \sum_{i=1}^k p_{k,i} \Phi(s_{k,i} z)$, for some known values of $p_{k,i}$ and $s_{k,i}$ optimizing the approximation

error. Then, we get that the approximations of the integrals,

$$\int \sigma\left(m + vz\right) \varphi\left(z\right) \, \mathrm{d}z \approx \sum_{i=1}^{k} p_{k,i} \Phi\left(\frac{m s_{k,i}}{\sqrt{1 + s_{k,i}^2 v^2}}\right),$$

have bounded error, uniformly in (m, v^2) and with bound depending only on the number of components k. For example, $k = 8$ produces an error bound of the order of 10^{-9} (Monahan and Stefanski, 1989).

A.1.7 ELBO calculations

The ELBO takes the following form:

$$\begin{aligned}\mathrm{ELBO}_\phi(q) = &\sum_{(u,j)\in\mathcal{O}} \mathbb{E}_q\{\log P(x_{uj} \mid \mathbf{z}_u, \phi)\} + \sum_{u<v} \mathbb{E}_q\{\log P(a_{uv} \mid \mathbf{z}_u, \mathbf{z}_v, \alpha_u, \alpha_v)\} \\ &- \sum_u \mathrm{KL}(q(\mathbf{z}_u) \| p_0(\mathbf{z}_u)) - \sum_u \mathrm{KL}(q(\alpha_u) \| p_0(\alpha_u)).\end{aligned}$$

The two KL terms can be directly computed from the KL between two Gaussian densities. Similarly, for continuous covariates, the ELBO contribution is a Gaussian cross-entropy term. The main difficulty occur for logistic fragments, arising from the adjacency likelihood or binary covariates. The discussion in Section A.1.4 suggests multiple options to compute Gaussian-logistic integrals. The quadratic bound (A.9) and the tilted bound (A.10) both provide a lower bound to the ELBO contribution, so we can use these bounds directly. Alternatively, we can use a quadrature to approximate the integral directly; however, the approximate ELBO is no longer guaranteed to be a lower bound for the model evidence. We monitor convergence using the quadrature method, though experimentation indicate that the two other bounds yield similar early stopping decisions and predictive performance.

A.1.8 Model selection

The main quantity to tune is the number of latent dimensions K. Larger values of K can improve adjustment to training data, but overfitting may be a concern (see Section A.4 for some experiments).

The ELBO provides a natural quantity to perform model selection: maximizing the ELBO amounts to balancing the positive and negative terms: we can increase the ELBO by increasing adjustment to data (expected log-likelihood) or by decreasing the KL between the approximate posterior and the prior. In particular, the KL term for $\mathbf{Z}$ scales with NK and

be interpreted as a penalty on complexity. We have that

$$\begin{aligned}\mathrm{KL}(q(\mathbf{z}_u)\|p_0(\mathbf{z}_u)) &= \mathrm{KL}(\varphi_K(\cdot \mid \boldsymbol{\mu}_{\mathbf{z}_u}, \Sigma_{\mathbf{z}_u})\|\varphi_K(\cdot \mid \mathbf{0}, \mathbf{I})) \\ &= \frac{1}{2}\left[\log \frac{\det \mathbf{I}}{\det \Sigma_{\mathbf{z}_u}} - K + \mathrm{Tr}(\Sigma_{\mathbf{z}_u}) + \boldsymbol{\mu}_{\mathbf{z}_u}^\top \boldsymbol{\mu}_{\mathbf{z}_u}\right] \\ &= \frac{1}{2}\left[\log\det \Sigma_{\mathbf{z}_u}^{-1} - K + \mathrm{Tr}(\Sigma_{\mathbf{z}_u}) + \boldsymbol{\mu}_{\mathbf{z}_u}^\top \boldsymbol{\mu}_{\mathbf{z}_u}\right]\end{aligned}$$

To get a sense of this term, we provide a heuristic asymptotic argument. The posterior for $\mathbf{z}_u$ is dominated by the $\mathcal{O}(N+p)$ messages coming from the edges and attributes. In particular, $\Sigma_{\mathbf{z}_u}^{-1}$ should grow at the same rate, meaning that the log determinant should grow with $K\log(N+p)$. Then, $\Sigma_{\mathbf{z}_u}$ should go to 0 (posterior contraction) with $N+p$, so the trace should go to 0 with $N+p$. Because of the prior, $\boldsymbol{\mu}_{\mathbf{z}_u}$ should not grow with $N+p$, but its squared norm should still grow with K. This means the KL term will be dominated by $\mathcal{O}(K\log(N+p)+K)$, which are essentially the AIC and BIC penalties for a vector of size K with $N+p$ observations. Instead of using the asymptotic penalties, we use the KL term directly, since they are readily available.

Unfortunately, since we use point estimates for global parameters, they do not appear in the ELBO. Now, the M-step already requires the computation of messages to global parameters, so we can compute their approximate posterior. This allows us to define an extended ELBO which penalizes complexity due to model parameters. In particular, we are mostly interested in $\mathbf{B} \in \mathbb{R}^{K\times p}$. Since we do not impose a prior on $\mathbf{B}$, we are only left with the negative entropy term,

$$\mathrm{KL}(q(\mathbf{b}_j)\|p_0(\mathbf{b}_j)) = -\mathcal{H}(q(\mathbf{b}_j)) = -\frac{1}{2}\left[\log\det(2\pi\Sigma_{\mathbf{b}_j}) - K\right]$$

Using a similar argument as above, we find that this term asymptotically behaves as $\mathcal{O}(K\log(p)+K)$, which is again akin to AIC and BIC penalties for a vector of length K with p observations. Adding this term to the ELBO produces a stronger penalty on K along p; experimental results in Section A.4 show that it helps model selection when p is large. Again, this comes at no cost since we already compute the messages to B_j during the M-step.

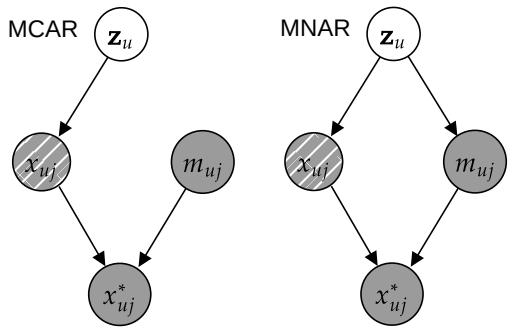

Figure A.2: Bayes net representation of the missingness model under MCAR and MNAR assumptions. White nodes represent unobserved quantities; grey nodes represent observed quantities; hashed nodes represent partially observed quantities.

A.2 Non-ignorable missingness model

Rubin (1976) argues that "statistician[s] should explicitly consider the process that causes missing data far more often." This statement is especially relevant to our context given the motivating example of social networks with user-provided node attributes which are subject to self-censoring. For this reason, we weaken the presumably false MCAR assumptions by directly modeling the missingness mechanism.

The spirit of the joint model, understood as a factor model, is that the latent variables $(\mathbf{z}_u, \alpha_u)$ capture all the systematic signal from a single node u: the adjacency, the covariates and whether we observe covariates or not. Thus, we define a missingness model for m_{uj} that is independent of the rest of the graphical model conditionally on $\mathbf{z}_u$. Then, as x^*_{uj} depends only on m_{uj} and x_{uj}, this defines a model for the observations $\mathbf{X}^*$ with independent entries x^*_{uj} conditionally on $\mathbf{Z}$. Since m_{uj} takes binary values, we simply treat it as a fully observed binary attribute, that is, we model it with the logistic model

$$m_{uj} \mid \Theta^M_{uj} \sim \text{Bernoulli}\left(\sigma\left(\Theta^M_{uj}\right)\right), \qquad \Theta^M_{uj} = c_{0j} + \mathbf{c}_j^\top \mathbf{z}_u,$$

where $\mathbf{c}_0 \in \mathbb{R}^p$ and $\mathbf{C} \in K \times p$ are respectively the bias and weights of the logistic regression model. This model allows capturing various missingness mechanisms. For example, if large values of x_{uj} are associated with a non-response, we should find that the corresponding Θ^X_{uj} and Θ^M_{uj} are positively correlated. Similarly, some regions of the latent space may be associated with non-response, in which case all $\mathbf{c}_j$'s would point toward these regions (i.e., have large inner products).

While we model an attribute and its missingness indicators independently conditionally

on the latent position $\mathbf{z}_u$, integrating out the latent variable reveals the dependence between these two quantities. The marginal distribution of the linear predictors is given by

$$\begin{pmatrix} \Theta_u^X \\ \Theta_u^M \end{pmatrix} \sim \mathcal{N}_{2p}\left(\begin{pmatrix} \mathbf{b}_0 \\ \mathbf{c}_0 \end{pmatrix}, \sigma_Z^2 \begin{pmatrix} \mathbf{B}^\top\mathbf{B} & \mathbf{B}^\top\mathbf{C} \\ \mathbf{C}^\top\mathbf{B} & \mathbf{C}^\top\mathbf{C} \end{pmatrix} \right).$$

We can learn about the correlation between an attribute value and it missingness by inspecting the diagonal elements in $\mathbf{B}^\top\mathbf{C}$ and about the missing patterns (e.g., co-occurrence) by inspecting the off-diagonal elements in $\mathbf{C}^\top\mathbf{C}$. For example, $\mathbf{c}_j = \mathbf{0}$, i.e., a MCAR assumption for the j-th attribute, leads to no correlation with any attribute ($\mathbf{c}_j^\top\mathbf{B} = \mathbf{0}$) nor with other missingness variables ($\mathbf{c}_j^\top\mathbf{C} = \mathbf{0}$).

The observed covariate model takes the form

$$P\left(\mathbf{X}^* \mid \mathbf{Z}\right) = \prod_{u=1}^{N}\prod_{j=1}^{p} P\left(x_{uj}^* \mid \mathbf{z}_u\right),$$

where $P(x_{uj}^* \mid \mathbf{z}_u)$ is a mixture of the observed component $P(x_{uj} \mid \mathbf{z}_u)$ and the point mass at $x_{uj}^* = \texttt{NA}$:

$$P\left(x_{uj}^* \mid \mathbf{z}_u\right) = \begin{cases} \mathbb{P}[m_{uj} = 1 \mid \mathbf{z}_u], & x_{uj}^* = \texttt{NA} \\ P(x_{uj} \mid \mathbf{z}_u)\mathbb{P}[m_{uj} = 0 \mid \mathbf{z}_u], & x_{uj}^* \neq \texttt{NA}. \end{cases}$$

This proposed missingness model is neither MCAR nor MAR because of the direct interdependence between the attribute x_{uj} and its missingness indicator m_{uj} through the latent position $\mathbf{z}_u$ (see Figure A.2).

Essentially, the proposed missingness model can be understood as adding p fully observed binary attributes m_{uj} to the set of covariates. This is enabled by the simplicity of the model, induced by the conditional independence on the latent positions $\mathbf{z}_u$. In particular, the only difference $\mathbf{M}$ and binary covariates of $\mathbf{X}$ is that $\mathbf{M}$ is fully observed.

A.3 Implementation details

A.3.1 NAIVI

We implement the variational message passing approach using `PyTorch` (Paszke et al., 2019). Our implementation rely on three main objects: *factors*, *variables* and *messages*. Factors contain references to the parent and child variables, and the messages between them, as well as the updating rules of inbound and outbound messages and the ELBO contribution. Variables contains references to their neighboring factors and their current posterior approximation. Messages link factors to variables and contain the messages in either directions. When a message to a variable is update, it triggers an update of the posterior using the quotient formula (A.5). We compute the ELBO at every iteration and stop iterating when the relative change is below some threshold (10^{-5}). Convergence is typically reached within the first 100 iterations.

Initialization We initialize all messages to unit messages represented by a Gaussian density with 0 precision. To break the symmetry, we initialize the entries of the weight matrix $\mathbf{B}$ to iid standard normal variates and we randomly initialize the messages from inner products to latent positions (messages 24) to a Gaussian density with iid standard normal variate as the mean. Only one of these two randomizations is really required, but we perform both to deal with cases where there are no network or no attributes. Then, we compute the current posterior approximation as the product of all inbound messages to each variable.

Complexity and memory Compared to its MLE counterpart (Zhang et al., 2022), `NAIVI` has prohibitive memory and computation requirements. In particular, the most memory-intensive messages are to the latent variables: there are $\mathcal{O}(N(N+p))$ of them, each with $\mathcal{O}(K^2)$ values. MLE have a much smaller memory footprint of $\mathcal{O}(N(N+p+K))$. For example, the Cora dataset experiment ($N = 2708$, $p = 1433$) would not fit in memory on a double-precision 16GB GPU for $K > 5$, but did on a single-precision 48GB GPU up to $K = 10$. The main computational bottleneck is the calculation of messages from latent positions. For each $\mathcal{O}(N^2)$ node pairs and $\mathcal{O}(Np)$ attributes we need to invert a $K \times K$ matrix, indicating a complexity of $\mathcal{O}(N(N+p)K^3)$, whereas gradient computation scales with $\mathcal{O}(N(N+p)K)$. Now, when N increases, posterior contraction kicks in, and we find little difference between the Bayesian and frequentist approaches (Figure 1, Setting A). Hence, for large N, we suggest reverting to point estimation. Still, for very large N, the $\mathcal{O}(N^2)$ pairs are prohibitive for any method modeling all edges and subsampling or amortized inference is warranted.

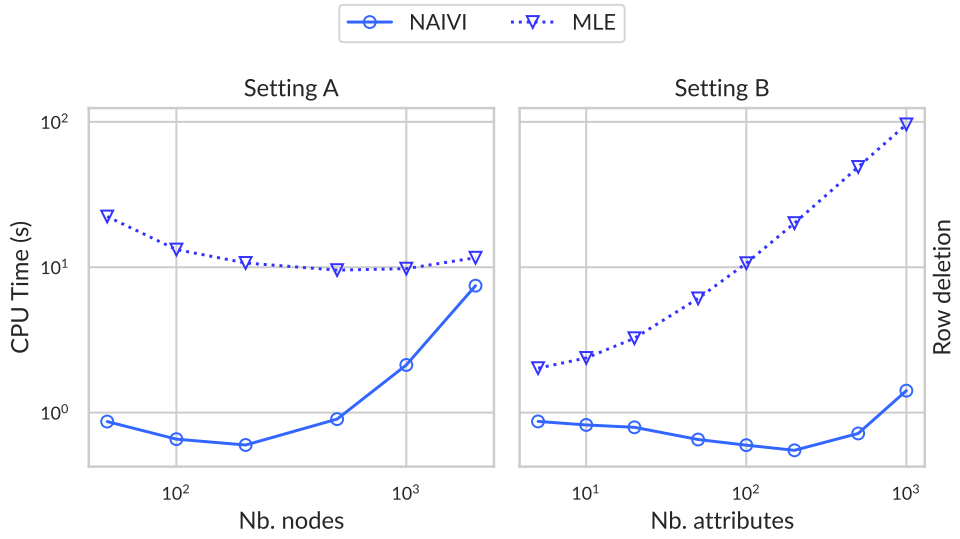

Figure A.3: Computation time comparison between variational inference and point estimation: `row deletion` case, settings A and B of the synthetic experiment of Section 4.1 in the main text, reported as the median across 30 replications.

In Figure A.3, we report the computing time of the synthetic experiments in Section 4.1 of the main text. We find that for larger networks (2,000 nodes) and large number of attributes (1,000), `NAIVI` reaches convergence in just a few seconds. Even though each iteration of MAP or MLE are much faster, convergences is also much slower; tracking the uncertainty around each quantity can be understood as a second order method, which explains the much faster convergence. [1]

A.3.2 Competing methods

Oracle When simulating data from our model, we can use the true generating means as our prediction: Θ^X_{uj} for continuous attributes and $\sigma(\Theta^X_{uj})$ for binary attributes.

GLFM We use the same implementation as `NAIVI` except that we drop the network model. Everything else remains identical.

MAP & PMLE We use automatic differentiation in `PyTorch` (Paszke et al., 2019) to obtain gradients and use the Rprop optimizer (Riedmiller and Braun, 1993) to perform updates of local and global parameters. For the projected MLE, we project the latent position after the gradient updates so that $\mathbf{Z}$ has column sums of 0, as in Ma et al. (2020);

[1] That being said, we do not claim to have a particularly optimized implementation of frequentist methods.

Zhang et al. (2022). We initialize latent positions using the singular value thresholding proposed in Ma et al. (2020); the regression parameters are then initialized to the estimates from the respective GLMs setting the latent positions as fixed covariates.

MICE & KNN We use the *scikit-learn* implementations (Pedregosa et al., 2011, `IterativeImputer` and `KNNImputer`.) The conditional models in `MICE` are Bayesian linear regressions with random imputation order and we truncate prediction to the observed support. We use 10 neighbors in `KNN`; all other parameters are set to their default values.

Smooth We start by imputing columnwise means in our attribute matrix $\mathbf{X}$ to produce $\mathbf{X}^{(0)}$. Then, until convergence of $\mathbf{X}^{(t)}$ we update the missing entries by computing the average over neighbors. For $\widetilde{\mathbf{A}}$, the row-standardized adjacency matrix, i.e., $\widetilde{\mathbf{A}} = \text{diag}^{-1}(\mathbf{A1})\mathbf{A}$, we compute

$$\mathbf{X}^{(t+1)} = \widetilde{\mathbf{A}}\mathbf{X}^{(t)}$$

and then replace the observed entries by their observed values in $\mathbf{X}$. Convergence is typically reached in just a handful of iterations.

GCN We consider the 2-layer architecture suggested in Kipf and Welling (2017) with 16 hidden nodes, ReLU activation and dropout of 50%. We adapt a PyTorch implementation provided by one of the authors (`https://github.com/tkipf/pygcn/`) and use the Adam optimizer (Kingma and Ba, 2014) with weight decay for gradient updates.

A.4 Additional experiments

A.4.1 Model selection

In a first set of experiments, we generate data from our model similarly to the settings in Section 4.1 In particular, we generate latent dimensions from $K = 3$ or $K = 7$ latent dimensions and estimate the model with K between 2 and 10. We repeat the experiment for various combinations of binary attributes p and number of nodes N. We consider the `Triangle` missing data mechanism with 50% missing values. We evaluate prediction accuracy using the AuROC averaged over all p predictors evaluated at the missing values. We repeat the experiment 30 times and plot the ELBO, the extended ELBO described in Section A.1.8 and AuROC as the relative difference to the best K within each seed. Results can be found in Figure A.4.

The ELBO curve show a clear "elbow" pattern right around the true generating value for K, indicating that the ELBO can be used directly for model selection. In particular, the best ELBO value is always attained at the true value of K, except when p is on the order of N; still, in that case, we find a sharp change in the curve, suggesting that visual inspection can be used to select K. The extended ELBO improves on the ELBO when p is large and appears to be a better criteria overall. Then, the predictive performance seems to remain good even beyond the true value of K, indicating that our method is fairly robust against overfitting. However, when p is on the order of N, we do find overfitting issues for K beyond its true value, indicating that model selection is more important for these cases.

Facebook Section 4.2 ... Figure A.5

For the two semi-supervised learning experiments of Section 4.3, we repeat the estimation and prediction process for varying latent dimension K. Results can be found in Figure A.6. For the Email dataset, the ELBO indicate that K between 3 and 6 should be selected, uniformly along the number of seeds per department. The extended ELBO, which penalized K more with the $p = 42$ classes, point towards K between 3 and 5. Previous experiments suggest to select K as large as possible, so we choose $K = 5$ for the main results. Looking at the F1 score of the predictions, we find prediction performance flattens after $K = 4$, with some small gains available beyond. For the Cora dataset, the ELBO suggests K between 3 and 6 across number of seeds per topic; the extended ELBO suggest a smaller K, between 2 and 3. However, the extended ELBO is less appropriate in this case since it is computed for all $p = 1433 + 7$ attributes, while only 7 attributes are of interest for prediction. We settled on using $K = 5$ for the main results. The F1 score is essentially flat beyond $K = 4$ or $K = 5$.

