

High-dimensional Multimodal Bayesian Learning

Mohamed Salem

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Statistics

Inyoung Kim, Chair
Christopher T. Franck
Robert B. Gramacy
Jennifer Van Mullekom

Dec 02, 2024
Blacksburg, Virginia

Keywords: Gaussian Process, High Dimensional Data, Variable Selection, Variational
Inference, Uncertainty Quantification

Copyright 2024, Mohamed Salem

High-dimensional Multimodal Bayesian Learning

Mohamed Salem

(ABSTRACT)

High-dimensional datasets are fast becoming a cornerstone across diverse domains, fueled by advancements in data-capturing technology like DNA sequencing, medical imaging techniques, and social media. This dissertation delves into the inherent opportunities and challenges posed by these types of datasets. We develop three Bayesian methods: (1) Multilevel Network Recovery for Genomics, (2) Network Recovery for Functional data, and (3) Bayesian Inference in Transformer-based Models. Chapter 2 in our work examines a two-tiered data structure; to simultaneously explore the variable selection and identify dependency structures among both higher and lower-level variables, we propose a multi-level nonparametric kernel machine approach, utilizing variational inference to jointly identify multi-level variables as well as build the network. Chapter 3 addresses the development of a simultaneous selection of functional domain subsets, selection of functional graphical nodes, and continuous response modeling given both scalar and functional covariates under semiparametric, nonadditive models, which allow us to capture unknown, possibly nonlinear, interaction terms among high dimensional functional variables. In Chapter 4, we extend our investigation of leveraging structure in high dimensional datasets to the relatively new transformer architecture; we introduce a new penalty structure to the Bayesian classification transformer, leveraging the multi-tiered structure of the transformer-based model. This allows for increased, likelihood-based regularization, which is needed given the high-dimensional nature of our motivating dataset. This new regularization approach allows us to integrate Bayesian inference via variational approximations into our transformer-based model and improves the calibration of probability estimates.

High-dimensional Multimodal Bayesian Learning

Mohamed Salem

(GENERAL AUDIENCE ABSTRACT)

In today's data-driven landscape, high-dimensional datasets have emerged as a cornerstone across diverse domains, fueled by advancements in technology like sensor networks, genomics, and social media platforms. This dissertation delves into the inherent opportunities and challenges posed by these datasets, emphasizing their potential for uncovering hidden patterns and correlations amidst their complexity. As high-dimensional datasets proliferate, researchers face significant challenges in effectively analyzing and interpreting them. This research focuses on leveraging Bayesian methods as a robust approach to address these challenges. Bayesian approaches offer unique advantages, particularly in handling small sample sizes and complex models. By providing robust uncertainty quantification and regularization techniques, Bayesian methods ensure reliable inference and model generalization, even in the face of sparse or noisy data. Furthermore, this work examines the strategic integration of structured information as a regularization technique. By exploiting patterns and dependencies within the data, structured regularization enhances the interpretability and resilience of statistical models across various domains. Whether the structure arises from spatial correlations, temporal dependencies, or coordinated actions among covariates, incorporating this information enriches the modeling process and improves the reliability of the results. By exploring these themes, this research contributes to advancing the understanding and application of high-dimensional data analysis. Through a thorough examination of Bayesian methods and structured regularization techniques, this dissertation aims to support researchers in effectively navigating and extracting meaningful insights from the complex landscape of high-dimensional datasets.

Dedication

To my wife, my family, and my friends, for their unwavering belief in me and their enduring support through every moment.

Acknowledgments

I would like to express my deepest gratitude to my advisor, Dr. Inyoung Kim, whose patience, guidance, and understanding have been invaluable throughout this journey. Her support and encouragement have shaped both my academic growth and personal development, and for that, I am eternally thankful.

I am also sincerely grateful to my committee members, Professors Chris Franck, Bobby Gramacy, and Jennifer Van Mullekom. Your willingness to volunteer your time, provide thoughtful feedback, and address my many questions has been instrumental to my progress. I deeply appreciate your graciousness, flexibility, and generosity throughout this process.

To everyone in the Statistics department, I owe a tremendous debt of gratitude. From the professors and staff who contributed to my education to the department as a whole, your unwavering support has been a cornerstone of this work. Thank you for fostering an environment where students are encouraged to thrive and for giving me the opportunity to pursue my dreams.

Finally, I would like to thank my wife, my family, and my friends for being my ever-present support system. Your love, encouragement, and belief in me have been my constant source of strength, and I could not have accomplished this without you.

Contents

List of Figures	x
------------------------	----------

List of Tables	xvi
-----------------------	------------

1 Introduction	1
1.1 Asymmetric LaPlace Distribution	1
1.2 The Ising Graphical Prior	2
1.3 Kernel Machine Regression	3
1.4 Gaussian Process Regression	4
1.5 Bayesian Optimization	6
1.6 Variational Inference	7
1.7 Transformers	9
1.8 Outline of the Dissertation	9
2 Bayesian Multi-level Network Recovery Selection	12
2.1 Introduction	13
2.2 Multi-level Network Model	14
2.3 Multi-level Recovery Method	16
2.3.1 Kernel Weights Estimation	16
2.3.2 The Selection Parameters	19
2.4 Theoretical Properties	26
2.4.1 Notations	26
2.4.2 Assumptions	26
2.4.3 Asymptotic Results under Multi-level Network Model	27

2.5	Simulation	30
2.5.1	Comparison between MLNR based on F-MCMC and MNLR based on M-MCVI	31
2.5.2	Comparison of MLNR with alternatives	32
2.5.3	Simulation Results	34
2.5.4	Visualization for Multi-Level Network	42
2.6	Real Data Application: Gene-pathway networks associated with Type 2 Dia- betes Mellitus	43
2.7	Discussion	47
3	Bayesian Multi-Tasking Inference using High-dimensional Correlated Functional Variables	49
3.1	Introduction	50
3.1.1	Our contribution	51
3.1.2	Outline of the chapter	53
3.2	Model Structure	54
3.2.1	Semiparametric Joint Functional Kernel Graphical Model	54
3.3	Functional Domain Subspace Selection: $[a_p, b_p]$	56
3.4	Functional Node and Graph Selection: $\xi_p, p = 1, \dots, P$	59
3.4.1	Markov chain Monte Carlo Estimation	60
3.5	Functional Association Testing	65
3.6	Algorithm for Bayesian multitasking inference	68
3.7	Theoretical Properties	69
3.7.1	Notations	70
3.7.2	Incoherence Condition for Recovery of Sparsity	71
3.7.3	Regularity conditions	72
3.7.4	Asymptotic Results	72

3.8	Simulation	74
3.8.1	Setting	76
3.8.2	Simulation Results	79
3.8.3	Sensitivity analysis of lengthscale parameters	92
3.9	Functional magnetic resonance imaging (fMRI) data in the human brain contributing to the development of Autism Spectrum Disorder	94
3.10	Discussion	102
4	Bayesian Inference in Transformer-based Models	103
4.1	Introduction	104
4.1.1	Main contribution	107
4.1.2	Outline of the paper	108
4.2	Likelihood-guided Regularization in Attention-Based Models	108
4.2.1	Classification model	109
4.2.2	Variational Bayes for Weights	109
4.3	Algorithm	115
4.4	Statistical properties	116
4.5	Simulations	117
4.5.1	Setting	118
4.5.2	Simulation Result	120
4.6	Autism Brain Imaging Data Exchange	126
4.6.1	Preprocessing data	127
4.6.2	Small dataset analysis	127
4.7	Conclusion and discussion	132
5	Conclusions and Discussion	134
	Bibliography	137

Appendices	154
Appendix A Bayesian Multilevel Network Recovery	155
A.1 Variational Bayes Formula Derivations	155
A.1.1 Deriving the Variational Distribution	155
A.1.2 Deriving the formula for updating the subnode selection parameters .	158
Appendix B Bayesian Multitasking Inference Method	161
B.1 Variational Bayes Formula Derivations	162
B.1.1 Deriving the Variational Distribution	162
B.1.2 Deriving the formula for updating the subnode selection parameters .	164
Appendix C Bayesian Inference in Transformer-based Models	166
C.1 Attention based crossformer	166
C.1.1 Two-Stage Attention	166

List of Figures

2.1	Simulation results for the normal distribution noise, normal distribution likelihood element case: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso. . . .	38
2.2	Simulation results for the normal distribution noise, normal distribution likelihood set case: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso. . . .	39
2.3	Simulation results for the asymmetric LaPlace distribution noise, asymmetric LaPlace distribution likelihood element case: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso.	40
2.4	Simulation results for the asymmetric LaPlace distribution noise, asymmetric LaPlace distribution likelihood set case: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso.	41
2.5	Sample output from the algorithm showing the elements nested within the sets. Connected dots indicate both a connection among the elements and to the response variable. In this run, only set 3 was connected to the response variable. This is indicated by the connection of a subset of the elements of set 3 and the lack of connections between the elements of sets 1 and 2. We note that this visualization approach will be misleading when a single element is selected from a pathway; however, this is a highly unlikely scenario in the application for which this method was developed (gene and pathway selection)	43

2.6	Sample output from the Variational Bayes algorithm for estimating the expectation of the response, blood glucose level. It shows the elements nested within the sets. Connected dots indicate both a connection among the elements/sets and to the response variable. These results visualize the MLNR (MVN) results from tables 2.7, 2.8.	46
2.7	Sample output from the Variational Bayes algorithm for estimating the 0.5 quantile of the response, blood glucose level. It shows the elements nested within the sets. Connected dots indicate both a connection among the elements/sets and to the response variable. These results visualize the MLNR (ALD) results from tables 2.7, 2.8.	47
3.1	Comparison of method performance between BMIM-1 and BMIM-2 when domain selection is applied .vs. when domain selection is not applied. Data is generated using Case 1 (scalar on functional regression with a coefficient function). BMIM-I is our method with K_p^1 applied to the unreduced functional data; BMIM-II is our method with K_p^2 applied to the unreduced functional data. This figure corresponds to table 3.1 with P=10	83
3.2	Comparison of functional domain selection performance among BMIM, Fused Lasso, Lasso, FLM-Bspline (B=13), and FLM-Bspline (B=23), in terms of TPR, FPR, FPV, and F1-score. Data is generated using scalar on functional regression with a coefficient function. This figure corresponds to the "Domains" section of table 3.2 with P=10	84
3.3	Comparison of node selection performance among BMIM, Fused Lasso, Lasso, FLM-Bspline (B=13), and FLM-Bspline (B=23) in terms of TPR, FPR, PPV, F1-Score. Data is generated using scalar on functional linear regression with a coefficient function. This figure corresponds to the "Nodes" section of table 3.2 with P=10	85

3.4	Comparison of method performance between BMIM-1 and BMIM-2 when domain selection is applied vs. when domain selection is not applied. Data is generated using the nonparametric model. BMIM-I is our method with K_p^1 applied to the unreduced functional data; BMIM-II is our method with K_p^2 applied to the unreduced functional data data. This figure corresponds to table 3.5 with P=10	89
3.5	Comparison of functional domain selection performance among BMIM, Fused Lasso, Lasso, FLM-Bspline (B=13), and FLM-Bspline (B=23), in terms of TPR, FPR, FPV, and F1-score. Data is generated using a functional non-parametric model. This figure corresponds to the "Domains" section of table 3.6 with P=10	90
3.6	Comparison of node selection performance between BMIM, Fused Lasso, Lasso, Bspline Regression (B=13), and Bspline Regression (B=23). Data is generated using a functional nonparametric model. This figure corresponds to the "Nodes" section of table 3.6 with P=10	91
3.7	Sensitivity analysis for choice of lengthscale versus the significance of the functional covariates when all functional covariates are unrelated to the response, which is null Case. The boxplots represent p-value estimates across 100 simulations per lengthscale setting. The dashed red line represents the 0.05 threshold.	93
3.8	Sensitivity analysis for choice of lengthscale versus the significance of the functional covariates when some functional covariates are related to the response, which is the alternate case. The boxplots represent p-value estimates across 100 simulations per lengthscale setting. The dashed red line represents the 0.05 threshold.	94

3.9	(a) shows predefined brain networks and the parcellations that form them. (b) shows a generated sample that mimics the observed functional MRI signals generated from each parcellation in the network. Visualization of parcellations from https://github.com/ThomasYeoLab/CBIG/tree/master/stable_pro	
	Original network source is from Yeo et al. (2011)	96
3.10	P-values of null hypothesis on real data networks, red dashed line indicates 0.05 p-value threshold	97
3.11	Illustrative visualization of connectivity within the Control network.	100
3.12	Illustrative visualization of connectivity within the Default Mode network.	101
4.1	The figure shows the components in our model. The selection variables ξ determine which distribution in the spike-slab prior a weight, w , is sampled from.	114
4.2	Simulation results to investigate the performance of different model architectures under $N=20, 50, 100$ and $P=20$ using ACC , TPR , FPR , PPV , F₁ . Crossformer (Zhang and Yan, 2023); Bayesian Crossformer (Sankaranarayanan et al., 2022); Full-Ising Bayesian Crossformer and Approximate-Ising Bayesian Crossformer are our methods.	121
4.3	Negative log-likelihood on the training set. $N=20$. We observe a bump at the starting epoch of the Ising implementation indicating the disruption of training loss due to the implementation of a new mask. the training loss smooths out as the weights are optimized over the next few epochs.	122
4.4	Negative log-likelihood on the validation set. $N=20$. We observe a clear sharp drop in validation loss in the Ising epochs due to greater regularization leading to better generalizability of the trained weights to the validation dataset.	123
4.5	Negative log-likelihood on the validation set. $N=100$. There is virtually no difference in validation loss as the larger dataset size negates the benefits of additional regularization.	123

4.6	Negative log-likelihood on the testing set. $N=20$. We observe a clear sharp drop in testing loss in the Ising epochs due to greater regularization leading to better generalizability of the trained weights to the testing dataset.	124
4.7	Negative log-likelihood for the first few epochs on the testing set. $N=100$. There is virtually no difference in test loss as the larger dataset size negates the benefits of additional regularization.	124
4.8	Real data results of classification performance using different model architectures under $N=20, 50, 100$ and $P=24$ parcellations within the Default Mode network of the brain as identified by Schaefer et al. (2017). Crossformer (Zhang and Yan, 2023); Bayesian Crossformer (Sankararaman et al., 2022); Full-Ising Bayesian Crossformer and Approximate-Ising Bayesian Crossformer are our methods.	131
4.9	Real data results of classification performance of different model architectures under $N=20, 50, 100$ and $P=24$ parcellations within the Default Mode network of the brain as identified by Schaefer et al. (2017). Crossformer (Zhang and Yan, 2023); Bayesian Crossformer (Sankararaman et al., 2022); Full-Ising Bayesian Crossformer and Approximate-Ising Bayesian Crossformer are our methods. The final column indicates the number of observations excluded due to uncertainty in classification, defined as a probability between 0.45 and 0.55.	131

4.10	Model probability estimates for $N=20$. The red rectangular boxes highlight misclassified observations. We observe that the typical Vision transformer assigns almost perfect probabilities even to misclassified observations, the Bayesian transformer produces close results, implying the models are over-confident and are producing poorly calibrated probability estimates. Both versions of our proposed regularization show more uncertainty. The approximate method, being the stronger regularizer shows even more uncertainty.	132
C.1	Model Architecture highlighting the two-stage attention process and the layer-to-layer backpropagation of the Ising prior-based dropout masks. The masks are learned from the likelihood only in the last layer, then they are backpropagated to earlier layers as per the strength of the inter-layer connectivity.	170

List of Tables

2.1	Simulation results comparing selection results from the exact posterior estimation via McMC to the approximate posterior estimation results via VB estimation.	33
2.2	Simulation results comparing selection results from the exact posterior estimation via McMC to the approximate posterior estimation results via VB estimation.	34
2.3	Simulation results for the normal distribution noise, normal distribution likelihood: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso.	35
2.4	Simulation results for the asymmetric LaPlace distribution noise, normal distribution likelihood: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso. . . .	36
2.5	Simulation results for the normal distribution noise, asymmetric LaPlace distribution likelihood: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso. . . .	36
2.6	Simulation results for the asymmetric LaPlace distribution noise, asymmetric LaPlace distribution likelihood: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso.	37
2.7	A real application result using competing methods: comparison of competing methods. For methods that are only selected at the gene level, we consider any single gene selection as a pathway selection in this table.	44

2.8	A real application result using competing methods: comparison of competing methods. For methods that only select at the gene level, we consider only all gene selections as a pathway selection in this table.	45
2.9	A comparison between genes selected under the multivariate normal likelihood assumption and the asymmetric LaPlace likelihood assumption.	45
3.1	Comparison of different BMIM node selection settings under Case 1: BMIM-1: McMC-on- K_p^1 ; BMIM-2: McMC-on- K_p^2 . The table shows each setting's performance on the task of node selection when N=183 and P=10	80
3.2	Comparison against other valid methodologies under Case1: BMIM (under the best performing node selection setting); Fused Lasso; Lasso; GP-on-FPCA; FLM-BS(13); FLM-BS(23). FLM-BS(B)=FLM-Bspline with the B number of Bsplines. The table shows the chosen models' (1) domain selection performance metrics, (2) node selection performance metrics, (3) overall significance under the null (Type-I error), and (4) overall significance under the alternate (Power) when N=183 and P=10	81
3.3	Comparison of different BMIM node selection settings under Case 1: BMIM-1: McMC-on- K_p^1 ; BMIM-2: McMC-on- K_p^2 . The table shows each setting's performance on the task of node selection when N=183 and P=30	82
3.4	Comparison against other valid methodologies under Case1: BMIM: BMIM (under the best performing node selection setting); Fused Lasso: Fused Lasso; Lasso: Lasso; FPCA-GP: GP-on-FPCA; FLM-BS(13): FLM-Bspline (B=13); FLM-BS(23): FLM-Bspline (B=23). The table shows the chosen models' (1) domain selection performance metrics, (2) node selection performance metrics, (3) overall significance under the null (Type-I error), and (4) overall significance under the alternate (Power) when N=183 and P=30	82

3.5	Comparison of different BMIM node selection settings conditional on selected domains under Case 2: BMIM-1: $\text{McMC-on-}K_p^1$; BMIM-2: $\text{McMC-on-}K_p^2$. The table shows each setting's performance on the task of node selection when $N=183$ and $P=10$	86
3.6	Comparison against other valid methodologies under Case 2: BMIM: BMIM (under the best performing node selection setting); Fused Lasso: Fused Lasso; Lasso: Lasso; FPCA-GP: GP-on-FPCA; FLM-BS(13): FLM-BSpline (B=13); FLM-BS(23): FLM-BSpline (B=23). The table shows the chosen models' (1) domain selection performance metrics, (2) node selection performance metrics, (3) overall significance under the null (Type-I error), and (4) overall significance under the alternate (Power) when $N=183$ and $P=10$	87
3.7	Comparison of different BMIM node selection settings conditional on selected domains under Case 2: BMIM-1: $\text{McMC-on-}K_p^1$; BMIM-2: $\text{McMC-on-}K_p^2$. The table shows each setting's performance on the task of node selection when $N=183$ and $P=30$	87
3.8	Comparison against other valid methodologies under Case 2: BMIM: BMIM (under the best performing node selection setting); Fused Lasso: Fused Lasso; Lasso: Lasso; FPCA-GP: GP-on-FPCA; FLM-BS(13): FLM-BSpline (B=13); FLM-BS(23): FLM-BSpline (B=23). The table shows the chosen models' (1) domain selection performance metrics, (2) node selection performance metrics, (3) overall significance under the null (Type-I error), and (4) overall significance under the alternate (Power) when $N=183$ and $P=30$	88
3.9	Selected Nodes within available networks using real data, LH denotes the left hemisphere of the brain, RH denotes the right hemisphere of the brain. . . .	98
3.10	Selected Nodes within available networks using real data, LH denotes the left hemisphere of the brain, RH denotes the right hemisphere of the brain. . . .	99

4.1	Simulation results to investigate the performance of different model architectures under N=20 and P=20 using ACC , TPR , FPR , PPV , F₁ ; Crossformer (Zhang and Yan, 2023); Bayesian Crossformer (Sankararaman et al., 2022); Full-Ising Bayesian Crossformer and Approximate-Ising Bayesian Crossformer are our methods.	125
4.2	Simulation results to investigate the performance of different model architectures under N=50 and P=20 using ACC , TPR , FPR , PPV , F₁ ; Crossformer (Zhang and Yan, 2023); Bayesian Crossformer (Sankararaman et al., 2022); Full-Ising Bayesian Crossformer and Approximate-Ising Bayesian Crossformer are our methods.	125
4.3	Simulation results to investigate the performance of different model architectures under N=100 and P=20 using ACC , TPR , FPR , PPV , F₁ ; Crossformer (Zhang and Yan, 2023); Bayesian Crossformer (Sankararaman et al., 2022); Full-Ising Bayesian Crossformer and Approximate-Ising Bayesian Crossformer are our methods.	126
4.4	Real data results of classification performance using different model architectures under N=20,50,100 and P=24 parcellations within the Default Mode network of the brain as identified by Schaefer et al. (2017); Vision transformer (Dosovitskiy et al., 2020); Bayesian transformer (Sankararaman et al., 2022); Full-Ising Bayesian transformer and Approximate-Ising Bayesian transformer are our methods.	129

4.5	Real data results of classification performance of different model architectures under N=20,50,100 and P=24 parcellations within the Default Mode network of the brain as identified by Schaefer et al. (2017); Vision transformer (Dosovitskiy et al., 2020); Bayesian transformer (Sankararaman et al., 2022); Full-Ising Bayesian transformer and Approximate-Ising Bayesian transformer are our methods; The final column indicates the number of observations excluded due to uncertainty in classification, defined as a probability between 0.45 and 0.55	130
-----	--	-----

Chapter 1

Introduction

In this chapter, we provide overviews of asymmetric Laplace distribution, Ising prior, non-parametric regression models, Bayesian optimization, Variational Bayes, and transformers which are fundamental concepts for Chapters 2-4.

1.1 Asymmetric LaPlace Distribution

Before introducing the Asymmetric Laplace distribution, we first review both the original, symmetric LaPlace distribution and quantile regression. The concept of regression, in general, is concerned with achieving as close as possible to an approximation for some target response variable in the space of available covariates. In other words, we aim to approximate or predict the response using given covariates. Regression models with a continuous variable as the target of interest typically assume a noisy response $\underline{\mathbf{y}}$ such that

$$\underline{\mathbf{y}} = f(\mathbf{X}) + \epsilon,$$

where $f(X)$ is some unknown function, and ϵ is epistemic, irreducible noise in the response. When assuming normally distributed noise terms (also often called error terms) and taking the minimization of mean squared error ($N^{-1} \sum_i (y_i - f(\underline{\mathbf{x}}_i))^2$) to be our goal, we obtain an estimate of the expected value of the response, $E[\underline{\mathbf{y}}|\mathbf{X}]$, in other words, the mean.

Quantile regression extends the regression concept to any quantile of the response variable. Beyond modelling a quantile for the sake of interest in a specific quantile, quantile

regression may avoid some outlier-induced pathologies. Furthermore, quantile regression targets the minimization of the mean absolute deviations from the response ($N^{-1} \sum_i |y_i - f(\mathbf{x}_i)|$) thereby making the approach more robust to violations of distributional assumptions (Koenker and Bassett, 1978).

As with most statistical models, there exists a Bayesian approach to performing quantile regression. One such possible approach proceeds by imposing an asymmetric LaPlace (Yu and Zhang, 2005) likelihood on the response. While this contradicts the distribution-free estimation procedure of the original, non-Bayesian approach, it has been shown to work well (Sriram et al., 2013). The probability density function of the asymmetric Laplace distribution is defined as

$$f(y|\mu, \sigma, \kappa) = \frac{\kappa(1 - \kappa)}{\sigma} \exp \left[- \frac{(y - \mu)}{\sigma} [\kappa - \mathbb{I}(y \leq \mu)] \right],$$

where μ is a location parameter, σ is a scale parameter, κ is a skew parameter, and $\mathbb{I}$ is the indicator function.

1.2 The Ising Graphical Prior

A central theme of this work is leveraging covariate connections as a model regularizer to overcome the challenges associated with parameter estimation in high-dimensional datasets. The Ising graphical prior is one such regularization method. A special case of the Boltzmann distribution, the Ising graphical prior has support over the Cartesian product $\{-1, 1\}^D$, where D represents the number of covariates in the model. The probability density function of the prior tends to place greater mass on configurations where “highly connected” random variables take on the same values, where

$$f(\underline{\xi}|\mathbf{X}) = \frac{1}{Z} \exp \left[\sum_{i < j} \xi_i \cdot \Omega(\mathbf{X})_{ij} \cdot \xi_j \right],$$

where each element ξ is an entry in the vector $\underline{\xi}$, $\Omega(\mathbf{X})$ is a matrix representing the prior expected strength of the connections between each element ξ and may be a function of covariates $\underline{\mathbf{X}}$, and Z is a normalizing constant representing the sum of the numerator over all possible 2^D configurations. While its form offers nice conjugacy with a Gaussian likelihood, the Ising prior is a discrete multivariate probability distribution which gives rise to its own set of computational difficulties, including the intractability of the normalizing constant for non-trivial problems, as well as typical difficulties associated with combinatorial optimization.

1.3 Kernel Machine Regression

The term kernel regression is often used to refer to several distinct concepts in statistics literature; perhaps it is most commonly associated with the Nadaraya-Watson model (Nadaraya, 1964, Watson, 1964), a method that approximates a regression function as a weighted sum of existing observations, with the weights being determined by a kernel function driven by some distance metric from the desired prediction location. This is derived by replacing the joint density function of the response and the covariates with a kernel density estimator. The Kernel Machine regression we utilize here is often called linear regression using a Gaussian basis function (Bishop, 2006). The Kernel Machine regression we utilize here is derived from a Bayesian perspective (Bobb et al., 2014), although equivalent derivations in an ordinary least squares fashion using a linear non-negative garrote also exist (Fang et al., 2016). Therefore, our models take the form,

$$\begin{aligned}\underline{\mathbf{y}}|\mathbf{K}, \underline{\boldsymbol{\alpha}}, \mathbf{V} &\sim \mathcal{N}(\mathbf{K}\underline{\boldsymbol{\alpha}}, \mathbf{V}), \\ \underline{\boldsymbol{\alpha}} &\sim \mathcal{N}(\underline{\mathbf{m}}_0, \mathbf{V}_0),\end{aligned}$$

where $\mathbf{V}$ is an unknown covariance matrix and $(\underline{\mathbf{m}}_0, \mathbf{V}_0)$ are a hyper parameters.

The weights may then be estimated as the maximum-a-posteriori of the resulting poste-

rior or sampled in a Monte Carlo setup.

1.4 Gaussian Process Regression

Gaussian process regression (GPR) can be considered a non-parametric approach to modelling functions as weighted combinations of the observations. One of the earliest and most cited references for this approach comes from the geostatistics literature, particularly Mathéron (1963). Works that were seminal in crystallizing the idea of a Gaussian prior over functions include Kushner (1964), which modelled observations as a Gaussian random function while introducing a probabilistic approach to finding maxima; Moćkus (1975) while also seeking to optimize functions, proposed to model them as realizations of a stochastic Gaussian field; and O’Hagan (1978) which offers what can be argued to be the complete Bayesian specification of a Gaussian process regression as a prior over functions.

For the remainder of this section, we present a brief mathematical overview of Gaussian process regression adapted from Rasmussen and Williams (2005). First, we define a Gaussian process as a collection of random processes that follow a joint Gaussian distribution

$$f(\underline{x}) \sim \mathcal{GP}(\mu(\underline{x}), k(\underline{x}, \underline{x}')),$$

where these random variables are the value the function takes on at a given $\underline{x}$, one characteristic that differentiates Gaussian processes from simple Gaussian distributions is that the function, f , is defined over infinitely many $\underline{x}$. Another feature of Gaussian processes is consistency, which in this context simply means that if $(y_1, y_2) \sim MVN(\underline{\mu}, \underline{\Sigma}) \implies y_1 \sim N(\underline{\mu}, \Sigma_{11})$. Given the above characteristics, if we choose an initial set of inputs, $\mathbf{X}_0$, we have

$$\underline{f}_0(\underline{x}_0) \sim \mathcal{GP}(\underline{\mu}_0(\underline{x}_0), \mathbf{K}(\mathbf{X}_0, \mathbf{X}_0')).$$

We are now left to choose both the function $\mu_0(\underline{x}_0)$ and the function $k(\mathbf{X}_0, \mathbf{X}'_0)$. It is often the case that the former is chosen as a constant or a linear function of the given inputs. The latter, however, calls for deeper discussion. This function $k(.,.)$, describes a kernel used to estimate the covariance matrix as a function of the inputs. As such, the choice of $k(.,.)$ plays a central role in determining the degree of variability in our random function. In our application, we use the *squared exponential covariance function* (also known as the Radial Basis Function)

$$k(\underline{x}, \underline{x}') = \exp \left[-\frac{1}{2} \|\underline{x} - \underline{x}'\|^2 \right].$$

Now that a choice of covariance function has been made, we can show how one iteration works based on our observed data $\mathbf{X}_0$. Consider again the characteristics of a Gaussian process. Given $\mathbf{X}_0$, we know that

$$\begin{bmatrix} \underline{f}' \\ \underline{f}_0 \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \underline{\mu}(\underline{x}') \\ \underline{\mu}_0(\underline{x}_0) \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) & \mathbf{K}(\mathbf{X}, \mathbf{X}_0) \\ \mathbf{K}(\mathbf{X}_0, \mathbf{X}) & \mathbf{K}(\mathbf{X}_0, \mathbf{X}_0) \end{bmatrix} \right).$$

We can use this relation to sample from the conditional distribution of $\underline{f}'$ given $\underline{f}_0$ (and given observed $\mathbf{X}_0$ and proposed $\mathbf{X}$)

$$\underline{f}' | \underline{f}_0, \mathbf{X}_0, \mathbf{X} \sim \mathcal{N} \left(\underline{\mu}^*, \Sigma^* \right),$$

where

$$\begin{aligned} \underline{\mu}^* &= \underline{\mu}(\underline{x}') + \mathbf{K}(\mathbf{X}, \mathbf{X}_0) [\mathbf{K}(\mathbf{X}_0, \mathbf{X}_0)]^{-1} (\underline{f}_0 - \underline{\mu}_0(\underline{x}_0)), \\ \Sigma^* &= \mathbf{K}(\mathbf{X}, \mathbf{X}) - \mathbf{K}(\mathbf{X}, \mathbf{X}_0) [\mathbf{K}(\mathbf{X}_0, \mathbf{X}_0)]^{-1} \mathbf{K}(\mathbf{X}, \mathbf{X}_0)^T. \end{aligned}$$

As such, we can probabilistically exclude functions that do not match our previously observed data points.

1.5 Bayesian Optimization

Bayesian optimization is an optimization approach that, unlike many popular optimization algorithms, is not gradient-based. In fact, it is exactly this derivative-free characteristic that makes Bayesian optimization a popular choice for optimizing black-box functions. At its core, Bayesian optimization is a sampling procedure that works by first building an approximation of the function under examination based on the function's evaluation at a limited set of observations sampled from its domain, an approach known as surrogate modelling. This approximation aspect is useful not only in cases where the function under examination is unknown but also when acquiring evaluations of said function is prohibitively costly or infeasible.

The optimization procedure produces a ranking of sorts over the input space, which guides our sampling process in choosing what portions of the function to acquire more information about based on a metric - the acquisition function - that balances between uncertainty reduction in sparsely sampled domain subsets, and quality improvement in domain subsets that show promise for yielding solution candidates. The acquisition function decides which point in the input space is next in line to be evaluated by our surrogate model. It does this by managing a trade-off between reducing uncertainty surrounding the function's value in some weakly explored subset of the sample space and the potential gain from over-examining subsets of the sample space close to observations that, at the algorithm's current iteration, relatively outperform other observations as candidates for the optimum. In the literature, this is known as "exploration vs. exploitation".

Over the years, many such acquisition functions have been proposed: the probability of improvement (Kushner, 1964), expected improvement (Jones et al., Moćkus, 1975), and Thompson Sampling (Thompson, 1933), among others. We will only examine the mathematics behind expected improvement under a Gaussian process surrogate, which is utilized in our simulation.

It is common in Bayesian optimization literature to choose surrogates that lead to tractable closed-form expressions for the estimates of the function and any associated uncertainty measures. The latter offers a strong argument for the abundant use of Gaussian process regressions as surrogates in the literature. We do note, however, that while the Bayesian optimization procedure requires that we are able to quantify the uncertainty associated with the surrogate model estimates, there is no requirement that these estimates be generated directly by the model in closed form; rather, one can apply other estimation methods, such as bootstrapping, to obtain uncertainty estimates. This is demonstrated in using Random Forest models as successful surrogates (Hutter et al., 2011).

1.6 Variational Inference

Variational Inference (or Variational Bayes) is, in addition to Markov-chain Monte Carlo, another asset in our arsenal when it comes to working with an intractable posterior distribution. However, while MCMC is a sampling method, and therefore yields random results (especially under computational restrictions); Variational Inference is a deterministic approach to posterior inference. Tracing its roots back to variational calculus (Bishop, 2006), the machinery on which Variational Inference rests involves minimizing the output of a functional – a mapping from functions to values. In variational inference, we attempt to approximate our distribution of interest using some other tractable distribution. This tractable distribution is the input to our functional (since probability density functions are functions), and considering how distributions, given their family, are perfectly characterized by their parameters, we aim to optimize the value of our functional by choosing the optimum parameter values for a given family of distributions. The final ingredient here is now functional, which is often chosen as a measure of divergence between the input distribution and the distribution that is the target of approximation; typically, Kullback-Leibler (KL) divergence Kullback and Leibler (1951). This approach, therefore, transforms posterior inference into an optimization problem with

a deterministic solution, and the objective takes on a consistent form when working with KL divergence

$$\begin{aligned}\mathcal{L}(q) &= \ln p(\mathbf{X}) - \mathbb{KL}(q||p) = \mathbb{E}_q \left[\ln \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right] \leq \ln p(\mathbf{X}) \\ \implies obj &: \max_q \mathcal{L}(q),\end{aligned}$$

where $\ln p(\mathbf{X})$ is the log-likelihood of the observed data, q is the approximate distribution, $\mathbb{KL}(q||p)$ is the KL-divergence, and $p(\mathbf{X}, \mathbf{Z})$ is the joint probability density function of the observed data and the parameters of interest. This lower bound we aim to maximize is often called the evidence lower bound (ELBO).

Perhaps one of the most commonly utilized variational inference frameworks is the mean-field assumption (Opper and Saad, 2001). The mean field assumption restricts the space of probability density functions to the family of factorizable distributions such that the following relationship holds

$$q(\mathbf{Z}) = \prod_i q_i(\mathbf{z}_i),$$

where $q(\cdot)$ is our approximating distribution, $\mathbf{Z}$ is a vector of all parameters we are interested in, and $\mathbf{z}_i$ represents a grouping of these parameters into sub-vectors or even scalars. This particular assumption allows for optimizing a single parameter at a time while marginalizing the rest. However, given that we have no access to the distributions of the remaining parameters, this is implemented in an iterative algorithm that cycles through each parameter and updates its estimates conditional on previous estimates of the remaining parameters such that

$$\ln q_j^*(\mathbf{z}_i) = \mathbb{E}_{i \neq j} \left[\ln p(\mathbf{X}, \mathbf{Z}) \right] + const.,$$

Given the convexity of the Evidence Lower Bound (Boyd and Vandenberghe, 2004), this approach is guaranteed to converge. However, one notes that it is guaranteed to converge to the best attainable approximation, which can sometimes be an insufficient substitute for

the true distribution.

1.7 Transformers

Transformers (Vaswani et al., 2017) are a neural network architecture that is particularly defined by the presence of an attention mechanism. Originally developed for natural language processing, these models can learn interdependencies within an observation by leveraging a sub-model that produces an attention map. An attention map is a learned weighting of the strength of the connection between elements of an observation, such as words in a sentence, subpatches in an image, or subdomains in a function transformers can accommodate both temporal and spatial connections as they can be fed both relative and absolute positional input, offering a simple way to encode connectivity-related domain knowledge. More recently, some work has emerged that leverages the power of transformers in analyzing functional magnetic resonance imaging data (fMRI). Malkiel et al. (2022) applied transformers to 3D fMRI scans to predict patient characteristics; the first application of the transformer architecture to functional fMRI time series data was implemented in Li et al. (2022), where domain knowledge of functional brain connectivity was fed to the transformer as a positional input. Also working with functional fMRI data, Bedel et al. (2022) introduce the fused window transformer for fMRI data, which implements a windowed attention mechanism over the function. More recently, Zhang and Yan (2023) proposed the crossformer, a transformer architecture designed for multivariate time series data which handles dependencies across both time and dimension by introducing a two-stage-attention mechanism.

1.8 Outline of the Dissertation

This dissertation is structured as follows, in Chapter 2 we explore variable selection and network estimation in settings involving non-negligible dependency structures among vari-

ables. In addition to the conditional dependencies between the covariates, there also exists a multi-level dependency structure involving groups of networks acting in unison. The work in this chapter examines inference in this two-level structure where higher-level variables contain lower-level variables nested within them; our main interest is to simultaneously select variables relevant to the response and identify dependency structures among both higher and lower-level variables. Given data from heterogeneous classes, we propose a multi-level nonparametric kernel machine approach, utilizing both Markov-chain Monte Carlo and Variational Bayes inference to jointly identify multi-level variables as well as recover the network. In addition to the variable selection and network recovery capabilities, our approach is capable of producing both mean and quantile estimations of the original response variable of interest. We demonstrate the advantages of our approach using both simulation studies and a genetic pathway-based analysis.

In Chapter 3, we explore dependency structures in functional data. Most methods used for functional data analysis either leverage the assumption of additive functional models with sparse penalty functions of coefficients of basis functions or proceed by decomposing functional variables into scalar spaces. In addition, these methods do not typically consider a multilevel dependency structure, being relegated to only dealing with temporal dependencies. Given our motivating dataset, which is the identification of regions in the human brain contributing to the development of Autism Spectrum Disorder within an individual, where our data is a mixture of both scalar and high-dimensional functional covariates, such as demographic data and resting-state functional magnetic resonance imaging (fMRI) data of the brain locations; our main interest is to identify subsets on the domain of the functional covariates, recover the graph structure among locations, and test significant association of subsets and graph selection with response. No existing functional variable selection methods are used to conduct these multi-inference tasks. In this chapter, we develop a Bayesian multi-tasking inference method under a joint functional kernel graphical model framework. Our

method unifies Bayesian optimization, an Ising graphical model, and a score-type test using a Bayesian approach so that we can perform (1) the selection of subsets within the domain of the functional covariates, (2) the identification of relevant brain locations to correct for the confounding effect of correlation between the signals, and (3) test the significance of selection and identification. The advantage of our multi-tasking inference method is demonstrated using simulations and applied to our motivating example.

Chapter 2

Bayesian Multi-level Network Recovery Selection

2.1 Introduction

Highly correlated high-dimensional data (HCHD) are frequently observed in many areas, including predicting protein-protein interaction, detecting significant biosensing signals, and identifying important disease-related pathways. The main challenge is that the number of variables can be much larger than the sample size, and these variables often have multi-level correlated structures. Due to higher-level variables possibly working together to accomplish certain tasks, simultaneously exploring conditional dependency structures among higher-level and lower-level variables is our main interest. We define a “set” as a group of HCHD variables that possess a certain practical meaning or control a certain process and an “element” as one of the HCHD variables within a certain set.

The goal of this study is to select set variables from P potential candidates by exploring the set configuration space of $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_P)^T$, as well as selecting element variables from the L_p potential candidates within a selected p th set by exploring the element configuration space of $\boldsymbol{\xi}_p = (\xi_{1p}, \dots, \xi_{L_p p})$, where $p = 1, \dots, P$ and $l = 1 \dots, L_p$.

Numerous variable selection methods were developed (Fang et al., 2016, Li and Zhang, 2010, OHara and Sillanpää, 2009, Polson and Scott, 2012, Reich et al., 2009, Sanyal et al., 2017, Scheipl, 2011, Shin et al., 2018, Tibshirani, 1996, Zhang et al., 2018); more recently, Shin et al. (2018) proposed estimating a mean function under a parametric model using a nonlocal prior distribution. Chen et al. (2016) developed Bayesian sparse group selection. Tang et al. (2017) proposed the group spike and slab lasso method. Bai et al. (2020) also developed sparse Bayesian additive nonparametric regression, introducing spike-slab group lassos priors and B-splines basis functions. Hence, these existing variable selection methods provided single-level variable selection tools under either a parametric model or a nonparametric model but not for multi-level selection. To the best of our knowledge, there are no variable selection methods that develop a multi-level selection approach under a multi-kernel nonparametric model.

Conducting multi-level variable selection and modelling between the response and multi-level variables are statistically challenging because (1) elements within a set are often highly correlated and high dimensional; (2) sets are also highly correlated; (3) parametric models may not be appropriate because of unknown high-order interactions among multi-level variables; (4) there may exist a need to identify multi-level variables associated with a quantile of the response variable.

In this chapter, we build a multi-level network model under a nonparametric model framework to select set variables and element variables simultaneously within a set associated with the mean and quantile of the response variable and to model unknown relationships among the response, the set, and the element variables. Our multi-level network recovery method can simultaneously select sets and elements that highly correlate with each other.

This chapter is organized as follows: In Section 2.2, we propose the multi-level network model. In Section 2.3, we develop the network recovery method. In Sections 2.3.2-2.3.2, we provide the Markov-chain Monte Carlo (MCMC) based inference and algorithm and the variational Bayes (VB) inference and algorithm for multi-level variable selection. In Section 2.4, we provide the asymptotic properties. In Section 2.5, we conduct simulation studies to evaluate the performances of our proposed method. In Section 2.6, we demonstrate the advantage of our method using genetic pathway data. Lastly, we present concluding remarks in Section 2.7.

2.2 Multi-level Network Model

Let $\mathbf{y}$ be an $N \times 1$ vector of the response variables, and $\mathbf{X}_p$ be an $N \times L_p$ covariate matrix for the p^{th} set variables, which contains L_p element variables. We build a multi-level network model under a nonparametric model framework as follows,

$$\underline{\mathbf{y}} = \sum_{p=1}^P \gamma_p f_p(\mathbf{X}_p; \underline{\boldsymbol{\xi}}_p) + \underline{\boldsymbol{\epsilon}}, \quad (2.1)$$

where $f_p(\cdot; \cdot)$ is an unknown high dimensional function in a reproducing kernel Hilbert space ($\mathcal{H}$) and $\underline{\boldsymbol{\epsilon}}$ is a random noise vector.

Because $f_p \in \mathcal{H}$ and reproducing property holds in Hilbert space, our multi-level network model can be expressed as:

$$\underline{\mathbf{y}} = \gamma_1 K_1(\underline{\boldsymbol{\xi}}_1, \mathbf{X}_1) \underline{\boldsymbol{\alpha}}_1 + \gamma_2 K_2(\underline{\boldsymbol{\xi}}_2, \mathbf{X}_2) \underline{\boldsymbol{\alpha}}_2 + \cdots + \gamma_P K_P(\underline{\boldsymbol{\xi}}_P, \mathbf{X}_P) \underline{\boldsymbol{\alpha}}_P + \underline{\boldsymbol{\epsilon}}, \quad (2.2)$$

where $\underline{\boldsymbol{\alpha}}_p$ is the unknown kernel weight, p is the set number, K_p is a $n \times n$ kernel matrix such that the $(i, j)^{th}$ element of $K_p(\underline{\boldsymbol{\xi}}_p, \mathbf{X}_p)$ is

$$k_p(\mathbf{x}_i, \mathbf{x}_j) = \exp \left[- \frac{\sum_{l=1}^{L_p} \xi_l (x_{li} - x_{lj})^2}{d_l} \right].$$

We note that this setup is similar to one found in Bobb et al. (2014). To select for relevant element lower level variables, we introduce the binary parameter ξ_{lp} , $\xi_{lp} = \{0, 1\}$, $l = 1, \dots, L_p$, indicating the presence of variable lp which follows a Boltzmann distribution of the Ising graphical model (ISGM). Define $\tilde{\xi}_{lp} = (-1)^{\xi_{lp}+1}$ and $\tilde{\xi}_{lp} \in \{-1, 1\}$. Given a graph $G = (V, E)$, where V represents vertex (or node) and E represents discrete edges, we consider the Boltzman distribution, $P_{\underline{\boldsymbol{\xi}}_p}(\xi_{1p}, \dots, \xi_{L_p p}) \propto \exp\{\sum_{s_p \in V} q_{s_p} \tilde{\xi}_{s_p} + \sum_{(s_p, l_p) \in E} J_{s_p l_p} \tilde{\xi}_{s_p} \tilde{\xi}_{l_p}\}$, where $\underline{\boldsymbol{\xi}}_p \in R^{|V|+|E|}$. Here $\mathbf{q}$ is called an “external field” and $\mathbf{J}$ is the interaction matrix. In our study, lp^{th} variables refer to l^{th} node within the p^{th} pathway and interaction among lower-level variables refers to graph/network. Hence $|V| = \sum_{p=1}^P L_p$ and $|E| = \sum_{p=1}^P [L_p(L_p - 1)/2]$.

In our study, we consider two distributions of $\underline{\boldsymbol{\epsilon}}$: (1) the distribution of $\underline{\boldsymbol{\epsilon}}$ is assumed to follow a multivariate Gaussian distribution, $MVN(\underline{0}, \sigma^2 \mathbf{I})$, where σ^2 is the variance of $\underline{\boldsymbol{\epsilon}}$, to estimate the conditional expectation of the response variable; (2) $\underline{\boldsymbol{\epsilon}}$ is assumed to follow a three-parameter asymmetric LaPlace distribution, $ALD(0, \sigma, q)$, where q is the quantile.

The latter can be used to estimate the conditional quantiles of the response variable.

In order to incorporate prior knowledge of multi-level correlated structure such that p sets interact with each other at the higher level and the L_p elements also interact among themselves in the lower level, we consider the Boltzman distribution of the Ising graphical model for both γ 's and ξ 's.

The intuition behind using the Ising graphical model of binary parameters (Fang and Kim, 2016) lies within the assumption that both sets and elements, when acting purposely, tend to act in unison. Even if the observed change in a single object is relatively small, it is necessarily associated with similar effects in related objects. Our context captures this effect through its amplification whenever multiple objects change simultaneously. The Boltzman distribution of the Ising graphical model for binary parameters is well suited for this purpose, as the distribution allows for the probability of a given object taking a specific value to be a function of its neighbours' own values as well as an external effect. This creates a network structure with pairing effects (Dai et al., 2013). This can be extended to any higher levels of the objects as well.

2.3 Multi-level Recovery Method

In this section, we develop the multi-level nonparametric kernel machine approach as a multi-level network recovery method (MLNR), utilizing both MCMC and VB to identify multi-level variables as well as build the networks jointly. Using our method, we estimate conditional mean and quantiles of the response in Section 2.3.1 and Section 2.3.1, respectively.

2.3.1 Kernel Weights Estimation

We estimate kernel weights ($\underline{\alpha}_p$) under both multivariate normal and asymmetric Laplace distributions.

Estimation of Conditional Mean

Under the normality assumption, the sampling of our kernel weights reduces to the typical form of ¹,

$$\begin{aligned}\underline{\alpha}_p|rest &\sim MVN\left(\underline{\mu}_\alpha^*, \mathbf{V}_\alpha^*\right), \\ \underline{\mu}_\alpha^* &= \mathbf{V}_\alpha^* \mathbf{V}_\alpha^{-1} \underline{\alpha}_{p0} + \frac{1}{\sigma^2} \mathbf{V}_\alpha^* \mathbf{K}_p^T \left(\underline{y} - w \sum_{q \neq p} \gamma_q K_q \underline{\alpha}_q \right), \\ \mathbf{V}_\alpha^* &= \sigma^2 \left(\sigma^2 \mathbf{V}_\alpha^{-1} + \mathbf{K}_p^T \mathbf{K}_p \right)^{-1},\end{aligned}\tag{2.3}$$

which is obtained under the assumption of a multivariate normal prior on our weights such that $\underline{\alpha}_p \sim \mathcal{MVN}(\underline{\alpha}_{p0}, \mathbf{V}_\alpha)$ and $\sigma^2 \sim \text{Inv.Gamma}(\frac{a}{2}, \frac{b}{2})$.

We note that in our Gibbs-sampling implementation, $\underline{\alpha}_p$ are sampled independently for each p^{th} set, as the full conditional changes with respect to p . As for σ^2 , the relevant full conditional distribution is

$$\sigma^2|rest = \text{Inv.Gamma}\left(\frac{N+a}{2}, \frac{1}{2} \left(\underline{\mathbf{y}} - \sum_{p=1}^P \gamma_p \mathbf{K}_p \underline{\alpha}_p \right)^2 + \frac{b}{2} \right).\tag{2.4}$$

Prior hyperparameters $a, b, \underline{\alpha}_{p0}, \mathbf{V}_\alpha$ are assumed to be known.

Estimation of Conditional Quantiles

Our estimation of the kernel weights under the three-parameter ALD for the estimation of response quantiles extends the work of Kozumi and Kobayashi (2011) to the case of robust quantile regression as follows

$$\underline{\epsilon} = \theta \underline{\mathbf{z}} + \tau \sqrt{\underline{\mathbf{z}}} \underline{\mathbf{u}},$$

¹Note: We use K_p as shorthand notation for $K_p(\underline{\xi}_p, \mathbf{X}_p)$; w is a shrinkage tuning parameter

where $\mathbf{u} \sim MVN(\mathbf{0}, \mathbf{I})$, $z_i \stackrel{iid}{\sim} \exp(1)$, and θ and τ^2 are

$$\theta = \frac{1-2q}{q(1-q)}, \quad \tau^2 = \frac{2}{q(1-q)},$$

and q is the quantile of interest.

We update our estimate for the kernel weights as

$$\underline{\alpha}_p | rest \sim MVN(\underline{\mu}_\alpha^*, \hat{\mathbf{B}}_p), \quad (2.5)$$

$$\underline{\mu}_\alpha^* = \hat{\mathbf{B}}_p \left[\mathbf{K}_p \mathbf{Z}^{-1} (\underline{\mathbf{y}} - \omega \sum_{q \neq p} \gamma_q \mathbf{K}_q \underline{\alpha}_q - \theta \underline{\mathbf{z}}) \frac{1}{\tau^2} + \mathbf{B}_{p0}^{-1} \underline{\alpha}_{p0} \right],$$

$$\hat{\mathbf{B}}_p^{-1} = \frac{1}{\tau^2} \mathbf{K}_p^T \mathbf{Z}^{-1} \mathbf{K}_p + \mathbf{B}_{p0},$$

where we define $\mathbf{Z}$ as $\mathbf{I} \odot \underline{\mathbf{z}} \cdot \underline{\mathbf{1}}^T$, $\underline{\alpha}_{p0}$ is our prior mean on the kernel weights, and $\mathbf{B}_{p0}$ represents the covariance matrix on the kernel weights prior. Finally, the individual elements of z_i are each sampled from a generalized inverse Gaussian such that:

$$z_{i|\underline{y}, \underline{\alpha}_1, \dots, \underline{\alpha}_P} \sim \mathcal{GG}\left(\frac{1}{2}, \hat{\delta}_i^2, \hat{\kappa}_i^2\right),$$

where $\hat{\kappa}_i^2 = 2/\sigma + \theta^2/\tau^2\sigma$ and $\hat{\delta}_i^2 = \left(\underline{\mathbf{y}} - \sum_p^P \gamma_p \mathbf{K}_p \underline{\alpha}_p\right)/\tau^2\sigma$.

