Minimizing Negative Transfer of Knowledge in Multivariate Gaussian Processes: A Scalable and Regularized Approach

Raed Kontar, Garvesh Raskutti, and Shiyu Zhou

Abstract—Recently there has been an increasing interest in the multivariate Gaussian process (MGP) which extends the Gaussian process (GP) to deal with multiple outputs. One approach to construct the MGP and account for non-trivial commonalities amongst outputs employs a convolution process (CP). The CP is based on the idea of sharing latent functions across several convolutions. Despite the elegance of the CP construction, it provides new challenges that need yet to be tackled. First, even with a moderate number of outputs, model building is extremely prohibitive due to the huge increase in computational demands and number of parameters to be estimated. Second, the negative transfer of knowledge may occur when some outputs do not share commonalities. In this paper we address these issues. We propose a regularized pairwise modeling approach for the MGP established using CP. The key feature of our approach is to distribute the estimation of the full multivariate model into a group of bivariate GPs which are individually built. Interestingly pairwise modeling turns out to possess unique characteristics, which allows us to tackle the challenge of negative transfer through penalizing the latent function that facilitates information sharing in each bivariate model. Predictions are then made through combining predictions from the bivariate models within a Bayesian framework. The proposed method has excellent scalability when the number of outputs is large and minimizes the negative transfer of knowledge between uncorrelated outputs. Statistical guarantees for the proposed method are studied and its advantageous features are demonstrated through numerical studies.

Index Terms—Negative Transfer, Multivariate Gaussian process, Convolution process, Pairwise models, Regularization.

1 Introduction

Gaussian process regression models are widely used in several fields due to their desirable properties, such as flexibility, ease of implementation, uncertainty quantification and natural Bayesian interpretation [1]. Recently, there has been increasing interest in extending GP models to deal with multivariate outputs (also known as *cokriging*) due to their prevalence in many applications. For example, in manufacturing plants, hard to sample performance indicators can be predicted from other correlated and cheap to sample indicators [2]. Also, the future evolution of sensor signals from in-service devices can be predicted using previously observed signals from similar devices in the historical database [3]. Other applications arise in geostatistics, wireless networks and computer experiments.

The multivariate Gaussian process draws its roots from multitask learning. When multiple datasets from related outputs exist, integrative analysis can be advantageous relative to learning outputs independently. This integrative analysis, which leverages commonalities among related outputs to improve prediction and learning accuracy is referred to as multitask learning [4], [5]. The key feature in multitask learning is to provide a shared representation between training and testing outputs to allow the inductive transfer of knowledge. From an MGP perspective, this transfer of knowledge is achieved through specifying a valid positive semidefinite covariance function that models the dependencies of all data points within an output and across different outputs [6].

Traditionally, in MGP models, outputs are jointly modeled with a separable covariance structure, i.e. correlation over the same the input space is separable from between-output correlation [7]–[9]. In such methods, the separable covariance function is of the form $t \times \text{cov}(\boldsymbol{x}, \boldsymbol{x}')$ where t is the between-output covariance matrix and $\text{cov}(.,.)$ is a covariance function over inputs $\boldsymbol{x} \in \mathcal{R}^D$, the same for all outputs. This assumption is appealing due to the simplified covariance structure and significant reduction of model parameters, however it restricts all marginal GP's for all outputs to share the same set of covariance parameters defined in $\text{cov}(.,.)$.

On the other hand, nonseparable covariance functions allow outputs to possess both shared and unique features, as different outputs share information through different covariance parameters, therefore, accounting for non-trivial commonalities in the data. Recent work on nonseparable covariance functions are mainly based on convolution processes (CP) [10]–[12]. Earlier work, is known in the geostatistics literature as the linear model of coregionalization (LMC) [13] and can be seen as a special case of the CP framework [14]. The CP is based on the idea that a GP can be constructed by convolving a latent function with a smoothing kernel [15]. Thus, if each output is expressed as a convolution of a latent function drawn from a GP, and if we share these latent functions across multiple convolutions, then, multiple outputs can be expressed as a jointly distributed GP [16], [17]. As referred to by [18], the key feature of the CP approach is facilitating the non-instantaneous

- *R. Kontar is with the Department of Industrial & Operations Engineering Engineering, Universty of Michigan, Ann Arbor, MI, 48109. E-mail: alkontar@umich.edu*
- *G. Raskutti and S. Zhou are with the Universty of Wisconsin Madison.*

mixing of base processes, where, for instance, each output can be described using its own length-scale, which gives an added flexibility for describing the data.

Despite the elegance of the MGP established using CP, denoted as MGCP, model building is extremely prohibitive and often impractical even with a moderate number of outputs. Further, it is not uncommon for modern engineering systems to have a large number of outputs [19], [20]. For instance, when considering qualitative factors, the number of outputs in computer experiments increases dramatically based on all potential combinations [21]. This leads to a set of considerable challenges.

- *Challenge 1* (*Computational complexity*): The fact that the full covariance function of the joint GP should be considered results in significant computational burden and numerical issues. This challenge, in fact, is a direct consequence of accounting for multiple outputs and has been recently tackled in some literature [18], [22].

However, two other challenges arise with the CP construction and have yet to be tackled.

- *Challenge 2* (*High dimensional parameter space — Number of parameters*): The flexibility of the CP is based on providing different covariance parameters for different output levels, therefore, even for a moderate-scale problem, the number of parameters in the covariance function can easily reach hundreds or even thousands. This will lead to significant difficulties in solving the optimization problem to find the maximum likelihood estimator (MLE) for the parameters, specifically under non-convex and highly nonlinear settings.
- *Challenge 3* (*Negative transfer of knowledge*): The integrative analysis of multiple outputs implicitly assumes that these outputs share some commonalities. However, if this does not hold, negative transfer of knowledge may occur, which leads to decreased performance relative to learning tasks separately [23]. This is specifically important in the CP approach which, unlike separable approaches, implicitly implies that functions have unique features. Even though negative transfer is a very important issue, no research has handled this issue in the context of MGP models.

In the current literature, nonseparable modeling using MGCP is prohibitive even for a moderate number of outputs, due to *challenges 1* and *2*. Also, no literature has addressed the negative transfer of knowledge (*challenge 3*) in MGCP that results from the integrative analysis of outputs that share no commonalities. This article aims to simultaneously overcome these challenges through proposing a regularized pairwise modeling approach for MGCP models. The proposed method has excellent scalability when the number of outputs is large and minimizes the negative transfer of knowledge between uncorrelated outputs. Our approach is based on breaking down the high dimensional MGCP into a group of bivariate GP models, where the shared latent function parameters that facilitate the sharing of information in each bivariate model are penalized to prevent information sharing between outputs with no commonalities. Consistency in estimation and variable selection are then established. In summary, our contributions can be summarized as follows.

1) We introduce the notion of negative transfer in MGP's and provide the necessary and sufficient conditions for an MGP to collapse into independent GPs.
2) We provide a generic pairwise framework that addresses negative transfer while reducing the parameter space and computational complexity. Our approach
    a) scales to arbitrarily large datasets by parallelization, where each pairwise model can be estimated separately with only a small number of parameters.
    b) is generic regarding the choice of kernel and allows any sparse MGP approximation to be applied within.
3) We prove the consistency of our method in both selection (an oracle property) and estimation. Here variable selection implies selecting whether functions should be predicted independently or not .

Empirical evidence demonstrates that the proposed method can: (1) achieve similar prediction performance as the full multivariate approach when the output dimension is low, (2) outperform the full multivariate approach, with only a fraction of its computational needs, when the output dimension is high, (3) outperform the full multivariate approach when some functions are uncorrelated even when the output dimension is low.

The rest of the article is organized as follows. Section 2 provides some preliminaries related to the CP construction. In Section 3, we motivate the proposed method through expanding on the proposed challenges. Section 4 introduces our regularized pairwise modeling approach and proves some corresponding statistical properties. The advantageous features of our proposed method are then demonstrated through benchmarking our method with other reference methods in Section 5. Finally, Section 6 concludes this article with discussions. Technical details are deferred to the appendix.

2 Preliminaries

Consider a set of N output functions $\boldsymbol{y}(\boldsymbol{x}) = [y_1(\boldsymbol{x}), y_2(\boldsymbol{x}), ..., y_N(\boldsymbol{x})]^\top$ from inputs $\boldsymbol{x}$ lying in some input space $\mathcal{X} \subset \mathcal{R}^D$. The MGP is defined as

$$\mathcal{Y}(\boldsymbol{x}) = \begin{bmatrix} y_1(\boldsymbol{x}) \\ y_2(\boldsymbol{x}) \\ \vdots \\ y_N(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ \vdots \\ f_N(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} \epsilon_1(\boldsymbol{x}) \\ \epsilon_2(\boldsymbol{x}) \\ \vdots \\ \epsilon_N(\boldsymbol{x}) \end{bmatrix} = \mathcal{F}(\boldsymbol{x}) + \mathcal{E}(\boldsymbol{x}), \tag{1}$$

where the function $\mathcal{F} : \mathcal{R}^D \to \mathcal{R}^N$ is zero mean multivariate process with covariance $\text{cov}_{ij}^f(\boldsymbol{x}, \boldsymbol{x}') = \text{cov}_{ij}^f(f_i(\boldsymbol{x}), f_j(\boldsymbol{x}'))$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$, $i, j \in \mathcal{I} = \{1, 2, ..., N\}$ and $\epsilon_i(\boldsymbol{x}) \sim \mathcal{N}(0, \sigma_i^2)$ represents additive noise. For the ith output the observed data is denoted as $\mathcal{D}_i = \{(\boldsymbol{y}_i, \boldsymbol{X}_i)\}$, where

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.2987482, IEEE Transactions on Pattern Analysis and Machine Intelligence

$\boldsymbol{y}_i = [y_i^1, y_i^2, \ldots, y_i^{p_i}]^\top$ $y_i^c := y_i(\boldsymbol{x}_c)$ $\boldsymbol{X}_i = [\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{ip_i}]^\top$

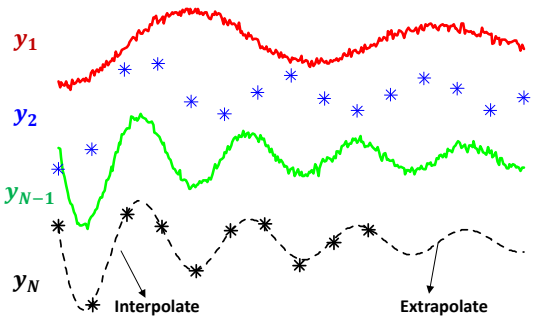


Fig. 1. Illustration of General MGP Setting

As shown in Figure 1, the underlying principle of the MGP is to borrow strength from a sample of curves, through the shared representation in (1) which enables integrative analysis, in order to predict individual outputs. For instance, assuming $P = \sum p_i$, at any new input $\boldsymbol{x}_0 \in \mathcal{X}$ belonging to output $i \in \mathcal{I}$ the joint distribution of the observed values from all outputs and the target function value at the test location $y_i^0 = y_i(\boldsymbol{x}_0)$ is given by

$$\begin{pmatrix} \boldsymbol{y} \\ y_i^0 \end{pmatrix} \Big| \boldsymbol{X} \sim \mathcal{N}\left(\boldsymbol{0}_{P+1}, \begin{bmatrix} \boldsymbol{C}_{\boldsymbol{f},\boldsymbol{f}} + \boldsymbol{\Sigma} & \boldsymbol{C}_{\boldsymbol{f},f_i^0} \\ \boldsymbol{C}_{\boldsymbol{f},f_i^0}^\top & C_{f_i^0,f_i^0} + \sigma_i^2 \end{bmatrix}\right), \tag{2}$$

where $\boldsymbol{y} = [\boldsymbol{y}_1^\top, \boldsymbol{y}_2^\top, ..., \boldsymbol{y}_N^\top]^\top$ are the noisy observed targets corresponding to the latent function values $\boldsymbol{f} = [\boldsymbol{f}_1^\top, \boldsymbol{f}_2^\top, ..., \boldsymbol{f}_N^\top]^\top$, $f_i^c := f_i(\boldsymbol{x}_{ic})$ such that $\boldsymbol{f}_i = f_i(\boldsymbol{X}_i)$, $\boldsymbol{C}_{\boldsymbol{f},\boldsymbol{f}} \in \mathcal{R}^{P\times P}$ is the covariance matrix relating all input points for all outputs with $\text{cov}_{ij}^f(\boldsymbol{x}, \boldsymbol{x}')$, $\boldsymbol{\Sigma} = \text{diag}[\sigma_1^2 \boldsymbol{I}_{p_1}, ..., \sigma_N^2 \boldsymbol{I}_{p_N}]$ is a block diagonal matrix in which the ith block corresponds to a $p_i \times p_i$ matrix, $\boldsymbol{C}_{\boldsymbol{f},f_i^0} = [\boldsymbol{C}_{\boldsymbol{f}_1,f_i^0}^\top, ..., \boldsymbol{C}_{\boldsymbol{f}_N,f_i^0}^\top]^\top$; $\boldsymbol{C}_{\boldsymbol{f}_c,f_i^0} = [\text{cov}_{ic}^f(\boldsymbol{x}_0, \boldsymbol{x}_{c1}), ..., \text{cov}_{ic}^f(\boldsymbol{x}_0, \boldsymbol{x}_{cp_c})]^\top$ and $C_{f_i^0,f_i^0} = \text{cov}_{ii}^f(\boldsymbol{x}_0, \boldsymbol{x}_0)$ where $f_i^0 := f_i(\boldsymbol{x}_0)$. Following multivariate normal theory, the predictive distribution of $y_i(\boldsymbol{x}_0)$ denoted as $\text{pr}(\cdot|\boldsymbol{y})$ is given as

$$\begin{aligned} \text{pr}(y_i(\boldsymbol{x}_0)|\boldsymbol{y}) = \mathcal{N}\Big(&\boldsymbol{C}_{\boldsymbol{f},f_i^0}^\top (\boldsymbol{C}_{\boldsymbol{f},\boldsymbol{f}} + \boldsymbol{\Sigma})^{-1} \boldsymbol{y},\ C_{f_i^0,f_i^0} + \\ &\sigma_i^2 - \boldsymbol{C}_{\boldsymbol{f},f_i^0}^\top (\boldsymbol{C}_{\boldsymbol{f},\boldsymbol{f}} + \boldsymbol{\Sigma})^{-1} \boldsymbol{C}_{\boldsymbol{f},f_i^0}\Big). \end{aligned} \tag{3}$$

The mean in (3) is the empirical best linear unbiased estimator (EBLUP) of $y_i(\boldsymbol{x}_0)$, while the variance is divided into three parts, the first is the variance of the variable under study, σ_i^2 represents additive noise and the last part is the variance reduction due to the EBLUP approximation [24].

As shown from (3), the sharing of information is achieved through $\text{cov}_{ij}^f(\boldsymbol{x}, \boldsymbol{x}')$ which models the variations both within and across different outputs, to capture their relatedness and improve prediction accuracy. This covariance function is typically assumed to belong to a known parametric class [25] of covariance functions $\{\text{cov}_{ij}^f(\cdot, \cdot\,; \boldsymbol{\theta}_f), \boldsymbol{\theta}_f \in \Theta_f\}$, where $\forall \boldsymbol{\theta}_f \in \Theta_f$, $\text{cov}_{ij}^f(\cdot, \cdot\,; \boldsymbol{\theta}_f)$ is a valid positive semidefinite covariance function, such that Θ_f is a set that contains the true parameters $\boldsymbol{\theta}_f^*$ for the covariance of $\boldsymbol{f}(\boldsymbol{x})$. Although in univariate settings there are many well known valid autocovariance functions, however, in the MGP it is extremely challenging to define cross-covariance functions that result in valid covariance matrices [26].

An alternative to directly parametrizing a covariance function is to construct a GP, $f_i : \mathcal{R}^D \to \mathcal{R}$, through convolving a Gaussian white noise process $X(\boldsymbol{x})$ with a smoothing kernel $K_i(\boldsymbol{x}) = \alpha_i k_i(\boldsymbol{x})$ where $\alpha_i \in \mathcal{R}$ and $k_i : \mathcal{R}^D \to \mathcal{R}$. Since the base process is a GP, and a convolution is a linear operator on a function, then the convolved process is also a GP [27]–[29].

$$f_i(\boldsymbol{x}) = K_i(\boldsymbol{x}) \star X(\boldsymbol{x}) = \int_{-\infty}^{\infty} K_i(\boldsymbol{x} - \boldsymbol{u}) X(\boldsymbol{u}) d\boldsymbol{u}\,, \tag{4}$$

where $\star$ denotes a convolution, $\text{cov}(X_i(\boldsymbol{u}), X_i(\boldsymbol{u}')) = \delta(\boldsymbol{u} - \boldsymbol{u}') = \delta(\boldsymbol{u}' - \boldsymbol{u}) = \delta_{\boldsymbol{u}\boldsymbol{u}'}$ and δ is the Dirac delta function. This approach is equivalent to applying a stable linear filter, where the output $f_i(\boldsymbol{x})$ is a weighted integral over the input signal $X(\boldsymbol{x})$, weighted according to the impulse response $K_i(\boldsymbol{x})$. This requires the filter to be stable, where the output is bounded for all bounded input signals [30], i.e. for a positive real finite number a, $|X(\boldsymbol{x})| \leq a \implies |f_i(\boldsymbol{x})| \leq a \int_{-\infty}^{\infty} |k_i(\boldsymbol{u})| d\boldsymbol{u}$. Therefore the only restriction for constructing a valid GP is that the impulse response/kernel is absolutely integrable $\int_{-\infty}^{\infty} |k_i(\boldsymbol{u})| d\boldsymbol{u} < \infty$. Some applications and extensions of the CP approach for the single output case are presented in [31] and [25].