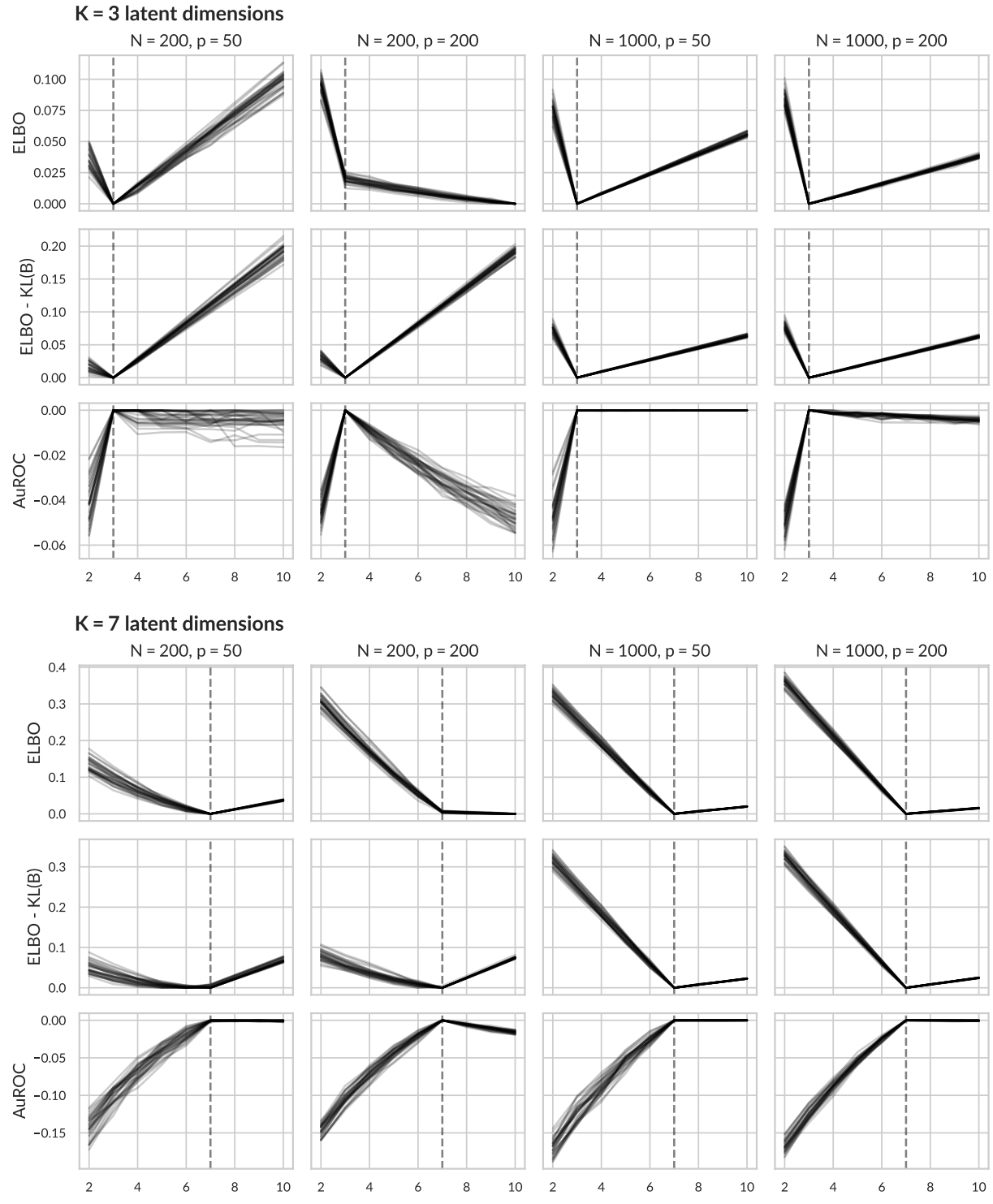

Figure A.4: Model selection metrics across fitted latent dimension in simulated data: ELBO, extended ELBO (see Section A.1.8) and prediction AuROC for 30 data generations for $K = 3$ and $K = 7$ generating latent dimensions. All metrics are displayed as the relative difference to the best value in each curve.

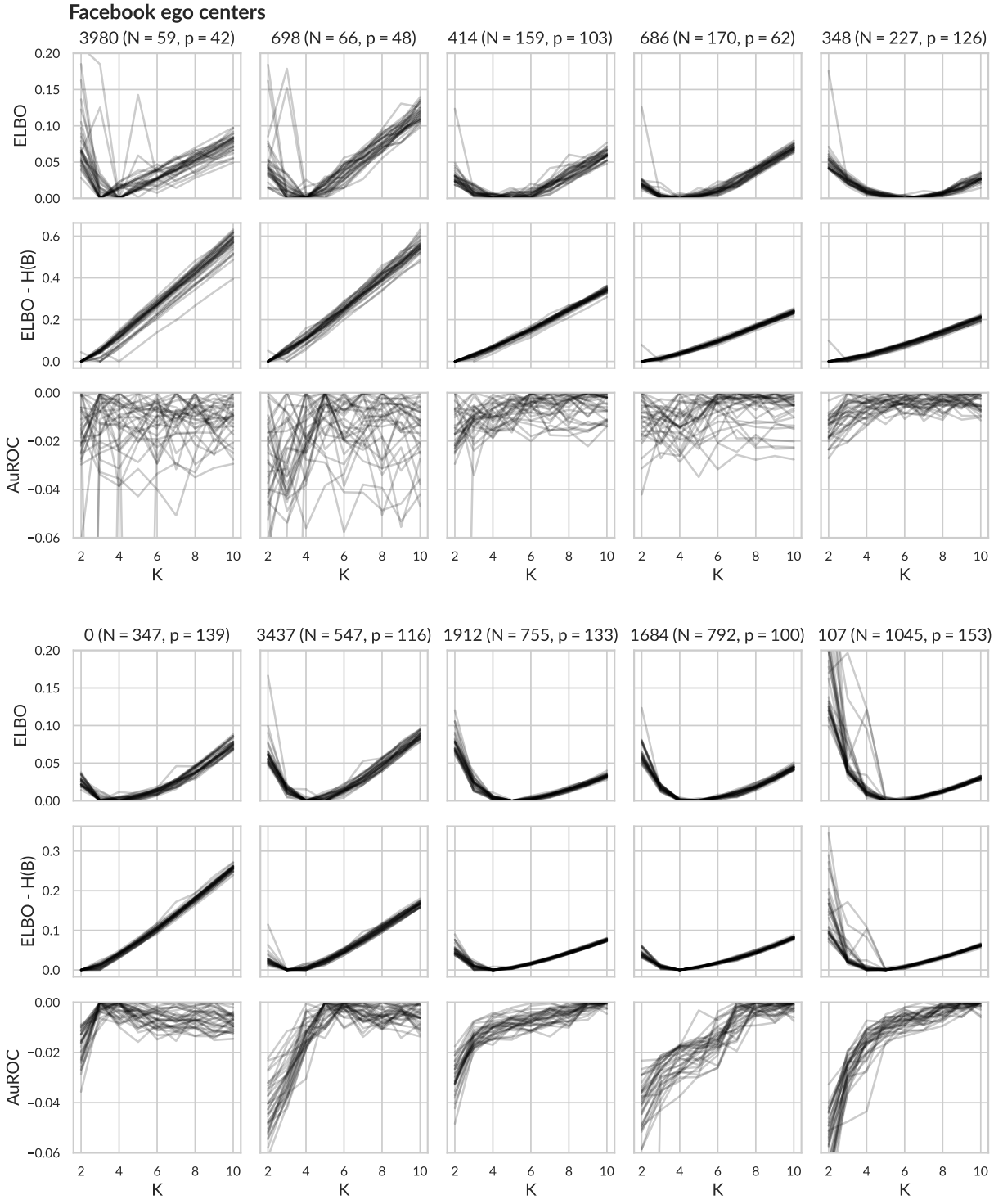

Figure A.5: Model selection metrics across fitted latent dimension for the Facebook ego centers: ELBO, extended ELBO (see Section A.1.8) and prediction AuROC for 30 seed samplings. All metrics are displayed as the relative difference to the best value in each curve.

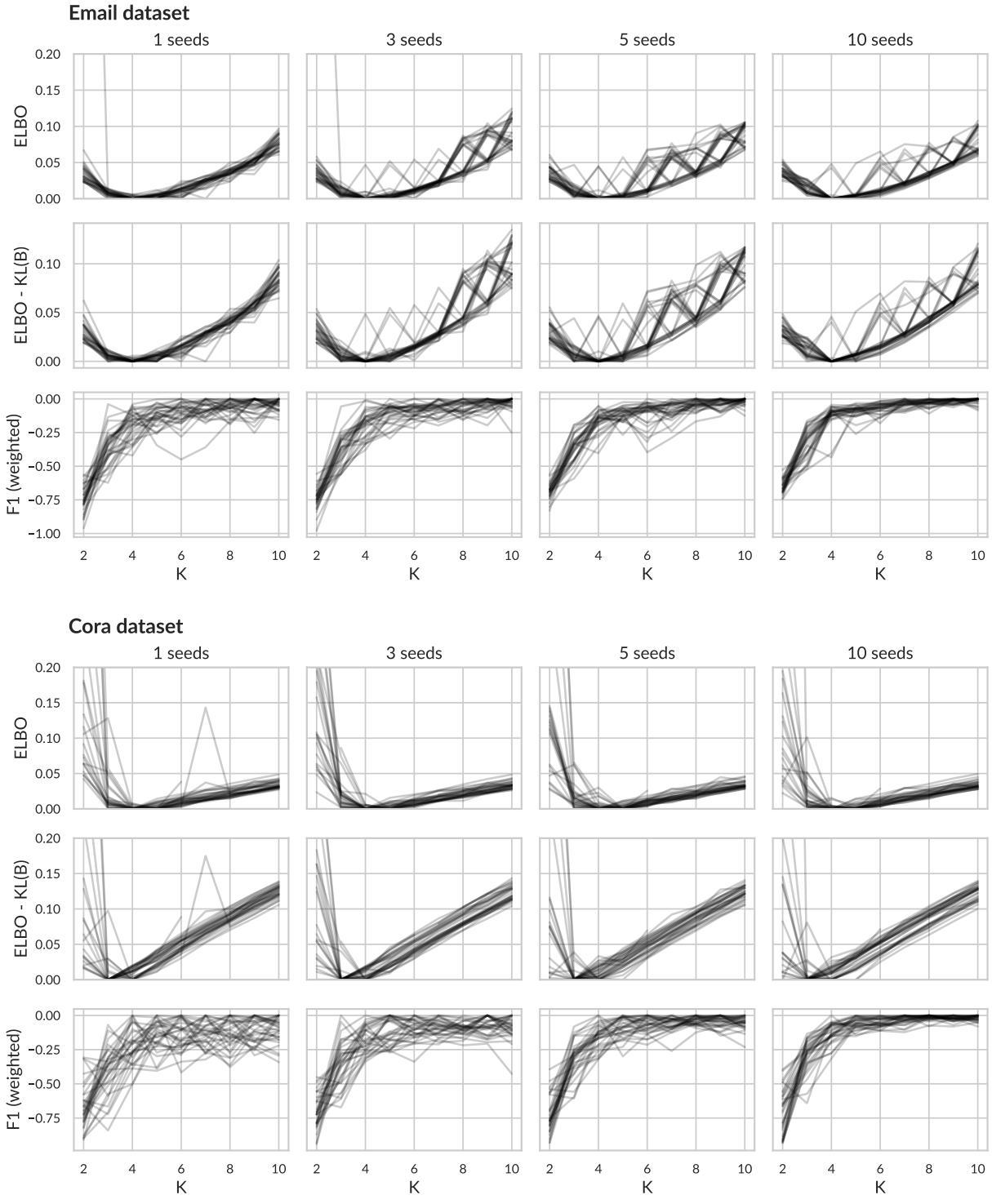

Figure A.6: Model selection metrics across fitted latent dimension for the Email and Cora datasets: ELBO, extended ELBO (see Section A.1.8) and prediction AuROC for 30 seed samplings varying number of seeds per class. All metrics are displayed as the relative difference to the best value in each curve.

APPENDIX B

Supplementary Materials to Chapter 3

B.1 Additional estimation details

B.1.1 Mean parameter updates

B.1.1.1 Constant term

Let $\mathbf{U}_i$ be the $n_i \times p_u$ matrix with rows $\mathbf{u}_i(t_{ij})$. The estimating function for $\boldsymbol{\alpha}$ is given by

$$U_{\boldsymbol{\alpha}} = -\sum_{i=1}^{N} \mathbf{U}_i^\top \mathbf{P}_i \mathbf{r}_i.$$

B.1.1.2 Proximal details

The hierarchical structure of the sparse group Lasso penalty enables its proximal to be computed by composing the two proximal operators emerging from the two penalties (Jenatton et al., 2010) using the tree ordering. Let $S_1(x, a) = \text{sgn}(x)(|x| - a)_+$ denote the soft thresholding operator with element-wise comprehension $[S_1(\mathbf{x}, \mathbf{a})]^{(s)} = S_1(x^{(s)}, a^{(s)})$, corresponding to the proximal operator for the Lasso penalty $a\|\mathbf{x}\|_1$. Let $S_2(\mathbf{x}, a) = (1 - a/\|\mathbf{x}\|_2)_+ \mathbf{x}$ denote the shrinkage operator corresponding to the proximal operator of the group Lasso penalty $a\|\mathbf{x}\|_2$. Then, the sparse group Lasso proximal operator is given by

$$\text{prox}_{\eta P_{\lambda,\alpha(\cdot;\Omega)}}(\mathbf{b}_j^*) = S_2\left(S_1\left(\mathbf{b}_j^*, \eta\lambda\alpha\boldsymbol{\omega}_j\right), \eta\lambda(1-\alpha)\sqrt{S}\omega_j\right),$$

where $\boldsymbol{\omega}_j = (\omega_j^{(s)}, s = 1, \ldots, S)$.

B.1.1.3 Boundary considerations

One notable concern about the penalized estimating equations is that the penalty strength, determined by the regularization parameter λ, is the same for all time points. However, the

estimating function $U_{\boldsymbol{\beta}(t)}$ is defined as the sum over observations, meaning that time points with fewer observations surrounding them will have a smaller score function. Then, since the shrinkage is applied uniformly, time points with fewer observations will be shrunk more. This behavior is particularly noticeable at the boundary where there typically is only half as many observations: then, the shrinkage will be twice as strong. To alleviate this issue, we propose to rescale the estimating function by the total weight:

$$U_{\boldsymbol{\beta}(t)} \leftarrow -\frac{\sum_{i=1}^{N} \mathbf{X}_i^\top \operatorname{diag}(\mathbf{k}_i(t))\mathbf{P}_i\mathbf{r}_i}{\sum_{i=1}^{N} \mathbf{1}^\top \mathbf{k}_i(t)},$$

Equivalently, this corresponds to normalizing the weights cross-sectionally:

$$\mathbf{k}_i(t) \leftarrow \mathbf{k}_i(t) / \textstyle\sum_{i=1}^{N} \mathbf{1}^\top \mathbf{k}_i(t).$$

Of note, Kong et al. (2015) also recognize the scaling issue between the estimating function and the penalty. Instead of rescaling the weights, the opt for rescaling the penalty, though it leads to a similar adjustment.

B.1.2 Covariance parameter updates

By parameterizing the covariance function as a multiple of σ^2, we can write $\mathbf{V}_i = \sigma^2\mathbf{C}_i$, where $\mathbf{C}_i = \mathbf{K}_\theta(\boldsymbol{\tau}) + \mathbf{I}$. Then, the estimate for σ^2 is given by

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{N} \mathbf{r}_i^\top \mathbf{C}_i^{-1}\mathbf{r}_i$$

where $n = \sum_{i=1}^{N} n_i$ is the total number of observations. To update the parameters of $\mathbf{C}_i$, we perform Newton-Raphson steps. Directly, the first derivative may be computed using the following expressions

$$\begin{aligned}
\partial\ell(\boldsymbol{\tau}) &= -\frac{1}{2}\sum_{i=1}^{N} \partial \log\det \mathbf{C}_i + \sigma^{-2}\partial[\mathbf{r}_i^\top \mathbf{C}_i^{-1}\mathbf{r}_i] \\
\partial \log\det \mathbf{C}_i &= \operatorname{Tr}\left(\mathbf{C}_i^{-1}\partial\mathbf{C}_i\right) \\
\partial[\mathbf{r}_i^\top \mathbf{C}_i^{-1}\mathbf{r}_i] &= -\mathbf{r}_i^\top\left(\mathbf{C}_i^{-1}\partial\mathbf{C}_i\mathbf{C}_i^{-1}\right)\mathbf{r}_i,
\end{aligned}$$

where $\partial \mathbf{C}_i$ depends on the parameterization. For example, in the compound symmetry structure, namely $\mathbf{C}_i = \mathbf{I} + r_\theta \mathbf{1}\mathbf{1}^\top$, we find $\partial \mathbf{C}_i = \mathbf{1}\mathbf{1}^\top$. For the second derivative, we have

$$\begin{aligned}
\partial^2 \ell(\boldsymbol{\tau}) &= -\frac{1}{2} \sum_{i=1}^{N} \partial^2 \log\det \mathbf{C}_i + \sigma^{-2} \partial^2 [\mathbf{r}_i^\top \mathbf{C}_i^{-1} \mathbf{r}_i] \\
\partial^2 \log\det \mathbf{C}_i &= \operatorname{Tr}\left(-\mathbf{C}_i^{-1} \partial \mathbf{C}_i \mathbf{C}_i^{-1} \partial \mathbf{C}_i + \mathbf{C}_i^{-1} \partial^2 \mathbf{C}_i\right) \\
\partial^2 [\mathbf{r}_i^\top \mathbf{C}_i^{-1} \mathbf{r}_i] &= -\mathbf{r}_i^\top \left(-2\mathbf{C}_i^{-1} \partial \mathbf{C}_i \mathbf{C}_i^{-1} \partial \mathbf{C}_i \mathbf{C}_i^{-1} + \mathbf{C}_i^{-1} \partial^2 \mathbf{C}_i \mathbf{C}_i^{-1}\right) \mathbf{r}_i.
\end{aligned}$$

Under the compound symmetry structure, $\partial^2 \mathbf{C}_i = \mathbf{0}$, so the second terms of each component vanishes.

B.1.3 Tuning parameter selection

The time points at which to estimate the varying coefficients models, $\mathbf{t}$, need to be specified before estimation. When the number of sample time points is relatively small, we suggest using those directly. Otherwise, if observations are sampled at irregular times or with high frequency, we suggest using a suitably fine grid over the observed domain. The main principle guiding this choice is about the desired output: what are the time points of interest, and what is a relevant timescale to identify differential intervals?

Second, the smoothing is mostly controlled by the choice of the kernel function k and the kernel scale h. We use the squared exponential kernel $k(t, t') = \exp(-|t - t'|^2)$ by default, but bounded-support kernels can be useful, especially in implementations exploiting sparsity. The choice of h can be done manually by the user, informed by the sampling frequency and expected smoothness. Alternatively, we propose an information criteria described below to select h empirically.

Third, the sparsity-inducing penalty contains three main parameters. The regularization strength $\lambda > 0$ will be selected similarly to h using information criteria. The global-local weight $\alpha \in [0, 1]$ should be user-selected informed by the expected or desired sparsity patterns. For example, if many covariates are included, it might be of increased interest to find which are associated with the response, in which case a pure group Lasso penalty ($\alpha = 0$) should be preferred. If a single covariate is include, such as a group indicator, identification of differential time points is of greater importance, suggesting a pure Lasso penalty ($\alpha = 1$). The adaptive strength $\gamma \geqslant 0$ is best set to be $1/2$.

To select the kernel scale h and the regularization parameter λ, we propose an extended

Bayesian information criterion (EBIC, Chen and Chen, 2008):

$$\text{EBIC} = -2\ell(\mathbf{B}, \boldsymbol{\alpha}) + \sum_{s=1}^{S} \hat{\text{df}}^{(s)} \log \hat{n}_h^{(s)} + \nu \hat{\text{df}} \log \text{df}_{\max},$$

for $\nu \geqslant 0$ ($\nu = 1/2$ by default), for $\hat{\text{df}} = \sum_{s=1}^{S} \text{df}^{(s)}$ and where $\text{df}_{\max} = Tp_x$, corresponding to the actual number of parameters defining $\boldsymbol{\beta}(\cdot)$ over $\mathbf{t}$, that is, the dimension of $\mathbf{B}$. Two quantities enter the EBIC penalty which do not have obvious values in our model: the localized sample size $\hat{n}_h^{(s)}$ and degrees of freedom $\hat{\text{df}}^{(s)}$.