Under the asymmetric LaPlace kernel weight estimation and the prior $\sigma \sim Inv.Gamma(\frac{a}{2}, \frac{b}{2})$, the full conditional distribution of σ is obtained as

$$\sigma | rest = Inv.Gamma\left(\frac{3N+a}{2}, \frac{b}{2} + \sum_{i=1}^N z_i + \frac{1}{2\tau^2} \left(\underline{\mathbf{y}} - \sum_{p=1}^P \mathbf{K}_p \underline{\alpha} - \theta \underline{\mathbf{z}}\right)^T [diag(\underline{\mathbf{z}})]^{-1} \left(\underline{\mathbf{y}} - \sum_{p=1}^P \mathbf{K}_p \underline{\alpha} - \theta \underline{\mathbf{z}}\right)\right). \quad (2.6)$$

Once more, the prior hyperparameters $a, b, \underline{\alpha}_{p0}, \mathbf{B}_{p0}$ are assumed to be known.

2.3.2 The Selection Parameters

As shown in Model (2.1), the approach proposed includes two selection variables: γ and ξ , regardless of the distribution of ϵ . Given the Bayesian nature of our approach, we need to identify our prior assumptions for these parameters. Starting with ξ , which is the selection variable for the lowest level of our network, we assume a prior of the form

$$P(\underline{\xi}_p) \propto \left[\gamma_p \exp\left\{ \sum_{l < j} \tilde{\xi}_{lp} \omega_{ljp} \tilde{\xi}_{jp} + (1 - \gamma_p) \delta_{0_p} \right\}, \quad \xi_{lp} \in \{0, 1\}, \right]$$

where $\tilde{\xi} = (-1)^{\xi+1}$ and $\tilde{\xi}_{lp} \in \{-1, 1\}$. The prior on our latent variable, $\tilde{\xi}_{lp}$, is a spike-slab prior between an Ising distribution and a Dirac delta distribution.

Here, ω_{ljp} captures a researcher's prior belief of the strength of the co-movement between elements l and j within the set p . Should the researcher assume zero co-movement across the elements, this prior reduces to the form of a discrete uniform distribution, while assuming a positive constant w_{kjp} for all pairs tilts the probability towards outcomes with many matching values of ξ .

The choice for a prior on γ is simple assuming the independence of the sets, where a simple Bernoulli prior suffices such that $P(\gamma_p) \sim \rho_p^{\gamma_p} (1 - \rho_p)^{1-\gamma_p}$ and $\gamma_p \in \{0, 1\}$.

One could also extend a similar prior γ if sufficient interaction is assumed to exist between the sets within the network such that $P(\underline{\gamma}) \propto \exp \left[\sum_{p < q} \tilde{\gamma}_p \omega_{pq} \tilde{\gamma}_q \right]$ where $\gamma_p \in \{0, 1\}$, ω_{pq} captures a researcher's prior belief of the strength of the co-movement between sets p and q , and $\tilde{\gamma} = (-1)^{\gamma+1}$. Given that the sets are the top level element, a spike-slab prior is unnecessary when implementing the Ising prior over sets. The selection of sets then follows the work of Kuo and Mallick (1998) extended to the non-parametric setting such that

$$P(\gamma_p = 1 | \gamma_{-p}, \underline{\alpha}, \sigma_y^2, \underline{\mathbf{y}}) = \frac{c_p}{c_p + d_p},$$

where $c_p = P(\gamma_p = 1) \cdot \mathcal{L}(\underline{\mathbf{y}} | \underline{\mathbf{X}}, \phi, \gamma_{-p}, \gamma_p = 1)$, $\mathcal{L}$ denotes the likelihood function, ϕ is

a catch-all notation for the parameters in the distribution of $\underline{\mathbf{y}}$, and $d_p = P(\gamma_p = 0) \cdot \mathcal{L}(\underline{\mathbf{y}}|\underline{\mathbf{X}}, \phi, \gamma_{-p}, \gamma_p = 0)$.

The process and derivations for the element variable selection parameters differ based on the desired approach. In the following two subsections 2.3.2-2.3.2, we describe a McMC-based approach and a VB-based approach for estimating the element selection parameters.

Markov chain Monte Carlo Estimation

Our McMC-based approach builds on the work of Fang and Kim (2016), which developed a graphical variable selection via an Ising full conditional distribution under an additive linear model. In our study, we develop a graphical variable selection under the multi-level recovery model (2.2).

Our approach begins from the previous expressions for $\underline{\alpha}_p$ and σ^2 , where we scale the response variable and covariates such that $\tilde{y}_i = (y_i - \bar{y})/\hat{\sigma}$ and $\tilde{x}_{ik} = (x_{ik} - \bar{x}_{.k})/\hat{\sigma}_{x_k}$. By proposing a penalized Ising prior to our selection variables, our full conditional distributions for $\tilde{\xi}$ s take the following form

$$p(\tilde{\xi}_{lp}|\underline{\tilde{\mathbf{y}}}, \underline{\tilde{\xi}}_{-lp}, \underline{\alpha}, \sigma^2, \gamma_p = 1) \propto \exp \left[- \underbrace{\left\{ - \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} \tilde{\xi}_{lp} \tilde{\xi}_{jp} - \mathcal{L}(\tilde{\xi}_{lp}|\underline{\tilde{\xi}}_{-lp}, \underline{\tilde{\mathbf{y}}}, \underline{\alpha}, \sigma^2) \right\}}_{\text{energy}} \right], \quad (2.7)$$

$$p(\tilde{\xi}_{lp}|\underline{\tilde{\mathbf{y}}}, \underline{\tilde{\xi}}_{-lp}, \underline{\alpha}, \sigma^2, \gamma_p = 0) \propto \text{constant},$$

where ω_{ljp} represents a researcher's prior belief on the connection strength between elements l and j in pathway p . $\mathcal{L}(\cdot)$ is the log-likelihood function for the selection variable of interest, representing the contribution of the l^{th} covariate to the log-likelihood of our response. We note the dependence of the likelihood term on remaining selection variables $\underline{\tilde{\xi}}_{-lp}$. Given our assumption of coordinated action among our covariates, the likelihood term becomes unstable as the inclusion of one relevant covariate of the same set in our model nullifies the information

contribution that can be extracted from the remaining connected covariates. This is due to the assumption of high multicollinearity between covariates contributing to the response and near-zero contribution for irrelevant covariates. Bobb et al. (2014) effectively utilizes a similar assumption by drawing from a multinomial distribution a single covariate to enter the model. Likewise, we evaluate the likelihood under the condition that $\tilde{\underline{\xi}}_{-lp} = \underline{\mathbf{0}}$.

Returning to our expression of interest, the first term of equation (2.7) represents the neighbour/network effect and the second term, which is the equivalent of the external field in a typical Ising distribution application, represents the strength of the evidence for a relationship between each element and the response variable. Given the flexible nature of the Kernel regression model, we augment the full conditional with a log-likelihood penalty term within our Metropolis algorithm, such that our acceptance ratio is

$$\Delta = (\mathcal{U}_{\xi_{proposed}} - \xi_{proposed}\lambda_{\xi}f_{\lambda}(\mathbf{K}_{\mathbf{p}})) - (\mathcal{U}_{\xi_{current}} - \xi_{current}\lambda_{\xi}f_{\lambda}(\mathbf{K}_{\mathbf{p}})).$$

Note the penalty only applies when $\xi_{lp} = 1$ as otherwise, the model is simply a constant and thus carries the lowest possible degree of flexibility. We apply multiple choices of the penalty function f_{λ} as a function of the lengthscale parameter of our kernel, which given its automatic relevance determination (ARD) properties (Neal, 2012) in a Gaussian kernel offers insight regarding the flexibility of our single covariate model under examination, as larger lengthscales, conditional on knowledge of the noise parameter, indicate that the function is more loosely correlated along that covariate. Similar lengthscale penalties have been implemented in Yi et al. (2011). Hence, we estimate all the parameters using MCMC. We present the full MCMC (F-MCMC) algorithm 5 for recovering a multi-level network model:

Algorithm 1 F-MCMC algorithm for multi-level network recovery

- 1: Initialize model parameters $\sigma^{2(0)}, \underline{\gamma}^{(0)}, \underline{\xi}^{(0)}$
 - 2: Set priors using hyper-parameters $\mathbf{W}, \sigma_{\alpha}^2, a, b, \rho_{ald}$
 - 3: **for** each iteration **do**:
 - 4: Sample $\underline{\alpha}^{(t+1)}, \sigma^{2(t+1)}$ from their full conditionals
 - 5: **for** each p in P sets **do**:
 - 6: Update the posterior probability on each γ_p
 - 7: Sample γ_p using the updated posterior
 - 8: **end for**
 - 9: **for** each p in P sets **do**:
 - 10: Sample $\underline{\xi}_p$ in MH-sampler using $\sigma^{2(t+1)}$
 - 11: Retrieve median values for $\underline{\xi}_p$ after allowing for sufficient burn-in
 - 12: Pass updated selection parameters to covariates in a kernel of choice
 - 13: **end for**
 - 14: **end for**
 - 15: Fix γ_p to their estimated values
 - 16: **for** each iteration **do**:
 - 17: Sample $\underline{\alpha}^{(t+1)}, \sigma^{2(t+1)}$ from their full conditionals
 - 18: **for** each p in P sets **do**:
 - 19: Sample $\underline{\xi}_p$ in MH-sampler using $\sigma^{2(t+1)}$
 - 20: Retrieve mode values for $\underline{\xi}_p$ after allowing for sufficient burn-in
 - 21: Pass updated selection parameters to covariates in a kernel of choice
 - 22: **end for**
 - 23: **end for**
-

Variational Bayes Estimation

To reduce the computational cost of our algorithm, we resort instead to variational methods. In particular, rather than sampling from the posterior of the variable selection parameters for the lower levels, which we expect to have an intractable distribution, hence our previous use of Metropolis-Hastings sampling, we fit a variational distribution to approximate the posterior on our element selection variables. Therefore, we will approximate the posterior distribution of the latent variable conditional on the observed response variable; this translates to the distribution of the second level selection variable, $\tilde{\xi}_{lp}$, being approximated by some distribution $Q(\tilde{\xi}_{lp}|\underline{\mathbf{y}}, \gamma_p)$. In the next section, we show the reasoning behind our choice of $Q(\tilde{\xi}_{lp}|\underline{\mathbf{y}}, \gamma_p)$, which is based on Murphy (2013).

As is commonplace in mean-field variational inference, we assume that our proposed posterior is fully factorizable such that

$$Q(\underline{\tilde{\xi}}_p|\underline{\mathbf{y}}, \gamma_p) = \prod_{l \in \mathcal{P}_p} Q_l(\tilde{\xi}_{lp}|\underline{\mathbf{y}}, \gamma_p). \quad (2.8)$$

Our objective can now be stated as follows

$$\min_Q KL(Q_{\underline{\tilde{\xi}}_p|\underline{\mathbf{y}}, \gamma_p} || P_{\underline{\tilde{\xi}}_p|\underline{\mathbf{y}}, \gamma_p}),$$

where KL refers to the Kullback-Leibler divergence.

Next we define $P(\underline{\mathbf{y}}, \underline{\tilde{\xi}}_p|\underline{\gamma}, \underline{\alpha}, \sigma^2) = P(\underline{\tilde{\xi}}_p|\underline{\gamma}, \underline{\alpha}, \sigma^2) \cdot P(\underline{\mathbf{y}}|\underline{\tilde{\xi}}_p, \underline{\gamma}, \underline{\alpha}, \sigma^2)$ as

$$= \prod_{l \in \mathcal{P}_p} \left[E[\gamma_p] \exp\left\{\sum_{j < l} \tilde{\xi}_{lp} \omega_{ljp} \tilde{\xi}_{jp} + (1 - E[\gamma_p]) \delta_{0_p}\right\} \right] \cdot P(\underline{\mathbf{y}}|\underline{\tilde{\xi}}_p, \underline{\gamma}, \underline{\alpha}, \sigma^2), \quad (2.9)$$

where we set the prior on our latent variable, $\tilde{\xi}_{lp}$, as a spike-slab prior between an Ising distribution and a Dirac delta distribution. For the weight term of the Ising distribution, ω_{ljp} , we define $\omega_{ljp} = \text{corr}(x_{lp}, x_{jp})$.

The expectation of $\tilde{\xi}_{lp}$ for the VB approach can be derived as follows

$$\begin{aligned}
E_{\tilde{\xi}_{lp}}^{(t+1)} [\tilde{\xi}_{lp}] &\approx q(\xi_{lp} = 1 | \underline{\mathbf{y}}, \underline{\boldsymbol{\gamma}}, \underline{\boldsymbol{\alpha}}, \sigma^2) = \\
E[\gamma_p] \tanh &\left(\sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} E_{-\tilde{\xi}_{lp}}^{(t)} [\tilde{\xi}_{jp}] + \frac{1}{2} \log \left(\frac{P(\underline{\mathbf{y}} | \tilde{\xi}_{lp} = +1, E_{-\tilde{\xi}_{lp}} [\tilde{\boldsymbol{\xi}}_{-lp}], \underline{\boldsymbol{\gamma}}, \underline{\boldsymbol{\alpha}}, \sigma^2)}{P(\underline{\mathbf{y}} | \tilde{\xi}_{lp} = -1, E_{-\tilde{\xi}_{lp}} [\tilde{\boldsymbol{\xi}}_{-lp}], \underline{\boldsymbol{\gamma}}, \underline{\boldsymbol{\alpha}}, \sigma^2)} \right) \right) \\
&+ (1 - E[\gamma_p])(-1).
\end{aligned} \tag{2.10}$$

The detailed derivation of the above results for the VB approach is described in Appendix A. Hence, we estimate parameters using mixing MCMC with VI: ξ s are estimated via VB, but other parameters are using MCMC. We refer to this algorithm as M-MCVI. We present the M-MCVI algorithm 4 for recovering the multi-level network model:

Algorithm 2 M-MCVI-based algorithm for multi-level network recovery

```

1: Initialize model parameters  $\sigma^{2(0)}, \underline{\gamma}^{(0)}, \tilde{\underline{\xi}}^{(0)}, M$ 
2: Set priors using hyper-parameters  $\mathbf{W}, \sigma_\alpha^2, a, b, \rho_{ald}$ 
3: for each iteration do:
4:   Sample  $\underline{\alpha}^{(t+1)}, \sigma^{2(t+1)}$  from their full conditionals
5:   for each p in P sets do:
6:     Update the posterior probability on each  $\gamma_p$ 
7:     Sample  $\gamma_p$  from the updated posterior
8:   end for
9:   if t mod M = 0 then:
10:    for each p in P sets do:
11:      Update variational distribution expectation estimates for  $\tilde{\underline{\xi}}_p$ 
12:      Pass updated selection parameters to covariates in a kernel of choice
13:    end for
14:  end if
15: end for
16: Fix  $\gamma_p$  to their estimated values
17: for each iteration do:
18:   Sample  $\underline{\alpha}^{(t+1)}, \sigma^{2(t+1)}$  from their full conditionals
19:   if t mod M = 0 then:
20:    for each p in P sets do:
21:      Update variational distribution expectation estimates for  $\tilde{\underline{\xi}}_p$ 
22:      Pass updated selection parameters to covariates in kernel of choice
23:    end for
24:  end if
25: end for

```

2.4 Theoretical Properties

We provide the asymptotic properties of our multi-level network recovery method regarding posterior contraction rate and sparsity recovery. The incoherence condition for sparsity recovery is also described. The incoherence condition was originally proposed by Fang et al. (2016) under the single-level kernel model when ξ is continuous. We can implement this condition under the multi-level kernel model when ξ is binary.

We consider n , P , and l_p are going to infinity under some assumptions. We first define some notations.

2.4.1 Notations

Define the true α_p vector as α_p^* . Let γ^* and $\hat{\gamma}$ represents the true γ and estimator of higher level, respectively. Suppose vector γ^* is sparse, i.e., $\gamma_j^* = 0$. Without loss of generality, denote $\gamma^* = (\gamma_1^*, \dots, \gamma_P^*)^T = (\gamma_1^{*T}, \gamma_0^{*T})$, where γ_1^* is the vector of the first p^* ones of γ_P s. Denote the one index set of γ as $\mathcal{A} = \{p \in \{1, \dots, P\} | \gamma_p^* = 1\}$ and γ as $\hat{\mathcal{A}} = \{p \in \{1, \dots, P\} | \hat{\gamma}_p = 1\}$. Note that $\mathcal{A}$ has relatively small cardinality $p^* = |\mathcal{A}|$, the number of true one γ_P 's.

Let ξ_p^* and $\hat{\xi}_p$ represents the true ξ_p and estimator of the lower level, respectively. Suppose vector ξ_p^* is sparse. Without loss of generality, denote $\xi_p^* = (\xi_{1_p}^*, \dots, \xi_{l_p}^*)^T = (\xi_{1_p}^{*T}, \xi_{0_p}^{*T})$, where $\xi_{1_p}^*$ is the vector of the first ω_p^* ones of ξ_p . Denote the one index set of ξ_p as $\mathcal{B} = \{l \in \{1_p, \dots, l_p\} | \xi_l^* = 1\}$ and ξ as $\hat{\mathcal{B}} = \{l \in \{1_p, \dots, l_p\} | \hat{\xi}_l = 1\}$. Note that $\mathcal{B}$ has also relatively small cardinality $\omega_p^* = |\mathcal{B}|$, the number of true one ξ_l 's.

2.4.2 Assumptions

We have the following assumptions (A1)-(A6) for the multi-level model with multi-kernel functions, while provided similar assumptions Bai et al. (2020), Wei et al. (2020) but were used for a single-level model with B-splines basis functions:

(A1) Assume that $P \gg n$, $l_p \gg n$, $\log P = o(n)$, $\log l_p = o(n)$, $p^* \approx n^{1/(2k_1+1)}$, and $\omega_p^* \approx n^{1/(2k_2+1)}$;

(A2) The number of true nonzero functions satisfies

$$p^* = o(\max\{n/\log(P), n/\log(l_p), n^{2k_1/(2k_1+1)}, n^{2k_2/2k_2+1}\});$$

(A3) There exists a constant $k > 0$ so that for all n , $\lambda_{\max}(\tilde{Z}_H^T \tilde{Z}_H) \leq k n$;

(A4) Let $\tilde{Z}_{\mathcal{A}}$ denote the submatrix of $\tilde{Z}_H$ that contains the submatrices indexed by $\mathcal{A}$. Let $\tilde{Z}_{\mathcal{B}}$ denote the submatrix of $\tilde{Z}_L$ that contains the submatrices indexed by $\mathcal{B}$. There exists constants $c_1 > 0$ and $c_2 > 0$ and an integer $\tilde{P}$ satisfying $p^* = o(\tilde{P})$ and $\tilde{P} = o(p^* \log(n))$, so that $\lambda_{\min}(\tilde{Z}_{\mathcal{A}}^T \tilde{Z}_{\mathcal{A}}) \geq nc_1$ and $\tilde{l}_p$ satisfying $\omega_p^* = o(\tilde{l}_p)$ and $\lambda_{\min}(\tilde{Z}_{\mathcal{B}}^T \tilde{Z}_{\mathcal{B}}) \geq nc_2$ for any model of size $|\mathcal{A}| \leq \tilde{P}$ and $|\mathcal{B}| \leq \tilde{l}_p$;

(A5) $|\mathcal{A}| = O(\log P)$ and $|\mathcal{B}| = O(\log l_p)$

(A6) The bias Δ satisfy $\|\Delta\|_2 \preceq \min\{\sqrt{p^* n} P^{-\kappa}, \sqrt{\omega_p^* n l_p^{-\kappa}}\}$, where Δ denotes the bias induced kernel machine approximation in the model

$$f(X_1, \dots, X_p) = \sum_{p=1}^P \gamma_p K(X_p, \xi_p) \alpha + \sigma \Delta;$$

and $f \in C^\kappa[0, 1]$, where true function f are all at least κ -times continuously differentiable over $[0, 1]$.

2.4.3 Asymptotic Results under Multi-level Network Model

Let $\tilde{\mathbb{P}}_0$ denote the probability measure underlying the truth (2.1) and $\Pi(\cdot|Y)$ denote the posterior distribution under model (2.2) with prior for $(\gamma, \xi, \alpha, \sigma^2)$. Further, let $\tilde{f}(X_i) = \sum_{p=1}^P \gamma_p f_p(X_{ip}; \xi_p)$ and $\tilde{f}_0(X_i) = \sum_{p=1}^P \gamma_p^* f_{0p}(X_{ip}; \xi_p^*)$, and define the empirical norm $\|\cdot\|_n$ as

$$\|\tilde{f} - \tilde{f}_0\|_n^2 = \frac{1}{n} \sum_{i=1}^n [\tilde{f}(X_i) - \tilde{f}_0(X_i)]^2.$$

Let $\mathcal{H}$ denote the infinite-dimensional set of all possible functions $\tilde{f} = \sum_{p=1}^P \tilde{f}_p$, where each $\tilde{f}_p$ belongs to a reproducing kernel Hilbert space (RKHS). Using the reproducing property, the p th component function f_p can be written in the form $f_p(\cdot) = \sum_{i=1}^n R_p \langle \cdot, X_{ip} \rangle \alpha_{ip}$, where the function $R_p \langle \cdot, x \rangle$ is a member of $\mathcal{H}$ and $\mathbf{R}_p \in \mathbf{R}^{n \times n}$ is a Gram matrix defined by the kernel-in particular with entries $(\mathbf{R}_p)_{l,l'} = R_p(x_{lp}, x_{l'p})$.

It was shown that the minimax estimation rate for $\tilde{f}_0 = \sum_{p=1}^P \tilde{f}_{0j}$ under squared ℓ_2 error loss is $\epsilon_n^2 \approx p^* \log P / n + p^* n^{-2\kappa/(2\kappa+1)}$, for some $\kappa \in \mathcal{N}$. Theorem 1 established that our multi-level network model achieves this minimax posterior contraction rate.

Theorem 1 (Posterior contraction rates)

Let $\epsilon_n^2 = p^* \log P / n + n^{-2k/(2k+1)}$. Suppose that Assumptions (A1)-(A6) hold. Under multi-level model (2.2), suppose that we endow $(\alpha_p, \sigma^2, \gamma, \xi)$ with the priors described in Sections 2.3.1-2.3.1. With the hyperparameter in the $Bin(1, \tilde{p}_p)$ prior on $pr(\tau_p = 1)$ and in the $Bin(1, \tilde{p}_{lp})$, $\tilde{p}_p \sim Beta(1, p_u^c)$ prior on $pr(\xi_{lp} = 1)$, $\tilde{p}_{lp} \sim Beta(1, p_l^c)$, $c > 2$,

$$\begin{aligned} & \prod(f \in \mathcal{H} : \|\tilde{f} - \tilde{f}_0\|_n \geq \tilde{M}_1 \epsilon_n | \mathbf{Y}) \rightarrow 0 \text{ a.s.}, \tilde{\mathbb{P}}_0 \text{ as } n, P \rightarrow \infty, \text{ and } l_p \rightarrow \infty \\ & \prod(\sigma^2; |\sigma^2 - \sigma_0^2| \geq 4\sigma_0^2 \epsilon_n | Y) \rightarrow 0, \text{ as } \sigma \rightarrow \infty, \text{ a.s. } \tilde{\mathbb{P}}_0 \text{ as } n, P \rightarrow \infty, \text{ and } l_p \rightarrow \infty, \end{aligned}$$

for some $\tilde{M}_1 > 0$ and σ_0^2 is the variance of the random error term of the true model.

The proof of Theorem 1 can be derived by extending single-level models (Bai et al., 2020, Wei et al., 2020) to multi-level models. Wei et al. (2020) developed sparse Bayesian additive nonparametric regression using spike-slab prior and B-splines basis functions, and Bai et al. (2020) also developed sparse Bayesian additive nonparametric regression introducing spike-slab group lassos priors and B-splines basis. In Bai et al. (2020), f_p was approximated with the form $g_p(X_p)B_p$, where $g_p(X_p)$ is B splines basis functions of X_p and B_p are the vector of basis coefficient, while our study approximates f_p with $\gamma_p K(X_p, \xi_p) \alpha_p$. Hence, the proof can be shown by replacing $g_p(X_p)B_p$ with $\gamma_p K(X_p, \xi_p) \alpha$.

Let us define dimensionality as $|\gamma| = \sum_{p=1}^P I(\gamma_p = 1)$. Theorem 2 shows that under our

multi-level network model, the expected posterior probability that the dimension size is a constant multiple larger than the true model size p^* and ω_p^* asymptotically vanishes.

Theorem 2 (Dimensionality) Suppose that the same conditions as those in Theorem 1 hold. Then under our multi-level network model (2.2), for sufficiently large $\tilde{M}_2 > 0$,

$$\sup_{\gamma^*} \tilde{E}_{\gamma^*} \prod(\gamma : |\gamma| > \tilde{M}_2 p^* | Y) \rightarrow 0, \text{ as } n, P \rightarrow \infty$$

and

$$\sup_{\xi^*} \tilde{E}_{\xi^*} \prod(\xi : |\xi(\gamma)| > \tilde{M}_3 \omega_p^* | Y) \rightarrow 0, \text{ as } n, l_p \rightarrow \infty,$$

for some $\tilde{M}_1 > 0$ and $\tilde{M}_2 > 0$.

The proof of Theorem 2 can be derived by extending single-level models (Bai et al., 2020) to multi-level models. For a high level, Theorem 2 can be proved by replacing $g_p(X_p)B_p$ with $\gamma K_p(X_p)\alpha_p$ in a higher level, where $g_p(X_p)$ is B-spline basis functions of X_p and B_p are the vector of basis coefficient. For the lower level, when a high level is given p , Theorem 2 can be shown by replacing it with $K_{l_p}(X_{l_p})\xi_{l_p}$.

Theorem 3 (Equivalence) The Markov-chain-Monte-Carlo-based approach and the Variational Inference converge conditional on the univariate approximation

$$P(\tilde{\xi}_{l_p} = 1 | \underline{\mathbf{y}}, \underline{\boldsymbol{\nu}}, \gamma_p = 1) \propto \exp \left[\sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} \tilde{\xi}_{jp} + \ln P(\underline{\mathbf{y}} | \tilde{\xi}_{l_p} = +1, \tilde{\boldsymbol{\xi}}_{-l_p}, \underline{\boldsymbol{\nu}}) \right],$$

For the variational distribution, we have

$$Q_i(\tilde{\xi}_{l_p} = 1 | \underline{\mathbf{y}}, \underline{\boldsymbol{\nu}}, \gamma_p = 1) \propto \exp \left[\sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} E_{-\tilde{\xi}_{jp}} [\tilde{\xi}_{jp}] + \ln P(\underline{\mathbf{y}} | \tilde{\xi}_{l_p} = +1, E_{-\tilde{\xi}_{l_p}} [\tilde{\boldsymbol{\xi}}_{-l_p}], \underline{\boldsymbol{\nu}}) \right].$$

Where $\underline{\boldsymbol{\nu}} = (\underline{\boldsymbol{\gamma}}, \underline{\boldsymbol{\alpha}}, \sigma^2)$ for notational brevity. We also have the following three properties:

Property 1 The P -dimensional binary random variable $\boldsymbol{\gamma}$ of the multi-level network model (2.2) is a class of stochastic processes on a finite random undirected graph model $G_H =$

(V_H, E_H) , where $V_H = \{1, \dots, P\}$ is the set of nodes, corresponding to P sets, and $E_H \subset V_H \times V_H$ is the set of edges. $\gamma \in \Gamma_H = \{(\gamma_1, \dots, \gamma_P) : \gamma_p, p = 1, \dots, P\}$ is indexed by V_H with probability measure on Γ_H .

Property 2 If $K(X; \xi)$ is linear function, the l_p -dimensional binary random variable ξ_p of the multi-level network model (2.2) is also a class of stochastic processes on a finite random undirected graph model $G_L = (V_L, E_L)$, where $V_L = \{1, \dots, l_p\}$ is the set of nodes, corresponding to l_p elements, and $E_L \subset V_L \times V_L$ is the set of edges. $\xi_p \in \Gamma_L = \{(\xi_1, \dots, \xi_{l_p}) : \xi_l, l = 1, \dots, l_p\}$ is indexed by V_L with the probability measure on Γ_L .

Property 3 If all correlations $w_{ljp} = 0 \ \forall \ l, i, j$ in eq.(A.5) is set to zero, then the variable selection approach based on VB reverts to a simple likelihood ratio test of each of the individual sets.

2.5 Simulation

We conduct simulation studies to evaluate the performance of the proposed method (MLNR). In Section 2.5.1, we conduct a simulation study to compare MLNR using MCMC with MLNR using VB in Section 2.5.1. In Section 2.5.2, we compare our method with four existing methods : Lasso (Tibshirani, 1996), Group Lasso (GL) (Yuan and Lin, 2006), Sparse Group Lasso (SGL) (Simon et al., 2013), Spike-and-Slab Group Lasso (SSGL)(Bai et al., 2020).

We compare them in terms of four evaluation metrics: the true positive rate (TPR), the false positive rate (FPR), the positive predictive value (PPV), and the f1 score using both set and element variables, defined as follows,

$$\begin{aligned} \mathbf{TPR}_{\text{set}} &= \frac{\sum_{\mathcal{P}} \mathbf{I}(\hat{\gamma}_p = 1, \gamma_p = 1)}{\sum_{\mathcal{P}} \mathbf{I}(\gamma_p = 1)} \quad ; \quad \mathbf{TPR}_{\text{element}} = \frac{\sum_{\mathcal{P}} \sum_{\mathcal{L}_p} \mathbf{I}(\hat{\xi}_{lp} = 1, \xi_{lp} = 1)}{\sum_{\mathcal{P}} \sum_{\mathcal{L}_p} \mathbf{I}(\xi_{lp} = 1)} \\ \mathbf{FPR}_{\text{set}} &= \frac{\sum_{\mathcal{P}} \mathbf{I}(\hat{\gamma}_p = 1, \gamma_p = 0)}{\sum_{\mathcal{P}} \mathbf{I}(\gamma_p = 0)} \quad ; \quad \mathbf{FPR}_{\text{element}} = \frac{\sum_{\mathcal{P}} \sum_{\mathcal{L}_p} \mathbf{I}(\hat{\xi}_{lp} = 1, \xi_{lp} = 0)}{\sum_{\mathcal{P}} \sum_{\mathcal{L}_p} \mathbf{I}(\xi_{lp} = 0)} \end{aligned}$$

$$\begin{aligned} \mathbf{PPV}_{\text{set}} &= \frac{\sum_{\mathcal{P}} \mathbf{I}(\hat{\gamma}_p = 1, \gamma_p = 1)}{\sum_{\mathcal{P}} \mathbf{I}(\hat{\gamma}_p = 1)} \quad ; \quad \mathbf{PPV}_{\text{element}} = \frac{\sum_{\mathcal{P}} \sum_{\mathcal{L}_p} \mathbf{I}(\hat{\xi}_{lp} = 1, \xi_{lp} = 1)}{\sum_{\mathcal{P}} \sum_{\mathcal{L}_p} \mathbf{I}(\hat{\xi}_{lp} = 1)} \\ \mathbf{F1}_{\text{set}} &= \frac{2\mathbf{PPV}_{\text{set}} \times \mathbf{TPR}_{\text{set}}}{\mathbf{PPV}_{\text{set}} + \mathbf{TPR}_{\text{set}}} \quad ; \quad \mathbf{F1}_{\text{element}} = \frac{2\mathbf{PPV}_{\text{element}} \times \mathbf{TPR}_{\text{element}}}{\mathbf{PPV}_{\text{element}} + \mathbf{TPR}_{\text{element}}}. \end{aligned}$$

2.5.1 Comparison between MLNR based on F-MCMC and MNLR based on M-MCVI

To compare our F-McMC and M-MCVI approaches, we run some 100 simulations with N , the number of observations equal to 25, with 5 sets and 38 elements. Our response is generated as follows,

$$\underline{y} = 10 \cos(\underline{\mathbf{x}}_{(1,1)}) + 3\underline{\mathbf{x}}_{(1,2)}^2 + \underline{\mathbf{x}}_{(1,1)}\underline{\mathbf{x}}_{(1,2)}\underline{\mathbf{x}}_{(1,4)} - \frac{1}{4} \sin(\underline{\mathbf{x}}_{(2,3)})\underline{\mathbf{x}}_{(2,2)} + \frac{2}{5}\underline{\mathbf{x}}_{(2,2)}\underline{\mathbf{x}}_{(2,3)}\underline{\mathbf{x}}_{(2,4)} + \underline{\epsilon},$$

where the first index on the independent variables $(\underline{\mathbf{x}}_{(1,2)}, \dots, \underline{\mathbf{x}}_{(2,4)})$ represents the sets, and the second represents the elements. We focus on this function as a nonlinear function that highlights the strength of our method in capturing nonlinear effects. We are also mindful of maintaining a similar effect magnitude of different sets, as having a dominating set will unnecessarily lead to eliminating sets with weaker effects.

With respect to our observed variables, we draw the elements within each set simultaneously from a multivariate normal distribution because we assume that connected elements tend to display stronger, same-direction correlations, which we encode in the covariance matrix such that

$$\mathbf{X}_{\mathbf{p},.} \sim \mathcal{MVN}(\underline{\mu}_{\mathbf{p}}, \Sigma_{\mathbf{p}}),$$

where $\underline{\mu}_{\mathbf{p}} \sim \mathcal{MVN}(\mathbf{0}, \mathbf{I})$, and $\Sigma_{\mathbf{p}}$ is generated such that the diagonal terms are set to 1 and the off-diagonal terms are sampled as a product of uniform distributions, with the exception of Σ_1 and Σ_2 , where the off-diagonal terms are manually set to maintain high correlations between the connected elements. We generate $\underline{\epsilon}$ from two different scenarios:

the first being a standard normal distribution and the second being a symmetric standard LaPlace distribution.

Tables 2.1-2.2 show the set and element selection results. We observe that while both methods perform similarly on set selection, the McMC algorithm tends to experience difficulty in quickly discarding irrelevant elements regardless of the choice of likelihood or noise-generating distribution. We attribute this to convergence difficulties associated with a single-step update in an Ising model (Fang and Kim, 2016). The average computation time for the given settings is around 2.3 minutes with a 6-second standard deviation for the F-McMC algorithm and around 59 seconds with a 1-second standard deviation for the M-MCVB algorithm on a desktop with Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz, 32.0 GB RAM, 64-bit Windows 10. As such, we proceed throughout the remaining simulations and real data application with the M-MCVI-based version of our algorithm.

2.5.2 Comparison of MLNR with alternatives

In our simulations, we mainly utilize the VI based approach, which we do not consider a limitation on the generalizability of the simulation results given Theorem 3. For our simulations, we consider $|\mathcal{P}| = 25$. This value was chosen randomly, with the only restrictions considered being $|\mathcal{P}| \geq 2$, as a single set makes the first level of selection redundant, and that $|\mathcal{P}|$ is not too large as to prohibit an average user device from replicating our work. We chose the first two pathways to be the relevant pathways, while the remaining 23 contains irrelevant. For each of the relevant sets, we generated 4 elements. For each of the irrelevant sets, we generated 10 elements. The total number of covariates, therefore, was 238. We set the number of observations (n) to be 100; these values maintain the high-dimensional setting while being computationally feasible given the $\mathcal{O}(n^3)$ time complexity associated with recovering the kernel weights.

With respect to our dependent variables, we adopt the same generation procedure as in

Section 2.5.1, such that

$$\underline{y} = 10 \cos(\underline{x}_{(1,1)}) + 3\underline{x}_{(1,2)}^2 + \underline{x}_{(1,1)}\underline{x}_{(1,2)}\underline{x}_{(1,4)} - \frac{1}{4} \sin(\underline{x}_{(2,3)})\underline{x}_{(2,2)} + \frac{2}{5}\underline{x}_{(2,2)}\underline{x}_{(2,3)}\underline{x}_{(2,4)} + \underline{\epsilon}.$$

Once more, we generate $\underline{\epsilon}$ from two different scenarios: the first being a standard normal distribution and the second being a symmetric standard LaPlace distribution. The average computation time for the given settings is around 12 minutes with a 40-second standard deviation on a desktop with Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz, 32.0 GB RAM, 64-bit Windows 10.

	$\epsilon \sim N$		$\epsilon \sim ALD$		
$L \sim N$	McMC	VB	McMC	VB	Truth
γ_1	90	84	85	86	100
γ_2	39	44	50	49	100
γ_3	11	23	13	22	0
γ_4	13	6	8	9	0
γ_5	11	2	6	4	0
$\xi_{1,1}$	84	60	84	56	100
$\xi_{1,2}$	89	76	85	69	100
$\xi_{1,3}$	54	1	65	2	0
$\xi_{1,4}$	81	46	82	61	100
$\xi_{2,1}$	35	7	46	5	0
$\xi_{2,2}$	39	34	47	43	100
$\xi_{2,3}$	39	36	48	38	100
$\xi_{2,4}$	39	33	48	35	100
ξ_{rest}	11	4	8	4	0

Table 2.1: Simulation results comparing selection results from the exact posterior estimation via McMC to the approximate posterior estimation results via VB estimation.

	$\epsilon \sim N$		$\epsilon \sim ALD$		
$L \sim ALD$	McMC	VB	McMC	VB	Truth
γ_1	84	91	91	88	100
γ_2	56	59	51	55	100
γ_3	4	1	16	6	0
γ_4	8	3	8	5	0
γ_5	8	3	9	9	0
$\xi_{1,1}$	84	56	89	53	100
$\xi_{1,2}$	84	70	91	64	100
$\xi_{1,3}$	78	4	85	1	0
$\xi_{1,4}$	83	51	89	53	100
$\xi_{2,1}$	56	13	51	14	0
$\xi_{2,2}$	56	44	51	33	100
$\xi_{2,3}$	56	43	51	36	100
$\xi_{2,4}$	55	41	51	35	100
ξ_{rest}	7	1	11	3	0

Table 2.2: Simulation results comparing selection results from the exact posterior estimation via McMC to the approximate posterior estimation results via VB estimation.

2.5.3 Simulation Results

We generate 100 datasets each from normal and symmetric standard LaPlace distribution. We compare our method (MLNR) with four others (Lasso, Group Lasso, Sparse Group Lasso, Spike-and-Slab Group Lasso) in terms of four evaluation metrics for both set and element variables. We note that some methods, such as Lasso, are single-level variable selection methods. Group Lasso-type methods can be used only for higher-level variable selection. However, our method is for multi-level variable selection. To compare at the higher level, we

take the inclusion of any element by these methods as an inclusion of the associated higher level as well.

For a normal error case, Figure 2.1 displays the average values of four metrics of each method for element selections across all 100 simulations, while Figure 2.2 shows those of each method for the set selections. For a symmetric standard LaPlace error case, Figures 2.3-2.4 display the average values of four metrics of each method for element selections and for the set selections, respectively. The detailed average values of the four evaluation metrics are summarized in Tables 2.3-2.6.

		MLNR	Lasso	GL	SGL	SSGL
Sets	TPR	0.72	0.25	0.935	0.575	0.555
	FPR	0.054	0.0	0.177	0.0	0.004
	PPV	0.609	0.4	0.467	0.83	0.765
	F1	0.619	0.3	0.57	0.66	0.625
Elements	TPR	0.565	0.628	0.935	0.79	0.78
	FPR	0.017	0.03	0.184	0.031	0.197
	PPV	0.582	0.448	0.246	0.485	0.212
	F1	0.542	0.49	0.339	0.567	0.231

Table 2.3: Simulation results for the normal distribution noise, normal distribution likelihood: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso.

		MLNR	Lasso	GL	SGL	SSGL
Sets	TPR	0.7	0.245	0.975	0.605	0.59
	FPR	0.053	0.0	0.195	0.0	0.002
	PPV	0.595	0.42	0.467	0.87	0.75
	F1	0.608	0.303	0.576	0.693	0.643
Elements	TPR	0.578	0.643	0.975	0.825	0.793
	FPR	0.017	0.03	0.202	0.045	0.206
	PPV	0.565	0.481	0.261	0.467	0.198
	F1	0.552	0.505	0.352	0.547	0.216

Table 2.4: Simulation results for the asymmetric LaPlace distribution noise, normal distribution likelihood: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso.

		MLNR	Lasso	GL	SGL	SSGL
Sets	TPR	0.710	0.270	0.960	0.595	0.530
	FPR	0.003	0.0	0.190	0.0	0.006
	PPV	0.982	0.480	0.438	0.950	0.715
	F1	0.797	0.340	0.550	0.713	0.588
Elements	TPR	0.505	0.665	0.960	0.827	0.777
	FPR	0.002	0.032	0.197	0.039	0.208
	PPV	0.968	0.474	0.224	0.478	0.217
	F1	0.610	0.519	0.314	0.571	0.223

Table 2.5: Simulation results for the normal distribution noise, asymmetric LaPlace distribution likelihood: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso.

		MLNR	Lasso	GL	SGL	SSGL
Sets	TPR	0.745	0.315	0.955	0.585	0.505
	FPR	0.002	0.0	0.188	0.0	0.006
	PPV	0.980	0.570	0.472	0.890	0.683
	F1	0.821	0.400	0.575	0.687	0.559
Elements	TPR	0.536	0.668	0.955	0.810	0.757
	FPR	0.001	0.037	0.194	0.041	0.214
	PPV	0.971	0.424	0.260	0.474	0.211
	F1	0.647	0.494	0.351	0.560	0.219

Table 2.6: Simulation results for the asymmetric LaPlace distribution noise, asymmetric LaPlace distribution likelihood: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso.

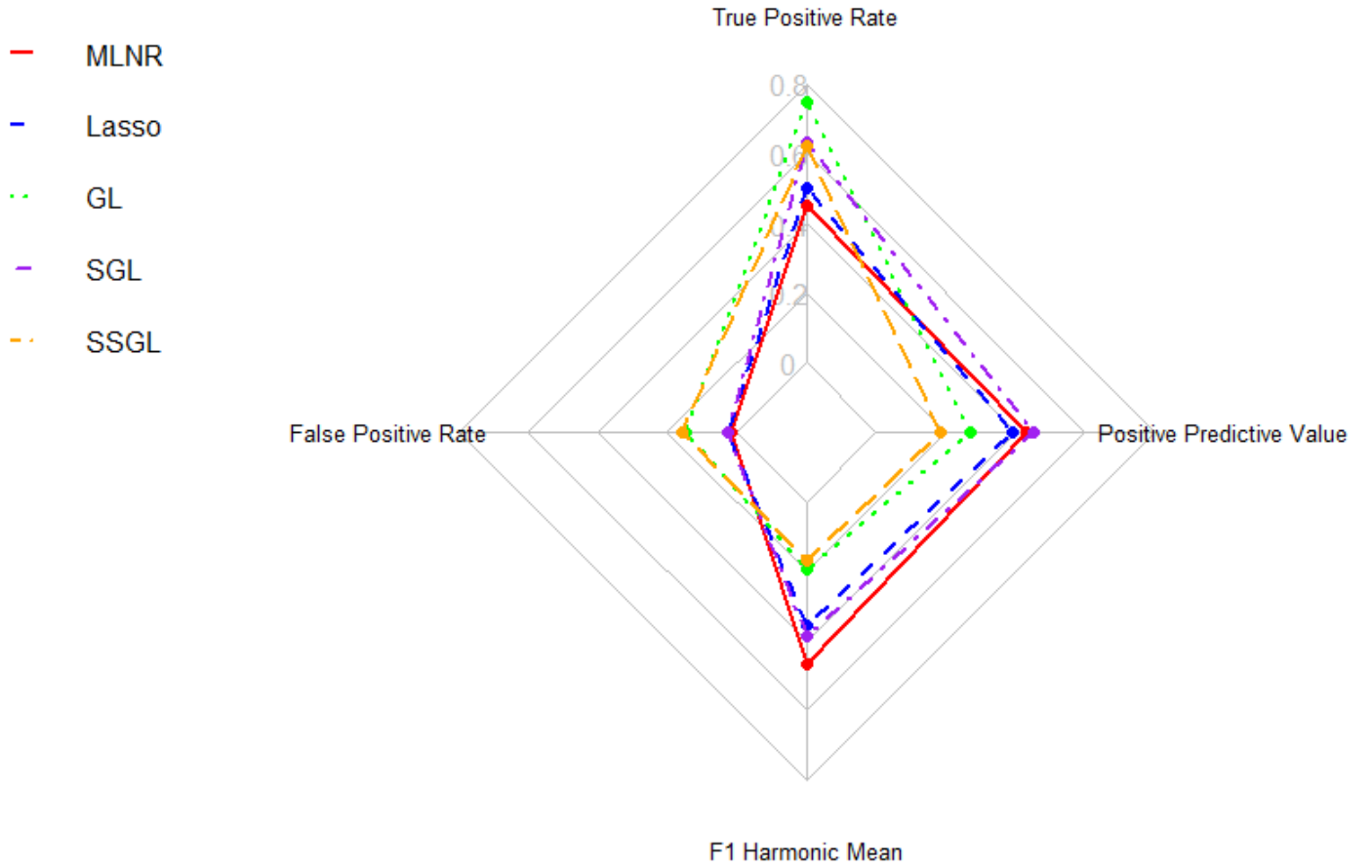


Figure 2.1: Simulation results for the normal distribution noise, normal distribution likelihood element case: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso.

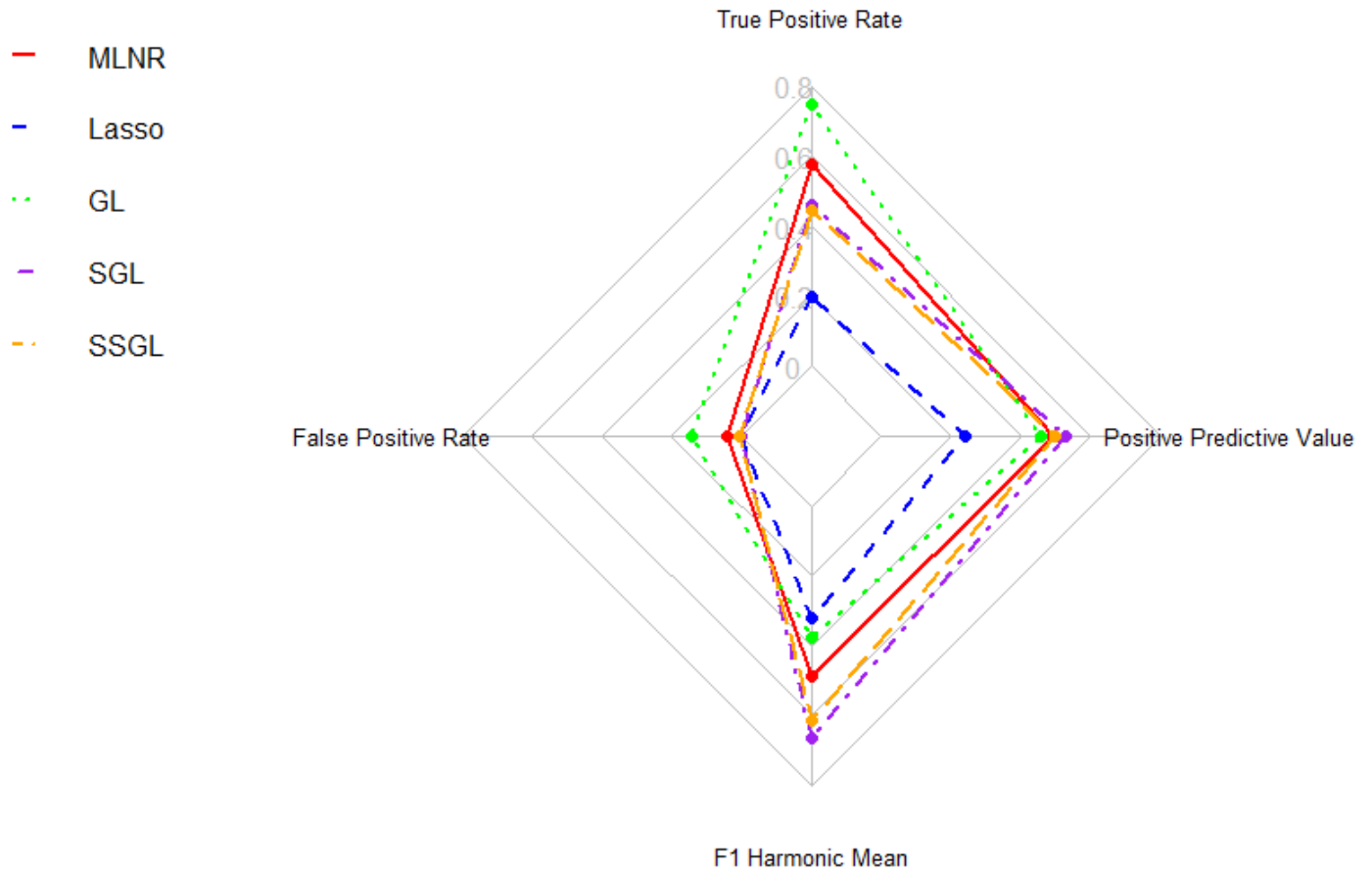


Figure 2.2: Simulation results for the normal distribution noise, normal distribution likelihood set case: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso.