Under the CP construction as shown in (4), if we share the same latent function $X(\boldsymbol{x})$, across multiple outputs $f_i(\boldsymbol{x}), i \in \mathcal{I}$, then all outputs can be expressed as a jointly distributed GP, i.e. MGP [29]. The resulting covariance function will then only depend on the displacement vector $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{x}' \in \mathcal{R}^D$ and is given as

$$\begin{aligned} \text{cov}_{ij}^f(\boldsymbol{x}, \boldsymbol{x}') &= \int_{-\infty}^{\infty} K_i(\boldsymbol{x} - \boldsymbol{u}) K_j(\boldsymbol{x}' - \boldsymbol{u}) d\boldsymbol{u} \\ &= \int_{-\infty}^{\infty} K_i(\boldsymbol{u}) K_j(\boldsymbol{u} - \boldsymbol{d}) d\boldsymbol{u}. \end{aligned} \tag{5}$$

As shown in (5), instead of directly parametrizing a positive semi-definite covariance function we only need to specify the parameters of a smoothing kernel, where the resulting covariance function must be valid by construction. Therefore, the key advantages is that we can exploit influence of multiple latent functions, $X_q(\boldsymbol{x})$ where $q \in \{1, 2, .., Q\}$, to share information across different output levels through different covariance parameters encoded in the kernels $K_{qi}(\boldsymbol{x})$ as shown in (6) [11], [32].

$$f_i(\boldsymbol{x}) = \sum_{q=1}^{Q} K_{qi}(\boldsymbol{x}) \star X_q(\boldsymbol{x}) = \sum_{q=1}^{Q} \int_{-\infty}^{\infty} K_{qi}(\boldsymbol{x} - \boldsymbol{u}) X_q(\boldsymbol{u}) d\boldsymbol{u}. \tag{6}$$

3 Challenges and Motivation

In this section, we expand on the MGCP challenges and present the motivation for our proposed method.

3.1 Computational complexity

As shown in the previous section, the MGCP is fully parametrized through the kernel parameters $\boldsymbol{\theta}_f$ and measurement noise $\boldsymbol{\sigma} = \{\sigma_1, ..., \sigma_N\}$. Now denote $\boldsymbol{\theta} = \{\boldsymbol{\theta}_f^\top, \boldsymbol{\sigma}^\top\}^\top$ and all the observed data as $\mathcal{D} = \{\mathcal{D}_i, ..., \mathcal{D}_N\}$, then the likelihood of the joint MGCP is given as

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) = (2\pi)^{-P/2} |\boldsymbol{C}_{\boldsymbol{f},\boldsymbol{f}} + \boldsymbol{\Sigma}|^{-1/2} \times \exp(-\boldsymbol{y}(\boldsymbol{C}_{\boldsymbol{f},\boldsymbol{f}} + \boldsymbol{\Sigma})^{-1}\boldsymbol{y}^\top/2). \quad (7)$$

As shown in (7), learning from the likelihood requires the inversion of $\boldsymbol{C}_{\boldsymbol{f},\boldsymbol{f}}+\boldsymbol{\Sigma}$ and calculating its determinant at each step/iteration. The covariance matrix of the CP, assuming p observations for each of the N outputs, scales as Np leading to $\mathcal{O}(P^3) = \mathcal{O}(N^3p^3)$ computational complexity and $\mathcal{O}(N^2p^2)$ storage. Therefore, CP modeling approaches are plagued by extremely high computational loads and numerical issues. In addition, more input points are usually required for multivariate outputs, resulting in a further increase in the complexity. Although separable approaches consider multiple outputs however their restrictive covariance functions often lead to structured covariances which are easily manipulated [8], [9]. On the other hand, some recent approaches have tried to tackle this computational issue in the context of nonseparable covariances [18], [33]. For instance, [18] proposed an inducing variable approximation similar to the well-known partially independent training conditional (PITC) approximation [1]. Their key assumption is that that outputs $\{y_i(\boldsymbol{x}) : i \in \mathcal{I}\}$ would be conditionally independent if all latent functions $X_q(\boldsymbol{u})$ are observed at $\mathcal{S}$ inducing locations, i.e. $\text{pr}(\{y_i(\boldsymbol{x})\}_{i=1}^N | \{X_q(\boldsymbol{u})\}_{q=1}^Q) = \prod_{i=1}^N \text{pr}(y_i(\boldsymbol{x}) | \{X_q(\boldsymbol{u})\}_{q=1}^Q)$. This assumption leads to the inversion of a block diagonal covariance matrix which reduces the complexity to $\mathcal{O}(Np^3) + \mathcal{O}(NSp^2)$. If $\mathcal{S} = p$ then this matches the complexity of modeling with N independent GPs. Further, [3], [22] assumed training output as independent and proposed sharing latent functions $X_q(\boldsymbol{u})$ only between test and training outputs. This lead to a block arrowhead covariance matrix which also reduced the complexity of learning from the likelihood to that of modeling

General Multivariate GP Setting

Despite the disadvantages associated with high computational complexity, the main drawback, which renders nonseparable modeling prohibitive even for a moderate N, is the large number of the parameters that need to be estimated. This large number of parameters is a direct consequence of the CP construction which inherits its flexibility from the ability to share information through different kernels resulting in different covariance parameters for different output levels. It is clear from (3) that prediction accuracy is greatly dependent on the parameter estimates which are obtained from minimizing the negative log-likelihood function $\ell(\boldsymbol{\theta}; \mathcal{D}) = -\log \mathcal{L}(\boldsymbol{\theta}; \mathcal{D})$. Up to an additive constant, $\ell(\boldsymbol{\theta}; \mathcal{D}) = \frac{1}{2}\langle \boldsymbol{Y}, (\boldsymbol{C}_{\boldsymbol{f},\boldsymbol{f}} + \boldsymbol{\Sigma})^{-1} \rangle + \frac{1}{2}\log|\boldsymbol{C}_{\boldsymbol{f},\boldsymbol{f}} + \boldsymbol{\Sigma}|$ where $\langle \boldsymbol{A}, \boldsymbol{A}' \rangle = \text{trace}(\boldsymbol{A}\boldsymbol{A}')$ and $\boldsymbol{Y} = \boldsymbol{y}\boldsymbol{y}^\top$. Since, $\ell(\boldsymbol{\theta}; \mathcal{D})$ is highly nonlinear and non-convex in $\boldsymbol{\theta}$, its minimization in a high dimensional parameter space is extremely challenging, time consuming and suboptimal as one should anticipate poor parameter estimates in such high dimensions [34]–[36].

Furthermore, as shown in (6), the number of parameters depends on Q which is the number of latent functions induced in the model. For instance, consider the case in Figure 2 where we share latent functions between each pair of outputs, i.e. $Q = N(N-1)/2$ resulting in $N(N-1)$ kernels. This is comparable to two way interaction effects in ANOVA where [37] only pairwise interactions are considered through the shared latent GP. Now assume the kernels $K_{iq}(\boldsymbol{x}) = \alpha_{qi} k_{qi}(\boldsymbol{x})$ follow the most commonly used exponential kernel, $k_{qi}(\boldsymbol{x}) = \exp((\boldsymbol{x} - \boldsymbol{x}')^\top \boldsymbol{\Lambda}_{qi}(\boldsymbol{x} - \boldsymbol{x}'))$ where $\boldsymbol{\Lambda}_{qi}$ is a $D \times D$ positive definite diagonal matrix allowing different length scales for each dimension. As a result the total number of parameters in the model is $N(N-1)(1+D)+N$ where the first part $N(N-1)$ represents the number of kernels multiplied by the number of parameters $(1+D)$ in each kernel, while the second part N represents the number of parameters in $\boldsymbol{\sigma}$. Note that this case is a bit conservative as we use the exponential kernel which is able to provide a large degree of flexibility with a small number of hyper parameters [34], [38]. Even for a moderate scale case where $N = 30$ and $D = 1$, we are required to estimate 1770 parameters under a non-convex setting. In another case, considering the more restrictive approach in [39] and [40] where $Q = 1$ and all outputs possess the same noise, i.e. $\sigma = \sigma_1 =, .., = \sigma_N$, the number of parameters still scales as $N(N+2D+1)/2+1$, therefore for $N = 30$ and $D = 1$ we are estimating 991 parameters. In conclusion, obtaining good estimates in such a high dimensional parameters space is an impractical task, for this reason the practical applications of the MGCP are limited. Note that once the parameters are learned, prediction complexity at any new test point $\boldsymbol{x}_0$ is $\mathcal{O}(Np)$ for mean and $\mathcal{O}(N^2p^2)$ for variance, which can be done rather efficiently.

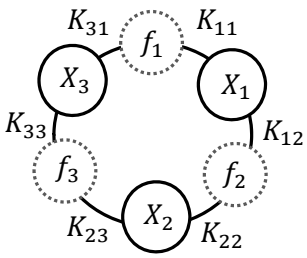


Fig. 2. CP model structure example

3.3 Negative transfer

As previously mentioned, a major challenge in the MGP which to the best of our knowledge has not been tackled yet, is the negative transfer of knowledge that occurs when we integratively analyze outputs that share no commonalities. Similar to the second challenge, negative transfer is specifically important in nonseperabale approaches which are used when functions have unique features. We start with

an example to illustrative the impact of negative transfer in MGCP models. Consider a simple case with two outputs and a one dimensional input $x \in \mathcal{X} \subset \mathcal{R}$. The outputs are generated according to $y_1(x) = 1 + \sin(x) + \epsilon_1(x)$ and $y_2(x) = 4 + 0.5\sin(1.5x) + \epsilon_2(x)$ and $x \in [0, 10]$. The number of observations per signal is $p_1 = p_2 = 20$ evenly spaced points and the measurement noise is set as $\sigma_1 = \sigma_2 = 0.1$. We analyze this data separately using a univariate GP applied to each output and integratively using an MGCP. In the univariate GP, we assume a Gaussian/squared exponential covariance where $\text{cov}_{ii}^f(x, x') = \alpha_i^2 \exp(-d^2/(4\nu^2))$. In the MGCP we adopt the covariance function in [11] and [40], where $\text{cov}_{ij}^f(x, x') = \alpha_i \alpha_j \sqrt{2|\nu_i \nu_j|/(\nu_i^2 + \nu_j^2)} \exp(-d^2/(2\nu_i^2 + 2\nu_j^2))$. Results are shown in Figure 3.

The results clearly indicate that separate modeling of each function is significantly better than their integrative analysis. This is specifically clear for y_2, which interestingly, was predicted using a larger length scale than the truth due to sharing of information with a smoother function y_1. Note that this problem occurred in our dense input example ($p_i = 20$) which implies that the challenge of negative transfer becomes exceedingly significant with sparse data. For these reasons, and since negative transfer is a crucial issue in MGP models, the main goal of this article is to handle the negative transfer of knowledge while maintaining the scalability of the model.

3.4 Motivation: Pairwise estimation and the precision matrix

In this article we propose a pairwise distributed estimation scheme motivated by both the distributed estimation literature for univariate GP models [41], [42] and pairwise modeling of longitudinal profiles [39], [43]. The proposed approach is based on distributing MGCP estimation through bivariate GP submodels which are individually estimated. Predictions are then made through combining predictions from the bivariate models within a Bayesian framework. Not only does this approach scale to arbitrarily large datasets by parallelization, also each bivariate model can be efficiently built with a limited of parameters and a small-scale covariance matrix.

While, pairwise modeling seems to address computational challenges, negative transfer remains an important issue. Interestingly, *pairwise modeling turns out to possess unique characteristics which allows us to tackle the challenge of negative transfer*. While few literature [44] have aimed to establish the number of latent functions to be shared, such approaches do not imply avoiding negative transfer. As we will show in this subsection, information sharing and independent predictions can only be avoided through independence and hence sparsity on the precision matrix. However, since MGCP models are based on modeling the covariance through latent functions not the precision matrix then we can only control the precision matrix under specific structures. It is clear from (3) that prediction accuracy for the GP/MGP is dependent on the inverse covariance matrix, also known as precision matrix $\boldsymbol{\Omega} = (\boldsymbol{C}_{\boldsymbol{f},\boldsymbol{f}} + \boldsymbol{\Sigma})^{-1} \in \mathcal{R}^{P \times P}$. The precision matrix carries conditional independence information. This matrix consists of block matrices $\boldsymbol{\Omega}_{ij} \in \mathcal{R}^{p_i \times p_j}$, where the $(c, c')^{th}$ entry of each block is denoted as $\Omega_{ij}^{c,c'}$. One can directly show that $\text{cov}(y_i^c, y_j^{c'} | \ddot{\boldsymbol{y}}) = 0$ if and only if $\Omega_{ij}^{c,c'} = 0$, where $\ddot{\boldsymbol{y}} = \{\boldsymbol{y}\}/\{y_i^c, y_j^{c'}\}$ ($\boldsymbol{y}$ excluding y_i^c and $y_j^{c'}$). Thus, conditionally independent variables lead to zero entries in the precision matrix [45]. As mentioned previously, GP/MGP models are characterized through a positive semidefinite covariance function (ex: $\text{cov}_{ij}^f(\boldsymbol{x}, \boldsymbol{x}')$) rather than a conditional covariance function to generate the precision matrix. However, the remarks below illustrate some useful properties in the case of a bivariate GP.

Lemma 1. (*Multivariate*) *Given that* $\text{pr}(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{0}_P, \boldsymbol{\Omega}^{-1})$, *then* $\boldsymbol{\Omega}_{ij} = \boldsymbol{0}$ *if and only if the multivariate Gaussian random vectors* $\boldsymbol{y}_i$ *and* $\boldsymbol{y}_j$ *are conditionally independent, i.e.* $cov(y_i^c, y_j^{c'} | \ddot{\boldsymbol{y}}) = 0$ *for every* $c \in \{1, .., p_i\}$ *and* $c' \in \{1, .., p_j\}$.

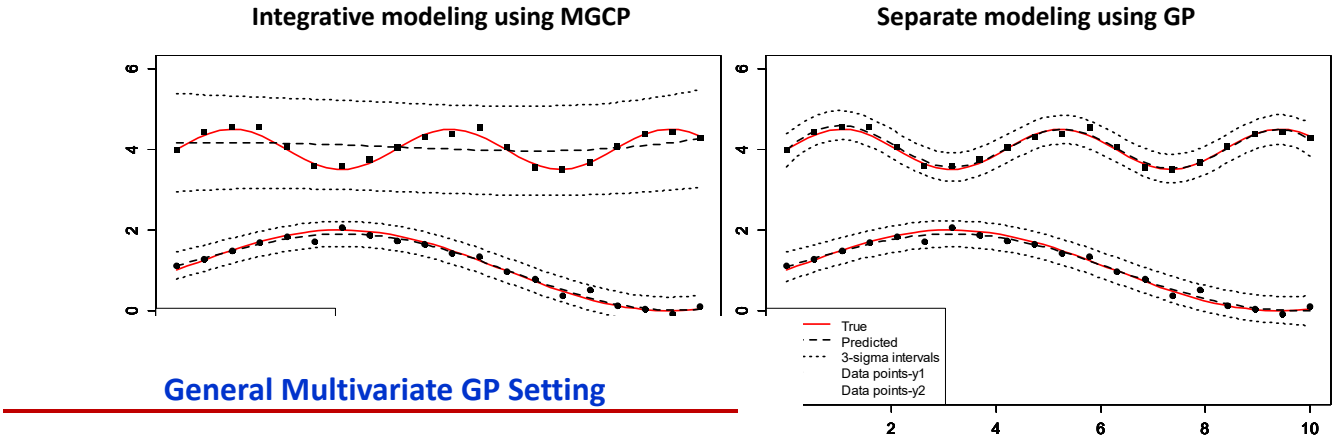
Lemma 2. (*Bivariate*) *Given that* $\text{pr}(\boldsymbol{y}_i, \boldsymbol{y}_j | \boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{\theta}') = \mathcal{N}\left(\boldsymbol{0}_{p_i+p_j}, \begin{pmatrix} \boldsymbol{\Omega}_{ii} & \boldsymbol{\Omega}_{ij} \\ \boldsymbol{\Omega}_{ij}^\top & \boldsymbol{\Omega}_{jj} \end{pmatrix}^{-1}\right)$, *then* $\boldsymbol{\Omega}_{ij} = \boldsymbol{0}$ *if and only if, the multivariate Gaussian random vectors* $\boldsymbol{y}_i$ *and* $\boldsymbol{y}_j$ *are independent, i.e.* $cov(y_i^c, y_j^{c'}) = 0$ *for every* $c \in \{1, .., p_i\}$ *and* $c' \in \{1, .., p_j\}$.

A brief proof of both Lemmas and further references are provided in Appendix A. The key conclusion from the Lemmas are as follows. Lemma 1 shows that an MGP collapses into independent GP's if and only if the inverse covariance off-diagonal blocks are zero. While Lemma 2 shows that through pairwise modeling, we are able to control the precision matrix through parameters in the covariance function used to construct the bivariate GP.

The utilization of pairwise modeling, distributed estimation and this direct mapping from the covariance function to the precision matrix is detailed in the following sections.

4 Model Development

The proposed framework presents a flexible alternative that can scale to a large number of outputs while avoiding the negative transfer of knowledge. The nature of our proposed pairwise approach circumvents any need to find or establish latent functions between pairs. While within each pair, model selection is automatically done through our regularization approach which is consistently able to infer whether information should be shared or not. In Section 4.1, we establish our pairwise model based on a CP construction. Our pairwise scheme is based on distributing the estimation of the high dimensional MGCP into bivariate GP's which are individually built. An MGCP with N outputs as a result decomposes into $N(N-1)/2$ pairwise submodels to predict all outputs. However, for the sake of notational simplicity, and building on Figure 1, we focus on predicting one output through sharing information from the remaining $N-1$ outputs as shown in Figure 4 below. In Section, 4.2 we provide some statistical guarantees for the proposed method. Section 4.3, provides a direct approach to applying our regularized pairwise approach to separable modeling. Finally, Section 4.4, provides the methodology to combine predictions from the bivariate submodels.



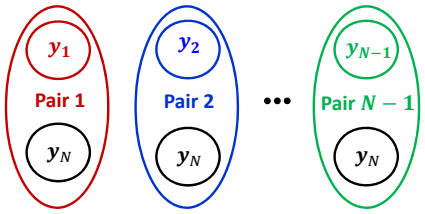


Fig. 4. Paired submodels

4.1 The pairwise and regularized MGCP

Based on (1) for each pairwise submodel we have $\begin{bmatrix} y_i(\boldsymbol{x}) \\ y_j(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} f_i(\boldsymbol{x}) \\ f_j(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} \epsilon_i(\boldsymbol{x}) \\ \epsilon_j(\boldsymbol{x}) \end{bmatrix}$, for $i, j \in \mathcal{I}$, where the input data is $\mathcal{D}_i = \{(\boldsymbol{y}_i, \boldsymbol{X}_j)\}$ and $\mathcal{D}_j = \{(\boldsymbol{y}_j, \boldsymbol{X}_j)\}$. Further, we assume that $\boldsymbol{y}_{ij} = [\boldsymbol{y}_i^\top, \boldsymbol{y}_j^\top]^\top$ represents the noisy observations corresponding to the latent function values $\boldsymbol{f}_{ij} = [\boldsymbol{f}_i^\top, \boldsymbol{f}_j^\top]^\top$. In order to capture both the unique properties of each output

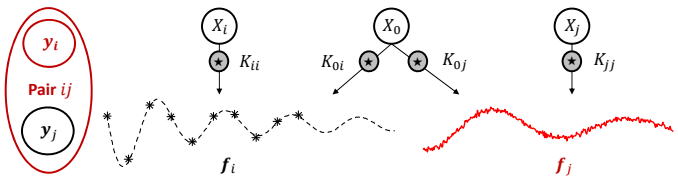


Fig. 5. Bivariate submodel structure

As shown in Figure 5, independent features are encoded through dependence on a latent function that has no effect on the other output (X_i, X_j) while dependent features are encoded through the common dependence on X_0.