B.1.3.1 Effective degrees of freedom

The number of non-zero entries in $\mathbf{B}$, namely $\|\mathbf{B}\|_0$, is not an appropriate estimate of the degrees of freedom. Indeed, suppose we have a large h inducing estimated functions that are essentially constant but non-zero. Then, $\|\mathbf{b}_j\|_0 = S$, while there is really only a single degree of freedom. Kernel smoothing methods usually use $k_h(0)|\mathcal{D}|$, where $\mathcal{D} \subset \mathbb{R}$ is the time domain, as the estimated degrees of freedom. This quantity naturally scales with h since $k_h(0) = k(0)/h$. We propose to replace $\mathcal{D}$ by $\widehat{\mathcal{D}}_{h,\lambda}$ defined by the domain estimated to be non-zero with tuning parameters h, λ. In practice, we only estimate $\widehat{\mathcal{D}}_{h,\lambda}$ at a finite set of points $\mathcal{T}$, so we estimate $|\widehat{\mathcal{D}}_{h,\lambda}| \approx |\mathcal{D}| \|\mathbf{b}_j\|_0 / S$, that is, the original domain length $|\mathcal{D}|$ multiplied by the proportion of time points estimated to be non-zero $\|\mathbf{b}_j\|_0/S$.

We also note that $k(0)|\mathcal{D}|/hT$ can be understood as a rescaling of the number of parameter $\|\mathbf{b}_j\|_0$. For small h, that scale can exceed 1, leading to an estimated degree of freedom exceeding the actual number of parameters. At the opposite end, when h gets larger, this quantity will tend to 0, even though there is still one effective parameter in the model that can be understood as the time average. Hence, we finally suggest to clamp this scale between $1/S$ and 1 and define the estimated degrees of freedom at time $t^{(s)}$ by $\hat{\text{df}}^{(s)} = \|\mathbf{b}^{(s)}\|_0 (k(0)|\mathcal{D}|/hT \wedge 1 \vee 1/S)$.

B.1.3.2 Effective sample size

Many sparse kernel regression method utilizes a BIC penalty of the form $\|\mathbf{B}\|_0 \log(nh)/nh$ (Wang and Xia, 2009), which implicitly assumes an effective sample size of nh. We have found this penalty not very effective for selection purposes, especially with respect to h, so we propose a penalty using better estimates of the effective sample size.

For the localized model at time $t \in \mathbf{t}$, not all n samples are used. In fact, the weights $k_h(t_{ij}, t)$ can be informative in determining the effective sample size. The maximum contribution towards the likelihood a sample can have is when $t_{ij} = t$, in which case, its weight

will be $k_h(0)$. This suggests the estimated sample size

$$\hat{n}_h^{(s)} = \sum_{i=1}^{N} \sum_{j=1}^{n_i} \frac{k_h(t_{ij}, t^{(s)})}{k_h(0)},$$

which has the nice property of being bounded between the number of samples at time $t^{(s)}$ $\sum_{i=1}^{N} \sum_{j=1}^{n_i} \mathbb{1}[t_{ij} = t^{(s)}]$ (when $h \to 0$) and the total number of samples n (when $h \to \infty$.) Of note, Kong et al. (2015) utilize the same expression for their effective sample size.

It turns out that nh is a crude estimate (or special case) of $\hat{n}_h(t)$. Consider uniformly distributed sampling times $t_{ij} \sim \text{Uniform}[0, 1]$ with a uniform kernel of radius h. Then, there will be, on average, $2nh$ samples falling within h of t (for t far enough from the boundary); the uniform kernel has a scaling factor of $1/2$, so we recover nh. The proposed estimate $\hat{n}_h(t)$ better captures the effective number of samples used to estimate the model at time t than nh, especially at the boundary or for non-uniform sample times. For large h, all n samples will have $k_h(t_{ij}, t) \approx k_h(0)$ and $\hat{n}_h(t)$ is capped by n from the scaling by the maximal value $k_h(0)$, whereas nh can exceed n. For very small $h \to 0$, only the t_{ij} exactly equal to t will contribute: if there are $n(t)$ such points, then $\hat{n}_h(t) \to n(t)$, whereas $nh \to 0$, which is undesirable.

B.1.3.3 Solution path

To select the regularization parameter λ, we first find the smallest value such that we find a completely sparse solution, for a fixed value of kernel scale h. This value can be determined by setting $\mathbf{B} = \mathbf{0}$ and checking what conditions the gradients must satisfy in order for the proximal update to remain $\mathbf{0}$. We find $\mathbf{b}_j^* = \eta U_{\mathbf{b}_j}$ so that the Lasso penalty shrinks back to 0 provided

$$|U_{b_j^{(s)}}| \leqslant \lambda \alpha \omega_j^{(s)}, \qquad s = 1, \ldots, S.$$

Similarly, the group Lasso penalty proximal will remain null provided

$$\|U_{\mathbf{b}_j}\|_2 \leqslant \lambda(1 - \alpha)\sqrt{S}\omega_j.$$

This leads to the condition

$$\lambda \geqslant \max_j \left[\max_s \frac{1}{\alpha \omega_j^{(s)}} |U_{\mathbf{b}_j^{(s)}}| \right] \vee \frac{1}{(1 - \alpha)\sqrt{S}\omega_j} \|U_{\mathbf{b}_j}\|_2$$

Of course, anytime we would divide by zero, we simply omit the corresponding bound.

B.1.4 Simultaneous confidence bands

We consider *sup-t* simultaneous confidence bands based on the bootstrap (Montiel Olea and Plagborg-Møller, 2019). Specifically, we resample the subjects B times with replacement to obtain B bootstrap estimates of the varying coefficients $\mathbf{B}^{(b)}$, $b = 1, \dots, B$. Sup-t bands are constructed by adjusting the pointwise interval's confidence level to ensure simultaneous coverage. Let $1 - \alpha \in (0, 1)$ be the desired coverage, and let $\hat{q}_\zeta^{(B)}(\cdot)$ denote the ζ empirical quantile of some quantity of interest over the B bootstrap samples. The ζ-tailed pointwise interval for $b_j^{(s)} = \beta_j(t^{(s)})$ is then given by $[\hat{q}_\zeta^{(B)}(b_j^{(s)}), \hat{q}_{1-\zeta}^{(B)}(b_j^{(s)})]$, with pointwise empirical coverage $1 - 2\zeta$. To obtain simultaneous coverage, we find the largest ζ preserving simultaneous empirical coverage:

$$\hat{\zeta} = \sup \left\{ \zeta \in (0,1) \;\middle|\; \frac{1}{B} \sum_{b=1}^{B} \prod_{s=1}^{S} \mathbb{1}\left[b_j^{(s,b)} \in [\hat{q}_\zeta^{(B)}(b_j^{(s)}), \hat{q}_{1-\zeta}^{(B)}(b_j^{(s)})] \right] \geqslant 1 - \alpha \right\}.$$

The simultaneous confidence band is finally given by the rectangle spanned by the pointwise confidence intervals using $\hat{\zeta}$:

$$\left\{ \mathbf{b}_j \in \mathbb{R}^S \;\middle|\; b_j^{(s)} \in [\hat{q}_{\hat{\zeta}}^{(B)}(b_j^{(s)}), \hat{q}_{1-\hat{\zeta}}^{(B)}(b_j^{(s)})], s = 1, \dots, S \right\}$$

In practice, we can restrict the search to $\zeta \in [\alpha/2S, \alpha/2]$ along a grid with increments $1/B$. While this is defined for a single varying coefficient $\beta_j(\cdot)$, the procedure can be naturally extended to any contrast or transformation of $\boldsymbol{\beta}(\cdot)$ as well as to multiple confidence bands (e.g., across multiple VCs).

B.2 Additional results

B.2.1 Additional performance metrics

To complement Figures 3 and 4 of the main text, we report additional evaluation metrics to better understand the differences observed between methods. First, we split the estimation error over the null region, i.e., where the true $\beta_j(t) = 0$, and the non-null region. Second, we report the proportion of time points selected to be non-zero. Third, we split the accuracy into false discovery rate (FDR), calculated as the proportion of estimated non-zeros that are true zeros, and into true positive rate (TPR), also referred to as *recall* or *power*, calculated as the proportion of true zeros estimated to be non-zero.

For the missing data scenario (Figure 3 of the main text, Figure B.1 here), we first note the large difference in proportion of time points estimated to be non-zero, which has direct implications on the other metrics. The true proportion of non-zeros is $4/10 = 0.4$. `SPFDA` selects far more time points than the other and than required, leading to inflated FDR and larger power. Still, the estimation error on the non-null region is lower for `LSVCMM`, even given the lower power. Interestingly, the estimation error of the null region is generally lower for `SPFDA`, suggesting that it selects null time points, but the estimate remains close to 0. Conversely, the estimation error on the null region for `LSVCMM` and `LSVCM` is higher with larger variance. The cross-secrional approach, `ALasso`, selects fewer time points, leading to reduced FDR and power and, correspondingly, lower error for null time points and higher error for non-null time points. The gap in estimation error for low variance in Experiment (a) between `LSVCMM` and `SPFDA` is mostly explained by the estimation error over the non-null region; a possible explanation is that we utilize an adaptive Lasso penalty which aims at removing the bias induced by the shrinkage, whereas `SPFDA` uses a group bridge penalty with power $0 < \alpha < 1$. For strong signal, this bias tends to dominate the variance.

For the irregular sampling scenario (Figure 4 of the main text, Figure B.2 here), the true proportion of non-null time points is $45/100 = 0.45$. Similar conclusions can be reached as for the previous experiment, with some notable differences. In this experiment, `SPFDA` exhibits large estimation error over the null region for sparse sampling (Experiments (a) and (c), and for small proportion in Experiment (c). In this regime, `SPFDA` relies heavily on the fPCA imputation. The drop in classification performance of `LSVCM` compared to `LSVCMM` when the sampling becomes dense is attributable to a larger proportion of selected time points and thus larger FDR.

The comparison of support recovery metrics suggest that `LSVCMM`, with its EBIC tuning parameter selection, is much better calibrated than `SPFDA`, which also using an EBIC criterion. Indeed, FDR is lower (around 10-20%) and roughly constant along the three pa-

rameters explored for our proposed method, while `SPFDA` has a more variable FDR centered around 30-50%. This obviously comes at a cost of lower power, but the overall discrimination (Figure 3 of the main text) is significanlty better for `LSVCMM`.

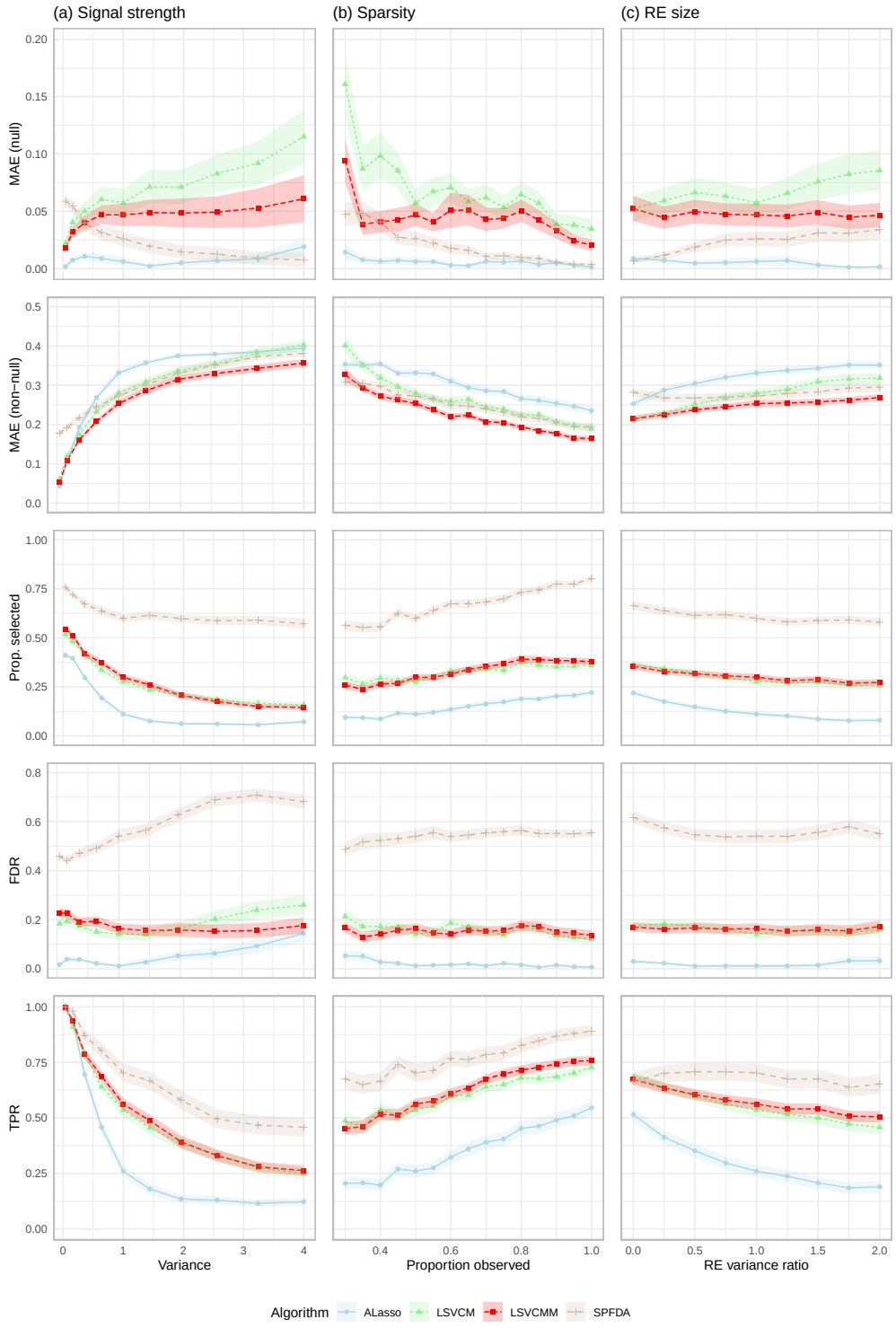

Figure B.1: Additional evaluation metrics in the missing data scenario: mean (line) and standard error (band) across 100 replications.

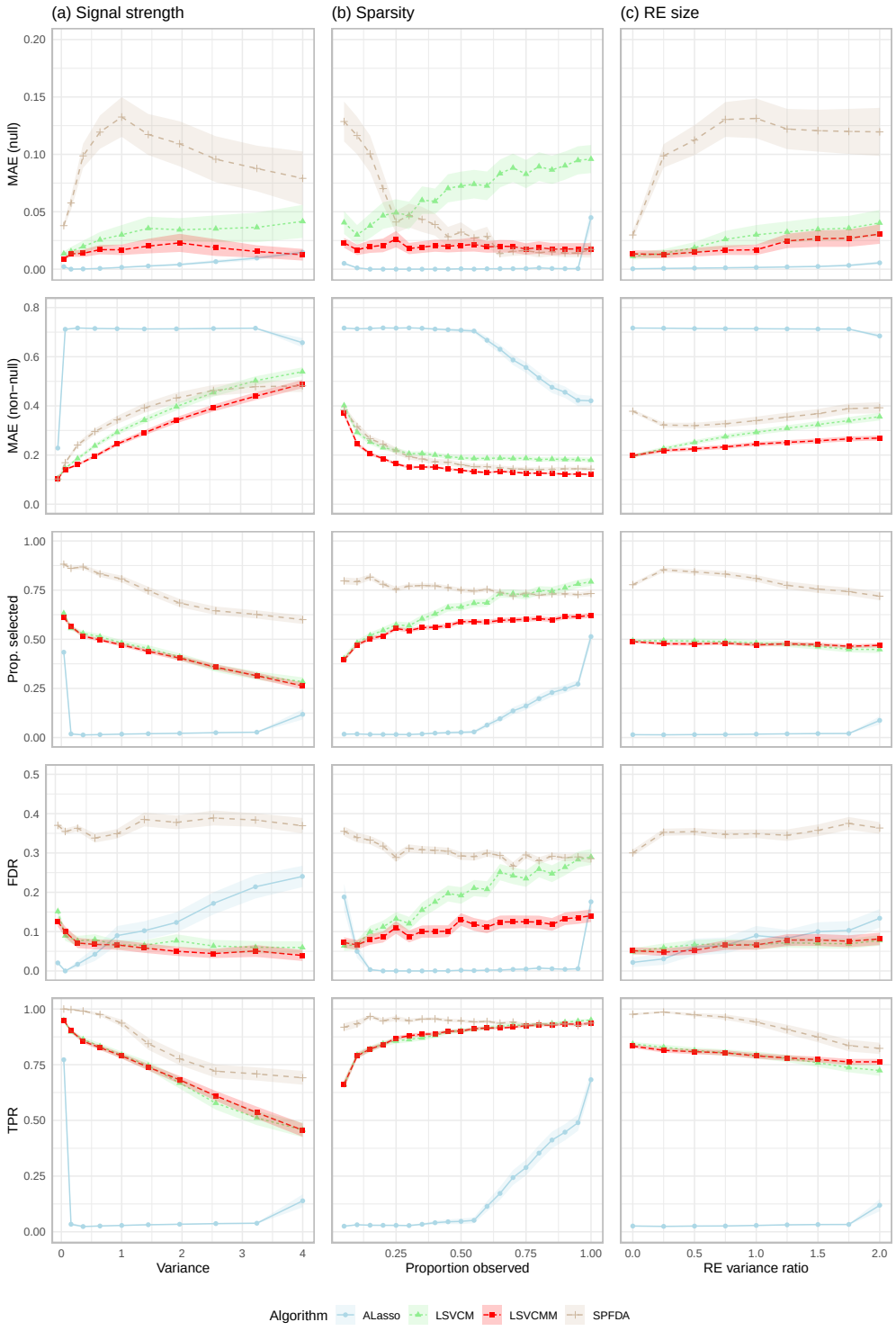

Figure B.2: Additional evaluation metrics in the irregular sampling scenario: mean (line) and standard error (band) across 100 replications.

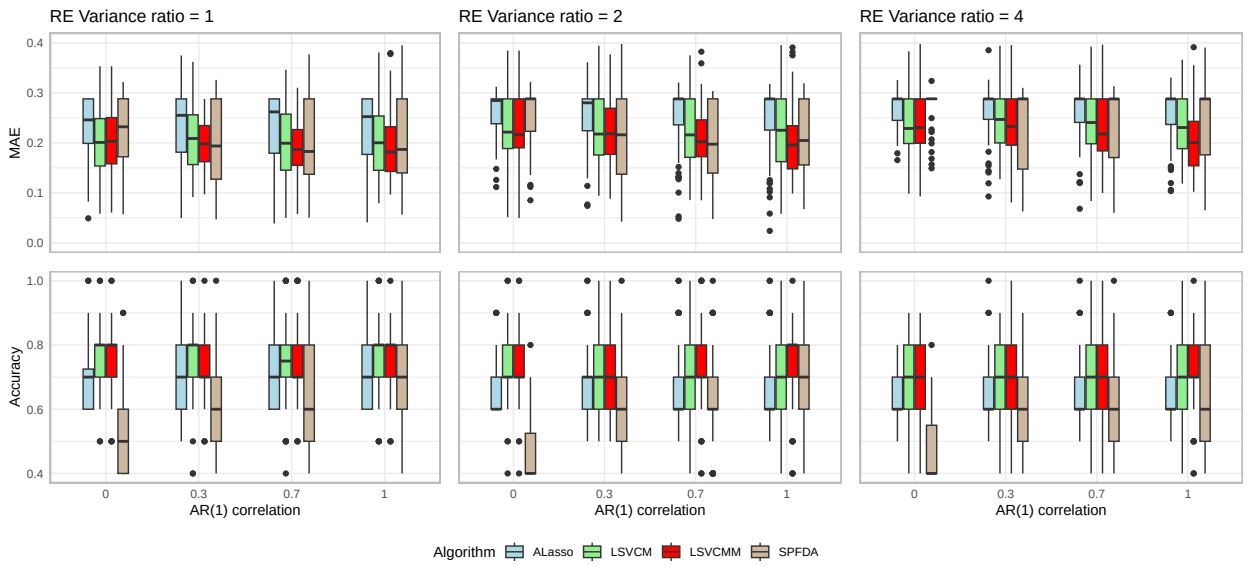

Figure B.3: Evaluation metrics for the misspecified working covariance scenario.