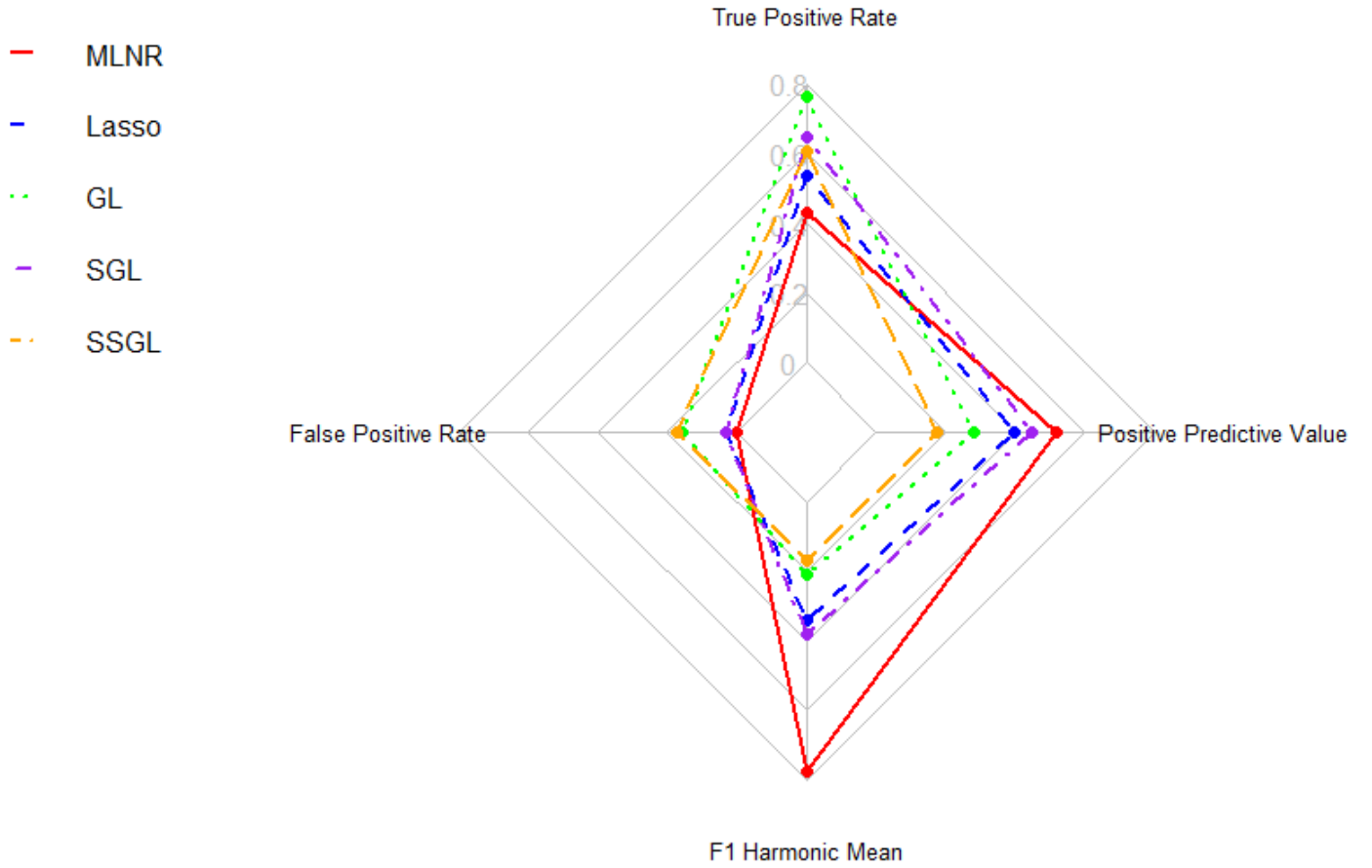


Figure 2.3: Simulation results for the asymmetric LaPlace distribution noise, asymmetric LaPlace distribution likelihood element case: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso.

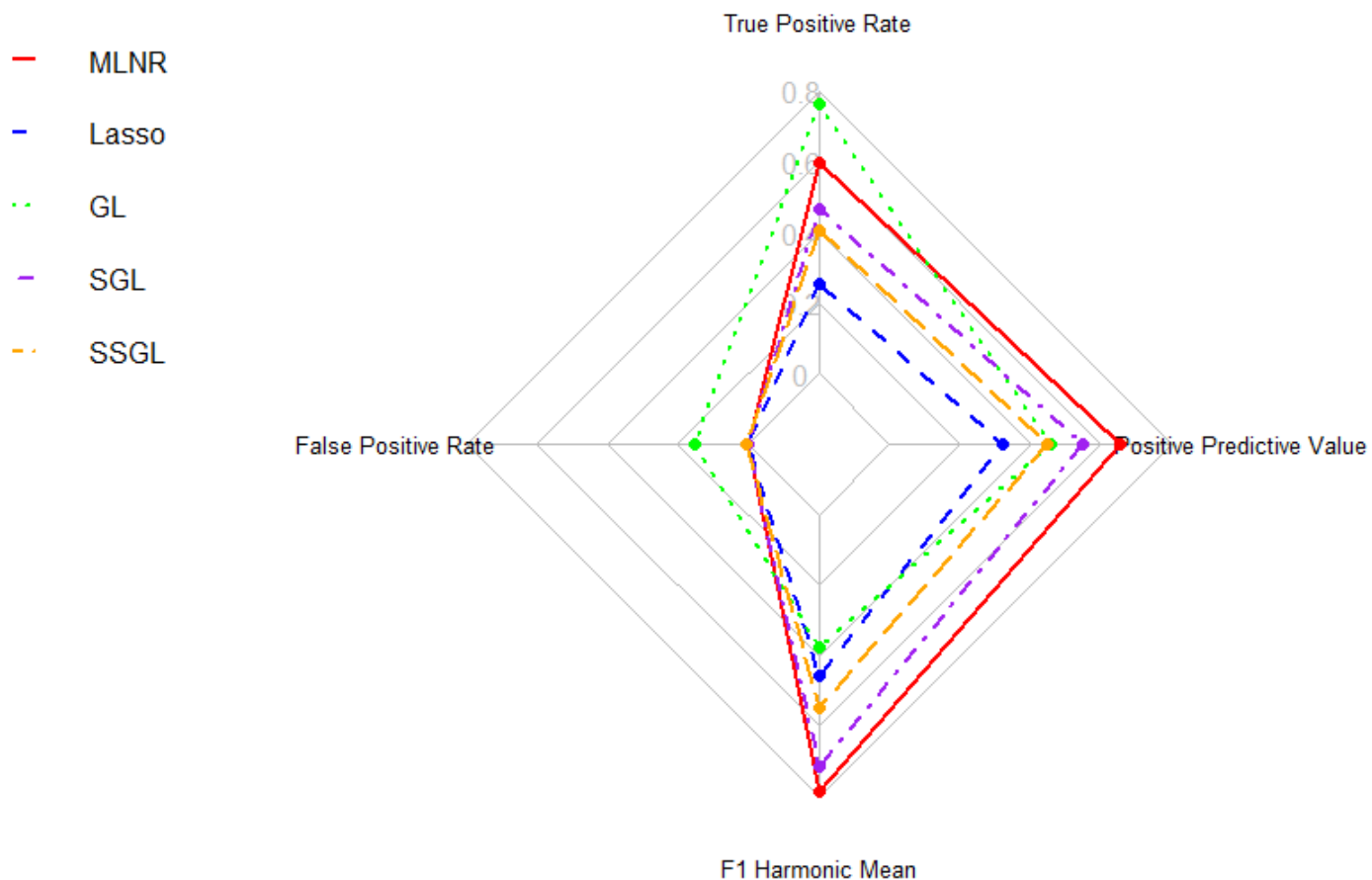


Figure 2.4: Simulation results for the asymmetric LaPlace distribution noise, asymmetric LaPlace distribution likelihood set case: MLNR=multi-level Network Recovery; Lasso; GL=Group Lasso; SGL=Sparse Group Lasso; SSGL=Spike and Slab Group Lasso.

Examining element performance, we observe that when the data is generated with a normally distributed noise component, our method (implemented with a Gaussian prior on kernel weights) produces more conservative recall rates (true positive rates) than competing methods; this is due to the adjustment of the uncertainty of including a set and the feedback

loop between set and element selection. On the other hand, our method excels at controlling FPR while maintaining high precision (PPV), which is comparable to the best-competing methods. In terms of set performance, our method’s two-level nature distinguishes itself. The two-level selection structure allows us to control the FPRs better on set selection, producing far better performance than competing methods that are either restricted to sets or elements. This also feeds into element selection, causing our positive selections to have low uncertainty, as indicated by high PPV values for set selection.

When employing a symmetric Laplace distribution, our method becomes more robust, producing higher recall (TPR) values with even lower FPR. While Group Lasso still maintains the best recall capabilities, this comes at the cost of significantly higher false positive rates. In this regard, our method offers a worthy compromise, sacrificing some recall capability to greatly limit type-I error. Despite this more conservative approach, the set selection of our method still significantly outperforms competing methods on TPR, while for the element case, which is dampened by uncertainty in set selection, true positive rates are comparable to competing methods.

2.5.4 Visualization for Multi-Level Network

We also create a visualization tool to display a multi-level network model using R software. Our visualization tool has been developed specifically for multi-level network problems. In this visualization tool, large circles represent the sets, while the dots within a set represent the elements. A sample output is displayed in Figure 2.5. This figure has three sets: one set has a large number of elements, and two sets have a small number of elements. The connections displayed among the elements indicate that the model has identified a relevant network among these elements, and the absence of connections across sets indicates that the model finds no supporting evidence for interaction between them. We use this visualization tool to identify the multi-level networks in our application as well.

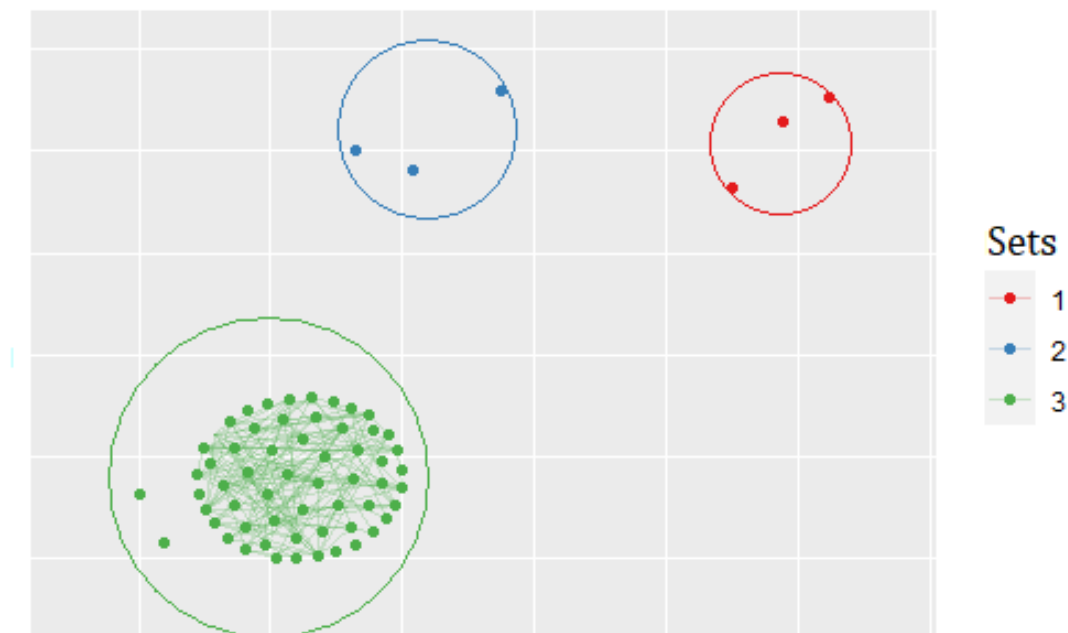


Figure 2.5: Sample output from the algorithm showing the elements nested within the sets. Connected dots indicate both a connection among the elements and to the response variable. In this run, only set 3 was connected to the response variable. This is indicated by the connection of a subset of the elements of set 3 and the lack of connections between the elements of sets 1 and 2. We note that this visualization approach will be misleading when a single element is selected from a pathway; however, this is a highly unlikely scenario in the application for which this method was developed (gene and pathway selection)

2.6 Real Data Application: Gene-pathway networks associated with Type 2 Diabetes Mellitus

In this section, we apply our method to pathway-gene-expression profile data associated with Type 2 diabetes mellitus (DM-2) (Mootha et al., 2003). The motivation is identifying gene expressions - within predefined pathways - that show association with fasting blood glucose levels, a marker which may indicate the presence of type-II diabetes. The data lends itself to the correlation assumptions inherent in our modelling strategies as gene sets that show co-regulated expressions exist. The dataset under examination consists of 651 genes and 10 pathways, forming the covariates of interest. Our sample size is 35 individuals (17 normal samples and 18 with Type 2 diabetes). Due to the fact that higher-level variables

(pathways) work together to accomplish certain tasks with other pathways and lower-level variables (genes), simultaneously exploring dependency structures among higher-level and lower-level variables is the main interest.

Because we do not have knowledge of the true contributing genes and pathways, we use some knowledge from the literature. There are pathways believed to be contributors based on previous statistical evidence. We observe and assess the results of our method against evidence from the literature for pathways involved in modulating fasted glucose levels. (Kim et al., 2012, Mootha et al., 2003, Pang et al., 2015).

In our study, Y is a 35×1 vector of the fasted glucose level, $\mathbf{X}_p$ is a $35 \times L_p$ matrix of gene expression, where $p = 1, \dots, 10$ (pathways) and L_p varies from 16 to 133, leading to a total of 651 individual elements. We identify pathways-genes associated with glucose levels. Information on ten pathways was collected from the literature. The names of the pathways and the number of genes within each pathway are summarized in Tables 2.7 and 2.8. We also summarized the results of methods based on Random Forests (RF), Lasso, GL, SGL, and SSGL. For our MLNR method, we construct our Ising prior weight matrix among pathways by using the average value of all pairwise correlations between genes within a pathway. That is, if pathways i and j have L_i and L_j genes, respectively, then we average across $(L_i L_j / 2)$ pairs. We applied our MLNR selection approach using the M-MCVB algorithm. Genes for unselected pathways are automatically not selected.

Pathway ID	No. of Genes	Pathway Names	MLNR _{MVN}	MLNR _{ALD}	RF	Lasso	GL	SGL	SSGL
230	121	OXPHOS HG-U133A probes	✗	✗	✓	✗	✗	✓	✗
229	133	Oxidative Phosphorylation	✗	✗	✓	✓	✗	✓	✗
110	71	JAK-Stat signaling pathway	✗	✗	✓	✓	✗	✓	✗
44	67	c23 U133 probes	✗	✗	✓	✓	✗	✓	✗
43	95	c22 U133 probes	✓	✗	✓	✓	✗	✓	✗
16	49	ATP Synthesis	✗	✗	✓	✓	✗	✓	✗
232	40	Parkinson's disease	✗	✗	✓	✗	✗	✓	✗
88	41	Gamma-Hexachlorocyclohexane Degradation	✗	✗	✓	✓	✗	✓	✗
4	18	Alanine and Aspartate Metabolism	✓	✓	✗	✗	✓	✓	✗
272	16	Ubiquinone Biosynthesis	✗	✗	✗	✗	✓	✓	✗

Table 2.7: A real application result using competing methods: comparison of competing methods. For methods that are only selected at the gene level, we consider any single gene selection as a pathway selection in this table.

2.6. REAL DATA APPLICATION: GENE-PATHWAY NETWORKS ASSOCIATED WITH TYPE 2 DIABETES MELLITUS

45

Pathway ID	No. of Genes	Pathway Names	MLNR _{MVN}	MLNR _{ALD}	RF	Lasso	GL	SGL	SSGL
230	121	OXPHOS HG-U133A probes	✗	✗	✗	✗	✗	✗	✗
229	133	Oxidative Phosphorylation	✗	✗	✗	✗	✗	✗	✗
110	71	JAK-Stat signaling pathway	✗	✗	✗	✗	✗	✗	✗
44	67	c23 U133 probes	✗	✗	✗	✗	✗	✗	✗
43	95	c22 U133 probes	✓	✗	✗	✗	✗	✗	✗
16	49	ATP Synthesis	✗	✗	✗	✗	✗	✗	✗
232	40	Parkinson's disease	✗	✗	✗	✗	✗	✗	✗
88	41	Gamma-Hexachlorocyclohexane Degradation	✗	✗	✗	✗	✗	✗	✗
4	18	Alanine and Aspartate Metabolism	✓	✓	✗	✗	✓	✗	✗
272	16	Ubiquinone Biosynthesis	✗	✗	✗	✗	✓	✗	✗

Table 2.8: A real application result using competing methods: comparison of competing methods. For methods that only select at the gene level, we consider only all gene selections as a pathway selection in this table.

We also compare our method's performance under an assumed Multivariate Normal Likelihood and a Symmetric LaPlace likelihood, shown in 2.9. As evident in the simulations, assuming an (A)symmetric LaPlace distribution results in a more conservative selection procedure, likely arising from the robustness of the distribution to outliers.

Pathway ID	Pathway Names	Gene ID	MLNR _{MVN}	MLNR _{ALD}
4	Alanine and Aspartate Metabolism	NARS	✓	✓
4	Alanine and Aspartate Metabolism	DARS...7	✓	✗
4	Alanine and Aspartate Metabolism	ADSL...9	✓	✓
4	Alanine and Aspartate Metabolism	PC	✓	✓
4	Alanine and Aspartate Metabolism	CRAT...11	✓	✓
4	Alanine and Aspartate Metabolism	ASPA	✓	✓
4	Alanine and Aspartate Metabolism	ABAT	✓	✓
4	Alanine and Aspartate Metabolism	GAD2...14	✓	✓
4	Alanine and Aspartate Metabolism	CRAT...17	✓	✗
4	Alanine and Aspartate Metabolism	ADSL...18	✓	✗
4	Alanine and Aspartate Metabolism	AGXT	✓	✗
4	Alanine and Aspartate Metabolism	GAD2...20	✓	✓
43	c22 U133 probes	SDHC...102	✓	✗
43	c22 U133 probes	SDHD...103	✓	✗
43	c22 U133 probes	ATP5J2...118	✓	✗
43	c22 U133 probes	NDUFC1...123	✓	✗
43	c22 U133 probes	UQCRB...135	✓	✗

Table 2.9: A comparison between genes selected under the multivariate normal likelihood assumption and the asymmetric LaPlace likelihood assumption.

When examining the selection of any gene from a pathway, we observe that with the exception of group lasso, Random Forest, Lasso, and Sparse Group Lasso tend to select at most 1-3 genes from each pathway, while Spike-and-Slab Group Lasso fails to make any selections given the dimensionality of the data. Group Lasso overlaps with our own method in

one of its two selected pathways. However, the second pathway it selects seems an unlikely candidate for influencing fasted blood glucose levels. All methods did not select the Oxidative Phosphorylation pathway identified by (Kim et al., 2012, Mootha et al., 2003, Pang et al., 2015). However, both Group Lasso and MLNR have selected Alanine and Aspartate metabolism, which is supported by multiple studies. (Chen et al., 2008, Kawamoto et al., 2012, Schindhelm et al., 2006)

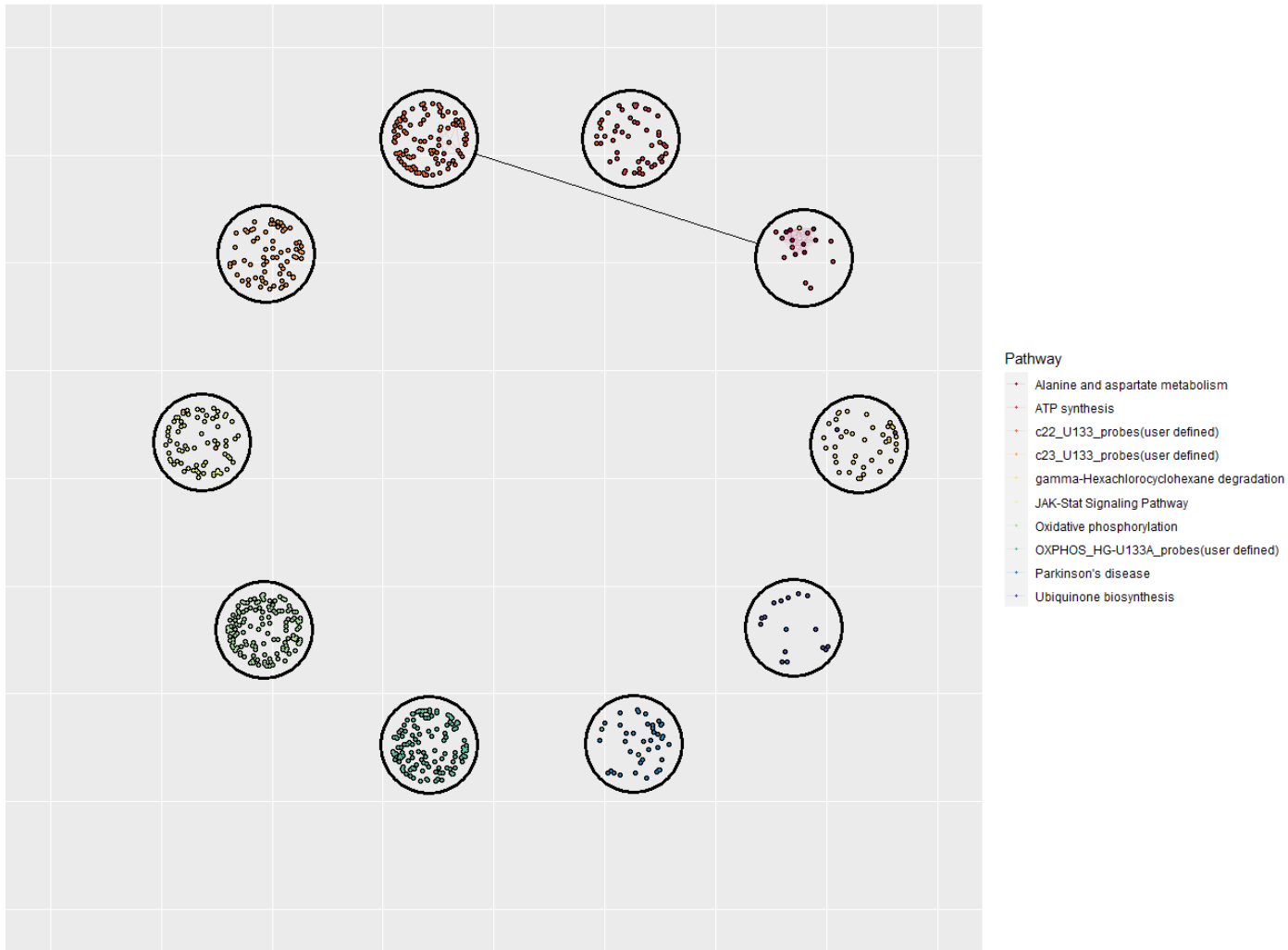


Figure 2.6: Sample output from the Variational Bayes algorithm for estimating the expectation of the response, blood glucose level. It shows the elements nested within the sets. Connected dots indicate both a connection among the elements/sets and to the response variable. These results visualize the MLNR (MVN) results from tables 2.7, 2.8.

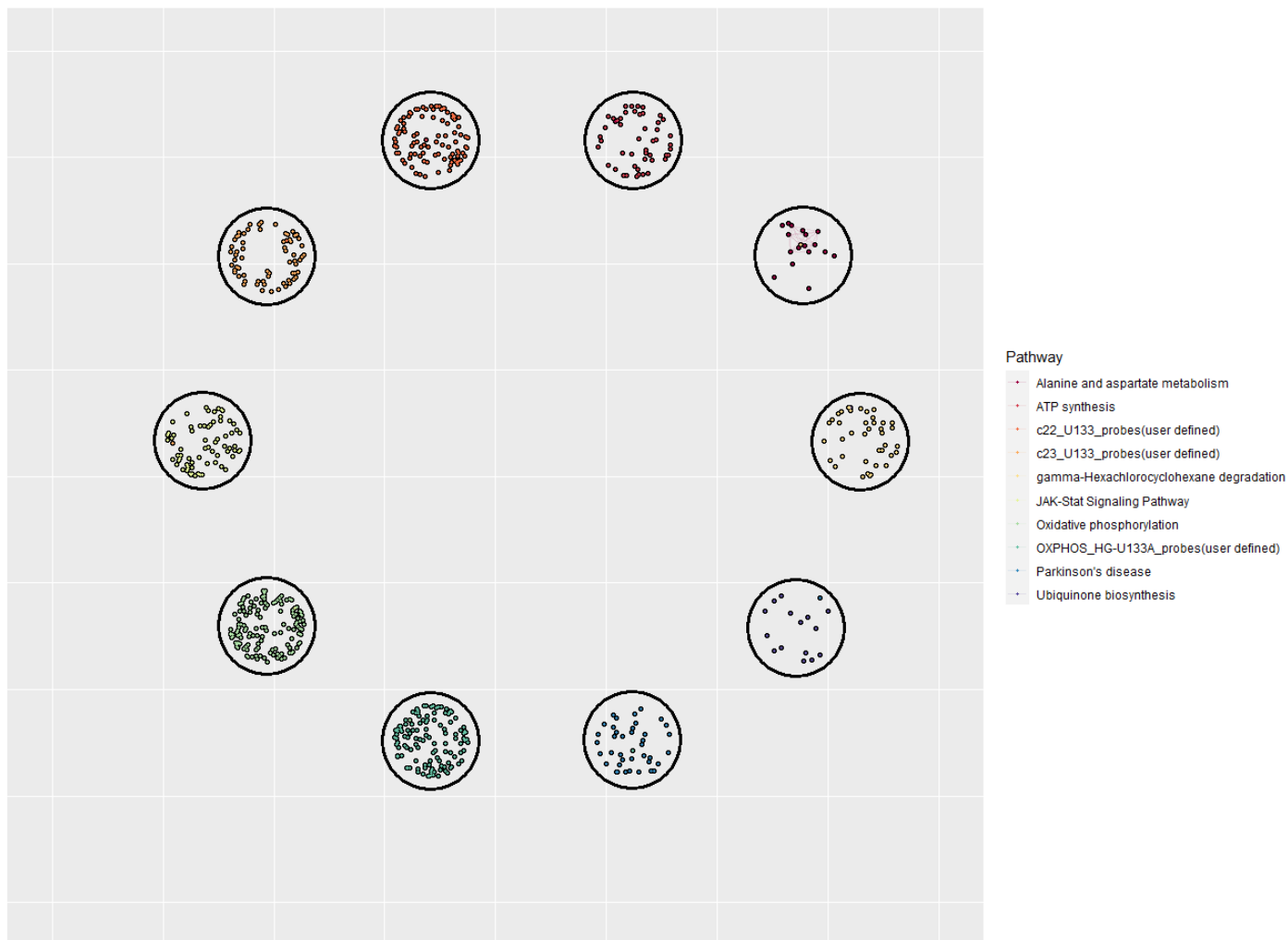


Figure 2.7: Sample output from the Variational Bayes algorithm for estimating the 0.5 quantile of the response, blood glucose level. It shows the elements nested within the sets. Connected dots indicate both a connection among the elements/sets and to the response variable. These results visualize the MLNR (ALD) results from tables 2.7, 2.8.

2.7 Discussion

In this chapter, we proposed a multi-level nonparametric kernel machine approach, utilizing both MCMC and VB to identify jointly multi-level variables as well as build the network. The variational inference approach is novel in its utilization of the sampled set structure within the proposed spike-slab prior. In addition to the variable selection and network recovery

capabilities, our approach is capable of producing both mean and quantile estimations of the original response variable of interest for a multiple kernel machine regression. In addition to reducing computational cost by employing a variational approach and proposing a multi-kernel quantile regression approach, our method shows better performance than some other potential approaches for variable selection in a multi-level correlated variable setting.

We have developed our method using a univariate response variable. However, we can observe multiple response values. In this case, we can extend our approach to the multivariate response variables for a multitask learning approach. We will consider this problem for future research.

Chapter 3

Bayesian Multi-Tasking Inference using High-dimensional Correlated Functional Variables

3.1 Introduction

The task of variable selection in the field of statistics has a rich and well-studied background. There are numerous methods that operate through modifying the likelihood under consideration by introducing penalty terms. For example, Ridge regression (Hoerl and Kennard, 1970), the Lasso (Tibshirani, 1996), Elastic Net (Zou and Hastie, 2005), group-lasso (Zou, 2006), fused lasso (Tibshirani et al., 2005), and other variational methods Reid and Tibshirani (2016), She (2010) were developed. This penalty can be viewed as the priors for Bayesian variable selection approaches, which involve the specification of a prior such as spike-slab prior, either over binary selection variables (Bai et al. (2020), Chen et al. (2016), George and McCulloch (1993) or over model selection (Kuo and Mallick, 1998). The introduction of a prior in the former also generates a penalized likelihood, however, inference is typically done on the posterior distribution of the binary selection variables, while the latter approach outputs a Bayes factor over pairwise comparisons of proposed models.

Variable selection methods are further extended to functional variable selection. Most functional variable selection methods are developed under the additive functional models with sparse penalty functions of coefficients of basis functions (Zhengwu Zhang and Zhu, 2022) or decomposing functional variables into scalar spaces (Haolun Shi and Cao, 2022). In these functional variable selection methods, given an appropriate approach has been chosen, the resulting functional coefficients automatically account for correlation across a functional covariate's domain through the introduction of smoothness assumptions inherent to the choice of basis function used to model the coefficient function. However, the presence of interactions across whole functional covariates is typically not considered in a model with multiple functional components. And also, these existing approaches tend to specify or estimate basis functions as coefficients for the functional covariates, where a zero value for the coefficient function at a point on the domain indicates the irrelevance of that point with respect to providing information about the response. Hence, the development of simultane-

ous selections of functional domain subsets, selections of functional variables, and networks among functional variables are still quite limited due to complications in modeling unknown interaction terms among high-dimensional and highly correlated functional variables. The development of functional variable selection in nonadditive and nonparametric models with high dimensional correlated functional variables is even challenged.

3.1.1 Our contribution

In this chapter, we develop a unified functional variable selection method that can simultaneously perform (1) the selections of functional domain subsets, (2) the selections of functional variables, and (3) the selection of networks among functional variables in the nonadditive and nonparametric model and (4) finally test whether these selected variable sets are significantly associated to a response variable.

Our approach seeks to avoid the specification of basis functions or, indeed, the imposition of any assumptions on both the functional covariate coefficients and the response variable. By incorporating a black box hyperparameter optimization problem, which we tackle using Bayesian optimization and discrete Ising graphical model, into a semiparametric kernel machine model framework, we propose a semiparametric joint functional kernel graphical model and then achieve our multi-tasking inference described in (1)-(4). To the best of our knowledge, no methods can achieve these multi-tasks simultaneously. Hence, in this chapter, we develop a Bayesian multi-tasking inferential method (BMIM) to provide proper posterior inference on domains, regions, networks, and the overall model simultaneously.

The development of BMIM is motivated by the functional magnetic resonance imaging (fMRI) dataset, which consists of both scalar demographic variables for autism spectrum disorder (ASD) study and functional brain imaging data formatted as functional covariates, all collected as part of a coordinated, multi-laboratory effort to study ASD. Existing literature on brain imaging data obtained via fMRI indicates the existence of interrelationships

between spatially separate brain regions, which tends to introduce correlation across in addition to within, the functional covariates (see *Functional Connectivity* e.g. Allen et al. (2012), Rogers et al. (2007)). One would need to introduce additional assumptions to identify relevant functional covariates. A reasonable approach would be to introduce a group-lasso-like penalty over the functional coefficients, with groups being identified by degree of functional connectivity.

Our use of Bayesian optimization for domain selection is rooted in machine learning literature, particularly hyperparameter optimization (Brochu et al. (2010), Shahriari et al. (2016), Snoek et al. (2012)). Our approach is tantamount to a sliding window approach where Bayesian optimization is employed to select the size and placement of the window by constructing a surrogate for the validation error of a scalar response on a single-component-functional-covariate regression.

We also utilize the Bayesian approach adapted from Fang and Kim (2016) to account for functional connectivity by introducing binary selection variables with a discrete Ising graphical model linking various brain regions based on their degree of functional connectivity. Within our algorithm, the prior acts to increase the posterior probability of selection on the binary variables whenever their “neighbors” (referring to other, spatially distinct brain regions that tend to simultaneously activate in relation to given tasks) also show a high posterior probability of selection.

Our implementation of the Ising model utilizes a Markov-chain Monte Carlo sampler (McMC) with a proposal distribution acting as a regularizer by first approximating the posterior on our selection variables using a mean-field approximation variational Bayes inference approach adapted from Murphy (2013), and built on a simpler model. This limits the exploration of the McMC algorithm, causing it to focus on the exploitation of covariates that are promising even when using less flexible models. Domain selection is carried out prior to brain region selection and follows Jones et al., such that the region selection operates on

screened functional covariates.

Finally, we supplement our approach with an overall, score-based test of significance for our semi-parametric functional kernel graphical model, which will be explained in Section 3.2. The score-based test was chosen for its complementarity with Gaussian processes, which provides a natural extension to our overall modeling strategy. Our method, therefore, is simultaneously selecting functional variables from P potential candidates by exploring the brain region configuration space of $\boldsymbol{\xi} = (\xi_1, \dots, \xi_P)^T$ as well as selecting a relevant domain subset from the P potential candidates by exploring the configuration space of our sliding window hyperparameters.

3.1.2 Outline of the chapter

This chapter is organized as follows: In Section 3.2, we propose a semiparametric joint functional kernel graphical model; in Section 3.3, we describe the domain selection method based on Bayesian optimization. In Section 4.2, we explain the functional node and graph selection for brain region selection. The Markov-chain Monte Carlo (MCMC) based inference and algorithm and also the variational Bayes inference (VB) derived proposal and algorithm are provided in 3.4.1 and 3.4.1, respectively. In Section 3.5, we provide a score-type test to determine whether the selections are significantly associated with the response. In Section 3.7, we provide the asymptotic properties. In Section 3.8, we conduct the simulation studies to evaluate the performance of our proposed method. In Section 3.9, we demonstrate the advantage of our method using resting state fMRI brain activity related to autism spectrum disorder (ASD). Lastly, our concluding remarks are presented in Section 3.10.

3.2 Model Structure

Let $\underline{\mathbf{y}}$ be an $N \times 1$ vector of continuous response variable, where $\underline{\mathbf{y}} = [y_1, \dots, y_i, \dots, y_N]^T$. Let $X_i^{(p)}(t)$ be the functional variable for the i th individual obtained at the p^{th} region ($p = 1, \dots, P$) on a time interval $t \in [1, T]$, $i = 1, \dots, N$. We define $\mathbf{X}^{(p)} = [\mathbf{X}_1^{(p)}, \dots, \mathbf{X}_N^{(p)}]^T$ to be the functional covariates generated at the p^{th} region for all N individuals across the full time interval. We assume all individuals are observed at the same time interval. Let $\mathbf{Z}$ represent individual-specific scalar covariates, where for individual i we observe $\mathbf{z}_i = [z_{i1}, \dots, z_{iS}]$, where S is the number of scalar covariates.

3.2.1 Semiparametric Joint Functional Kernel Graphical Model

We consider the following semiparametric functional machine learning model, which consists of two components, the scalar and the functional covariates

$$\underline{\mathbf{y}} = \mathbf{Z}\underline{\boldsymbol{\beta}} + \underline{\mathbf{h}}(\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(P)}) + \underline{\boldsymbol{\epsilon}}, \quad \underline{\boldsymbol{\epsilon}} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}),$$

where $\boldsymbol{\beta}$ is unknown coefficient parameter, σ^2 is a unknown variance parameter, and $\underline{\mathbf{h}}(\dots)$ is an unknown and nonadditive function of P dimensional high dimensional functional variables.

To select for relevant regions, we introduce the binary parameter ξ_p , $\xi_p = \{0, 1\}$, $p = 1, \dots, P$, indicating the presence of region p which follows the Boltzmann distribution of the Ising graphical model (ISGM). Define $\tilde{\xi}_p = (-1)^{\xi_p+1}$ and $\tilde{\xi}_p \in \{-1, 1\}$. Given a graph $G = (V, E)$, where V represents vertex (or node) and E represents discrete edges, we consider the Boltzman distribution, $P_{\boldsymbol{\xi}}(\xi_1, \dots, \xi_P) \propto \exp\{\sum_{s \in V} q_s \tilde{\xi}_s + \sum_{(s,l) \in E} w_{sl} \tilde{\xi}_s \tilde{\xi}_l\}$, where q_s and w_{sl} are the elements of vector $\mathbf{q}$ and the elements of the interaction matrix $\mathbf{W}$. Here $\mathbf{q}$ is called an “external field”, $\mathbf{W}$ is the interaction matrix, and $\boldsymbol{\xi} \in R^{|V|+|E|}$. In our study, p th region refers to p th node, and interaction among regions refers to graph/network. Hence

$|V| = P$ and $|E| = P(P - 1)/2$.

We also introduce the notation $[a_p, b_p]$, which indicates the selected time domain within the functional domain subspace of the p th region. By jointly combining the semiparametric functional machine learning model with ISGM, we propose a semiparametric joint functional kernel graphical model,

$$\mathbf{E}(\mathbf{y}|\mathbf{Z}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(P)}) = \mathbf{Z}\underline{\boldsymbol{\beta}} + \mathbf{h}\left(\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(P)}, \xi_1 \dots, \xi_P, t_p \in [a_p, b_p]\right). \quad (3.1)$$

Since $\mathbf{h}(\dots)$ is an unknown and nonadditive function of some subset of the P dimensional high dimensional functional variables, we assume that $\mathbf{h}(\dots)$ is in Hilbert Kernel space, $\mathcal{H}_{\mathcal{K}}$. Using the reproducing property, our model can be re-expressed using some kernel matrix $\mathbf{K}$ as

$$\mathbf{E}(\mathbf{y}|\mathbf{Z}, \mathbf{X}) = \mathbf{Z}\underline{\boldsymbol{\beta}} + \mathbf{K}\left(\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(P)}, \xi_1 \dots, \xi_P, t_p \in [a_p, b_p]\right)\underline{\boldsymbol{\alpha}}, \quad (3.2)$$

where $\underline{\boldsymbol{\alpha}}$ are the unknown kernel weights.

We build our kernel matrix $\mathbf{K}$ using a binary parameter ξ_p indicating the presence of region p , and a functional-input radial basis function (RBF) kernel, in which the (i, i') th element of $\mathbf{K}$ consists of $k(\mathbf{X}_i, \mathbf{X}_{i'})$, $i, i' = 1, \dots, N$, where

$$k(\mathbf{X}_i, \mathbf{X}_{i'}) = \exp\left(-\frac{\sum_{p=1}^P \xi_p \left[\int_{t \in [a_p, b_p]} \frac{1}{b_p - a_p} \{X_i^p(t) - X_{i'}^p(t)\} dt\right]^2}{d}\right) \quad (3.3)$$

and

$$k(\mathbf{X}_i, \mathbf{X}_{i'}) = \exp\left(-\frac{\sum_{p=1}^P \xi_p \left[\int_{t \in [a_p, b_p]} \frac{1}{b_p - a_p} \{X_i^p(t) - X_{i'}^p(t)\} dt\right]^2}{d} - \frac{\sum_{p=1}^P \xi_p \left[\int_{t \in [a_p, b_p]} \{X_i^p(t) - \frac{1}{b_p - a_p} \int_{t \in [a_p, b_p]} X_i^p(t) dt\}^2 - \{X_{i'}^p(t) - \frac{1}{b_p - a_p} \int_{t \in [a_p, b_p]} X_{i'}^p(t) dt\}^2 dt\right]^2}{d}\right) \quad (3.4)$$

The kernel function in (3.3) contains the average of the differences between functional

variables obtained from i and i' th samples, while the kernel function in (3.4) includes both the average and variability of them. This semiparametric joint functional kernel graphical model (3.1) can be connected with a mixed model framework using a functional-input Gaussian process (f -GP) (Chih-Li Sung and Hungc, 2024). Because GP is a family of any unknown functions, we use a functional-input GP to approximate $\underline{\mathbf{h}}(\cdots)$ for correlated functional input variables. That is, $\underline{\mathbf{h}}(\cdots) \sim MVN(0, \tau^2 \mathbf{K})$, where $\mathbf{K}$ is an $N \times N$ matrix, the (i, i') entry of $\mathbf{K}$ is $k(\mathbf{X}_i, \mathbf{X}_{i'})$, and τ^2 is a unknown variance component parameter. Note that if $\tau^2 = 0$, we can say that the correlated functional variable set is not associated with the response. The equivalence of the least squares kernel machine regression (LSKM) with a functional-input Gaussian process regression (f -GPR) can be seen in the predictive model where for the LSKM we have $\hat{\underline{\mathbf{h}}} = \gamma^{-1} \mathbf{K}(\mathbf{I} + \gamma^{-1} \mathbf{K})^{-1}(\underline{\mathbf{y}} - \mathbf{Z}\hat{\underline{\boldsymbol{\beta}}})$ (Liu et al., 2007), where $\hat{\underline{\boldsymbol{\beta}}}$ is a LSKM estimator; while for the GPR, our posterior predictive mean is $\hat{\underline{\mathbf{h}}} = \mathbf{K}(\gamma \mathbf{I} + \mathbf{K})^{-1}(\underline{\mathbf{y}} - \mathbf{Z}\hat{\underline{\boldsymbol{\beta}}})$ (Gramacy, 2020). For scalar γ we see that these two quantities are the same.

Our semiparametric joint functional kernel graphical model framework allows a flexible function for the joint effect of multiple P functional nodes by specifying a functional-input kernel function and allows for the possibility that the relationship between each functional covariate and the response is nonlinear and does not preclude the nodes (regions) from interacting with each other in a complicated way.

3.3 Functional Domain Subspace Selection: $[a_p, b_p]$

We explain how to incorporate Bayesian optimization for functional domain subspace selection $[a_p, b_p]$. We assume that for all $t \in [1, T]$ and for each region p that has predictive power of the response, there exists some subset $[a_p, b_p] \subseteq [1, T]$ such that for all $t \notin [a_p, b_p]$, $X^{(p)}(t)$ has no predictive power of the response. For each p , we consider the following model

$$\mathbf{E}(\underline{\mathbf{y}}|\underline{\mathbf{Z}}, \mathbf{X}^{(\mathbf{p})}) \approx \underline{\mathbf{Z}}\underline{\boldsymbol{\beta}} + \mathbf{f}_p\left(X^{(p)}(t), t \in [a_p, b_p]\right), \quad (3.5)$$

where the choice of $\mathbf{f}_p(\cdot)$ can range from a fully parametric model (e.g., a functional linear model) to a fully non-parametric model (e.g., kernel machine model). For example, the model (3.5) can also be semiparametric kernel machine regression, but with a single functional covariate so that we build $\mathbf{f}_p(X^{(p)}(t), t \in [a_p, b_p]) = \mathbf{K}_p(X^{(p)})\underline{\alpha}_p$, where $\underline{\alpha}_p$ is a kernel weight vector.

We fit the model under different settings of a_p and b_p to enforce $a_p < b_p$, i.e., we work with a_p and s_p , where $b_p = a_p + s_p$, and solve Bayesian optimization to search for the optimal settings of a_p and b_p . We denote these optimal values a^* and $b^* = a_p^* + s_p^*$. To prevent our algorithm from selecting the whole region $[1, T]$, we need to incorporate an additional penalty term that encourages smaller domain selections. We utilize a functional-input Gaussian process regression (f -GPR). Assuming a functional-input kernel machine regression model, our setup is as follows,

$$\mathbf{f}_p(\cdot) \sim \mathcal{GP}(\underline{\mu}_p, \tau_p^2 \mathbf{K}_p + \sigma_p^2 \mathbf{I}),$$

where $\mathbf{f}_p$ follows a f -GP with mean $\underline{\mu}_p$ and covariance $\tau_p^2 \mathbf{K}_p + \sigma_p^2 \mathbf{I}$, where τ_p^2 is a scale parameter on $\mathbf{K}_p$, and σ_p^2 represents the variance of normal noise. Finally, since $\xi_p = 0$ implies $\tau_p^2 = 0$, then under the null assumption for a covariate, the covariate is generated by noise and is not relevant to the response.

We build a functional-input kernel machine regression using two types of kernel matrices for $\mathbf{K}_p$, denoted as $\mathbf{K}_p^1$ and $\mathbf{K}_p^2$. The (i, i') th element of $\mathbf{K}_p^1$ and $\mathbf{K}_p^2$ are defined as

$$\begin{aligned} \mathbf{K}_p^1(X_i^{(p)}, X_{i'}^{(p)}) &= \exp[-\{\int_{t=a_p}^{b_p} (X_i^p(t) - X_{i'}^p(t))dt\}^2]; \\ \mathbf{K}_p^2(X_i^{(p)}, X_{i'}^{(p)}) &= \exp\left(-[\int_{t=a_p}^{b_p} \frac{1}{b_p - a_p} \{X_i^p(t) - X_{i'}^p(t)\}dt]^2 \right. \\ &\quad \left. - [\int_{t=a_p}^{b_p} \{(X_i^p(t) - \frac{1}{b_p - a_p} \int_t X_i^p(t)dt\}^2 - \{X_{i'}^p(t) - \frac{1}{b_p - a_p} \int_t X_{i'}^p(t)dt\}^2 dt]^2\right). \end{aligned}$$

Integral is numerically calculated using Riemann sum approximation. The $\mathbf{K}_1^p$ contains the average of the differences between functional variables obtained from i and i' th samples, while $\mathbf{K}_p^2$ includes both the average and variability of them.

We perform k-fold cross-validation treating a_p and b_p as hyperparameters, and then our surrogate searches for the optimum setting of our hyperparameters by interpolating the root mean squared error (RMSE) as a function of the hyperparameters. We can incorporate the required domain selection penalty into the following objective function

$$\min_{a_p, b_p} \frac{1}{\mathcal{K}} \sum_{i \in \mathcal{K}} \left[\frac{\sum_{i=1}^{N_{\mathcal{K}}} \{y_i - \hat{\mathbf{E}}(\mathbf{y}|\mathbf{Z}, \mathbf{X}^{(p)})\}^2}{N_{\mathcal{K}}} \right]^{1/2} + \lambda(s_p - c)^2, \quad (3.6)$$

where $\mathcal{K}$ represents the number of cross-validation folds, $N_{\mathcal{K}}$ is the number of observations in the $\mathcal{K}^{th}$ fold, $\hat{\mathbf{E}}(y|\mathbf{Z}, \mathbf{X}^{(p)})$ represents a model's estimation of the expectation of the response variable, and λ is a penalty parameter on the size of the selected domain and c is a suggested anchor for the domain size such that larger values of λ push s_p towards c .

Here, λ is unlike typical penalty terms, which tend to push penalized parameters to zero; in the highly correlated structure of functional variable, a typical penalty parameter on the coefficients would reduce the optimization procedure to selecting the smallest domain that provides the highest prediction potential for the response which can, in turn, reduce the domain size to a singular point (which in a functional covariate setting, tends to capture overall local neighborhood correlations to the response). In our setting in equation (3.6), we push the hyperparameter s_p towards some predetermined minimum value c , where any expansions in the domain size will only be induced by large reductions to validation RMSE. The purpose of λ is not regularization for better generalization, rather it represents a trade-off between dimensionality reduction and model performance. Therefore, we cannot use typical cross-validation RMSE to establish a λ choice, as the optimal choice would be $\lambda = 0$. In contrast, a very high choice of λ leads to $s_p = c$ regardless of cross-validation RMSE values; as such, we propose varying λ and monitoring the variability in the value of s_p as a

guide for setting its value. We finally obtain a^* and b^* .

3.4 Functional Node and Graph Selection: $\xi_p, p = 1, \dots, P$

In this section, we lay out our proposed node selection method under the semiparametric joint kernel graphical model (3.1), utilizing both MCMC and VB, which jointly identify functional variables (nodes) and graphs among nodes that are associated to the response variable.

As described in the model (3.1), the approach proposed here includes a set of binary selection variables: $\underline{\xi}$. As is typical in Bayesian variable selection, we need to assign a prior distribution for our binary selection variables. Given the connected nature of our functional covariates, we assume an Ising prior to the form

$$p(\underline{\xi}) \propto \exp \left[\sum_{l < j} \tilde{\xi}_l \omega_{lj} \tilde{\xi}_j \right] \quad \xi_l \in \{0, 1\} \quad ; \quad l, j \in [1, \dots, P], \quad (3.7)$$

where $\tilde{\xi} = (-1)^{\xi+1}$ and $\tilde{\xi}_l \in \{-1, 1\}$.

Here, ω_{lj} is a term that captures a researcher's prior belief of the strength of the co-movement between elements l and j within the region p . Should the researcher assume zero co-movement across the elements, this prior reduces to the form of a discrete uniform distribution, while assuming a positive constant ω_{lj} for all pairs tilts the probability towards outcomes with many matching values of ξ . This formulation has been referred to as the Ising prior (Dai et al., 2013, Fang and Kim, 2016, Murphy, 2013) and is well suited for modeling the coordinated behavior of particles. We adopt this here to model the coordinated behavior of our functional covariates.

3.4.1 Markov chain Monte Carlo Estimation

Our McMC-based approach builds on the work of Gramacy and Lee (2008b), which provides robust full conditionals associated with the parameters in a GP model with a linear mean. While the original work uses a McMC scheme to explore tree space, we adopt a McMC scheme that utilizes the Ising prior to selecting correlated variables. Typically, when utilizing RBF kernels, automatic relevance determination (ARD) of each covariate is accomplished by instituting an anisotropic lengthscale and observing the resulting posterior samples of lengthscales associated with each covariate; Gramacy and Lee (2008a) extends on the former idea, allowing for some dimensions to be completely “turned off” in a sense, such that they only contribute to the response via a linear model. We note that the idea is not straightforward to adopt in our setting given the correlated nature of our covariates, as the individual lengthscale parameters would no longer be independent. In our study, we utilize an isotropic f -GP to increase the stability of the Gibbs samples of the lengthscale hyperparameter.

We can then express the model (3.1) as the following Bayesian hierarchical framework,

$$\begin{aligned} \underline{\mathbf{y}}|\underline{\boldsymbol{\beta}}, \tau^2, \mathbf{K} &\sim MVN(\mathbf{Z}\underline{\boldsymbol{\beta}}, \tau^2\mathbf{K}), & \tau^2 &\sim IG(\alpha_\tau/2, q_\tau/2) \\ \underline{\boldsymbol{\beta}}|\sigma_\beta^2, \tau^2, \underline{\boldsymbol{\beta}}_0, \mathbf{W} &\sim MVN(\underline{\boldsymbol{\beta}}_0, \sigma_\beta^2\tau^2\mathbf{W}), & \sigma_\beta^2 &\sim IG(\alpha_\beta/2, q_\beta/2) \\ \underline{\boldsymbol{\beta}}_0 &\sim MVN(\underline{\boldsymbol{\mu}}_0, \mathbf{B}_0), & \mathbf{W}^{-1} &\sim W((\rho\mathbf{V})^{-1}, \rho) \end{aligned} \quad (3.8)$$

where $\mathbf{W}$ is denoted as the covariance matrix of prior distribution $\boldsymbol{\beta}$ and MVN , IG , G , and W refer to the Multivariate Normal, Inverse-Gamma, Gamma, and Wishart distributions, respectively. $\rho, \mathbf{V}, \underline{\boldsymbol{\mu}}_0, \mathbf{B}_0, \alpha_\tau, q_\tau, \alpha_\beta, q_\beta$ are hyperparameters. The i, i' component of the kernel matrix $\mathbf{K}$ is

$$k(\mathbf{X}_i, \mathbf{X}_{i'}) + g\delta_{ij},$$

where δ_{ij} is the Kronecker delta function, and g is the nugget parameter (also known as

Tikhonov regularization parameter) where $\tau^2 g = \sigma^2$.