Following the CP construction in (6), we have that $y_i(\boldsymbol{x}) = f_i(\boldsymbol{x}) + \epsilon_i(\boldsymbol{x}) = K_{ii}(\boldsymbol{x}) \star X_i(\boldsymbol{x}) + K_{0i}(\boldsymbol{x}) \star X_0(\boldsymbol{x}) + \epsilon_i(\boldsymbol{x})$, similarly, $y_j(\boldsymbol{x}) = K_{jj}(\boldsymbol{x}) \star X_j(\boldsymbol{x}) + K_{0j}(\boldsymbol{x}) \star X_0(\boldsymbol{x}) + \epsilon_j(\boldsymbol{x})$. This model is quite flexible, as it provides both shared and unique latent functions for both outputs. Based on this modeling framework, the cross covariance function between the two outputs is given as

$$\begin{aligned} \text{cov}_{ij}^{y}(\boldsymbol{x}, \boldsymbol{x}') &= \text{cov}_{ij}^{f}(\boldsymbol{x}, \boldsymbol{x}') + \text{cov}_{ij}^{\epsilon}(\boldsymbol{x}, \boldsymbol{x}') \\ &= \text{cov}_{ij}^{f}(\boldsymbol{x}, \boldsymbol{x}') + \sigma_i^2 \tau_{ij} \tau_{x,x'}, \end{aligned} \tag{8}$$

where, τ_{ij} is the Kronecker delta function, which is equal to one if $i = j$, and is zero otherwise. In a more general case to that of Section 2, we define the latent functions $\{X_q : q = 0, i, j\}$ as $\text{cov}(X_q(\boldsymbol{u}), X_q(\boldsymbol{u}')) = \xi_q^2 \delta_{\boldsymbol{u}\boldsymbol{u}'}$ where $\xi_q \in \mathcal{R}$. Then given the fact that $X_q(\boldsymbol{u})$ and $X_{q'}(\boldsymbol{u}')$ are independent latent functions which only covary if $q = q'$ and $\boldsymbol{u} = \boldsymbol{u}'$ and utilizing the commutativity of the convolution and the "sifting" property of the Dirac delta function, i.e. $\int f(\boldsymbol{u}) \delta(\boldsymbol{u} - \boldsymbol{x}) d\boldsymbol{z} = f(\boldsymbol{x})$, we have that $\text{cov}_{ij}^{f}(\boldsymbol{x}, \boldsymbol{x}') = E(f_i(\boldsymbol{x}) f_j(\boldsymbol{x}')) =$

$$\begin{aligned} & E\Big(\sum_{q \in \{0,i\}} \int_{-\infty}^{\infty} K_{qi}(\bar{\boldsymbol{x}}) X_q(\boldsymbol{u}) d\boldsymbol{u} \\ & \times \sum_{q' = \{0,j\}} \int_{-\infty}^{\infty} K_{qj}(\bar{\bar{\boldsymbol{x}}}) X_{q'}(\boldsymbol{u}') d\boldsymbol{u}' \Big) \\ &= \sum_{q \in \{0,i,j\}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} K_{qi}(\boldsymbol{u}) K_{qj}(\boldsymbol{u}') E(X_q(\bar{\boldsymbol{x}}) X_q(\bar{\bar{\boldsymbol{x}}})) d\boldsymbol{u} d\boldsymbol{u}' \\ &= \sum_{q \in \{0,i,j\}} \xi_q^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} K_{qi}(\boldsymbol{u}) K_{qj}(\boldsymbol{u}') \delta(\bar{\boldsymbol{x}} - \bar{\bar{\boldsymbol{x}}}) d\boldsymbol{u} d\boldsymbol{u}' \\ &= \sum_{q \in \{0,i,j\}} \xi_q^2 \int_{-\infty}^{\infty} K_{qi}(\boldsymbol{u}) K_{qj}(\boldsymbol{u} - \boldsymbol{d}) d\boldsymbol{u} \end{aligned} \tag{9}$$

where $\bar{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{u}$, $\bar{\bar{\boldsymbol{x}}} = \boldsymbol{x}' - \boldsymbol{u}'$. Note that the derivation in (9) is a wrapper function for $\text{cov}_{ii}^{f}(\boldsymbol{x}, \boldsymbol{x}')$, $\text{cov}_{jj}^{f}(\boldsymbol{x}, \boldsymbol{x}')$ and $\text{cov}_{ij}^{f}(\boldsymbol{x}, \boldsymbol{x}')$. Based on our model framework illustrated in Figure 5 and due to the shared dependence on only X_0, the autocovariance and cross covariance functions simplify as

$$\begin{cases} \text{cov}_{ii}^{f}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{q=\{0,i\}} \xi_q^2 \int_{-\infty}^{\infty} K_{qi}(\boldsymbol{u}) K_{qi}(\boldsymbol{u} - \boldsymbol{d}) d\boldsymbol{u} \\ \text{cov}_{jj}^{f}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{q=\{0,j\}} \xi_q^2 \int_{-\infty}^{\infty} K_{qj}(\boldsymbol{u}) K_{qj}(\boldsymbol{u} - \boldsymbol{d}) d\boldsymbol{u} \\ \text{cov}_{ij}^{f}(\boldsymbol{x}, \boldsymbol{x}') = \xi_0^2 \int_{-\infty}^{\infty} K_{0i}(\boldsymbol{u}) K_{0j}(\boldsymbol{u} - \boldsymbol{d}) d\boldsymbol{u} \end{cases} \tag{10}$$

This modeling framework is generic in terms of the choice of the kernel function. However, a general purpose kernel can be constructed through assuming the kernels follow a Gaussian form. As mentioned previously, the Gaussian kernel is the most common choice of kernels due to its flexibility and correspondence to Bayesian linear regression with an infinite number of basis functions [34]. Further, similar constructions using such kernels have been utilized in both a GP and MGP setting [18], [46].

Now assume the kernels $K_{qi}(\boldsymbol{x}) = \alpha_{qi}(4\pi)^{\frac{D}{4}}|\boldsymbol{\Lambda}_{qi}|^{-\frac{1}{4}}\mathcal{N}(\boldsymbol{x}|\boldsymbol{0},\boldsymbol{\Lambda}_{qi}^{-1})$ to be scaled Gaussian kernels. Also, denote $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{qi},\boldsymbol{\Lambda}_{qi}^{-1})\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{qj},\boldsymbol{\Lambda}_{qj}^{-1}) = \mathcal{N}(\boldsymbol{\mu}_{qi}-\boldsymbol{\mu}_{qj}|\boldsymbol{0},\boldsymbol{\Lambda}_{qi}^{-1}+\boldsymbol{\Lambda}_{qj}^{-1})\mathcal{N}(\boldsymbol{x}|\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Lambda}})$ where $\tilde{\boldsymbol{\Lambda}}^{-1} = (\boldsymbol{\Lambda}_{qi}+\boldsymbol{\Lambda}_{qj})^{-1}$ and $\tilde{\boldsymbol{\mu}} = \tilde{\boldsymbol{\Lambda}}^{-1}(\boldsymbol{\Lambda}_{qi}\boldsymbol{\mu}_{qi}+\boldsymbol{\Lambda}_{qj}\boldsymbol{\mu}_{qj})$, to be the identity for the product of two Gaussian distributions. Then, we have that $\text{cov}_{ij}^{f}(\boldsymbol{x},\boldsymbol{x}')$

$$
\begin{aligned}
&= \sum_{q\in\{0,i,j\}} \xi_q^2\omega_{ij}^q \int_{-\infty}^{\infty} \mathcal{N}(\boldsymbol{u}|-\boldsymbol{d},\boldsymbol{\Lambda}_{qi}^{-1})\mathcal{N}(\boldsymbol{u}|\boldsymbol{0},\boldsymbol{\Lambda}_{qj}^{-1})d\boldsymbol{u}\\
&= \sum_{q\in\{0,i,j\}} \xi_q^2\omega_{ij}^q \int_{-\infty}^{\infty} \mathcal{N}(\boldsymbol{u}|-\boldsymbol{d},\boldsymbol{\Lambda}_{qi}^{-1}+\boldsymbol{\Lambda}_{qj}^{-1})\mathcal{N}(\boldsymbol{u}|\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Lambda}})d\boldsymbol{u}\\
&= \sum_{q\in\{0,i,j\}} \xi_q^2\omega_{ij}^q \mathcal{N}(\boldsymbol{d}|\boldsymbol{0},\boldsymbol{\Lambda}_{qi}^{-1}+\boldsymbol{\Lambda}_{qj}^{-1})\\
&= \sum_{q\in\{0,i,j\}} \xi_q^2\tilde{\omega}_{ij}^q \text{exp}(-\frac{1}{2}\boldsymbol{d}^\top {\boldsymbol{\Phi}_{ij}^q}^{-1}\boldsymbol{d}),
\end{aligned}
\tag{11}
$$

where $\omega_{ij}^q = \alpha_{qi}\alpha_{qj}(4\pi)^{\frac{D}{2}}|\boldsymbol{\Lambda}_{qi}|^{-\frac{1}{4}}|\boldsymbol{\Lambda}_{qj}|^{-\frac{1}{4}}$, $\tilde{\omega}_{ij}^q = 2^{\frac{D}{2}}\alpha_{qi}\alpha_{qj}|\boldsymbol{\Lambda}_{qi}|^{\frac{1}{4}}|\boldsymbol{\Lambda}_{qj}|^{\frac{1}{4}}/|\boldsymbol{\Lambda}_{qi}+\boldsymbol{\Lambda}_{qj}|^{\frac{1}{2}}$, and ${\boldsymbol{\Phi}_{ij}^q}^{-1} = (\boldsymbol{\Lambda}_{qi}^{-1}+\boldsymbol{\Lambda}_{qj}^{-1})^{-1} = \boldsymbol{\Lambda}_{qi}(\boldsymbol{\Lambda}_{qi}+\boldsymbol{\Lambda}_{qj})^{-1}\boldsymbol{\Lambda}_{qj}$. A nice feature of (11), is that the marginal process i.e. $\text{cov}_{ii}^{f}(\boldsymbol{x},\boldsymbol{x}') = \sum_{q=\{0,i\}}\xi_q^2\alpha_{qi}^2\text{exp}(-\frac{1}{4}\boldsymbol{d}^\top\boldsymbol{\Lambda}_{qi}\boldsymbol{d})$ has the most common Gaussian covariance function resulting from the convolution of two Gaussian kernels. Therefore, (11) can be viewed as the extension of the Gaussian covariance function to the multivariate case. Once again we note that (11) is a wrapper function where for $i \neq j$, we have that $\text{cov}_{ij}^{f}(\boldsymbol{x},\boldsymbol{x}') = \xi_0^2\tilde{\omega}_{ij}^0\text{exp}(-\frac{1}{2}\boldsymbol{d}^\top{\boldsymbol{\Phi}_{ij}^0}^{-1}\boldsymbol{d})$.

Now, we let $\boldsymbol{\theta}_{f_{ij}} \in \Theta_{f_{ij}}$ represent the parameters in $\text{cov}_{ij}^{f}(\boldsymbol{x},\boldsymbol{x}')$ where $\Theta_{f_{ij}}$ is a set that contains the true parameters $\boldsymbol{\theta}_{f_{ij}}^*$, and we denote $\boldsymbol{\theta}_{ij}^\top = \{\boldsymbol{\theta}_{f_{ij}}^\top, \boldsymbol{\sigma}_{ij}^\top\}^\top$, where $\boldsymbol{\sigma}_{ij} = \{\sigma_i,\sigma_j\}^\top$, to be the set of all parameters in the bivariate submodel. Then, given our CP formulation and following (8), the marginal density of the bivariate submodel is expressed as $\text{pr}(\boldsymbol{y}_i,\boldsymbol{y}_j|\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{\theta}_{ij}) =$

$$
\begin{aligned}
&\int \text{pr}(\boldsymbol{y}_{ij}|\boldsymbol{f}_{ij})\,\text{pr}(\boldsymbol{f}_{ij}|\boldsymbol{\theta}_{f_{ij}})d\boldsymbol{f}_{ij}\\
&= \int \prod_{c=1}^{p_i} pr(y_i^c|f_i^c) \prod_{c'=1}^{p_j} \text{pr}(y_j^{c'}|f_j^{c'})\,\text{pr}(\boldsymbol{f}_{ij}|\boldsymbol{\theta}_{f_{ij}})d\boldsymbol{f}_{ij},
\end{aligned}
\tag{12}
$$

where $\text{pr}(\boldsymbol{f}_{ij}|\boldsymbol{\theta}_{f_{ij}}) = \mathcal{N}(\boldsymbol{0}_{P'},\boldsymbol{C}_{\boldsymbol{f}_{ij},\boldsymbol{f}_{ij}})$, $\text{pr}(\boldsymbol{y}_{ij}|\boldsymbol{f}_{ij}) = \mathcal{N}(\boldsymbol{0}_{P'},\boldsymbol{\Sigma}_{ij})$ and $\text{p} = p_i + p_j$. Therefore $\text{pr}(\boldsymbol{y}_i,\boldsymbol{y}_j|\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{\theta}_{ij}) = \mathcal{N}(\boldsymbol{0}_{\text{p}},\boldsymbol{C}_{\boldsymbol{f}_{ij},\boldsymbol{f}_{ij}}+\boldsymbol{\Sigma}_{ij})$ where

$$
\boldsymbol{C}_{\boldsymbol{f}_{ij},\boldsymbol{f}_{ij}}+\boldsymbol{\Sigma}_{ij} = \begin{pmatrix}\boldsymbol{C}_{\boldsymbol{f}_i,\boldsymbol{f}_i} & \boldsymbol{C}_{\boldsymbol{f}_i,\boldsymbol{f}_j}\\ \boldsymbol{C}_{\boldsymbol{f}_i,\boldsymbol{f}_j}^\top & \boldsymbol{C}_{\boldsymbol{f}_j,\boldsymbol{f}_j}\end{pmatrix} + \begin{pmatrix}\sigma_i^2\boldsymbol{I}_{p_i} & \boldsymbol{0}\\ \boldsymbol{0} & \sigma_j^2\boldsymbol{I}_{p_j}\end{pmatrix},
\tag{13}
$$

such that $\boldsymbol{C}_{\boldsymbol{f}_{ij},\boldsymbol{f}_{ij}} \in \mathcal{R}^{\text{p}\times\text{p}}$ is the covariance matrix relating all input points of outputs i and j with $\text{cov}_{ij}^{f}(\boldsymbol{x},\boldsymbol{x}')$ in (10) and (11). As previously mentioned, parameter estimates are obtained from minimizing the negative log-likelihood function $\ell(\boldsymbol{\theta}_{ij};\mathcal{D}_i,\mathcal{D}_j) = -\log\ \text{pr}(\boldsymbol{y}_i,\boldsymbol{y}_j|\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{\theta}_{ij})$. Denoting $\boldsymbol{Y}_{ij} = \boldsymbol{y}_{ij}\boldsymbol{y}_{ij}^\top$, up to an additive constant, the bivariate likelihood and its derivatives are given as

$$
\begin{aligned}
\ell(\boldsymbol{\theta}_{ij};\mathcal{D}_i,\mathcal{D}_j) = &\frac{1}{2}\langle\boldsymbol{Y}_{ij},(\boldsymbol{C}_{\boldsymbol{f}_{ij},\boldsymbol{f}_{ij}}+\boldsymbol{\Sigma}_{ij})^{-1}\rangle\\
&+\frac{1}{2}\log|\boldsymbol{C}_{\boldsymbol{f}_{ij},\boldsymbol{f}_{ij}}+\boldsymbol{\Sigma}_{ij}|\ .
\end{aligned}
\tag{14}
$$

Further, through denoting $\boldsymbol{C}_{ij} \triangleq \boldsymbol{C}_{\boldsymbol{y}_{ij},\boldsymbol{y}_{ij}} = \boldsymbol{C}_{\boldsymbol{f}_{ij},\boldsymbol{f}_{ij}}+\boldsymbol{\Sigma}_{ij}$, $\boldsymbol{\Psi}_{ij} = \boldsymbol{C}_{ij}^{-1}\boldsymbol{y}_{ij}$ and $\boldsymbol{\Xi}_{nm} = \frac{\partial\boldsymbol{C}_{ij}}{\partial\theta_{ij}^{(n)}}\boldsymbol{C}_{ij}^{-1}\frac{\partial\boldsymbol{C}_{ij}}{\partial\theta_{ij}^{(m)}}$ the gradient and second derivatives with respect to any parameter $\theta_{ij}^{(n)} \in \boldsymbol{\theta}_{ij}$ are then given as (in Appendix B we expand on $\partial\boldsymbol{C}_{ij}/\partial\theta_{ij}^{(n)}$)

$$
\begin{aligned}
\frac{\partial\ell}{\partial\theta_{ij}^{(n)}} &= \frac{1}{2}\left\langle\boldsymbol{\Psi}_{ij}\boldsymbol{\Psi}_{ij}^\top - \boldsymbol{C}_{ij}^{-1}, \frac{\partial\boldsymbol{C}_{ij}}{\partial\theta_{ij}^{(n)}}\right\rangle\\
\frac{\partial^2\ell}{\partial\theta_{ij}^{(n)}\partial\theta_{ij}^{(m)}} &= \frac{1}{2}\left\langle\boldsymbol{\Psi}_{ij}\boldsymbol{\Psi}_{ij}^\top - \boldsymbol{C}_{ij}^{-1}, \left(\frac{\partial^2\boldsymbol{C}_{ij}}{\partial\theta_{ij}^{(n)}\partial\theta_{ij}^{(m)}} - \boldsymbol{\Xi}_{nm}\right)\right\rangle\\
&\quad -\frac{1}{2}\left\langle\boldsymbol{\Psi}_{ij}\boldsymbol{\Psi}_{ij}^\top, \boldsymbol{\Xi}_{mn}\right\rangle.
\end{aligned}
$$

The computational complexity of learning from the bivariate likelihood in (14) is reduced to $\mathcal{O}((2p)^3)$. More importantly, the total number of parameters in the model is reduced to $4(1+D)+2+3$ where 4 represents the number of kernels $(K_{ii},K_{jj},K_{0i},K_{0j})$ multiplied by the number of parameters $(1+D)$ in each kernel, while 2 represents σ_i and σ_j, and 3 represents the parameters (ξ_0,ξ_i,ξ_j) of the latent functions (X_0,X_i,X_j) respectively. Note that reductions in parameter number can be done through assuming $K_{0i} = K_{0j}$ or $\sigma_i = \sigma_j$. As shown distributed estimation using bivariate submodels, is able to handle both the computational complexity and large parameter number to be estimated. Also, all pairwise models can be parallelized and thus our model can scale to an arbitrarily large number of outputs by parallelization.