B.2.2 Misspecification experiment

We investigate the estimation and classificatino performance of `LSVCMM` when the working covariance does not match the true generating covariance of the data. Specifically, we modify the missing data scenario (Section 3.1 of the main text) where the within-subject correlation is now AR(1) with varying correlation, and with varying values of ratio with the noise variance. `LSVCMM` is still fitted with a compund symmetry working covariance; `SPFDA` is fitted with an unrestricted covariance which is estimated in a prior step.

Figure B.3 contains the mean absolute estimation error (MAE) and classification accuracy for support recovery. First, comparing `LSVCMM` to its independent counterpart `LSVCM`, we still find efficiency gains with a misspecified covariance, which becomes more noticeable with increase variance ratio and stronger correlation. Of note, `LSVCM` is correctly specified with correlation 0 and does not outperform `LSVCMM`, indicating there is no significant loss of performance by including dependency when there are none. That does not seem to be the case for `SPFDA` which drops in classification accuracy for small correlation. `SPFDA` is correctly specified for all cases: in particular, for correlation 0.3 and 0.7, it is the only correctly-specified method and performs worse in terms of classification accuracy.

APPENDIX C

Supplementary Materials to Chapter 4

C.1 Additional details on inference

C.1.1 Prior specification

Similarly to Ma et al. (2022), we put SMGP priors on the scaling signals $\beta^{\xi}_{y,k}$, $y = 0, 1$ and the factor signals $\beta^{z}_{y,k}$, $y = 0, 1$, independently across latent dimensions $k = 1, \ldots, K$:

$$\begin{aligned} \alpha^{z}_{0,k}, \alpha^{z}_{1,k} &\sim \mathcal{GP}(0, \kappa^{z}_{\alpha}) \\ \zeta^{z}_{k} &\sim \mathcal{TGP}_{[0,1]}(0.5, \kappa^{z}_{\zeta}) \\ \beta^{z}_{0,k} &= \alpha^{z}_{0,k} \\ \beta^{z}_{1,k} &= \zeta^{z}_{k}\alpha^{z}_{0,k} + (1 - \zeta^{z}_{k})\alpha^{z}_{1,k} \end{aligned} \qquad \begin{aligned} \alpha^{\xi}_{0,k}, \alpha^{\xi}_{1,k} &\sim \mathcal{GP}(0, \kappa^{\xi}_{\alpha}) \\ \zeta^{\xi}_{k} &\sim \mathcal{TGP}_{[0,1]}(0.5, \kappa^{\xi}_{\zeta}) \\ \beta^{\xi}_{0,k} &= \alpha^{\xi}_{0,k} \\ \beta^{\xi}_{1,k} &= \zeta^{\xi}_{k}\alpha^{\xi}_{0,k} + (1 - \zeta^{\xi}_{k})\alpha^{\xi}_{1,k}, \end{aligned}$$

where $\kappa^{z}_{\alpha}, \kappa^{z}_{\zeta}, \kappa^{\xi}_{\alpha}, \kappa^{\xi}_{\zeta}$ are covariance kernels. We put inverse Gamma priors on the noise variances:

$$\sigma_e^{-2} \sim \text{Gamma}(a_\sigma, b_\sigma).$$

For the loading matrix Θ, we put heterogeneous Gaussian priors on each entry:

$$\theta_{ek} \sim \mathcal{N}(0, \phi_{ek}), \qquad \phi^{-1}_{ek} \sim \text{Gamma}(\gamma/2, \gamma/2).$$

We experimented with the multiplicative Gamma process prior (Bhattacharya and Dunson, 2011; Durante, 2017) to force a decreasing ordering of the columns as well as a Horseshoe prior (Carvalho et al., 2009) to induce sparsity, but found both to be of little influence on the posterior since the data likelihood dominates. In particular, the ordering of the K components is rather guided by the initialization (Section C.1.3).

C.1.2 Additional details on Gibbs sampling

C.1.2.1 Joint distribution

We decompose the joint distribution into four terms:

$$p_{\text{data}} \cdot p_{\text{local processes}} \cdot p_{\text{global processes}} \cdot p_{\text{parameters}}.$$

The data likelihood is given by the normal observation model 4.3, that is,

$$p_{\text{data}} = \prod_{\ell,r} \prod_{s} p(\mathbf{x}_{\ell r}(t_s) \mid \Theta, \mathbf{z}_{\ell,r}(t_s), \boldsymbol{\xi}_{\ell r}(t_s), \Sigma),$$

where

$$p(\mathbf{x}_{\ell r}(t_s) \mid \Theta, \mathbf{z}_{\ell,r}(t_s), \boldsymbol{\xi}_{\ell r}(t_s), \Sigma) = \varphi_E(\mathbf{x}_{\ell r}(t_s) \mid \Theta \operatorname{diag}[\boldsymbol{\xi}_{\ell r}(t_s)]\mathbf{z}_{\ell,r}(t_s), \Sigma),$$

and where $\varphi_D(\cdot \mid \boldsymbol{\mu}, \Sigma)$ denotes the D-variate Gaussian density with mean $\boldsymbol{\mu}$ and covariance Σ. The local processes include primarily the mean factor processes $\mathbf{z}_{\ell r}$, which are given by $z_{\ell r,k}(\cdot) \mid \bar{z}_{\ell r,k}(\cdot) \sim \mathcal{GP}(\bar{z}_{\ell r,k}(\cdot), \kappa_z)$. Let $\mathbf{K}_z$ be the $S \times S$ matrix of covariance kernel evaluations over $\mathcal{T} = \{t_s, \ldots, t_{s'}\}$, i.e., $[\mathbf{K}_z]_{ss'} = \kappa_z(t_s, t_{s'})$. Then,

$$p_{\text{local processes}} = \prod_{\ell,r} \prod_{k} p(z_{\ell r,k}(\cdot) \mid \bar{z}_{\ell r,k}(\cdot)),$$

where

$$p(z_{\ell r,k}(\cdot) \mid \bar{z}_{\ell r,k}(\cdot)) = \varphi_S(z_{\ell r,k}(\mathcal{T}) \mid \bar{z}_{\ell r,k}(\mathcal{T}), \mathbf{K}_z).$$

The local processes also include the superpositions

$$\bar{z}_{\ell r,k}(\cdot) = \sum_{j=1}^{12} \beta^{z}_{y_{\ell,j},k}(\tau_{\ell r,j}(\cdot)) \qquad \log \xi_{\ell r,k}(\cdot) = \sum_{j=1}^{12} \beta^{\xi}_{y_{\ell,j},k}(\tau_{\ell r,j}(\cdot)),$$

which are deterministic relations, so we omit them from the joint distribution for simplicity. The global processes term corresponds to the priors for the scaling and mean factor signals, given by the SMGP prior (Ma et al., 2022). Specifically,

$$p_{\text{global processes}} = \prod_{k} p(\alpha^{z}_{0,k}(\cdot)) p(\alpha^{z}_{1,k}(\cdot)) p(\zeta^{z}_{k}(\cdot)) p(\alpha^{\xi}_{0,k}(\cdot)) p(\alpha^{\xi}_{1,k}(\cdot)) p(\zeta^{\xi}_{k}(\cdot)).$$

We also omit the deterministic relationship from the combination of these processes into $\beta^z_{y,k}, \beta^\xi_{y,k}$ for $y = 0, 1$ and $k = 1, \ldots, K$, given by

$$\beta_{0,k} = \alpha_{0,k} \qquad\qquad \beta_{1,k} = \zeta_k \alpha_{1,k} + (1 - \zeta_k)\alpha_{0,k}.$$

Let $\mathcal{T}_z = \{\tau_1, \ldots, \tau_{S_z}\}$ denote the time points of evaluation of the response, i.e., a discrete grid over $[0, T_z]$. Further denote the corresponding $S_z \times S_z$ matrices of kernel evaluation over $\mathcal{T}_z$ by $\mathbf{K}^z_\alpha, \mathbf{K}^z_\zeta, \mathbf{K}^\xi_\alpha, \mathbf{K}^\xi_\zeta$. The α signals have GP priors,

$$\begin{aligned} p(\alpha^z_{y,k}(\cdot)) &= \varphi_{S_z}(\alpha^z_{y,k}(\mathcal{T}_z) \mid \mathbf{0}_{S_z}, \mathbf{K}^z_\alpha), \\ p(\alpha^\xi_{y,k}(\cdot)) &= \varphi_{S_z}(\alpha^\xi_{y,k}(\mathcal{T}_z) \mid \mathbf{0}_{S_z}, \mathbf{K}^\xi_\alpha), \end{aligned}$$

and the ζ mixing processes have TGP priors, namely,

$$\begin{aligned} p(\zeta^z_k(\cdot)) &\propto \varphi_{S_z}(\zeta^z_{y,k}(\mathcal{T}_z) \mid 0.5\mathbf{1}_{S_z}, \mathbf{K}^z_\zeta)\mathbb{1}[0 \preceq \zeta^z_{y,k}(\mathcal{T}_z) \preceq 1], \\ p(\zeta^\xi_k(\cdot)) &\propto \varphi_{S_z}(\zeta^\xi_{y,k}(\mathcal{T}_z) \mid 0.5\mathbf{1}_{S_z}, \mathbf{K}^\xi_\zeta)\mathbb{1}[0 \preceq \zeta^\xi_{y,k}(\mathcal{T}_z) \preceq 1]. \end{aligned}$$

The remaining parameters' priors are aggregated in the last term of the joint distribution:

$$p_{\text{parameters}} = \prod_e p(\sigma^2_e) \prod_k p(\theta_{ek} \mid \phi_{ek}) p(\phi_{ek}),$$

where

$$\begin{aligned} p(\sigma^2_e) &\propto \sigma_e^{-2(a_\sigma+1)} \exp\{-b_\sigma/\sigma^2_e\} \\ p(\theta_{ek} \mid \phi_{ek}) &= \varphi(0, \theta_{ek}) \\ p(\phi_{ek}) &\propto \phi_{ek}^{-(\gamma/2+1)} \exp\{-\gamma/2\phi_{ek}\}. \end{aligned}$$

C.1.2.2 Posterior updates

We derive the full conditional distributions of all quantities. Many of them lie within known families because of conjugacy relationship: in those cases, we proceed with the standard Gibbs updates. Otherwise, the procedure will be describe along the conditional.

We define a few useful intermediary quantities that will be used throughout. The fitted mean is given by

$$m_{\ell r,e}(t_s) = \Theta_{\cdot e} \operatorname{diag}[\boldsymbol{\xi}_{\ell r}(t_s)]\mathbf{z}_{\ell r}(t_s),$$

which is aggregated into the vector $\mathbf{m}_{\ell r}(t_s)$ across electrodes $e = 1, \ldots, E$.

Observation noise variance By conjugacy, the full conditional of σ_e^2 is given by

$$\sigma_e^2 \mid \text{rest} \sim \text{InvGamma}\left(a_\sigma + \frac{1}{2}LRS, b_\sigma + \frac{1}{2}\sum_{\ell,r,s}(x_{\ell r,e}(t_s) - m_{\ell r,e}(t_s))^2\right).$$

Heterogeneity parameters By conjugacy, the full conditional of ϕ_{ek} is given by

$$\phi_{ek} \mid \text{rest} \sim \text{InvGamma}\left(\frac{\gamma+1}{2}, \frac{\gamma+\theta_{ek}^2}{2}\right).$$

Loadings By conjugacy, the full conditional of $\Theta_{e\cdot}$ is given by

$$\begin{aligned}
\Theta_{e\cdot} \mid \text{rest} &\sim \mathcal{N}_K\left(\boldsymbol{\mu}_{e|\text{rest}}, \Sigma_{e|\text{rest}}\right), \\
\Sigma_{e|\text{rest}}^{-1} &= \frac{1}{\sigma_e^2}\sum_{\ell,r,s}\mathbf{z}_{\ell r}(t_s)\,\text{diag}[\boldsymbol{\xi}_{\ell r}(t_s)]^2\mathbf{z}_{\ell r}(t_s)^\top + \text{diag}(\phi_{ek}^{-1}, k=1,\ldots,K) \\
\Sigma_{e|\text{rest}}^{-1}\boldsymbol{\mu}_{e|\text{rest}} &= \frac{1}{\sigma_e^2}\sum_{\ell,r,s}x_{\ell r,e}(t_s)\,\text{diag}[\boldsymbol{\xi}_{\ell r}(t_s)]\mathbf{z}_{\ell r}(t_s).
\end{aligned}$$

Factor processes and mean factor signals For all of $z_{\ell r,k}(\mathcal{T})$ and $\alpha_{y,k}^z(\mathcal{T}_z)$, $y = 0, 1$, we proceed in a similar way. We observe the relevant terms of the joint distribution are their respective prior and the observations. Now, all enter the observation terms through the mean $m_{\ell r,e}(t_s) = \Theta_{\cdot e}\,\text{diag}[\boldsymbol{\xi}_{\ell r}(t_s)]\mathbf{z}_{\ell r}(t_s)$, which is a linear function of any one of these processes, say $\mathbf{u}$, with prior $\mathcal{N}(\boldsymbol{\mu}_u, \Sigma_u)$. Then, let $\mathbf{L}_{\ell r,s}$ denote the mapping from either T or T_z to E such that $\mathbf{m}_{\ell r}(t_s) = \tilde{\mathbf{m}}_{\ell r}(t_s)\mathbf{L}_{\ell r,s}\mathbf{u}$, where $\tilde{\mathbf{m}}_{\ell r}(t_s)$ is the fitted mean, except that it is computed with $\mathbf{u} = \mathbf{0}$. We can then write

$$\begin{aligned}
p(\mathbf{u} \mid \text{rest}) &\propto \varphi(\mathbf{u} \mid \boldsymbol{\mu}_u, \Sigma_u)\prod_{\ell,r,s}\varphi(\mathbf{x}_{\ell r}(t_s) \mid \mathbf{L}_{\ell r,s}\mathbf{u}, \Sigma) \\
&\propto \varphi\left(\mathbf{u} \mid \boldsymbol{\mu}_\mathbf{u}, \Sigma_\mathbf{u}\right) \\
\Sigma_\mathbf{u}^{-1} &= \Sigma_u^{-1} + \sum_{\ell,r,s}\mathbf{L}_{\ell r,s}^\top\Sigma^{-1}\mathbf{L}_{\ell r,s} \\
\Sigma_\mathbf{u}^{-1}\boldsymbol{\mu}_\mathbf{u} &= \Sigma_u^{-1}\boldsymbol{\mu}_u + \sum_{\ell,r,s}\mathbf{L}_{\ell r,s}^\top\Sigma^{-1}[\mathbf{x}_{\ell r}(t_s) - \tilde{\mathbf{m}}_{\ell r}(t_s)],
\end{aligned}$$

In practice, we compute $\mathbf{L}_{\ell r,s}$ as the partial derivative of $\mathbf{m}_{\ell r}(t_s)$ with respect to $\mathbf{u}$ using automatic differentiation.

Mixing processes and scaling signals For all of $\zeta_k^\xi(\mathcal{T}_z)$ and $\alpha_{y,k}^\xi(\mathcal{T}_z)$, y=0,1, we do not have conjugacy because of the exponential activation function use to define $\boldsymbol{\xi}_{\ell r}(\cdot)$. For

$\zeta_k^z(\mathcal{T}_z)$, we do have TGP conjugacy, but we employ the same update scheme as the other just mentioned as we found it more efficient (sampling from highly-correlated and high-dimensional truncated Gaussians is difficult). Similarly to above, to update one such process $\mathbf{u}$, we get that the relevant terms lead to the factorization

$$p(\mathbf{u} \mid \text{rest}) \;\propto\; p(\mathbf{u} \mid \boldsymbol{\mu}_u, \Sigma_u) \prod_{\ell,r,s} \varphi(\mathbf{x}_{\ell r}(t_s) \mid f_{\ell r,s}(\mathbf{u}), \Sigma). \tag{C.1}$$

Notice that now the mean is no longer a linear function of the process of interest (and the prior may or may not be Gaussian: some of them will be truncated Gaussians). Hence, we utilize the MALA sampler (Roberts and Tweedie, 1996), which requires only the evaluation of (C.1) along its gradient w.r.t. $\mathbf{u}$, which can easily be obtained by automatic differentiation. We also experimented with the elliptical slice sampler (Murray et al., 2010), but we found it to be much less efficient that MALA.

C.1.3 Initialization

To quicken convergence of the MCMC, we propose an initialization scheme based on a simplification of the model. Indeed, we start from the static covariance model (`LR-SC`) and further drop the structure in the mean factor process as well as the temporal dependence, leading to a K-dimensional factor analysis (FA) model with $L \times R \times S$ "independent" observations in E dimensions:

$$\mathbf{x} \mid \mathbf{z} \sim \mathcal{N}_E(\Theta\mathbf{z}, \Sigma), \qquad\qquad \mathbf{z} \sim \mathcal{N}_K(\mathbf{0}, \mathbf{I}).$$

Using an *expectation-maximization* (EM) algorithm, we find starting values for Θ and Σ, as well as posterior mean estimates for the factor processes $\mathbf{z}_{\ell r}(\cdot)$. The processes are then smoothed within each component k and sequence ℓr before being used as initial values. We note that the K components estimated from the FA model are reordered in decreasing order of the column norm of Θ after a Varimax rotation. Finally, the chain is initialized at a *maximum a posteriori* (MAP) obtained using automatic differentiation. The scaling processes are two zero, meaning that the superpostion $\boldsymbol{\xi}_{\ell r}(\cdot)$ is initialized to zero.