The prior distribution of (d, g) in kernel function is denoted as $p(d, g)$ which is specified as the following distribution

$$p(d, g) = p(d)p(g) = G(\alpha_g, q_g) \cdot \left[\frac{1}{2}G(\alpha_{d_1}, q_{d_1}) + \frac{1}{2}G(\alpha_{d_2}, q_{d_2}) \right],$$

where $\alpha_g, q_g, \alpha_{d_1}, q_{d_1}, \alpha_{d_2}, q_{d_2}$ are hyperparameters.

Given the above hierarchical Bayesian model, we obtain the following full conditional distributions for our parameters, which we implement in a Gibbs sampling scheme

$$\underline{\beta}|rest \sim MVN(\tilde{\underline{\beta}}, \tau^2 \mathbf{V}_{\tilde{\underline{\beta}}}), \quad \underline{\beta}_0|rest \sim MVN(\tilde{\underline{\beta}}_0, \mathbf{V}_{\tilde{\underline{\beta}}_0}), \quad (3.9)$$

where

$$\begin{aligned} \mathbf{V}_{\tilde{\underline{\beta}}} &= (\mathbf{Z}^T \mathbf{K}^{-1} \mathbf{Z} + \mathbf{W}^{-1}/\sigma_{\beta}^2)^{-1}, & \tilde{\underline{\beta}} &= \mathbf{V}_{\tilde{\underline{\beta}}}(\mathbf{Z}^T \mathbf{K}^{-1} \underline{\mathbf{y}} + \mathbf{W}^{-1} \underline{\beta}_0/\sigma_{\beta}^2)^{-1} \\ \mathbf{V}_{\tilde{\underline{\beta}}_0} &= (\mathbf{B}^{-1} + \mathbf{W}^{-1}/\tau^2 \sigma_{\beta}^2)^{-1}, & \tilde{\underline{\beta}}_0 &= \mathbf{V}_{\tilde{\underline{\beta}}_0}(\mathbf{B}^{-1} \underline{\mu}_0 + \mathbf{W}^{-1} \underline{\beta}/\tau^2 \sigma_{\beta}^2), \end{aligned}$$

and

$$\sigma_{\beta}^2|rest \sim IG((\alpha_{\beta} + S)/2, q_{\beta}/2 + (\underline{\beta} - \underline{\beta}_0)^T \mathbf{W}^{-1} (\underline{\beta} - \underline{\beta}_0)/2\tau^2), \quad (3.10)$$

$$\mathbf{W}^{-1}|rest \sim W\left(\rho \mathbf{V} + \frac{(\underline{\beta} - \underline{\beta}_0)(\underline{\beta} - \underline{\beta}_0)^T}{\tau^2 \sigma_{\beta}^2}, \rho + 1\right). \quad (3.11)$$

For sampling the hyperparameters of the kernel, we utilize a marginal posterior for the hyperparameters by integrating out the parameters $\underline{\beta}, \tau^2$ as shown in Berger et al. (2001) such that

$$p(d, g|\underline{\mathbf{y}}, \underline{\beta}_0, \mathbf{W}, \sigma_{\beta}^2) = \left(\frac{|\mathbf{V}_{\tilde{\underline{\beta}}}|}{(2\pi)^n |\mathbf{K}| |\mathbf{W}| \sigma_{\beta}^{2S}} \right)^{\frac{1}{2}} \frac{(q_{\tau}/2)^{\alpha_{\tau}/2}}{[(q_{\tau} + \psi)/2]^{(\alpha_{\tau} + N)/2}} \frac{\Gamma[(\alpha_{\tau} + N)/2]}{\Gamma[\alpha_{\tau}/2]} p(d, g), \quad (3.12)$$

where

$$\psi = \underline{\mathbf{y}}^T \mathbf{K}^{-1} \underline{\mathbf{y}} + \underline{\boldsymbol{\beta}}_0^T \mathbf{W}^{-1} \underline{\boldsymbol{\beta}}_0 / \sigma_\beta^2 - \underline{\tilde{\boldsymbol{\beta}}}^T \mathbf{V}_{\tilde{\boldsymbol{\beta}}}^{-1} \underline{\tilde{\boldsymbol{\beta}}}.$$

By integrating out $\underline{\boldsymbol{\beta}}$, we can also obtain a conditional distribution for τ^2 as

$$\tau^2 | \mathbf{K}, \underline{\boldsymbol{\beta}}_0, \mathbf{W} \sim IG \left((\alpha_\tau + N)/2, (q_\tau + \psi)/2 \right). \quad (3.13)$$

Finally, we develop our selection procedure on the posterior generated by applying an Ising prior to the likelihood such that

$$p(\tilde{\boldsymbol{\xi}} | \underline{\mathbf{y}}, \mathbf{K}, \tau^2) \propto \exp \left[-\frac{1}{2} \left\{ \sum \left(\tilde{\boldsymbol{\xi}} \tilde{\boldsymbol{\xi}}^T \odot \boldsymbol{\Omega} \right) + \frac{1}{\tau^2} \tilde{\mathbf{y}}^T \mathbf{K}^{-1} \tilde{\mathbf{y}} \right\} \right] \quad (3.14)$$

$$p(\tilde{\boldsymbol{\xi}}) \propto \exp \left[\frac{1}{2} \sum \left\{ \tilde{\boldsymbol{\xi}} \tilde{\boldsymbol{\xi}}^T \odot \boldsymbol{\Omega} \right\} \right],$$

where $\tilde{\mathbf{y}} = (\underline{\mathbf{y}} - \mathbf{Z}\underline{\boldsymbol{\beta}})$ and $\boldsymbol{\Omega}$ encodes a researcher's prior belief in the strength of the pairwise connections between the functional covariates at the individual nodes. The first term of equation (3.14) represents the neighbor/network effect, and the second term, which is equivalent to the external field in an Ising distribution, represents the strength of the evidence for a relationship between each element and the response variable. The expression represents the “energy” of the “system,” where a low energy system, one with “neighbors” taking on similar values, is one that is encouraged and therefore has a higher probability of occurrence.

We sample $\tilde{\boldsymbol{\xi}}$ using a Metropolis-Hastings (MH). Since the full conditional distribution of $\boldsymbol{\xi}$ is discrete, it is important to choose a proposal distribution carefully. The MCMC chain can often be stuck in a local mode due to the characteristics of the discrete full conditional distribution. Hence, we derive the variational Bayes (VB) proposal distribution, which is explained in detail in Section 3.4.1. By combining functional domain subspace selection with node and graph selection, the MH-based algorithm using the VB proposal distribution for

the Bayesian multi-tasking inference method is summarized in Algorithm 5.

Variational Bayes Proposal Distribution in MH

To increase the stability of our MCMC algorithm, we derive the proposal distribution rooted in variational methods. In particular, rather than utilizing a typical proposal defined over the $\{0, 1\}$ set, we first fit a variational proposal for our selection variables. Our approach involves assuming a likelihood over a summary measure of the relationship between $\underline{\mathbf{y}}$ and $\underline{\mathbf{X}}^{(P)}$, which we define as, $\mathcal{G}(\underline{\mathbf{y}}, \underline{\mathbf{X}}^{(P)}) : \mathbb{R}^{2N} \rightarrow \mathbb{R}$. In this study, we explore using Pearson correlation and mutual information as our function $\mathcal{G}(\cdot, \cdot)$. We denote observed values of $\mathcal{G}(\cdot, \cdot)$, which we denote by $\underline{\mathbf{g}}$. In what follows, we show the reasoning behind our choice of the variational distribution, $Q(\tilde{\xi}_p | \underline{\mathbf{g}})$. We use mean-field variational inference and make the assumption that our target distribution is fully factorizable such that

$$Q(\tilde{\xi} | \underline{\mathbf{g}}) = \prod_{p=1}^P Q_p(\tilde{\xi}_p | \underline{\mathbf{g}}) \quad (3.15)$$

and our objective can now be stated as follows

$$\min_Q KL(Q_{\tilde{\xi} | \underline{\mathbf{g}}} || P_{\tilde{\xi} | \underline{\mathbf{g}}}), \quad (3.16)$$

where KL refers to Kullback-Leibler divergence, and $P_{\tilde{\xi} | \underline{\mathbf{g}}}$ represents the posterior distribution of $\tilde{\xi}$ conditional on $\underline{\mathbf{g}}$.

Next we define $p(\underline{\mathbf{g}}, \tilde{\xi}) = p(\underline{\mathbf{g}} | \tilde{\xi}) \cdot p(\tilde{\xi})$ as

$$\mathbf{L}(\underline{\mathbf{g}} | \tilde{\xi}) \cdot \prod_{p=1}^P \left[\exp \left\{ \sum_{j < l} \tilde{\xi}_l \omega_{lj} \tilde{\xi}_j \right\} \right].$$

where $\mathbf{L}(\underline{\mathbf{g}} | \tilde{\xi})$ refers to the likelihood of $\underline{\mathbf{g}}$ for a given value of $\tilde{\xi}$.

Note that we do not separate the likelihood function into the product of its components

and include it within the product operator because the independence of the likelihood terms relies on the nature of our chosen association function $\mathfrak{G}(\cdot, \cdot)$. Some measures of association exhibit transitive properties, which would violate independence given that covariates are not independent, which we assume they are not (Tao, 2014). Assuming a Gaussian likelihood, expression (3.16) becomes

$$N(\underline{g}|\mu(\tilde{\underline{\xi}}), \sigma_g^2 \Sigma) \cdot \prod_{p=1}^P \left[\exp\left\{ \sum_{j < l} \tilde{\xi}_l \omega_{lj} \tilde{\xi}_j \right\} \right], \quad (3.17)$$

where we assume $\mu(\cdot)$ is a function such that $\mu : \mathbb{R} \rightarrow \mathbb{R}$ and for all l, j if $\tilde{\xi}_l = \tilde{\xi}_j$ then $\mu(\tilde{\xi}_l) = \mu(\tilde{\xi}_j)$. σ_g^2 is a scalar representing the variance of our proposed normal distribution. In the case where our observed similarity measures $\underline{g}_p$ are independent for all p , $\Sigma = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Otherwise, $\Sigma = \Omega_g$, where Ω_g is a matrix representing the degree of connectedness of the $\underline{g}_p$'s. We show below why we need to impose some sparsity restrictions on Ω_g , and in section 3.8, we show how we propose to implement those.

The expectation of $\tilde{\xi}_p$ at the $t + 1$ th iteration of VB therefore takes the following form:

$$E_{\tilde{\xi}_p}^{(t+1)} [\tilde{\xi}_p] = \tanh \left[\sum_{j \neq p} \omega_{pj} E_{-\tilde{\xi}_p}^{(t)} [\tilde{\xi}_j] + \frac{1}{2} E_{-\tilde{\xi}_p}^{(t)} \left\{ \log \left(\frac{N(\underline{g}|\mu(\tilde{\xi}_p = +1, -\tilde{\underline{\xi}}_p), \sigma_g^2 \Sigma)}{N(\underline{g}|\mu(\tilde{\xi}_p = -1, -\tilde{\underline{\xi}}_p), \sigma_g^2 \Sigma)} \right) \right\} \right] \quad (3.18)$$

The second term inside the parentheses requires taking an expectation over all possible combinations of the binary variables other than ξ_p which we denote by $-\tilde{\underline{\xi}}_p$. Computing the expectation is an operation of order $\mathcal{O}(2^{P-1})$ that can quickly become intractable for large P nodes (i.e. with large numbers of regions). To estimate the equation (3.18), we must assume sparsity in the off-diagonal terms of the Σ matrix, such that we only need to consider combinations of those $-\tilde{\underline{\xi}}_p$ that are strongly connected to $\tilde{\xi}_p$ when computing the expectation. Assuming M connections at most per node (region), the equation (3.18) reduces to a computation on the order of $\mathcal{O}(2^{M-1})$. In the special case assuming complete

independence, $\Sigma = \mathbf{I}$, the equation (3.18) becomes the following simplification

$$E_{\tilde{\xi}_p}^{(t+1)}[\tilde{\xi}_p] = \tanh \left[\sum_{j \neq p} \omega_{pj} E_{-\tilde{\xi}_p}^{(t)}[\tilde{\xi}_j] + \frac{1}{2} \log \left\{ \frac{N(\mathbf{g} | \mu(\tilde{\xi}_p = +1), \sigma_g^2)}{N(\mathbf{g} | \mu(\tilde{\xi}_p = -1), \sigma_g^2)} \right\} \right]. \quad (3.19)$$

which is tantamount to assuming that, conditional on the selection variable, ξ , any variability in the observed measure of similarity, $\mathbf{g}$, is attributable solely to noise. In other words, a connection between nodes (regions) only affects whether or not the nodes (regions) contain relevant information to the response, but not the amount of information that is drawn randomly.

The detailed derivation of the above results for the variational Bayes distribution is described in Appendix B.1.

3.5 Functional Association Testing

In this section, we present a framework for testing the overall functional association with the response variable. The proposed approach equates testing model significance with testing the null hypothesis $H_0 : \mathbf{h}(\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(P)}, \xi_1, \dots, \xi_P, t_p \in [a_p^*, b_p^*]) = 0$ which means that $H_0 : \tau^2 = 0$. This is equivalent to testing that once scalar covariates are accounted for and the variability of the error term is estimated; there is no variability left to warrant the inclusion of our functional covariates associated with the response variable. In other words, the functional covariates contribute little to explain the variability beyond the scalar covariates and random error. Clearly, testing the scale parameter $\tau^2 = 0$ lies on the boundary of the parameter space. As such, the method outlined here is a score-based test of the likelihood, making it computationally inexpensive as only the scalar covariates are required to be fit (and the error variance estimated). We still require the estimated node selection variables to construct our kernel matrix, which is a necessary part of the score-test statistic, as will be explained shortly.

In our estimation of the score test statistic, we apply a different approach towards the estimation of the necessary quantities; while in Section 3.4.1, we use a formulation that estimates the variance of the noise variables in the model, $\underline{\epsilon}$, as $\sigma^2 = \tau^2 g$, In this section we estimate the variance of $\underline{\epsilon}$ directly. The choice in the former allows for easier estimation procedures; however, this formulation precludes the estimation of σ^2 under the restricted likelihood, which assumes $\tau^2 = 0$ under the null. Therefore, in the following section, σ^2 is estimated via REML. We note that both formulations are equivalent (Gramacy, 2020). Hence, REML is a function of β , σ^2 , τ^2 , and d .

We begin by expressing the restricted log-likelihood of our model

$$\begin{aligned} \ell(\beta, \sigma^2, \tau^2, d) \propto & -\frac{1}{2} \log \left| \left(\tau^2 \mathbf{K} + \sigma^2 \mathbf{I} \right) \right| - \frac{1}{2} (\underline{\mathbf{y}} - \mathbf{Z}\underline{\beta})^T (\tau^2 \mathbf{K} + \sigma^2 \mathbf{I})^{-1} (\underline{\mathbf{y}} - \mathbf{Z}\underline{\beta}) \\ & - \frac{1}{2} \log \left| \mathbf{Z}^T (\tau^2 \mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{Z} \right|, \end{aligned} \quad (3.20)$$

which is obtained by treating $\underline{\mathbf{h}}$ as a random effect in the mixed effects model.

The score equations for τ^2, σ^2 , and d given $\hat{\beta}$ are then

$$\begin{aligned} \tau^2 : & -\frac{1}{2} \text{tr}(\mathbf{K}\mathbf{P}) + \frac{1}{2} (\underline{\mathbf{y}} - \mathbf{Z}\hat{\beta})^T (\tau^2 \mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{K} (\tau^2 \mathbf{K} + \sigma^2 \mathbf{I})^{-1} (\underline{\mathbf{y}} - \mathbf{Z}\hat{\beta}) = 0; \\ \sigma^2 : & -\frac{1}{2} \text{tr}(\mathbf{P}) + \frac{1}{2} (\underline{\mathbf{y}} - \mathbf{Z}\hat{\beta})^T (\tau^2 \mathbf{K} + \sigma^2 \mathbf{I})^{-1} (\tau^2 \mathbf{K} + \sigma^2 \mathbf{I})^{-1} (\underline{\mathbf{y}} - \mathbf{Z}\hat{\beta}) = 0; \\ d : & -\frac{1}{2} \text{tr} \left(\tau^2 \frac{\partial \mathbf{K}}{\partial d} \mathbf{P} \right) + \frac{1}{2} (\underline{\mathbf{y}} - \mathbf{Z}\hat{\beta})^T (\tau^2 \mathbf{K} + \sigma^2 \mathbf{I})^{-1} \left(\tau^2 \frac{\partial \mathbf{K}}{\partial d} \right) (\tau^2 \mathbf{K} + \sigma^2 \mathbf{I})^{-1} (\underline{\mathbf{y}} - \mathbf{Z}\hat{\beta}) = 0, \end{aligned} \quad (3.21)$$

where $\mathbf{V} = (\tau^2 \mathbf{K} + \sigma^2 \mathbf{I})$, $\hat{\beta} = (\mathbf{X}^T \mathbf{V}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{V}^{-1} \underline{\mathbf{y}}$, and $\mathbf{P} = \mathbf{V}^{-1} - \mathbf{V}^{-1} \mathbf{Z} (\mathbf{Z}^T \mathbf{V}^{-1} \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{V}^{-1}$.

To obtain the score statistic for testing $H_0 : \tau^2 = 0$, we simply evaluate equation (3.21) under the null to obtain the following test statistic

$$-\frac{1}{2} \text{tr}(\mathbf{K}\mathbf{P}_0) + \frac{1}{2\sigma^2} (\underline{\mathbf{y}} - \mathbf{Z}\hat{\beta})^T \mathbf{K} (\underline{\mathbf{y}} - \mathbf{Z}\hat{\beta}) \quad (3.22)$$

where $\mathbf{P}_0 = \mathbf{I} - \mathbf{Z}(\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T$.

This score statistic follows a mixture of chi-square distributions, with the weights being the ordered non-zero eigenvalues of $\mathbf{K}\mathbf{P}_0/2$ as described in Zhang and Lin (2003). The p-value using this score statistic is calculated using the algorithm presented in Davies (1980).

3.6 Algorithm for Bayesian multitasking inference

Algorithm 3 Algorithm for Bayesian multi-tasking inference method

- 1: Initialize model parameters $\underline{\xi}^{(0)}$
 - 2: Set prior hyper-parameters for $\rho, \mathbf{V}, \underline{\mu}_0, \mathbf{B}_0, \alpha_\tau, q_\tau, \alpha_\beta, q_\beta, \alpha_g, q_g, \alpha_{d_1}, q_{d_1}, \alpha_{d_2}, q_{d_2}, \Omega$
 - 3: **Task 1: Domain Selection**
 - 4: **for** each p in P nodes **do**:
 - 5: Randomly sample a set of m initial design points for a_p, s_p
 - 6: Construct a surrogate model for $RMSE_{CV}$ on a_p, s_p
 - 7: **repeat**:
 - 8: Compute Expected Information to acquire a new design point
 - 9: **until** convergence or maximum number of iterations is reached
 - 10: Truncate $\mathbf{X}^{(p)}$ from $[1, T]$ to $[a_p^*, a_p^* + s_p^*]$
 - 11: **end for**
 - 12: **Task 2: Node Selection**
 - 13: Truncate each $\mathbf{X}^{(p)}$ from $[1, T]$ to $[a_p^*, a_p^* + s_p^*]$
 - 14: Sample $\underline{\xi}$ in MH-sampler using (13)
 - 15: Retrieve posterior mean value for $\underline{\xi}$ after sufficient burn-in
 - 16: **Task 3: Model Significance Testing**
 - 17: Using selected nodes, $\underline{\xi}^*$, and selected functional domains, $[a_p^*, b_p^*]$, we construct the covariance matrix, $\mathbf{K}$
 - 18: Estimate $\underline{\hat{\beta}}$ via maximum-likelihood conditional on $\mathbf{K}$
 - 19: Compute the test statistic $\mathcal{T} = -\frac{1}{2}\text{tr}(\mathbf{K}\mathbf{P}_0) + \frac{1}{2\sigma^2}(\underline{\mathbf{y}} - \mathbf{Z}\underline{\hat{\beta}})^T \mathbf{K}(\underline{\mathbf{y}} - \mathbf{Z}\underline{\hat{\beta}})$ and calculate p-value using the mixture Chi-squared distribution
-

We note that the choice of uniform random sampling of initial design points a_p, s_p was shown to provide superior performance to other alternatives in special cases, including the

optimization of blackbox functions, which we are performing here (Boya Zhang and Gramacy, 2021). $RMSE_{CV}$ represents the root mean square error arising from applying a generic, simple model to various subsets of the functional covariate domain in a cross-validation scheme. a_p and s_p represent the starting point in our chosen functional covariate domain, and s_p represents the length of the domain subset.

3.7 Theoretical Properties

We provide the asymptotic properties of our method in terms of consistency and sparsity for variable selection on $\boldsymbol{\xi}$ under an independent Bernoulli prior of $\boldsymbol{\xi}$, $p(\boldsymbol{\xi}) \propto \exp(\mathbf{q}'\boldsymbol{\xi})$, $\mathbf{q}'$ is prior belief, which is a special case of Ising prior.

The incoherence condition for sparsity recovery is also described. We note that Fang et al. (2016) originally showed asymptotic properties of the kernel machine model using high-dimensional scalar variables when the selection parameters are nonnegative cases. However, our study's kernel model is built using functional variables, and our selection parameters ($\boldsymbol{\xi}$) are binary.

In our study, we introduce an nonnegative activation function $\xi_\epsilon(s)$, which lies in $[0,1]$, is an increasing function, and satisfies the following properties

$$\begin{aligned}\xi_\epsilon(s) &\rightarrow \chi_{[0,\infty)} \text{ as } \epsilon \rightarrow 0, \\ \frac{d\xi_\epsilon(s)}{ds} &\rightarrow \delta(0) \text{ as } \epsilon \rightarrow 0,\end{aligned}$$

where $\chi_{[0,\infty)}$ and $\delta(0)$ represent a characteristic function and Dirac-delta, respectively. That is,

$$\lim_{\epsilon \rightarrow 0} \xi_\epsilon(s) = \begin{cases} 1 & \text{if } s \in [0, \infty) \\ 0 & \text{otherwise} \end{cases}$$

We show that such an activation function exists in Lemma 1. Let $\boldsymbol{\xi}_\epsilon(s)$ be activation functions

corresponding to $\boldsymbol{\xi}$. We simply denote $\boldsymbol{\xi}_\epsilon(s) \equiv \boldsymbol{\xi}_\epsilon$, where $\boldsymbol{\xi}_\epsilon = (\xi_{1\epsilon}, \dots, \xi_{P\epsilon})$ and $\xi_{l\epsilon}$ is an activation function of ξ_l , $l = 1, \dots, P$.

The log of the full conditional distribution of $\boldsymbol{\xi}_\epsilon$ is equivalently written as the regularized penalized objective function with penalty parameters λ_0 and λ

$$\frac{1}{2} \|\tilde{\mathbf{y}} - \mathbf{K}(\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(P)}, \xi_{1\epsilon}, \dots, \xi_{P\epsilon}) \underline{\boldsymbol{\alpha}}\|^2 + \frac{1}{2} \lambda_0 \underline{\boldsymbol{\alpha}}^T \mathbf{K}(\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(P)}, \xi_{1\epsilon}, \dots, \xi_{P\epsilon}) \underline{\boldsymbol{\alpha}} + n\lambda \sum_l^P \xi_{l\epsilon}, \quad (3.23)$$

which is the same as the objective function of Fang et al. (2016).

Hence, theoretical properties using $\xi_\epsilon(s)$ can be similarly derived by following Fang et al. (2016)'s results. We consider n and P to be infinity under some assumptions. We first define some notations.

3.7.1 Notations

Let $\boldsymbol{\xi}^*$ and $\hat{\boldsymbol{\xi}}$ represents the true ξ_p and estimator of the functional variable, respectively. Suppose vector $\boldsymbol{\xi}^*$ is sparse. Without loss of generality, denote $\boldsymbol{\xi}^* = (\xi_{1_p}^*, \dots, \xi_{l_p}^*)^T = (\boldsymbol{\xi}_1^{*T}, \boldsymbol{\xi}_0^{*T})$, where $\boldsymbol{\xi}_1^*$ is the vector of the first ω^* ones of $\boldsymbol{\xi}$. Denote the one index set of $\boldsymbol{\xi}$ as $\mathcal{B} = \{l \in \{1, \dots, P\} | \xi_l^* = 1\}$ and $\hat{\boldsymbol{\xi}}$ as $\hat{\mathcal{B}} = \{l \in \{1, \dots, P\} | \hat{\xi}_l = 1\}$. Note that $\mathcal{B}$ has also relatively small cardinality $\omega^* = |\mathcal{B}|$, the number of true one ξ_l 's.

Let $\boldsymbol{\xi}_\epsilon^*(s)$ and $\hat{\boldsymbol{\xi}}_\epsilon(s)$ be activation functions corresponding to $\boldsymbol{\xi}^*$ and $\hat{\boldsymbol{\xi}}$. We simply denote $\boldsymbol{\xi}_\epsilon^*(s) \equiv \boldsymbol{\xi}_\epsilon^*$ and $\hat{\boldsymbol{\xi}}_\epsilon(s) \equiv \hat{\boldsymbol{\xi}}_\epsilon$. Also let us denote $\frac{d\xi_\epsilon(s)}{ds} \equiv \delta_\epsilon$. We also simply denote $\mathbf{K}(\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(P)}, \xi_{1\epsilon}, \dots, \xi_{P\epsilon}) \equiv \mathbf{K}(\boldsymbol{\xi}_\epsilon)$.

Define the true $\underline{\boldsymbol{\alpha}}$ vector as $\underline{\boldsymbol{\alpha}}^*$. For a given $\underline{\boldsymbol{\alpha}}^*$ and estimate $\tilde{\underline{\boldsymbol{\alpha}}}$, we define the following

matrices and their respective partitions for functional variables. We can define partitions as

$$\begin{aligned}\tilde{Z} &= [\tilde{Z}_1, \dots, \tilde{Z}_P] = [\tilde{\mathbf{Z}}_1, \tilde{\mathbf{Z}}_0] \\ &= [\{\mathbf{K}'(\xi_{l\epsilon}^*)\underline{\alpha}^*\}_{1 \leq l \leq \omega^*}, \{\mathbf{K}'(\xi_{l\epsilon}^*)\underline{\alpha}^*\}_{\omega^*+1 \leq l \leq P}]; \\ \tilde{\tilde{Z}} &= [\tilde{\tilde{Z}}_1, \dots, \tilde{\tilde{Z}}_P] = [\tilde{\tilde{\mathbf{Z}}}_1, \tilde{\tilde{\mathbf{Z}}}_0] \\ &= [\{\mathbf{K}(\xi_{l\epsilon}^*)\underline{\alpha}^*\}_{1 \leq l \leq \omega^*}, \{\mathbf{K}'(\xi_{l\epsilon}^*)\underline{\alpha}^*\}_{\omega^*+1 \leq l \leq P}],\end{aligned}$$

where $\mathbf{K}'_l(\xi_{l\epsilon}^*) = (\partial \mathbf{K} / \partial \xi_{l\epsilon}) \delta_\epsilon$, obtained by taking the partial derivative of the componentwise entries of $\mathbf{K}$.

Some covariance matrices are also defined as

$$\begin{aligned}\tilde{\Sigma}_{11} &= (n^{-1} \tilde{\mathbf{Z}}_1^T \tilde{\mathbf{Z}}_1), \quad \tilde{\Sigma}_{01} = (\mathbf{n}^{-1} \tilde{\mathbf{Z}}_0^T \tilde{\mathbf{Z}}_1) \\ \tilde{\tilde{\Sigma}}_{11} &= (n^{-1} \tilde{\tilde{\mathbf{Z}}}_1^T \tilde{\tilde{\mathbf{Z}}}_1), \quad \tilde{\tilde{\Sigma}}_{01} = (\mathbf{n}^{-1} \tilde{\tilde{\mathbf{Z}}}_0^T \tilde{\tilde{\mathbf{Z}}}_1),\end{aligned}$$

where $\tilde{\Sigma}_{11}$ and $\tilde{\tilde{\Sigma}}_{11}$ are also assumed to be invertible.

3.7.2 Incoherence Condition for Recovery of Sparsity

The incoherence condition for recovery of sparsity is

$$\tilde{\Sigma}_{01} \tilde{\Sigma}_{11}^{-1} \mathbf{1} - \frac{\sigma^2 \sigma_\xi^2}{2n \sigma_\alpha^2} \tilde{\mathbf{Z}}_0^T \tilde{P} \underline{\alpha}^* \preceq (1 - \delta) \mathbf{1},$$

where $\delta \in (0, 1]$ and $\tilde{P} = [I - \tilde{Z}_1(\tilde{Z}_1^T \tilde{Z}_1)^{-1} \tilde{Z}_1^T]$ is a projection matrix. It is equivalent to $\tilde{\tilde{\Sigma}}_{01} \tilde{\tilde{\Sigma}}_{11}^{-1} \mathbf{1} - \frac{\lambda_0}{2n\lambda} \tilde{\tilde{Z}}_0^T \tilde{\tilde{P}} \underline{\alpha}^* \preceq (1 - \delta) \mathbf{1}$, $\tilde{\tilde{P}} = [I - \tilde{\tilde{Z}}_1(\tilde{\tilde{Z}}_1^T \tilde{\tilde{Z}}_1)^{-1} \tilde{\tilde{Z}}_1^T]$.

We consider δ -consistent $\tilde{\alpha}$ and $\tilde{Z}$ matrix only. We can recover the sparsity of $\boldsymbol{\xi}^*$ based on $\tilde{\alpha}$ under this incoherence condition. The sparsity of $\boldsymbol{\xi}^*$ is recovered through the techniques of a primal-dual witness on model selection consistency (Ravikumar et al., 2009, 2010). The derivation of this condition can be obtained by following Fang et al. (2016).

3.7.3 Regularity conditions

Define the objective function $Q(\boldsymbol{\xi}_\epsilon) = Q_0(\boldsymbol{\xi}_\epsilon) + \lambda pen(\boldsymbol{\xi}_\epsilon) = \frac{\lambda_0}{2} y^T (\lambda_0 I + \tau^2 \mathbf{K}(\boldsymbol{\xi}_\epsilon))^{-1} + n\lambda pen(\boldsymbol{\xi}_\epsilon)$, where $pen(\boldsymbol{\xi}_\epsilon)$ is a penalty function of $\boldsymbol{\xi}$.

Let $\Delta = \lambda_0 I + \tau^2 K(\boldsymbol{\xi}_\epsilon)$, and we define the $P \times 1$ vector $\mathbf{v}_n(\boldsymbol{\xi}_\epsilon)$ and $P \times P$ matrix $M_n(\boldsymbol{\xi}_\epsilon)$ as follows:

$$v_n(\boldsymbol{\xi}_\epsilon) = \lambda_0^{-1} n^{-1/2} \frac{\partial Q_0(\boldsymbol{\xi}_\epsilon)}{\partial \boldsymbol{\xi}_\epsilon} = - \left\{ \frac{1}{\sqrt{n}} \mathbf{y}^T \Delta^{-1} \left(\frac{\partial \mathbf{K}}{\partial \xi_{j\epsilon}} \delta_\epsilon \right) \Delta^{-1} \mathbf{y} \right\}_{1 \leq j \leq P}^T$$

$$M_n(\boldsymbol{\xi}_\epsilon) = (\lambda_0 n)^{-1} \frac{\partial^2 Q_0(\boldsymbol{\xi}_\epsilon)}{\partial \boldsymbol{\xi}_\epsilon^T \partial \boldsymbol{\xi}_\epsilon} = \left\{ \frac{1}{2n} y^T \Delta^{-1} \left[\frac{\partial K}{\partial \xi_{i\epsilon}} \delta_\epsilon \Delta \frac{\partial \mathbf{K}}{\partial \xi_{j\epsilon}} \delta_\epsilon + \frac{\partial \mathbf{K}}{\partial \xi_j} \delta_\epsilon \Delta \frac{\partial \mathbf{K}}{\partial \xi_{i\epsilon}} \delta_\epsilon - \frac{\partial^2 \mathbf{K}}{\partial \xi_{i\epsilon} \partial \xi_{j\epsilon}} \delta_\epsilon - \frac{\partial \mathbf{K}}{\partial \xi_{i\epsilon}} \frac{\partial \delta_\epsilon}{\partial t} \right] \Delta^{-1} y \right\}_{1 \leq i, j \leq P}$$

With these notations, we have the following regularity conditions

$$\lim_{n \rightarrow \infty} \mathbf{v}_n(\boldsymbol{\xi}_\epsilon) \rightarrow v^*, \quad \|\mathbf{v}^*\|_\infty < \infty$$

$$\lim_{n \rightarrow \infty} M_n(\boldsymbol{\xi}_\epsilon) \rightarrow M^*, \quad \|M^*\|_\infty < \infty$$

3.7.4 Asymptotic Results

Lemma 1 (The existence of nonnegative activation function) There exists a non-negative activation function $\xi_\epsilon(s)$ which lies on $[0,1]$ and is an increasing function.

Proof

There exists $\eta(u) \geq 0$ such that $\sup_u \eta(u) \in (-1, 1)$ and $\int_{-\infty}^{\infty} \eta(u) du = \int_{-1}^1 \eta(u) du = 1$. Let $\eta_\epsilon(u) = \frac{1}{\epsilon} \eta(u/\epsilon)$. Then $\sup_u \eta_\epsilon(u) \in (-\epsilon, \epsilon)$ and $\eta_\epsilon(u) \geq 0$.

$$\int_{-\infty}^{\infty} \eta_\epsilon(u) du = \int_{-\epsilon}^{\epsilon} \eta_\epsilon(u) du = \int_{-\epsilon}^{\epsilon} \frac{1}{\epsilon} \eta\left(\frac{u}{\epsilon}\right) du = \int_{-1}^1 \eta(v) dv = 1$$

Let $\xi_\epsilon(s) = \int_{-\infty}^s \eta_\epsilon(u) du$. If $s < -\epsilon$, then $\xi_\epsilon(s) = 0$. If $s > \epsilon$, then $\xi_\epsilon(s) = 1$. Hence, $\xi_\epsilon(s) \geq 0$.

Theorem 1 (Consistency of $\boldsymbol{\xi}_\epsilon$): Under the regularity conditions, if $n\lambda\lambda_0 a = O_p(1)$ with some positive constant a less than n , there exists a local minimum activation function $\boldsymbol{\xi}_\epsilon$ of $Q(\boldsymbol{\xi}_\epsilon)$ such that $\|\hat{\boldsymbol{\xi}}_\epsilon - \boldsymbol{\xi}_\epsilon^*\| = O_p(\sqrt{\frac{\log(P)}{n}})$.

Theorem 2 (Recovery of Sparsity): Under the following conditions

- (c1) the initial estimate $\tilde{\underline{\alpha}}_p$ is δ consistent, i.e, $\|\tilde{\underline{\alpha}}_p - \underline{\alpha}_p^*\|_\infty = O_p(\delta)$ for some $\delta \rightarrow 0$,
 - (c2) $\Lambda_{\min}(\tilde{\Sigma}_{11}) = \tilde{C}_{\min} > 0$, where $\Lambda_{\min}(\cdot)$ denotes the minimum nonnegative eigenvalue of M_n ,
 - (c3) $\tilde{\Sigma}_{01}\tilde{\Sigma}_{11}^{-1} \rightarrow \Sigma_{01}\Sigma_{11}^{-1}$ with a rate no slower than δ , and
 - (c4) incoherence condition $\Sigma_{01}\Sigma_{11}^{-1}\mathbf{1}_{\frac{\lambda_0}{2n\lambda}}Z_0^TP\boldsymbol{\alpha}^* \preceq (1-\gamma)\mathbf{1}$, there exists some λ with $n\lambda^2 \rightarrow \infty$ such that for some constant $\eta_1 > 0$, with probability $1 - \exp(-\eta_1 n\lambda^2) \rightarrow 1$ we have results (a)-(b):
- (a) $\hat{\mathcal{B}} \subseteq \mathcal{B}$ and the upper bound of $\|\hat{\boldsymbol{\xi}}_{1\epsilon} - \boldsymbol{\xi}_{1\epsilon}^*\|$ converges to

$$\rho(\lambda) = \lambda \left[\frac{4\sigma}{\sqrt{\tilde{C}_{\min}}} + \|\tilde{\Sigma}_{11}^{-1}\|_\infty \frac{\lambda_0}{\lambda} (n^{-1/2} \|\mathbf{v}^*\|_\infty + O_p(\delta)) + \|\tilde{\Sigma}_{11}^{-1}\|_\infty \right]$$

- (b) If $\rho(\lambda) < \min_{j \in \mathcal{B}} \xi_j^*$, then we have sparsistency of $\hat{\boldsymbol{\xi}}$, i.e. $\hat{\mathcal{B}} = (\mathcal{B})$.

Because of the equivalence (3.23), the proofs of these theorems can be obtained by following Fang et al. (2016) by replacing $(\frac{\partial K}{\partial \xi_j})$ by $(\frac{\partial K}{\partial \xi_{j\epsilon}})\delta_\epsilon$. The detailed proofs are omitted.

We also have the following property due to the Ising model:

Property 1 The P -dimensional binary random variable $\boldsymbol{\xi}$ of semiparametric joint functional kernel graphical model (3.1) can be viewed as a class of stochastic processes on a finite random undirected graph model $G = (V, E)$, where $V = \{1, \dots, P\}$ is the set of nodes, corresponding to l th predictors, and $E \subset V \times V$ is the set of edges. $\boldsymbol{\xi} \in \Gamma = \{(\xi_1, \dots, \xi_P) : \xi_l, l = 1, \dots, P\}$ is indexed by V with the probability measure on Γ on as (3.14), when w_{ij} and q_j are all random with some distributions.

3.8 Simulation

We conduct a simulation study to evaluate the performance of our BMIM and to compare it against the following five existing methods that qualify as suitable approaches for the motivating problem:

- Lasso: the functional covariates are treated as scalar variables concatenated to a single matrix $X_{N \times (TP)}$ and then fit the linear model by penalizing the log-likelihood during the optimization process using an L_1 penalty,

$$\hat{\underline{\beta}} = \underset{\underline{\beta}}{\operatorname{argmax}} L(\underline{\beta}|\underline{\mathbf{y}}, \mathbf{X}) - \lambda \sum_{p=1}^P \sum_{t=1}^T |\beta_p(t)|$$

- Fused Lasso: similar to the lasso, fit the linear model by penalizing the log-likelihood using the fused penalty,

$$\hat{\underline{\beta}} = \underset{\underline{\beta}}{\operatorname{argmax}} L(\underline{\beta}|\underline{\mathbf{y}}, \mathbf{X}) - \lambda_1 \sum_{p=1}^P \sum_{t=1}^T |\beta_p(t)| - \lambda_2 \sum_{p=1}^P \sum_{t=1}^T |\beta_p(t+1) - \beta_p(t)|$$

- GP-on-FPCA: Gaussian process regression built using functional principal component analysis (FPCA) scores: First, we apply functional PCA to decompose functional variables into a set of basis functions (often orthogonal), which capture most of the variability in the functional data. By retaining a subset of these basis functions, one can approximate the original functions using a finite number of coefficients, effectively reducing the dimensionality of the data while retaining most of its variability. The number of principal components (K) used is based on the *fraction of variance explained* criterion such that principal components are included until 90% of variability in our functional data is explained by the chosen principal components. And then we consider

the following GPR

$$\underline{\mathbf{y}} = f(\mathcal{X}) + \underline{\epsilon};$$

$$\mathcal{X} \xrightarrow{\text{FPCA}} \Psi;$$

$$f(\mathcal{X}) \sim \mathcal{GP}(\underline{\mathbf{0}}, \tau^2 \mathbf{K}) \quad ; \quad \underline{\epsilon} \sim N(\underline{\mathbf{0}}, \sigma^2 \mathbf{I});$$

$$k(\underline{\Psi}_i, \underline{\Psi}_j) = \exp \left[- \frac{\sum_{p=1}^P \sum_{k=1}^K (\Psi_{p,k}^i - \Psi_{p,k}^j)^2}{d_{p,k}} \right] + g\delta_{ij},$$

where K is the number of functional principal components, $d_{p,k}$ is the lengthscale parameter for the k^{th} principal component score arising from the p^{th} node, g is the noise parameter (nugget), and $\underline{\Psi}_i$ is the vector of principal component scores for observation i , and $\underline{\Psi}_i = \{\Psi_{p,k}^i; p = 1 \dots P, k = 1, \dots, K\}$. Hyperparameters for this GPR fit by maximizing log-likelihood.

- Scalar-on-functional linear model (FLM) using B-splines: As the coefficient function is defined over the domain of the functional covariates, we can obtain coefficients for each point on the function's domain. The functional smoothing parameter is estimated via generalized cross-validation (Ramsay and Silverman, 2002). The model formulation is

$$\begin{aligned} y_i &= \sum_{p=1}^P \int \beta_p(t) x_{ip}(t) dt + \epsilon_i \\ &\approx \sum_{p=1}^P \sum_{b=1}^B c_{p,b} \int B_{p,b}(t) x_{ip}(t) dt + \epsilon_i, \end{aligned}$$

where $\beta_p(t)$ is the functional coefficient, $x_{ip}(t)$ is the p th functional variable of i th sample, B is the number of BSpline bases, $B_{p,b}(t)$ is the BSpline basis function for the p^{th} node evaluated at timepoint t , and $c_{p,b}$ is the coefficient for the b^{th} BSpline in the p^{th} node.

In our simulation study, the (P, N) setting is chosen to match our motivating dataset, the Autism Brain Imaging Data Exchange dataset. We then consider two generating models with two different values of P . For each combination setting, we evaluate the performance

of the presented methods on all domain selection, node selection, and model significance using four evaluation metrics: the true positive rate (TPR), the false positive rate (FPR), the positive predictive value (PPV), and the F1 score, respectively defined as follows,

$$\mathbf{TPR} = \frac{\sum_{p=1}^P \mathbb{I}(\hat{\xi}_p = 1, \xi_p = 1)}{\sum_{p=1}^P \mathbb{I}(\xi_p = 1)} \quad ; \quad \mathbf{FPR} = \frac{\sum_{p=1}^P \mathbb{I}(\hat{\xi}_p = 1, \xi_p = 0)}{\sum_{p=1}^P \mathbb{I}(\xi_p = 0)}.$$

$$\mathbf{PPV} = \frac{\sum_{p=1}^P \mathbb{I}(\hat{\xi}_p = 1, \xi_p = 1)}{\sum_{p=1}^P \mathbb{I}(\hat{\xi}_p = 1)} \quad ; \quad \mathbf{F}_1 = \frac{2PPV \times TPR}{PPV + TPR}$$

3.8.1 Setting

We consider $P \in \{10, 30\}$, with $P^* = 3$, the true number of relevant variables contributing to the response, which is on the same order of magnitude as our motivating dataset. P is the number of functional covariates, with each functional covariate corresponding to a node location. Within each functional covariate, we generate $T = 100$ timepoints for a total of 1000 covariates. For each simulation iteration, we generate $N = 341$ samples.

Functional variables

To generate our functional covariates, we begin by randomly 3 indices from $1, 2, \dots, P$ without replacement. These indices mark nodes to be set as relevant to the response. We generate our functional covariates each over 100 time points. The relevant time points always be set between $t = 61$ to $t = 80$. We then sample an observation from a Bernoulli distribution, $Bern(0.5)$, such that if our sampled value is 0, we generate a single functional covariate prototype per node, while if our sampled value is 1, we generate two functional covariate prototypes only for the relevant domains of the three chosen nodes, while the remaining nodes - as well as irrelevant domains within the chosen nodes - only hold a single functional prototype.

For our functional covariate prototypes, $X_1^{(P)}$ and $X_2^{(p)}$, we generate each from a modified

Weiner process such that

$$\underline{\mathbf{x}}_{1,2}(0) \sim MVN(\underline{\mathbf{0}}, \mathbf{C}_P) \quad , \quad \underline{\mathbf{x}}_{1,2}(t) \sim MVN(\underline{\mathbf{x}}_{1,2}(t-1), \mathbf{C}_P) \quad (3.24)$$

where $\underline{\mathbf{x}}(t)$ is a vector of covariate observations across all nodes P at a single time point t and $\mathbf{C}_P$ is a $T \times T$ matrix representing the desired connectedness between the nodes such that the $(t, t')^{th}$ element of $\mathbf{C}_P(t, t')$ is

$$c_p(t, t') = \tau_u \exp \left[- \frac{||t - t'||}{\rho_u} \right],$$

where τ_u is a tuning hyperparameter responsible for the overall scale of the generated data; likewise, ρ_u is a lengthscale parameter that controls the degree of locality of connection between time points. We use $\mathbf{C}_P$ to induce correlation between functional covariates which are to be considered neighbors.

Finally, to generate $X_i^{(P)}$, we again sample from $Bern(0.5)$. Each observation is assigned one of the functional prototypes based on its sampled Bernoulli value plus a vector of independent Gaussian noise, such that if we desire to assign an individual, i , to the first functional prototype, we have

$$\mathbf{X}_i^{(P)} = \mathbf{X}_1^{(P)} + \underline{\mathbf{u}} \quad , \quad \underline{\mathbf{u}} \sim MVN(\underline{\mathbf{0}}, \sigma_u^2 \mathbf{I}); \quad i = 3, \dots, N \quad (3.25)$$

where σ_u^2 is a tuning parameter controlling the deviation of an individual's functional covariate from its underlying functional prototype.

Two generating models

With respect to our response variables ($\mathbf{y}$), we adopt two different cases of generating models: functional linear and nonparametric models, with $N = 341$, $T = 100$, $P^* = 3$, and $P \in$

$\{10, 30\}$:

- Case 1: Functional linear model

$$\underline{\mathbf{y}} = \sum_{p \in \mathcal{C}} \int_1^T \underline{\mathbf{x}}^{(p)}(t) \beta_p(t) dt + \underline{\epsilon}, \quad \underline{\epsilon} \sim MVN(\underline{\mathbf{0}}, \sigma^2 \mathbf{I}), \quad (3.26)$$

where

$$\begin{aligned} \beta_1(t) &= 0.1 \cdot N(t | \mu = 0.675, \sigma = 0.03) \cdot \mathbb{I}\{t \in [61, 80]\}, \\ \beta_2(t) &= -0.083 \cdot N(t | \mu = 0.7, \sigma = 0.05) \cdot \mathbb{I}\{t \in [61, 80]\}, \\ \beta_3(t) &= -0.133 \cdot N(t | \mu = 0.725, \sigma = 0.03) \cdot \mathbb{I}\{t \in [61, 80]\}. \end{aligned}$$

Here, $\mathcal{C}$ denotes the chosen relevant locations and $\mathbb{I}$ denotes the indicator function. The coefficient functions were chosen to induce a similar distribution of $\underline{\mathbf{y}}$ to the motivating dataset.

- Case 2: Functional nonparametric model

$$\underline{\mathbf{y}} = \mathbf{f}(\mathbf{X}^{(3)}) + \underline{\epsilon}, \quad (3.27)$$

where $\mathbf{X}^{(3)} = [\mathbf{X}_1^{(1)}, \dots, \mathbf{X}_N^{(3)}]$ to be the functional covariates generated at the 3 regions for all N individuals across the time interval.

$$\mathbf{f}(\mathbf{X}^{(3)}) = \begin{cases} \mu_1 \cdot \mathbb{I}\{\mathbf{K}(\mathbf{X}^{(3)})\underline{\alpha} \geq 0, t \in [61, 80]\} \\ \mu_2 \cdot \mathbb{I}\{\mathbf{K}(\mathbf{X}^{(3)})\underline{\alpha} < 0, t \in [61, 80]\} \end{cases}$$

and

$$\underline{\epsilon} \sim N(\underline{\mathbf{0}}, \Sigma(\mathbf{X}^{(p)})),$$

where μ_1 and μ_2 are constants, estimated from the motivating dataset such that:

$$\mu_1 = \frac{\sum_{i=1}^N y_i \mathbb{I}\{Label_i = 0\}}{\mathbb{I}\{Label_i = 0\}} \quad ; \quad \mu_2 = \frac{\sum_{i=1}^N y_i \mathbb{I}\{Label_i = 1\}}{\mathbb{I}\{Label_i = 1\}},$$

where $Label_i = 1$ represents a subject diagnosed with ASD; $Label_i = 0$ represents a subject diagnosed with no ASD and

$$\Sigma(\mathbf{X}^{(3)}) = \begin{cases} \sigma_1^2 \cdot \mathbb{I}\{\mathbf{K}(\mathbf{X}^{(3)})\underline{\alpha} \geq 0\} \\ \sigma_2^2 \cdot \mathbb{I}\{\mathbf{K}(\mathbf{X}^{(3)})\underline{\alpha} < 0\}, \end{cases}$$

where

$$\sigma_1^2 = \frac{\sum_{i=1}^N (y_i - \mu_1)^2 \mathbb{I}\{Label_i = 0\}}{\mathbb{I}\{Label_i = 0\}} \quad ; \quad \sigma_2^2 = \frac{\sum_{i=1}^N (y_i - \mu_2)^2 \mathbb{I}\{Label_i = 1\}}{\mathbb{I}\{Label_i = 1\}}.$$

We consider Case 2 to generate simulated data similar to real data of interest and does not require imposing any assumptions on a coefficient function.

3.8.2 Simulation Results

We compare our method (BMIM)'s performance to five other methods: Lasso, Fused Lasso, GP-on-FPCA, and FLM-Bsplines. We note that while we can recover a relevant domain from most of these methods, that is out of reach when applying FPCA to our functional covariates. We summarize simulation results in Tables 3.1-3.8 and in Figures 3.1-3.6.

First, under Case (1), we compare our two methods: one is McMC-on- K_p^1 (BMIM-1) which conducts MCMC for node selection and kernel machine regression with K_p^1 for domain selection, described in Section 3.3 and the other is McMC-on- K_p^2 approach (BMIM-2) which conduct MCMC for node selection and kernel machine regression with K_p^2 for domain selection, we also described in Section 3.3. We observe that BMIM-2 significantly

outperforms the BMIM-1, as shown in Table 3.1.

		Domain Selection		No Domain Selection	
		BMIM-1	BMIM-2	BMIM-1	BMIM-2
Nodes	TPR	0.828	0.934	0.839	0.942
	FPR	0.087	0.078	0.078	0.062
	PPV	0.803	0.836	0.821	0.866
	F₁	0.815	0.883	0.830	0.902

Table 3.1: Comparison of different BMIM node selection settings under Case 1: BMIM-1: McMC-on- K_p^1 ; BMIM-2: McMC-on- K_p^2 . The table shows each setting's performance on the task of node selection when $N=183$ and $P=10$

We also observe a higher TPR of node selection under no functional domain selection but a lower FPR and higher PPV of node selection. We note, however, that despite domain selection reducing the dimensionality of the functional covariate by about 80%, its performance in terms of TPR is still at a comparable level to that of no reduction. Next, we compare our method with other alternatives, as shown in Table 3.2 for domain selection performance. FLM with 23 Bsplines has the highest TPR, followed by FLM with 13 Bsplines. Both of the aforementioned methods, however, have higher FPR than BMIM. In terms of node selection, the GP-on-FPCA scores stand out as a top performer on FPR and PPV. Fused Lasso has better TPR, but the GP-on-FPCA is not far behind. We summarized the simulation results of Case (1) in Figures 3.1-3.3, in which we can see that our method outperforms in terms of TPR, FPR, PPV, and F1. We also observe similar simulation results when $P = 30$ as shown in Tables 3.3-3.4.