Now, recall Figure (5), X_0, defined by ξ_0, represents the latent function which facilitates the sharing of information between outputs i and j. Therefore, in order to handle negative transfer of knowledge we use the bivariate likelihood in (14) but with ξ_0 penalized. Following Lemma 2, we will show in the following section that shrinking ξ_0 decreases the cross correlation amongst the outputs and that $\xi_0 = 0$ ensures that each output is predicted independently. The penalized negative log-likelihood function is defined as

$$
\ell_{\mathbb{P}}(\boldsymbol{\theta}_{ij};\mathcal{D}_i,\mathcal{D}_j,\lambda) = \ell(\boldsymbol{\theta}_{ij};\mathcal{D}_i,\mathcal{D}_j) + \mathbb{P}_\lambda(|\xi_0|)\ ,
\tag{15}
$$

where $\mathbb{P}_\lambda(\xi_0)$ is a penalty function. Different types of penalty functions can be used, examples include: ridge penalty $\mathbb{P}_\lambda(|\xi_0|) = \lambda\xi_0^2$, ℓ_1 penalty $\mathbb{P}_\lambda(|\xi_0|) = \lambda|\xi_0|$, bridge penalty $\mathbb{P}_\lambda(|\xi_0|) = \lambda|\xi_0|^{0<\cdot<1}$, and scad penalty $\mathbb{P}_\lambda(|\xi_0|) = \lambda|\xi_0|$ if $|\xi_0| \leq \lambda$, $(\xi_0^2-2\gamma\lambda|\xi_0|+\lambda^2)/(2\gamma-2)$ if $\lambda < |\xi_0| \leq \gamma\lambda$, $\lambda^2(\gamma+1)/2$ if $|\xi_0| > \gamma\lambda$. The tuning parameter λ (λ and γ in Scad) has an important effect on predictions. For instance

in the Lasso, as λ increases, ξ_0 shrinks to zero. Typically, the optimal tuning parameter is found using a grid search such as generalized cross validation (GCV), specifically b-fold GCV [47]. The GCV is based on splitting the data into b groups. For a given λ, first, we exclude one group as the testing dataset, and the rest $b-1$ groups are used as the training data, from which predictions at test locations are obtained. This is repeated b times for each group. The tuning parameter is then chosen as the minimizer of some accuracy measure (ex: absolute error, mean squared error, etc..) based on the GCV procedure.

Prior to discussing the statistical properties of our model, we note that the pairwise model $\mathrm{pr}(\boldsymbol{y}_i, \boldsymbol{y}_j|\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{\theta}_{ij}) = \mathcal{N}(\mathbf{0}_{\mathrm{p}}, \boldsymbol{C}_{\boldsymbol{y}_{ij},\boldsymbol{y}_{ij}})$ is in itself an MGP. Thus, any sparse MGP approximation can be directly used to make the pairwise model faster, be it an inducing point/variational approximation, a state space approximation, a matrix tapering approach or just a faster matrix inversion/determinant calculation scheme. For example, considering $\boldsymbol{u}$ inducing variables, one can directly implement the partially independent training conditional approximation (PITC) [18], where $\mathrm{pr}(\boldsymbol{y}_i, \boldsymbol{y}_j|\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{\theta}_{ij}) \approx \mathcal{N}(\mathbf{0}_{\mathrm{p}}, \mathrm{blockdiag}(\boldsymbol{C}_{\boldsymbol{y}_{ij},\boldsymbol{y}_{ij}} - \boldsymbol{C}_{\boldsymbol{y}_{ij},\boldsymbol{u}}\boldsymbol{C}^{-1}_{\boldsymbol{u},\boldsymbol{u}}\boldsymbol{C}_{\boldsymbol{u},\boldsymbol{y}_{ij}})) + \boldsymbol{C}_{\boldsymbol{y}_{ij},\boldsymbol{u}}\boldsymbol{C}^{-1}_{\boldsymbol{u},\boldsymbol{u}}\boldsymbol{C}_{\boldsymbol{u},\boldsymbol{y}_{ij}})$ or even just the Nystrom approximation $pr(\boldsymbol{y}_i, \boldsymbol{y}_j|\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{\theta}_{ij}) \approx \mathcal{N}(\mathbf{0}_{\mathrm{p}}, \boldsymbol{C}_{\boldsymbol{y}_{ij},\boldsymbol{u}}\boldsymbol{C}^{-1}_{\boldsymbol{u},\boldsymbol{u}}\boldsymbol{C}_{\boldsymbol{u},\boldsymbol{y}_{ij}})$.

4.2 Statistical properties

We now discuss some of the structural properties of our model. We first introduce the main theorem which guarantees that our regularized model in (15) is able to avoid the negative transfer of knowledge.

Theorem 1. *Suppose that $\xi_0 = 0$, then the predictive distribution of the bivariate model at any new input $\boldsymbol{x}_0 \in \mathcal{X}$ reduces to that of a univariate model where $\mathrm{pr}(y_i(\boldsymbol{x}_0)|\boldsymbol{y}_{ij}) = \mathrm{pr}(y_i(\boldsymbol{x}_0)|\boldsymbol{y}_i)$ and $\mathrm{pr}(y_j(\boldsymbol{x}_0)|\boldsymbol{y}_{ij}) = \mathrm{pr}(y_j(\boldsymbol{x}_0)|\boldsymbol{y}_j)$ such that*

$$\begin{cases} \mathrm{pr}(y_i(\boldsymbol{x}_0)|\boldsymbol{y}_{ij}) = \mathcal{N}\Big(\boldsymbol{C}^{\top}_{\boldsymbol{f}_i,f^0_i}\boldsymbol{\Omega}_{ii}\boldsymbol{y}_i,\, C_{f^0_i,f^0_i} + \\ \qquad\qquad \sigma^2_i - \boldsymbol{C}^{\top}_{\boldsymbol{f}_i,f^0_i}\boldsymbol{\Omega}_{ii}\boldsymbol{C}_{\boldsymbol{f}_i,f^0_i}\Big) \\ \mathrm{pr}(y_j(\boldsymbol{x}_0)|\boldsymbol{y}_{ij}) = \mathcal{N}\Big(\boldsymbol{C}^{\top}_{\boldsymbol{f}_j,f^0_j}\boldsymbol{\Omega}_{jj}\boldsymbol{y}_j,\, C_{f^0_j,f^0_j} + \\ \qquad\qquad \sigma^2_j - \boldsymbol{C}^{\top}_{\boldsymbol{f}_j,f^0_j}\boldsymbol{\Omega}_{jj}\boldsymbol{C}_{\boldsymbol{f}_j,f^0_j}\Big) \end{cases}$$

where for $c \in \{i,j\}$, $\boldsymbol{\Omega}_{cc} = (\boldsymbol{C}_{\boldsymbol{f}_c,\boldsymbol{f}_c} + \sigma^2_c\boldsymbol{I}_{p_c})^{-1} \in \mathcal{R}^{p_c\times p_c}$, $\boldsymbol{C}_{\boldsymbol{f}_c,f^0_c} = [cov^f_{cc}(\boldsymbol{x}_0,\boldsymbol{x}_{c1}), ..., cov^f_{cc}(\boldsymbol{x}_0,\boldsymbol{x}_{cp_c})]^{\top}$, $C_{f^0_c,f^0_c} = cov^f_{cc}(\boldsymbol{x}_0,\boldsymbol{x}_0)$ and $cov^f_{cc}(\boldsymbol{x},\boldsymbol{x}') = \xi^2_c\int_{-\infty}^{\infty} K_{cc}(\boldsymbol{u})K_{cc}(\boldsymbol{u}-\boldsymbol{d})d\boldsymbol{u}$.

The proof is detailed in Appendix C. The key feature of this theorem is that penalizing only one variable, in our initial parameter set $\xi_0 \in \boldsymbol{\theta}_{f_{ij}} \subset \boldsymbol{\theta}_{ij}$, will lead to separating the bivariate model into two models equivalent to the univariate GP established through a CP in (4). Our regularization approach is flexible to any specified kernel function and not based on the Gaussian covariance derived in (11), where theorem 1 holds for any valid kernel function K_{iq}. We note that the result of theorem 1, is based on the fact that for $c \in \{i,j\}$ as $\xi_0 \to 0$

$$\begin{cases} \mathrm{cov}^f_{ij}(\boldsymbol{x},\boldsymbol{x}') = \xi^2_0\int_{-\infty}^{\infty} K_{0i}(\boldsymbol{u})K_{0j}(\boldsymbol{u}-\boldsymbol{d})d\boldsymbol{u} \to 0 \\ \mathrm{cov}^f_{cc}(\boldsymbol{x},\boldsymbol{x}') = \sum\limits_{q\in\{0,c\}}\xi^2_q\int_{-\infty}^{\infty} K_{qi}(\boldsymbol{u})K_{qi}(\boldsymbol{u}-\boldsymbol{d})d\boldsymbol{u} \\ \qquad\qquad \to \xi^2_c\int_{-\infty}^{\infty} K_{cc}(\boldsymbol{u})K_{cc}(\boldsymbol{u}-\boldsymbol{d})d\boldsymbol{u}. \end{cases}$$

This is important to note since non-sparse penalties such as the ridge penalty can still minimize the negative transfer of knowledge through shrinking ξ_0. This however comes at the expense of variable selection implied in sparse penalties.

Next, we discuss some asymptotic properties of our regularized bivariate model. In order to investigate the asymptotic properties of our regularized model, we first need to examine the properties of $\boldsymbol{\theta}_{ij}$ obtained from minimizing the unpenalized likelihood in (14). Here we note that one advantage of our model is that for both the penalized $\ell_{\mathbb{P}}(\boldsymbol{\theta}_{ij})$ and unpenalized likelihood $\ell(\boldsymbol{\theta}_{ij})$ we are minimizing over the same set of parameters $\boldsymbol{\theta}_{ij}$ since $\xi_0 \in \boldsymbol{\theta}_{ij}$. Based on mild regularity conditions for dependent observations, it has been shown that the maximum likelihood estimator of $\boldsymbol{\theta}_{ij}$ is r_{p} consistent. Refer to Appendix D, for more details [48]–[50]. Now for the penalized model, let $\boldsymbol{\theta}^{*\,t}_{ij} = \{\boldsymbol{\theta}^{*\,t}_{f_{ij}}, \boldsymbol{\sigma}^{*\,t}_{ij}\}^{\top}$ be the true parameter values corresponding to $\boldsymbol{\theta}^{\top}_{ij} = \{\boldsymbol{\theta}^{\top}_{f_{ij}}, \boldsymbol{\sigma}^{\top}_{ij}\}^{\top}$, and let $\hat{\boldsymbol{\theta}}_{ij}$ be the estimated parameters obtained from minimizing $\ell_{\mathbb{P}}(\boldsymbol{\theta}_{ij})$. Hence ξ^*_0 and $\hat{\xi}_0$ respectively represent the true and estimated value of ξ_0 . For the penalty function $\mathbb{P}_\lambda(|\xi_0|)$, we assume that the penalty is non-negative; $\mathbb{P}_\lambda(|\xi_0|) \geq 0$ and $\mathbb{P}_\lambda(0) = 0$, and that larger coefficients are penalized no less than smaller ones; $\mathbb{P}_\lambda(|\xi'_0|) \geq \mathbb{P}_\lambda(|\xi_0|)$ if $|\xi'_0| \geq |\xi_0|$. These are typical assumptions and are satisfied by the aforementioned penalties [51]. Further, we assume that the first and second derivatives of $\mathbb{P}_\lambda(|\xi_0|)$ are continuous at $\xi^*_0 \neq 0$. We next provide two theorems that establish parameter estimation and selection consistency. The theorems provide similar results as in [51], but defined within our model specifications and based on dependent observations. Note that the "$\prime$" notation on a function implies a derivative.

Theorem 2. *If $z_2 = max\{|\mathbb{P}''_\lambda(|\xi^*_0|)| : \xi^*_0 \neq 0\} \to 0$, then there exits a local minimzer $\hat{\boldsymbol{\theta}}_{ij}$ for $\ell_{\mathbb{P}}(\boldsymbol{\theta}_{ij})$, such that $||\hat{\boldsymbol{\theta}}_{ij} - \boldsymbol{\theta}^*_{ij}|| = O(r^{-1}_{\mathrm{p}} + z_1)$, where $z_1 = max\{\mathbb{P}'_\lambda(|\xi^*_0|) : \xi^*_0 \neq 0\}$.*

Theorem 3. *Assume that $\xi^*_0 = 0$ and the parameters $\ddot{\boldsymbol{\theta}}_{ij} = \{\boldsymbol{\theta}_{ij}\}/\{\xi_0\}$ satisfy r_{p} consistency in theorem 2. Then if $\lim\limits_{\mathrm{p}\to\infty} inf \lim\limits_{\xi_0\to 0^+} inf\ \frac{1}{\lambda}\mathbb{P}'_\lambda(\xi_0) > 0$, $\lambda \to 0$ and $\mathrm{p}\lambda/r_{\mathrm{p}} \to \infty$ as $\mathrm{p} \to \infty$, we have that $\lim\limits_{\mathrm{p}\to\infty} \mathrm{pr}(\hat{\xi}_0 = 0) = 1$.*

The proof for theorems 2 and 3 is detailed in Appendix D. In the above theorem "*lim inf*" denoted the infimum of the limit points. It is clear from theorem 2 and 3, that if we choose a proper tuning parameter λ and penalty function $\mathbb{P}_\lambda(|\xi_0|)$ there exists an r_{p} consistent estimator for the penalized likelihood $\ell_{\mathbb{P}}(\boldsymbol{\theta}_{ij})$, which possesses the sparsity property $\hat{\xi}_0 = 0$, i.e. asymptotically performs as well as knowing that $\xi_0 = 0$ beforehand. This result is also known as an oracle property which provides consistency in variable selection [52]. Here variable selection implies select-

ing whether functions should be predicted independently or not, a stated in theorem 1.

4.3 Application to separable covariance

In this section, we provide a direct approach to applying our regularized pairwise approach to separable modeling. Here we recall that a key aspect of separable covariance is that all functions share the same marginal covariance, i.e. the within correlation function. Thus, negative transfer is inevitable due to this restriction. However, through regularizing the between-output correlation matrix, one can decrease negative transfer where some pairs can be predicted independently as shown here. The covariance function in a separable model is of the form $\text{cov}_{ij}^{f}(\boldsymbol{x}, \boldsymbol{x}') = T_{ij}\text{cov}(\boldsymbol{x}, \boldsymbol{x}')$, where T_{ij} is the between-output covariance matrix and $\text{cov}(.,.)$ is a covariance function over inputs $\boldsymbol{x} \in \mathcal{R}^D$, the same for all outputs. For $\text{cov}_{ij}^{f}(\boldsymbol{x}, \boldsymbol{x}')$ to be a valid covariance function, it is required that $T_{ij} = \{t_{c,c'}\}$ be a positive definite matrix with unit diagonal elements (PDUDE) [9]. Therefore, in a bivariate case with two outputs i and j, $T_{ij} = \begin{pmatrix} 1 & t_{ij} \\ t_{ij} & 1 \end{pmatrix}$, such that t_{ij} measures the correlation between output i and j. It is interesting to note that the covariance between outputs only varies through t_{ij}, this is why outputs in separable functions are instantaneously mixed as they are directly derived by a scaling or a rotation to an output space of dimension D. Inspired by Theorem 1 and Appendix C, one can directly show that $t_{ij} = 0$ ensures that each output is predicted independently. Therefore, in separable modeling, we only need to adjust the penalty function to $\mathbb{P}_\lambda(|t_{ij}|)$, and optimize the penalized likelihood while restricting T_{ij} to be PDUDE. It is interesting to mention here the intrinsic coregionalization (IC) model [53] which is a simplified version of the LMC previously mentioned in the introduction. The IC is a separable construction that reduces to independent predictions over each output under an isotopic data case and if outputs are modeled as noise free. Unfortunately, despite its ability to avoid negative transfer, such a model cannot make use of commonalities across outputs.

