Towards Trustworthy Machine Learning with Kernels

Siu Lun Chau

St.Peter's College

University of Oxford

A thesis presented for the degree of

Doctor of Philosophy

Trinity 2023

To myself, friends, and family.

Acknowledgements

First and foremost, I would like to express my deepest appreciation to my supervisor Professor Dino Sejdinovic for his continuous guidance and encouragements throughout my DPhil, without which I would not have been able to complete this journey. His wisdom and attitude towards both research and life have inspired me to become a better academic and a better individual.

I have also had the privilege of working with several incredible researchers and mentors over the past four years. They have all impacted my thinking and approach to research, and I am eternally grateful for their contributions. I extend my sincere appreciation to Dr. Thibaut Lienart for introducing me to Machine Learning in 2016 and supporting my research and professional development since then. Without his contribution, my life would be significantly different. Many thanks to Professor Mihai Cucuringu for his patience and support, especially during the early stages of my DPhil. I am also grateful to Professor Xiaowen Dong, who has always been caring and kind to me. Thanks are also due to Dr. Javier Gonzalez, who had confidence in me during the most challenging period of my DPhil. Lastly, I cannot express my gratitude enough to Dr. Krikamol Muandet. Our discussion on what makes a piece of research meaningful during my stay at the MPI has changed my way of approaching machine learning research. I look forward to continuing working with you all in the future.

Aside from my mentors, I would also like to thank the EPSRC and the MRC for their generous support of my research. Also, thank you to the Department of Statistics, where I have stayed since I was an undergraduate, and where I always found refuge when I felt lost and doubtful in the past eight years. Thank you to all the staff and researchers who have made this possible.

This endeavour would not have been possible without the support of my lovely colleagues as well. Huge thanks to Bobby He, Deborah Sulem, Edwin Fong, Fan Wu, Henry Kenlay, Jake Fawkes, Leon Law, Lorenzo Pacchiadrdi, Lucian Chan, Michael Zhu Li, Stacy Pu, Veit Wild, William Thomas, and Yin Cong Zhi. Special thanks to Shahine Bouabid, Jean-Francois Ton, and Robert Hu, whom I am lucky enough to have collaborated with and learnt so much from. Their moral support is what kept me sane during the toughest times.

I would be in remiss in not mentioning my friends and family, who have supported my journey from the beginning. To Alan Chan, Angel Wong, Bryan Ng, Charig Yang, Cherie Wong, Churchill Ngai, Dominic Li, Gloria Ma, Kenneth Lee, Nicole Hui, Nicholas Yung, Terence Tsui, and Tooki Chu, thank you all for making my Oxford life less nerdy and more fun. To my partner Yidan Gao, a big thank you for bringing joy and new perspectives to my life always. I wish you will have as much fun in your DPhil as I had in mine. To my Grandma: Thank you for raising and feeding me very well all these years. To my sister:

Thank you for being there whenever I need you. You have always been my biggest fan and soul mate, and I couldn't be where I am today without your sacrifices and love. To mum: Thank you for your faith in whatever decision I make to pursue my dream. Your wise words are today still my motto today. To dad: Thank you for your tough love, generous support, and understanding all these years. Hope you all are proud of me and will keep watching over me in the coming years.

Declaration

I hereby declare that except where specific reference is made to the work of others, the intellectual contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification. The particulars of my personal contributions can be found in Section 1.2 of the introduction. This dissertation is my own work except as specified in the text.

Siu Lun Chau

Trinity 2023

Abstract

Machine Learning has become an indispensable aspect of various safety-critical industries like healthcare, law, and automotive. Hence, it is crucial to ensure that our machine learning models function appropriately and instil trust among their users. This thesis focuses on improving the safety and transparency of Machine Learning by advocating for more principled uncertainty quantification and more effective explainability tools. Specifically, the use of Kernel Mean Embeddings (KME) and Gaussian Processes (GP) is prevalent in this work since they can represent probability distribution with minimal distributional assumptions and capture uncertainty well, respectively. I dedicate Chapter 2 to introduce these two methodologies. Chapter 3 demonstrates an effective use of these methods in conjunction with each other to tackle a statistical downscaling problem, in which a Deconditional Gaussian process is proposed. Chapter 4 considers a causal data fusion problem, where multiple causal graphs are combined for inference. I introduce BayesIMP, an algorithm built using KME and GPs, to draw causal conclusion while accounting for the uncertainty in the data and model. In Chapter 5, I present RKHS-SHAP to model explainability for kernel methods that utilizes Shapley values. Specifically, I propose to estimate the value function in the cooperative game using KMEs, circumventing the need for any parametric density estimations. A Shapley regulariser is also proposed to regulate the amount of contributions certain features can have to the model. Chapter 6 presents a generalised preferential Gaussian processes for modelling preference with non-rankable structure, which sets the scene for Chapter 7, where I built upon my research and propose Pref-SHAP to explain preference models.

Table of Contents

List of Figures

List of Tables

1 | Introduction

This thesis is presented in an integrated format, consisting of 8 chapters that work together to provide a cohesive narrative. The first and last chapters serve to offer an overview of the thesis and a discussion of its implications, respectively. In Chapter 2, relevant background materials are presented for context.

The subsequent 5 chapters are based on the papers I have published over the course of my DPhil studies. Each of them is self-contained and includes a comprehensive introduction, literature review, and their separate contribution to contextualize the research within its specific topic. Appendices are compiled and included at the end of the thesis.

In the remainder of this chapter, I outline the motivation and structure of the thesis in detail. Additionally, I will provide a list of projects I have undertaken during my DPhil, including those not included in this thesis.

1.1 Motivation

Machine learning (ML) has become an indispensable aspect of our day-to-day activities owing to its ability to improve the efficiency, precision, and quality of conventional practices across different domains. ML models have vast applications, ranging from diagnosing illnesses [Alaa et al., 2017] and prescribing personalized treatment strategies [Bica and van der Schaar, 2022], detecting fraudulent transactions [Shirgave et al., 2019] and projecting stock market trends [Gogas and Papadimitriou, 2021], analysing legal documents [Robaldo et al., 2019] and assessing policy outcomes [Kreif and DiazOrdaz, 2019], to screening resumes [Sinha et al., 2021] and automating transportation modes [Yurtsever et al., 2020]. Given the life-changing potential of these models, it is crucial to ensure that they function as intended and instil trust among their users. This is the primary objective of the field of Trustworthy ML.

Ensuring the trustworthiness of machine learning (ML) models goes beyond the accuracy and includes consideration of privacy, fairness, safety, and transparency, as noted in Schmitz et al. [2022]. Protecting individual privacy is paramount to preventing the misuse of sensitive personal information, and approaches such as differentially private ML [Abadi et al., 2016] and decentralised learning [Li et al., 2020] have been developed to address this objective. In contrast, the potential bias in data and ML models may result in discrimination against specific user groups [Mehrabi et al., 2021], underscoring the importance of fairness in algorithmic design. Enhancing the safety of ML is another crucial topic that involves research on uncertainty quantification [Hüllermeier and Waegeman, 2021], adversarial attacks [Kurakin et al., 2016], data poisoning [Goldblum et al., 2022], domain adaptation and generalisation [Zhang et al., 2019].

Increasing transparency in ML [Roscher et al., 2020] is akin to opening up the algorithmic “black box“ to understand how the model operates. This not only enables end-users to better comprehend how results are generated, but also supports learning from algorithmic training.

In this thesis, I present my contributions towards improving the safety and transparency of machine learning by advocating for more principled uncertainty quantification and for more effective explainability tools. Throughout this thesis, I rely heavily on the use of kernel mean embeddings [Muandet et al., 2017] and Gaussian processes [Rasmussen and Williams, 2005b] as non-parametric methods to capture and model probability distributions. Both class of methods rely on positive definite kernels. The former utilise them as reproducing kernels to construct function space and represent distributions as element therein, whereas the latter utilise them as covariance functions. Details of the two class of methods are introduced formally in Chapter 2. Additionally, in Chapter 3, I present a project that demonstrates the use of both methods in conjunction with each other to tackle a challenging image reconstruction problem.