		BMIM	Fused Lasso	Lasso	FPCA-GP	FLM-BS(13)	FLM-BS(23)
Domains	TPR	0.337	0.333	0.042	-	0.000	0.387
	FPR	0.184	0.390	0.033	-	0.000	0.243
	PPV	0.314	0.176	0.246	-	-	0.285
	F₁	0.325	0.230	0.072	-	-	0.328
Nodes	TPR	0.934	0.997	0.693	0.997	-	0.497
	FPR	0.078	0.091	0.464	0.062	0.000	0.359
	PPV	0.836	0.825	0.390	0.873	-	0.372
	F₁	0.883	0.903	0.500	0.931	-	0.426
Overall	$P(\text{Reject } H_0 H_0)$	0.032	-	-	-	-	-
	$P(\text{Reject } H_0 H_1)$	1.00	-	-	-	-	-

Table 3.2: Comparison against other valid methodologies under Case1: BMIM (under the best performing node selection setting); Fused Lasso; Lasso; GP-on-FPCA; FLM-BS(13); FLM-BS(23). FLM-BS(B)=FLM-Bspline with the B number of Bsplines. The table shows the chosen models' (1) domain selection performance metrics, (2) node selection performance metrics, (3) overall significance under the null (Type-I error), and (4) overall significance under the alternate (Power) when N=183 and P=10

		Domain Selection		No Domain Selection	
		BMIM-1	BMIM-2	BMIM-1	BMIM-2
Nodes	TPR	0.749	0.893	0.817	0.939
	FPR	0.037	0.101	0.036	0.102
	PPV	0.690	0.496	0.716	0.506
	F₁	0.718	0.638	0.763	0.657

Table 3.3: Comparison of different BMIM node selection settings under Case 1: BMIM-1: McMC-on- K_p^1 ; BMIM-2: McMC-on- K_p^2 . The table shows each setting's performance on the task of node selection when N=183 and P=30

		BMIM	Fused Lasso	Lasso	FPCA-GP	FLM-BS(13)	FLM-BS(23)
Domains	TPR	0.329	0.327	0.040	-	0.000	0.000
	FPR	0.186	0.378	0.028	-	0.000	0.000
	PPV	0.306	0.178	0.265	-	-	-
	F₁	0.317	0.230	0.070	-	-	-
Nodes	TPR	0.749	0.997	0.648	1.000	0.000	0.000
	FPR	0.037	0.090	0.195	-	0.000	0.000
	PPV	0.690	0.553	0.270	0.1	-	-
	F₁	0.718	0.711	0.381	0.182	-	-
Overall	$P(\text{Reject } H_0 H_0)$	0.149	-	-	-	-	-
	$P(\text{Reject } H_0 H_1)$	0.991	-	-	-	-	-

Table 3.4: Comparison against other valid methodologies under Case1: BMIM: BMIM (under the best performing node selection setting); Fused Lasso: Fused Lasso; Lasso: Lasso; FPCA-GP: GP-on-FPCA; FLM-BS(13): FLM-BSpline (B=13); FLM-BS(23): FLM-BSpline (B=23). The table shows the chosen models' (1) domain selection performance metrics, (2) node selection performance metrics, (3) overall significance under the null (Type-I error), and (4) overall significance under the alternate (Power) when N=183 and P=30

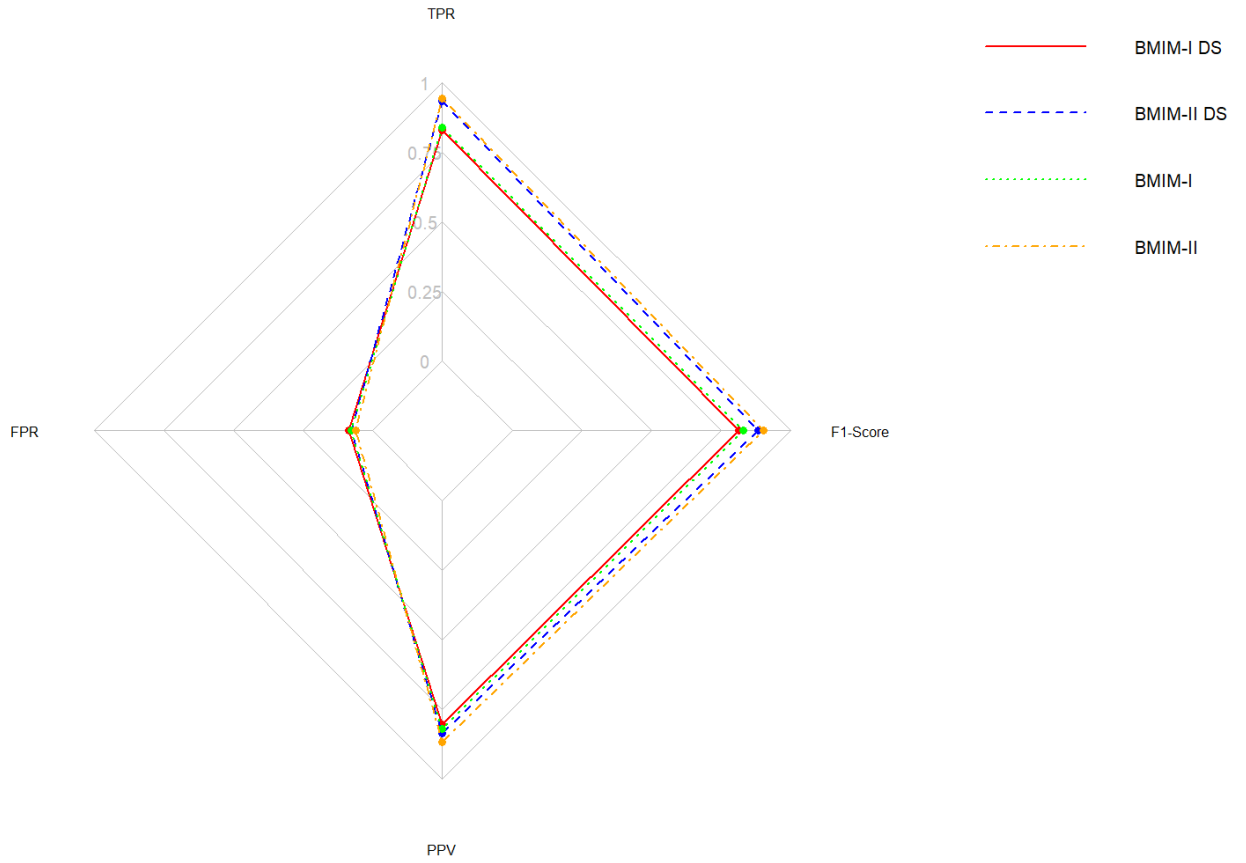


Figure 3.1: Comparison of method performance between BMIM-1 and BMIM-2 when domain selection is applied .vs. when domain selection is not applied. Data is generated using Case 1 (scalar on functional regression with a coefficient function). BMIM-I is our method with K_p^1 applied to the unreduced functional data; BMIM-II is our method with K_p^2 applied to the unreduced functional data. This figure corresponds to table 3.1 with $P=10$

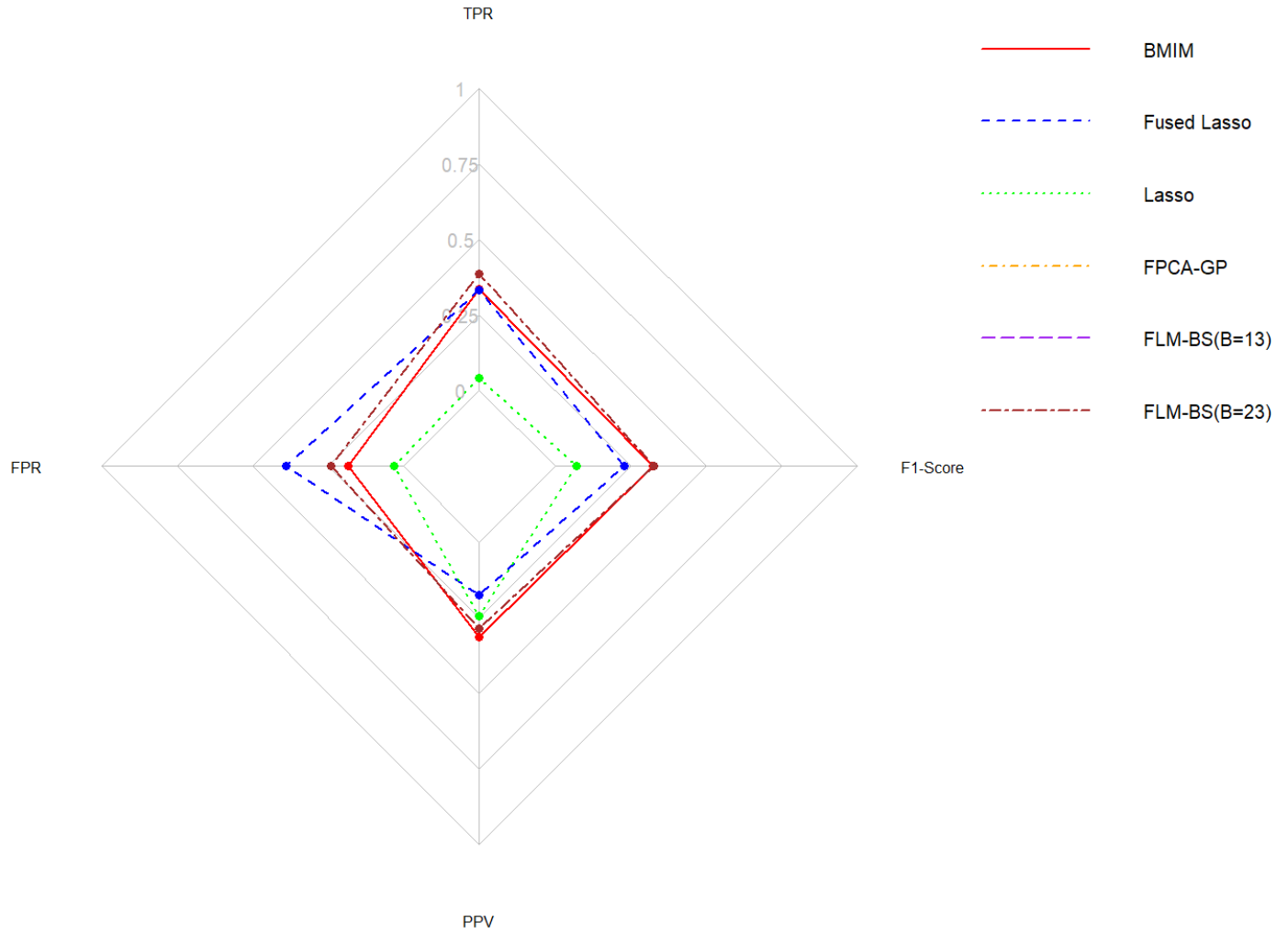


Figure 3.2: Comparison of functional domain selection performance among BMIM, Fused Lasso, Lasso, FLM-Bspline (B=13), and FLM-Bspline (B=23), in terms of TPR, FPR, PPV, and F1-score. Data is generated using scalar on functional regression with a coefficient function. This figure corresponds to the "Domains" section of table 3.2 with $P=10$

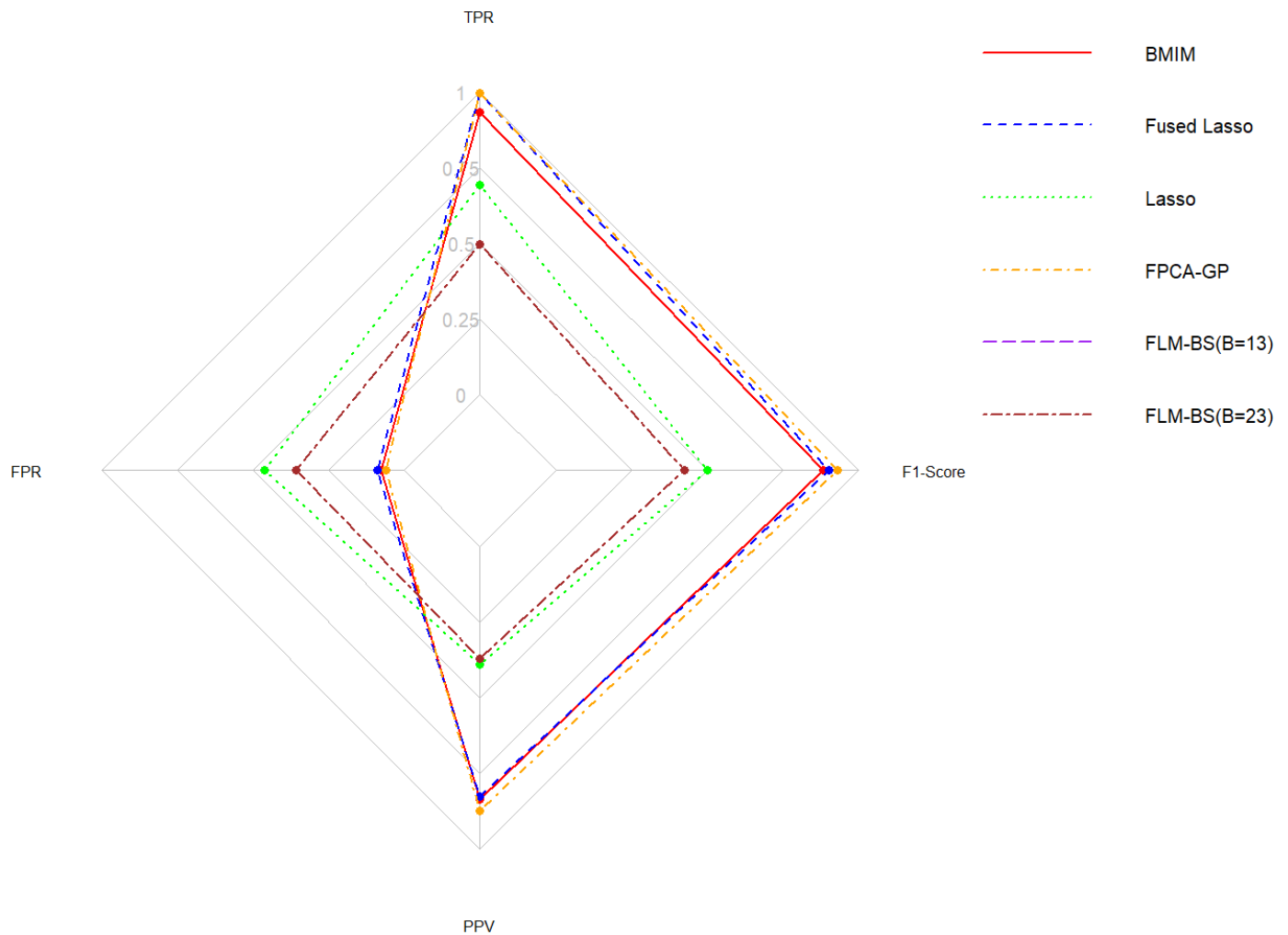


Figure 3.3: Comparison of node selection performance among BMIM, Fused Lasso, Lasso, FLM-Bspline (B=13), and FLM-Bspline (B=23) in terms of TPR, FPR, PPV, F1-Score. Data is generated using scalar on functional linear regression with a coefficient function. This figure corresponds to the "Nodes" section of table 3.2 with $P=10$

Overall, our BMIM follows closely behind, representing a good compromise between a method that offers both domain selection and node selection while only taking a minor hit to node selection performance.

Second, under Case (2), as shown in Table 3.5, once more, we observe that BMIM-2 significantly outperforms BMIM-1, both with and without functional domain selection. Among BMIM-2, we observe a higher TPR under no functional domain selection but a lower FPR and higher PPV when we select a functional domain. This is opposite to our previous data generation case, as such we are inclined to believe that domain selection performs on par with no domain selection despite reducing the dimensionality of the functional covariate by about 80%.

		Domain Selection		No Domain Selection	
		BMIM-1	BMIM-2	BMIM-1	BMIM-2
Nodes	TPR	0.836	0.997	0.873	0.995
	FPR	0.122	0.053	0.119	0.045
	PPV	0.745	0.889	0.759	0.904
	F₁	0.788	0.940	0.812	0.947

Table 3.5: Comparison of different BMIM node selection settings conditional on selected domains under Case 2: BMIM-1: McMC-on- K_p^1 ; BMIM-2: McMC-on- K_p^2 . The table shows each setting's performance on the task of node selection when $N=183$ and $P=10$

Compared to other methods, as shown in Table 3.6, on domain selection performance, BMIM is the overall best performer based on the F_1 metric, offering the best FPR and PPV with the exclusion of the typical lasso. In Case 2, where non-parametric model generation setting, BMIM also performs best on node selection, with the closest competitor being Fused Lasso, which can produce domain selection but cannot provide an overall test for functional covariates due to its utilization of a penalized likelihood in estimation. We summarized the simulation results of Case (2) in Figures 3.4-3.6, in which we can see that our method outperforms in terms of TPR, FPR, PPV, and F_1 . We also observe similar simulation results when $P = 30$ as shown in Tables 3.7-3.8.

		BMIM	Fused Lasso	Lasso	FPCA-GP	FLM-BS(13)	FLM-BS(23)
Domains	TPR	0.393	0.310	0.028	-	-	0.258
	FPR	0.162	0.508	0.016	-	-	0.234
	PPV	0.377	0.132	0.303	-	-	0.216
	F₁	0.385	0.186	0.052	-	-	0.235
Nodes	TPR	0.997	1.000	0.868	0.667	-	0.468
	FPR	0.053	0.100	0.234	0.273	0.000	0.417
	PPV	0.889	0.811	0.614	0.511	-	0.325
	F₁	0.940	0.896	0.719	0.579	-	0.384
Overall	$P(\text{Reject } H_0 H_0)$	0.089	-	-	-	-	-
	$P(\text{Reject } H_0 H_1)$	1.00	-	-	-	-	-

Table 3.6: Comparison against other valid methodologies under Case 2: BMIM: BMIM (under the best performing node selection setting); Fused Lasso: Fused Lasso; Lasso: Lasso; FPCA-GP: GP-on-FPCA; FLM-BS(13): FLM-BSpline (B=13); FLM-BS(23): FLM-BSpline (B=23). The table shows the chosen models' (1) domain selection performance metrics, (2) node selection performance metrics, (3) overall significance under the null (Type-I error), and (4) overall significance under the alternate (Power) when N=183 and P=10

		Domain Selection		No Domain Selection	
		BMIM-1	BMIM-2	BMIM-1	BMIM-2
Nodes	TPR	0.863	0.995	0.835	1.000
	FPR	0.047	0.056	0.042	0.067
	PPV	0.669	0.662	0.686	0.625
	F₁	0.754	0.795	0.753	0.769

Table 3.7: Comparison of different BMIM node selection settings conditional on selected domains under Case 2: BMIM-1: McMC-on- K_P^1 ; BMIM-2: McMC-on- K_P^2 . The table shows each setting's performance on the task of node selection when N=183 and P=30

		BMIM	Fused Lasso	Lasso	FPCA-GP	FLM-BS(13)	FLM-BS(23)
Domains	TPR	0.387	0.260	0.024	-	0.000	0.000
	FPR	0.171	0.462	0.013	-	0.000	0.000
	PPV	0.361	0.123	0.322	-	-	-
	F₁	0.374	0.168	0.045	-	-	-
Nodes	TPR	0.995	1.000	0.801	1.000	-	-
	FPR	0.056	0.088	0.072	0.998	0.000	0.000
	PPV	0.662	0.559	0.551	0.100	-	-
	F₁	0.795	0.717	0.653	0.182	-	-
Overall	$P(\text{Reject } H_0 H_0)$	0.009	-	-	-	-	-
	$P(\text{Reject } H_0 H_1)$	0.942	-	-	-	-	-

Table 3.8: Comparison against other valid methodologies under Case 2: BMIM: BMIM (under the best performing node selection setting); Fused Lasso: Fused Lasso; Lasso: Lasso; FPCA-GP: GP-on-FPCA; FLM-BS(13): FLM-BSpline (B=13); FLM-BS(23): FLM-BSpline (B=23). The table shows the chosen models' (1) domain selection performance metrics, (2) node selection performance metrics, (3) overall significance under the null (Type-I error), and (4) overall significance under the alternate (Power) when N=183 and P=30

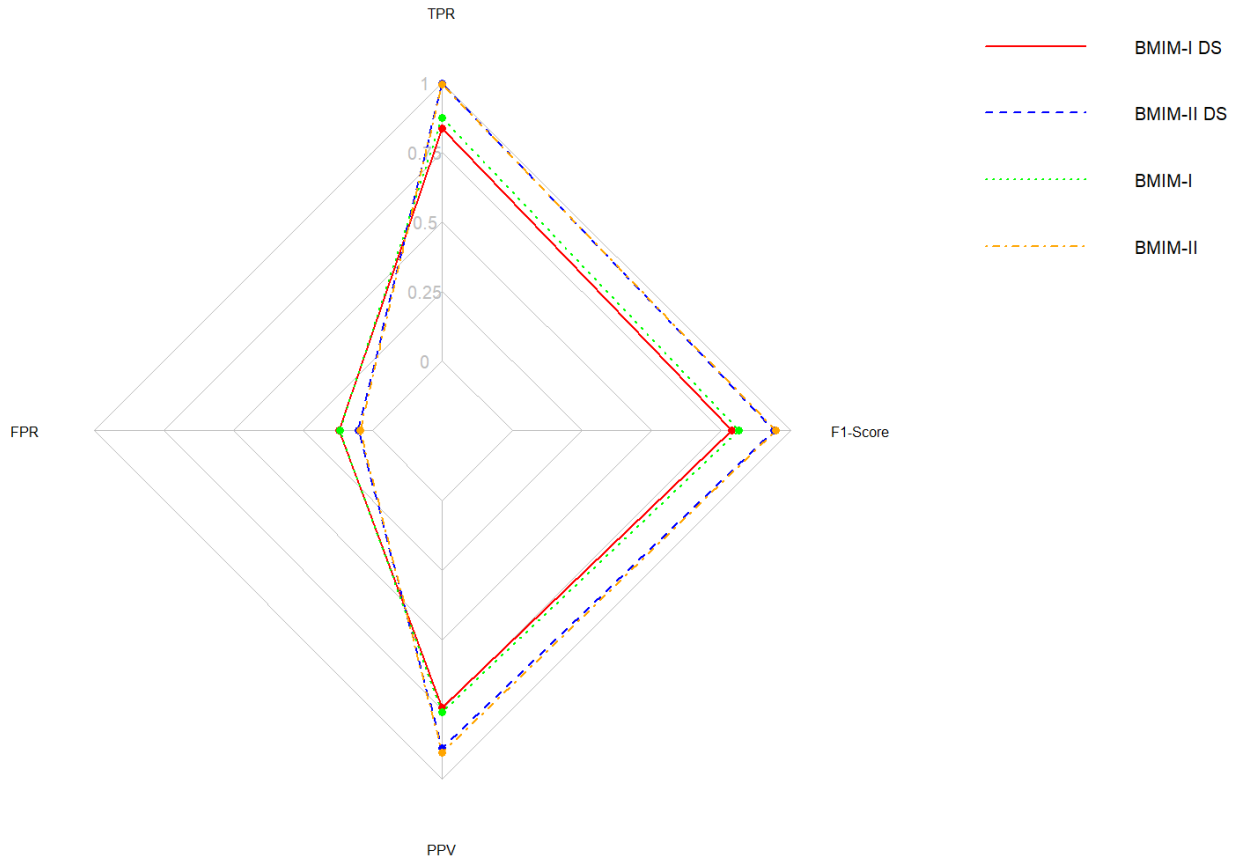


Figure 3.4: Comparison of method performance between BMIM-1 and BMIM-2 when domain selection is applied vs. when domain selection is not applied. Data is generated using the nonparametric model. BMIM-I is our method with K_p^1 applied to the unreduced functional data; BMIM-II is our method with K_p^2 applied to the unreduced functional data data. This figure corresponds to table 3.5 with $P=10$

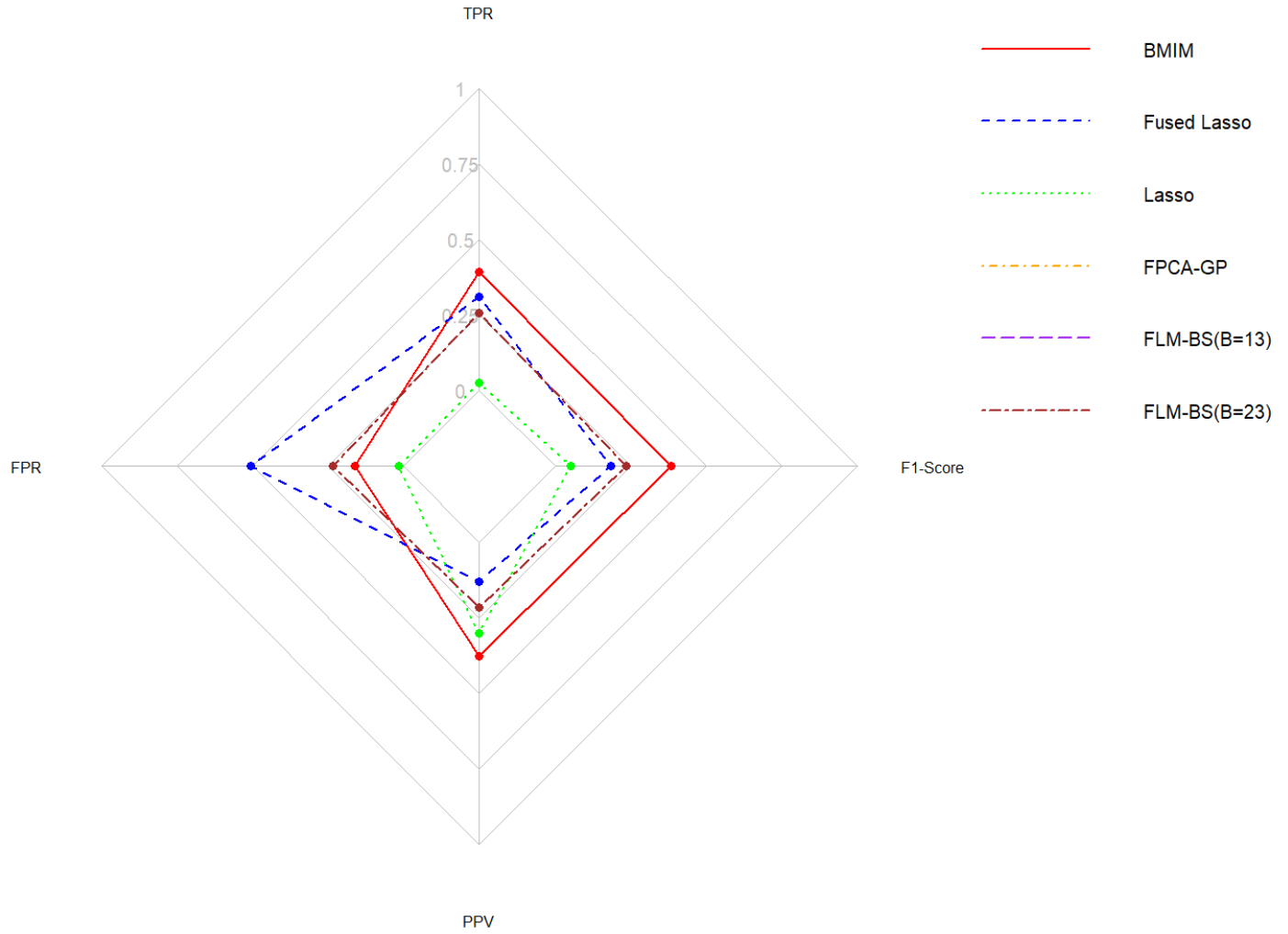


Figure 3.5: Comparison of functional domain selection performance among BMIM, Fused Lasso, Lasso, FLM-Bspline (B=13), and FLM-Bspline (B=23), in terms of TPR, FPR, PPV, and F1-score. Data is generated using a functional nonparametric model. This figure corresponds to the "Domains" section of table 3.6 with $P=10$

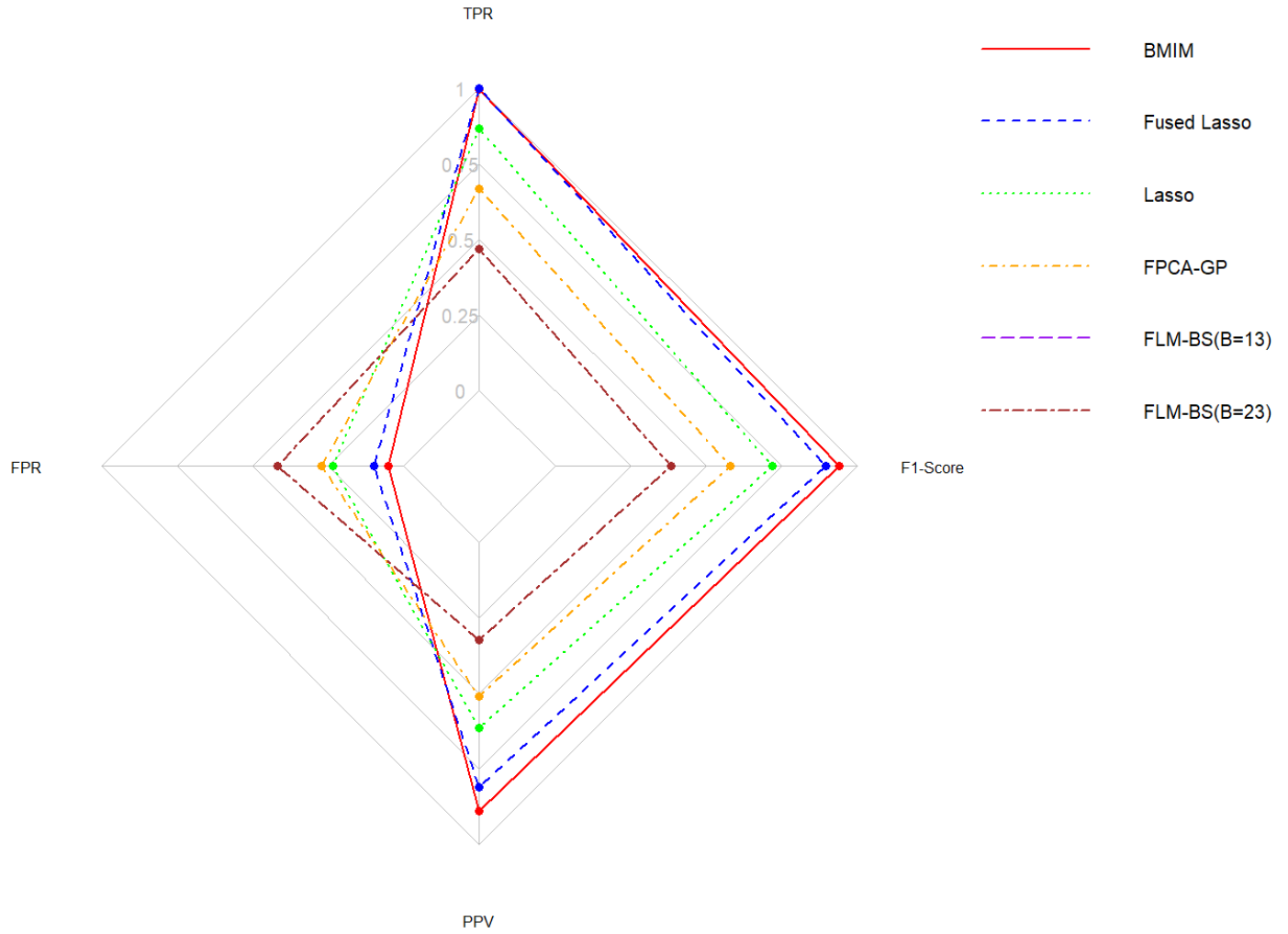


Figure 3.6: Comparison of node selection performance between BMIM, Fused Lasso, Lasso, Bspline Regression (B=13), and Bspline Regression (B=23). Data is generated using a functional nonparametric model. This figure corresponds to the "Nodes" section of table 3.6 with $P=10$

3.8.3 Sensitivity analysis of lengthscale parameters

Finally, we show the results of a sensitivity analysis of our testing approach to the choice of the lengthscale parameter in Figures 3.7-3.8. The horizontal axis represents different values of the lengthscale parameter in our functional input RBF kernel, and the vertical axis represents the probability of observing our test statistic under the null hypothesis, $P(T \geq t|H_0)$. Figure 3.7 shows the results when the data is generated under the null hypothesis, $\tau^2 = 0$, and figure 3.8 shows the results when the data is generated under the alternate hypothesis, $\tau^2 \neq 0$. The boxplots represent p-value estimates across 100 simulations per length-scale setting. The dashed red line represents the 0.05 threshold.

The optimal length scale parameter cannot be estimated as it vanishes in the restricted likelihood due to existing inside the kernel. We observe that our testing results are robust to different lengthscale choices.

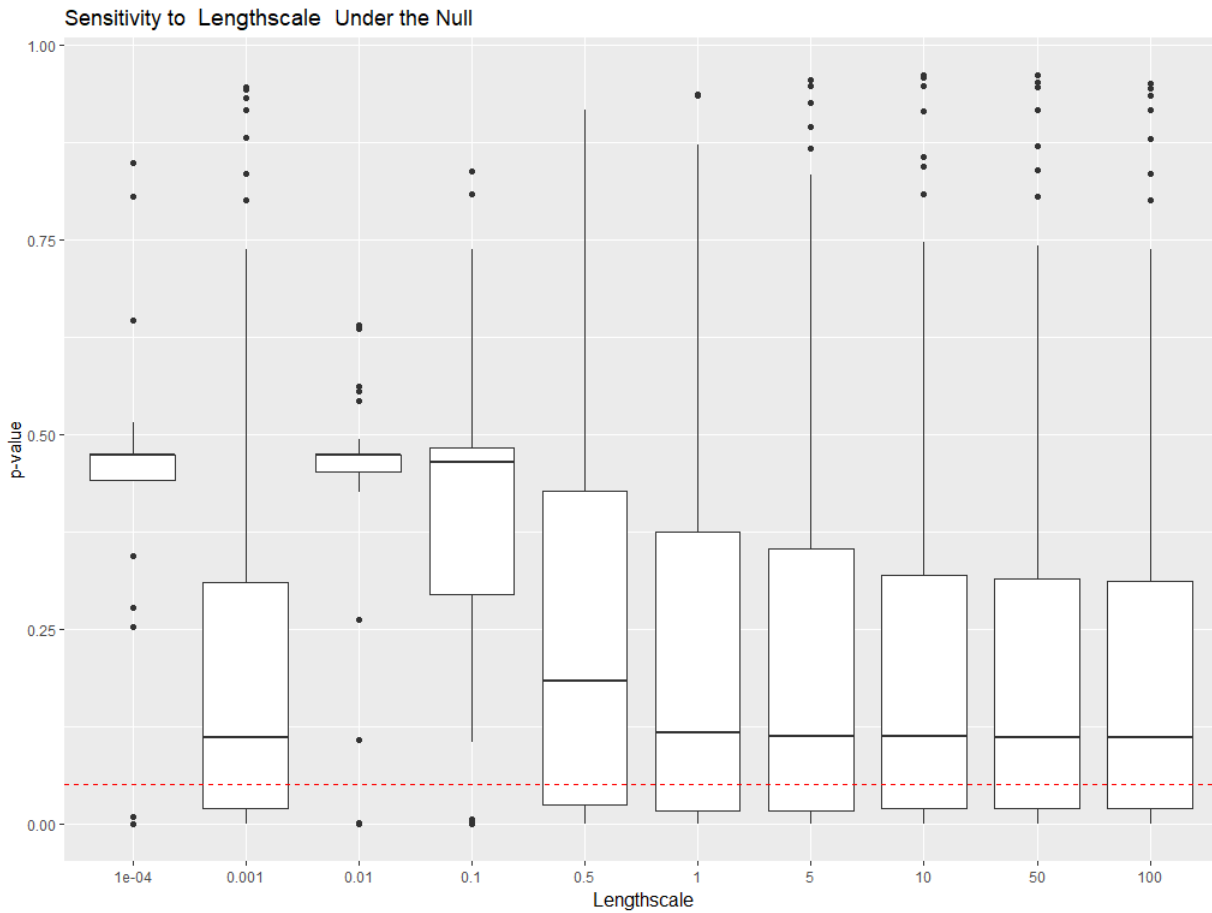


Figure 3.7: Sensitivity analysis for choice of lengthscale versus the significance of the functional covariates when all functional covariates are unrelated to the response, which is null Case. The boxplots represent p-value estimates across 100 simulations per lengthscale setting. The dashed red line represents the 0.05 threshold.

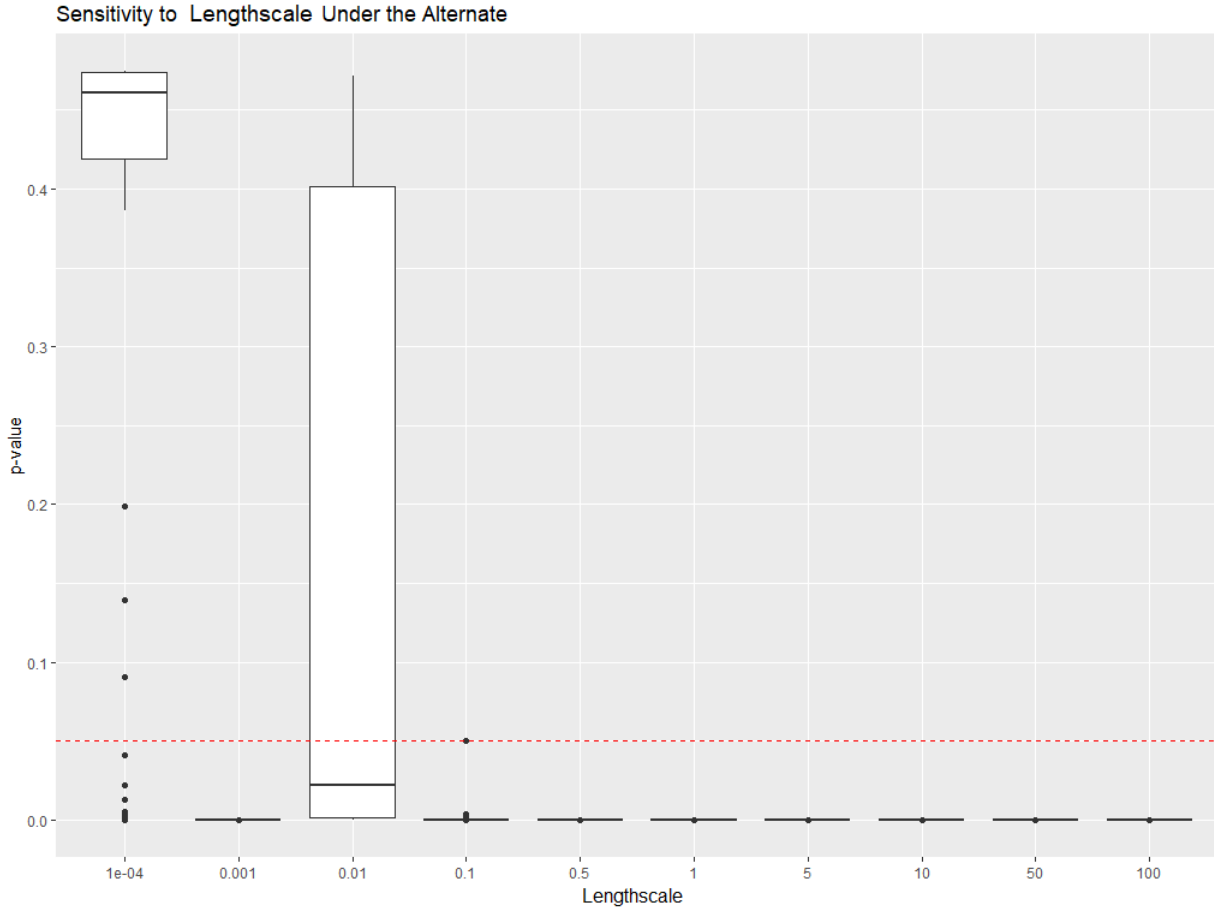


Figure 3.8: Sensitivity analysis for choice of lengthscale versus the significance of the functional covariates when some functional covariates are related to the response, which is the alternate case. The boxplots represent p-value estimates across 100 simulations per lengthscale setting. The dashed red line represents the 0.05 threshold.

3.9 Functional magnetic resonance imaging (fMRI) data in the human brain contributing to the development of Autism Spectrum Disorder

Our driving problem is the identification of regions in the human brain contributing to the development of autism spectrum disorder (ASD) within an individual. ASD, caused by structural differences in the brain, is a neurological developmental disability that affects how people communicate, learn, and behave. People diagnosed with ASD suffer difficulties in

self-behavior and social communication. According to the latest available data from the U.S. Centers for Disease Control, the estimated prevalence of ASD is about three percent among American children, highlighting the importance of building a better understanding of this disease through identifying brain regions associated with ASD.

Our data source is the autism brain imaging data exchange (ABIDE) dataset, which observes individuals across five countries with ages ranging from 2 to 64 years. The dataset was collected over a time span of 5 years. It is a collection of functional magnetic resonance imaging (fMRI) and demographics data on subjects with and without ASD. fMRI is a neuroimaging technique used to measure brain activity by detecting changes in blood flow and oxygenation levels in response to some neural activity, and this is based on the premise that when neurons become active, they require more oxygen, leading to an increase in blood flow to the active brain regions, resulting in a localized increase in the concentration of oxygenated hemoglobin, which can be detected using MRI. The type of fMRI data we are working with is called resting-state fMRI, where the imaging process was done while the subject was at rest (i.e., not engaged in any particular task).

We use a subset of the ABIDE data consisting of seven functional networks within the brain Yeo et al. (2011): Visual, SomatoMotor, Dorsal Attention, Salience/Ventral Attention, Limbic, Control, Default Mode, which are displayed in Figure 3.9 and also mimic functional networks of FMRI signals which we want to build. We observe a sample of 341 individuals, and our response variable is the autism diagnostic observation schedule (ADOS) score, which is generated from the ADOS tool, a diagnostic tool for ASD that has been shown to be highly correlated to an individual's diagnosis. For our functional covariates, we choose fMRI data at a frequency of 2 TR/sec (repetition time) for a total of 100 observed time points, as that had the greatest number of associated available observations. Restricting our data to only samples with an fMRI frequency of 2 TR/sec reduced our sample size to 183, with 173 patients and 10 controls. The demographic variables we used were age at scan, sex, and

handedness (left or right handed).

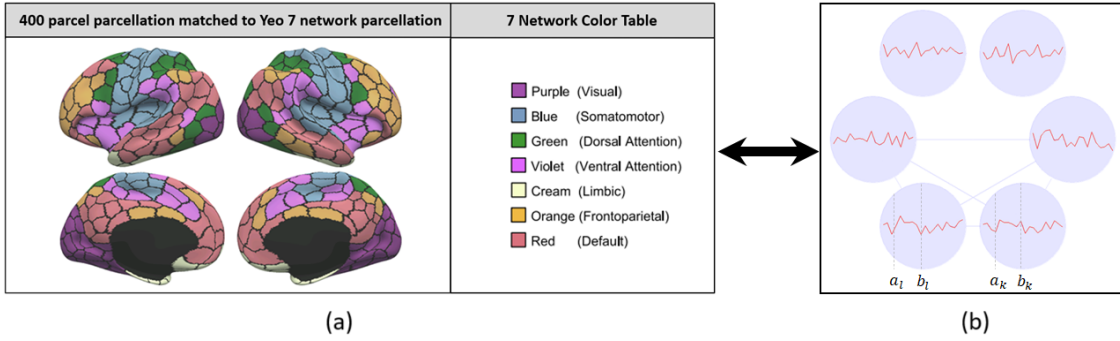


Figure 3.9: (a) shows predefined brain networks and the parcellations that form them. (b) shows a generated sample that mimics the observed functional MRI signals generated from each parcellation in the network. Visualization of parcellations from https://github.com/ThomasYeoLab/CBIG/tree/master/stable_projects/brain_parcellation/Schaefer2018_LocalGlobal. Original network source is from Yeo et al. (2011)

In our study, we denote $Y = \text{ADOS scores}$, $X^{(p)}(t) = \text{an fMRI signal curve at time } t \text{ from the } p^{\text{th}} \text{ parcellation}$, these are treated as functional covariates. The number of functional variables (P) are varied by network: $P = (17, 14, 12, 5, 15, 24, 13)$ for seven networks (Visual, SomatoMotor, Salience/Ventral Attention, Limbic, Dorsal Attention, Default Mode, Control). Our main interest is to identify subsets on the domain of the functional covariates and graph structure among locations that are relevant to the ADOS scores. We also aim to test the overall significance of the association between the functional covariates and the response for each of the examined structural brain networks.

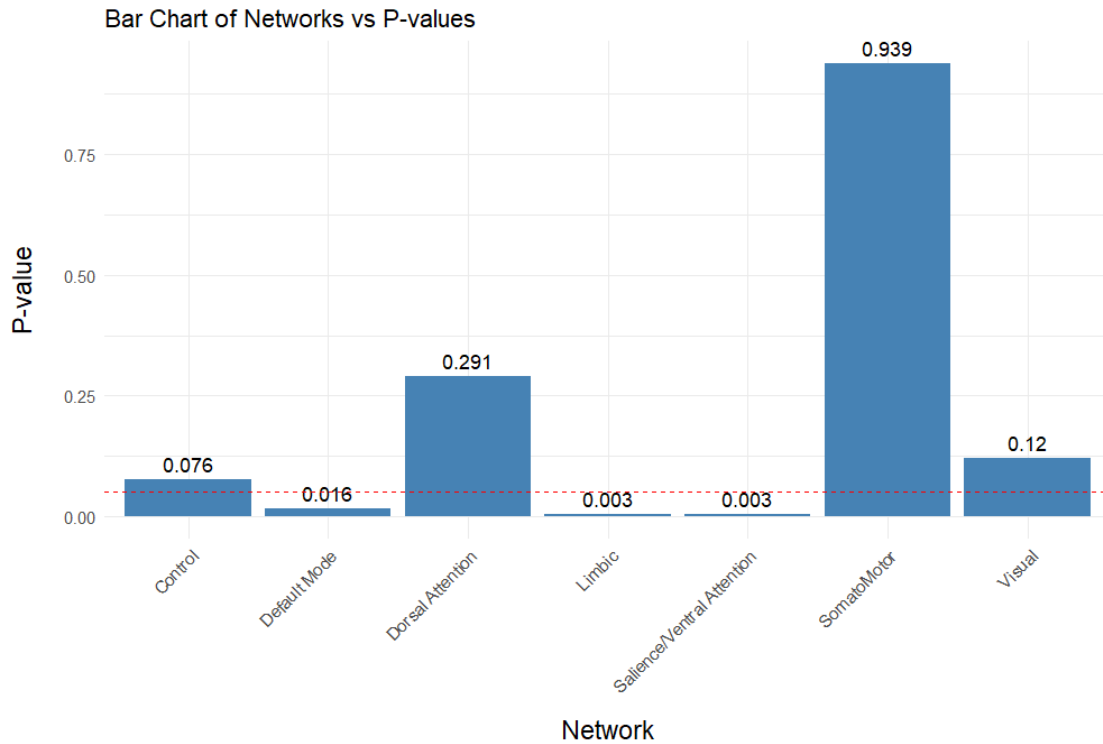


Figure 3.10: P-values of null hypothesis on real data networks, red dashed line indicates 0.05 p-value threshold

We apply our BMIM to each network. Selected time intervals, nodes, and score tests are summarized in Tables 3.9 - 3.10. Our method significantly detects 3 out of 7 networks, which are “the Saliency/Ventral Attention network”, “Limbic network”, and “the Default Mode network”, as potential predictors of ADOS with p-values (0.003, 0.003, 0.016). Our findings are also supported by neuroscience literature. Based on Uddin (2015), the “Saliency/Ventral Attention network” is crucial in sensory integration during the initial sensory/attentional processing state, prioritizing incoming sensory inputs from multiple neural sources to efficiently utilize limited neural resources for subsequent higher cognitive processing networks, such as the frontoparietal system. Lee and Telzer (2016) supported that the “Limbic network” plays a role in adjusting bottom-up sensory gains based on emotional and motivational values, facilitating rapid and automatic behavioral responses. Consistently, abnormalities in the neural activity within these two networks have been reported in individuals with ASD

(Green et al., 2016, Haznedar et al., 2000). Furthermore, a recent neural network dynamic model suggests that the activation of the “Default-mode network” and “Frontoparietal network” is determined by the results of early sensory integration processes within the anterior insula (AI) and anterior cingulate cortex (ACC) (Chen et al., 2022, Menon and Uddin, 2010).

Network	Total # of nodes	Selected time interval	Selected nodes	Score test
1: Visual	17	(25,43) (78,96) (78,92) (38,67) (62,83) (10,29) (72,93) (43,59) (82,99) (44,66) (37,57) (69,87)	LH-1 LH-3 LH-4 LH-5 LH-6 LH-7 RH-1 RH-2 RH-3 RH-4 RH-5 RH-6	0.120
2: SomatoMotor	14	(29,48) (24,41) (47,70)	LH-4 LH-6 RH-2	0.939
3: Salience/Ventral Attention	12	(3,24)	LH-FrOperIns-2	0.003
4: Limbic	5	(35,55)	LH-TempPole-2	0.003
5: Dorsal Attention	15	(55,72) (65,86) (47,70) (4,25) (40,61)	LH-DorsAttn-Post-1 LH-PrCv-1 LH-FEF-1 RH-Post-3 RH-FEF-1	0.291

Table 3.9: Selected Nodes within available networks using real data, LH denotes the left hemisphere of the brain, RH denotes the right hemisphere of the brain.