C.1.4 Additional details on prediction

C.1.4.1 Marginalization

We wish to compute

$$p(\mathbf{X}_r^* \mid \mathbf{y}, \mathbf{w}_r^*, \phi) = \int p(\mathbf{X}_r^* \mid \mathbf{y}, \mathbf{w}_r^*, \mathbf{Z}_r^*, \phi) p(\mathbf{Z}_r^* \mid \mathbf{y}, \mathbf{w}_r^*, \phi) \, \mathrm{d}\mathbf{Z}_r^*,$$

given some posterior sample ϕ. The distribution of the factor processes is given by independent GPs:

$$p(\mathbf{Z}_r^* \mid \mathbf{y}, \mathbf{w}_r^*, \phi) = \prod_{k=1}^{K} p(\mathbf{z}_{r,k}^* \mid \bar{\mathbf{z}}_{r,k}^*) = \prod_{k=1}^{K} \varphi_S\left(\mathbf{z}_{r,k}^* \mid \bar{\mathbf{z}}_{r,k}^*, \mathbf{K}_z\right).$$

The factor processes enter the observation likelihood linearly in the mean:

$$p(\mathbf{X}_r^* \mid \mathbf{y}, \mathbf{w}_r^*, \mathbf{Z}_r^*, \phi) = \prod_s \varphi_E\left(\mathbf{x}_r^*(t_s) \mid \Theta \operatorname{diag}[\boldsymbol{\xi}_{\mathbf{r}}^*(t_s)] \mathbf{z}_r^*(t_s), \Sigma\right).$$

We rewrite in the form of structural equations as:

$$\begin{aligned} \mathbf{x}_r^*(t_s) &= \Theta \operatorname{diag}[\boldsymbol{\xi}_{\mathbf{r}}^*(t_s)] \mathbf{z}_r^*(t_s) + \boldsymbol{\varepsilon}_r^*(t_s), & \boldsymbol{\varepsilon}_r^*(t_s) &\sim \mathcal{N}_E(\mathbf{0}, \Sigma) \\ \mathbf{z}_{r,k}^* &= \bar{\mathbf{z}}_{r,k}^* + \boldsymbol{\varepsilon}_{r,k}^*, & \boldsymbol{\varepsilon}_{r,k}^* &\sim \mathcal{N}_S(\mathbf{0}, \mathbf{K}_z). \end{aligned}$$

Since $\mathbf{X}_r^*$ is a linear combination of Gaussian random vectors, is it also a Gaussian matrix. Hence, we only need to compute its marginal first and second moments. Directly, the marginal expectation is given by the law of total expectations

$$\begin{aligned} \mathbb{E}\{\mathbf{x}_r^*(t_s) \mid \phi\} &= \mathbb{E}\{\mathbb{E}\{\mathbf{x}_r^*(t_s) \mid \phi, \mathbf{z}_r^*(t_s)\} \mid \phi\} \\ &= \mathbb{E}\{\Theta \operatorname{diag}[\boldsymbol{\xi}_r^*(t_s)] \mathbf{z}_r^*(t_s) \mid \phi\} \\ &= \Theta \operatorname{diag}[\boldsymbol{\xi}_r^*(t_s)] \bar{\mathbf{z}}_r^*(t_s). \end{aligned}$$

The marginal spatial covariance at a given t_s can be derived by the law of total variance:

$$\begin{aligned} \operatorname{Cov}(\mathbf{x}_r^*(t_s) \mid \phi) &= \operatorname{Cov}(\mathbb{E}\{\mathbf{x}_r^*(t_s) \mid \phi, \mathbf{z}_r^*(t_s)\} \mid \phi) + \mathbb{E}\{\operatorname{Cov}(\mathbf{x}_r^*(t_s) \mid \phi, \mathbf{z}_r^*(t_s)) \mid \phi\} \\ &= \operatorname{Cov}(\Theta \operatorname{diag}[\boldsymbol{\xi}_r^*(t_s)] \mathbf{z}_r^*(t_s) \mid \phi) + \mathbb{E}\{\Sigma \mid \phi\} \\ &= \kappa_z(t_s, t_s) \Theta \operatorname{diag}[\boldsymbol{\xi}_r^*(t_s)]^2 \Theta^\top + \Sigma. \end{aligned}$$

The marginal temporal cross-covariance between t_s and $t_{s'}$ is given by

$$\begin{aligned}
&\mathrm{Cov}(\mathbf{x}_r^*(t_s), \mathbf{x}_r^*(t_{s'}) \mid \phi) \\
&= \mathrm{Cov}(\mathbb{E}\{\mathbf{x}_r^*(t_s) \mid \phi, \mathbf{z}_r^*(t_s)\}, \mathbb{E}\{\mathbf{x}_r^*(t_{s'}) \mid \phi, \mathbf{z}_r^*(t_s)\} \mid \phi) + \mathbb{E}\{\mathrm{Cov}(\mathbf{x}_r^*(t_s), \mathbf{x}_r^*(t_{s'}) \mid \phi, \mathbf{z}_r^*(t_s)) \mid \phi\} \\
&= \mathrm{Cov}(\Theta\, \mathrm{diag}[\boldsymbol{\xi}_r^*(t_s)]\mathbf{z}_r^*(t_s), \Theta\, \mathrm{diag}[\boldsymbol{\xi}_r^*(t_{s'})]\mathbf{z}_r^*(t_{s'}) \mid \phi) + \mathbb{E}\{\mathbf{0} \mid \phi\} \\
&= \kappa_z(t_s, t_{s'})\Theta\, \mathrm{diag}[\boldsymbol{\xi}_r^*(t_s)]\, \mathrm{diag}[\boldsymbol{\xi}_r^*(t_{s'})]\Theta^\top.
\end{aligned}$$

Hence, we can construct the mean and covariance for the vectorization of $\mathbf{X}_r^*$ and compute the corresponding density at $\mathbf{X}_r^*$.

This construction as also quite revealing of the differences with Ma et al. (2022), where the matrix $\mathbf{X}_r^*$ has a matrix-normal distribution since the spatial covariance is not allowed to change with time. Indeed, without the scaling processes $\boldsymbol{\xi}_r^*$, we would have the Kronecker structure $\mathbf{K}_z \otimes \Theta\Theta^\top$.

C.1.4.2 Aggregation

In practice, approximate predictive posteriors are obtained as follows:

1. For each repetition $r = 1 \ldots, R^*$, for each character $\mathbf{y} \in \mathcal{Y}$ and for each posterior sample $\phi^{(n)}$, $n = 1, \ldots, N$, compute the marginal log-likelihood $l_r^{(n)}(\mathbf{y}) = \log p(\mathbf{X}_r^* \mid \mathbf{y}, \mathbf{w}_r^*, \phi^{(n)})$.

2. Aggregate across repetitions and posterior samples into an unnormalized probability mass function $\tilde{p}^*(\mathbf{y})$, $\mathbf{y} \in \mathcal{Y}$.

3. Standardize across characters: $\hat{p}^*(\mathbf{y}) = \tilde{p}^*(\mathbf{y}) / \sum_{\mathbf{y}' \in \mathcal{Y}} \tilde{p}^*(\mathbf{y}')$

The natural aggregation procedure is to replace integration in 4.7 of the main text by the Monte Carlo average across posterior samples:

$$p(\{\mathbf{X}_r^*\}_r \mid \mathbf{y}, \{\mathbf{w}_r^*\}_r, \mathcal{D}) \approx \tilde{p}^*(\mathbf{y}) := \frac{1}{N} \sum_{n=1}^{N} \exp\left\{ \sum_{r=1}^{R^*} l_r^{(n)}(\mathbf{y}) \right\}. \tag{C.2}$$

More generally, let $\mathbf{L}(\mathbf{y})$ denote the $N \times R^*$ matrix with entries $l_r^{(n)}(\mathbf{y})$. In (C.2), we map $\mathbf{L}(\mathbf{y})$ to $\tilde{p}^*(\mathbf{y})$ by first computing the exponential sum across the second dimension of $\mathbf{L}(\mathbf{y})$ before taking the arithmetic mean across its first dimension. However, alternative means and orders of operation can be utilized.

Figure C.1 shows the binary cross-entropy evaluated along the sequence of testing repetitions over four posterior sample aggregation methods (arithmetic mean, geometric mean,

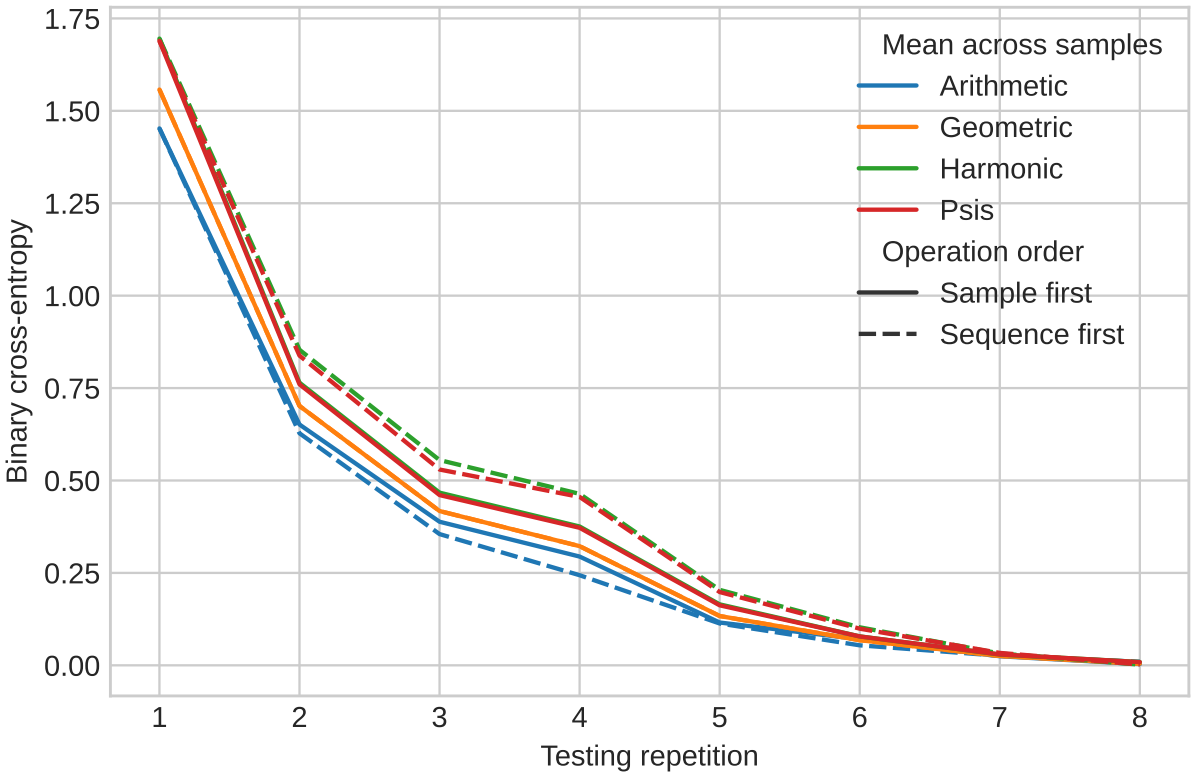

Figure C.1: Binary cross-entropy evaluated across testing repetitions for a variety of aggregation methods.

harmonic mean, and PSIS, which corresponds to the harmonic mean estimator with a prior Pareto-smoothing step) and the two orders of operation (aggregate across sample first, aggregate across sequences first). For larger number of repetitions, the difference is quite small, meaning the aggregation method does not matter much. However, for fewer repetitions, it appears that aggregating across sequences first and using the arithmetic mean to aggregate across posterior samples produces better posterior predictive probabilities. Figure C.2 shows the binary cross-entropy evaluated along the sequence of testing repetitions for a varying number of posterior samples used: we find that satisfactory performance is reached with just a few posterior samples (around 10-20 samples). This finding is significant since it means that it is quite cheap to compute accurate predictions, even though training is expensive. Conversely, using too few posterior samples (say, fewer than 10) for aggregation can be as hurtful as having one or two fewer repetitions.

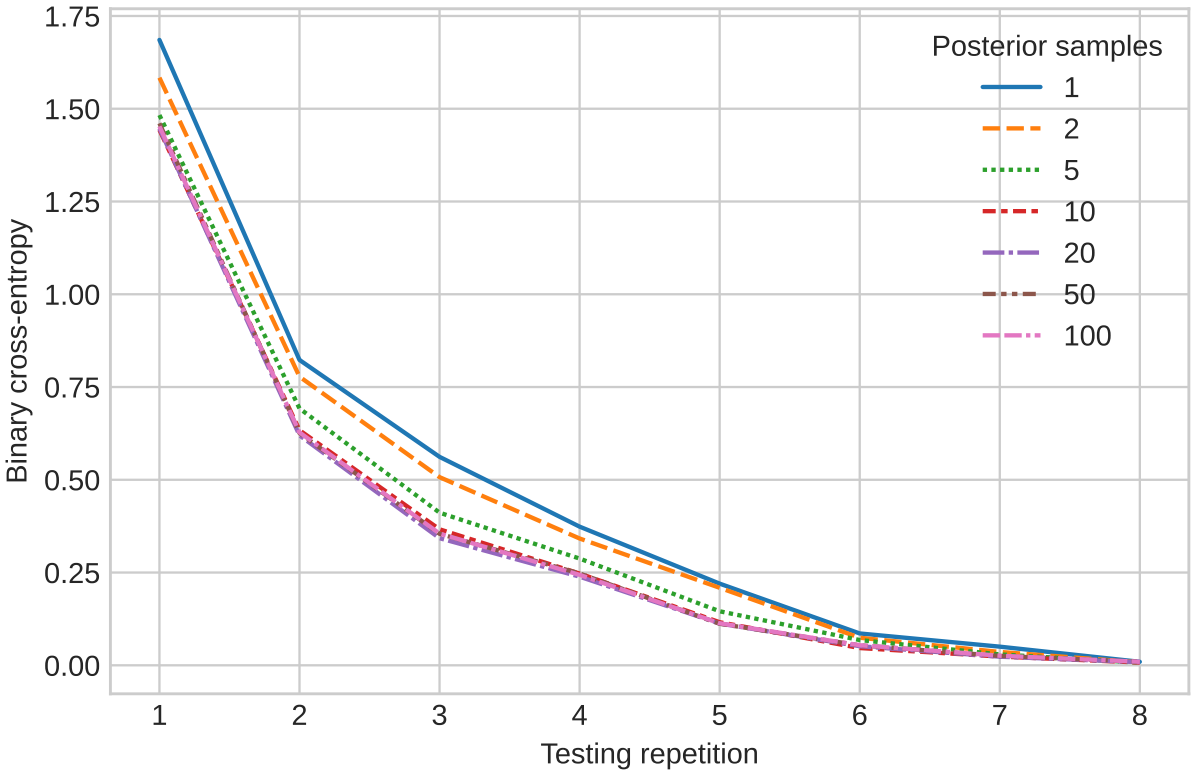

Figure C.2: Binary cross-entropy evaluated across testing repetitions for a varying number of posterior samples used for prediction.

C.2 Equivalence with FR-CS (SMGP, Ma et al., 2022)

We can express the full-rank mean and static compound symmetric covariance model of Ma et al. (2022) within our modeling framework through $K = E+1$ latent components. Indeed, Ma et al. (2022) allows each electrode's mean to vary independently of each other and the correlation among them is assumed to follow a compound symmetry structure. Hence, we set $\Theta = [\mathbf{I}_{E\times E}; \xi\mathbf{1}_{E\times 1}] \in \mathbb{R}^{E\times E+1}$. Then, we set the variance of the factor processes $z_{\ell r,k}(\cdot)$ to 0 for $k = 1, \ldots, E$, i.e., $z_{\ell r,k}(\cdot) \equiv \bar{z}_{\ell r,k}(\cdot)$ so that they do not influence the covariance structure. The mean factor process for the $E+1$th component is set to zero so it does not affect the mean structure. The scaling processes are set to zero. From this construction, we recover the mean and covariance structures of Ma et al. (2022), namely,

$$\begin{aligned}
\mathbb{E}\left\{\mathbf{x}_{\ell r,e}(t) \mid \phi\right\} &= z_{\ell r,e}(t) \\
\mathrm{Cov}\left(\mathbf{x}_{\ell r}(t) \mid \phi\right) &= \xi^2\mathbf{1}\mathbf{1}^\top + \mathbf{I} + \Sigma \\
\mathrm{Cov}\left(\mathbf{x}_{\ell r}(t), \mathbf{x}_{\ell r}(t') \mid \phi\right) &= \kappa_z(t,t')[\xi^2\mathbf{1}\mathbf{1}^\top + \mathbf{I}], \qquad t \neq t,
\end{aligned}$$

except for the extra $+\mathbf{I}$ terms. For the spatial covariance, this is absorbed into Σ (the observation variance is much larger than 1). Hence, the only difference is that the autocorrelation has an extra $\kappa_z(t,t')\mathbf{I}$, which is not present in Ma et al. (2022); again, the $\mathbf{I}$ is quite small compared to $\xi\mathbf{1}\mathbf{1}^\top$, so the difference is minor. In summary, `FR-CS` is both simpler in terms of covariance and more flexible in terms of mean than our proposed model

C.3 Additional simulation results

Figure C.3 contains an example of simulated loadings and mean factor signals used in Section 4.4 of the main text.

C.3.1 Variant selection

The selection criterion discussed in Section 4.3.3 of the main text can also be used for selecting between the variants of our proposed model discussed in Section 4.2.7 of the main text. We generate data according to each `LR-DCR`, `LR-DC` and `LR-SC`, and fit those same three models. We then compute the PSIS-LOO-CV selection criterion using the data likelihood ($p(x \mid y)$) as well as the likelihood on a held-out test set of the same dimensions. Results, contained in Figure C.5, indicate that the PSIS-LOO-CV criterion is able to exclude models that are not flexible enough to adjust to the data. When the true generating model is `LR-DCR`, the simpler models (`LR-DC` and `LR-SC`), which do not account for changes in covariance between stimulus types, have a lower value of the criterion and the test likelihood is lower as well. When the true generating model is `LR-DC`, both `LR-DCR` and `LR-DC` are correctly specified, though `LR-DCR` is over-specified. We find that both achieve higher PSIS-LOO-CV than the incorrectly-specified `LR-SC`, but all produce similar test likelihood. When data is generated according to `LR-SC`, all three models perform similarly. These results suggests that our approach is robust to some overfitting where including regression in the covariance when there are no difference in covariance between stimulus type does not hurt performance. In Figures C.6 and C.7, we include the estimation error of the mean and the covariance functions. The mean function estimation error does not change significantly across all nine cases, but the estimation error of the spatial covariance function is generally lower when a sufficiently flexible model is used and remains comparable when an over-specified model is used.

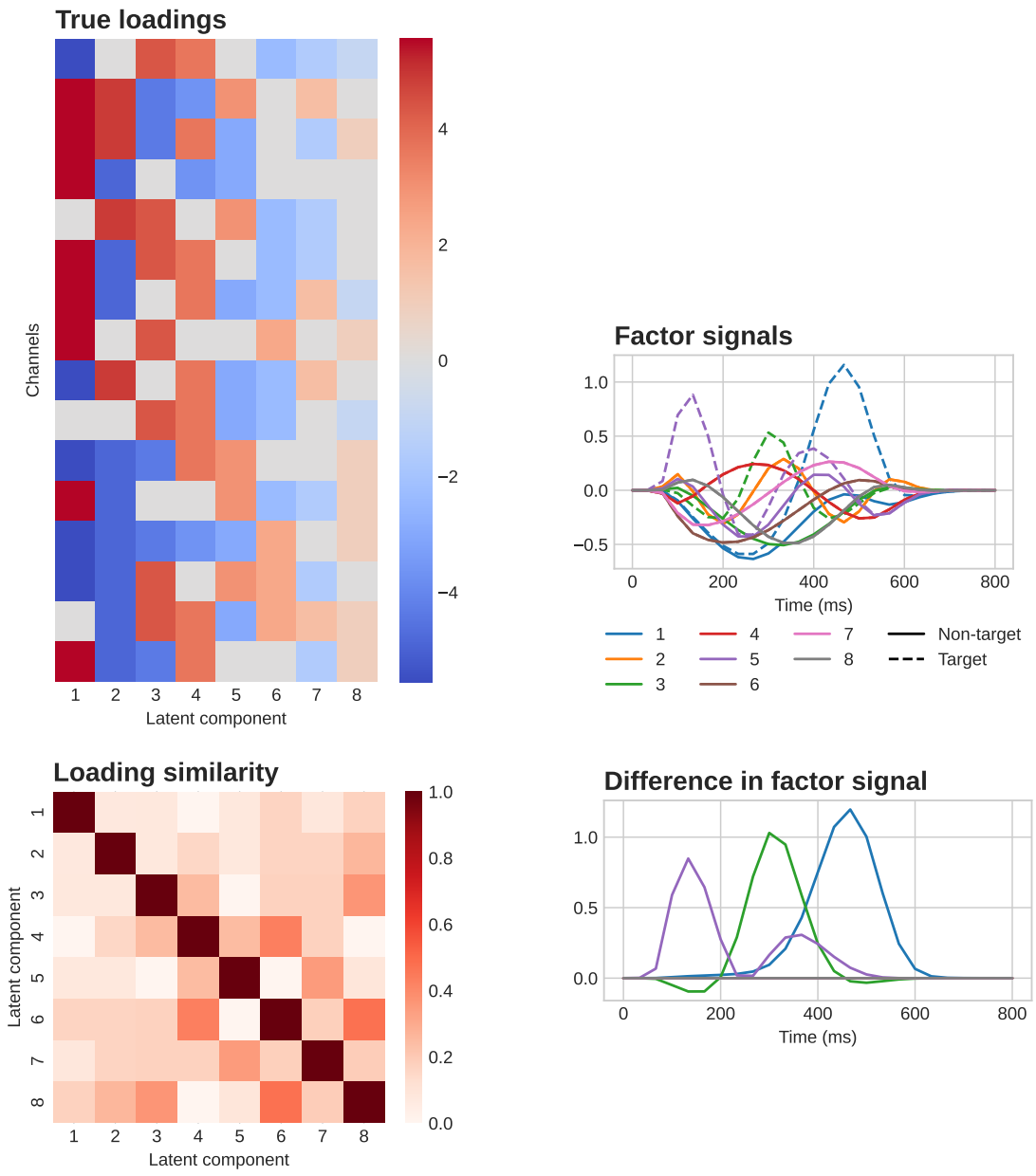

Figure C.3: Simulated global parameters for the simulation studies of Section 4.4 of the main text.