4.4 Combining the predictions

Without loss of generality, we focus on predicting output N through sharing information from the remaining $N-1$ outputs as shown in Figure 4. Based on (3), for each submodel in Figure 4, the predictive equation for any new input $\boldsymbol{x}_0 \in \mathcal{X}$ for output N is expressed as

$$\begin{aligned}\text{pr}(y_N(\boldsymbol{x}_0)|\boldsymbol{y}_{iN}) = \mathcal{N}\Big(\boldsymbol{C}_{\boldsymbol{f}_{iN},f_N^0}^{\top}\boldsymbol{\Omega}_{iN}\boldsymbol{y}_{iN},\ C_{f_i^N,f_i^N} + \\ \sigma_N^2 - \boldsymbol{C}_{\boldsymbol{f}_{iN},f_N^0}^{\top}\boldsymbol{\Omega}_{iN}\boldsymbol{C}_{\boldsymbol{f}_{iN},f_N^0}\Big),\end{aligned} \tag{16}$$

where $i \in \mathcal{I}^{-N} = \{1, ..., N-1\}$, $\boldsymbol{\Omega}_{iN} = (\boldsymbol{C}_{\boldsymbol{f}_{iN},\boldsymbol{f}_{iN}} + \boldsymbol{\Sigma}_{iN})^{-1}$, $\boldsymbol{C}_{\boldsymbol{f}_{iN},f_N^0} = [\boldsymbol{C}_{\boldsymbol{f}_i,f_N^0}^{\top}, \boldsymbol{C}_{\boldsymbol{f}_N,f_N^0}^{\top}]^{\top}$ and $\boldsymbol{C}_{\boldsymbol{f}_c,f_N^0} = [\text{cov}_{Nc}^{f}(\boldsymbol{x}_0, \boldsymbol{x}_{c1}), ..., \text{cov}_{Nc}^{f}(\boldsymbol{x}_0, \boldsymbol{x}_{cp_c})]^{\top}$ for $c \in \{i, N\}$. Our goal is to efficiently combine the predictions from the $N-1$ bivariate submodels to form an overall result. To this end, we utilize the product of GP experts (PoE) model, used in univariate GP's, however implemented within the specifications of our pairwise model [42], [54]. Here, we aim to combine predictions from $N-1$ "experts", where each expert is a regularized bivariate GP. The PoE model combines the predictions by the product of all expert predictions. In our pairwise model, the PoE implies that $\bar{\text{pr}}(y_N(\boldsymbol{x}_0)|\boldsymbol{y}) = \prod_{c=1}^{N-1}\text{pr}(y_N(\boldsymbol{x}_0)|\boldsymbol{y}_{cN})$. PoE models are straightforward and theoretically appealing as each expert is weighted by the inverse covariance, therefore experts which are uncertain about their predictions are automatically weighted less than experts that are certain about their predictions. However, a major shortcoming of PoE models is that as N increases, the combined prediction tends to be overconfident. For instance, assume that all functions in $\mathcal{I}^{-N}$ are exactly equivalent, then we have that $\text{pr}(y_N(\boldsymbol{x}_0)|\boldsymbol{y}_{cN}) = \mathcal{N}(\mathcal{M}, \mathcal{V})\ \forall c \in \mathcal{I}^{-N}$ for some mean $\mathcal{M}$ and variance $\mathcal{V}$. Therefore, $\bar{\text{pr}}(y_N(\boldsymbol{x}_0)|\boldsymbol{y}) = \mathcal{N}(\bar{\mathcal{M}} = \mathcal{M}, \bar{\mathcal{V}} = \mathcal{V}/(N-1))$ and as $N \to \infty \implies \bar{\mathcal{V}} \to 0$. Naturally, in such a case we would want $\bar{\text{pr}}(y_N(\boldsymbol{x}_0)|\boldsymbol{y}) = \mathcal{N}(\bar{\mathcal{M}} = \mathcal{M}, \bar{\mathcal{V}} = \mathcal{V})$. To this end, we weight the contributions of each expert with a weight $\beta_c = 1/(N-1)$ for $c \in \mathcal{I}^{-N}$. As a result, given that $\text{pr}(y_N(\boldsymbol{x}_0)|\boldsymbol{y}_{cN}) = \mathcal{N}(\mathcal{M}_c, \mathcal{V}_c)\ \forall c \in \mathcal{I}^{-N}$, and following the identity that the product of Gaussian distributions is Gaussian, we have that $\bar{\text{pr}}(y_N(\boldsymbol{x}_0)|\boldsymbol{y}) = \mathcal{N}(\bar{\mathcal{M}}, \bar{\mathcal{V}})$, where

$$\bar{\mathcal{V}}^{-1} = \sum_{c=1}^{N-1}\beta_c\mathcal{V}_c^{-1}, \quad \bar{\mathcal{M}} = \bar{\mathcal{V}}\sum_{c=1}^{N-1}\beta_c\mathcal{V}_c^{-1}\mathcal{M}_c\ , \tag{17}$$

This efficient closed form inference for combining the bivariate models is independent of the computational graph, and, consequently, facilitates the ability to scale to arbitrarily large datasets by parallelization, where each bivariate model is efficiently built with a limited of parameters and a small-scale covariance matrix. Note that (17) is similar to the log opinion pool model [55] and the Generalized product of experts [42], [56], therefore, the key feature of the PoE model is still retained as experts that are uncertain about their predictions are weighted less, also, since $\sum_{c\in\mathcal{I}^{-N}}\beta_c = 1$, then it ensures a consistent model that falls back to the prior. Some slight modifications to the traditional PoE have been also proposed such as the Bayesian Committee Machine (BCM) or the robust BCM [41], [57] . The BCM is based on adjusting the variance of the unweighted PoE, by a prior variance $\text{pr}(y_N(\boldsymbol{x}_0))$, while the robust BCM adjusts the BCM with weights similar to those in our model. However, the BCM is based upon assuming a block diagonal covariance where all experts share the same parameters, which hinders its application in our model.

5 Numerical Case Studies

We conduct case studies to demonstrate the advantageous features of our regularized and distributed multivariate Gaussian convolution process denoted as MGCP-RD. In Section 5.1, we discuss benchmarked methods and the general setting for our numerical case studies. Then Section 5.2 uses simulated functions to demonstrate the performance of the proposed method under four different model settings. Further an illustrative example is provided in Section 5.2.

5.1 General settings

In this section, we discuss the settings used to assess the MGCP-RD performance using simulated data. To evaluate

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.2987482, IEEE Transactions on Pattern Analysis and Machine Intelligence

the performance of our proposed method, we randomly generate N signals from different model settings, in which the first $N-1$ outputs are used as a training set, while the Nth output is selected as the testing function. We repeat the study for $W = 1000$ times. For each replication, we use the mean absolute error (MAE) between the true signal value and its predicted value at $p_{\text{test}} = 50$ points as the criterion to evaluate our prediction accuracy. We then report the distribution of the MAE across the W replications using a group of boxplots, with respectiv represented as black dots. The MAE values are as prediction errors in all boxplots. Kernel param obtained through minimizing the negative log-li using a scaled congugate gradient algorithm ([[39]). For the MGCP-RD, we distribute the comp over only two systems, where each system was res for sequentially fitting $(N-1)/2$ of the bivariate All computations are done on R-3.2.2 in a 64-bit W 7 setting. Further, in our simulation studies, we be our method with four other reference methods for ison: 1) The individual GP established using a CP, as GCP, where the test function is fitted separatel The full MGCP model, denoted as MGCP, describe tion 3.2; 3) The inducing variable approximation, as MGCP-I, which tackles the computational co challenge [18], [32]; 4) The pairwise model for lon profiles, denoted as MGCP-P [39], [40], [43]. To provide consistent results we utilize the Gaussian kernel in Section 4.1 for each of the benchmarked methods. In the GCP, this kernel reduces to the Gaussian/squared exponential covariance function in the univariate case. For the MGCP-I and since no specific latent structure is proposed we use that of the full MGCP in Section 3.2. However, for the MGCP-P, we utilize the proposed latent structure in the paper which only involves one common latent function for each pairwise model. Finally, throughput the numerical study we use a 3-fold cross validation method to find the tuning parameters for our approach.

5.2 Results

We simulate functions from three different settings to demonstrate the benefits of the MGCP-RD. The model settings and results are shown below.

5.2.1 Setting I

In this setting, we aim to compare the performance of the MGCP-RD in a simple case with few number of outputs ($N = 5$) and no negative transfer. In order to establish a setting with no negative transfer, the multivariate output model for the N curves are generated according to the same functional form $y_i(x) = 1 + \sin(x) + \epsilon_i(x)$ for $x \in [0, 10]$ and $i \in \mathcal{I}$. The number of observations per signal is $p = p_1 =, .., = p_N = 10$ evenly spaced points, the test points are evenly spaced across $[0, 10]$ and measurement noise standard deviation is set to $\sigma = \sigma_1 =, .., = \sigma_N = 0.1$ for all outputs. For the MGCP-I, all p design points are used as inducing variables. Also, in this setting we implement the ridge penalization. The importance of this case is that the three challenges of nonseparable modeling can be readily handled in a low dimensional setting with no negative transfer. Therefore, this scenario is able to evaluate the performance of the MGCP-RD relative to the full MGCP. The results are shown in Figure 6.

The results in Figure (6) indicate that there is an insignificant difference in the prediction error between the full model (MGCP) and our proposed method. First, amongst the models that considered multiple outputs the MGCP-P had the worst performance. The MGCP-P is based on aver-

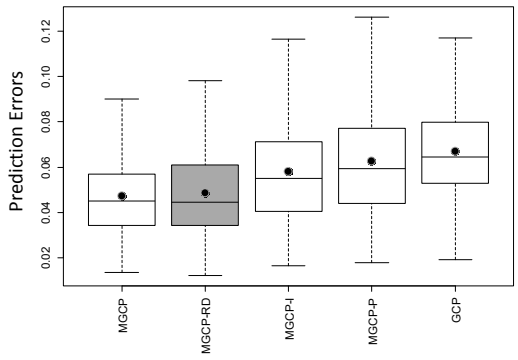


Fig. 6. Setting I results

As mentioned in [34], different parameters estimates correspond to different interpretations of the data, however, predictions will be moderately effected as the Bayesian correspondence in the GP implies that predictions should pass through or close to the design points. As will be shown in later settings, this approach of averaging parameter estimates becomes specifically dangerous with high noise levels, large parameter space and outputs with varying forms and characteristics where parameter estimates fluctuate widely between different submodels and iterations. Second, and following the intuition why we average predictions rather than parameter estimates, Figure (7) below shows the advantageous features of the weighted PoE model. In Figure (7), we compare the MGCP-RD results with MGCP-RD prediction errors before averaging the $N-1$ pairwise submodels from each iteration (denoted as BIVARIATE). Note that since we consider the marginal errors from each submodel in the BIVARIATE then we have $W(N-1)$ errors compared to MGCP-RD with G errors. As shown in the figure, the straightforward mechanism of PoE models, where experts that are uncertain about their predictions are automatically weighted less by the inverse covariance, provides both a simple yet efficient solution for distributed modeling.

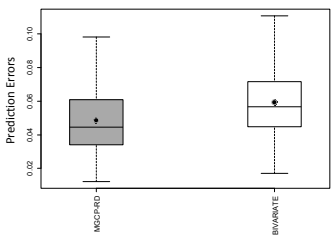


Fig. 7. PoE results

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.2987482, IEEE Transactions on Pattern Analysis and Machine Intelligence

5.2.2 Setting II

In this setting, we aim to compare the performance of the MGCP-RD in a case with a moderate number of outputs ($N = 50$) and no negative transfer. Also, we also aim to illustrate the importance of MGP models in extrapolation which has not been fully exploited in literature. We adopt the quadratic function example from [21] with some modifications. The multivariate output model for the N curves are generated according to $y_i(x) = 1 + e^{II}x^2 + \epsilon_i(x)$ for $i \in \mathcal{I}$ and $e^{II} \sim \text{uniform}(0.8, 1.2)$. The number of observations for the $N-1$ training output is $p = 20$ evenly spaced points for $x \in [0, 10]$, while for the Nth function to be predicted we generated $p = 10$ evenly spaced points for $x \in [0, 7]$. The test points are evenly spaced across $[0, 10]$ and measurement noise standard deviation is set to $\sigma = 1$ for all outputs. Due to the long model building time, only $W = 50$ iterations for the MGCP and MGCP-I are conducted. For the MGCP-I, all $p = 20$ design points in $[0, 10]$ are used as inducing variables. Also, in this setting we implement the ℓ_1 penalization. The importance of this case is that we are able to test all benchmarked models in a rather high dimensional parameter space with higher computational complexity. The results are shown in Figure 8, while an illustrative example

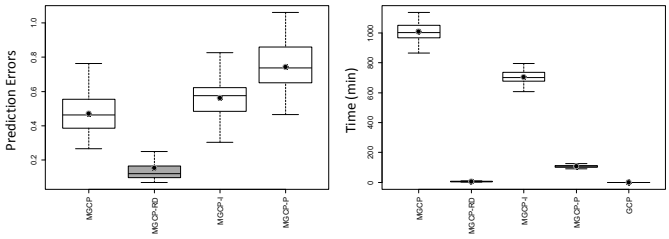


Fig. 8. Setting II results

Based on the results we can obtain some important insights. First, from a computational perspective, and as shown in Figure 8, the model building time of the MGCP and MGCP-I is extremely prohibitive. It takes on average 13 hours to build one MGCP model and this one dimensional input and a moderate number (($N = 50$). The mean building time for each iter half a second for the GCP, 6.8 minutes for the M and 104 minutes for the MGCP-P. Although the considers paired models however to predict the N we need to fit $N(N-1)/2$ pairwise submodels, method where only $N-1$ models need to be bu results illustrate why non-separable MGP model used in low dimensional settings. As mentioned p for the MGCP-RD, we parallelized computational 2 systems (25 models sequentially fit in each syst ever, with more computational power, the buildin the MGCP-RD can be significantly reduced.

Second, in addition to the severe computatio back of the MGCP, its predictive accuracy greatly in such a high dimensional space. This is intui derstandable, as minimizing the negative likelihood in such a high dimensional (4950) search space is a prohibitive task for any search algorithm. The MLE will directly get trapped in a local minima and will not be able to move even if different starting points are tried, therefore leading to suboptimal parameter estimates with undesirable properties. Besides that, the computation is plagued by numerical issues associated with inverting the 990×990 covariance matrix at each iteration of the search algorithm. Similar results have also been shown in [39], [40], [50], where MGP models tend to loose accuracy in a high dimensional parameter space. This issue is also faced in the MGCP-I, which is not able to address the large parameter space challenge, despite tackling the computational complexity where it only requires the inversion of 20×20 covariance matrices. It is important to note that multivariate statistical modeling often encounters functional data with $N >> 50$ [20], [58], thus the aforementioned drawbacks significantly increase in severity with more outputs.

Third, in Figure 9 below we compare our the MGCP-RD results in cases with different output number N under setting II specifications. As shown in the figure, as we increase N the prediction errors significantly decrease. This results further highlights the efficiency of the weighted PoE model specifically when N is large.

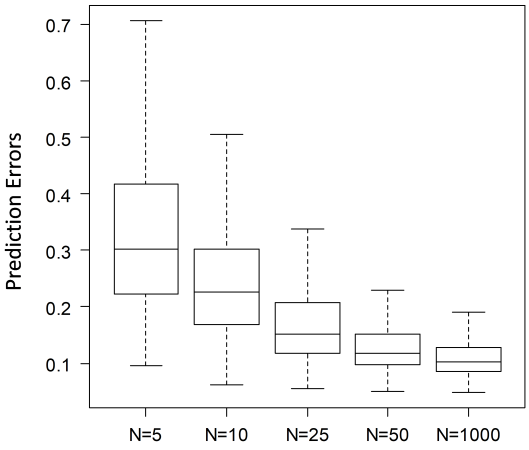


Fig. 9. PoE results with different N

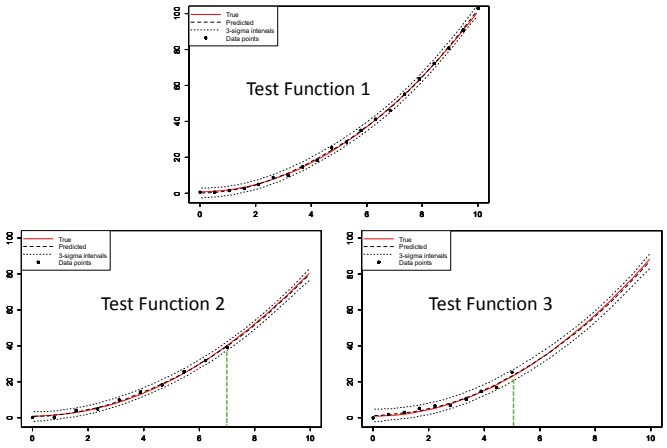


Fig. 10. Intrapolation vs extrapolation

Finally, in Figure 8 we do not report the GCP prediction accuracy to maintain scale, since the mean prediction error is 21.78. The reason for this poor performance is related to the

fact the GP in traditional settings cannot extrapolate as correlation goes to zero for data points that are far away from the observed data. However, the MGP is able to borrow strength from other observed outputs to predict the future evolution of a specific output. In other words, extrapolation in the MGP can be seen as interpolation across different output. To highlight this aspect, Figure 10 illustrates the MGCP-RD performance in extrapolation.

In Figure 10, for function 1 we perform interpolation, while for functions 2 and 3 we extrapolate following setting II specifications. However, in function 3, we generate $p = 10$ evenly spaced points for $x \in [0, 5]$ instead of $x \in [0, 7]$. The figure clearly shows that one main advantage of MGP models is the ability to extrapolate an output when other correlated outputs are observed over a larger domain. This advantage of MGP models indeed has many practical applications in cases where extrapolation might be also of interest to the user.

5.2.3 Setting III

In this setting, we aim to compare the performance of the MGCP-RD when negative transfer of knowledge exists. We establish a simple setting with few number of outputs ($N = 8$). Motivated by $M/M/1$ queuing systems, the multivariate output for the N functions is generated according to:

- $i \in \{1, 2, 3, 4\}$: $y_i^{(1)}(x) = x^2 + \epsilon_i(x)$ for $x \in [0, 0.8]$
- $i \in \{5, 6\}$: $y_i^{(2)}(x) = x^2/(2(1-x)) + \epsilon_i(x)$ for $x \in [0, 0.8]$
- $i \in \{7, 8\}$: $y_i^{(3)}(x) = x^2/(1-x) + \epsilon_i(x)$ for $x \in [0, 0.8]$

The number of observations per signal is $p = 7$ evenly spaced points for $x \in [0, 0.8]$, the test points are evenly spaced across $[0, 0.8]$ and measurement noise standard deviation is set to $\sigma = 0.005$ for all outputs. In this setting, if we assume x to be the system utilization, then $y_i^{(2)}(x)$ and $y_i^{(3)}(x)$ respectively define the steady state closed form equations of the expected Queue time and Queue length in an $M/M/1$ system, where the inter-arrival time is exponentially distributed with rate 2 [59]. All $p = 7$ design points in $[0, 0.8]$ are used as inducing variables for the MGCP-I and we implement the ℓ_1 penalization for the MGCP-RD. Following our general settings, for each iteration we find the MAE of the Nth function to be predicted which belongs to $y_i^{(3)}(x)$ and represents the expected queue length. Also, we benchmark with two other methods denoted as MGCP-Sep and Spectral. In MGCP-Sep, the MGCP is used to predict the Nth output using only the training signals with the same functional form, i.e. we fit outputs $i \in \{7, 8\}$ separately using the MGCP. While, Spectral is an MGP based on a recently proposed spectral mixture kernel in [60]. The Spectral method is added to check whether more expressive kernels such as the spectral mixture are able to automatically address negative transfer without using our proposed framework. Note that, the importance of our Model Setting III, is that the we are able to test all benchmarked models in a setting where outputs behave according to different functional forms. The results are shown in Figure 11.