Network	Total # of nodes	Selected time interval	Selected nodes	Score test
6: Default Mode	24	(62,82)	LH-Temp-1	0.016
		(35,55)	LH-Par-1	
		(78,100)	LH-PFC-1	
		(13,32)	LH-PFC-2	
		(59,83)	LH-PFC-3	
		(61,82)	LH-PFC-5	
		(62,81)	LH-PFC-6	
		(40,60)	LH-PFC-7	
		(12,31)	LH-pCunPCC-1	
		(26,46)	RH-Temp-1	
		(29,50)	RH-Temp-3	
		(8,26)	RH-PFCdPFCm-1	
		(56,74)	RH-PFCdPFCm-3	
		(72,93)	RH-pCunPCC-1	
		(29,48)	RH-pCunPCC-2	
7: Control	13	(69,93)	LH-Par-1	0.076
		(77,97)	LH-PFCl-1	
		(53,72)	LH-Cing-1	
		(50,70)	RH-Par-1	
		(73,92)	RH-Par-2	
		(27,49)	RH-PFCl-2	
		(69,91)	RH-PFCl-3	
		(54,72)	RH-Cing-1	
		(3,21)	RH-PFCmp-1	
		(57,77)	RH-pCun-1	

Table 3.10: Selected Nodes within available networks using real data, LH denotes the left hemisphere of the brain, RH denotes the right hemisphere of the brain.

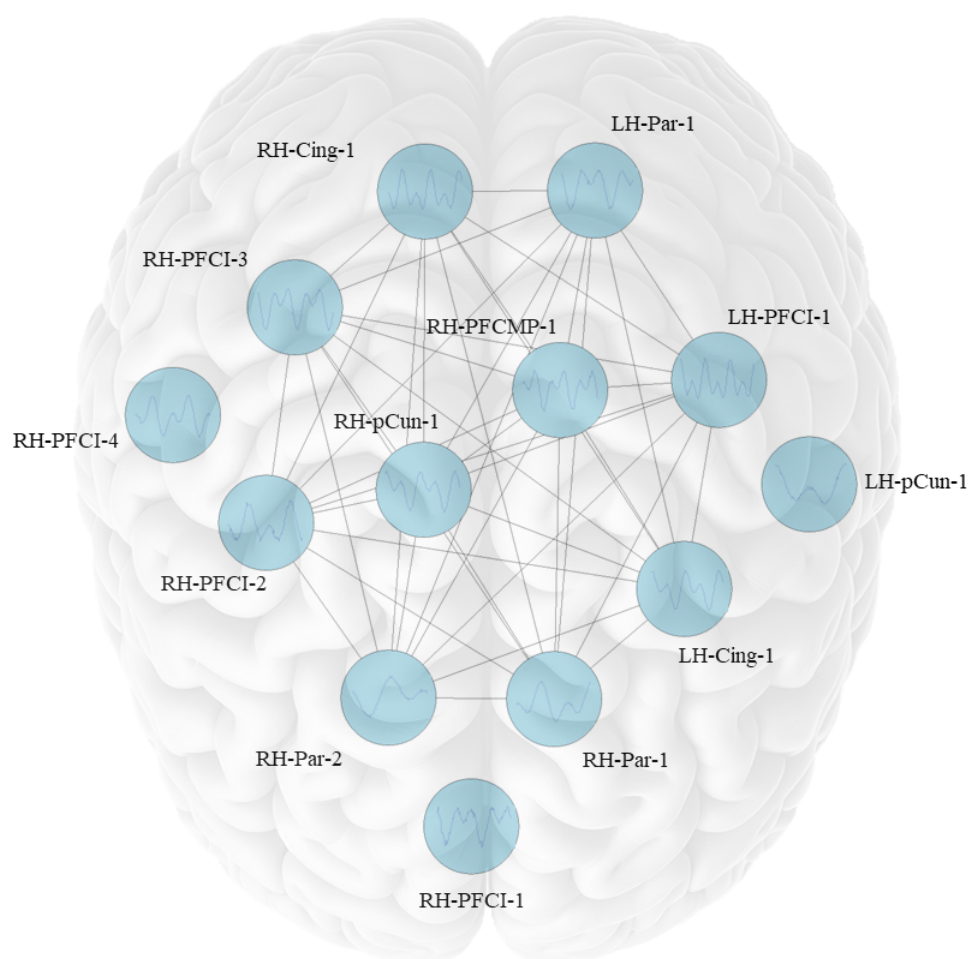


Figure 3.11: Illustrative visualization of connectivity within the Control network.

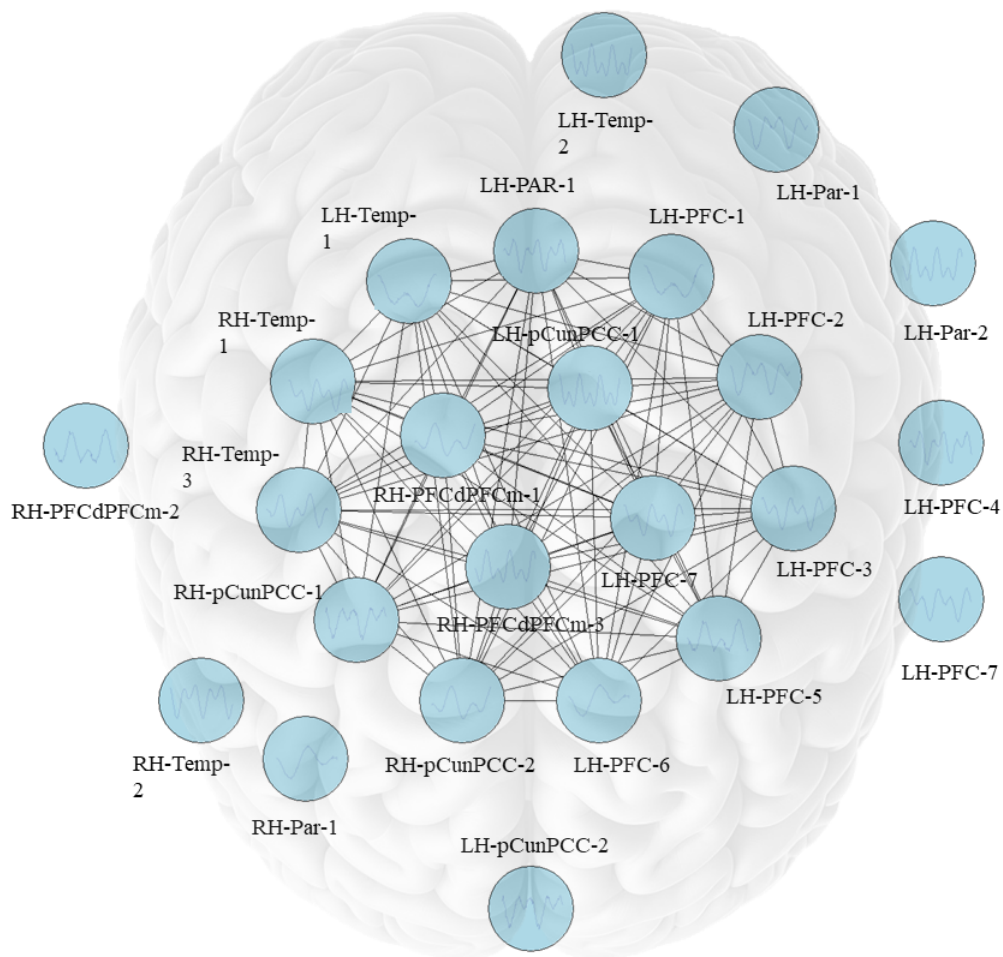


Figure 3.12: Illustrative visualization of connectivity within the Default Mode network.

3.10 Discussion

In this chapter, we develop a Bayesian multi-tasking inference method under a joint functional kernel graphical model framework. Our method unifies Bayesian optimization, an Ising graphical model, and a score-type test using a Bayesian approach so that we can simultaneously perform (1) the selections of functional domain subsets, (2) selections of functional variables, and (3) the selection of network among functional variables in the nonadditive and nonparametric model and (4) finally test whether these selected variable sets is significantly associated to a response variable.

Our approach seeks to avoid the specification of basis functions or, indeed the imposition of any assumptions on both the functional covariate coefficients and the response variable. By incorporating a black-box hyperparameter optimization problem, which we tackle using Bayesian optimization and discrete Ising graphical model into semiparametric kernel machine model framework, we propose semiparametric joint functional kernel graphical model and then achieve our multi-taking inference described in (1)-(4). To the best of our knowledge, no methods can achieve these multi-tasks simultaneously.

Our method can be extended to sets of heterogeneous functional covariates so that we can select more than one functional domain. Our method can also be further developed under unknown heterogeneous populations. We can build our method using a finite mixture of kernel models. However, it will require more computation cost and careful optimization procedure.

We note that we have explored introducing knowledge of functional node interconnectivity via priors on selection variables. This approach is equivalent to introducing connectivity knowledge via the lengthscales parameters within the nonparametric portion of our model. Introducing the connectivity via a single prior on the lengthscales parameters may offer greater efficiency. We will further consider this problem for future research.

Chapter 4

Bayesian Inference in Transformer-based Models

4.1 Introduction

With the recent rise in popularity of deep learning-based models came a flurry of research exploring the application of these models to multiple data modalities. Deep learning-based models have been successfully applied to tabular data, functional data and image data. Some such applications aimed to capitalize on deep neural nets' extensive learning and feature extraction capabilities to improve the classification of diseases and abnormalities. In this avenue, several studies have focused purely on the task of identifying disease markers present within functional covariates to produce good classification performance. Early approaches, however, lacked a coherent method of introducing domain knowledge into the architecture, focusing instead on the model's ability to extract features.

More recent work on the matter has attempted to incorporate structure within the data, such as the idea of functional connectivity, (Allen et al., 2012, Rogers et al., 2007), which acknowledges the existence of network interactions between separate functional covariates. For instance, using resting state fMRI (rs-fMRI) data, Suk et al. (2016) leverage an autoencoder to learn the functional connectivities and cast them into an embedding space, after which a hidden Markov model is applied to the transformed space to predict mild cognitive impairment. Riaz et al. (2020) also incorporated functional connectivity as a submodel within their architecture and showed that its inclusion markedly improves attention-deficit/hyperactivity disorder(ADHD) classification performance. Likewise, Wang et al. (2023) calculates functional connectivity using a phase synchronization index, then uses a convolutional neural network (CNN) to extract features relevant to the prediction of Alzheimer's. Shao et al. (2021) combine a graph convolutional neural network (GCNN) preceded by sparsity-inducing multilayer perceptron in their proposed architecture to first select for a subset of functional connections that are then passed onto the GCNN. Using image data, Kong et al. (2019) used stacked sparse autoencoders to select functional connections, producing high classification performance. Heinsfeld et al. (2018) also employs an autoencoder approach, although they

use denoising autoencoders to cast the data into an embedding space onto which a classifier is trained. Other approaches involving convolutions include Gao et al. (2020), Liang et al. (2021), and Nogay and Adeli (2024). A common theme across these attempts is the incorporation of a pre-CNN model that first accounts for connectivity across multiple functional covariates before passing on to the CNN model, which then performs the task of capturing local information, possibly in a transformed space. As such, CNN-based models require a custom pre-processing step with no proven guarantees or general guidance on how to set up an effective pre-processing model. Additionally, the translation invariance property of CNNs acts as a double-edged sword; while CNNs learn filters that are convolved with the input data for prediction, this approach eschews positioning knowledge in favour of pattern-matching, which may not be ideal for data where position matters.

Increasingly, CNNs have been facing a new competitor model, the transformer. First introduced in (Vaswani et al., 2017), transformers are a neural network architecture that is characterized by the presence of an attention mechanism. Originally developed for natural language processing (NLP), these models can learn interdependencies present within an observation by leveraging a sub-model that produces an attention map. An attention map is a learned weighting of the strength of the connection between elements of an observation, such as words in a sentence, which makes transformers ideal for NLP. More recent work has expanded the applications of transformers; Dosovitskiy et al. (2020), through a minor modification of the architecture that eschews the decoder and introduces a class token indicating the class label, extended transformers to image classification, in the process making it the new state-of-the-art (SOTA) model architecture for image classification. As such, the transformer architecture has been shown to possess the ability to learn and leverage both temporal and spatial connections as it can be fed both relative and absolute positional input, offering a simple way to encode and learn connectivity-related domain knowledge without the need for custom pre-processing steps. Some work has already shown that transformers

can achieve similar classification performance as the CNN-based models without the need for a secondary, pre-processing model; in Li et al. (2022), the authors leverage the relatively new transformer architecture to incorporate functional connectivity information into the modelling approach with promising results.

More recently, Zhang and Yan (2023) proposed the crossformer, a transformer architecture designed for multivariate time series data which handles dependencies across both time and dimension by introducing a two-stage-attention (TSA) mechanism. The goal of TSA was the forecasting of univariate time-series data using multivariate time-series data as covariates, a function-on-function(s) regression. Yet, the structure is general enough that it can be extended to classification in the fashion of Dosovitskiy et al. (2020), which we aim to accomplish in this work.

Early results suggest that transformer-based models can be a promising tool for the classification problem given a functional dataset. However, sensitive applications such as medical diagnosis also require a strong understanding of model uncertainty, knowing when we should be less trusting of our model's results. The Bayesian approach offers an intuitive way to incorporate uncertainty quantification into our models. Fully Bayesian neural nets, however, are typically computationally intensive and difficult to train (Blundell et al., 2015, Wang and Yeung, 2016). Given the often over-parametrized nature of neural net architectures, sampling from the posterior of the neural net weights quickly becomes intractable for non-trivial problems. As such, approximate methods are typically utilized in lieu of a full Bayesian approach (He and Jiang, 2023). Furthermore, Zeng et al. (2018) shows that adopting only a partial Bayesian approach can substitute for a full Bayesian model, where it was demonstrated that one can achieve virtually full epistemic uncertainty (model uncertainty) quantification by implementing a Bayesian approach only on a limited number of layers near the output of a neural network. Sankararaman et al. (2022) extends this work to the transformer architecture and connecting to Gal and Ghahramani (2016), provides a

simple dropout-based implementation of Bayesian inference.

Given the relatively smaller size of our motivating dataset as compared to typical datasets used with transformer architectures, the need for regularization becomes critical. While dropout itself is considered a regularization approach, shown in Gal and Ghahramani (2016) to produce an L2-regularization equivalence, we expect more may be needed. Both Graves (2011) and Titsias and Lázaro-Gredilla (2014) proposed utilizing an empirical Bayes approach to learn the hyperparameters of the proposed prior on the weights of the network. Blundell et al. (2015) observes that this is typically slow to converge in practice and may not work well. The former proposed a spike-slab (George and McCulloch, 1993) type prior on the weights which leads to an easily updated variational posterior on the weights,; however, the mixture parameters were assumed known and constant for all weights in the network. In this work, we propose a new regularizing approach that simultaneously performs regularization to handle smaller datasets and prune the model, effectively performing an architecture search.

Our proposed approach utilizes a variational Ising distribution, which has been utilized before to prune neural networks in the literature (Salehinejad and Valaee, 2019, 2021). However, the similarities between our proposed approach and the former are restricted to the utilization of the same distribution for pruning, with the implementations being significantly different.

4.1.1 Main contribution

The main contributions of our work here are as follows: we aim to leverage new, state-of-the-art attention-based architectures to solve the problem of Bayesian inference for classification tasks. We introduce a new regularization approach to enhance the model’s capabilities on smaller datasets. These contributions allow for simultaneous model fitting and model selection by finding the optimal neural network architecture. Our proposed approach is

completely guided by the model likelihood, thereby naturally extending uncertainty quantification to the attention structure to produce uncertainty estimates on both the model classification output and the attention maps on the input space, thereby allowing for inference on the relevance of different portions of the input data. This is of utmost importance for our motivating problem, a high dimensional dataset representing brain locations that display local and global network-like connectivities that simultaneously activate in relation to the disease classification process. Our proposed approach is computationally efficient, and we provide an approximate implementation that is linear in the number of model parameters, and requires little change in the typical transformer training procedure.

4.1.2 Outline of the paper

This paper is organized as follows: In section 4.2, we describe both the typical variational inference approach in attention-based models and our proposed approach. Section 4.3 presents our proposed algorithm. In section 4.4, we present the statistical properties of our approach. Section 4.5, investigates the performance of our proposed approach, given our choice of model architecture, on a set of simulated datasets and compare it to alternative approaches that are typically applied in this context. In section 4.6, we demonstrate the advantage of our approach using the ABIDE dataset. Section 4.7 concludes and summarizes the results of this work. Details of the utilized architecture can be found in the Appendix.

4.2 Likelihood-guided Regularization in Attention-Based Models

In this section, following Sankararaman et al. (2022), we show how to implement Bayesian inference using our proposed new regularizer in Attention-based models.

4.2.1 Classification model

Let $\mathcal{D} = \{y_n, \mathbf{X}_n\}_{n=1}^N$ be a finite dataset, where N is the number of observations. Let y_n be a binary response variable indicating the class label associated with observation, $\mathbf{X}_n = (\mathbf{x}_1, \dots, \mathbf{x}_P)^{(n)}$ represents all the P continuous functional covariates for observation n , and $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_N)^T$ is our full $N \times P \times T$ covariate tensor. Then, an attention-based model can be expressed as

$$Pr(y_n = 1 | \mathbf{X}, \mathbf{W}, \boldsymbol{\xi}) = f_{\mathbf{y}, \mathbf{W}, \boldsymbol{\xi}}(\mathbf{X})$$

where $f_{\mathbf{y}, \mathbf{W}, \boldsymbol{\xi}}(\mathbf{X})$ is the output of an attention based model on a finite dataset $\mathcal{D} = \{y_n, \mathbf{X}_n\}_{n=1}^N$, $\mathbf{W}$ represents the weights of all layers in our model, and finally $\boldsymbol{\xi}$ represents a matrix of binary dropout variables acting as a dropout mask for weights in the model. The dropout scheme proposed here is weight-specific, in the essence of DropConnect Wan et al. (2013), rather than the node-based dropout of Sankararaman et al. (2022).

4.2.2 Variational Bayes for Weights

Adopting a full Bayesian approach for our neural network model would involve placing a prior distribution, $p(\cdot)$, on weights (W) of all linear transformations in the model such that:

$$\mathbf{W} \sim p(\mathbf{W}).$$

Typically, a $\mathcal{N}(\mathbf{W}; 0, \sigma^2)$ prior is placed on the weights with a well chosen σ^2 . Blundell et al. (2015) propose a singular-spike slab prior for all $\mathbf{W}$ as,

$$p(\mathbf{W}) = \prod \pi \mathcal{N}(\mathbf{W}; 0, \sigma_1^2) + (1 - \pi) \mathcal{N}(\mathbf{W}; 0, \sigma_2^2),$$

where π is a mixing probability of two normal distributions and $\sigma_1^2 < \sigma_2^2$ causing some weights to cluster around zero. The formulation in which the prior shares its hyperparameters across

all weights offers greater computational efficiency for optimization. However, this contrasts with the method of assigning hierarchical priors and learning separate hyperparameters for each weight, which, according to Blundell et al. (2015), tends to be ineffective in practice.

While we propose a weight-specific hyperparameter, we will show how to efficiently perform the computation using stochastic gradient descent with low computational cost. Our proposed prior is

$$p(\mathbf{W}|\boldsymbol{\xi}) = \prod_{j,j'} \xi_{j,j'} \mathcal{N}(\mathbf{W}; 0, \sigma_1^2) + (1 - \xi_{j,j'}) \mathcal{N}(\mathbf{W}; 0, \sigma_2^2), \quad \xi_j \in \{0, 1\} \quad \forall j$$

$$\boldsymbol{\xi} \sim p(\boldsymbol{\xi}),$$

where $p(\boldsymbol{\xi})$ is a prior distribution. We are therefore interested in obtaining a posterior on the joint distribution $p(\mathbf{W}, \boldsymbol{\xi}|\mathbf{X}, \mathbf{y}) = p(\mathbf{W}|\boldsymbol{\xi}, \mathbf{X}, \mathbf{y}) \cdot p(\boldsymbol{\xi}|\mathbf{X}, \mathbf{y})$. Recovery of $p(\mathbf{W}|\boldsymbol{\xi}, \mathbf{X}, \mathbf{y})$ allows for quantifying both uncertainty about the weight estimates as well as uncertainty about the prediction through the posterior predictive distribution as,

$$p(\mathbf{y}|\mathbf{X}) = \sum_{\boldsymbol{\xi}} \int p(\mathbf{y}|\mathbf{X}, \mathbf{W}, \boldsymbol{\xi}) p(\mathbf{W}|\boldsymbol{\xi}, \mathbf{X}, \mathbf{y}) p(\boldsymbol{\xi}|\mathbf{X}, \mathbf{y}) d\mathbf{W}.$$

However, given the dimensionality of $\mathbf{W}$ - which is a function of the number of layers, the dimensionality of the embedding space (d_{model}), the number of segments (S), and the number of locations (P) - as well as the discrete nature of $\boldsymbol{\xi}$ this becomes an intractable task, particularly in a high-dimensional setting.

As such, we resort to an approximation of this posterior via Variational Inference (VI) whereby we approximate $p(\mathbf{W}, \boldsymbol{\xi}|\mathbf{X}, \mathbf{y})$ using some distribution $q(\mathbf{W}, \boldsymbol{\xi})$ that is easier to compute. We then attempt to minimize the KL divergence between our proposed approximation and the true posterior distribution. To formulate our objective, we consider the function

$L(q)$,

$$L(q) = \log p(\mathbf{y}|\boldsymbol{\xi}, \mathbf{X}, \mathbf{W}) - \mathbb{KL}(q(\mathbf{W}|\boldsymbol{\xi})||p(\mathbf{W}, \boldsymbol{\xi}|\mathbf{X}, \mathbf{y})) \leq \log p(\mathbf{y}|\boldsymbol{\xi}, \mathbf{X}, \mathbf{W}),$$

where $\mathbb{KL}$ is the Kullback-Leibler divergence; which being a non-negative quantity, implies the inequality. Maximizing this lower bound on the marginal log-likelihood thereby indirectly maximizes the marginal log-likelihood of the data, and as our function of choice q only appears in the KL-divergence term, this is equivalent to minimizing KL-divergence, which, in turn, is equivalent to the following optimization objective,

$$\min_q - \sum_{\boldsymbol{\xi}} q(\boldsymbol{\xi}) \left(\int q(\mathbf{W}|\boldsymbol{\xi}) \cdot \log[p(\mathbf{y}|\mathbf{X}, \mathbf{W}, \boldsymbol{\xi})] d\mathbf{W} + \mathbb{KL}(q(\mathbf{W}|\boldsymbol{\xi})||p(\mathbf{W}|\boldsymbol{\xi})) \right) + \mathbb{KL}(q(\boldsymbol{\xi})||p(\boldsymbol{\xi})).$$

The integral in this expression computes the expectation of $\log p(\mathbf{y}_n|\mathbf{X}_n, \mathbf{W}, \boldsymbol{\xi})$ over all $\mathbf{W}$ assuming $\mathbf{W} \sim q(\mathbf{W}|\boldsymbol{\xi})$. Therefore, we can obtain an unbiased estimate for the integral using Monte Carlo integration (Gal and Ghahramani, 2016) such that,

$$\int q(\mathbf{W}|\boldsymbol{\xi}) \cdot \log[p(\mathbf{y}_n|\mathbf{X}_n, \mathbf{W}, \boldsymbol{\xi})] d\mathbf{W} \approx p(\mathbf{y}_n|\mathbf{X}_n, \widehat{\mathbf{W}}, \boldsymbol{\xi}).$$

Employing our sampling of $\widehat{\mathbf{W}}$ within our stochastic gradient descent leads to the following minimization over this quantity across our observed dataset, which leads to the following form for empirical risk minimization:

$$\min_q - \sum_n \sum_{\boldsymbol{\xi}} q(\boldsymbol{\xi}) \left(\log[p(\mathbf{y}_n|\mathbf{X}_n, \widehat{\mathbf{W}}, \boldsymbol{\xi})] + \mathbb{KL}(q(\mathbf{W}|\boldsymbol{\xi})||p(\mathbf{W}|\boldsymbol{\xi})) \right) + \mathbb{KL}(q(\boldsymbol{\xi})||p(\boldsymbol{\xi})).$$

We then employ a similar Monte Carlo approximation for $\boldsymbol{\xi}$ which has the same dimensionality as $\mathbf{W}$ necessitating the same approach,

$$\sum_{\boldsymbol{\xi}} q(\boldsymbol{\xi}) \left(\log[p(\mathbf{y}_n|\mathbf{X}_n, \widehat{\mathbf{W}}, \boldsymbol{\xi})] + \mathbb{KL}(q(\mathbf{W}|\boldsymbol{\xi})||p(\mathbf{W}|\boldsymbol{\xi})) \right) \approx \left(p(\mathbf{y}_n|\mathbf{X}_n, \widehat{\mathbf{W}}, \hat{\boldsymbol{\xi}}) + \mathbb{KL}(q(\mathbf{W}|\hat{\boldsymbol{\xi}})||p(\mathbf{W}|\boldsymbol{\xi})) \right).$$

Therefore, our objective function is

$$\min_q - \sum_n \log[p(\mathbf{y}_n | \mathbf{X}_n, \widehat{\mathbf{W}}, \hat{\boldsymbol{\xi}})] + \mathbb{KL}(q(\mathbf{W} | \hat{\boldsymbol{\xi}}) || p(\mathbf{W} | \boldsymbol{\xi})) + \mathbb{KL}(q(\boldsymbol{\xi}) || p(\boldsymbol{\xi})). \quad (4.1)$$

The above derivations (4.1) express our problem in the most general terms; for our specific classification problem, we can express it as the following,

$$\log[p(\mathbf{y}_n | \mathbf{X}_n, \widehat{\mathbf{W}}, \hat{\boldsymbol{\xi}})] = \mathbf{y}_n \log \left(f_{\widehat{\mathbf{W}}, \hat{\boldsymbol{\xi}}}(\mathbf{X}_n) \right) + (1 - \mathbf{y}_n) \log \left(1 - (f_{\widehat{\mathbf{W}}, \hat{\boldsymbol{\xi}}}(\mathbf{X}_n)) \right).$$

Next, for any weight matrix from within our architecture, denoted by $\mathbf{W}$, we posit the following variational distribution, $q_{\mathbf{M}}(\mathbf{W} | \boldsymbol{\xi})$, parametrized by a matrix $\mathbf{M}$ of hyperparameters, where $\mathbf{M} = \left(m_{1,1}, m_{1,2}, \dots, m_{J(L-1), J(L)} \right)$,

$$q_{\mathbf{M}}(\mathbf{W} | \boldsymbol{\xi}) = \prod_{j,j'} \xi_{j',j} \mathcal{N}(w_{j',j}; 0, \sigma^2) + (1 - \xi_{j',j}) \mathcal{N}(w_{j',j}; m_{j',j}, \sigma^2). \quad (4.2)$$

For our selection variables, we propose the following variational distribution,

$$q(\xi_{j',j}^{(l)} = 1) = \left[1 + \exp \left\{ -2 \frac{\sum_{j'' \in (l+1)} w_{j'',j'}^2 E_q[\xi_{j'',j'}^{(l+1)}]}{\sum_{j'' \in (l+1)} w_{j'',j'}^2} - (L_j^+ - L_j^-) \right\} \right]^{-1}, \quad (4.3)$$

where $\xi_{j',j}^{(l)}$ is the binary dropout variable for the weight connecting node j in layer l to node j' in layer $l+1$ (before application of the chosen activation function), L_j^+ and L_j^- denote $\sum_n \log[p(\mathbf{y}_n | \mathbf{X}_n, \mathbf{W}, \mathbf{E}_q[\boldsymbol{\xi}_{-(j',j)}], \xi_{j',j}^{(l)} = 1)]$ and $\sum_n \log[p(\mathbf{y}_n | \mathbf{X}_n, \mathbf{W}, \mathbf{E}_q[\boldsymbol{\xi}_{-(j',j)}], \xi_{j',j}^{(l)} = 0)]$, respectively. $E_q[\xi_{j'',j'}^{(l+1)}]$ denotes the expectation over binary dropout variables in the following layer $(l+1)$, and $w_{j',j}$ denotes the weights originating at the output of the activation of the j^{th} node from layer l and going into the j' node in layer $l+1$.

The dependence on $\mathbf{W}$ may seem contradictory at first glance. However, there are no outbound weights at the last layer of the network, L , causing the first term to vanish.

As such, for any layer l , $q(\xi_{j',j}^{(l)})$ only depends on $\mathbf{W}^{(l+1)}$ which has already been updated by backpropagation independently of $\xi_{j',j}^{(l)}$. We note that the Ising prior has been applied for neural network regularization before (Salehinejad and Valaee, 2019, 2021), but to our knowledge never in a backward fashion, as all previous applications followed a forward looking scheme based on previous node activation.

The estimation of this probability directly is computationally intractable, as it requires implementing a forward pass for each proposed value of $\xi_{j',j}$, leading to a total of $2^{|\mathcal{E}|}$ required evaluations per batch. To overcome this computational bottleneck, we leverage the approximation utilized in LeCun et al. (1989) to compute saliency scores by using a Taylor expansion to approximate the log-likelihood around a previously obtained local optimum. This leads to the following approximation involving terms of the Hessian obtained from the proposed likelihood function,

$$L_j^+ - L_j^- \approx \frac{\partial^2 L(q)}{\partial w_{j',j}^2} = \frac{\partial^2 L(q)}{\partial a_{j'}^2} x_j^2. \quad (4.4)$$

This can be made even more computationally efficient using the Levenberg-Marquardt approximation to the Hessian, which requires only the entries of the gradient, where for all layers $l \neq L$,

$$\frac{\partial^2 L(q)}{\partial a_{j'}^2} = f'(a_{j'})^2 \sum_{j''} w_{j'',j'}^2 \frac{\partial^2 L(q)}{\partial a_{j''}^2} - f''(a_{j'}) \frac{\partial L(q)}{\partial x_{j'}} \quad (4.5)$$

and for the final layer, L we have the boundary condition,

$$\frac{\partial^2 L(q)}{\partial a_{j'}^2} = 2f'(a_{j'})^2 - 2(d_{j'} - x_{j'})f''(a_{j'}), \quad (4.6)$$

where $x_{j'}$ is the output of the j'^{th} neuron in layer L , $f'(a_{j'})$ is the gradient with respect to the activation function at node j' , $f''(a_{j''})$ is the diagonal component of the Hessian at the node j'' , and $d_{j'}$ is the ground truth or desired output of the node. The Levenberg-Marquardt

approximation proceeds by eliminating all $f''(\cdot)$ terms such that the quantity $L_j^+ - L_j^-$ can be efficiently approximated simultaneously with the typical backpropagation training procedure, thereby requiring no additional computational overhead whereas computing the diagonal hessian requires one additional pass per parameter. Regardless of the choice of utilizing the Levenberg-Marquardt approximation, the algorithm remains the same. The structure of our approach is summarised below in figure 4.1.

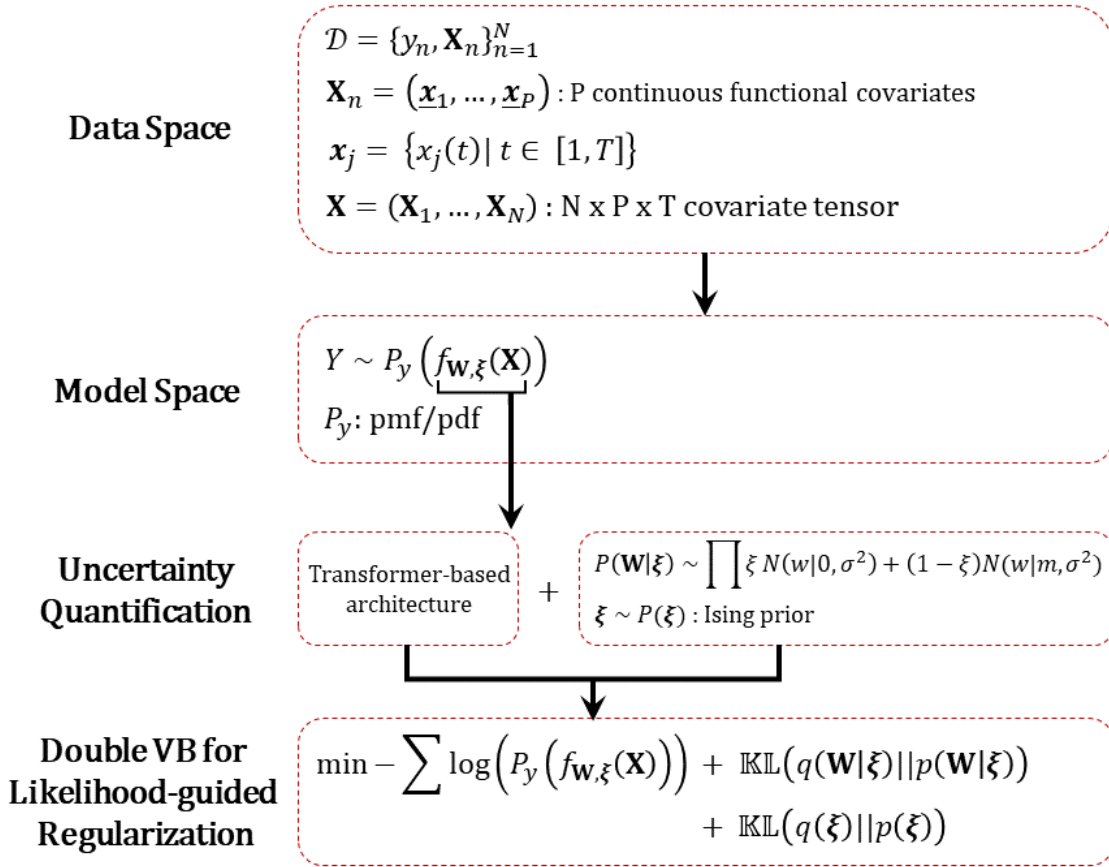


Figure 4.1: The figure shows the components in our model. The selection variables ξ determine which distribution in the spike-slab prior a weight, w , is sampled from.

4.3 Algorithm

We present the likelihood-guided regularization method fitting procedure in Algorithm 4.

Algorithm 4 Model fitting via Likelihood-guided Regularization

- 1: Given the architecture and its initializing hyperparameters
- 2: Train the model by backpropagating the negative log likelihood,

$$-\log[p(\mathbf{y}_n|\mathbf{X}_n, \widehat{\mathbf{W}}, \hat{\boldsymbol{\xi}})] = -\mathbf{y}_n \log \left(f_{\widehat{\mathbf{W}}, \hat{\boldsymbol{\xi}}}(\mathbf{X}_n) \right) - (1 - \mathbf{y}_n) \log \left(1 - (f_{\widehat{\mathbf{W}}, \hat{\boldsymbol{\xi}}}(\mathbf{X}_n)) \right)$$

onto the means, $\mathbf{M}$ of the proposed variational posterior on $\mathbf{W}$ while setting $\boldsymbol{\xi} = \mathbf{1}$ until a reasonable minimum of the loss function is obtained

- 3: For each new epoch compute the diagonal of the Hessian for each element of $\mathbf{M}$
- 4: Recover the saliency approximation for each parameter in $\mathbf{M}$ as

$$L_j^+ - L_j^- \approx \frac{\partial^2 L(q)}{\partial w_{j',j}^2},$$

- 5: Sample $\boldsymbol{\xi}$ from the proposed variational posterior

$$q(\xi_{j',j}^{(l)} = 1) = \left[1 + \exp \left\{ -2 \frac{\sum_{j'' \in (l+1)} w_{j'',j'}^2 E_q[\xi_{j'',j'}^{(l+1)}]}{\sum_{j'' \in (l+1)} w_{j'',j'}^2} - (L_j^+ - L_j^-) \right\} \right]^{-1},$$

- 6: Apply the sampled mask, do a forward pass then backpropagate the error
- 7: Sample $\mathbf{W}$ from the updated variational posterior

$$q_{\mathbf{M}}(\mathbf{W}|\boldsymbol{\xi}) = \prod_{j,j'} \xi_{j,j'} \mathcal{N}(\mathbf{W}; 0, \sigma^2) + (1 - \xi_{j,j'}) \mathcal{N}(\mathbf{W}; m_{j,j'}, \sigma^2),$$

- 8: Iterate through steps 3-6 until a reasonable minimizer is attained
 - 9: Fix the current masks, dropping the weights based on the variational posterior of $\boldsymbol{\xi}$ and train the smaller network
-

4.4 Statistical properties

Given the variational distributions $q(\mathbf{W}|\boldsymbol{\xi})$ and $q(\boldsymbol{\xi})$, we have the following statistical properties.

Proposition 4.1. *For any input $\mathbf{X}$, we have the following properties of our likelihood-guided regularization approach:*

- (a) *Computing the model output $f_{\mathbf{y}, \mathbf{W}, \boldsymbol{\xi}}(\mathbf{X})$ is equivalent to performing a forward pass on the input $\mathbf{X}$ and applying the drop with probability $\text{pr}(\xi_{j,j'} = 1)$ in likelihood-guided regularization of attention based model.*
- (b) *The posterior $p(y, \mathbf{X}) \approx \frac{1}{T} \sum_{t=1}^T f_{\mathbf{y}, \widehat{\mathbf{W}}_t, \hat{\boldsymbol{\xi}}_t}(\mathbf{X})$, where $f_{\mathbf{y}, \widehat{\mathbf{W}}_t, \hat{\boldsymbol{\xi}}_t}(\mathbf{X})$ is the output of the t^{th} forward pass through the likelihood-guided regularization approach.*
- (c) *The prediction uncertainty over the prediction $p(\mathbf{y}^*|\mathbf{X}^*)$ can be approximated by the Bayesian lower and upper credible intervals of the T forward passes.*

This proposition describes how our approach can provide uncertain quantification effectively.

Proposition 4.2. *If $\text{pr}(\xi_{j,j'} = 1) = \pi$ for all (j, j') and $q(\boldsymbol{\xi}) = O(1)$, our likelihood-guided regularization of attention based model aligns with BayesFormer(Sankararaman et al., 2022).*

This property describes our approach is a generalization of existing BayesFormer.

4.5 Simulations

We conduct a simulation study to evaluate the performance of our proposed model and to compare it against the following existing, non-neural network type methods that qualify as suitable approaches for the motivating problem: Fused Lasso Logistic Regression Yu et al. (2015), Random Forest-on-FPCA-scores, and a Support Vector Machine (with a linear kernel)-on-FPCA-scores. We also include a vanilla Multilayer Perceptron for evaluation of any performance improvements attributable to our choice of architecture. For our transformer-based architecture, we utilize the Crossformer introduced in Zhang and Yan (2023). This architecture was initially proposed for time series forecasting using multivariate time series. It includes an Encoder-Decoder structure which we believe would favorably contribute to the regularization procedure. It also has the benefit of allowing for mixing of information between multiple functional covariates at different smoothness scalings. We adapt the original implementation here for the classification problem. In addition to the non-penalized classification Crossformer, we also implement a penalized Bayesian Classification Crossformer; for evaluation of the effect of classic regularization methods. Finally, we implement our two proposed approaches for the Bayesian-Ising regularization method.

In our simulation study, we examine multiple combinations of settings for the number of parameters of the model and the sample size. For the sample size we apply the model to datasets of size $N = [20, 50, 100]$. The number of model parameters is modified by adding more layers to the Crossformer architecture and changing the number of neurons within the hidden layers of some MLP layers in the Crossformer. For our Crossformer models, we chose a unified setting across all four implementations with 4 attention heads, single encoder and decoder layers, segment length of 5, 20 routers, a hidden state dimension of 8, no segment merging, and a train-validation-test split of (0.7, 0.1, 0.2). Other settings were implemented as per the default architecture settings. Our sample sizes were chosen to be within an order of magnitude of our motivating dataset, the Autism Brain Imaging Data Exchange dataset.

We evaluate the performance of the presented methods on the classification problem using four evaluation metrics: the true positive rate (TPR), the false positive rate (FPR), the positive predictive value (PPV), and the F1 score, respectively defined as follows,

$$\mathbf{TPR} = \frac{\sum_{i=1}^N \mathbb{I}(\hat{y}_i = 1, y_i = 1)}{\sum_{i=1}^N \mathbb{I}(y_i = 1)} \quad ; \quad \mathbf{FPR} = \frac{\sum_{i=1}^N \mathbb{I}(\hat{y}_i = 1, y_i = 0)}{\sum_{i=1}^N \mathbb{I}(y_i = 0)}.$$

$$\mathbf{PPV} = \frac{\sum_{i=1}^N \mathbb{I}(\hat{y}_i = 1, y_i = 1)}{\sum_{i=1}^N \mathbb{I}(\hat{y}_i = 1)} \quad ; \quad \mathbf{F}_1 = \frac{2PPV \times TPR}{PPV + TPR}$$

4.5.1 Setting

We consider $P = 20$, which is on the same order of magnitude as our motivating dataset, and $|P^*| = 3$, the true number of relevant variables contributing to the response. Each $\mathbf{x}_i^{(p)}$ represents the observed p^{th} functional covariate, for the i^{th} individual, we generate $T = 100$ timepoints for a total of 2000 covariates.

Functional variables

To generate our functional covariates, we fix the three indices from 2, 5, 7 to be the indices for nodes relevant to the generation of the binary response. We generate each of our functional covariates over 100 time points. The relevant time points are always set over two regions, from $t = 21$ to $t = 40$ and $t = 61$ to $t = 80$. We then generate two functional covariate prototypes only for the relevant domains of the three chosen indices, while for the nodes indexed by the remaining indices – which we consider irrelevant with respect to the response – only hold a single functional prototype.

For our functional covariate prototypes, $\mathbf{X}_1^{(p \in P^*)}$ and $\mathbf{X}_2^{(p \in P^*)}$, we generate each from a modified Weiner process such that

$$\mathbf{x}_{1,2}(0) \sim MVN(\mathbf{0}, \mathbf{C}_P) \quad , \quad \mathbf{x}_{1,2}(t) \sim MVN(\mathbf{x}_{1,2}(t-1), \mathbf{C}_P) \quad (4.7)$$

where $\underline{\mathbf{x}}(t)$ is a vector of covariate observations across all nodes P at a single time point t and $\mathbf{C}_P$ is a $T \times T$ matrix representing the desired connectedness between the nodes such that the $(t, t')^{th}$ element of $\mathbf{C}_P(t, t')$ is

$$c_p(t, t') = \tau_u \exp \left[- \frac{||t - t'||}{\rho_u} \right],$$

where τ_u is a tuning hyperparameter responsible for the overall scale of the generated data; likewise, ρ_u is a lengthscale parameter that controls the degree of locality of connection between time points. We use $\mathbf{C}_P$ to induce correlation between functional covariates which are to be considered neighbors.

Finally, to generate $\underline{\mathbf{X}}_1^{(p)}$, we sample from $Bern(0.5)$, representing the response label. Each observation is assigned one of the functional prototypes based on its sampled Bernoulli value plus a vector of independent Gaussian noise, such that if we desire to assign an individual, i , to the first functional prototype, we have

$$\mathbf{X}_i^{(p \in P^*)} = \mathbf{X}_1^{(P)} + \underline{\mathbf{u}} \quad , \quad \underline{\mathbf{u}} \sim MVN(\mathbf{0}, \sigma_u^2 \mathbf{I}); \quad i = 3, \dots, N, \quad (4.8)$$

where σ_u^2 is a tuning parameter controlling the deviation of an individual's functional covariate from its underlying functional prototype. In our simulation, we set $\tau_u = 2$ and $\sigma_u^2 = 4$, this induces high levels of noise in the covariates and allows for significant overlap between the functional covariates. Therefore we have,

$$\underline{\mathbf{y}} = \mathbb{I}\{\mathbf{K}(\mathbf{X}^*)\underline{\boldsymbol{\alpha}} \geq 0\} \quad , \quad t \in [21, 40] \cup [61, 80],$$

where $\mathbb{I}\{\cdot\}$ is an indicator function and $\mathbf{X}^* = [\mathbf{X}^{(2)}, \mathbf{X}^{(5)}, \mathbf{X}^{(7)}]$ are the functional covariates generated at the three regions for all N individuals across the time interval.

4.5.2 Simulation Result

Simulation results are summarized in Tables 4.1- 4.3. We randomly generate 100 datasets for each setting of N , using the process described in Section 4.5.1. Tables 4.1, 4.2, and 4.3 contain reference results for $N = 20, 50, 100$, respectively. Results displayed in the tables represent the average performance across all 100 generated datasets for each setting of N .

We observe that for smaller sample sizes Table 4.1, and Table 4.2, our two methods (both approximate and full implementation) tend to outperform the original non-Bayesian architecture, indicated by the “Crossformer” row, on all metrics for the approximate approach and on PPV and FPR for the exact approach. They also outperform the original architecture when the linear layers are replaced with their Bayesian counterparts on all metrics, as shown in the “Bayesian Crossformer” row, indicating that our variational posterior on the dropout probabilities has an additional impact over the regularizing effect of a typical, fixed dropout probability Bayesian layer. Differences between the approximate and fully implemented methods are minor and, in general, represent a small improvement over their competitors. The benefits from our proposed regularization approach tend to dilute as the sample size increases, which we observe for $N = 100$ in Table 4.3 with all models yielding virtually similar results, except for the “Bayesian Crossformer”; being the only model that is forced to a fixed dropout probability regardless of sample size. This dilution of the importance of regularization is a common feature of any Bayesian approach. We displayed all of our simulation results in Figure 4.2.

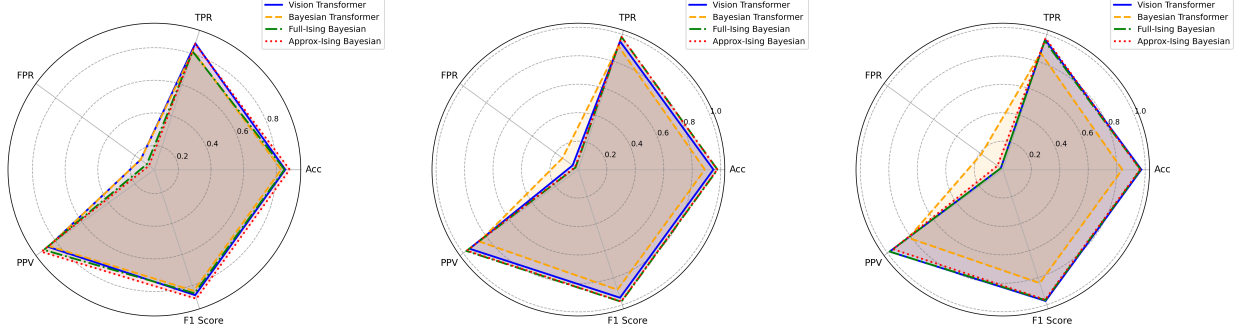


Figure 4.2: Simulation results to investigate the performance of different model architectures under $N=20, 50, 100$ and $P=20$ using **ACC**, **TPR**, **FPR**, **PPV**, **F1**. Crossformer (Zhang and Yan, 2023); Bayesian Crossformer (Sankararaman et al., 2022); Full-Ising Bayesian Crossformer and Approximate-Ising Bayesian Crossformer are our methods.

For each setting of N we also show the performance of a vanilla MLP, to account for any performance improvements resulting from our choice of architecture. The MLP performs worse on every metric than any implemented transformer architectures. Despite the performance gains from the additional regularization, however, less flexible models such as the Fused Lasso Logistic Regression and the Linear Kernel Support Vector Machine still outperform their more flexible competitors as has been previously observed in many cases in existing literature. Random Forests, being a highly flexible non-parametric model, tends to overfit on smaller sample sizes, observed in Table 4.1, but tends to do well once sufficient data is provided (see Table 4.3).

Tracking the negative log-likelihood over the validation and testing sets further confirms the regularizing effect of our approach. We see that the full implementation of our regularizer leads to a marked decrease in the negative log-likelihood for the validation and testing sets as displayed in Figures 4.4 and 4.6; this effect weakens as the sample size increases until it disappears completely in Figures 4.5 and 4.7. Epistemic uncertainty also declines steadily under our approach, while the “Bayesian Crossformer” suffers from fixed uncertainty even as the number of epochs goes to infinity. This decrease in the value of the negative log-likelihood has not manifested greatly in terms of classification performance; instead, it leads

to more confident probability estimates on the output labels in the form of more excellent separability of the resulting logits.

Expectedly, this decline in negative log-likelihood is absent when examining performance on the training set (see Figure 4.3), indicating a ceiling on the generalizability of the non-regularized models to the validation and testing data.

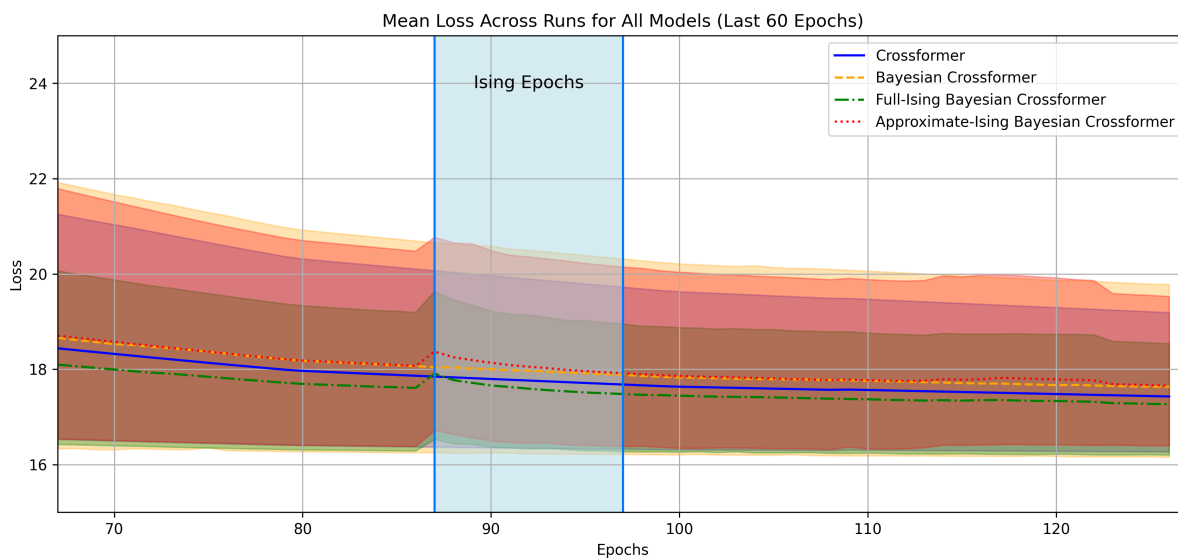


Figure 4.3: Negative log-likelihood on the training set. $N=20$. We observe a bump at the starting epoch of the Ising implementation indicating the disruption of training loss due to the implementation of a new mask. the training loss smooths out as the weights are optimized over the next few epochs.