1.2 Thesis Layout and Statements of Authorships

Chapter 2 provides an introduction to kernel methods and Gaussian processes, two crucial methods employed in my research. This chapter also contains an overview of uncertainty modelling and model explainability, two main areas of trustworthy ML I address in this thesis.

In Chapter 3, I demonstrate the use of Gaussian processes with kernel mean embeddings to tackle the problem of reconciling multiple resolutions present in data, also known as statistical downscaling, with the main focus on satellite imaging data. The approach involves formulating low resolution pixels as a conditional expectation of high resolution pixels, and seeking to undo the “aggregation" processes. A novel theoretical analysis of the learning of the *Deconditional Mean Operator* introduced by Hsu and Ramos [2019] is also presented. The work was conducted under the supervision of Prof. Dino Sejdinovic, and I am a joint first author with Shahine Bouabid. The manuscript was accepted at the 2021 Conference on Neural Information Processing Systems (NeurIPS). In this work, I contributed by formulating Bayesian deconditioning as a Gaussian process with conditional expectations as inter-domain features. I also proposed a data efficient conditional mean operator for set inputs, named the *Conditional Mean Shrinkage Operator*, and contributed to the theoretical analysis of deconditioning by expressing the problem as a two-staged vector-valued regression. Bouabid designed and conducted the experiments and developed the variational formulation of the model.

In Chapter 4, the problem of uncertainty quantification when combining multiple causal graphs for inference, also known as causal data fusion, is studied. A novel Bayesian Conditional Mean Embedding

is introduced to model an average treatment effect while capturing both the aleatoric and epistemic uncertainty. We demonstrated the effectiveness of such uncertainty model by applying it to a Causal Bayesian Optimisation problem and demonstrated superior results to baselines. In this work, I am a joint first author with Dr. Jean Francois Ton, and we are jointly supervised by Dr. Javier Gonzalez, Prof. Yee Whye Teh, and Prof. Dino Sejdinovic. Our work has been accepted at the 2021 Conference on Neural Information Processing Systems (NeurIPS). Together with Dr. Ton, I contributed to deriving the Bayesian Conditional Mean Embedding and subsequently, the Bayesian Interventional Mean Processes. Additionally, with Prof. Sejdinovic's guidance, we derived the approximation of the inner product between two Gaussian processes. During this project, Dr. Ton designed and conducted the Causal Bayesian Optimisation experiments, while I designed and conducted the ablation studies.

Chapter 5 presents a novel approach to model explainability for kernel methods in machine learning. Specifically, the RKHS-SHAP algorithm is proposed for models residing in a reproducing kernel Hilbert space (RKHS). In addition, a Shapley attribution prior is formulated to allow for regularisation of attribute contributions to the model. The analysis of robustness of RKHS-SHAP is provided by analysing Shapley values from a functional perspective. For this work, I am the first author, with Dr. Robert Hu as the second author. We were jointly supervised by Dr. Javier Gonzalez and Prof. Dino Sejdinovic. This work was accepted at the 2022 Conference on Neural Information Processing Systems (NeurIPS). I contributed by deriving the RKHS-SHAP and attribution prior formulation, and running most of the experiments. Hu contributed by providing a large scale code implementation code for some appendix experiments. Gonzalez gave valuable insights in early stage and Sejdinovic supervised the whole project.

Chapter 6 deviates from model explainability, serving as a prerequisite for Chapter 7. This chapter presents a non-parametric method for modelling non-rankable preference structures using Gaussian processes, challenging the assumption of rankability in classical preference models. Dr. Javier Gonzalez and Prof. Dino Sejdinovic supervised this work, which was accepted at the 25th International Conference on Artificial Intelligence and Statistics (AISTATS). Collaboratively, we developed the generalized preferential Gaussian processes. I demonstrated that the corresponding kernels fulfil the appropriate notion of universality and proposed a spectral decomposition method for extracting clusters of comparable items from preferential data.

In Chapter 7, I build upon my research from Chapters 5 and 6 to investigate explainability for preference models. It was observed that naive application of existing Shapley explanation algorithms to preference models ignore certain symmetry and that they result in inconsistent explanations. To address this, I proposed the Pref-SHAP algorithm. I am a first joint author of this work along with Dr. Robert Hu, Jaime Huertas is a third author, and the work is supervised by Prof. Dino Sejdinovic. This work is accepted at

the 2022 Conference on Neural Information Processing Systems (NeurIPS). I contributed via formalising and deriving the full Pref-SHAP algorithm. Hu contributed by running the experiments and developing the software.

Finally, in Chapter 8 we conclude the thesis and discuss possible extensions and future directions.

Specifically, the following papers are presented in this thesis:

1. Deconditional downscaling with Gaussian processes
   SL Chau*, S Bouabid*, D Sejdinovic
   Accepted at *Advances in Neural Information Processing Systems, 2021*

2. Bayesimp: Uncertainty quantification for causal data fusion
   SL Chau*, JF Ton*, J Gonzalez, YW Teh, D Sejdinovic
   Accepted at *Advances in Neural Information Processing Systems, 2021*

3. RKHS-SHAP: Shapley values for kernel methods
   SL Chau, R Hu, J Gonzalez, D Sejdinovic
   Accepted at *Advances in Neural Information Processing Systems, 2022*

4. Learning inconsistent preferences with Gaussian processes
   SL Chau, J Gonzalez, D Sejdinovic
   Accepted at *International Conference on Artificial Intelligence and Statistics, 2022*

5. Explaining Preferences with Shapley Values
   R Hu*, **SL Chau***, JF Huertas, D Sejdinovic
   Accepted at *Advances in Neural Information Processing Systems, 2022*

Although not included in this thesis, I have had the privilege of contributing to several other projects, which are listed below.

6. Spectral ranking with covariates
   SL Chau, M Cucuringu, D Sejdinovic
   Accepted at *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2022*

This work is related to the preference modelling problem discussed in Chapter 6. We consider the classical setting and aim to rank players given their incomplete and noisy pairwise comparisons, in light of player covariate information. We proposed three spectral ranking methods that incorporate player covariates and

are based on *seriation*, *low-rank structure* assumption, and *canonical correlation*, respectively. These approaches are then compared with state-of-the-art methods on both synthetic and real-world datasets to demonstrate their effectiveness.

7. Kernel-based graph learning from smooth signals: A functional viewpoint
   X Pu, **SL Chau**, X Dong, D Sejdinovic
   Accepted at *IEEE Transactions on Signal and Information Processing over Networks, 2021*

This work studies the problem of graph learning, where the goal is to construct an explicit topological structure revealing the relationship between nodes representing data entities. We treat observed graph signals - data that adhere to the underlying latent graph smoothness - as functions in the RKHS and integrate functional learning with smoothness constraints to recover the graph.

8. Giga-scale Kernel Matrix Vector Multiplication on GPU
   R Hu, **SL Chau**, D Sejdinovic, JA Glaunes
   Accepted at *Advances in Neural Information Processing Systems, 2022*

The last project concerns about the scalability of kernel matrix-vector multiplication (KMVM), a foundational operation in machine learning and scientific computing. Standard KMVM approach scale quadratically in both memory and time, thus limiting applications to large scale real-world problems. This work proposes an approximation procedure to address such scaling issues for tall (10^8) and skinny ($D \leq 7$) data.

2 | Background

Throughout the thesis, the use of kernel methods and Gaussian processes will be ubiquitous and hence will be introduced formally in Section 2.1 and 2.2. We then give an overview of the two aspects of trustworthy machine learning we tackle in the thesis, which are uncertainty modelling (Section 2.3) and model explainability (Section 2.4).