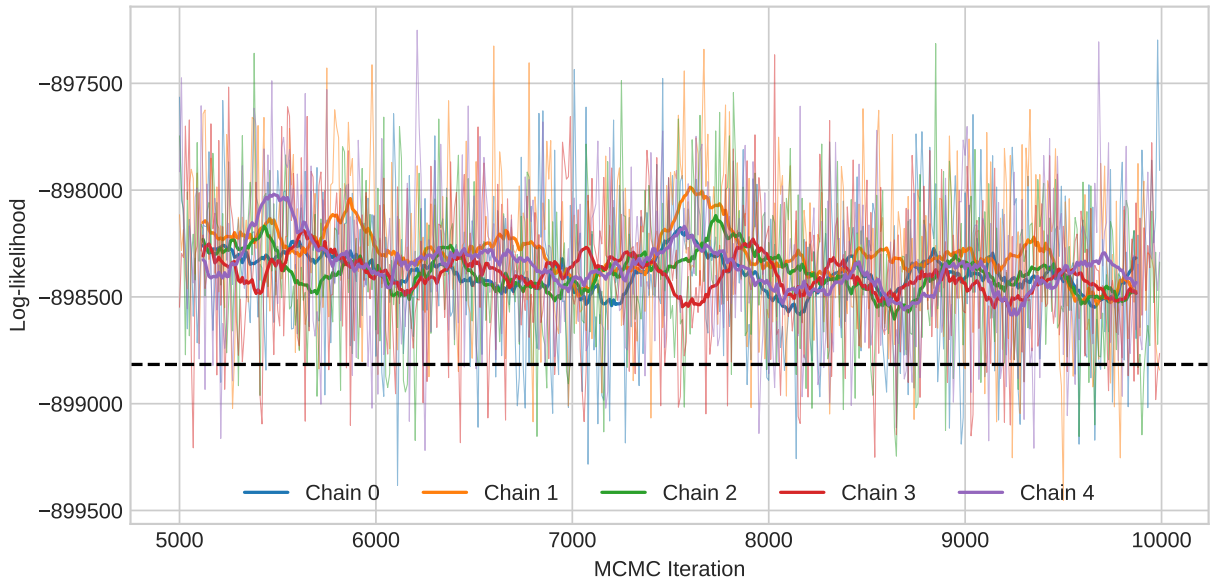

Figure C.4: Data log-likelihood evaluated along five MCMC chains (with moving average) for the simulated data of Section 4.4.

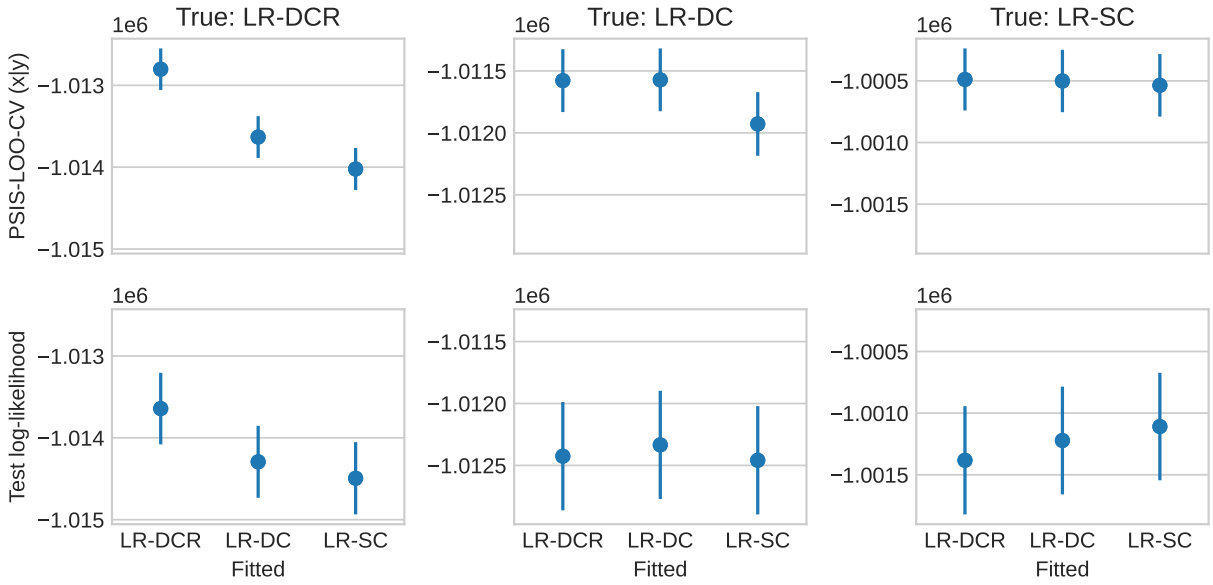

Figure C.5: PSIS-LOO-CV and data likelihood on a held-out test set (with standard error) where data is generating according to one variant and fitted using all variants.

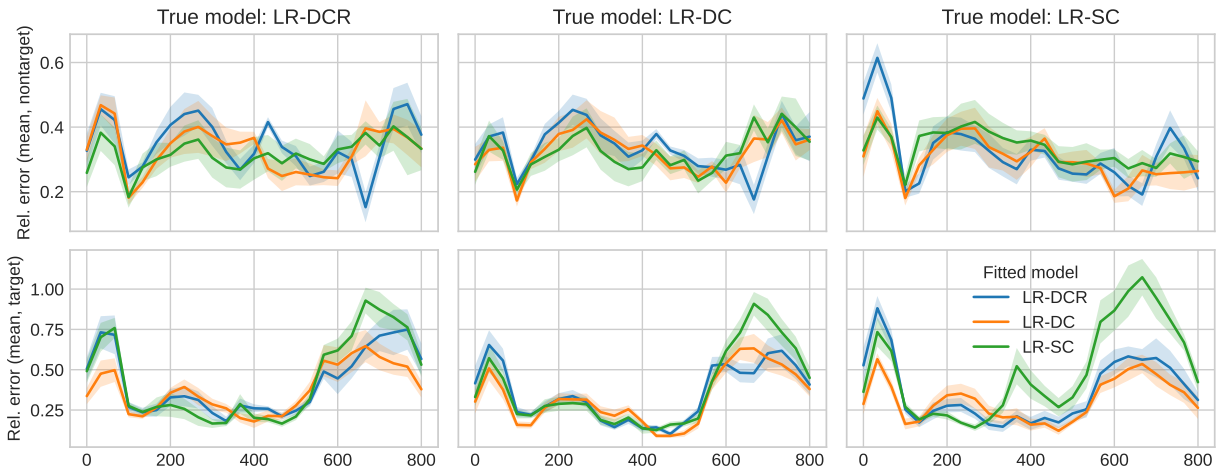

Figure C.6: Relative error in estimation of the mean function over the response window, computed as $\|\hat{\mathbf{m}} - \mathbf{m}\|/(1 + \|\mathbf{m}\|)$ where $\mathbf{m}$ is the true mean and $\hat{\mathbf{m}}$ is the fitted mean (the mean function is close to 0 for latencies close to the boundary). Lines corresponds to the average over posterior sample; bands corresponds to ± 1 posterior standard deviation.

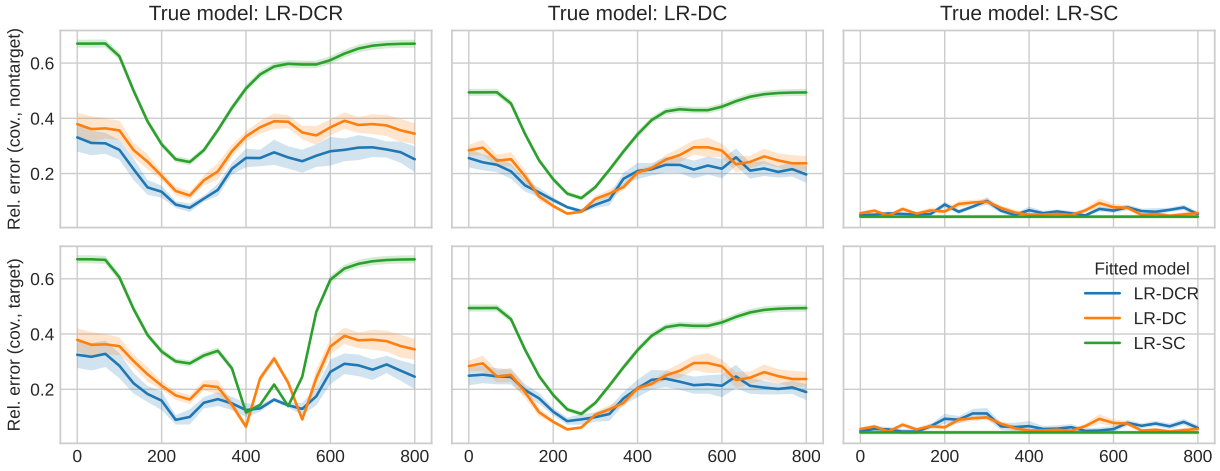

Figure C.7: Relative error in estimation of the spatial covariance function over the response window, computed as $\|\hat{\mathbf{V}} - \mathbf{V}\|_F/\|\mathbf{V}\|_)$ where $\mathbf{V}$ is the true mean and $\hat{\mathbf{V}}$ is the fitted spatial covariance. Lines corresponds to the average over posterior sample; bands corresponds to ± 1 posterior standard deviation.

C.4 Additional application results

C.4.1 Prediction performance

We investigate the key prediction performance of our proposed models in comparison to other existing approaches. To this end, we sample seven training repetitions and seven testing repetitions from the 15 original repetitions. Methods are fitted to the first set of repetitions and their prediction performance is assessed on the second set of repetitions through accuracy (out of 36 keys) and binary cross entropy.

The methods we compare are as follows. We include `LR-DCR` as well as the simpler variants `LR-SC` and `FR-CS` (Ma et al., 2022). These three models can be understood as *generative* in the sense that it fits the whole sequence rather than the conditional stimulus type probability. We include an additional simple generative model labelled `MN-LDA` consisting of fitting unconstrained matrix normal distributions to target and non-target stimulus responses and predicting stimulus type using Bayes' formula. The remaining methods are all discriminative in the sense that only the conditional stimilus type probabilities are produced. Two BCI-specific methods are also included: `swLDA` (Donchin et al., 2000; Krusienski et al., 2008) and `EEGNet` (Lawhern et al., 2018). We further consider some machine learning classifiers who have been used in similar applications: random forests (Okumus and Aydemir, 2017, `RF`), gradient boosting (Leoni et al., 2021, `GB`), and support vector machines (Kaper et al., 2004, `SVM`) with linear kernel. Specific implementation details can be found in Section C.4.2.

Figure C.8 contains the performance metrics of all methods. In terms of accuracy, `SVM` is the only method achieving perfect prediction for all 10 repetition resamplings. `LogReg` makes a single error for one of the 10 resampling, while `LR-SC` makes a single error for two of the 10 resamplings and `LR-DCR` for three resampling. Of note, `swLDA` achieves perfect accuracy for eight of the 10 resamplings, but only correctly predicts 6/19 keys for the other two, suggesting some strong sensibility to the training data. The binary cross-entropy follows a similar trend where `LogReg` and `SVM` produce the best adjusted probabilities, followed closely by `LR-DCR` and `LR-SC`.

Comparing our low-rank methods (`LR-DCR` and `LR-SC`) to the full-rank approach (Ma et al., 2022, `FR-CS`), we find improved prediction performance, suggesting that aggregating channels into fewer latent components helps generalizability. Figure 4.6 of the main text suggested, through information criterion, that `LR-DCR` might have slightly more predictive power than `LR-SC`, but this was computed using all 15 repetitions. In this seven repetition experiments, the simpler model generalizes slightly better instead. The `MN-LDA` method performance indicate that even a simple generative model can achieve almost as well as discriminative ones, though improvements can be achieved by adding structure.

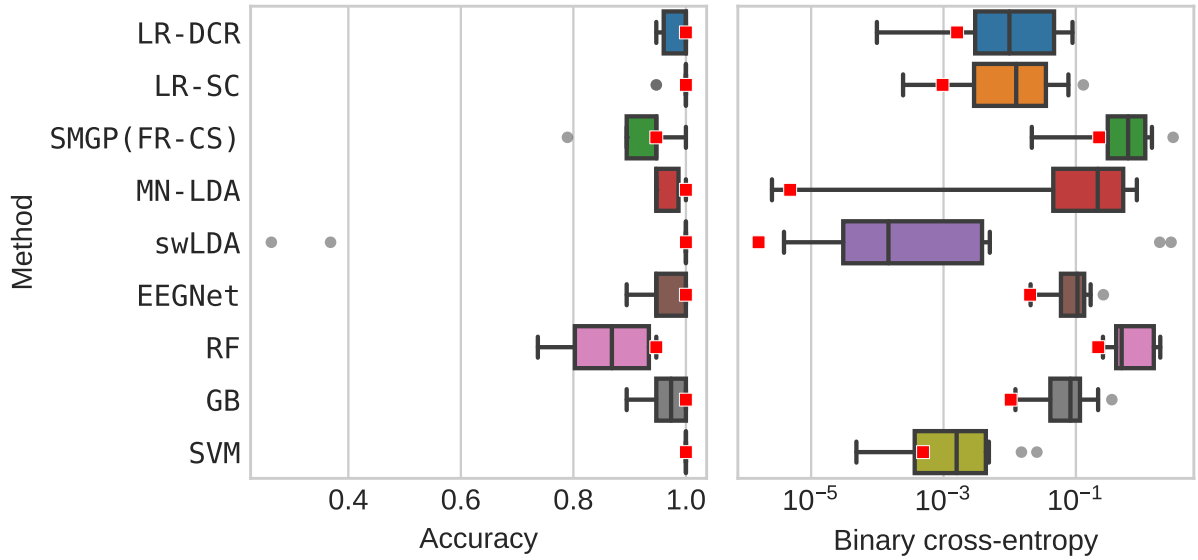

Figure C.8: Testing accuracy and binary cross-entropy of competing method across 10 random resampling of 7 training and testing repetitions. Squares indicate the performance for the particular split of odd repetitions in the training set and even repetitions in the testing set, excluding the first.

Figure C.8 also contains the performance metrics for the particular split used in Ma et al. (2022) where the training set consists of all odd repetitions and the testing set of all even repetitions, excluding the first. This split avoids some of the heterogeneity along repetitions we find with random splitting, so most methods achieve perfect accuracy.

C.4.2 Competing methods

All discriminative methods are implemented as binary classifiers of stimulus type. Hence, we chunk the EEG readings according to stimuli rather than sequences, meaning that we get $L \times R \times J$ examples, of which 1/6th are target stimuli. We obtain the predicted probability of a target stimuli and aggregate them into a predictive key probability as follows. Given a key $k \in \mathcal{K}$ in row $r(k)$ and column $c(k)$, we add the log probabilities that $r(k)$ and $c(k)$ are target across and standardize the exponential sum to 1 across keys. We then aggregate across repetitions in a similar way: sum the log probabilities across repetitions and standardize the exponential sum to 1 across keys. Since `swLDA` and `SVM` do not directly provide probabilities, we use Platt scaling (Platt and others, 1999) to get probabilities. We note that for `EEGNet` (Lawhern et al., 2018), we did not decimate the EEG readings by a factor of 8, as we found the performance to be much worse. For `FR-CS` (Ma et al., 2022), we use our implementation

with the modifications in Section C.2 as their implementation was difficult to adapt to our settings.

C.4.3 Additional subjects

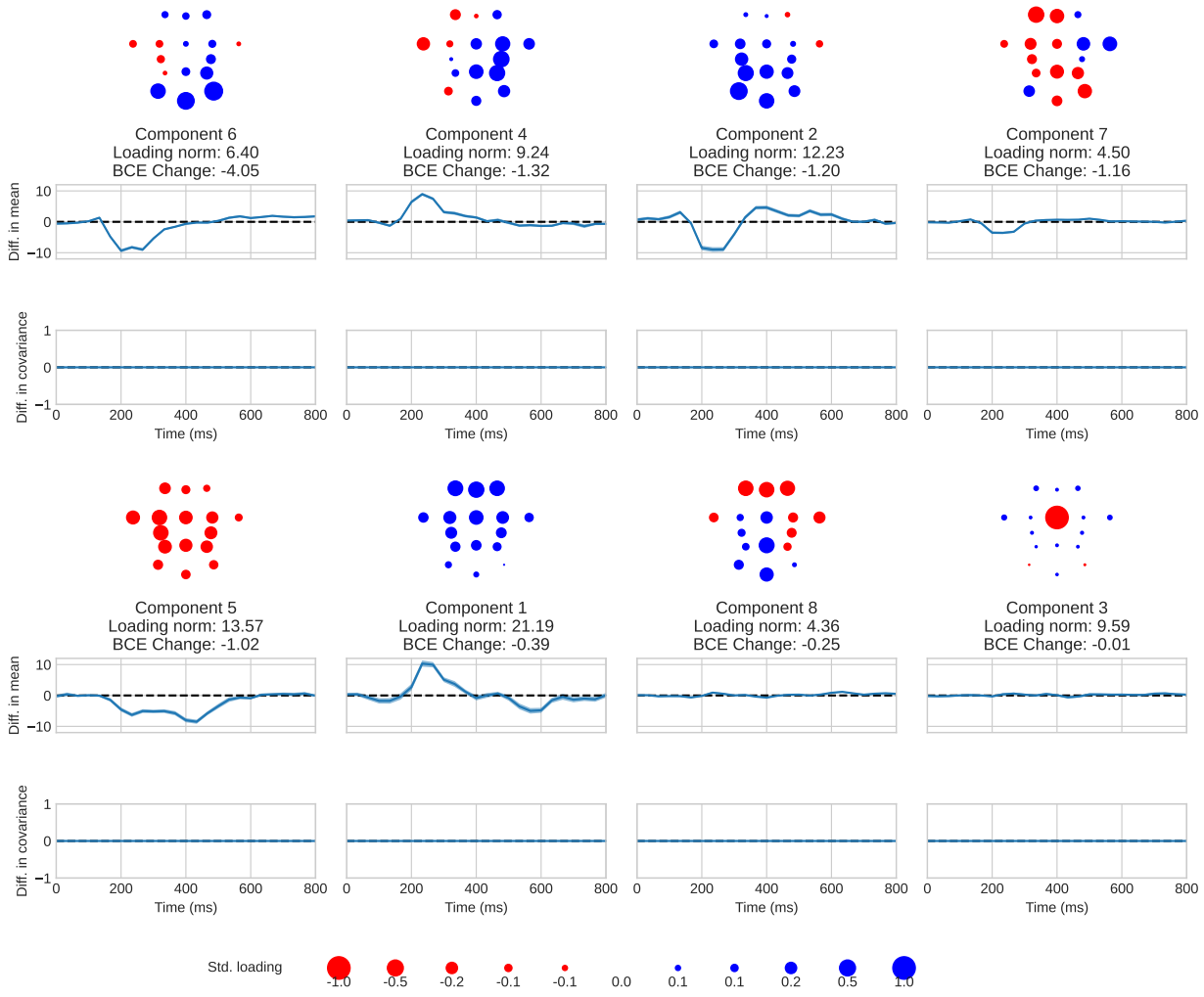

Figure C.9: Posterior summaries for the $K = 8$ components estimated using all 15 training repetitions of subject 117 in Thompson et al. (2014). Components are ordered by predictive importance, as measured by the change in binary cross-entropy (BCE) when dropped. For each component: (top) posterior mean of the loading standardized to norm one, (middle) posterior mean and pointwise standard deviation difference in mean (4.9), and (bottom) posterior mean and pointwise standard deviation difference in scaling (4.10). Refer to Figure 4.5 for electrode labels.