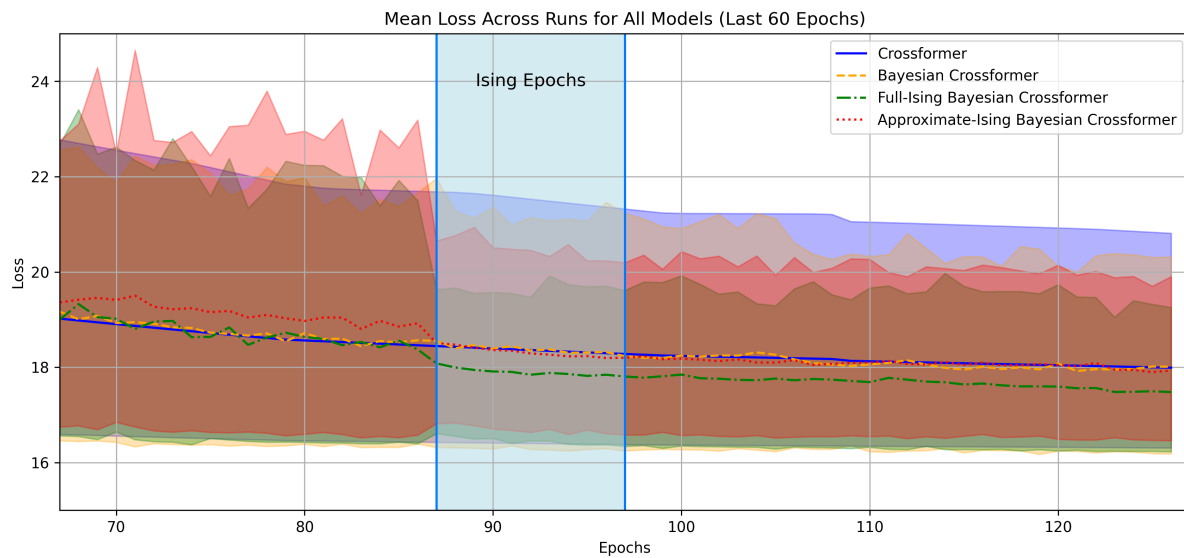


Figure 4.4: Negative log-likelihood on the validation set. $N=20$. We observe a clear sharp drop in validation loss in the Ising epochs due to greater regularization leading to better generalizability of the trained weights to the validation dataset.

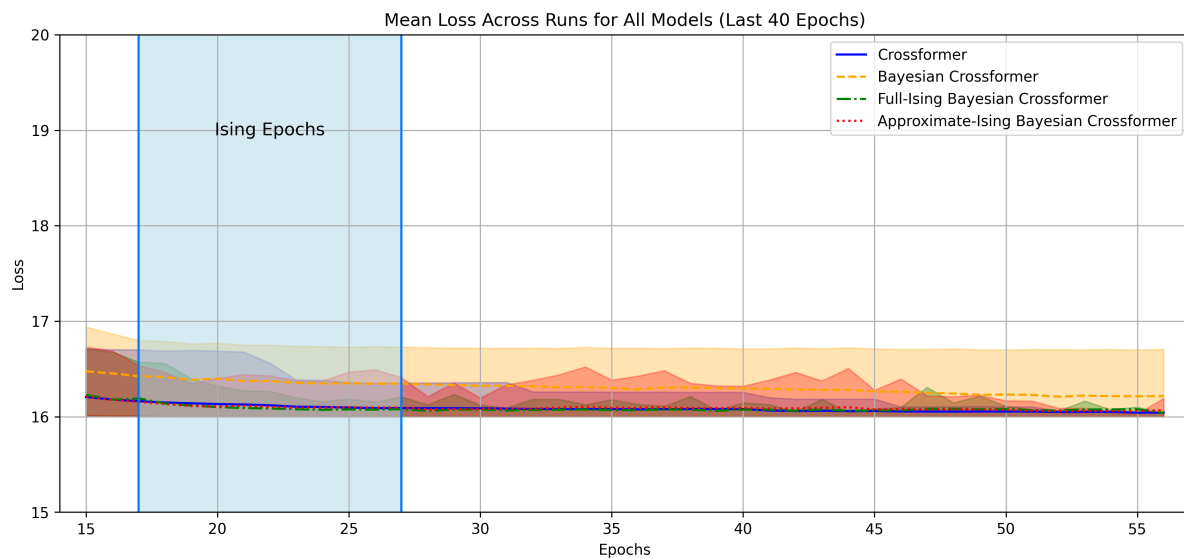


Figure 4.5: Negative log-likelihood on the validation set. $N=100$. There is virtually no difference in validation loss as the larger dataset size negates the benefits of additional regularization.

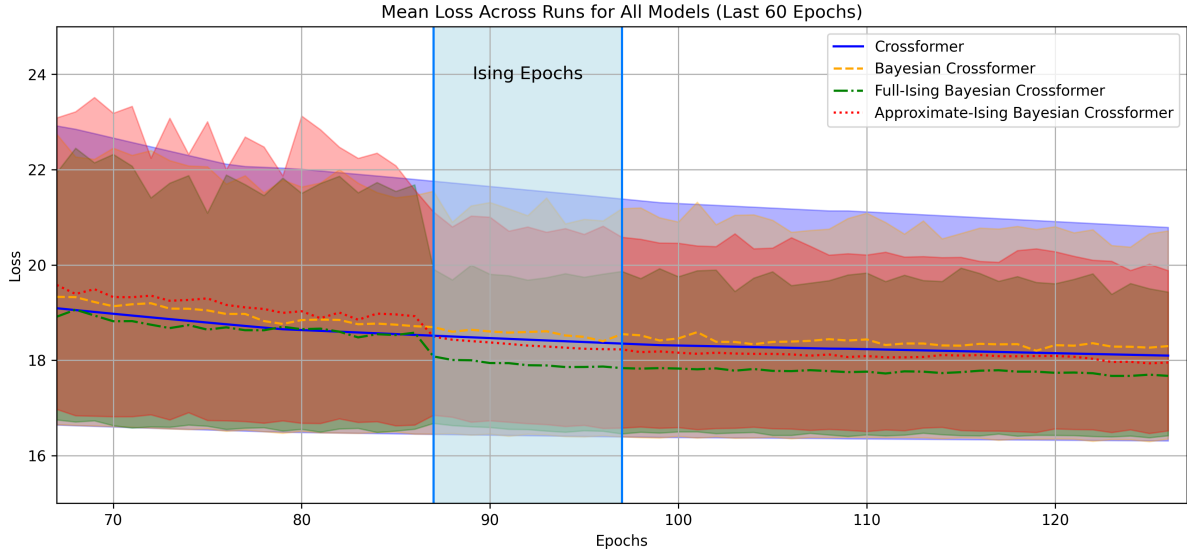


Figure 4.6: Negative log-likelihood on the testing set. $N=20$. We observe a clear sharp drop in testing loss in the Ising epochs due to greater regularization leading to better generalizability of the trained weights to the testing dataset.

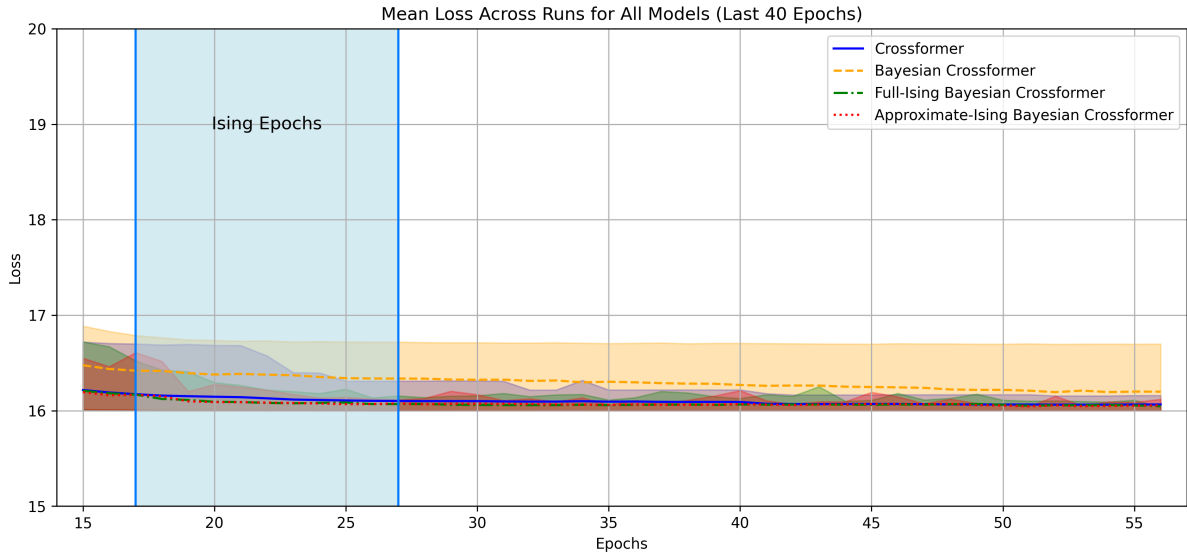


Figure 4.7: Negative log-likelihood for the first few epochs on the testing set. $N=100$. There is virtually no difference in test loss as the larger dataset size negates the benefits of additional regularization.

(N,p)	Method	ACC	TPR	FPR	PPV	F ₁
(20,20)	Crossformer	0.855	0.866	0.157	0.858	0.862
	Bayesian Crossformer	0.830	0.818	0.157	0.851	0.834
	Full-Ising Bayesian Crossformer	0.850	0.813	0.110	0.890	0.850
	Approximate-Ising Bayesian Crossformer	0.882	0.862	0.094	0.909	0.885
(20,20)	Fused Lasso Log.Reg	0.980	0.961	0.000	1.000	0.980
	Random Forests	0.688	0.484	0.089	0.856	0.618
	Support Vector Machine	1.000	1.000	0.000	1.000	1.000
	Multilayer Perceptron	0.693	0.952	0.592	0.638	0.764

Table 4.1: Simulation results to investigate the performance of different model architectures under N=20 and P=20 using **ACC**, **TPR**, **FPR**, **PPV**, **F₁**; Crossformer (Zhang and Yan, 2023); Bayesian Crossformer (Sankararaman et al., 2022); Full-Ising Bayesian Crossformer and Approximate-Ising Bayesian Crossformer are our methods.

(N,p)	Method	ACC	TPR	FPR	PPV	F ₁
(50,20)	Crossformer	0.949	0.948	0.050	0.946	0.947
	Bayesian Crossformer	0.889	0.912	0.138	0.860	0.888
	Full-Ising Bayesian Crossformer	0.978	0.981	0.025	0.973	0.977
	Approximate-Ising Bayesian Crossformer	0.976	0.981	0.029	0.969	0.975
(50,20)	Fused Lasso Log.Reg	0.992	0.986	0.002	0.998	0.992
	Random Forests	0.981	0.961	0.000	1.000	0.980
	Support Vector Machine	1.000	1.000	0.000	1.000	1.000
	Multilayer Perceptron	0.728	0.964	0.488	0.645	0.773

Table 4.2: Simulation results to investigate the performance of different model architectures under N=50 and P=20 using **ACC**, **TPR**, **FPR**, **PPV**, **F₁**; Crossformer (Zhang and Yan, 2023); Bayesian Crossformer (Sankararaman et al., 2022); Full-Ising Bayesian Crossformer and Approximate-Ising Bayesian Crossformer are our methods.

(N,p)	Method	ACC	TPR	FPR	PPV	F ₁
(100,20)	Crossformer	0.973	0.962	0.017	0.982	0.972
	Bayesian Crossformer	0.839	0.863	0.186	0.818	0.840
	Full-Ising Bayesian Crossformer	0.968	0.952	0.017	0.982	0.967
	Approximate-Ising Bayesian Crossformer	0.964	0.973	0.046	0.953	0.963
(100,20)	Fused Lasso Log.Reg	0.998	0.995	0.001	0.999	0.997
	Random Forests	1.000	1.000	0.000	1.000	1.000
	Support Vector Machine	1.000	1.000	0.000	1.000	1.000
	Multilayer Perceptron	0.794	0.984	0.391	0.710	0.825

Table 4.3: Simulation results to investigate the performance of different model architectures under $N=100$ and $P=20$ using **ACC, TPR, FPR, PPV, F₁**; Crossformer (Zhang and Yan, 2023); Bayesian Crossformer (Sankararaman et al., 2022); Full-Ising Bayesian Crossformer and Approximate-Ising Bayesian Crossformer are our methods.

4.6 Autism Brain Imaging Data Exchange

Our problem of interest is classifying autism spectrum disorder motivated by the autism brain imaging data exchange (ABIDE) dataset. This data contains functional magnetic resonance imaging (fMRI) as a BOLD signal time series (a series of observations of a patient’s Blood Oxygenation level-dependent signals over time). In particular, we use time series data from ABIDE I and II to classify patients versus controls. The ABIDE dataset is a publicly available dataset consisting of two repositories (ABIDE I and ABIDE II). The full dataset was aggregated from data collected across multiple testing sites in Europe and North America, leading to some differences in the data collection procedure (e.g., different sampling rates). The data can be preprocessed using several methods available directly at the host source. The version we use here combines observations from both repositories with an overall sample size of $N = 1,036$, with data observed from several brain regions. The time series data is extracted from BOLD signals observed in pre-specified brain regions. There exist multiple

atlases for defining a brain region; the one we use here is the Schaefer100 parcellation Schaefer et al. (2017), which maps the brain to 7 networks, each consisting of multiple regions. The ABIDE dataset also includes Structural MRI information as well as Phenotypic information about the observed individuals; however, our analysis focuses only on the fMRI time series output signals.

4.6.1 Preprocessing data

The preprocessing procedure on our version of the ABIDE dataset includes 0.0010.08 Hz band-pass filtering, cutting out of the first ten volumes, motion correction, 5-mm smoothing, slice-timing correction, grand-mean intensity normalization, regressing out CSF/WM nuisance signal (individual segmented masks, with a 90% threshold), and ICA denoising and registration to standard MNI 2-mm EPI brain template (Calhoun et al., 2017). Furthermore, we restrict our working dataset to data collected at a sampling frequency of 2 Time Repetitions/sec so that we do not need imputation via upsampling and/or downsampling. Also, this sampling frequency of data contains the largest number of observations. The sample size of this reduced dataset is $N = 596$ with 266 patients and 330 controls.

We conduct data analysis on a single brain network, “the Default Mode network”, which has been identified in previous work as a promising marker-producing region for ASD classification. In our chosen atlas, the Default Mode network consists of $P = 24$ parcellations, each producing a BOLD signal time series. Additionally, we implement a version that includes all 7 networks by averaging across parcellations within a network and utilizing the data for all networks simultaneously.

4.6.2 Small dataset analysis

First, we conducted a dataset with $N = 596$. We found that results from both settings, the single network and the averaged parcellations from all networks, fail to approach the

current state of the art for time series-based transformers established. To hedge against potential implementation errors and model-specific limitations, we also implement several other models, including Fused Lasso Logistic Regression, SVM with a linear kernel, Random Forests, and a vanilla MLP with a hidden layer structure of 24, 15, 10, 15, 24. All models produce similar results with accuracies for all models reaching at most 59% across multiple runs, with high variability.

We note that successful undertakings of the classification problem using the ABIDE dataset have involved the introduction of domain knowledge information in one of several ways. For instance, Mahler et al. (2023) uses a combination of pretraining and introduction of predefined brain atlases into the model that inject knowledge on the form of brain atlases to achieve state-of-the-art accuracy at 83.70%, while Li et al. (2022) (74.18%) introduce predefined functional connectivity matrices into the model in lieu of positional decoder embeddings. In terms of transformer, models applied only to the time series version of the data, Deng et al. (2022) seems to be the best-performing model with 73% and 72% accuracies achieved on ABIDE I and ABIDE II, respectively. For our application, we have opted to rely solely on the model as a way of testing the architecture’s advertised cross-time and cross-position inference ability. However, the data seems to be generated by heterogeneous mechanisms that the model alone is incapable of adequately capturing.

Second, given the difficulties the Crossformer architecture has experienced, we restrict our focus to a smaller subset of the data that also has recorded autism diagnostic observation schedule (ADOS) scores. This data subset has shown significant separability between both classes based on fMRI time series alone and may represent a more homogeneous underlying data-generating process. The smaller dataset consists of 183 observations with 173 patients and 10 controls. For the resulting covariates, we also reduce the dimensionality of the time series data by applying functional principal component analysis (FPCA) and extracting the first 10 FPC scores for use as input into our models. To overcome the class imbalance,

we weight our loss function using the inverse class frequencies in the training data. While training our model, we hold 20% of the data for validation, a single observation for testing, and train the model on the remaining data. We consider 3 different data structures, the first with a sample size of 20, which includes all 10 available controls and 10 randomly selected patients, the second and third consist of unbalanced datasets of size 50 and 100 where we keep the 10 controls and sample the remainder of the dataset randomly from the patient population. We also discard the decoder portion of the Crossformer, leading to an architecture that is closer to a classic classification vision transformer. The classification results are summarized in Tables 4.4-4.5.

(N,p)	Method	ACC	TPR	FPR	PPV	F ₁
(20,24)	Vision transformer	0.900	0.900	0.100	0.900	0.900
	Bayesian transformer	0.900	0.900	0.100	0.900	0.900
	Full-Ising Bayesian transformer	0.900	0.900	0.100	0.900	0.900
	Approximate-Ising Bayesian transformer	0.950	0.900	0.000	1.000	0.947
(50,24)	Vision transformer	0.820	0.900	0.500	0.878	0.889
	Bayesian transformer	0.920	0.925	0.100	0.974	0.949
	Full-Ising Bayesian transformer	0.840	0.875	0.300	0.921	0.897
	Approximate-Ising Bayesian transformer	0.820	0.875	0.400	0.897	0.886
(100,24)	Vision Transformer	0.910	0.911	0.100	0.988	0.948
	Bayesian transformer	0.880	0.912	0.400	0.953	0.932
	Full-Ising Bayesian transformer	0.890	0.922	0.400	0.954	0.938
	Approximate-Ising Bayesian transformer	0.890	0.933	0.500	0.944	0.939

Table 4.4: Real data results of classification performance using different model architectures under N=20,50,100 and P=24 parcellations within the Default Mode network of the brain as identified by Schaefer et al. (2017); Vision transformer (Dosovitskiy et al., 2020); Bayesian transformer (Sankararaman et al., 2022); Full-Ising Bayesian transformer and Approximate-Ising Bayesian transformer are our methods.

(N,p)	Method	ACC	TPR	FPR	PPV	F ₁	# Excl.
(20,24)	Vision transformer	0.900	0.900	0.100	0.900	0.900	0
	Bayesian transformer	0.895	0.900	0.111	0.900	0.900	1
	Full-Ising Bayesian transformer	0.895	0.889	0.100	0.889	0.889	1
	Approximate-Ising Bayesian transformer	0.947	0.900	0.000	1.000	0.947	1
(50,24)	Vision transformer	0.851	0.897	0.375	0.921	0.909	3
	Bayesian transformer	0.920	0.925	0.100	0.974	0.949	0
	Full-Ising Bayesian transformer	0.860	0.868	0.200	0.971	0.917	7
	Approximate-Ising Bayesian transformer	0.886	0.944	0.375	0.919	0.932	6
(100,24)	Vision transformer	0.919	0.921	0.100	0.988	0.953	1
	Bayesian transformer	0.888	0.920	0.400	0.953	0.936	2
	Full-Ising Bayesian transformer	0.917	0.949	0.600	0.962	0.955	16
	Approximate-Ising Bayesian transformer	0.955	0.976	0.400	0.976	0.976	11

Table 4.5: Real data results of classification performance of different model architectures under $N=20,50,100$ and $P=24$ parcelations within the Default Mode network of the brain as identified by Schaefer et al. (2017); Vision transformer (Dosovitskiy et al., 2020); Bayesian transformer (Sankararaman et al., 2022); Full-Ising Bayesian transformer and Approximate-Ising Bayesian transformer are our methods; The final column indicates the number of observations excluded due to uncertainty in classification, defined as a probability between 0.45 and 0.55

As shown in Table 4.4, across all N , we observe that our methods show little variability in performance, with differences that could easily arise due to the random nature of the dropout implemented in all versions. This is in line with our simulation results in Section 4.5.2, which indicated only minor performance improvements. The greatest differences we see, however, are in the output probabilities by the model, where we clearly observe an overconfidence problem in both the Bayesian and non-Bayesian versions of the utilized transformer architectures, indicating poorly calibrated probability estimates. For both versions, the approximate and exact implementation of our regularization approach, the models produce far less extreme probabilities on the misclassified cases while maintaining similar performance levels. In Table 4.5, we show performance metrics when we introduce a third

“uncertain” classification based on the model probabilities. Real data results are visualized in 4.8 for the unamended probabilities, and in figure 4.9 with uncertain classifications removed. In addition, Figure 4.10 shows the probabilities output by the models for an observation that was misclassified by all 4 models under the $N = 20$ setting. We clearly observe that the models without our proposed regularization produce worse probability calibration under small sample sizes.

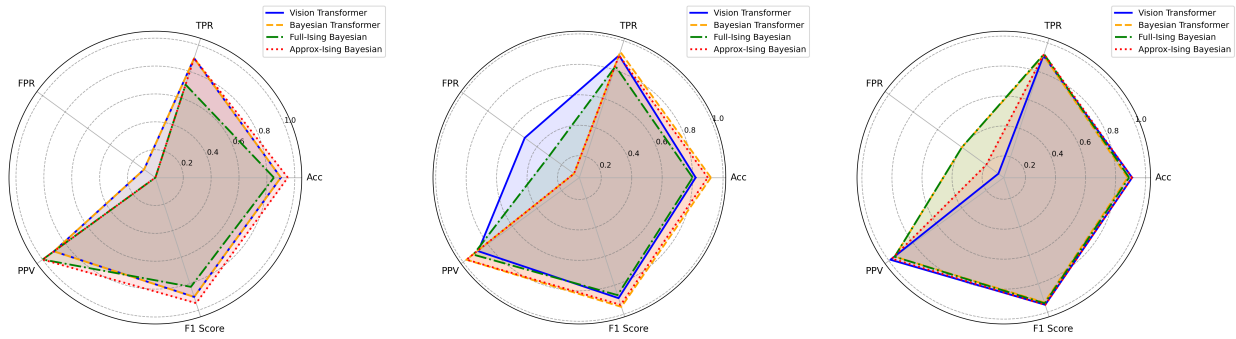


Figure 4.8: Real data results of classification performance using different model architectures under $N=20, 50, 100$ and $P=24$ parcellations within the Default Mode network of the brain as identified by Schaefer et al. (2017). Crossformer (Zhang and Yan, 2023); Bayesian Crossformer (Sankararaman et al., 2022); Full-Ising Bayesian Crossformer and Approximate-Ising Bayesian Crossformer are our methods.

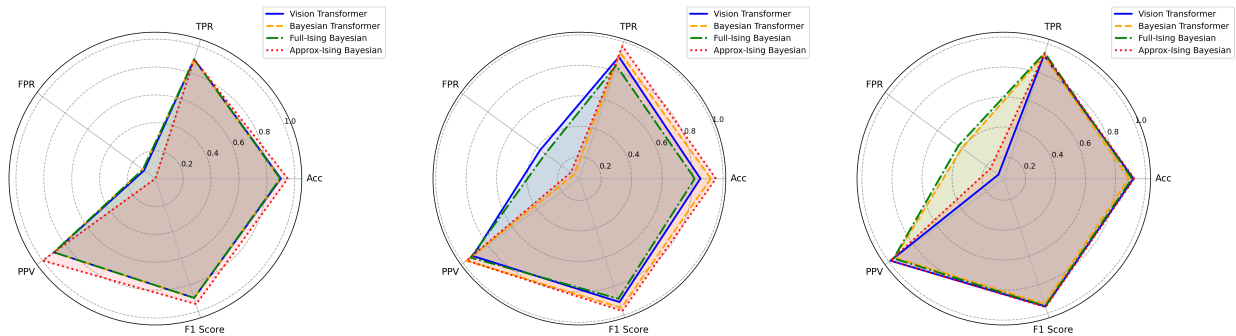


Figure 4.9: Real data results of classification performance of different model architectures under $N=20, 50, 100$ and $P=24$ parcellations within the Default Mode network of the brain as identified by Schaefer et al. (2017). Crossformer (Zhang and Yan, 2023); Bayesian Crossformer (Sankararaman et al., 2022); Full-Ising Bayesian Crossformer and Approximate-Ising Bayesian Crossformer are our methods. The final column indicates the number of observations excluded due to uncertainty in classification, defined as a probability between 0.45 and 0.55.

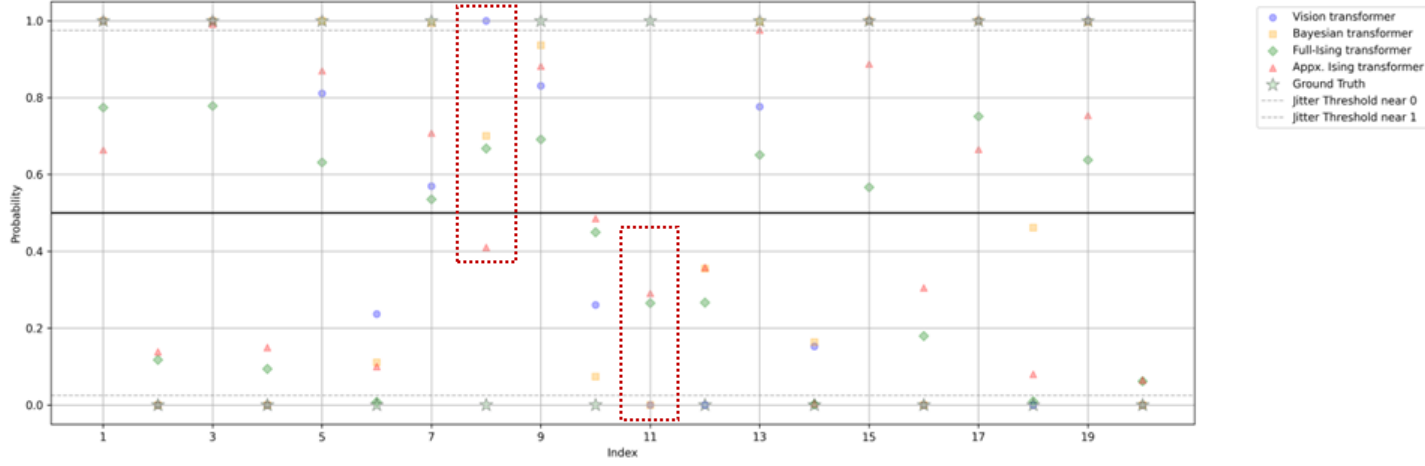


Figure 4.10: Model probability estimates for $N=20$. The red rectangular boxes highlight misclassified observations. We observe that the typical Vision transformer assigns almost perfect probabilities even to misclassified observations, the Bayesian transformer produces close results, implying the models are overconfident and are producing poorly calibrated probability estimates. Both versions of our proposed regularization show more uncertainty. The approximate method, being the stronger regularizer shows even more uncertainty.

4.7 Conclusion and discussion

In this paper, we develop a new and general regularization approach applicable to attention-based transformer architectures. Our approach is completely guided by the model likelihood, allowing us to provide uncertainty quantification. Our proposed likelihood-guided regularization offers improvements in performance, particularly for smaller datasets where models generally tend to overfit. Unlike typical Bayesian transformers, our approach does not require setting a dropout hyperparameter; instead, the dropout probability is inferred directly from the data in a computationally efficient manner. In addition, the advantage of using transformer-based architectures is that it allows the approach to be applied to all covariate modalities. We note that although our focus in this paper is on a classification task using functional covariates, our method generalizes directly to regression tasks as well.

We apply our approach to the ABIDE dataset using the BOLD signal time series as our input covariates. Our chosen architecture could not achieve current state-of-the-art classification levels without pretraining, transfer learning, or direct input of functional connectivity

information. However, we show that for a smaller subset of the data, our likelihood-guided regularization leads to improvements in model performance alongside the production of uncertainty estimates for the output. We believe that investigating the effects of our proposed regularization on models that utilize transfer learning, functional connectivity, and pre-training could provide valuable insights and enhance performance, particularly with smaller datasets.

Chapter 5

Conclusions and Discussion

In this dissertation, we have explored the application of Bayesian methods to high-dimensional data of different modalities; in particular, we have proposed effective new methods for analyzing high-dimensional, structured data, focusing on multilevel variable selection, network estimation, and functional data classification; all following a Bayesian approach, which deals with inference in both high-dimensional scalar-on-scalar regression, scalar-on-functional regression, and scalar-on-function classification. These methods offer a transparent and systematic way to incorporate prior domain knowledge into the proposed models, thus improving their performance.

In the second chapter, we introduced a Bayesian multilevel network recovery method, combining McMC and variational Bayes inference to jointly select multilevel variables and construct networks under a non-additive, non-parametric framework. Its novel multilevel Ising spike-slab prior, utilizing the sampled set structure, allows for efficient computational performance while producing both robust mean and quantile estimations. The multi-kernel approach also allows for separate non-parametric modeling of unique components within a dataset. The approach demonstrates superiority for the simultaneous regression and variable selection problem given a univariate response with a data generating process consisting of interacting sets of multilevel, correlated covariates. Future research may further extend this approach to multivariate response settings.

The third chapter builds on our previous work with correlated data structures to propose a Bayesian multi-tasking inference framework for functional data under a joint semiparametric functional kernel graphical model framework. By integrating a semiparametric

kernel graphical model with Bayesian optimization and Ising graphical models, the method achieves simultaneous domain subset selection, variable selection, and network estimation. Furthermore, the chosen modeling structure allows for the utilization of score-type tests for significance of the non-parametric components of the model in relation to the proposed response variable. This approach was shown to be particularly effective for non-parametric data-generating processes, while avoiding the need for basis functions and offering flexible extensions for heterogeneous covariates and populations. The approach introduced here is straightforward to adapt to heterogeneous sets of functional covariates requiring different model settings; the model can also accommodate finite mixtures of kernel-based models. We also note the introduction of functional connectivity domain knowledge via priors on selection variables. The approach proposed here is equivalent to introducing connectivity knowledge via the lengthscales parameters within the nonparametric portion of our model, given a Gaussian, or any other suitably similar choice of kernel. Introducing the connectivity via a single prior on the lengthscales parameters may offer greater efficiency which warrants consideration for further investigation.

The fourth chapter explores the use of transformer architectures for classification using functional covariates in a high-dimensional setting. The work shown there focuses on likelihood-guided regularization to improve performance under small datasets. A new and general regularization approach, completely guided by the model likelihood and applicable to all attention-based architectures, is proposed; offering improvements in performance, particularly for smaller datasets where models generally tend to overfit, without the need for hyperparameter optimization. While the utilized architecture does not yet achieve state-of-the-art results without additional pretraining or functional connectivity inputs, the approach demonstrates promise for future integration with advanced techniques like transfer learning and functional-connectivity-based positional embeddings.

Together, these chapters explore and emphasize the value of Bayesian methodologies

and their associated regularization effects in addressing the challenges of high-dimensional, structured data. By extending traditional approaches to account for known dependencies, this work invites further exploration into key areas such as the optimal penalty structure for multi-tiered datasets and model structures, effective summary statistics for functional covariates, and computationally feasible extensions of uncertainty quantification to over-parametrized model architectures, as applied to the analysis of high-dimensional data across diverse fields such as genetics, neuroscience, and machine learning.

Bibliography

- E. A. Allen, E. Damaraju, S. M. Plis, E. B. Erhardt, T. Eichele, and V. D. Calhoun. Tracking Whole-Brain Connectivity Dynamics in the Resting State. *Cerebral Cortex*, 24(3):663–676, 11 2012. ISSN 1047-3211. doi: 10.1093/cercor/bhs352. URL <https://doi.org/10.1093/cercor/bhs352>.
- R. Bai, G. E. Moran, J. L. Antonelli, Y. Chen, and M. R. Boland. Spike-and-slab group lassos for grouped regression and sparse generalized additive models. *Journal of the American Statistical Association*, 117(537):184–197, jun 2020. doi: 10.1080/01621459.2020.1765784. URL <https://doi.org/10.1080%2F01621459.2020.1765784>.
- H. Bedel, I. Şivgin, O. Dalmaz, S. U. H. Dar, and T. Cukur. Bolt: Fused window transformers for fmri time series analysis. 05 2022. doi: 10.48550/arXiv.2205.11578.
- J. O. Berger, V. de Oliveira, and B. Sansó. Objective bayesian analysis of spatially correlated data. *Journal of the American Statistical Association*, 96(456):1361–1374, 2001. ISSN 01621459. URL <http://www.jstor.org/stable/3085905>.
- C. M. Bishop. Pattern recognition and machine learning. *Springer google schola*, 2:5–43, 2006.
- C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.
- J. F. Bobb, L. Valeri, B. Claus Henn, D. C. Christiani, R. O. Wright, M. Mazumdar, J. J. Godleski, and B. A. Coull. Bayesian kernel machine regression for estimating the health effects of multi-pollutant mixtures. *Biostatistics*, 16(3):493–508, 12 2014. ISSN 1465-4644.

- doi: 10.1093/biostatistics/kxu058. URL <https://doi.org/10.1093/biostatistics/kxu058>.
- D. A. C. Boya Zhang and R. B. Gramacy. Distance-distributed design for gaussian process surrogates. *Technometrics*, 63(1):40–52, 2021. doi: 10.1080/00401706.2019.1677269. URL <https://doi.org/10.1080/00401706.2019.1677269>.
- S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- E. Brochu, V. M. Cora, and N. de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *ArXiv*, abs/1012.2599, 2010. URL <https://api.semanticscholar.org/CorpusID:1640103>.
- V. D. Calhoun, T. D. Wager, A. Krishnan, K. S. Rosch, K. E. Seymour, M. B. Nebel, S. H. Mostofsky, P. Nyalakanai, and K. Kiehl. The impact of t1 versus epi spatial normalization templates for fmri data analyses. *Human Brain Mapping*, 38(11):5331–5342, 2017. doi: <https://doi.org/10.1002/hbm.23737>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.23737>.
- R.-B. Chen, C.-H. Chu, S. Yuan, and Y. N. Wu. Bayesian sparse group selection. *Journal of Computational and Graphical Statistics*, 25(3):665–683, 2016. doi: 10.1080/10618600.2015.1041636. URL <https://doi.org/10.1080/10618600.2015.1041636>.
- Y.-Y. Chen, M. Uljarevic, J. Neal, S. Greening, H. Yim, and T.-H. Lee. Excessive functional coupling with less variability between salience and default mode networks in autism spectrum disorder. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, 7(9):876–884, 2022.
- Z.-w. Chen, L.-y. Chen, H.-l. Dai, J.-h. Chen, and L.-z. Fang. Relationship between alanine aminotransferase levels and metabolic syndrome in nonalcoholic fatty liver disease. *Journal*

- of Zhejiang University. Science. B*, 9(8):616–622, Aug. 2008. ISSN 1673-1581. doi: 10.1631/jzus.B0720016. Place: China.
- F. C. I. H. Chih-Li Sung, Wenjia Wang and Y. Hungc. Functional-input gaussian processes with applications to inverse scattering problems. *Statistica Sinica*, 2024. URL [arXiv: 2201.01682](https://arxiv.org/abs/2201.01682).
- B. Dai, S. Ding, and G. Wahba. Multivariate bernoulli distribution. *Bernoulli*, 19(4), Sep 2013. ISSN 1350-7265. doi: 10.3150/12-bejsp10. URL <http://dx.doi.org/10.3150/12-BEJSP10>.
- R. B. Davies. Algorithm as 155: The distribution of a linear combination of 2 random variables. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29(3):323–333, 1980. ISSN 00359254, 14679876. URL <http://www.jstor.org/stable/2346911>.
- X. Deng, J. Zhang, R. Liu, and K. Liu. Classifying asd based on time-series fmri using spatial-temporal transformer. *Computers in Biology and Medicine*, 151:106320, 2022. ISSN 0010-4825. doi: <https://doi.org/10.1016/j.combiomed.2022.106320>. URL <https://www.sciencedirect.com/science/article/pii/S0010482522010289>.
- A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL <https://arxiv.org/abs/2010.11929>.
- Z. Fang and I. Kim. Bayesian ising graphical model for variable selection. *Journal of Computational and Graphical Statistics*, 25(2):589–605, 2016. doi: 10.1080/10618600.2015.1035438. URL <https://doi.org/10.1080/10618600.2015.1035438>.
- Z. Fang, I. Kim, and P. Schaumont. Flexible variable selection for recovering sparsity in

- nonadditive nonparametric models. *Biometrics*, 72(4):1155–1163, 2016. ISSN 0006341X, 15410420. URL <http://www.jstor.org/stable/44695333>.
- Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL <https://proceedings.mlr.press/v48/gal16.html>.
- J. Gao, M. Chen, Y. Li, Y. Gao, Y. Li, S. Cai, and J. Wang. Multisite Autism Spectrum Disorder Classification Using Convolutional Neural Network Classifier and Individual Morphological Brain Networks. *Frontiers in neuroscience*, 14:629630, 2020. ISSN 1662-4548 1662-453X. doi: 10.3389/fnins.2020.629630. Place: Switzerland.
- E. I. George and R. E. McCulloch. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993. ISSN 01621459. URL <http://www.jstor.org/stable/2290777>.
- R. B. Gramacy. *Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences*. Chapman Hall/CRC, Boca Raton, Florida, 2020. <http://bobby.gramacy.com/surrogates/>.
- R. B. Gramacy and H. K. Lee. Gaussian processes and limiting linear models. *Computational Statistics Data Analysis*, 53(1):123–136, 2008a. ISSN 0167-9473. doi: <https://doi.org/10.1016/j.csda.2008.06.020>. URL <https://www.sciencedirect.com/science/article/pii/S0167947308003307>.
- R. B. Gramacy and H. K. H. Lee. Bayesian Treed Gaussian Process Models with an Application to Computer Modeling. *Journal of the American Statistical Association*, 103(483):

- 1119–1130, 2008b. ISSN 01621459. URL <http://www.jstor.org/stable/27640148>. Publisher: [American Statistical Association, Taylor & Francis, Ltd.].
- A. Graves. Practical variational inference for neural networks. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/7eb3c8be3d411e8ebfab08eba5f49632-Paper.pdf.
- S. A. Green, L. Hernandez, S. Y. Bookheimer, and M. Dapretto. Salience network connectivity in autism is related to brain and behavioral markers of sensory overresponsivity. *Journal of the American Academy of Child & Adolescent Psychiatry*, 55(7):618–626, 2016.
- L. W. D. M. M. F. B. J. P. Haolun Shi, Yuping Yang and J. Cao. Two-dimensional functional principal component analysis for image feature extraction. *Journal of Computational and Graphical Statistics*, 31(4):1127–1140, 2022. doi: 10.1080/10618600.2022.2035738. URL <https://doi.org/10.1080/10618600.2022.2035738>.
- M. M. Haznedar, M. S. Buchsbaum, T.-C. Wei, P. R. Hof, C. Cartwright, C. A. Bienstock, and E. Hollander. Limbic circuitry in patients with autism spectrum disorders studied with positron emission tomography and magnetic resonance imaging. *American Journal of Psychiatry*, 157(12):1994–2001, 2000.
- W. He and Z. Jiang. A survey on uncertainty quantification methods for deep neural networks: An uncertainty source perspective. *arXiv preprint arXiv:2302.13425*, 2023.
- A. S. Heinsfeld, A. R. Franco, R. C. Craddock, A. Buchweitz, and F. Meneguzzi. Identification of autism spectrum disorder using deep learning and the ABIDE dataset. *NeuroImage. Clinical*, 17:16–23, 2018. ISSN 2213-1582. doi: 10.1016/j.nicl.2017.08.017. Place: Netherlands.

- A. E. Hoerl and R. W. Kennard. Ridge regression: Applications to nonorthogonal problems. *Technometrics*, 12(1):69–82, 1970. ISSN 00401706. URL <http://www.jstor.org/stable/1267352>.
- F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In C. A. C. Coello, editor, *Learning and Intelligent Optimization*, pages 507–523. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-25566-3.
- D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. 13(4):455–492. ISSN 1573-2916. doi: 10.1023/A:1008306431147. URL <https://doi.org/10.1023/A:1008306431147>.
- R. Kawamoto, K. Kohara, T. Kusunoki, Y. Tabara, M. Abe, and T. Miki. Alanine aminotransferase/aspartate aminotransferase ratio is the best surrogate marker for insulin resistance in non-obese Japanese adults. *Cardiovascular Diabetology*, 11(1):117, Oct. 2012. ISSN 1475-2840. doi: 10.1186/1475-2840-11-117. URL <https://doi.org/10.1186/1475-2840-11-117>.
- I. Kim, H. Pang, and H. Zhao. Bayesian semiparametric regression models for evaluating pathway effects on continuous and binary clinical outcomes. *Statistics in medicine*, 31(15):1633–1651, July 2012. ISSN 1097-0258 0277-6715. doi: 10.1002/sim.4493. Place: England.
- R. Koenker and G. Bassett. Regression quantiles. *Econometrica*, 46(1):33–50, 1978. ISSN 00129682, 14680262. URL <http://www.jstor.org/stable/1913643>.
- Y. Kong, J. Gao, Y. Xu, Y. Pan, J. Wang, and J. Liu. Classification of autism spectrum disorder by combining brain connectivity and deep neural network classifier. *Neurocomputing*, 324:63–68, 2019. ISSN 0925-2312. doi: <https://doi.org/10.1016/j>.

- neucom.2018.04.080. URL <https://www.sciencedirect.com/science/article/pii/S0925231218306234>. Deep Learning for Biological/Clinical Data.
- H. Kozumi and G. Kobayashi. Gibbs sampling methods for bayesian quantile regression. *Journal of Statistical Computation and Simulation*, 81(11):1565–1578, 2011. doi: 10.1080/00949655.2010.496117. URL <https://doi.org/10.1080/00949655.2010.496117>.
- S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951. doi: 10.1214/aoms/1177729694. URL <https://doi.org/10.1214/aoms/1177729694>.
- L. Kuo and B. Mallick. Variable selection for regression models. *Sankhyā: The Indian Journal of Statistics, Series B (1960-2002)*, 60(1):65–81, 1998. ISSN 05815738. URL <http://www.jstor.org/stable/25053023>.
- H. J. Kushner. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering*, 86(1):97–106, 03 1964. ISSN 0021-9223. doi: 10.1115/1.3653121. URL <https://doi.org/10.1115/1.3653121>.
- Y. LeCun, J. Denker, and S. Solla. Optimal brain damage. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. URL https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.
- T.-H. Lee and E. H. Telzer. Negative functional coupling between the right fronto-parietal and limbic resting state networks predicts increased self-control and later substance use onset in adolescence. *Developmental cognitive neuroscience*, 20:35–42, 2016.
- F. Li and N. R. Zhang. Bayesian variable selection in structured high-dimensional covariate spaces with applications in genomics. *Journal of the American Statistical Association*, 105: 1202–1214, 2010.

- W. Li, S. Wang, and G. Liu. Transformer-based model for fmri data: Abide results. In *2022 7th International Conference on Computer and Communication Systems (ICCCS)*, pages 162–167, 2022. doi: 10.1109/ICCCS55155.2022.9845999.
- Y. Liang, B. Liu, and H. Zhang. A Convolutional Neural Network Combined With Prototype Learning Framework for Brain Functional Network Classification of Autism Spectrum Disorder. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 29:2193–2202, 2021. ISSN 1558-0210 1534-4320. doi: 10.1109/TNSRE.2021.3120024. Place: United States.
- D. Liu, X. Lin, and D. Ghosh. Semiparametric regression of multidimensional genetic pathway data: Least-squares kernel machines and linear mixed models. *Biometrics*, 63(4):1079–1088, 2007. doi: <https://doi.org/10.1111/j.1541-0420.2007.00799.x>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1541-0420.2007.00799.x>.
- L. Mahler, Q. Wang, J. Steiglechner, F. Birk, S. Heczko, K. Scheffler, and G. Lohmann. Pretraining is all you need: A multi-atlas enhanced transformer framework for autism spectrum disorder classification. In A. Abdulkadir, D. R. Bathula, N. C. Dvornek, S. T. Govindarajan, M. Habes, V. Kumar, E. Leonardsen, T. Wolfers, and Y. Xiao, editors, *Machine Learning in Clinical Neuroimaging*, pages 123–132, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-44858-4.
- I. Malkiel, G. Rosenman, L. Wolf, and T. Hendler. Self-supervised transformers for fmri representation. In E. Konukoglu, B. Menze, A. Venkataraman, C. Baumgartner, Q. Dou, and S. Albarqouni, editors, *Proceedings of The 5th International Conference on Medical Imaging with Deep Learning*, volume 172 of *Proceedings of Machine Learning Research*, pages 895–913. PMLR, 06–08 Jul 2022. URL <https://proceedings.mlr.press/v172/malkiel22a.html>.

- G. Matheron. Principles of geostatistics. *Economic Geology*, 58:1246–1266, 1963. URL <https://api.semanticscholar.org/CorpusID:128762998>.
- V. Menon and L. Q. Uddin. Saliency, switching, attention and control: a network model of insula function. *Brain structure and function*, 214:655–667, 2010.
- J. Močkus. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference: Novosibirsk, July 1–7, 1974*, pages 400–404. Springer, 1975.
- V. K. Mootha, C. M. Lindgren, K.-F. Eriksson, A. Subramanian, S. Sihag, J. Lehar, P. Puigserver, E. Carlsson, M. Ridderstråle, E. Laurila, N. Houstis, M. J. Daly, N. Patterson, J. P. Mesirov, T. R. Golub, P. Tamayo, B. Spiegelman, E. S. Lander, J. N. Hirschhorn, D. Altshuler, and L. C. Groop. PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature genetics*, 34(3): 267–273, July 2003. ISSN 1061-4036. doi: 10.1038/ng1180. Place: United States.
- K. P. Murphy. *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.], 2013. ISBN 9780262018029 0262018020. URL https://www.amazon.com/Machine-Learning-Probabilistic-Perspective-Computation/dp/0262018020/ref=sr_1_2?ie=UTF8&qid=1336857747&sr=8-2.
- E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1): 141–142, 1964. doi: 10.1137/1109020. URL <https://doi.org/10.1137/1109020>.
- R. Neal. *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics. Springer New York, 2012. ISBN 9781461207450. URL <https://books.google.com/books?id=LHHrBwAAQBAJ>.
- H. S. Nogay and H. Adeli. Multiple Classification of Brain MRI Autism Spectrum Disorder by Age and Gender Using Deep Learning. *Journal of Medical Systems*, 48(1):15, Jan.

2024. ISSN 1573-689X. doi: 10.1007/s10916-023-02032-0. URL <https://doi.org/10.1007/s10916-023-02032-0>.
- A. O'Hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(1):1–42, 1978. ISSN 00359246. URL <http://www.jstor.org/stable/2984861>.
- R. OHara and M. Sillanpää. A review of bayesian variable selection methods: what, how and which. *Bayesian Analysis*, 4:85–117, 2009.
- M. Opper and D. Saad. *Advanced Mean Field Methods: Theory and Practice*. The MIT Press, 2001. ISBN 978-0-262-28105-8. doi: 10.7551/mitpress/1100.001.0001. URL <https://doi.org/10.7551/mitpress/1100.001.0001>.
- H. Pang, I. Kim, and H. Zhao. Random Effects Model for Multiple Pathway Analysis with Applications to Type II Diabetes Microarray Data. *Statistics in biosciences*, 7(2):167–186, Oct. 2015. ISSN 1867-1764 1867-1772. doi: 10.1007/s12561-014-9109-1.
- N. Polson and J. Scott. Local shrinkage rule, levy process and regularization regression. *Journal of the Royal Statistical Society, Series B*, 74:287–311, 2012.
- J. O. Ramsay and B. W. Silverman, editors. *Applied Functional Data Analysis*. Springer New York, New York, NY, 2002. ISBN 978-0-387-22465-7. doi: 10.1007/978-0-387-22465-7_1. URL https://doi.org/10.1007/978-0-387-22465-7_1.
- C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005. ISBN 978-0-262-25683-4. doi: 10.7551/mitpress/3206.001.0001. URL <https://doi.org/10.7551/mitpress/3206.001.0001>.
- P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman. Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):1009–1030, 2009. doi:

- <https://doi.org/10.1111/j.1467-9868.2009.00718.x>. URL <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2009.00718.x>.
- P. Ravikumar, M. J. Wainwright, and J. D. Lafferty. High-dimensional Ising model selection using ℓ_1 -regularized logistic regression. *The Annals of Statistics*, 38(3):1287 – 1319, 2010. doi: 10.1214/09-AOS691. URL <https://doi.org/10.1214/09-AOS691>.
- B. Reich, C. Storlie, and H. Nondell. Variable selection in bayesian smoothing spline anova models. *Journal of Econometrics*, 51:110–119, 2009.
- S. Reid and R. Tibshirani. Sparse regression and marginal testing using cluster prototypes. *Biostatistics*, 17(2):364–376, 11 2016. ISSN 1465-4644. doi: 10.1093/biostatistics/kxv049. URL <https://doi.org/10.1093/biostatistics/kxv049>.
- A. Riaz, M. Asad, E. Alonso, and G. Slabaugh. Deepfmri: End-to-end deep learning for functional connectivity and classification of adhd using fmri. *Journal of Neuroscience Methods*, 335:108506, 2020. ISSN 0165-0270. doi: <https://doi.org/10.1016/j.jneumeth.2019.108506>. URL <https://www.sciencedirect.com/science/article/pii/S0165027019303632>.
- B. P. Rogers, V. L. Morgan, A. T. Newton, and J. C. Gore. Assessing functional connectivity in the human brain by fmri. *Magnetic Resonance Imaging*, 25(10):1347–1357, 2007. ISSN 0730-725X. doi: <https://doi.org/10.1016/j.mri.2007.03.007>. URL <https://www.sciencedirect.com/science/article/pii/S0730725X07002238>.
- H. Salehinejad and S. Valaee. Ising-dropout: A regularization method for training and compression of deep neural networks. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3602–3606, 2019. URL <https://api.semanticscholar.org/CorpusID:67855504>.
- H. Salehinejad and S. Valaee. Pruning of convolutional neural networks using ising energy model. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Sig-*

- nal Processing (ICASSP)*, pages 3935–3939, 2021. URL <https://api.semanticscholar.org/CorpusID:231861759>.
- K. A. Sankararaman, S. Wang, and H. Fang. Bayesformer: Transformer with uncertainty estimation, 2022.
- N. Sanyal, M. Lo, K. Kauppi, S. Djurovic, O. Andreassen, V. Jonshon, and C. Chen. Gwasinlps non local prior based iterative snp selection tool for genome wide association studies. *Bioinformatics*, 35:1–11, 2017.
- A. Schaefer, R. Kong, E. M. Gordon, T. O. Laumann, X.-N. Zuo, A. J. Holmes, S. B. Eickhoff, and B. T. T. Yeo. Local-global parcellation of the human cerebral cortex from intrinsic functional connectivity mri. *Cerebral Cortex*, 28(9):3095–3114, 07 2017. ISSN 1047-3211. doi: 10.1093/cercor/bhx179. URL <https://doi.org/10.1093/cercor/bhx179>.
- F. Scheipl. spikeslabgam: Bayesian variable selection, model choice, and regularization for generalized additive mixed models in r. *Journal of Statistical Software*, 43:1–24, 2011.
- R. K. Schindhelm, M. Diamant, J. M. Dekker, M. E. Tushuizen, T. Teerlink, and R. J. Heine. Alanine aminotransferase as a marker of non-alcoholic fatty liver disease in relation to type 2 diabetes mellitus and cardiovascular disease. *Diabetes/Metabolism Research and Reviews*, 22(6):437–443, 2006. doi: <https://doi.org/10.1002/dmrr.666>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/dmrr.666>.
- B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016. doi: 10.1109/JPROC.2015.2494218.
- L. Shao, C. Fu, Y. You, and D. Fu. Classification of ASD based on fMRI data with deep learning. *Cognitive Neurodynamics*, 15(6):961–974, Dec. 2021. ISSN 1871-4099. doi: 10.1007/s11571-021-09683-0. URL <https://doi.org/10.1007/s11571-021-09683-0>.