2.1 Kernel methods

Kernel methods are non-linear algorithms that formulate learning and estimation problems as linear tasks in a reproducing kernel Hilbert space (RKHS) associated with a kernel [Hofmann et al., 2006]. This versatile framework allows us to deploy our algorithm on any data type, e.g. tabular data, images [Mairal, 2016], texts [Lodhi et al., 2002], and graphs [Kriege, 2022], as long as a kernel is well-defined with respect to that data type. The introduction to these concepts here follows closely that of Sejdinovic [2019]. Further details can be found therein as well as in textbooks such as Steinwart and Christmann [2008b], Berlinet and Thomas-Agnan [2004] and Paulsen and Raghupathi [2016]. We begin by defining the notion of a kernel:

Definition 2.1.1 (Kernel function)**.** *Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a kernel if there exists a Hilbert space $\mathcal{H}$ and a map $\phi : \mathcal{X} \to \mathcal{H}$ such that $\forall x, x' \in \mathcal{X}$,*

$$k(x, x') := \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} \tag{2.1}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is an inner product function on $\mathcal{H}$.

The Hilbert space $\mathcal{H}$ of kernel k and the map ϕ are often denoted as feature space and feature map respectively. There is no particular restriction to the dimensionality of $\phi(x)$ and it can be an infinite-dimensional vector. Another important notion for kernel methods is the idea of a reproducing kernel, defined as

Definition 2.1.2 (Reproducing kernel)**.** *Let $\mathcal{H}$ be a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ defined on a non-empty set $\mathcal{X}$. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called reproducing kernel of $\mathcal{H}$ if it satisfies*

- *$\forall x \in \mathcal{X}$, $k_x := k(\cdot, x) \in \mathcal{H}$,*
- *$\forall x \in \mathcal{X}, \forall f \in \mathcal{H}, \langle f, k(\cdot, x) \rangle_{\mathcal{H}} = f(x)$, also known as the reproducing property.*

If $\mathcal{H}$ has a reproducing kernel, it is called a reproducing kernel Hilbert space (RKHS), denoted by $\mathcal{H}_k$.

Note that every reproducing kernel is a kernel with a canonical feature map $\phi : x \mapsto k(\cdot, x)$ and, by Moore-Aronszajn theorem [Berlinet and Thomas-Agnan, 2011], every kernel defines a unique RKHS, therefore the two notions are equivalent. We refer the reader to Paulsen and Raghupathi [2016] for a deeper introduction to the different properties of RKHSs.

Examples of kernels Here we give some examples of kernel functions defined on $\mathbb{R}^d$ for some $d \in \mathbb{N}$.

- **Linear Kernel:** the simplest kernel one could think of: $k(x, x') = x^\top x'$
- **Polynomial kernel:** given a fixed degree p, the polynomial kernel is $k(x, x') = (1 + x^\top x')^p$
- **Radial Basis Function (RBF) kernel:** This infinitely differentiable kernel is arguably the most well known kernel: $k(x, x') = \exp\left(-\frac{||x-x'||_2^2}{l^2}\right)$, where l is a hyperparameter known as the lengthscale. This is the kernel we used throughout the thesis unless specified otherwise.
- **Matérn kernel** takes the following form

$$k(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{d}{\rho}\right) K_\nu \left(\sqrt{2\nu}\frac{d}{\rho}\right)$$

  where Γ is the gamma function, K_ν is the modified Bessel function of the second kind, and ρ and ν are hyperparameter controlling the smoothness of the corresponding function from this RKHS. As functions from the RKHS built using the RBF kernel are infinitely differentiable, they might be considered too smooth for practice. In comparison, Mateérn kernel gives rise to $\nu - 1$ differentiable functions. This kernel is deployed in the experiments from Chapter 3.

Kernel Ridge Regression. Reproducing kernel Hilbert spaces are often used as a function class for empirical risk minimisations (ERM). Let k be a kernel on $\mathcal{X}$ such that $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Recall that the goal in ERM with hypothesis class $\mathcal{H}_k$ is to find a function $f^* \in \mathcal{H}_k$ that solves:

$$\min_{f \in \mathcal{H}_k} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \Omega\left(||f||_{\mathcal{H}_k}^2\right) \tag{2.2}$$

given data $\{x_i, y_i\}_{i=1}^n \overset{i.i.d}{\sim} p(X, Y)$ with p some distribution over domains $\mathcal{X} \times \mathcal{Y}$, loss function L, and non-decreasing regularisation function Ω. The first part of the equation denotes the empirical risk, and the latter part denotes a regularisation of the model complexity via penalising the RKHS norm of the function, which can be interpreted as encouraging smoothness. The representer theorem thus tells us that the optimal solution lies in the span of canonical feature maps of input data points, i.e. $f^* = \sum_{i=1}^n \alpha_i k(\cdot, x_i)$.

When $\mathcal{Y} = \mathbb{R}$, L is the squared loss function, and $\Omega(||f||_{\mathcal{H}_k}^2) = \lambda||f||_{\mathcal{H}_k}^2$ for some $\lambda > 0$, we are solving

a kernel ridge regression (KRR) problem:

$$\sum_i (y_i - f(x_i))^2 + \lambda ||f||^2_{\mathcal{H}_k} = \sum_i (y_i - \langle f, k(\cdot, x_i)\rangle)^2 + \lambda \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) \tag{2.3}$$

$$= (\mathbf{y} - \mathbf{K}\boldsymbol{\alpha})^\top (\mathbf{y} - \mathbf{K}\boldsymbol{\alpha}) + \lambda \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \tag{2.4}$$

where $\mathbf{K}$ is the kernel matrix of size $n \times n$. Differentiate the objective with respect to $\boldsymbol{\alpha}$ and setting to 0, we get the optimal $\boldsymbol{\alpha} = (\mathbf{K} + \lambda I_n)^{-1}\mathbf{y}$. KRR is ubiquitous throughout this thesis. In Chapter 3 we formulate and analyse the deconditional mean operator as a two-staged kernel ridge regressor. Chapter 4 took inspiration from vector-valued KRR and formulate Bayesian conditional mean embedding as a vector-valued Gaussian process. At last, Chapter 5 developed an explanation algorithm for RKHS models and demonstrated its effectiveness using KRR.

Kernel mean embeddings of distributions. Kernel mean embeddings of distributions provide a powerful framework for representing probability distributions in an RKHS [Song et al., 2013, Muandet et al., 2017]. We can define the (marginal) kernel mean embedding of a distribution $\mathbb{P}$ for a random variable X as follows:

$$\mu_{\mathbb{P}_X} := \int k(\cdot, X) d\mathbb{P}(X). \tag{2.5}$$

To obtain an empirical estimate, we simply compute the empirical averages of the feature maps of the available data. For a wide range of kernels, including RBF and Matérn family, this mapping is injective, which is a powerful result that has been used to design metrics between probability distributions, such as the Maximum mean discrepancy (MMD). The MMD equation is defined as $MMD_k^2(\mathbb{P}, \mathbb{Q}) = ||\mu_k(\mathbb{P}) - \mu_k(\mathbb{Q})||^2_{\mathcal{H}_k}$, and $MMD_k(\mathbb{P}, \mathbb{Q}) = 0$ implies that $\mathbb{P} = \mathbb{Q}$ assuming the corresponding kernel is characteristic [Szabó and Sriperumbudur, 2017]. The marginal kernel mean embedding has been applied to various machine learning applications, including two-sample testing [Gretton et al., 2012], independence testing [Gretton et al., 2007], distributional regression [Law et al., 2019], and for training deep generative models [Li et al., 2017].