The results in Figure 11 clearly illustrate the ability of our model to minimize negative transfer while borrowing strength from other correlated output. As shown in the figure, MGCP-Sep outperformed MGCP. This confirms that negative transfer is occurring since if outputs belonging to $y_i^{(3)}(x)$ are analyzed separately the results are better than the full model (MGCP). More interestingly, we have that MGCP-RD outperformed MGCP-Sep. The reason is that $y_i^{(3)}(x)$ and $y_i^{(2)}(x)$ are highly correlated, therefore, MGCP-RD was able to learn this cross correlation of $y_i^{(3)}(x)$ with $y_i^{(2)}(x)$ while at the same time avoiding negative transfer with $y_i^{(1)}(x)$. However, MGCP-Sep is not able to learn from the cross correlation between $y_i^{(3)}(x)$ and $y_i^{(2)}(x)$. Indeed, in this model setting we observe that $\hat{\xi}_0 = 0$ for pairwise models including $y_i^{(3)}(x)$ and $y_i^{(1)}(x)$ which indicates that these outputs possess no common features and should be predicted independently.

Also, we notice that Spectral behaved similar to MGCP and thus was not able to address the negative transfer of knowledge. This is expected as the spectral kernel possesses a large number of parameters, thus making optimization rather difficult and prone to become trapped in local minima and there is no incentive in the spectral kernel to penalize spurious correlations. This also illustrates that one can have models as expressive as needed to model within-output (auto) correlation however handling across-output (cross) correlation should be done with care to avoid negative transfer. Finally, one important observation is that MGCP-P behaved worse than GCP. As previously mentioned, averaging parameter estimates is specifically dangerous in cases with different functional forms as parameter estimates from different submodels will greatly fluctuate.

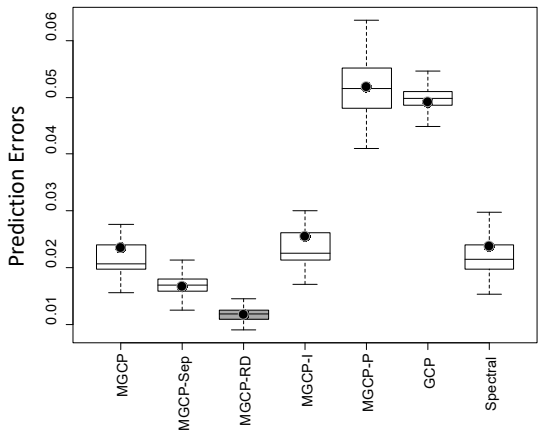


Fig. 11. Setting III results

5.2.4 Setting IV

In this setting, the goal is to predict the foreign exchange rate compared to the United States dollar currency. The data comes from the pacific exchange rate service (http://fx.sauder.ubc.ca/data.html). Our analysis utilized the exchange rates of the top ten international currencies (Canadian Dollar CAD/USD, Euro EUR/USD, Japanese Yen JPY/USD, Great British Pound GBP/USD, Swiss Franc CHF/USD, Australian Dollar AUD/USD, Hong Kong Dollar HKD/USD, New Zealand Dollar NZD/USD, South Koreon Won KRW/ USD, Mexican Peso MXN/USD) during the 52 weeks of the 2017 calender year. The data is illustrated in Figure 12. Each output is adjusted to have zero mean

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.2987482, IEEE Transactions on Pattern Analysis and Machine Intelligence

and unit variance. We use a leave-one-out cross validation approach to evaluate the performance of the MGCP-RD and the benchmarked methods. We iteratively treat one exchange currency rate as the test output and the remaining 9 currencies as the training set. This procedure is repeated for the 10 currencies. For each test output we randomly remove 13 data points (25% of the data from a specific output) and test the model capability to recover the true underlying values at these test points. For the MGCP, all remaining input points are used as inducing variables for the MGCP-I and we implement the ℓ_1 penalization for the MGCP-RD.

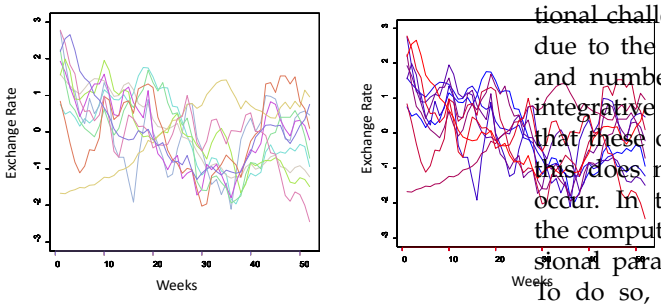


Fig. 12. Data illustration

First we provide some illustrative results in Figure 13. As shown in the figure, the MGCP-RD was able to able to efficiently recover the underlying truth. For performance accuracy comparison we also use the standardized mean square error (SMSE) defined in [34]. The results in in Table 1 show that the MGCP-RD was significantly able to outperform the benchmarked methods. This result in intuitively understandable as based on Figure 12, the trends display clear heterogeneity and thus negative transfer is a key issue when integratively modeling the exchange rates.

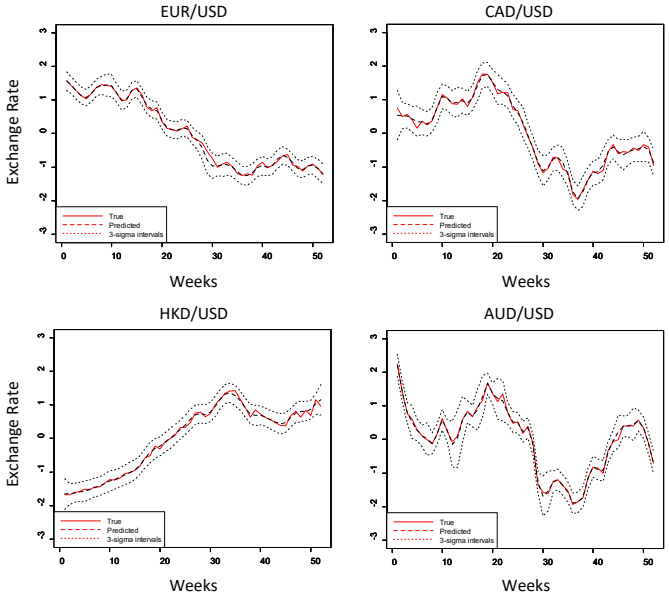


Fig. 13. MGCP-RD predictive results

6 Conclusion

MGP models established using a CP construction offer a general and flexible solution for multiple output regression.

TABLE 1
Setting IV results

	MGCP	MGCP-RD	MGCP-I	MGCP-P	GCP
MAE	0.203	0.156	0.218	0.952	0.247
std.error	(0.160)	(0.068)	(0.236)	(0.544)	(0.182)
SMSE	1.666	1.293	1.788	3.261	1.703
std.error	(1.783)	(0.861)	(2.250)	(6.002)	(1.935)

Despite that, the added flexibility arises serious computational challenges even with a moderate number of outputs due to the significant increase in computational demands and number of parameters to be estimated. Further, the integrative analysis of multiple outputs implicitly assumes that these outputs share some commonalities. However, if this does not hold, negative transfer of knowledge may occur. In this paper, we try to simultaneously address the computational (computational complexity, high dimensional parameter space) and negative transfer challenges. To do so, we propose a regularized pairwise modeling approach for MGCP models that has excellent scalability and minimizes the negative transfer of knowledge between uncorrelated outputs. The proposed approach is based on distributing MGCP estimation through bivariate GP submodels which are individually estimated. Predictions are then made through combining predictions from the bivariate models within a Bayesian framework. Interestingly, pairwise modeling turns out to possess unique characteristics which allows to tackle the challenge of negative transfer through penalizing shared latent functions. The modeling framework is generic in terms of the choice of the kernel function and can scale to arbitrarily large datasets by parallelization. We also provide statistical guarantees for the proposed method, extent our method to separable molding cases and demonstrate its advantageous features through numerical studies. The numerical studies illustrate that we can (1) achieve similar prediction performance as the full multivariate approach when the output dimension is low, (2) outperform the full multivariate approach, with only a fraction of its computational needs, when the output dimension is high, (3) outperform the full multivariate approach when some functions are uncorrelated even when the output dimension is low.

One important extension of this model lies in the domain of functional graphical models. In such models, nodes are functions rather than random variables. In fact, and since a GP itself is an undirected graphical model, the MGP represents a fully connected undirected functional graphical model where each node represents an output. In our pairwise approach, we are encouraging independence between pairs of functions through our regularization framework. However, an interesting extension would be to extend this regularization framework, to build conditional independence amongst functions in a similar sense to Lemma 1. The main challenge however remains in providing a mapping between the covariance matrix and the precision matrix which control conditional independence between the outputs. We will work along this line and report the results in the future.

APPENDIX A

In this appendix we prove the following Lemmas from Section 3.4:

Lemma 1. (*Multivariate*) *Given that* $\text{pr}(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}_P, \boldsymbol{\Omega}^{-1})$, *then* $\boldsymbol{\Omega}_{ij} = \mathbf{0}$ *if and only if the multivariate Gaussian random vectors* $\boldsymbol{y}_i$ *and* $\boldsymbol{y}_j$ *are conditionally independent, i.e.* $cov(y_i^c, y_j^{c'}|\ddot{\boldsymbol{y}}) = 0$ *for every* $c \in \{1,..,p_i\}$ *and* $c' \in \{1,..,p_j\}$.

Lemma 2. (*Bivariate*) *Given that* $\text{pr}(\boldsymbol{y}_i, \boldsymbol{y}_j|\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{\theta}') = \mathcal{N}\left(\mathbf{0}_{p_i+p_j}, \begin{pmatrix}\boldsymbol{\Omega}_{ii} & \boldsymbol{\Omega}_{ij}\\ \boldsymbol{\Omega}_{ij}^\top & \boldsymbol{\Omega}_{jj}\end{pmatrix}^{-1}\right)$, *then* $\boldsymbol{\Omega}_{ij} = \mathbf{0}$ *if and only if, the multivariate Gaussian random vectors* $\boldsymbol{y}_i$ *and* $\boldsymbol{y}_j$ *are independent, i.e.* $cov(y_i^c, y_j^{c'}) = 0$ *for every* $c \in \{1,..,p_i\}$ *and* $c' \in \{1,..,p_j\}$.

Proof. We first prove Lemma 2 and deduce Lemma 1 accordingly. Let $\boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_j}$ denote the covariance between the random vectors $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$, where $\begin{pmatrix}\boldsymbol{\Omega}_{ii} & \boldsymbol{\Omega}_{ij}\\ \boldsymbol{\Omega}_{ij}^\top & \boldsymbol{\Omega}_{jj}\end{pmatrix} = \begin{pmatrix}\boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_i} & \boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_j}\\ \boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_j}^\top & \boldsymbol{C}_{\boldsymbol{y}_j,\boldsymbol{y}_j}\end{pmatrix}^{-1} = \text{cov}(\boldsymbol{y}_i,\boldsymbol{y}_j)^{-1}$. Using the inverse variance Lemma, we have that $\begin{pmatrix}\boldsymbol{\Omega}_{ii} & \boldsymbol{\Omega}_{ij}\\ \boldsymbol{\Omega}_{ij}^\top & \boldsymbol{\Omega}_{jj}\end{pmatrix} =$

$$\begin{pmatrix}\boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_i}^{-1} + \boldsymbol{H}_2^\top \text{cov}(\boldsymbol{y}_j|\boldsymbol{y}_i)^{-1}\boldsymbol{H}_2 & -\boldsymbol{H}_2^\top \text{cov}(\boldsymbol{y}_j|\boldsymbol{y}_i)^{-1}\\ -\text{cov}(\boldsymbol{y}_j|\boldsymbol{y}_i)^{-1}\boldsymbol{H}_2 & \text{cov}(\boldsymbol{y}_j|\boldsymbol{y}_i)^{-1}\end{pmatrix}, \tag{18}$$

where $\boldsymbol{H}_2 = \boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_j}^\top \boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_i}^{-1}$ and $\text{cov}(\boldsymbol{y}_j|\boldsymbol{y}_i) = \boldsymbol{C}_{\boldsymbol{y}_j,\boldsymbol{y}_j} - \boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_j}^\top \boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_i}^{-1}\boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_j}$. We now have that $\boldsymbol{\Omega}_{ij} = -\boldsymbol{H}_2^\top \text{cov}(\boldsymbol{y}_j|\boldsymbol{y}_i)^{-1} = -\boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_i}^{-1}\boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_j}\text{cov}(\boldsymbol{y}_j|\boldsymbol{y}_i)^{-1}$. However, since $\text{cov}(\boldsymbol{y}_i,\boldsymbol{y}_j)^{-1}$ is positive definite then both $\boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_i}$ and $\text{cov}(\boldsymbol{y}_j|\boldsymbol{y}_i)$ are positive definite and thus $\boldsymbol{\Omega}_{ij} = \mathbf{0}$ if and only if $\boldsymbol{C}_{\boldsymbol{y}_i,\boldsymbol{y}_j} = \mathbf{0}$.

To prove Lemma 1, we consider the random vectors $\boldsymbol{y}_i$, $\boldsymbol{y}_j$ and $\ddot{\boldsymbol{y}}$ where $\ddot{\boldsymbol{y}} = \{\boldsymbol{y}\}/\{y_i^c, y_j^{c'}\}$. Now replace $\boldsymbol{y}_i$ by $\ddot{\boldsymbol{y}}$ and replace $\boldsymbol{y}_j$ by the partitioned vector $[\boldsymbol{y}_i^\top, \boldsymbol{y}_j^\top]^\top$. Following (18), we observe that $\boldsymbol{\Omega}_{jj} = \text{cov}(\boldsymbol{y}_i,\boldsymbol{y}_j|\ddot{\boldsymbol{y}})^{-1}$. By applying the inverse variance Lemma again, but this time to $\text{cov}(\boldsymbol{y}_i,\boldsymbol{y}_j|\ddot{\boldsymbol{y}})^{-1}$ instead of $\text{cov}(\boldsymbol{y}_i,\boldsymbol{y}_j)^{-1}$ we have that $\boldsymbol{\Omega}_{jj} = \text{cov}(\boldsymbol{y}_i,\boldsymbol{y}_j|\ddot{\boldsymbol{y}})^{-1} =$

$$\begin{pmatrix}\boldsymbol{H}_3 & -\boldsymbol{H}_1^\top \text{cov}(\boldsymbol{y}_j|\boldsymbol{y}_i,\ddot{\boldsymbol{y}})^{-1}\\ -\text{cov}(\boldsymbol{y}_j|\boldsymbol{y}_i,\ddot{\boldsymbol{y}})^{-1}\boldsymbol{H}_1 & \text{cov}(\boldsymbol{y}_j|\boldsymbol{y}_i,\ddot{\boldsymbol{y}})^{-1}\end{pmatrix}, \tag{19}$$

where $\boldsymbol{H}_3 = \text{cov}(\boldsymbol{y}_i,\boldsymbol{y}_i|\ddot{\boldsymbol{y}})^{-1} + \boldsymbol{H}_1^\top \text{cov}(\boldsymbol{y}_j|\boldsymbol{y}_i,\ddot{\boldsymbol{y}})^{-1}\boldsymbol{H}_1$ and $\boldsymbol{H}_1 = \text{cov}(\boldsymbol{y}_i,\boldsymbol{y}_j|\ddot{\boldsymbol{y}})^\top \text{cov}(\boldsymbol{y}_i,\boldsymbol{y}_i|\ddot{\boldsymbol{y}})^{-1}$. Then following Lemma 2, the off-diagonal block is zero in this case, if and only if the multivariate Gaussian random vectors $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are conditionally independent where $\text{cov}(y_i^c, y_j^{c'}|\ddot{\boldsymbol{y}}) = 0$ for every $c \in \{1,..,p_i\}$ and $c' \in \{1,..,p_j\}$.