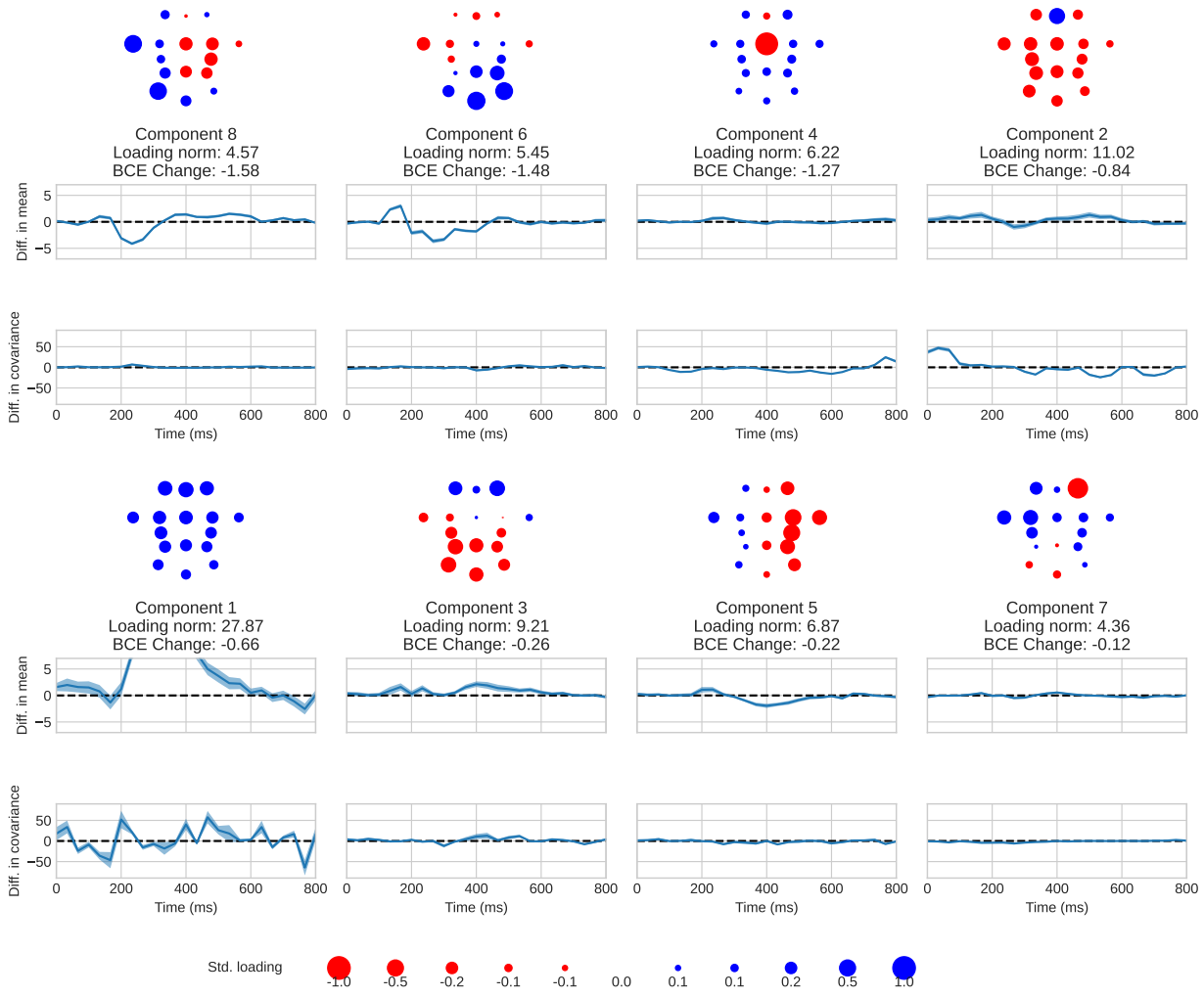

Figure C.10: Posterior summaries for the $K = 8$ components estimated using all 15 training repetitions of subject 146 in Thompson et al. (2014). Components are ordered by predictive importance, as measured by the change in binary cross-entropy (BCE) when dropped. For each component: (top) posterior mean of the loading standardized to norm one, (middle) posterior mean and pointwise standard deviation difference in mean (4.9), and (bottom) posterior mean and pointwise standard deviation difference in scaling (4.10). Refer to Figure 4.5 for electrode labels.

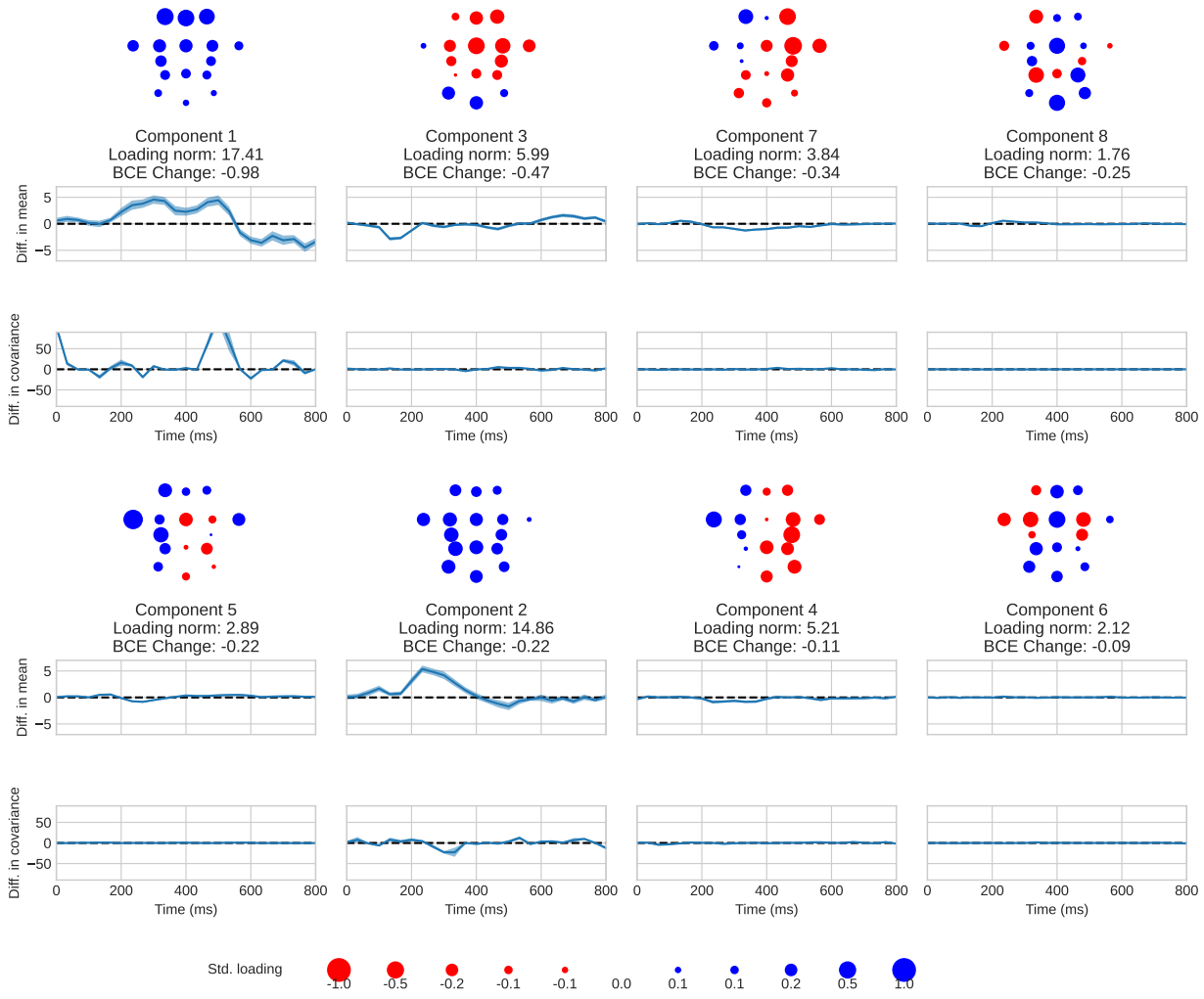

Figure C.11: Posterior summaries for the $K = 8$ components estimated using all 15 training repetitions of subject 171 in Thompson et al. (2014). Components are ordered by predictive importance, as measured by the change in binary cross-entropy (BCE) when dropped. For each component: (top) posterior mean of the loading standardized to norm one, (middle) posterior mean and pointwise standard deviation difference in mean (4.9), and (bottom) posterior mean and pointwise standard deviation difference in scaling (4.10). Refer to Figure 4.5 for electrode labels.

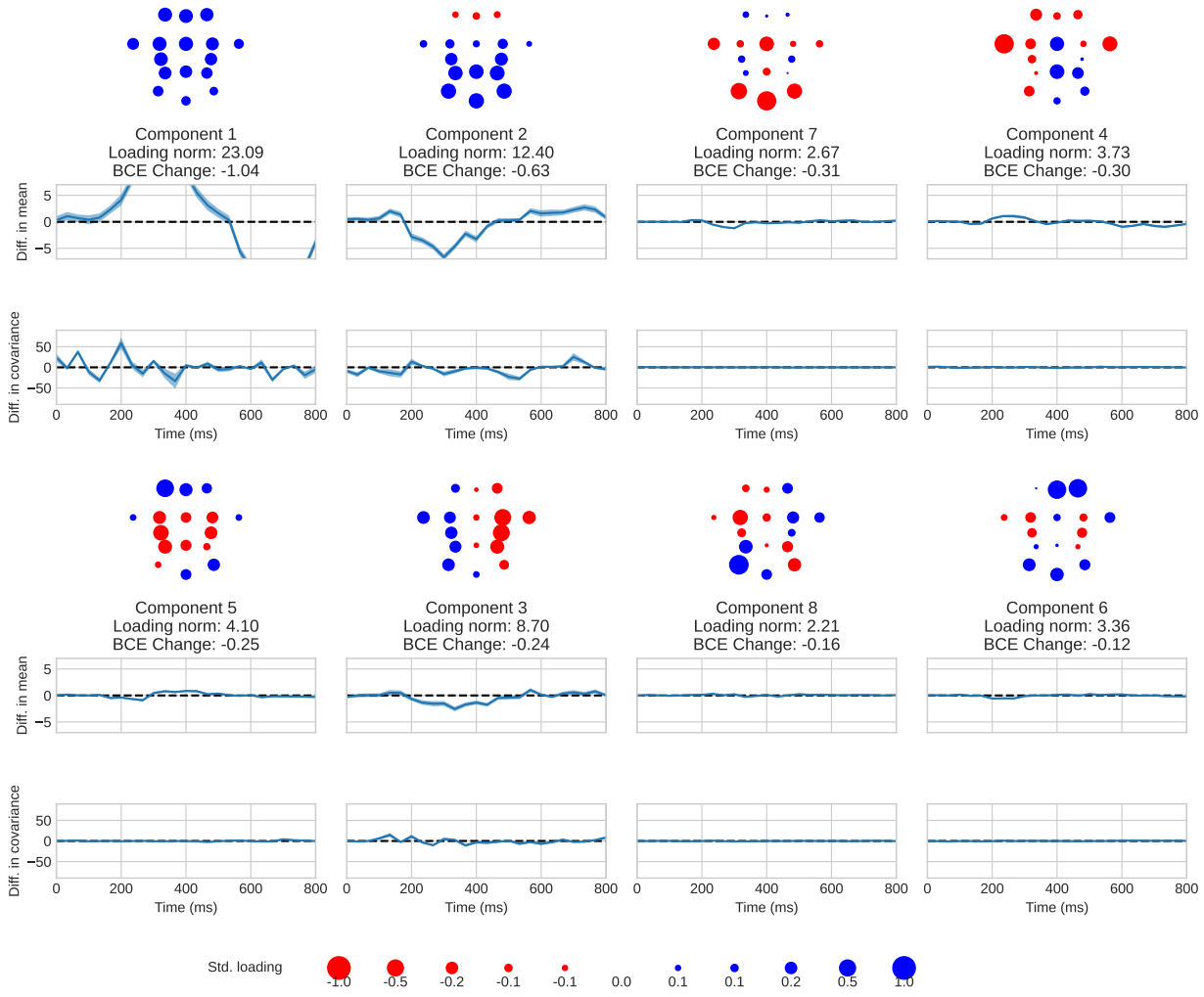

Figure C.12: Posterior summaries for the $K = 8$ components estimated using all 15 training repetitions of subject 183 in Thompson et al. (2014). Components are ordered by predictive importance, as measured by the change in binary cross-entropy (BCE) when dropped. For each component: (top) posterior mean of the loading standardized to norm one, (middle) posterior mean and pointwise standard deviation difference in mean (4.9), and (bottom) posterior mean and pointwise standard deviation difference in scaling (4.10). Refer to Figure 4.5 for electrode labels.

BIBLIOGRAPHY

Abiri, R., Borhani, S., Sellers, E. W., Jiang, Y., and Zhao, X. (2019). A comprehensive review of EEG-based brain–computer interface paradigms. *Journal of Neural Engineering*, 16(1):011001.

Ahn, J., Segers, S., and Hayes, R. B. (2012). Periodontal disease, Porphyromonas gingivalis serum antibody levels and orodigestive cancer mortality. *Carcinogenesis*, 33(5):1055–1058.

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*.

Andersen, M., Winther, O., Hansen, L. K., Poldrack, R., and Koyejo, O. (2018). Bayesian Structure Learning for Dynamic Brain Connectivity. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 1436–1446. PMLR.

Bhattacharya, A. and Dunson, D. B. (2011). Sparse Bayesian infinite factor models. *Biometrika*, 98(2):291–306.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Cai, H., Zheng, V. W., and Chang, K. C.-C. (2018). A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637.

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2009). Handling Sparsity via the Horseshoe. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, pages 73–80. PMLR. ISSN: 1938-7228.

Chakrabarti, D., Funiak, S., Chang, J., and Macskassy, S. A. (2017). Joint label inference in networks. *Journal of Machine Learning Research*, 18(1):1941–1979.

Che, C., Jin, I. H., and Zhang, Z. (2021). Network Mediation Analysis Using Model-Based Eigenvalue Decomposition. *Structural Equation Modeling: A Multidisciplinary Journal*, 28(1):148–161.

Chen, J. and Chen, Z. (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771.

Comerchero, M. D. and Polich, J. (1999). P3a and P3b from typical auditory and visual stimuli. *Clinical Neurophysiology*, 110(1):24–30.

Daudin, J.-J., Picard, F., and Robin, S. (2008). A mixture model for random graphs. *Statistics and Computing*, 18(2):173–183.

Dauwels, J., Eckford, A., Korl, S., and Loeliger, H.-A. (2009). Expectation Maximization as Message Passing - Part I: Principles and Gaussian Messages. arXiv:0910.2832 [cs, math].

Daye, Z. J., Xie, J., and Li, H. (2012). A Sparse Structured Shrinkage Estimator for Nonparametric Varying-Coefficient Model With an Application in Genomics. *Journal of Computational and Graphical Statistics*, 21(1):110–133.

Dinteren, R. v., Arns, M., Jongsma, M. L. A., and Kessels, R. P. C. (2014). P300 Development across the Lifespan: A Systematic Review and Meta-Analysis. *PLOS ONE*, 9(2):e87347.

Donchin, E., Spencer, K., and Wijesinghe, R. (2000). The mental prosthesis: assessing the speed of a P300-based brain-computer interface. *IEEE Transactions on Rehabilitation Engineering*, 8(2):174–179.

Dong, Z., Chen, B., Pan, H., Wang, D., Liu, M., Yang, Y., Zou, M., Yang, J., Xiao, K., Zhao, R., Zheng, X., Zhang, L., and Zhang, Y. (2019). Detection of Microbial 16S rRNA Gene in the Serum of Patients With Gastric Cancer. *Frontiers in Oncology*, 9.

Durante, D. (2017). A note on the multiplicative gamma process. *Statistics & Probability Letters*, 122:198–204.

Fan, J., Huang, T., and Li, R. (2007). Analysis of Longitudinal Data With Semiparametric Estimation of Covariance Function. *Journal of the American Statistical Association*, 102(478):632–641.

Fan, J. and Wu, Y. (2008). Semiparametric Estimation of Covariance Matrixes for Longitudinal Data. *Journal of the American Statistical Association*, 103(484):1520–1533.

Farwell, L. A. and Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology*, 70(6):510–523.

Flemer, B., Warren, R. D., Barrett, M. P., Cisek, K., Das, A., Jeffery, I. B., Hurley, E., O'Riordain, M., Shanahan, F., and O'Toole, P. W. (2018). The oral microbiota in colorectal cancer is distinctive and predictive. *Gut*, 67(8):1454–1463.

Folstein, J. R. and Van Petten, C. (2008). Influence of cognitive control and mismatch on the N2 component of the ERP: A review. *Psychophysiology*, 45(1):152–170.

Fontaine, S., D'Silva, N. J., de Medeiros, M. C., Chen, G. Y., Zhu, J., and Li, G. (2024a). Locally sparse varying coefficient mixed model with application to longitudinal microbiome differential abundance. *(under review at the Journal of the American Statistical Association).*

Fontaine, S., Huggins, J. E., Zhu, J., and Kang, J. (2024b). Dynamic Latent Factor Regression for EEG-Based Brain-Computer Interfaces. *(to be submitted)*.

Fontaine, S., Zhu, J., and Kang, J. (2024c). Missing Value Imputation in Relational Data using Variational Inference. *(under review at the Journal of Computational and Graphical Statistics)*.

Fosdick, B. K. and Hoff, P. D. (2015). Testing and Modeling Dependencies Between a Network and Nodal Attributes. *Journal of the American Statistical Association*, 110(511):1047–1056.

Fox, E. B. and Dunson, D. B. (2015). Bayesian Nonparametric Covariance Regression. *Journal of Machine Learning Research*, 16(77):2501–2542.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). A note on the group lasso and a sparse group lasso. arXiv:1001.0736 [math, stat].

Gao, S., Liu, Y., Duan, X., Liu, K., Mohammed, M., Gu, Z., Ren, J., Yakoumatos, L., Yuan, X., Lu, L., Liang, S., Li, J., Scott, D. A., Lamont, R. J., Zhou, F., and Wang, H. (2021). Porphyromonas gingivalis infection exacerbates oesophageal cancer and promotes resistance to neoadjuvant chemotherapy. *British Journal of Cancer*, 125(3):433–444.

Goldsmith, J., Greven, S., and Crainiceanu, C. (2013). Corrected Confidence Bands for Functional Data Using Principal Components. *Biometrics*, 69(1):41–51.

Goldsmith, J., Scheipl, F., Huang, L., Wrobel, J., Di, C., Gellar, J., Harezlak, J., McLean, M. W., Swihart, B., Xiao, L., Crainiceanu, C., and Reiss, P. T. (2023). *refund: Regression with Functional Data*.

Gollini, I. and Murphy, T. B. (2016). Joint modeling of multiple network views. *Journal of Computational and Graphical Statistics*, 25(1):246–265.

Gomaa, E. Z. (2020). Human gut microbiota/microbiome in health and diseases: a review. *Antonie van Leeuwenhoek*, 113(12):2019–2040.

Gonzalez-Navarro, P., Marghi, Y. M., Azari, B., Akçakaya, M., and Erdoğmuş, D. (2019). An Event-Driven AR-Process Model for EEG-Based BCIs With Rapid Trial Sequences. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(5):798–804.

Gu, J. and Yu, P. L. H. (2022). Joint latent space models for ranking data and social network. *Statistics and Computing*, 32(3):51.

Guhaniyogi, R. and Rodriguez, A. (2020). Joint Modeling of Longitudinal Relational Data and Exogenous Variables. *Bayesian Analysis*, 15(2):477–503. Publisher: International Society for Bayesian Analysis.

Higuchi, R., Goto, T., Hirotsu, Y., Otake, S., Oyama, T., Amemiya, K., Ohyama, H., Mochizuki, H., and Omata, M. (2021). Sphingomonas and Phenylobacterium as Major Microbiota in Thymic Epithelial Tumors. *Journal of Personalized Medicine*, 11(11):1092.

Ho, Q., Song, L., and Xing, E. (2011). Evolving cluster mixed-membership blockmodel for time-evolving networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 342–350.

Hoff, P. D. (2003). Random effects models for network data. In *Proceedings of the National Academy of Sciences: Symposium on Social Network Analysis for National Security*, pages 302–322. National Academies Press.