In addition to capturing marginal distributions, we can also embed conditional distributions like $\mathbb{P}(Y \mid X = x)$ via the conditional mean embedding (CME):

$$\mu_{\mathbb{P}(Y|X=x)} = \int \ell(\cdot, Y) d\mathbb{P}(Y \mid X = x) \tag{2.6}$$

where ℓ is a kernel for Y. However, estimating CME can be more challenging than the marginal case,

as we only have access to pairs of $\{x_i, y_i\}_{i=1}^n$ from the joint distribution. There are two approaches to establish the estimation of CME - the classical operator perspective first considered in Song et al. [2009], and the vector-valued regression perspective proposed in Grünewälder et al. [2012] - which are used in Chapter 3 and Chapter 4 respectively. Despite their different assumptions, both approaches lead to the same empirical estimation using the equation $\hat{\mu}_{\mathbb{P}(Y|X=\cdot)} = k(\cdot, x)^\top \Phi_X (\mathbf{K} + n\lambda I)^{-1} \Phi_Y$, where Φ_X is the collection of feature maps over the data X, k is a kernel, $\mathbf{K}$ is the Gram matrix, and $\lambda > 0$ is a regularisation term. Conditional mean embeddings are widely used to capture complicated relations between random variables. It has been used in a broad range of applications, such as conditional independence tests [Park and Muandet, 2020], dynamical systems [Song et al., 2009], meta learning [Ton et al., 2021b], reinforcement learning [Lever et al., 2016], causal inference [Park et al., 2021, Chau et al., 2021c], and explainability [Chau et al., 2021b] to name a few.

2.2 Gaussian processes

Kernel methods and Gaussian processes are both popular non-parametric approaches based on positive definite kernels. The former utilise them as reproducing kernels to construct RKHSs, whereas the latter utilise them as covariance functions. We refer the reader to Kanagawa et al. [2018] for a deeper discussion on the deep mathematical connection between the two approaches. We start with a classical definition of Gaussian processes, taken from Dudley [2002],

Definition 2.2.1 (Gaussian processes)**.** *Let $\mathcal{X}$ be a nonempty set, $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel and $m : \mathcal{X} \to \mathbb{R}$ be any real-valued function. Then a random function $f : \mathcal{X} \to \mathbb{R}$ is said to be a Gaussian processes (GP) with mean function m and covariance kernel k, denoted by (m, k), if the following holds: For any finite set $X = [x_1, \ldots, x_n]^\top \subset \mathcal{X}$ of size $n \in \mathbb{N}$, the random vector*

$$\mathbf{f}_X := [f(x_1), ..., f(x_n)]^\top \in \mathbb{R}^n$$

follows the multivariate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{K})$ with $\mathbf{m} = [m(x_1), ..., m(x_n)]^\top$ and $\mathbf{K} = [k(x_i, x_j)]_{i,j=1}^n$ the mean vector and covariance matrix, respectively.

Consider the standard regression model $Y = f(X) + \epsilon$ for data $\{(x_i, y_i)\}_{i=1}^n$ and noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ with σ some hyperparameter. When we place a GP prior (m, k) over f, it induces a multivariate normal distribution over the labels $\mathbf{y} := [y_1, ..., y_n]^\top$ with mean $\mathbf{m} = [m(x_1), ..., m(x_n)]^\top$ and covariance structure $\mathbf{K}_{\mathbf{x},\mathbf{x}} = [k(x_i, x_j) + \sigma^2 \delta(i, j)]_{i,j=1}^n$ where δ is the Dirac function. Now given new data $\mathbf{x}_*$, the

predictions $\mathbf{f}_{\mathbf{x}_*}$ and the labels $\mathbf{y}$ follows a joint Gaussian distribution,

$$\begin{bmatrix} \mathbf{f}_{\mathbf{x}_*} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m}_* \\ \mathbf{m} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{\mathbf{x}_*,\mathbf{x}_*} & \mathbf{K}_{\mathbf{x}_*,\mathbf{x}} \\ \mathbf{K}_{\mathbf{x},\mathbf{x}_*} & \mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2 I \end{bmatrix} \right) \tag{2.7}$$

where m_* is the vector of mean function applied on $\mathbf{x}_*$. To obtain the posterior predictive distribution of $p(\mathbf{f}_{\mathbf{x}_*} \mid \mathbf{y})$, we apply the Gaussian conditioning rule,

$$\mathbf{f}_{\mathbf{x}_*} \mid \mathbf{y} \sim \mathcal{N}\left(\mathbf{m}_* + \mathbf{K}_{\mathbf{x}_*,\mathbf{x}_*}(\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2 I)^{-1}(\mathbf{y} - \mathbf{m}), \mathbf{K}_{\mathbf{x}_*,\mathbf{x}_*} - \mathbf{K}_{\mathbf{x}_*,\mathbf{x}}(\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2 I)^{-1}\mathbf{K}_{\mathbf{x},\mathbf{x}_*}\right) \tag{2.8}$$

this gives us a closed-form expression for the posterior distribution of the evaluations of f at any collection of inputs.

If we pick $m = 0$ as the zero mean function, the posterior mean function coincides with the estimates of a KRR with the same kernel k and regularisation term $\lambda = \sigma^2$. In practice, a GP model might be preferred because of two reasons: First, the posterior variance could be used as a measure of uncertainty, which can be an important indicator when making predictions far away from the input data. Second, the hyperparameters in KRR are often chosen by cross-validation and grid search, while the hyperparameters in GP can be optimised using the maximum marginal likelihood principle. These advantages led us to focus on GP models in Chapters 3, 4, and 6.

Gaussian processes are widely used in this thesis. In Chapter 3, we utilise the joint normality between the latent function of interests and their observed conditional expectations to recover a posterior over the function of interests. In Chapter 4, we introduce a vector-valued Gaussian Process formulation to model a Bayesian Conditional Mean Embedding. Furthermore, we demonstrate how to estimate uncertainties between the inner products of GPs. In Chapter 6, a Gaussian process over pairwise comparison is proposed to model preferences with inconsistent structures, such as cyclic relationships.

2.3 Uncertainty Quantification

The preceding sections provided an overview of the model classes that constitute the core focus of this thesis. In the following, we delve into the specific challenges we aim to address, starting with the topic of uncertainty modeling.

Uncertainty modeling is a fundamental component of machine learning, particularly when deploying models in safety-critical domains such as forensics and self-driving car. Uncertainty can manifest at different stages of the machine learning pipeline, stemming from factors like collecting noisy or imprecise

data [He et al., 2009], incorrect model assumptions [Aydogan et al., 2020], or encountering distribution shifts during testing [Wen et al., 2014]. Given the practical challenges of verifying or falsifying such scenarios, uncertainty is inherently present in our predictions. Therefore, it is desirable to have a trustworthy representation of uncertainty as a key feature of any machine learning model.

There are two main sources of uncertainty, often referred to as *aleatoric* and *epistemic* uncertainty [Hüllermeier and Waegeman, 2021]. Aleatoric uncertainty describes the intrinsic randomness inherent in the phenomena we model whereas epistemic uncertainty describes the estimation uncertainty we incur under finite observations of the phenomena. Unlike aleatoric uncertainty, epistemic uncertainty can, in principle, be reduced when additional information is provided to the model.

Distinguishing between aleatoric and epistemic uncertainty is important to avoid overconfident predictions, i.e. it is important to distinguish between a doctor being very confident that there is a 50% chance the tumour is cancerous versus they being very not confident but estimate that the tumour is cancerous with 100% chance. Gaussian process can help achieve this separation. Referring to the notations from the previous section, the variance of the additive error term σ^2 in a regression setting corresponds to the aleatoric uncertainty, while the posterior covariance for a query x_* is a meaningful indicator of the total uncertainty. Thus, the epistemic uncertainty could be obtained by subtracting the total uncertainty from the aleatoric uncertainty. In Chapter 4, we will show how we can combine uncertainties from two data sources to estimate a conditional treatment effect while presenting an uncertainty estimate that captures both aleatoric and epistemic uncertainties from two data sources.