APPENDIX B

In this appendix we expand on $\partial \boldsymbol{C}_{ij}/\partial\theta_{ij}^{(n)}$ from Section 4.1. Recall that $\theta_{ij}^{(n)} \in \boldsymbol{\theta}_{ij}^\top = \{\boldsymbol{\theta}_{f_{ij}}^\top, \boldsymbol{\sigma}_{ij}^\top\}^\top$, $\boldsymbol{C}_{ij} = \boldsymbol{C}_{\boldsymbol{f}_{ij},\boldsymbol{f}_{ij}} + \boldsymbol{\Sigma}_{ij}$ and $\boldsymbol{\Lambda}_{qi}$ is a $D \times D$ positive definite diagonal matrix allowing different length scales for each dimension. For instance if $D = 2$ then $\boldsymbol{\Lambda}_{qi} = \begin{pmatrix}\nu_{qi(1)}^2 & 0\\ 0 & \nu_{qi(2)}^2\end{pmatrix}$. We first expand on $\partial \boldsymbol{C}_{ii}/\partial\theta_{ij}^{(n)}$, where, as shown in Section 4.1, the covariance of the marginal process is $\text{cov}_{ii}^f(\boldsymbol{x},\boldsymbol{x}') = \sum_{q=\{0,i\}} \xi_q^2\alpha_{qi}^2\text{exp}(-\frac{1}{4}\boldsymbol{d}^\top\boldsymbol{\Lambda}_{qi}\boldsymbol{d}) = \sum_{q=\{0,i\}} \xi_q^2\alpha_{qi}^2\text{exp}\big(-\frac{1}{4}\sum_{c=1}^D d_{(c)}^2\nu_{0i(c)}^2\big)$ for $c \in \{1,...,D\}$.

$$\frac{\partial \text{cov}_{ii}^f}{\partial \xi_0^2} = 2\xi_0\alpha_{0i}^2\text{exp}(-\frac{1}{4}\boldsymbol{d}^\top\boldsymbol{\Lambda}_{0i}\boldsymbol{d});$$
$$\frac{\partial \text{cov}_{ii}^f}{\partial \alpha_{0i}^2} = 2\xi_0^2\alpha_{0i}\text{exp}(-\frac{1}{4}\boldsymbol{d}^\top\boldsymbol{\Lambda}_{0i}\boldsymbol{d});$$
$$\frac{\partial \text{cov}_{ii}^f}{\partial \nu_{0i(c)}^2} = -\frac{1}{2}\xi_0^2\alpha_{0i}^2 d_{(c)}^2\nu_{0i(c)}^2\text{exp}(-\frac{1}{4}\boldsymbol{d}^\top\boldsymbol{\Lambda}_{0i}\boldsymbol{d});$$
$$\frac{\partial \text{cov}_{ii}^f}{\partial \sigma_i} = 2\sigma_i\tau_{ij}\tau_{x,x'}.$$

We exclude $\frac{\partial \text{cov}_{ii}^f}{\partial \xi_i^2}$, $\frac{\partial \text{cov}_{ii}^f}{\partial \alpha_{ii}^2}$ and $\frac{\partial \text{cov}_{ii}^f}{\partial \nu_{ii(c)}^2}$ due to similarity with their counterparts above. Now when $i \neq j$, we have that $\text{cov}_{ij}^f(\boldsymbol{x},\boldsymbol{x}') = \xi_0^2\tilde{\omega}_{ij}^0\text{exp}(-\frac{1}{2}\boldsymbol{d}^\top{\boldsymbol{\Phi}_{ij}^0}^{-1}\boldsymbol{d})$ where $\tilde{\omega}_{ij}^0 = 2^{\frac{D}{2}}\alpha_{0i}\alpha_{0j}|\boldsymbol{\Lambda}_{0i}|^{\frac{1}{4}}|\boldsymbol{\Lambda}_{0j}|^{\frac{1}{4}}/|\boldsymbol{\Lambda}_{0i}+\boldsymbol{\Lambda}_{0j}|^{\frac{1}{2}}$, and ${\boldsymbol{\Phi}_{ij}^q}^{-1} = (\boldsymbol{\Lambda}_{qi}^{-1} + \boldsymbol{\Lambda}_{qj}^{-1})^{-1}$. Let $c' = \{1,...,D\}/\{c\}$ then

$$\frac{\partial \text{cov}_{ij}^f}{\partial \xi_0^2} = 2\xi_0\tilde{\omega}_{ij}^0\text{exp}(-\frac{1}{2}\boldsymbol{d}^\top{\boldsymbol{\Phi}_{ij}^0}^{-1}\boldsymbol{d});$$
$$\frac{\partial \text{cov}_{ij}^f}{\partial \alpha_{0i}^2} = \xi_0^2 2^{\frac{D}{2}}\alpha_{0j}\frac{|\boldsymbol{\Lambda}_{0i}|^{\frac{1}{4}}|\boldsymbol{\Lambda}_{0j}|^{\frac{1}{4}}}{|\boldsymbol{\Lambda}_{0i}+\boldsymbol{\Lambda}_{0j}|^{\frac{1}{2}}}\text{exp}(-\frac{1}{2}\boldsymbol{d}^\top{\boldsymbol{\Phi}_{ij}^0}^{-1}\boldsymbol{d});$$
$$\frac{\partial \text{cov}_{ij}^f}{\partial \nu_{0i(c)}^2} = \bar{B}\xi_0^2 2^{\frac{D}{2}}\alpha_{0i}\alpha_{0j}\text{exp}(-\frac{1}{2}\boldsymbol{d}^\top{\boldsymbol{\Phi}_{ij}^0}^{-1}\boldsymbol{d}) +$$
$$\xi_0^2 2^{\frac{D}{2}}\alpha_{0i}\alpha_{0j}\frac{|\boldsymbol{\Lambda}_{0i}|^{\frac{1}{4}}|\boldsymbol{\Lambda}_{0j}|^{\frac{1}{4}}}{|\boldsymbol{\Lambda}_{0i}+\boldsymbol{\Lambda}_{0j}|^{\frac{1}{2}}}\bar{\bar{B}};$$
$$\bar{\bar{B}} = -d_{(c)}^2\frac{\nu_{0i(c)}\nu_{0j(c)}^4}{(\nu_{0i(c)}^2+\nu_{0j(c)}^2)^2};$$
$$\bar{B} = \left[\frac{\nu_{0i(c)}|\boldsymbol{\Lambda}_{0j}|^{\frac{1}{4}}\Big(\frac{1}{2}\prod_{c'}\nu_{0i(c')}^2|\boldsymbol{\Lambda}_{0i}|^{-\frac{3}{4}}|\boldsymbol{\Lambda}_{0i}+\boldsymbol{\Lambda}_{0j}|^{\frac{1}{2}} - |\boldsymbol{\Lambda}_{0i}|^{\frac{1}{4}}\nu_{0i(c)}|\boldsymbol{\Lambda}_{0i}+\boldsymbol{\Lambda}_{0j}|^{-\frac{1}{2}}\prod_{c'}(\nu_{0i(c')}^2+\nu_{0j(c')}^2)\Big)}{|\boldsymbol{\Lambda}_{0i}+\boldsymbol{\Lambda}_{0j}|}\right]$$

We also exclude $\frac{\partial \text{cov}_{ij}^f}{\partial \alpha_{0j}^2}$ and $\frac{\partial \text{cov}_{ij}^f}{\partial \nu_{0j(c)}^2}$ due to similarity with their counterparts above.

APPENDIX C

In this appendix we prove the following theorem from Section 4.2:

Theorem 1. *Suppose that* $\xi_0 = 0$, *then the predictive distribution of the bivariate model at any new input* $\boldsymbol{x}_0 \in \mathcal{X}$ *reduces to that of*

a univariate model where $\text{pr}(y_i(\boldsymbol{x}_0)|\boldsymbol{y}_{ij}) = \text{pr}(y_i(\boldsymbol{x}_0)|\boldsymbol{y}_i)$ *and* $\text{pr}(y_j(\boldsymbol{x}_0)|\boldsymbol{y}_{ij}) = \text{pr}(y_j(\boldsymbol{x}_0)|\boldsymbol{y}_j)$ *such that*

$$\begin{cases} \text{pr}(y_i(\boldsymbol{x}_0)|\boldsymbol{y}_{ij}) = \mathcal{N}\Big(\boldsymbol{C}_{\boldsymbol{f}_i,f_i^0}^\top \boldsymbol{\Omega}_{ii}\boldsymbol{y}_i,\, C_{f_i^0,f_i^0} + \\ \qquad\qquad \sigma_i^2 - \boldsymbol{C}_{\boldsymbol{f}_i,f_i^0}^\top \boldsymbol{\Omega}_{ii}\boldsymbol{C}_{\boldsymbol{f}_i,f_i^0}\Big) \\ \text{pr}(y_j(\boldsymbol{x}_0)|\boldsymbol{y}_{ij}) = \mathcal{N}\Big(\boldsymbol{C}_{\boldsymbol{f}_j,f_j^0}^\top \boldsymbol{\Omega}_{jj}\boldsymbol{y}_j,\, C_{f_j^0,f_j^0} + \\ \qquad\qquad \sigma_j^2 - \boldsymbol{C}_{\boldsymbol{f}_j,f_j^0}^\top \boldsymbol{\Omega}_{jj}\boldsymbol{C}_{\boldsymbol{f}_j,f_j^0}\Big) \end{cases}$$

where for $c \in \{i,j\}$, $\boldsymbol{\Omega}_{cc} = (\boldsymbol{C}_{\boldsymbol{f}_c,\boldsymbol{f}_c} + \sigma_c^2\boldsymbol{I}_{p_c})^{-1} \in \mathcal{R}^{p_c\times p_c}$, $\boldsymbol{C}_{\boldsymbol{f}_c,f_c^0} = [cov_{cc}^f(\boldsymbol{x}_0,\boldsymbol{x}_{c1}),...,cov_{cc}^f(\boldsymbol{x}_0,\boldsymbol{x}_{cp_c})]^\top$, $C_{f_c^0,f_c^0} = cov_{cc}^f(\boldsymbol{x}_0,\boldsymbol{x}_0)$ *and* $cov_{cc}^f(\boldsymbol{x},\boldsymbol{x}') = \xi_c^2\int_{-\infty}^{\infty} K_{cc}(\boldsymbol{u})K_{cc}(\boldsymbol{u}-\boldsymbol{d})d\boldsymbol{u}$.

Proof. Based on (9), $\xi_0 = 0$ implies that, for $i \neq j$, $\text{cov}_{ij}^f(\boldsymbol{x},\boldsymbol{x}') = \xi_0^2\int_{-\infty}^{\infty} K_{0i}(\boldsymbol{u})K_{0j}(\boldsymbol{u}-\boldsymbol{d})d\boldsymbol{u} = 0$ for every $\boldsymbol{d}$. Therefore, we have that $\boldsymbol{C}_{ij} = \boldsymbol{C}_{\boldsymbol{f}_{ij},\boldsymbol{f}_{ij}} + \boldsymbol{\Sigma}_{ij} = \begin{pmatrix} \boldsymbol{C}_{\boldsymbol{f}_i,\boldsymbol{f}_i} & \boldsymbol{0}_{p_i\times p_j} \\ \boldsymbol{0}_{p_i\times p_j} & \boldsymbol{C}_{\boldsymbol{f}_j,\boldsymbol{f}_j}\end{pmatrix} + \begin{pmatrix} \sigma_i^2\boldsymbol{I}_{p_i} & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_j^2\boldsymbol{I}_{p_j}\end{pmatrix}$. Applying (3) under a bivariate setting we we have that

$$\text{pr}(y_i(\boldsymbol{x}_0)|\boldsymbol{y}_{ij}) = \mathcal{N}\Big(\boldsymbol{C}_{\boldsymbol{f}_{ij},f_i^0}^\top \boldsymbol{C}_{ij}^{-1}\boldsymbol{y}_{ij},\; C_{f_i^0,f_i^0} + \sigma_i^2 - \boldsymbol{C}_{\boldsymbol{f}_{ij},f_i^0}^\top \boldsymbol{C}_{ij}^{-1}\boldsymbol{C}_{\boldsymbol{f}_{ij},f_i^0}\Big). \tag{20}$$

Recall, $\boldsymbol{C}_{\boldsymbol{f}_{ij},f_i^0} = [\boldsymbol{C}_{\boldsymbol{f}_i,f_i^0}^\top, \boldsymbol{C}_{\boldsymbol{f}_j,f_i^0}^\top]^\top$ where $\boldsymbol{C}_{\boldsymbol{f}_i,f_i^0}^\top = [\text{cov}_{ii}^f(\boldsymbol{x}_0,\boldsymbol{x}_{i1}),...,\text{cov}_{ii}^f(\boldsymbol{x}_0,\boldsymbol{x}_{ip_i})]^\top$ and $\boldsymbol{C}_{\boldsymbol{f}_j,f_i^0} = [\text{cov}_{ij}^f(\boldsymbol{x}_0,\boldsymbol{x}_{j1}),...,\text{cov}_{ij}^f(\boldsymbol{x}_0,\boldsymbol{x}_{jp_j})]^\top$. However, since $\text{cov}_{ij}^f(\boldsymbol{x},\boldsymbol{x}') = 0$ for $i \neq j$ then $\boldsymbol{C}_{\boldsymbol{f}_j,f_i^0} = \boldsymbol{0}_{p_j}^\top$, therefore

$$\begin{aligned}\boldsymbol{C}_{\boldsymbol{f}_{ij},f_i^0}^\top \boldsymbol{C}_{ij}^{-1}\boldsymbol{y}_{ij} &= [\boldsymbol{C}_{\boldsymbol{f}_i,f_i^0}^\top, \boldsymbol{0}_{p_j}]\times \\ &\begin{pmatrix}(\boldsymbol{C}_{\boldsymbol{f}_i,\boldsymbol{f}_i} + \sigma_i^2\boldsymbol{I}_{p_i})^{-1} & \boldsymbol{0}_{p_i\times p_j} \\ \boldsymbol{0}_{p_i\times p_j} & (\boldsymbol{C}_{\boldsymbol{f}_j,\boldsymbol{f}_j} + \sigma_j^2\boldsymbol{I}_{p_j})^{-1}\end{pmatrix}[\boldsymbol{y}_i^\top, \boldsymbol{y}_j^\top]^\top \\ &= \boldsymbol{C}_{\boldsymbol{f}_i,f_i^0}^\top \boldsymbol{\Omega}_{ii}\boldsymbol{y}_i\end{aligned}$$

$$\begin{aligned}&\boldsymbol{C}_{\boldsymbol{f}_{ij},f_i^0}^\top \boldsymbol{C}_{ij}^{-1}\boldsymbol{C}_{\boldsymbol{f}_{ij},f_i^0} = [\boldsymbol{C}_{\boldsymbol{f}_i,f_i^0}^\top, \boldsymbol{0}_{p_j}]\times \\ &\begin{pmatrix}(\boldsymbol{C}_{\boldsymbol{f}_i,\boldsymbol{f}_i} + \sigma_i^2\boldsymbol{I}_{p_i})^{-1} & \boldsymbol{0}_{p_i\times p_j} \\ \boldsymbol{0}_{p_i\times p_j} & (\boldsymbol{C}_{\boldsymbol{f}_j,\boldsymbol{f}_j} + \sigma_j^2\boldsymbol{I}_{p_j})^{-1}\end{pmatrix}[\boldsymbol{C}_{\boldsymbol{f}_i,f_i^0}^\top, \boldsymbol{0}_{p_j}]^\top \\ &= \boldsymbol{C}_{\boldsymbol{f}_i,f_i^0}^\top \boldsymbol{\Omega}_{ii}\boldsymbol{C}_{\boldsymbol{f}_i,f_i^0}\end{aligned}$$

Similarly, we can obtain the proof $\text{pr}(y_j(\boldsymbol{x}_0)|\boldsymbol{y}_{ij})$.

APPENDIX D

In this appendix we prove the following theorems from Section 4.2:

Theorem 2. *If* $z_2 = max\{|\mathbb{P}_\lambda''(|\xi_0^*|)| : \xi_0^* \neq 0\} \to 0$, *then there exits a local minimzer* $\hat{\boldsymbol{\theta}}_{ij}$ *for* $\ell_{\mathbb{P}}(\boldsymbol{\theta}_{ij})$, *such that* $||\hat{\boldsymbol{\theta}}_{ij} - \boldsymbol{\theta}_{ij}^*|| = O(r_{\text{p}}^{-1} + z_1)$, *where* $z_1 = max\{\mathbb{P}_\lambda'(|\xi_0^*|) : \xi_0^* \neq 0\}$.

Theorem 3. *Assume that* $\xi_0^* = 0$ *and the parameters* $\ddot{\boldsymbol{\theta}}_{ij} = \{\boldsymbol{\theta}_{ij}\}/\{\xi_0\}$ *satisfy* r_{p} *consistency in theorem 2. Then if* $\liminf_{\text{p}\to\infty}\liminf_{\xi_0\to 0^+}\frac{1}{\lambda}\mathbb{P}_\lambda'(\xi_0) > 0$, $\lambda \to 0$ *and* $\text{p}\lambda/r_{\text{p}} \to \infty$ *as* $\text{p}\to\infty$, *we have that* $\lim_{\text{p}\to\infty}\text{pr}(\hat{\xi}_0 = 0) = 1$.

Proof. First we note that for the the negative log-likelihood (ℓ') and penalty function ($\mathbb{P}_\lambda'(|\xi_0|)$) the "$\prime$" notation on a function implies a derivative. As previously mentioned the MLE for the unpenalized likelihood $\ell(\boldsymbol{\theta}_{ij})$ is r_{p} consistent where r_{p} is a sequence such that $r_{\text{p}} \to \infty$ as $\text{p} \to \infty$. Therefore, we have that $r_{\text{p}}^{-1}\ell'(\boldsymbol{\theta}_{ij}) = O(1)$ and $||\hat{\boldsymbol{\theta}}_{ij} - \boldsymbol{\theta}_{ij}^*|| = O(r_{\text{p}}^{-1})$ [48], [49]. This result is a direct extension of the well known root-p consistency of the MLE based on independent and identically distributed normal observations, which holds under the usual regularity conditions (please refer to chapter 7 of [48]). In theorems 2 and 3 we aim to study the asymptotic properties of the penalized likelihood $\ell_{\mathbb{P}}(\boldsymbol{\theta}_{ij}) = \ell(\boldsymbol{\theta}_{ij}) + \mathbb{P}_\lambda(|\xi_0|)$. The proofs provide similar results as in [51], but defined within our model specifications and based on dependent observations. To be consistent with [51] notation, instead of minimizing the negative log-likelihood we maximize the log-likelihood whose form follows $\ell_{\mathbb{P}+}(\boldsymbol{\theta}_{ij}) = -\ell_{\mathbb{P}}(\boldsymbol{\theta}_{ij}) = -\ell(\boldsymbol{\theta}_{ij}) - \mathbb{P}_\lambda(|\xi_0|) = \ell_+(\boldsymbol{\theta}_{ij}) - \mathbb{P}_\lambda(|\xi_0|)$. Also we follow their convention by multiplying by p the penalty function, i.e. $\ell_{\mathbb{P}+}(\boldsymbol{\theta}_{ij}) = \ell_+(\boldsymbol{\theta}_{ij}) - \text{p}\mathbb{P}_\lambda(|\xi_0|)$. To prove theorem 2, we need to show that for any given $\varepsilon > 0$ there exists a large constant $\mathcal{G}$ such that

$$\text{pr}\Big(\sup_{||\boldsymbol{g}||=\mathcal{G}} \ell_{\mathbb{P}+}(\boldsymbol{\theta}_{ij}^* + \rho\boldsymbol{g}) < \ell_{\mathbb{P}+}(\boldsymbol{\theta}_{ij}^*)\Big) \geq 1-\varepsilon\,, \tag{21}$$

where $\rho = r_{\text{p}}^{-1} + z_1$. This equation implies that, with a probability at least $1-\varepsilon$, there exists a local maximum in the ball $\{\boldsymbol{\theta}_{ij}^* + \rho\boldsymbol{g} : ||\boldsymbol{g}|| \leq \mathcal{G}\}$, where the local maximizer $\hat{\boldsymbol{\theta}}_{ij}$ satisfies $||\hat{\boldsymbol{\theta}}_{ij} - \boldsymbol{\theta}_{ij}^*|| = O(\rho) = O(r_{\text{p}}^{-1} + z_1)$. Expanding on $\ell_{\mathbb{P}}(\boldsymbol{\theta}_{ij}^* + \rho\boldsymbol{g}) - \ell_{\mathbb{P}}(\boldsymbol{\theta}_{ij}^*)$, we have that