Hoff, P. D. (2009). Multiplicative latent factor models for description and prediction of social networks. *Computational and Mathematical Organization Theory*, 15(4):261.

Honda, T. (2021). The de-biased group Lasso estimation for varying coefficient models. *Annals of the Institute of Statistical Mathematics*, 73(1):3–29.

Huisman, M. and Krause, R. W. (2017). Imputation of Missing Network Data. In Alhajj, R. and Rokne, J., editors, *Encyclopedia of Social Network Analysis and Mining*, pages 707–705. Springer, New York, NY.

Jaakkola, T. S. and Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10(1):25–37.

Jeganathan, P., Callahan, B. J., Proctor, D. M., Relman, D. A., and Holmes, S. P. (2018). The Block Bootstrap Method for Longitudinal Microbiome Data. arXiv:1809.01832 [stat].

Jenatton, R., Mairal, J., Obozinski, G., and Bach, F. R. (2010). Proximal methods for sparse hierarchical dictionary learning. In *ICML*, volume 1, page 2. Citeseer.

Jin, I. H. and Jeon, M. (2019). A Doubly Latent Space Joint Model for Local Item and Person Dependence in the Analysis of Item Response Data. *Psychometrika*, 84(1):236–260.

Johnson, B. A., Lin, D. Y., and Zeng, D. (2008). Penalized Estimating Functions and Variable Selection in Semiparametric Regression Models. *Journal of the American Statistical Association*, 103(482):672–680.

Kabbara, A., Khalil, M., El-Falou, W., Eid, H., and Hassan, M. (2016). Functional Brain Connectivity as a New Feature for P300 Speller. *PLoS ONE*, 11(1):e0146282.

Kaper, M., Meinicke, P., Grossekathoefer, U., Lingner, T., and Ritter, H. (2004). BCI competition 2003-data set IIb: support vector machines for the P300 speller paradigm. *IEEE Transactions on Biomedical Engineering*, 51(6):1073–1076.

Kass, R. E. and Raftery, A. E. (1995). Bayes Factors. *Journal of the American Statistical Association*, 90(430):773–795.

Kim, D. I., Hughes, M. C., and Sudderth, E. B. (2012). The nonparametric metadata dependent relational model. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1411–1418.

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980. arXiv: 1412.6980.

Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs, stat].

Knowles, D. A. and Minka, T. (2011). Non-Conjugate Variational Message Passing for Multinomial and Binary Regression. In *Advances in Neural Information Processing Systems*, pages 1701–1709.

Kong, D., Bondell, H. D., and Wu, Y. (2015). Domain selection for the varying coefficient model via local polynomial regression. *Computational Statistics & Data Analysis*, 83:236–250.

Koskinen, J. H., Robins, G. L., Wang, P., and Pattison, P. E. (2013). Bayesian analysis for partially observed network data, missing ties, attributes and actors. *Social Networks*, 35(4):514 – 527.

Krusienski, D. J., Sellers, E. W., McFarland, D. J., Vaughan, T. M., and Wolpaw, J. R. (2008). Toward enhanced P300 speller performance. *Journal of Neuroscience Methods*, 167(1):15–21.

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., and Lance, B. J. (2018). EEGNet: a compact convolutional neural network for EEG-based brain-computer interfaces. *Journal of Neural Engineering*, 15(5):056013.

Lee, E. R. and Mammen, E. (2016). Local linear smoothing for sparse high dimensional varying coefficient models. *Electronic Journal of Statistics*, 10(1):855–894.

Lee, W., McCormick, T. H., Neil, J., Sodja, C., and Cui, Y. (2021). Anomaly detection in large scale networks with latent space models. *Technometrics*, pages 1–23.

Leoni, J., Strada, S. C., Tanelli, M., Jiang, K., Brusa, A., and Proverbio, A. M. (2021). Automatic stimuli classification from ERP data for augmented communication via Brain-Computer Interfaces. *Expert Systems with Applications*, 184:115572.

Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford Large Network Dataset Collection.

Leskovec, J. and Mcauley, J. (2012). Learning to Discover Social Circles in Ego Networks. In *Advances in Neural Information Processing Systems*, volume 25, pages 539–547.

Li, F., Chen, B., Li, H., Zhang, T., Wang, F., Jiang, Y., Li, P., Ma, T., Zhang, R., Tian, Y., Liu, T., Guo, D., Yao, D., and Xu, P. (2016). The Time-Varying Networks in P300: A Task-Evoked EEG Study. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(7):725–733.

Ligtenberg, T. J., Bikker, F. J., Groenink, J., Tornoe, I., Leth-Larsen, R., Veerman, E. C., Nieuw Amerongen, A. V., and Holmskov, U. (2001). Human salivary agglutinin binds to lung surfactant protein-D and is identical with scavenger receptor protein gp-340. *The Biochemical Journal*, 359(Pt 1):243–248.

Lin, W.-C. and Tsai, C.-F. (2020). Missing value imputation: a review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, 53(2):1487–1509.

Lin, Z., Cao, J., Wang, L., and Wang, H. (2017). Locally Sparse Estimator for Functional Linear Regression Models. *Journal of Computational and Graphical Statistics*, 26(2):306–318.

Little, R. J. and Rubin, D. B. (2019). *Statistical analysis with missing data, third edition*. John Wiley & Sons.

Liu, B. and Zhu, J. (2019). Generative Link Prediction for Incomplete Networks with Node Features. In *Statistical Learning for Networks with Node Features*. University of Michigan.

Liu, D., Bai, H.-Y., Li, H.-J., and Wang, W.-J. (2014). Semi-supervised community detection using label propagation. *International Journal of Modern Physics B*, 28(29):1450208.

Liu, F., Liu, A., Lu, X., Zhang, Z., Xue, Y., Xu, J., Zeng, S., Xiong, Q., Tan, H., He, X., Xu, W., Sun, Y., and Xu, C. (2019). Dysbiosis signatures of the microbial profile in tissue from bladder cancer. *Cancer Medicine*, 8(16):6904–6914.

Liu, H., Jin, I. H., and Zhang, Z. (2018). Structural Equation Modeling of Social Networks: Specification, Estimation, and Application. *Multivariate Behavioral Research*, 53(5):714–730.

Liu, H., Jin, I. H., Zhang, Z., and Yuan, Y. (2021a). Social Network Mediation Analysis: A Latent Space Approach. *Psychometrika*, 86(1):272–298.

Liu, W., Lin, H., Zheng, S., and Liu, J. (2021b). Generalized Factor Model for Ultra-High Dimensional Correlated Variables with Mixed Types. *Journal of the American Statistical Association*, 0(0):1–17.

Liu, Y. and Chen, Y. (2021). Variational Inference for Latent Space Models for Dynamic Networks. *arXiv preprint arXiv:2105.14093*.

Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., and Yger, F. (2018). A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. *Journal of Neural Engineering*, 15(3):031005.

Luo, C., Li, F., Li, P., Yi, C., Li, C., Tao, Q., Zhang, X., Si, Y., Yao, D., Yin, G., Song, P., Wang, H., and Xu, P. (2022). A survey of brain network analysis by electroencephalographic signals. *Cognitive Neurodynamics*, 16(1):17–41.

Luo, D., Ziebell, S., and An, L. (2017). An informative approach on differential abundance analysis for time-course metagenomic sequencing data. *Bioinformatics*, 33(9):1286–1292.

Luts, J. and Wand, M. P. (2015). Variational Inference for Count Response Semiparametric Regression. *Bayesian Analysis*, 10(4):991–1023.

Ma, T., Li, Y., Huggins, J. E., Zhu, J., and Kang, J. (2022). Bayesian Inferences on Neural Activity in EEG-Based Brain-Computer Interface. *Journal of the American Statistical Association*, 117(539):1122–1133.

Ma, Z., Ma, Z., and Yuan, H. (2020). Universal Latent Space Model Fitting for Large Networks with Edge Covariates. *Journal of Machine Learning Research*, 21(4):1–67.

Mariadassou, M., Robin, S., and Vacher, C. (2010). Uncovering latent structure in valued graphs: a variational approach. *The Annals of Applied Statistics*, 4(2):715–742.

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

McLean, M. W. and Wand, M. P. (2019). Variational Message Passing for Elaborate Response Regression Models. *Bayesian Analysis*, 14(2):371–398.

Medeiros, M. C. d., The, S., Bellile, E., Russo, N., Schmitd, L., Danella, E., Singh, P., Banerjee, R., Bassis, C., Murphy, G. R., Sartor, M. A., Lombaert, I., Schmidt, T. M., Eisbruch, A., Murdoch-Kinch, C. A., Rozek, L., Wolf, G. T., Li, G., Chen, G. Y., and D'Silva, N. J. (2023). Salivary microbiome changes distinguish response to chemoradiotherapy in patients with oral cancer. *Microbiome*, 11(1):1–23.

Metwally, A. A., Yang, J., Ascoli, C., Dai, Y., Finn, P. W., and Perkins, D. L. (2018). MetaLonDA: a flexible R package for identifying time intervals of differentially abundant features in metagenomic longitudinal studies. *Microbiome*, 6(1):32.

Metwally, A. A., Zhang, T., Wu, S., Kellogg, R., Zhou, W., Contrepois, K., Tang, H., and Snyder, M. (2022). Robust identification of temporal biomarkers in longitudinal omics studies. *Bioinformatics*, 38(15):3802–3811.

Miller, K., Jordan, M., and Griffiths, T. (2009). Nonparametric latent feature models for link prediction. *Advances in Neural Information Processing Systems*, 22:1276–1284.

Minka, T. (2005). Divergence measures and message passing. Technical report, Technical report, Microsoft Research.

Monahan, J. H. and Stefanski, L. A. (1989). Normal Scale Mixture Approximations to the Logistic Distribution with Applications. Technical report, Department of Statistics, North Carolina State University.

Montiel Olea, J. L. and Plagborg-Møller, M. (2019). Simultaneous confidence bands: Theory, implementation, and an application to SVARs. *Journal of Applied Econometrics*, 34(1):1–17.

Murray, I., Adams, R., and MacKay, D. (2010). Elliptical slice sampling. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 541–548. JMLR Workshop and Conference Proceedings. ISSN: 1938-7228.

Nardi, Y. and Rinaldo, A. (2008). On the asymptotic properties of the group lasso estimator for linear models. *Electronic Journal of Statistics*, 2(none):605–633.

Noh, H. S. and Park, B. U. (2010). Sparse Varying Coefficient Models for Longitudinal Data. *Statistica Sinica*, 20(3):1183–1202.

Nolan, T. H. and Wand, M. P. (2017). Accurate logistic variational message passing: algebraic and numerical details. *Stat*, 6(1):102–112.

Ogunrinola, G. A., Oyewale, J. O., Oshamika, O. O., and Olasehinde, G. I. (2020). The Human Microbiome and Its Impacts on Health. *International Journal of Microbiology*, 2020:e8045646.

Okumus, H. and Aydemir, O. (2017). Random forest classification for brain computer interface applications. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4.

Olsen, I. and Yilmaz, O. (2019). Possible role of Porphyromonas gingivalis in orodigestive cancers. *Journal of Oral Microbiology*, 11(1):1563410.

Ouzienko, V. and Obradovic, Z. (2014). Imputation of missing links and attributes in longitudinal social surveys. *Machine learning*, 95(3):329–356.

Parikh, N., Boyd, S., and others (2014). Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239. Publisher: Now Publishers, Inc.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Paulson, J. N., Talukder, H., and Bravo, H. C. (2017). Longitudinal differential abundance analysis of microbial marker-gene surveys using smoothing splines.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Platt, J. and others (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74. Publisher: Cambridge, MA.

Poignard, B. (2020). Asymptotic theory of the adaptive Sparse Group Lasso. *Annals of the Institute of Statistical Mathematics*, 72(1):297–328.

Polich, J. (2007). Updating P300: An integrative theory of P3a and P3b. *Clinical Neurophysiology*, 118(10):2128–2148.

Preti, M. G., Bolton, T. A., and Van De Ville, D. (2017). The dynamic functional connectome: State-of-the-art and perspectives. *NeuroImage*, 160:41–54.

Reichhardt, M. P., Holmskov, U., and Meri, S. (2017). SALSA-A dance on a slippery floor with changing partners. *Molecular Immunology*, 89:100–110.

Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In *IEEE International Conference on Neural Networks*, pages 586–591 vol.1.

Roberts, G. O. and Tweedie, R. L. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363.

Rubenstein, J. H., Fontaine, S., MacDonald, P. W., Burns, J. A., Evans, R. R., Arasim, M. E., Chang, J. W., Firsht, E. M., Hawley, S. T., Saini, S. D., and others (2023). Predicting Incident Adenocarcinoma of the Esophagus or Gastric Cardia Using Machine Learning of Electronic Health Records. *Gastroenterology*, 165(6):1420–1429. Publisher: Elsevier.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Rubin-Delanchy, P., Cape, J., Tang, M., and Priebe, C. E. (2022). A Statistical Interpretation of Spectral Embedding: The Generalised Random Dot Product Graph. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(4):1446–1473.

Rushby, J. A., Barry, R. J., and Johnstone, S. J. (2002). Event-related potential correlates of serial-position effects during an elaborative memory test. *International Journal of Psychophysiology*, 46(1):13–27.

Salter-Townshend, M. and Murphy, T. B. (2013). Variational Bayesian inference for the latent position cluster model for network data. *Computational Statistics & Data Analysis*, 57(1):661–671.

Saul, L. and Jordan, M. (1998). A Mean Field Learning Algorithm for Unsupervised Neural Networks. In Jordan, M. I., editor, *Learning in Graphical Models*, NATO ASI Series, pages 541–554. Springer Netherlands, Dordrecht.

Schalk, G., McFarland, D., Hinterberger, T., Birbaumer, N., and Wolpaw, J. (2004). BCI2000: a general-purpose brain-computer interface (BCI) system. *IEEE Transactions on Biomedical Engineering*, 51(6):1034–1043.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. (2008). Collective Classification in Network Data. *AI Magazine*, 29(3):93–93.

Sewell, D. K. and Chen, Y. (2017). Latent space approaches to community detection in dynamic networks. *Bayesian Analysis*, 12(2):351–377.

Shields-Cutler, R. R., Al-Ghalith, G. A., Yassour, M., and Knights, D. (2018). SplinectomeR Enables Group Comparisons in Longitudinal Microbiome Studies. *Frontiers in Microbiology*, 9.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A Sparse-Group Lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245.

Singh, P., Banerjee, R., Piao, S., Costa de Medeiros, M., Bellile, E., Liu, M., Damodaran Puthiya Veettil, D., Schmitd, L. B., Russo, N., Danella, E., Inglehart, R. C., Pineault, K. M., Wellik, D. M., Wolf, G., and D'Silva, N. J. (2021). Squamous cell carcinoma subverts adjacent histologically normal epithelium to promote lateral invasion. *The Journal of Experimental Medicine*, 218(6):e20200944.

Song, Z., Yang, X., Xu, Z., and King, I. (2022). Graph-Based Semi-Supervised Learning: A Comprehensive Review. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21.

Staicu, A.-M., Lahiri, S. N., and Carroll, R. J. (2015). Significance tests for functional data with complex dependence structure. *Journal of Statistical Planning and Inference*, 156:1–13.

Stige, S., Fjell, A. M., Smith, L., Lindgren, M., and Walhovd, K. B. (2007). The Development of Visual P3a and P3b. *Developmental Neuropsychology*, 32(1):563–584.

Sweet, T. M. (2015). Incorporating Covariates Into Stochastic Blockmodels. *Journal of Educational and Behavioral Statistics*, 40(6):635–664.

Thompson, D. E., Gruis, K. L., and Huggins, J. E. (2014). A plug-and-play brain-computer interface to operate commercial assistive technology. *Disability and rehabilitation. Assistive technology*, 9(2):144–150.

Tian, J. (2015). Missing at Random in Graphical Models. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38, pages 977–985. PMLR.

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525.

Tsai, K., Kolar, M., and Koyejo, O. (2022). A Nonconvex Framework for Structured Dynamic Covariance Recovery. *Journal of Machine Learning Research*, 23(200):1–91.

Tu, C. Y., Park, J., and Wang, H. (2020). Estimation of Functional Sparsity in Nonparametric Varying Coefficient Models for Longitudinal Data Analysis. *Statistica Sinica*, 30(1):439–465.

Van Buuren, S., Boshuizen, H. C., and Knook, D. L. (1999). Multiple imputation of missing blood pressure covariates in survival analysis. *Statistics in medicine*, 18(6):681–694.

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.

Vehtari, A., Simpson, D., Gelman, A., Yao, Y., and Gabry, J. (2022). Pareto Smoothed Importance Sampling.

Wand, M. P. (2017). Fast Approximate Inference for Arbitrarily Large Semiparametric Regression Models via Message Passing. *Journal of the American Statistical Association*, 112(517):137–168.

Wang, H. and Kai, B. (2015). Functional Sparsity: Global Versus Local. *Statistica Sinica*, 25(4):1337–1354.

Wang, H. and Xia, Y. (2009). Shrinkage Estimation of the Varying Coefficient Model. *Journal of the American Statistical Association*, 104(486):747–757.

Wang, L., Li, H., and Huang, J. Z. (2008). Variable Selection in Nonparametric Varying-Coefficient Models for Analysis of Repeated Measurements. *Journal of the American Statistical Association*, 103(484):1556–1569.

Wang, L., Zhou, J., and Qu, A. (2012). Penalized Generalized Estimating Equations for High-Dimensional Longitudinal Data Analysis. *Biometrics*, 68(2):353–360.

Wang, S., Paul, S., and De Boeck, P. (2023). Joint Latent Space Model for Social Networks with Multivariate Attributes. *Psychometrika*.

Wang, Z., Magnotti, J., Beauchamp, M. S., and Li, M. (2022). Functional group bridge for simultaneous regression and support estimation. *Biometrics*.

Watanabe, S. (2010). Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory. *The Journal of Machine Learning Research*, 11:3571–3594.

Wermuth, N. and Cox, D. R. (2005). Statistical Dependence and Independence. In *Encyclopedia of Biostatistics*. John Wiley & Sons, Ltd. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/0470011815.b2a15154.

Winn, J. and Bishop, C. M. (2005). Variational Message Passing. *Journal of Machine Learning Research*, 6(23):661–694.

Xing, E. P., Fu, W., and Song, L. (2010). A state-space mixed membership blockmodel for dynamic network tomography. *The Annals of Applied Statistics*, pages 535–566.

Xuan, C., Shamonki, J. M., Chung, A., DiNome, M. L., Chung, M., Sieling, P. A., and Lee, D. J. (2014). Microbial Dysbiosis Is Associated with Human Breast Cancer. *PLOS ONE*, 9(1):e83744.

Xue, L. and Qu, A. (2012). Variable selection in high-dimensional varying-coefficient models with global optimality. *The Journal of Machine Learning Research*, 13(null):1973–1998.

Yang, T., Chi, Y., Zhu, S., Gong, Y., and Jin, R. (2011). Detecting communities and their evolutions in dynamic social networks–a Bayesian approach. *Machine Learning*, 82(2):157–189.

Yin, H., Benson, A. R., Leskovec, J., and Gleich, D. F. (2017). Local Higher-Order Graph Clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 555–564, New York, NY, USA. Association for Computing Machinery.

Yoon, Y., Kim, G., Jeon, B.-N., Fang, S., and Park, H. (2021). Bifidobacterium Strain-Specific Enhances the Efficacy of Cancer Therapeutics in Tumor-Bearing Mice. *Cancers*, 13(5):957.

Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. (2019). Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):2008–2026.

Zhang, C.-H. and Zhang, S. S. (2014). Confidence Intervals for Low Dimensional Parameters in High Dimensional Linear Models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(1):217–242.

Zhang, X., Xu, G., and Zhu, J. (2022). Joint latent space models for network data with high-dimensional node variables. *Biometrika*, 109(3):707–720.

Zhong, R., Zhang, C., and Zhang, J. (2022). Locally sparse estimator of generalized varying coefficient model for asynchronous longitudinal data. arXiv:2206.04315 [stat].

Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation.

Zou, H. (2006). The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, 101(476):1418–1429.