2.4 Model Explanation

In addition to quantifying prediction uncertainty, it is crucial to understand a model's decision when deploying machine learning. Although the terms "explainability" and "interpretability" are often used interchangeably, there is no precise mathematical definition for either. However, it is important to differentiate between an explainable model and an explanation method. Explainable models have been around for over a century since the introduction of linear models. With a linear model, one can easily understand the results by examining the model coefficients. However, as the field of machine learning advances, we are developing more complex models to handle challenging data types like images, text, and graphs and to capture more detailed non-linear patterns in large datasets, which makes our models less interpretable and more like "black boxes." To address this issue, explanation methods have been developed.

There are various types of explanation methods:

- **Counterfactual explanation** provide users a certain level of recommendation on how they should modify their model inputs to receive a preferred model decision, see e.g. Wachter et al. [2017] for more.
- **Global explanation** focuses on the overall model behaviour and provides a pattern that the prediction model discovered in general. For example, a partial dependence analysis [Friedman, 2001, Greenwell et al., 2018] shows the marginal effect one or two features have on the predicted outcome of a model to determine feature importance.
- **Local explanation** is only concerned with model decision corresponding to specific data point. These methods [Ribeiro et al., 2016, Lundberg and Lee, 2017] often place interpretable (linear or tree) surrogate models around the decision boundary of the specific data point to explain the complex decision at a local level.

In this thesis, we focus on local explanation methods. We take the popular approach to formulate a local explanation as a cooperative game among model features. This allows us to utilise classical concepts such as Shapley values [Shapley, 1953] to arrive at explanations with several favourable axioms. In Chapter 5, we build on an existing line of research where model-specific Shapley value computation algorithms are proposed. Similar to how DeepSHAP [Lundberg and Lee, 2017] is proposed for deep learning and TreeSHAP [Lundberg et al., 2018] is proposed for tree-based models, we propose an RKHS-SHAP for computing Shapley values with RKHS models. In Chapter 7 we focus on designing an appropriate cooperative game to arrive at the right explanation for preference models.

3 | Deconditional Downscaling with Gaussian Processes

This chapter is based on the following publication:

> **Siu Lun Chau***, Shahine Bouabid*, and Dino Sejdinovic. "Deconditional Downscaling with Gaussian Processes." *Advances in Neural Information Processing Systems (NeurIPS), 2021*

Abstract

Refining low-resolution (LR) spatial fields with high-resolution (HR) information is challenging as the diversity of spatial datasets often prevents direct matching of observations. Yet, when LR samples are modeled as aggregate conditional means of HR samples with respect to a mediating variable that is globally observed, the recovery of the underlying fine-grained field can be framed as taking an "inverse" of the conditional expectation, namely a *deconditioning problem*. In this work, we introduce *conditional mean process* (CMP), a new class of Gaussian Processes describing conditional means. By treating CMPs as inter-domain features of the underlying field, a posterior for the latent field can be established as a solution to the deconditioning problem. Furthermore, we show that this solution can be viewed as a two-staged vector-valued kernel ridge regressor and show that it has a minimax optimal convergence rate under mild assumptions. Lastly, we demonstrate its proficiency in a synthetic and a real-world atmospheric field downscaling problem, showing substantial improvements over existing methods.

3.1 Introduction

Spatial observations often operate at limited resolution due to practical constraints. For example, remote sensing atmosphere products [Remer et al., 2005, Platnick et al., 2003, Stephens et al., 2002, Barnes et al., 1998] provide a measurement of atmospheric properties such as cloud top temperatures and optical thickness, but only at a low resolution. This hinders local scale analysis of variables playing a critical role in our understanding of the anthropogenic impact on climate.

By postulating an underlying detailed spatial field that aggregates into the coarse observations, it becomes possible to frame the inference of high-resolution samples as weakly-supervised learning [Zhou, 2017] given coarse aggregated targets. This problem, also known as *statistical downscaling* or *spatial disaggregation*, has been studied in the literature in various forms, notably giving it a probabilistic treatment [Zhang et al., 2020, Law et al., 2018c, Hamelijnck et al., 2019, Yousefi et al., 2019, Tanaka et al., 2019, Ville Tanskanen , Krista Longi, 2020], in which Gaussian processes (GP) [Rasmussen and Williams, 2005b] or deeper combinations of GPs [Wilson et al., 2012, Damianou and Lawrence, 2013] are typically used in conjunction with a sparse variational formulation [Titsias, 2009] to recover the underlying latent field at high resolution.

However, these models require access to bags of high-resolution (HR) covariates that are paired with aggregate targets, which in practice might be infeasible. For example, distinct satellite trajectories, instrument limitations or variability of atmospheric conditions often lead to coarse and fine-grained covariates that are unmatched spatially and temporally. Climate simulations [Eyring et al., 2016, Flato, 2011, Scholze et al., 2012], on the other hand, provide a comprehensive low-resolution (LR) coverage of meteorological variables that can be matched to both coarse and high-resolution covariates. This motivates us to introduce a less restrictive mediated formulation of statistical downscaling that only requires indirect matching between low and high-resolution covariates.

Formally, let ${}^b\boldsymbol{x} = \{x^{(1)}, \dots, x^{(n)}\} \subset \mathcal{X}$ be a general notation for bags of HR covariates, $f : \mathcal{X} \to \mathbb{R}$ the field of interest we wish to recover and $\tilde{z}$ observations from this field but at LR. We suppose that ${}^b\boldsymbol{x}$ and $\tilde{z}$ are unmatched, but that there exists LR covariates $y, \tilde{y} \in \mathcal{Y}$, such that y is jointly observed with ${}^b\boldsymbol{x}$ and likewise $\tilde{y}$ with $\tilde{z}$ as illustrated in Figure 5.1. Then y and $\tilde{y}$ can be used to mediate a two-staged learning process as follows: **(i)** Model the aggregate observation process by $\tilde{z} = \mathbb{E}[f(X)|Y = \tilde{y}] + \varepsilon$ with some noise ε ; **(ii)** Separately learn the conditional expectation operator $y \mapsto \mathbb{E}[f(X)|Y = y]$ using $\{{}^b\boldsymbol{x}, y\}$.

The task of recovering f under observation model $\tilde{z} = \mathbb{E}[f(X)|Y = \tilde{y}] + \varepsilon$ is referred to as *deconditioning*. Motivated by applications in likelihood-free inference and task-transfer regression, Hsu and Ramos [2019]

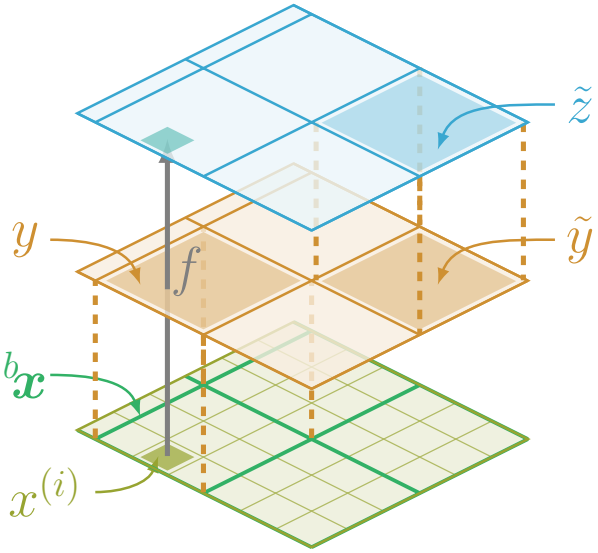


Figure 3.1: The LR response $\tilde{z}$ (blue) and the bag HR covariates ${}^{b}\boldsymbol{x}$ (green) are unmatched. The downscaling is mediated through bag-level LR covariates y and $\tilde{y}$ (orange).

first studied the deconditioning problem through the lens of reproducing kernel Hilbert space (RKHS) and introduced the framework of *Deconditional Mean Embeddings* (DME) as its solution. In this work, motivated by the application to mediated statistical downscaling, we extend deconditioning to a multi-resolution setup, which is not considered by Hsu and Ramos [2019]. Placing a GP prior on the sought field f, we first characterise the stochastic process resulting from taking the conditional expectations of f, which we term the *Conditional Mean Process* (CMP). By exploiting the joint normality of the sought field and its CMP, we can obtain the posterior distribution of f as a principled Bayesian solution to the downscaling task on indirectly matched data. The posterior mean under our model elegantly recovers the DME-based estimator of Hsu and Ramos [2019] and the posterior covariance reflects confidence on the deconditioning quality. While Hsu and Ramos [2019] also consider a Bayesian interpretation of DME, they do this through the lens of task-transformed regression and do not link f to its CMP. Rather, the likelihoods they use imply $\tilde{z} = f(x)$ which does not reflect a different granularity level of the aggregate response $\tilde{z}$.