$$\begin{aligned}\ell_{\mathbb{P}+}(\boldsymbol{\theta}_{ij}^* + \rho\boldsymbol{g}) - \ell_{\mathbb{P}+}(\boldsymbol{\theta}_{ij}^*) &= \big[\ell_+(\boldsymbol{\theta}_{ij}^* + \rho\boldsymbol{g}) - \ell_+(\boldsymbol{\theta}_{ij}^*)\big] \\ &\quad - \text{p}\big[\mathbb{P}_\lambda(|\xi_0^* + \rho g_{\xi_0}|) - \mathbb{P}_\lambda(|\xi_0^*|)\big],\end{aligned} \tag{22}$$

where g_{ξ_0} denotes the element in $\boldsymbol{g}$ corresponding to ξ_0. Under the assumption that $\mathbb{P}_\lambda(|\xi_0|) \geq 0$ and $\mathbb{P}_\lambda(0) = 0$, and if $\xi_0^* = 0$ then $\ell_{\mathbb{P}+}(\boldsymbol{\theta}_{ij}^* + \rho\boldsymbol{g}) - \ell_{\mathbb{P}+}(\boldsymbol{\theta}_{ij}^*) \leq \ell_+(\boldsymbol{\theta}_{ij}^* + \rho\boldsymbol{g}) - \ell_+(\boldsymbol{\theta}_{ij}^*)$ as $\mathbb{P}_\lambda(|\xi_0^* + g_{\xi_0}|) - \mathbb{P}_\lambda(|\xi_0^*|) = \mathbb{P}_\lambda(|\rho g_{\xi_0}|) \geq 0$. Using a Taylor expansion we have that $\ell_+(\boldsymbol{\theta}_{ij}^* + \rho\boldsymbol{g}) = \ell_+(\boldsymbol{\theta}_{ij}^*) + \rho\ell_+'(\boldsymbol{\theta}_{ij}^*)^\top\boldsymbol{g} - \frac{\rho^2}{2}\boldsymbol{g}^\top\mathbb{I}(\boldsymbol{\theta}_{ij}^*)\boldsymbol{g}\{1+o(1)\}$ where $\ell_+'(\boldsymbol{\theta}_{ij}^*)^\top$ is the gradient vector of ℓ_+ evaluated at $\boldsymbol{\theta}_{ij}^*$ and $\mathbb{I}$ is a finite positive definite information matrix at $\boldsymbol{\theta}_{ij}^*$. Therefore, through applying a Taylor expansion also on $\mathbb{P}_\lambda(|\xi_0^* + \rho g_{\xi_0}|)$, we have that

$$\begin{aligned}&\ell_{\mathbb{P}+}(\boldsymbol{\theta}_{ij}^* + \rho\boldsymbol{g}) - \ell_{\mathbb{P}+}(\boldsymbol{\theta}_{ij}^*) \\ &\leq \big[\rho\ell_+'(\boldsymbol{\theta}_{ij}^*)^\top\boldsymbol{g} - \frac{\rho^2}{2}\boldsymbol{g}^\top\mathbb{I}(\boldsymbol{\theta}_{ij}^*)\boldsymbol{g}\{1+o(1)\}\big] \\ &- \text{p}\big[\rho\mathbb{P}_\lambda'(|\xi_0^*|)\text{sign}(\xi_0^*)g_{\xi_0} + \rho^2\mathbb{P}_\lambda''(|\xi_0^*|)g_{\xi_0}^2\{1+o(1)\}\big].\end{aligned} \tag{23}$$

Note that $r_{\text{p}}^{-1}\ell'(\boldsymbol{\theta}_{ij}) = O(1)$, and the penalty form is similar to that of [51], thus the rest of the proof is identical to theorem 1 in [51]. Regarding theorem 3, we need to show that $\lim_{\text{p}\to\infty}\text{pr}(\hat{\xi}_0 = 0) = 1$ if $\xi_0^* = 0$. First we have that

$$\frac{\partial\ell_{\mathbb{P}+}(\boldsymbol{\theta}_{ij})}{\partial\xi_0} = \frac{\partial\ell_+(\boldsymbol{\theta}_{ij})}{\partial\xi_0} - \text{p}\mathbb{P}_\lambda'(|\xi_0|)\text{sign}(\xi_0). \tag{24}$$

Following a first order Taylor expansion, (24) can be written as

$$\frac{\partial \ell_{\mathbb{P}+}(\hat{\boldsymbol{\theta}}_{ij})}{\partial \xi_0} = \frac{\partial \ell_{+}(\boldsymbol{\theta}^*_{ij})}{\partial \xi_0} + \sum_n \frac{\partial^2 \ell_{+}(\boldsymbol{\theta}^*_{ij})}{\partial \xi_0 \partial \theta^{(n)}_{ij}}(\hat{\boldsymbol{\theta}}^{(n)}_{ij} - \boldsymbol{\theta}^{*(n)}_{ij}) + \sum_n \sum_{n'} \frac{\partial^3 \ell_{+}(\boldsymbol{\theta}^{**}_{ij})}{\partial \xi_0 \partial \theta^{(n)}_{ij} \theta^{(n')}_{ij}}(\hat{\boldsymbol{\theta}}^{(n)}_{ij} - \boldsymbol{\theta}^{*(n)}_{ij})(\hat{\boldsymbol{\theta}}^{(n')}_{ij} - \boldsymbol{\theta}^{*(n')}_{ij}) - \mathrm{p}\mathbb{P}'_{\lambda}(|\hat{\xi}_0|)\mathrm{sign}(\hat{\xi}_0), \tag{25}$$

for some $\boldsymbol{\theta}^{**}_{ij} \in (\boldsymbol{\theta}^*_{ij}, \hat{\boldsymbol{\theta}}_{ij})$. However, since $\ell_+(\hat{\boldsymbol{\theta}}_{ij})$ is of order $O(r_{\mathrm{p}})$ and given that $||\hat{\boldsymbol{\theta}}_{ij} - \boldsymbol{\theta}^*_{ij}|| = O(r_{\mathrm{p}}^{-1})$ we have $\frac{\partial \ell_{\mathbb{P}+}(\hat{\boldsymbol{\theta}}_{ij})}{\partial \xi_0} = \lambda \mathrm{p}\big[-\frac{1}{\lambda}\mathbb{P}'_{\lambda}(|\hat{\xi}_0|)\mathrm{sign}(\hat{\xi}_0) + O\Big(\frac{r_{\mathrm{p}}}{\lambda \mathrm{p}}\Big)\big]$. From here and since, $\liminf_{\mathrm{p}\to\infty} \liminf_{\xi_0 \to 0^+} \frac{1}{\lambda}\mathbb{P}'_{\lambda}(\xi_0) > 0$ and $\mathrm{p}\lambda/r_{\mathrm{p}} \to \infty$, then the sign of $\frac{\partial \ell_{\mathbb{P}+}(\hat{\boldsymbol{\theta}}_{ij})}{\partial \xi_0}$ is only determined by the sign of $\hat{\xi}_0$. Thus, $\frac{\partial \ell_{\mathbb{P}+}(\hat{\boldsymbol{\theta}}_{ij})}{\partial \xi_0} < 0$ for $0 < \hat{\xi}_0 < \varepsilon$ and $\frac{\partial \ell_{\mathbb{P}+}(\hat{\boldsymbol{\theta}}_{ij})}{\partial \xi_0} > 0$ for $-\varepsilon < \hat{\xi}_0 < 0$ for some small ε, which is a sufficient condition to prove theorem 3.

ACKNOWLEDGMENTS

The authors would like to acknowledge support for this project from the National Science Foundation (NSF grant #1561512 and NSF grant #1811767).

REFERENCES

[1] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 1939–1959, 2005.

[2] M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings, "Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes," in *Proceedings of the 7th international conference on Information processing in sensor networks*. IEEE Computer Society, 2008, pp. 109–120.

[3] R. Kontar, S. Zhou, C. Sankavaram, X. Du, and Y. Zhang, "Nonparametric-condition-based remaining useful life prediction incorporating external factors," *IEEE Transactions on Reliability*, 2017.

[4] R. Caruana, "Multitask learning," in *Learning to learn*. Springer, 1998, pp. 95–133.

[5] X.-T. Yuan, X. Liu, and S. Yan, "Visual classification with multitask joint sparse representation," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4349–4360, 2012.

[6] S. Conti, J. P. Gosling, J. E. Oakley, and A. O'Hagan, "Gaussian process emulation of dynamic computer codes," *Biometrika*, vol. 96, no. 3, pp. 663–676, 2009.

[7] S. Conti and A. O'Hagan, "Bayesian emulation of complex multi-output and dynamic computer models," *Journal of statistical planning and inference*, vol. 140, no. 3, pp. 640–651, 2010.

[8] P. Z. G. Qian, H. Wu, and C. J. Wu, "Gaussian process models for computer experiments with qualitative and quantitative factors," *Technometrics*, vol. 50, no. 3, pp. 383–396, 2008.

[9] Q. Zhou, P. Z. Qian, and S. Zhou, "A simple approach to emulation for computer models with qualitative and quantitative factors," *Technometrics*, vol. 53, no. 3, pp. 266–273, 2011.

[10] A. Majumdar and A. E. Gelfand, "Multivariate spatial modeling for geostatistical data using convolved covariance functions," *Mathematical Geology*, vol. 39, no. 2, pp. 225–245, 2007.

[11] T. E. Fricker, J. E. Oakley, and N. M. Urban, "Multivariate gaussian process emulators with nonseparable covariance structures," *Technometrics*, vol. 55, no. 1, pp. 47–56, 2013.

[12] A. Melkumyan and F. Ramos, "Multi-kernel gaussian processes," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 1, 2011, p. 1408.

[13] M. Goulard and M. Voltz, "Linear coregionalization model: tools for estimation and choice of cross-variogram matrix," *Mathematical Geology*, vol. 24, no. 3, pp. 269–286, 1992.

[14] M. Álvarez, D. Luengo, M. Titsias, and N. Lawrence, "Efficient multioutput gaussian processes through variational inducing kernels," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 25–32.

[15] J. M. Ver Hoef and R. P. Barry, "Constructing and fitting models for cokriging and multivariable spatial prediction," *Journal of Statistical Planning and Inference*, vol. 69, no. 2, pp. 275–294, 1998.

[16] P. Boyle and M. Frean, "Dependent gaussian processes," in *Advances in neural information processing systems*, 2005, pp. 217–224.

[17] M. Alvarez and N. D. Lawrence, "Sparse convolved gaussian processes for multi-output regression," in *Advances in neural information processing systems*, 2009, pp. 57–64.

[18] M. A. Álvarez and N. D. Lawrence, "Computationally efficient convolved multiple output gaussian processes," *Journal of Machine Learning Research*, vol. 12, no. May, pp. 1459–1500, 2011.

[19] D. Higdon, J. Gattiker, B. Williams, and M. Rightley, "Computer model calibration using high-dimensional output," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 570–583, 2008.

[20] J. McFarland, S. Mahadevan, V. Romero, and L. Swiler, "Calibration and uncertainty analysis for computer simulations with multivariate output," *AIAA journal*, vol. 46, no. 5, pp. 1253–1265, 2008.

[21] G. Han, T. J. Santner, W. I. Notz, and D. L. Bartel, "Prediction for computer experiments having quantitative and qualitative input variables," *Technometrics*, vol. 51, no. 3, pp. 278–288, 2009.

[22] R. Kontar, S. Zhou, C. Sankavaram, X. Du, and Y. Zhang, "Nonparametric modeling and prognosis of condition monitoring signals using multivariate gaussian convolution processes," *Technometrics*, vol. 60, no. 4, pp. 484–496, 2018.

[23] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[24] A. Stein and L. Corsten, "Universal kriging and cokriging as a regression procedure," *Biometrics*, pp. 575–587, 1991.

[25] C. A. Calder and N. Cressie, "Some topics in convolution-based spatial modeling," *Proceedings of the 56th Session of the International Statistics Institute*, pp. 22–29, 2007.

[26] P. Boyle, "Gaussian processes for regression and optimisation," 2007.

[27] H. J. Thiébaux and M. Pedder, *Spatial objetive analysis: with applications in atmospheric science*, 1987, no. 519.24 THI.

[28] R. P. Barry, M. Jay, and V. Hoef, "Blackbox kriging: spatial prediction without specifying variogram models," *Journal of Agricultural, Biological, and Environmental Statistics*, pp. 297–322, 1996.

[29] D. Higdon, "Space and space-time modeling using process convolutions," in *Quantitative methods for current environmental issues*. Springer, 2002, pp. 37–56.

[30] S. Haykin, *Communication systems*. John Wiley & Sons, 2008.

[31] C. K. Wikle, "A kernel-based spectral model for non-gaussian spatio-temporal processes," *Statistical Modelling*, vol. 2, no. 4, pp. 299–314, 2002.

[32] M. A. Alvarez, L. Rosasco, N. D. Lawrence *et al.*, "Kernels for vector-valued functions: A review," *Foundations and Trends® in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.

[33] T. V. Nguyen, E. V. Bonilla *et al.*, "Collaborative multi-output gaussian processes." in *UAI*, 2014, pp. 643–652.

[34] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.

[35] S. D. Tajbakhsh, N. S. Aybat, and E. Del Castillo, "Sparse precision matrix selection for fitting gaussian random field models to large data sets," *arXiv preprint arXiv:1405.5576*, 2014.

[36] K. Mardia and A. Watkins, "On multimodality of the likelihood in the spatial linear model," *Biometrika*, vol. 76, no. 2, pp. 289–295, 1989.

[37] T. H. Wonnacott and R. J. Wonnacott, *Introductory statistics*. Wiley New York, 1990, vol. 5.

[38] B. M. Colosimo, P. Cicorella, M. Pacella, and M. Blaco, "From profile to surface monitoring: Spc for cylindrical surfaces via gaussian processes," *Journal of Quality Technology*, vol. 46, no. 2, pp. 95–113, 2014.

[39] Y. Li and Q. Zhou, "Pairwise meta-modeling of multivariate output computer models using nonseparable covariance function," *Technometrics*, vol. 58, no. 4, pp. 483–494, 2016.

[40] Y. Li, Q. Zhou, X. Huang, and L. Zeng, "Pairwise estimation of multivariate gaussian process models with replicated observations: Application to multivariate profile monitoring," *Technometrics*, vol. 60, no. 1, pp. 70–78, 2018.
[41] V. Tresp, "A bayesian committee machine," *Neural computation*, vol. 12, no. 11, pp. 2719–2741, 2000.
[42] M. P. Deisenroth and J. W. Ng, "Distributed gaussian processes," *arXiv preprint arXiv:1502.02843*, 2015.
[43] S. Fieuws and G. Verbeke, "Pairwise fitting of mixed models for the joint modeling of multivariate longitudinal profiles," *Biometrics*, vol. 62, no. 2, pp. 424–431, 2006.
[44] C. Williams, S. Klanke, S. Vijayakumar, and K. M. Chai, "Multi-task gaussian process learning of robot inverse dynamics," in *Advances in Neural Information Processing Systems*, 2009, pp. 265–272.
[45] J. Whittaker, *Graphical models in applied multivariate statistics*. Wiley Publishing, 2009.
[46] C. J. Paciorek and M. J. Schervish, "Nonstationary covariance functions for gaussian process regression," in *Advances in neural information processing systems*, 2004, pp. 273–280.
[47] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1, no. 10.
[48] I. V. Basawa, *Statistical Inferences for Stochasic Processes: Theory and Methods*. Elsevier, 1980.
[49] I. Basawa, P. Feigin, and C. Heyde, "Asymptotic properties of maximum likelihood estimators for stochastic processes," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 259–270, 1976.
[50] J. Q. Shi and T. Choi, *Gaussian process regression analysis for functional data*. CRC Press, 2011.
[51] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.
[52] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the american statistical association*, vol. 90, no. 432, pp. 1200–1224, 1995.
[53] J. D. Helterbrand and N. Cressie, "Universal cokriging under intrinsic coregionalization," *Mathematical Geology*, vol. 26, no. 2, pp. 205–226, 1994.
[54] J. W. Ng and M. P. Deisenroth, "Hierarchical mixture-of-experts model for large-scale gaussian process regression," *arXiv preprint arXiv:1412.3078*, 2014.
[55] T. Heskes, "Selecting weighting factors in logarithmic opinion pools," in *Advances in neural information processing systems*, 1998, pp. 266–272.
[56] Y. Cao and D. J. Fleet, "Generalized product of experts for automatic and principled fusion of gaussian process predictions," *arXiv preprint arXiv:1410.7827*, 2014.
[57] A. Schwaighofer and V. Tresp, "Transductive and inductive methods for approximate gaussian process regression," in *Advances in Neural Information Processing Systems*, 2003, pp. 977–984.
[58] J. O. Ramsay, *Functional data analysis*. Wiley Online Library, 2006.
[59] L. Kleinrock, *Queueing systems, volume 2: Computer applications*. wiley New York, 1976, vol. 66.
[60] G. Parra and F. Tobar, "Spectral mixture kernels for multi-output gaussian processes," in *Advances in Neural Information Processing Systems*, 2017, pp. 6681–6690.

Raed Kontar is an Assistant Professor in the department of Industrial & Operations at University Michigan since 2018. He completed a PhD in Industrial and Systems Engineering at the University of Wisconsin-Madison in 2018, where he also received his masters in statistics in 2017. His research area include transfer/multitask learning, uncertainty quantification and kernel methods with application in Internet of Things (IoT) enabled products/systems.

Garvesh Raskutti is an Associate Professor in Statistics at University of Wisconsin-Madison. He completed a PhD in Statistics from UC Berkeley and was a Postdoctoral Fellow at the Statistical and Mathematical Sciences Institute (SAMSI). His research areas include high-dimensional statistics, statistical learning theory and algorithms, graphical models, and time series models.

Shiyu Zhou is the Vilas Distinguished Achievement Professor in the Department of Industrial and Systems Engineering at the University of Wisconsin-Madison. His research focuses on data-driven modeling, monitoring, diagnosis, and prognosis for engineering systems with particular emphasis on manufacturing and after-sales service systems. He has established methods for modeling, analysis, and control of Internet-of-Things (IoT) enabled smart and connected systems, variation modeling, analysis, and reduction for complex manufacturing processes, and process control methodologies for emerging nano-manufacturing processes. He is a recipient of a CAREER Award from the National Science Foundation and the Best Application Paper Award from IIE Transactions. He is now the director of IoT Systems Research Center at UW-Madison and a fellow of IISE, ASME, and SME.