- Y. She. Sparse regression with exact clustering. *Electronic Journal of Statistics*, 4(none):1055 – 1096, 2010. doi: 10.1214/10-EJS578. URL <https://doi.org/10.1214/10-EJS578>.
- M. Shin, A. Bhattacharya, and V. Johnson. Scalable bayesian variable selection using non-local prior densities in ultrahigh dimensional settings. *Stat Sinica*, 28:1053–1078, 2018.
- N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013. doi: 10.1080/10618600.2012.681250. URL <https://doi.org/10.1080/10618600.2012.681250>.
- J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.
- K. Sriram, R. Ramamoorthi, and P. Ghosh. Posterior Consistency of Bayesian Quantile Regression Based on the Misspecified Asymmetric Laplace Density. *Bayesian Analysis*, 8(2):479 – 504, 2013. doi: 10.1214/13-BA817. URL <https://doi.org/10.1214/13-BA817>.
- H.-I. Suk, C.-Y. Wee, S.-W. Lee, and D. Shen. State-space model with deep learning for functional dynamics estimation in resting-state fmri. *NeuroImage*, 129:292–307, 2016. ISSN 1053-8119. doi: <https://doi.org/10.1016/j.neuroimage.2016.01.005>. URL <https://www.sciencedirect.com/science/article/pii/S1053811916000100>.
- Z. Tang, Y. Shen, Y. Li, X. Zhang, J. Wen, C. Qian, W. Zhuang, X. Shi, and N. Yi. Group spike-and-slab lasso generalized linear models for disease prediction and associated genes detection by incorporating pathway information. *Bioinformatics*, 34(6):901–910, 10 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx684. URL <https://doi.org/10.1093/bioinformatics/btx684>.

- T. Tao. When is correlation transitive? [blog post]. 2014. URL <https://terrytao.wordpress.com/2014/06/05/when-is-correlation-transitive/>.
- W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. ISSN 00063444. URL <http://www.jstor.org/stable/2332286>.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246. URL <http://www.jstor.org/stable/2346178>.
- R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(1):91–108, 2005. ISSN 13697412, 14679868. URL <http://www.jstor.org/stable/3647602>.
- M. K. Titsias and M. Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML’14, page II–1971–II–1980. JMLR.org, 2014.
- L. Q. Uddin. Salience processing and insular cortical function and dysfunction. *Nature reviews neuroscience*, 16(1):55–61, 2015.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.
- L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus. Regularization of neural networks using dropconnect. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine*

- Learning Research*, pages 1058–1066, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL <https://proceedings.mlr.press/v28/wan13.html>.
- H. Wang and D.-Y. Yeung. Towards bayesian deep learning: A framework and some existing methods. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3395–3408, 2016.
- R. Wang, Q. He, C. Han, H. Wang, L. Shi, and Y. Che. A deep learning framework for identifying Alzheimer’s disease using fMRI-based brain network. *Frontiers in neuroscience*, 17:1177424, 2023. ISSN 1662-4548 1662-453X. doi: 10.3389/fnins.2023.1177424. Place: Switzerland.
- G. S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 26(4):359–372, 1964. ISSN 0581572X. URL <http://www.jstor.org/stable/25049340>.
- R. Wei, B. J. Reich, J. A. Hoppin, and S. Ghosal. Sparse bayesian additive nonparametric regression with application to health effects of pesticides mixtures. *Statistica Sinica*, 30: 55–79, 2020.
- B. T. T. Yeo, F. M. Krienen, J. Sepulcre, M. R. Sabuncu, D. Lashkari, M. Hollinshead, J. L. Roffman, J. W. Smoller, L. Zöllei, J. R. Polimeni, B. Fischl, H. Liu, and R. L. Buckner. The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of neurophysiology*, 106(3):1125–1165, Sept. 2011. ISSN 1522-1598 0022-3077. doi: 10.1152/jn.00338.2011. Place: United States.
- G. Yi, J. Q. Shi, and T. Choi. Penalized gaussian process regression and classification for high-dimensional nonlinear data. *Biometrics*, 67(4):1285–1294, 2011. ISSN 0006341X, 15410420. URL <http://www.jstor.org/stable/41434433>.

- D. Yu, S. J. Lee, W. J. Lee, S. C. Kim, J. Lim, and S. W. Kwon. Classification of spectral data using fused lasso logistic regression. *Chemometrics and Intelligent Laboratory Systems*, 142:70–77, 2015. ISSN 0169-7439. doi: <https://doi.org/10.1016/j.chemolab.2015.01.006>. URL <https://www.sciencedirect.com/science/article/pii/S0169743915000088>.
- K. Yu and J. Zhang. A three-parameter asymmetric laplace distribution and its extension. *Communications in Statistics - Theory and Methods*, 34(9-10):1867–1879, 2005. doi: 10.1080/03610920500199018. URL <https://doi.org/10.1080/03610920500199018>.
- M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006. doi: <https://doi.org/10.1111/j.1467-9868.2005.00532.x>. URL <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00532.x>.
- J. Zeng, A. Lesnikowski, and J. M. Alvarez. The relevance of bayesian layer positioning to model uncertainty in deep bayesian active learning, 2018.
- D. Zhang and X. Lin. Hypothesis testing in semiparametric additive mixed models. *Biostatistics*, 4(1):57–74, 01 2003. ISSN 1465-4644. doi: 10.1093/biostatistics/4.1.57. URL <https://doi.org/10.1093/biostatistics/4.1.57>.
- Y. Zhang and J. Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=vSVLM2j9eie>.
- Z. Zhang, S. Sinha, T. Maiti, and E. Shipp. Bayesian variable selection in the accelerated failure time model with an application to the surveillance, epidemiology, and end results breast cancer data. *Statistical methods in medical research*, 27(4):971–990, Apr. 2018. ISSN 1477-0334 0962-2802. doi: 10.1177/0962280215626947. Place: England.

- L. K. Zhengwu Zhang, Xiao Wang and H. Zhu. High-dimensional spatial quantile function-on-scalar regression. *Journal of the American Statistical Association*, 117(539):1563–1578, 2022. doi: 10.1080/01621459.2020.1870984. URL <https://doi.org/10.1080/01621459.2020.1870984>. PMID: 37008532.
- H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006. doi: 10.1198/016214506000000735. URL <https://doi.org/10.1198/016214506000000735>.
- H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2):301–320, 2005. ISSN 13697412, 14679868. URL <http://www.jstor.org/stable/3647580>.

Appendices

Appendix A

Bayesian Multilevel Network Recovery

A.1 Variational Bayes Formula Derivations

For this section, we denote $\boldsymbol{\nu} = (\underline{\gamma}, \underline{\alpha}, \sigma^2)$ for notational brevity.

A.1.1 Deriving the Variational Distribution

$$\min_Q KL(Q_{\tilde{\underline{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}} || P_{\tilde{\underline{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}}) = \min_{\tilde{\underline{\xi}}_p \sim Q} \left[\ln(Q(\tilde{\underline{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu})) - \ln(P(\tilde{\underline{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu})) \right],$$

where “ $\tilde{\underline{\xi}}_p \sim Q$ ” represents that $\tilde{\underline{\xi}}_p$ is sampled from Q . By applying Bayes’ rule, we have

$$KL(Q_{\tilde{\underline{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}} || P_{\tilde{\underline{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}}) = E_{\tilde{\underline{\xi}}_p \sim Q} \left[\ln(Q(\tilde{\underline{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu})) - \ln(P(\underline{\mathbf{y}}|\tilde{\underline{\xi}}_p, \boldsymbol{\nu})) - \ln(P(\tilde{\underline{\xi}}_p|\boldsymbol{\nu})) + \ln(P(\underline{\mathbf{y}}|\boldsymbol{\nu})) \right].$$

Since $P(\underline{\mathbf{y}}|\boldsymbol{\nu})$ is independent of $\tilde{\underline{\xi}}_p$, we can pull it out of the expectation so that

$$\ln(P(\underline{\mathbf{y}}|\boldsymbol{\nu})) - KL(Q_{\tilde{\underline{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}} || P_{\tilde{\underline{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}}) = -E_{\tilde{\underline{\xi}}_p \sim Q} \left[\ln(Q(\tilde{\underline{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu})) - \ln(P(\underline{\mathbf{y}}|\tilde{\underline{\xi}}_p, \boldsymbol{\nu})) - \ln(P(\tilde{\underline{\xi}}_p|\boldsymbol{\nu})) \right],$$

which implies

$$\ln(P(\underline{\mathbf{y}}, \boldsymbol{\nu})) \geq E_{\tilde{\underline{\xi}}_p \sim Q} \left[-\ln(Q(\tilde{\underline{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu})) + \ln(P(\underline{\mathbf{y}}|\tilde{\underline{\xi}}_p, \boldsymbol{\nu})) + \ln(P(\tilde{\underline{\xi}}_p|\boldsymbol{\nu})) \right].$$

Here the equality holds when our objective of minimizing $KL(Q_{\tilde{\underline{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}} || P_{\tilde{\underline{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}})$ is met at

$KL(Q_{\tilde{\xi}_p|\underline{\mathbf{y}}, \underline{\nu}} || P_{\tilde{\xi}_p|\underline{\mathbf{y}}, \underline{\nu}}) = 0$. As such, our problem now becomes:

$$\max_Q E_{\tilde{\xi}_p \sim Q} \left[\ln \left(\frac{P(\underline{\mathbf{y}}, \tilde{\xi}_p | \underline{\nu})}{Q(\tilde{\xi}_p | \underline{\mathbf{y}}, \underline{\nu})} \right) \right].$$

Assuming a discrete latent variable, we have

$$E_{\tilde{\xi}_p \sim Q} \left[\ln \left(\frac{P(\underline{\mathbf{y}}, \tilde{\xi}_p | \underline{\nu})}{Q(\tilde{\xi}_p | \underline{\mathbf{y}}, \underline{\nu})} \right) \right] = \sum_{\tilde{\xi}_p} \ln \left(\frac{P(\underline{\mathbf{y}}, \tilde{\xi}_p | \underline{\nu})}{Q(\tilde{\xi}_p | \underline{\mathbf{y}}, \underline{\nu})} \right) \cdot Q(\tilde{\xi}_p | \underline{\mathbf{y}}, \underline{\nu}). \quad (\text{A.1})$$

By the assumption of the factorizability (2.8) of $Q(\tilde{\xi}_p | \underline{\mathbf{y}}, \underline{\nu})$, the equation (B.1) can be rewrit-

ten as

$$\begin{aligned}
&= \sum_{\tilde{\xi}_p} \left(\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) - \ln \left(\prod_{l \in \mathcal{P}_p} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \right) \right) \cdot \prod_{l \in \mathcal{P}_p} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \\
&= \sum_{\tilde{\xi}_{lp}} \sum_{\xi_{-lp}} \left(\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) - \ln \left(\prod_{l \in \mathcal{P}_p} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \right) \right) \cdot Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \\
&= \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \sum_{\tilde{\xi}_{-lp}} \prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \left(\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) - \ln \left(\prod_{l \in \mathcal{P}_p} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \right) \right) \\
&= \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \sum_{\tilde{\xi}_{-lp}} \prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \left(\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) - \ln Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \right. \\
&\quad \left. - \ln \left(\prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \right) \right) \\
&= \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \sum_{\tilde{\xi}_{-lp}} \prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \left(\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) \right. \\
&\quad \left. - \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \sum_{\tilde{\xi}_{-lp}} \prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \left(\ln Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) + \ln \left(\prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \right) \right) \right) \\
&= \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) E_{-\tilde{\xi}_{lp}} \left[\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) \right] - \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) E_{-\tilde{\xi}_{lp}} \left[\ln Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \right] \\
&\quad - \underbrace{E_{-\tilde{\xi}_{lp}} \left[\ln \left(\prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \right) \right]}_{constant} \\
&= \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) E_{-\tilde{\xi}_{lp}} \left[\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) \right] - \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \ln Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) + constant,
\end{aligned}$$

which is clearly minimized by setting $\ln Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \propto E_{-\tilde{\xi}_{lp}} \left[\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) \right]$ such that

$$Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \propto \exp \left\{ E_{-\tilde{\xi}_{lp}} \left[\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) \right] \right\}.$$

Remark that we use proportionality but not equality since we are still restricted by $Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu)$ being a distribution, so it requires a normalizing constant.

A.1.2 Deriving the formula for updating the subnode selection parameters

Plugging (2.9) into our proposed $Q_l(\tilde{\xi}_{lp}|\underline{\mathbf{y}}, \boldsymbol{\nu})$, we have:

$$\begin{aligned}
Q_l(\tilde{\xi}_{lp}|\underline{\mathbf{y}}, \boldsymbol{\nu}) &\propto \exp \left\{ E_{-\tilde{\xi}_{lp}} \left[\ln \left(\prod_{l \in \mathcal{P}_p} \left[E[\gamma_p] \exp \left\{ \sum_{j \neq l \in \mathcal{P}_p} \tilde{\xi}_{lp} \omega_{ljp} \tilde{\xi}_{jp} \right\} + (1 - E[\gamma_p]) \delta_{0_{lp}} \right] P(\underline{\mathbf{y}}|\underline{\tilde{\xi}}_p, \boldsymbol{\nu}) \right) \right] \right\} \\
&= \exp \left\{ E_{-\tilde{\xi}_{lp}} \left[\ln \left(\prod_{l \in \mathcal{P}_p} \left[E[\gamma_p] \exp \left\{ \sum_{j \neq l \in \mathcal{P}_p} \tilde{\xi}_{lp} \omega_{ljp} \tilde{\xi}_{jp} \right\} + (1 - E[\gamma_p]) \delta_{0_{lp}} \right] \right) + \ln \left(P(\underline{\mathbf{y}}|\underline{\tilde{\xi}}_p, \boldsymbol{\nu}) \right) \right] \right\} \\
&= \exp \left\{ E_{-\tilde{\xi}_{lp}} \left[\sum_{l \in \mathcal{P}_p} \left(\ln \left[E[\gamma_p] \exp \left\{ \sum_{j \neq l \in \mathcal{P}_p} \tilde{\xi}_{lp} \omega_{ljp} \tilde{\xi}_{jp} \right\} + (1 - E[\gamma_p]) \delta_{0_{lp}} \right] \right) + \ln \left(P(\underline{\mathbf{y}}|\underline{\tilde{\xi}}_p, \boldsymbol{\nu}) \right) \right] \right\}. \tag{A.2}
\end{aligned}$$

Therefore, when $E[\gamma_p] = 1$, the equation (B.2) becomes

$$\begin{aligned}
&\exp \left\{ E_{-\tilde{\xi}_{lp}} \left[\sum_{l \in \mathcal{P}_p} \ln \left[\exp \left\{ \sum_{j \neq l \in \mathcal{P}_p} \tilde{\xi}_{lp} \omega_{ljp} \tilde{\xi}_{jp} \right\} + \ln \left(P(\underline{\mathbf{y}}|\underline{\tilde{\xi}}_p, \boldsymbol{\nu}) \right) \right] \right] \right\} \\
&= \exp \left\{ E_{-\tilde{\xi}_{lp}} \left[\sum_{j \neq l \in \mathcal{P}_p} \tilde{\xi}_{lp} \omega_{ljp} \tilde{\xi}_{jp} + \ln \left(P(\underline{\mathbf{y}}|\underline{\tilde{\xi}}_p, \boldsymbol{\nu}) \right) \right] \right\} \\
&= \exp \left\{ \tilde{\xi}_{lp} \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} E_{-\tilde{\xi}_{lp}} [\tilde{\xi}_{jp} | \gamma_p = 1] + \ln P(\underline{\mathbf{y}}|\tilde{\xi}_{lp}, E_{-\tilde{\xi}_{lp}} [\tilde{\xi}_{-lp}], \boldsymbol{\nu}) \right\},
\end{aligned}$$

where

$$E_{-\tilde{\xi}_{lp}} [\tilde{\xi}_{jp} | \gamma_p = 1] = (+1) \cdot Q_j(\tilde{\xi}_{jp} = 1 | \underline{\mathbf{y}}, \gamma_p = 1) + (-1) \cdot Q_j(\tilde{\xi}_{jp} = -1 | \underline{\mathbf{y}}, \gamma_p = 1).$$

By denoting $L_{lp}^+ = \ln P(\underline{\mathbf{y}}|\tilde{\xi}_{lp} = +1, E_{-\tilde{\xi}_{lp}} [\tilde{\xi}_{-lp}], \boldsymbol{\nu})$, $L_{lp}^- = \ln P(\underline{\mathbf{y}}|\tilde{\xi}_{lp} = -1, E_{-\tilde{\xi}_{lp}} [\tilde{\xi}_{-lp}], \boldsymbol{\nu})$, and $m_{lp} = \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} E_{-\tilde{\xi}_{lp}} [\tilde{\xi}_{jp} | \gamma_p = 1]$ we have

$$Q_i(\tilde{\xi}_{lp} = +1 | \underline{\mathbf{y}}, \boldsymbol{\nu}, \gamma_p = 1) = \frac{\exp\{m_{lp} + L_{lp}^+\}}{\exp\{m_{lp} + L_{lp}^+\} + \exp\{-m_{lp} + L_{lp}^-\}}. \tag{A.3}$$

When $E[\gamma_p] = 0$, the expression (B.3) reduces to:

$$Q_i(\tilde{\xi}_{lp} = +1 | \underline{\mathbf{y}}, \underline{\boldsymbol{\nu}}, \gamma_p = 0) = \frac{\exp\{\ln(\delta_{0_p}) + L_{lp}^+\}}{\exp\{\ln(\delta_{0_p}) + L_{lp}^+\} + \exp\{\ln(\delta_{0_p}) + L_{lp}^-\}} = 0.$$

Therefore, we finally have

$$Q_i(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \gamma_p) = \gamma_p \frac{\exp\{m_{lp} + L_{lp}^+\}}{\exp\{m_{lp} + L_{lp}^+\} + \exp\{-m_{lp} + L_{lp}^-\}} + (1 - \gamma_p)(1 - \tilde{\xi}_{lp}).$$

Here $\tilde{\xi}_{lp}$'s depend on $\tilde{\xi}_{jp}$ s which in turn depend on $\tilde{\xi}_{lp}$ s. Hence, by introducing an iteration superscript (t) , we practically express

$$m_{lp}^{(t+1)} = \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} E_{-\tilde{\xi}_{jp}} [\tilde{\xi}_{jp} | \gamma_p = 1]^{(t)}. \quad (\text{A.4})$$

Dividing through by $\exp\{m_{lp}^{(t)} + L_{lp}^+\}$, the first term of the equation (B.4) becomes

$$= \frac{1}{1 + \exp\{-2m_{lp}^{(t)} + L_{lp}^- - L_{lp}^+\}}.$$

Therefore,

$$\begin{aligned} m_{lp}^{(t+1)} &= \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} \left[(+1) \cdot Q_j(\tilde{\xi}_{jp} = +1 | \underline{\mathbf{y}})^{(t)} + (-1) \cdot Q_j(\tilde{\xi}_{jp} = -1 | \underline{\mathbf{y}})^{(t)} \right] \\ &= \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} \left[(+1) \cdot \left(\frac{1}{1 + \exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}} \right) + (-1) \cdot \left(1 - \frac{1}{1 + \exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}} \right) \right] \\ &= \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} \left[\left(\frac{1}{1 + \exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}} \right) - \left(\frac{\exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}}{1 + \exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}} \right) \right] \\ &= \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} \left[\left(\frac{1 - \exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}}{1 + \exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}} \right) \right]. \end{aligned}$$

Note that the term in squared brackets is the conditional mean for $\tilde{\xi}_{jp}$. As such, for any l ,

we can compute the conditional mean of $\tilde{\xi}_{lp}$ as:

$$= \tanh \left(m_{lp}^{(t)} + \frac{L_{lp}^- - L_{lp}^+}{2} \right).$$

Therefore, we can have the following forms,

$$E_{\tilde{\xi}_{lp}}^{(t+1)} [\tilde{\xi}_{lp} | \gamma_p = 1] = \tanh \left(\sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} E_{-\tilde{\xi}_{lp}}^{(t)} [\tilde{\xi}_{jp}] + \frac{1}{2} \log \left(\frac{P(\underline{\mathbf{y}} | \tilde{\xi}_{lp} = +1, E_{-\tilde{\xi}_{lp}} [\tilde{\boldsymbol{\xi}}_{-lp}], \boldsymbol{\nu})}{P(\underline{\mathbf{y}} | \tilde{\xi}_{lp} = -1, E_{-\tilde{\xi}_{lp}} [\tilde{\boldsymbol{\xi}}_{-lp}], \boldsymbol{\nu})} \right) \right)$$

$$E_{\tilde{\xi}_{lp}}^{(t+1)} [\tilde{\xi}_{lp} | \gamma_p = 0] = -1.$$

Hence, by taking expectation over γ_p yields, we have the following form

$$\begin{aligned} E_{\tilde{\xi}_{lp}}^{(t+1)} [\tilde{\xi}_{lp}] &= E[\gamma_p] \tanh \left(\sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} E_{-\tilde{\xi}_{lp}}^{(t)} [\tilde{\xi}_{jp}] + \frac{1}{2} \log \left(\frac{P(\underline{\mathbf{y}} | \tilde{\xi}_{lp} = +1, E_{-\tilde{\xi}_{lp}} [\tilde{\boldsymbol{\xi}}_{-lp}], \boldsymbol{\nu})}{P(\underline{\mathbf{y}} | \tilde{\xi}_{lp} = -1, E_{-\tilde{\xi}_{lp}} [\tilde{\boldsymbol{\xi}}_{-lp}], \boldsymbol{\nu})} \right) \right) \\ &+ (1 - E[\gamma_p])(-1). \end{aligned} \tag{A.5}$$

Appendix B

Bayesian Multitasking Inference Method

B.1 Variational Bayes Formula Derivations

For this section, we denote $\boldsymbol{\nu} = (\underline{\boldsymbol{\alpha}}, \sigma^2)$ for notational brevity.

B.1.1 Deriving the Variational Distribution

$$\min_Q KL(Q_{\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}} || P_{\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}}) = \min_{\tilde{\boldsymbol{\xi}}_p \sim Q} \left[\ln(Q(\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu})) - \ln(P(\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu})) \right],$$

where “ $\tilde{\boldsymbol{\xi}}_p \sim Q$ ” represents that $\tilde{\boldsymbol{\xi}}_p$ is sampled from Q . By applying Bayes’ rule, we have

$$KL(Q_{\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}} || P_{\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}}) = E_{\tilde{\boldsymbol{\xi}}_p \sim Q} \left[\ln(Q(\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu})) - \ln(P(\underline{\mathbf{y}}|\tilde{\boldsymbol{\xi}}_p, \boldsymbol{\nu})) - \ln(P(\tilde{\boldsymbol{\xi}}_p|\boldsymbol{\nu})) + \ln(P(\underline{\mathbf{y}}|\boldsymbol{\nu})) \right].$$

Since $P(\underline{\mathbf{y}}|\boldsymbol{\nu})$ is independent of $\tilde{\boldsymbol{\xi}}_p$, we can pull it out of the expectation so that

$$\ln(P(\underline{\mathbf{y}}|\boldsymbol{\nu})) - KL(Q_{\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}} || P_{\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}}) = -E_{\tilde{\boldsymbol{\xi}}_p \sim Q} \left[\ln(Q(\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu})) - \ln(P(\underline{\mathbf{y}}|\tilde{\boldsymbol{\xi}}_p, \boldsymbol{\nu})) - \ln(P(\tilde{\boldsymbol{\xi}}_p|\boldsymbol{\nu})) \right],$$

which implies

$$\ln(P(\underline{\mathbf{y}}, \boldsymbol{\nu})) \geq E_{\tilde{\boldsymbol{\xi}}_p \sim Q} \left[-\ln(Q(\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu})) + \ln(P(\underline{\mathbf{y}}|\tilde{\boldsymbol{\xi}}_p, \boldsymbol{\nu})) + \ln(P(\tilde{\boldsymbol{\xi}}_p|\boldsymbol{\nu})) \right].$$

Here the equality holds when our objective of minimizing $KL(Q_{\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}} || P_{\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}})$ is met at

$KL(Q_{\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}} || P_{\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu}}) = 0$. As such, our problem now becomes:

$$\max_Q E_{\tilde{\boldsymbol{\xi}}_p \sim Q} \left[\ln \left(\frac{P(\underline{\mathbf{y}}, \tilde{\boldsymbol{\xi}}_p|\boldsymbol{\nu})}{Q(\tilde{\boldsymbol{\xi}}_p|\underline{\mathbf{y}}, \boldsymbol{\nu})} \right) \right].$$

Assuming a discrete latent variable, we have

$$E_{\tilde{\xi}_p \sim Q} \left[\ln \left(\frac{P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu)}{Q(\tilde{\xi}_p | \underline{\mathbf{y}}, \nu)} \right) \right] = \sum_{\tilde{\xi}_p} \ln \left(\frac{P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu)}{Q(\tilde{\xi}_p | \underline{\mathbf{y}}, \nu)} \right) \cdot Q(\tilde{\xi}_p | \underline{\mathbf{y}}, \nu). \quad (\text{B.1})$$

By the assumption of the factorizability of $Q(\tilde{\xi}_p | \underline{\mathbf{y}}, \nu)$, the equation (B.1) can be rewritten as

$$\begin{aligned} &= \sum_{\tilde{\xi}_p} \left(\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) - \ln \left(\prod_{l \in \mathcal{P}_p} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \right) \right) \cdot \prod_{l \in \mathcal{P}_p} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \\ &= \sum_{\tilde{\xi}_{lp}} \sum_{\xi_{-lp}} \left(\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) - \ln \left(\prod_{l \in \mathcal{P}_p} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \right) \right) \cdot Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \\ &= \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \sum_{\xi_{-lp}} \prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \left(\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) - \ln \left(\prod_{l \in \mathcal{P}_p} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \right) \right) \\ &= \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \sum_{\xi_{-lp}} \prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \left(\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) - \ln Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \right. \\ &\quad \left. - \ln \left(\prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \right) \right) \\ &= \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \sum_{\xi_{-lp}} \prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \left(\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) \right. \\ &\quad \left. - \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \sum_{\xi_{-lp}} \prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \left(\ln Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) + \ln \left(\prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \right) \right) \right) \\ &= \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) E_{-\tilde{\xi}_{lp}} \left[\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) \right] - \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) E_{-\tilde{\xi}_{lp}} \left[\ln Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \right] \\ &\quad - \underbrace{E_{-\tilde{\xi}_{lp}} \left[\ln \left(\prod_{j \neq l \in \mathcal{P}_p} Q_j(\tilde{\xi}_{jp} | \underline{\mathbf{y}}, \nu) \right) \right]}_{\text{constant}} \\ &= \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) E_{-\tilde{\xi}_{lp}} \left[\ln P(\underline{\mathbf{y}}, \tilde{\xi}_p | \nu) \right] - \sum_{\tilde{\xi}_{lp}} Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) \ln Q_l(\tilde{\xi}_{lp} | \underline{\mathbf{y}}, \nu) + \text{constant}, \end{aligned}$$

which is clearly minimized by setting $\ln Q_l(\tilde{\xi}_{lp}|\underline{\mathbf{y}}, \boldsymbol{\nu}) \propto E_{-\tilde{\xi}_{lp}} [\ln P(\underline{\mathbf{y}}, \tilde{\boldsymbol{\xi}}_p|\boldsymbol{\nu})]$ such that

$$Q_l(\tilde{\xi}_{lp}|\underline{\mathbf{y}}, \boldsymbol{\nu}) \propto \exp \left\{ E_{-\tilde{\xi}_{lp}} [\ln P(\underline{\mathbf{y}}, \tilde{\boldsymbol{\xi}}_p|\boldsymbol{\nu})] \right\}.$$

Remark that we use proportionality but not equality since we are still restricted by $Q_l(\tilde{\xi}_{lp}|\underline{\mathbf{y}}, \boldsymbol{\nu})$ being a distribution, so it requires a normalizing constant.

B.1.2 Deriving the formula for updating the subnode selection parameters

Plugging the Ising prior into our proposed $Q_l(\tilde{\xi}_{lp}|\underline{\mathbf{y}}, \boldsymbol{\nu})$, we have:

$$\begin{aligned} Q_l(\tilde{\xi}_{lp}|\underline{\mathbf{y}}, \boldsymbol{\nu}) &\propto \exp \left\{ E_{-\tilde{\xi}_{lp}} \left[\ln \left(\prod_{l \in \mathcal{P}_p} \left[\exp \left\{ \sum_{j \neq l \in \mathcal{P}_p} \tilde{\xi}_{lp} \omega_{ljp} \tilde{\xi}_{jp} \right\} P(\underline{\mathbf{y}}|\tilde{\boldsymbol{\xi}}_p, \boldsymbol{\nu}) \right) \right] \right\} \\ &= \exp \left\{ E_{-\tilde{\xi}_{lp}} \left[\ln \left(\prod_{l \in \mathcal{P}_p} \left[\exp \left\{ \sum_{j \neq l \in \mathcal{P}_p} \tilde{\xi}_{lp} \omega_{ljp} \tilde{\xi}_{jp} \right\} \right] \right) + \ln \left(P(\underline{\mathbf{y}}|\tilde{\boldsymbol{\xi}}_p, \boldsymbol{\nu}) \right) \right] \right\} \\ &= \exp \left\{ E_{-\tilde{\xi}_{lp}} \left[\sum_{l \in \mathcal{P}_p} \left(\ln \left[\exp \left\{ \sum_{j \neq l \in \mathcal{P}_p} \tilde{\xi}_{lp} \omega_{ljp} \tilde{\xi}_{jp} \right\} \right] \right) + \ln \left(P(\underline{\mathbf{y}}|\tilde{\boldsymbol{\xi}}_p, \boldsymbol{\nu}) \right) \right] \right\} \\ &= \exp \left\{ \tilde{\xi}_{lp} \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} E_{-\tilde{\xi}_{lp}} [\tilde{\xi}_{jp}] + \ln P(\underline{\mathbf{y}}|\tilde{\xi}_{lp}, E_{-\tilde{\xi}_{lp}} [\tilde{\boldsymbol{\xi}}_{-lp}], \boldsymbol{\nu}) \right\}, \end{aligned} \quad (\text{B.2})$$

where

$$E_{-\tilde{\xi}_{lp}} [\tilde{\xi}_{jp}] = (+1) \cdot Q_j(\tilde{\xi}_{jp} = 1|\underline{\mathbf{y}}) + (-1) \cdot Q_j(\tilde{\xi}_{jp} = -1|\underline{\mathbf{y}}).$$

By denoting $L_{lp}^+ = \ln P(\underline{\mathbf{y}}|\tilde{\xi}_{lp} = +1, E_{-\tilde{\xi}_{lp}} [\tilde{\boldsymbol{\xi}}_{-lp}], \boldsymbol{\nu})$, $L_{lp}^- = \ln P(\underline{\mathbf{y}}|\tilde{\xi}_{lp} = -1, E_{-\tilde{\xi}_{lp}} [\tilde{\boldsymbol{\xi}}_{-lp}], \boldsymbol{\nu})$, and $m_{lp} = \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} E_{-\tilde{\xi}_{lp}} [\tilde{\xi}_{jp}]$ we have

$$Q_i(\tilde{\xi}_{lp} = +1|\underline{\mathbf{y}}, \boldsymbol{\nu}) = \frac{\exp\{m_{lp} + L_{lp}^+\}}{\exp\{m_{lp} + L_{lp}^+\} + \exp\{-m_{lp} + L_{lp}^-\}}. \quad (\text{B.3})$$

Here $\tilde{\xi}_{lp}$'s depend on $\tilde{\xi}_{jp}$ s which in turn depend on $\tilde{\xi}_{lp}$ s. Hence, by introducing an iteration

superscript (t) , we practically express

$$m_{lp}^{(t+1)} = \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} E_{-\tilde{\xi}_{jp}} [\tilde{\xi}_{jp}]^{(t)}. \quad (\text{B.4})$$

Dividing through by $\exp\{m_{lp}^{(t)} + L_{lp}^+\}$, the first term of the equation (B.4) becomes

$$= \frac{1}{1 + \exp\{-2m_{lp}^{(t)} + L_{lp}^- - L_{lp}^+\}}.$$

Therefore,

$$\begin{aligned} m_{lp}^{(t+1)} &= \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} \left[(+1) \cdot Q_j(\tilde{\xi}_{jp} = +1 | \underline{\mathbf{y}})^{(t)} + (-1) \cdot Q_j(\tilde{\xi}_{jp} = -1 | \underline{\mathbf{y}})^{(t)} \right] \\ &= \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} \left[(+1) \cdot \left(\frac{1}{1 + \exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}} \right) + (-1) \cdot \left(1 - \frac{1}{1 + \exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}} \right) \right] \\ &= \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} \left[\left(\frac{1}{1 + \exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}} \right) - \left(\frac{\exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}}{1 + \exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}} \right) \right] \\ &= \sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} \left[\left(\frac{1 - \exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}}{1 + \exp\{-2m_{jp}^{(t)} + L_{jp}^- - L_{jp}^+\}} \right) \right]. \end{aligned}$$

Note that the term in squared brackets is the conditional mean for $\tilde{\xi}_{jp}$. As such, for any l , we can compute the conditional mean of $\tilde{\xi}_{lp}$ as:

$$= \tanh \left(m_{lp}^{(t)} + \frac{L_{lp}^- - L_{lp}^+}{2} \right).$$

Therefore, we can have the following forms,

$$E_{\tilde{\xi}_{lp}}^{(t+1)} [\tilde{\xi}_{lp} | \gamma_p = 1] = \tanh \left(\sum_{j \neq l \in \mathcal{P}_p} \omega_{ljp} E_{-\tilde{\xi}_{lp}}^{(t)} [\tilde{\xi}_{jp}] + \frac{1}{2} \log \left(\frac{P(\underline{\mathbf{y}} | \tilde{\xi}_{lp} = +1, E_{-\tilde{\xi}_{lp}} [\tilde{\xi}_{-lp}], \boldsymbol{\nu})}{P(\underline{\mathbf{y}} | \tilde{\xi}_{lp} = -1, E_{-\tilde{\xi}_{lp}} [\tilde{\xi}_{-lp}], \boldsymbol{\nu})} \right) \right)$$

Appendix C

Bayesian Inference in Transformer-based Models

C.1 Attention based crossformer

Let $\underline{\mathbf{x}}_p = \{x_p(t) | 1 \leq t \leq T\}$ represent the p^{th} functional covariate, where $p = 1, \dots, P$ represents a spatial location index and $t = 1, \dots, T$ represent the time index. We begin by segmenting each functional covariate across the functional covariate's domain into segments of equal length equal to L_{seg} . For a single functional covariate observation, we would therefore have:

$$\underline{\mathbf{x}}_{s,p} = \{x_p(t) | (s-1) \times L_{seg} < t \leq s \times L_{seg}\},$$

where $\underline{\mathbf{x}}_{s,p}$ is the s^{th} segment of the functional covariate.

Given these segments, we describe the concept of a two-stage attention function (f_{TSA}). In particular, we describe the architecture of the two-stage attention function, which allows us to estimate dependencies across both time and location domains in Section C.1.1.

C.1.1 Two-Stage Attention

Two-stage attention consists of attention across time and spaces which are described in the following sections C.1.1-C.1.1.

Attention Across Time

The first step in TSA is to linearly project these segments into our embedding space using the encoder structure of the model and adding a positional embedding (positional information encoded as a covariate) such that:

$$\mathbf{Z}_{s,p}^{(0)} = \mathbf{W}_{s,p}^{(emb)} \mathbf{X}_{s,p} + \mathbf{W}^{(pos)}(s, p), \quad (\text{C.1})$$

where $\mathbf{Z}_{s,p}^{(0)}$ is a vector of dimensionality matching that of the embedding latent space (which we will henceforth denote by d_{model}), $\mathbf{W}_{s,p}^{(emb)}$ is a $d_{model} \times L_{seg}$ linear projection matrix casting $\mathbf{X}_{s,p}$ into the latent space, and $\mathbf{W}^{(pos)}(s, p)$ is our the s^{th} segment's positional information encoded in a d_{model} -vector.

We then collate our $\mathbf{Z}_{s,p}^{(0)}$ vectors across all s segments and p locations to obtain an $S \times P \times d_{model}$ tensor $\mathbf{Z}$. We denote $\mathbf{Z}_{s,:}$ as the matrix of segment s for all locations, and likewise $\mathbf{Z}_{:,p}$ as all segments in the p^{th} location.

Next, we then apply the self-attention function (C.2) and extend it to the multiheaded self-attention (MSA) (Vaswani et al., 2017) to each location p . The self-attention function form is

$$f_{sa}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \frac{1}{1 + \exp \left[-\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_K}} \right]} \cdot \mathbf{V}, \quad (\text{C.2})$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ represent the input to f_{sa} , which in our application represent the same time series segments linearly projected into an embedding space, i.e. Z , and d_k represents the dimensionality of input K (which also matches the dimensionality of input Q). Note that the first term is a scaled dot product capturing similarity between the inputs on a unit length scale. The resulting similarity matrix is dot producted with our third argument such that it now acts as weights in a linear transformation.

By applying the self-attention function to partitions of the covariate dimension, the MSA

function (f_{msa}) are defined and concatenated together such that:

$$f_{msa}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \left(f_{sa}(\mathbf{Q}\mathbf{W}_1^{(Q)}, \mathbf{K}\mathbf{W}_1^{(K)}, \mathbf{V}\mathbf{W}_1^{(V)}), \dots, f_{sa}(\mathbf{Q}\mathbf{W}_h^{(Q)}, \mathbf{K}\mathbf{W}_h^{(K)}, \mathbf{V}\mathbf{W}_h^{(V)}) \right) \cdot \mathbf{W}^{(O)}, \quad (\text{C.3})$$

where the number of heads, h , is a tuning parameter; $\mathbf{W}_h^{(Q)} \in \mathbb{R}^{d_{model} \times d_K}$, $\mathbf{W}_h^{(K)} \in \mathbb{R}^{d_{model} \times d_K}$, $\mathbf{W}_h^{(V)} \in \mathbb{R}^{d_{model} \times d_V}$, $\mathbf{W}^{(O)} \in \mathbb{R}^{hd_V \times d_{model}}$ are linear projection matrices, and $d_K = d_V = d_{model}/h$.

This process enables each self-attention application to learn distinct aspects from other applications. Application of the full f_{msa} takes the following approach:

$$\hat{\mathbf{Z}}_{:,d}^{(time)} = LN\left(\mathbf{Z}_{:,p} + \mathbf{f}_{msa}^{(time)}(\mathbf{Z}_{:,p}, \mathbf{Z}_{:,p}, \mathbf{Z}_{:,p})\right) \quad (\text{C.4})$$

and

$$\mathbf{Z}^{(time)} = LN\left(\hat{\mathbf{Z}}^{(time)} + \mathbf{f}_{MLP}(\hat{\mathbf{Z}}^{(time)})\right),$$

where f_{MLP} is a multilayer perceptron function, LN indicates a standard normalization of the output, and $\hat{\mathbf{Z}}^{(time)}$ is obtained by collating $\hat{\mathbf{Z}}_{:,p}^{(time)}$ across locations. Note that this induces dependencies across time within each location.

Attention Across Space

In the next step, we further extend dependencies across locations. We follow the formulation of Zhang and Yan (2023), which reduces computational burden by using an intermediary step dubbed “routers,” which allows the model to bypass the computation of all pairwise connections between locations by first aggregating data from multiple locations into a smaller, c , number of tensors where $c \ll P$, as such the pairwise connections need to be estimated only between the tensors and the different locations. Application of f_{msa} to this stage

proceeds as follows:

$$\begin{aligned}
\mathbf{B}_{s,:} &= f_{msa}^{(loc)}(\mathbf{R}_{s,:}, \mathbf{Z}_{s,:}^{(\text{time})}, \mathbf{Z}_{s,:}^{(\text{time})}); \\
\bar{\mathbf{Z}}_{s,:}^{(\text{loc})} &= f_{msa}^{(loc)}(\mathbf{Z}_{s,:}^{(\text{time})}, \mathbf{B}_{s,:}, \mathbf{B}_{s,:}); \\
\hat{\mathbf{Z}}^{(\text{loc})} &= LN(\mathbf{Z}^{(\text{time})} + \bar{\mathbf{Z}}^{(\text{loc})}); \\
\mathbf{Z}^{(\text{loc})} &= LN(\hat{\mathbf{Z}}^{(\text{loc})} + \mathbf{f}_{MLP}(\hat{\mathbf{Z}}^{(\text{loc})})),
\end{aligned} \tag{C.5}$$

where $\mathbf{R} \in \mathbb{R}^{S \times c \times d_{model}}$ is a learnable tensor of routers, $\mathbf{R}_{s,:}$ is a $c \times d_{model}$ slice of the aforementioned tensor containing the representation of the s^{th} segment's embedding across all c routers, $\mathbf{B} \in \mathbb{R}^{S \times c \times d_{model}}$ is the aggregated messages across all locations, and $\bar{\mathbf{Z}}_{s,:}^{(\text{loc})}$ captures the importance of a time-segment to a given location.

We can now define f_{TSA} as:

$$f_{TSA}(\mathbf{Z}) = \mathbf{Z}^{(\text{loc})}.$$

Given f_{TSA} , we can stack several encoder representations of the functional covariates and apply f_{TSA} to induce a granular approach with different attention maps at each level of granularity, we will investigate different attention maps in the simulation section 4.5.

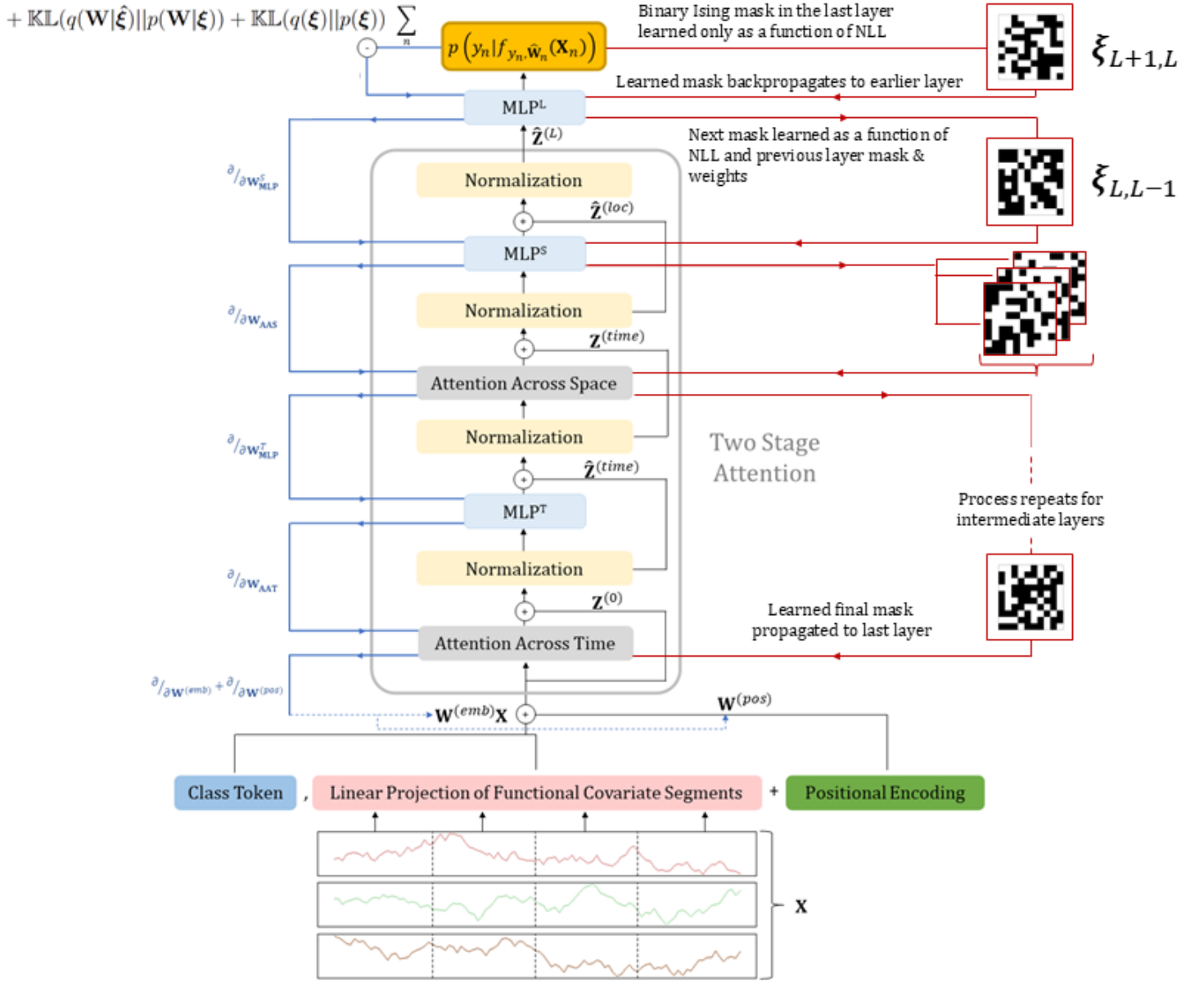


Figure C.1: Model Architecture highlighting the two-stage attention process and the layer-to-layer backpropagation of the Ising prior-based dropout masks. The masks are learned from the likelihood only in the last layer, then they are backpropagated to earlier layers as per the strength of the inter-layer connectivity.

Algorithm 5 Utilized Classification Crossformer Architecture

-
- 1: Initialize model parameters $L_{seg}, d_{model}, h, l_e, p$
 - 2: Segment functional covariates along time dimension such that for each p , $\mathbf{X}_p = (\mathbf{X}_{1,p}, \dots, \mathbf{X}_{S,p})$
 - 3: Apply a linear transformation to segments as: $\mathbf{X}_{s,p} \mathbf{W}_{s,p}^{(emb)}$
 - 4: Randomly Initialize positional embedding weights, $\mathbf{W}^{(pos)}$
 - 5: Add the positional embedding weights to the linearly transformed segments, $\mathbf{Z}_{s,p}^{(0)} = \mathbf{W}_{s,p}^{(emb)} \mathbf{X}_{s,p} + \mathbf{W}_p^{(pos)}$
 - 6: Collate $\mathbf{Z}_{s,p}^{(0)}$ into $N \times S \times d_{model}$ tensor, $\mathbf{Z}_p^{(0)}$
 - 7: Pass $\mathbf{Z}_p^{(0)}$ to the Encoder which performs the following operations:
 - 8: **Attention Across Time:**
 - 9: **for** each p **do**:
 - 10: Split each $\mathbf{Z}_p^{(0)}$ to h vectors each of length d_{model}/h , $\mathbf{Z}_{h,p}^{(0)}$
 - 11: **for** each h **do**:
 - 12: Define $\mathbf{Q} = \mathbf{Z}_{h,p}^{(0)} \mathbf{W}^{(Q)}$
 - 13: Define $\mathbf{K} = \mathbf{Z}_{h,p}^{(0)} \mathbf{W}^{(K)}$
 - 14: Define $\mathbf{V} = \mathbf{Z}_{h,p}^{(0)} \mathbf{W}^{(V)}$
 - 15: Apply $f_{sa}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{Z}_{h,p}^{(time)}$
 - 16: **end for**
 - 17: Collate output over h , $\mathbf{Z}_p^{(time)} = (\mathbf{Z}_{1,p}^{(time)}, \dots, \mathbf{Z}_{h,p}^{(time)})$
 - 18: apply a linear projection: $\mathbf{Z}_p^{(time)} \mathbf{W}_p^{(time)}$
 - 19: set $\hat{\mathbf{Z}}_{:,d}^{(time)} = LN(\mathbf{Z}_{:,d} + \mathbf{f}_{msa}^{(time)}(\mathbf{Z}_{:,d}, \mathbf{Z}_{:,d}, \mathbf{Z}_{:,d}))$
 - 20: set $\mathbf{Z}^{(time)} = LN(\hat{\mathbf{Z}}^{(time)} + \mathbf{f}_{MLP}(\hat{\mathbf{Z}}^{(time)}))$
 - 21: **end for**
 - 22: Initialize a random tensor of routers, $\mathbf{R}$, of dimensions $N \times S \times c \times d_{model}$
-

23: Attention Across Space:
24: for each s do:

 25: Split each $\mathbf{Z}_s^{(time)}$ to h vectors each of length d_{model}/h , $\mathbf{Z}_{h,s}^{(time)}$
26: for each h do:

 27: Define $\mathbf{Q} = \mathbf{Z}_{h,s}^{(time)} \mathbf{W}_B^{(Q)}$

 28: Define $\mathbf{K} = \mathbf{Z}_{h,s}^{(time)} \mathbf{W}_B^{(K)}$

 29: Define $\mathbf{V} = \mathbf{Z}_{h,s}^{(time)} \mathbf{W}_B^{(V)}$

 30: Apply $f_{sa}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{B}_{h,s}^{(time)}$
31: end for

 32: Collate output over h and apply a linear projection such that, $\mathbf{B}_{s,:}^{(loc)} = (\mathbf{B}_{1,s}^{(time)}, \dots, \mathbf{B}_{H,s}^{(time)}) \mathbf{W}_{s,:}^{(loc)}$

 33: Compute $\bar{\mathbf{Z}}_{s,:}^{(loc)} = f_{msa}^{(loc)}(\mathbf{Z}_{s,:}^{(time)}, \mathbf{B}_{s,:}, \mathbf{B}_{s,:})$
34: end for

 35: Collate output over s such that, $\bar{\mathbf{Z}}^{(loc)} = (\bar{\mathbf{Z}}_{s,:}^{(loc)}, \dots, \bar{\mathbf{Z}}_{s,:}^{(loc)})$

 36: set $\hat{\mathbf{Z}}^{(loc)} = LN(\mathbf{Z}^{(time)} + \bar{\mathbf{Z}}^{(loc)})$

 37: set $\mathbf{Z}^{(loc)} = LN(\hat{\mathbf{Z}}^{(loc)} + \mathbf{f}_{MLP}(\hat{\mathbf{Z}}^{(loc)}))$

 38: Append the class token, $\underline{\mathbf{y}}$ to $\mathbf{Z}^{(loc)}$