Our approach also admits a tractable variational inference approximation method, enabling scalability to large datasets. In addition, we study the theoretical properties of Deconditional Mean Operator (DMO) associated with DME, by establishing it as a two-staged vector-valued regressor with a natural reconstruction loss. This perspective allows us to leverage vector-valued regression theory and obtain novel convergence rate results for the DMO estimator. Under mild assumptions, we obtain conditions under which this rate is minimax optimal in terms of statistical-computational efficiency. Our contributions are summarized as follows:

- We propose a Bayesian formulation of the mediated statistical downscaling problem, and establish its posterior mean as a DME-based estimate and its posterior covariance as a gauge of the deconditioning quality.
- We demonstrate that the DMO estimate minimises a two-staged vector-valued regression and derives its convergence rate under mild assumptions, with conditions for minimax optimality.
- We benchmark our model against existing methods for spatial disaggregation tasks, on both synthetic and real-world multi-resolution atmospheric fields data, and show improved performance

3.2 Background Materials

3.2.1 Notations

Let X, Y be a pair of random variables taking values in non-empty sets $\mathcal{X}$ and $\mathcal{Y}$, respectively. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be positive definite kernels. The closure of the span of their canonical feature maps $k_x := k(x, \cdot)$ and $\ell_y := \ell(y, \cdot)$ for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ respectively induces RKHS $\mathcal{H}_k \subseteq \mathbb{R}^{\mathcal{X}}$ and $\mathcal{H}_\ell \subseteq \mathbb{R}^{\mathcal{Y}}$ endowed with inner products $\langle \cdot, \cdot \rangle_k$ and $\langle \cdot, \cdot \rangle_\ell$.

We observe realizations ${}^b\mathcal{D}_1 = \{{}^b\boldsymbol{x}_j, y_j\}_{j=1}^N$ of bags ${}^b\boldsymbol{x}_j = \{x_j^{(i)}\}_{i=1}^{n_j}$ from conditional distribution $\mathbb{P}_{X|Y=y_j}$, with bag-level covariates y_j sampled from $\mathbb{P}_Y$. We concatenate them into vectors $\mathbf{x} := \begin{bmatrix} {}^b\boldsymbol{x}_1 & \dots & {}^b\boldsymbol{x}_N \end{bmatrix}^\top$ and $\mathbf{y} := \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}^\top$. For simplicity, our exposition will initially use the notation without bagging of the observations, i.e. where $\mathcal{D}_1 = \{x_j, y_j\}_{j=1}^N$. We will come back to a bagged dataset formalism in Section 3.3.3, which corresponds to our motivating application of statistical downscaling.

With an abuse of notation, we define feature matrices by stacking feature maps along columns as $\boldsymbol{\Phi}_{\mathbf{x}} := \begin{bmatrix} k_{x_1} & \dots & k_{x_N} \end{bmatrix}$ and $\boldsymbol{\Psi}_{\mathbf{y}} := \begin{bmatrix} \ell_{y_1} & \dots & \ell_{y_N} \end{bmatrix}$ and we denote Gram matrices as $\mathbf{K}_{\mathbf{xx}} = \boldsymbol{\Phi}_{\mathbf{x}}^\top \boldsymbol{\Phi}_{\mathbf{x}} = [k(x_i, x_j)]_{1 \le i,j \le N}$ and $\mathbf{L}_{\mathbf{yy}} = \boldsymbol{\Psi}_{\mathbf{y}}^\top \boldsymbol{\Psi}_{\mathbf{y}} = [\ell(y_i, y_j)]_{1 \le i,j \le N}$. The notation abuse $(\cdot)^\top(\cdot)$ is a shorthand for the elementwise RKHS inner products when it is clear from the context.

Let Z denote the real-valued random variable stemming from the noisy conditional expectation of some unknown latent function $f : \mathcal{X} \to \mathbb{R}$, as $Z = \mathbb{E}[f(X)|Y] + \varepsilon$. We suppose one observes another set of realizations $\mathcal{D}_2 = \{\tilde{y}_j, \tilde{z}_j\}_{j=1}^M$ from $\mathbb{P}_{YZ}$, which is sampled independently from $\mathcal{D}_1$. Likewise, we stack observations into vectors $\tilde{\mathbf{y}} := \begin{bmatrix} \tilde{y}_1 & \dots & \tilde{y}_M \end{bmatrix}^\top$, $\tilde{\mathbf{z}} := \begin{bmatrix} \tilde{z}_1 & \dots & \tilde{z}_M \end{bmatrix}^\top$ and define feature map $\boldsymbol{\Psi}_{\tilde{\mathbf{y}}} := \begin{bmatrix} \ell_{\tilde{y}_1} & \dots & \ell_{\tilde{y}_M} \end{bmatrix}$.

3.2.2 Conditional and Deconditional Kernel Mean Embeddings

Marginal and Joint Mean Embeddings Kernel mean embeddings of distributions provide a powerful framework for representing and manipulating distributions without specifying their parametric form [Song et al., 2013, Muandet et al., 2016a]. The marginal mean embedding of measure $\mathbb{P}_X$ is defined as $\mu_X := \mathbb{E}[k_X] \in \mathcal{H}_k$ and corresponds to the Riesz representer of expectation functional $f \mapsto \mathbb{E}[f(X)]$. It can hence be used to evaluate expectations $\mathbb{E}[f(X)] = \langle f, \mu_X \rangle_k$. If the mapping $\mathbb{P}_X \mapsto \mu_X$ is injective, the kernel k is said to be characteristic [Fukumizu et al., 2004], a property satisfied for the Gaussian and Matérn kernels on $\mathbb{R}^d$ [Fukumizu et al., 2008]. In practice, Monte Carlo estimator $\hat{\mu}_X := \frac{1}{N}\sum_{i=1}^N k_{x_i}$ provides an unbiased estimate of μ_X [Sriperumbudur et al., 2012].

Extending this rationale to embeddings of joint distributions, we define $C_{YY} := \mathbb{E}[\ell_Y \otimes \ell_Y] \in \mathcal{H}_\ell \otimes \mathcal{H}_\ell$ and $C_{XY} := \mathbb{E}[k_X \otimes \ell_Y] \in \mathcal{H}_k \otimes \mathcal{H}_\ell$, which can be identified with the cross-covariance operators between Hilbert spaces $C_{YY} : \mathcal{H}_\ell \to \mathcal{H}_\ell$ and $C_{XY} : \mathcal{H}_\ell \to \mathcal{H}_k$. They correspond to the Riesz representers of bilinear forms $(g, g') \mapsto \mathrm{Cov}(g(Y), g'(Y)) = \langle g, C_{YY} g' \rangle_\ell$ and $(f, g) \mapsto \mathrm{Cov}(f(X), g(Y)) = \langle f, C_{XY} g \rangle_k$. As above, empirical estimates are obtained as $\hat{C}_{YY} = \frac{1}{N}\mathbf{\Psi_y}\mathbf{\Psi_y}^\top = \frac{1}{N}\sum_{i=1}^N \ell_{y_i} \otimes \ell_{y_i}$ and $\hat{C}_{XY} = \frac{1}{N}\mathbf{\Phi_x}\mathbf{\Psi_y}^\top = \frac{1}{N}\sum_{i=1}^N k_{x_i} \otimes \ell_{y_i}$.

Conditional Mean Embeddings Similarly, one can introduce RKHS embeddings for conditional distributions referred to as *Conditional Mean Embeddings* (CME). The CME of conditional probability measure $\mathbb{P}_{X|Y=y}$ is defined as $\mu_{X|Y=y} := \mathbb{E}[k_X | Y = y] \in \mathcal{H}_k$. As introduced by Fukumizu et al. [2004], it is common to formulate conditioning in terms of a Hilbert space operator $C_{X|Y} : \mathcal{H}_\ell \to \mathcal{H}_k$ called the *Conditional Mean Operator* (CMO). $C_{X|Y}$ satisfies by definition $C_{X|Y}\ell_y = \mu_{X|Y=y}$ and $C_{X|Y}^\top f = \mathbb{E}[f(X)|Y = \cdot\,], \forall f \in \mathcal{H}_k$, where $C_{X|Y}^\top$ denotes the adjoint of $C_{X|Y}$. Plus, the CMO admits expression $C_{X|Y} = C_{XY}C_{YY}^{-1}$, provided $\ell_y \in \mathrm{Range}(C_{YY})$, $\forall y \in \mathcal{Y}$ [Fukumizu et al., 2004, Song et al., 2013]. Song et al. [2009] show that a nonparametric empirical form of the CMO writes

$$\hat{C}_{X|Y} = \mathbf{\Phi_x}(\mathbf{L_{yy}} + N\lambda \mathbf{I}_N)^{-1}\mathbf{\Psi_y}^\top, \tag{3.1}$$

where $\lambda > 0$ is some regularisation ensuring the empirical operator is globally defined and bounded.

As observed by Grünewälder et al. [2012], since $C_{X|Y}$ defines a mapping from $\mathcal{H}_\ell$ to $\mathcal{H}_k$, it can be interpreted as the solution to a vector-valued regression problem. This perspective enables derivation of probabilistic convergence bounds on the empirical CMO estimator [Grünewälder et al., 2012, Singh et al., 2019].

Deconditional Mean Embeddings Introduced by Hsu and Ramos [2019] as a new class of embeddings, *Deconditional Mean Embeddings* (DME) are natural counterparts of CMEs. While CME $\mu_{X|Y=y} \in \mathcal{H}_k$ allows to take the conditional expectation of any $f \in \mathcal{H}_k$ through inner product $\mathbb{E}[f(X)|Y = y] = \langle f, \mu_{X|Y=y} \rangle_k$, the DME denoted $\mu_{X=x|Y} \in \mathcal{H}_\ell$ solves the inverse problem[1] and allows to recover the initial function of which the conditional expectation was taken, through inner product

$$\langle \mathbb{E}[f(X)|Y = \cdot], \mu_{X=x|Y} \rangle_\ell = f(x).$$

The associated Hilbert space operator, the *Deconditional Mean Operator* (DMO), is thus defined as the operator $D_{X|Y} : \mathcal{H}_k \to \mathcal{H}_\ell$ such that

$$D_{X|Y}^\top \mathbb{E}[f(X)|Y = \cdot] = f \, , \forall f \in \mathcal{H}_k.$$

It admits an expression in terms of CMO and cross-covariance operators

$$D_{X|Y} = (C_{X|Y} C_{YY})^\top (C_{X|Y} C_{YY} (C_{X|Y})^\top)^{-1}$$

provided $\ell_y \in \text{Range}(C_{YY})$ and $k_x \in \text{Range}(C_{X|Y} C_{YY} C_{X|Y}^\top) \, , \forall y \in \mathcal{Y}$ and $\forall x \in \mathcal{X}$. Later in Section 3.4 the connection between DMO and CMO will be made explicit by formulating DMO as a reconstruction operator involving CMO.

A nonparametric empirical estimate of the DMO using datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ as described above, is given by $\hat{D}_{X|Y} = \mathbf{\Psi}_{\tilde{\mathbf{y}}} \left(\mathbf{A}^\top \mathbf{K}_{\mathbf{xx}} \mathbf{A} + M\epsilon \mathbf{I}_M \right)^{-1} \mathbf{A}^\top \mathbf{\Phi}_{\mathbf{x}}^\top$ where $\epsilon > 0$ is a regularisation term and $\mathbf{A} := (\mathbf{L}_{\mathbf{yy}} + N\lambda \mathbf{I})^{-1} \mathbf{L}_{\mathbf{y}\tilde{\mathbf{y}}}$ can be interpreted as an aggregation operator. Applying the DMO to expected responses $\tilde{\mathbf{z}}$, Hsu and Ramos [2019] are able to recover an estimate of f as

$$\hat{f}(x) = k(x, \mathbf{x}) \mathbf{A} \left(\mathbf{A}^\top \mathbf{K}_{\mathbf{xx}} \mathbf{A} + M\epsilon \mathbf{I}_M \right)^{-1} \tilde{\mathbf{z}}. \tag{3.2}$$

Note that since separate samples $\tilde{\mathbf{y}}$ can be used to estimate C_{YY}, this naturally fits a mediating variables setup where $\mathbf{x}$ and the conditional means $\tilde{\mathbf{z}}$ are not jointly observed.

[1] we adopt slightly unusual notation $\mu_{X=x|Y}$ from Hsu and Ramos [2019] which is meant to contrast the usual conditioning $\mu_{X|Y=y}$

3.3 Methodology

In this section, we introduce *Conditional Mean Process* (CMP), a stochastic process stemming from the conditional expectation of a GP. We provide a characterisation of the CMP and show that the corresponding posterior mean of its integrand is a DME-based estimator. We also derive in Appendix A.3 a variational formulation of our model that scales to large datasets and demonstrate its performance in Section 3.5. In what follows, $\mathcal{X}$ is a measurable space, $\mathcal{Y}$ a Borel space, and feature maps k_x and ℓ_y are Borel-measurable functions for any $x \in \mathcal{X}$, $y \in \mathcal{Y}$. All proofs and derivations of the chapter are included in the appendix.

3.3.1 Conditional Mean Process

Bayesian quadrature [Larkin, 1972, Briol et al., 2019, Rasmussen and Williams, 2005b] is based on the observation that the integral of a GP with respect to some marginal measure is a Gaussian random variable. We start by probing the implications of integrating with respect to conditional distribution $\mathbb{P}_{X|Y=y}$ and considering such integrals as functions of the conditioning variable y. This gives rise to the notion of conditional mean processes.

Definition 3.3.1 (Conditional Mean Process)**.** *Let $f \sim \mathcal{GP}(m, k)$ with integrable sample paths, i.e. $\int_{\mathcal{X}} |f| \, \mathrm{d}\mathbb{P}_X < \infty$ a.s. The CMP induced by f with respect to $\mathbb{P}_{X|Y}$ is defined as the stochastic process $\{g(y) : y \in \mathcal{Y}\}$ given by $g(y) = \int_{\mathcal{X}} f(x) \, \mathrm{d}\mathbb{P}_{X|Y=y}(x)$.*

By linearity of the integral, it is clear that $g(y)$ is a Gaussian random variable for each $y \in \mathcal{Y}$. The sample paths integrability requirement ensures g is well-defined almost everywhere. The following result characterizes CMP as a GP on $\mathcal{Y}$.

Proposition 3.3.2 (CMP characterization)**.** *Suppose $\mathbb{E}[|m(X)|] < \infty$ and $\mathbb{E}[|k_X|_k] < \infty$ and let $(X', Y') \sim \mathbb{P}_{XY}$. Then g is a Gaussian process $g \sim \mathcal{GP}(\nu, q)$ a.s. , specified by*

$$\nu(y) = \mathbb{E}[m(X)|Y = y] \qquad q(y, y') = \mathbb{E}[k(X, X')|Y = y, Y' = y'] \tag{3.3}$$

$\forall y, y' \in \mathcal{Y}$. Furthermore $q(y, y') = \langle \mu_{X|Y=y}, \mu_{X|Y=y'} \rangle_k$ a.s.

Intuitively, the CMP can be understood as a GP on the conditional means where its covariance $q(y, y')$ is induced by the similarity between the CMEs at y and y'. Resorting to the kernel ℓ defined on $\mathcal{Y}$, we can reexpress the covariance using Hilbert space operators as $q(y, y') = \langle C_{X|Y}\ell_y, C_{X|Y}\ell_{y'} \rangle_k$. A natural nonparametric estimate of the CMP covariance thus comes using the CMO estimator from

(3.1) as $\hat{q}(y, y') = \ell(y, \mathbf{y}) \left(\mathbf{L_{yy}} + N\lambda \mathbf{I}_N\right)^{-1} \mathbf{K_{xx}} \left(\mathbf{L_{yy}} + N\lambda \mathbf{I}_N\right)^{-1} \ell(\mathbf{y}, y')$. When $m \in \mathcal{H}_k$, the mean function can be written as $\nu(y) = \langle \mu_{X|Y=y}, m \rangle_k$ for which we can also use an empirical estimate $\hat{\nu}(y) = \ell(y, \mathbf{y}) \left(\mathbf{L_{yy}} + N\lambda \mathbf{I}_N\right)^{-1} \mathbf{\Phi}_{\mathbf{x}}^\top m$. Finally, one can also quantify the covariance between the CMP g and its integrand f, i.e. $\mathrm{Cov}(f(x), g(y)) = \mathbb{E}[k(x, X)|Y = y]$. Under the same assumptions as Proposition 3.3.2, this covariance can be expressed using mean embeddings, i.e. $\mathrm{Cov}(f(x), g(y)) = \langle k_x, \mu_{X|Y=y} \rangle_k$ and admits empirical estimate $k(x, \mathbf{x}) \left(\mathbf{L_{yy}} + N\lambda \mathbf{I}_N\right)^{-1} \ell(\mathbf{y}, y)$.

3.3.2 Deconditional Posterior

Given independent observations introduced above, $\mathcal{D}_1 = \{\mathbf{x}, \mathbf{y}\}$ and $\mathcal{D}_2 = \{\tilde{\mathbf{y}}, \tilde{\mathbf{z}}\}$, we may now consider an additive noise model with CMP prior on aggregate observations $\tilde{\mathbf{z}}|\tilde{\mathbf{y}} \sim \mathcal{N}(g(\tilde{\mathbf{y}}), \sigma^2 \mathbf{I}_M)$. Let $\mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} := q(\tilde{\mathbf{y}}, \tilde{\mathbf{y}})$ be the kernel matrix induced by q on $\tilde{\mathbf{y}}$ and let $\mathbf{\Upsilon} := \mathrm{Cov}(f(\mathbf{x}), \tilde{\mathbf{z}}) = \mathbf{\Phi}_{\mathbf{x}}^\top C_{X|Y} \mathbf{\Psi}_{\tilde{\mathbf{y}}}$ be the cross-covariance between $f(\mathbf{x})$ and $\tilde{\mathbf{z}}$. The joint normality of $f(\mathbf{x})$ and $\tilde{\mathbf{z}}$ gives

$$\begin{bmatrix} f(\mathbf{x}) \\ \tilde{\mathbf{z}} \end{bmatrix} \mid \mathbf{y}, \tilde{\mathbf{y}} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{x}) \\ \nu(\tilde{\mathbf{y}}) \end{bmatrix}, \begin{bmatrix} \mathbf{K_{xx}} & \mathbf{\Upsilon} \\ \mathbf{\Upsilon}^\top & \mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2 \mathbf{I}_M \end{bmatrix} \right). \tag{3.4}$$

Using Gaussian conditioning, we can then readily derive the posterior distribution of the underlying GP field f given the aggregate observations $\tilde{\mathbf{z}}$ corresponding to $\tilde{\mathbf{y}}$. The latter can naturally be degenerated if observations are paired, i.e. $\mathbf{y} = \tilde{\mathbf{y}}$. This formulation can be seen as an example of the inter-domain GP [Rudner et al., 2020], where we utilise the observed conditional means $\tilde{\mathbf{z}}$ as inter-domain inducing features for inference of f.

Proposition 3.3.3 (Deconditional Posterior)**.** *Given aggregate observations $\tilde{\mathbf{z}}$ with homoscedastic noise σ^2, the deconditional posterior of f is defined as the Gaussian process $f|\tilde{\mathbf{z}} \sim \mathcal{GP}(m_\mathrm{d}, k_\mathrm{d})$ where*

$$m_\mathrm{d}(x) = m(x) + k_x^\top C_{X|Y} \mathbf{\Psi}_{\tilde{\mathbf{y}}} (\mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2 \mathbf{I}_M)^{-1} (\tilde{\mathbf{z}} - \nu(\tilde{\mathbf{y}})), \tag{3.5}$$

$$k_\mathrm{d}(x, x') = k(x, x') - k_x^\top C_{X|Y} \mathbf{\Psi}_{\tilde{\mathbf{y}}} (\mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2 \mathbf{I}_M)^{-1} \mathbf{\Psi}_{\tilde{\mathbf{y}}}^\top C_{X|Y}^\top k_{x'}. \tag{3.6}$$

Substituting terms by their empirical forms, we can define a nonparametric estimate of the m_d as

$$\hat{m}_\mathrm{d}(x) := m(x) + k(x, \mathbf{x}) \mathbf{A} (\hat{\mathbf{Q}}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2 \mathbf{I}_M)^{-1} (\tilde{\mathbf{z}} - \hat{\nu}(\tilde{\mathbf{y}}))) \tag{3.7}$$

which, when $m = 0$, reduces to the DME-based estimator in (3.2) by taking the noise variance $\frac{\sigma^2}{N}$ as the inverse regularization parameter. Hsu and Ramos [2019] recover a similar posterior mean expression

in their Bayesian interpretation of DME. However, they do not link the distributions of f and its CMP, which leads to much more complicated chained inference derivations combining fully Bayesian and MAP estimates, while we naturally recover it using simple Gaussian conditioning.

Likewise, an empirical estimate of the deconditional covariance is given by

$$\hat{k}_{\mathrm{d}}(x, x') := k(x, x') - k(x, \mathbf{x})\mathbf{A}(\hat{\mathbf{Q}}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2 \mathbf{I}_M)^{-1}\mathbf{A}^\top k(\mathbf{x}, x'). \tag{3.8}$$

Interestingly, the latter can be rearranged to write as the difference between the original kernel and kernel undergoing conditioning and deconditioning steps $\hat{k}_{\mathrm{d}}(x, x') = k(x, x') - \langle k_x, \hat{D}_{X|Y}\hat{C}_{X|Y}k_{x'}\rangle_k$. This can be interpreted as a measure of reconstruction quality, which degenerates in the case of perfect deconditioning, i.e. $\hat{D}_{X|Y}\hat{C}_{X|Y} = \mathrm{Id}_{\mathcal{H}_k}$.

3.3.3 Deconditioning and multiresolution data

Downscaling application would typically correspond to multiresolution data, with bag dataset ${}^b\mathcal{D}_1 = \{({}^b\boldsymbol{x}_j, y_j)\}_{j=1}^N$ where ${}^b\boldsymbol{x}_j = \{x_j^{(i)}\}_{i=1}^{n_j}$. In this setup, the mismatch in size between vector concatenations $\mathbf{x} = [x_1^{(1)} \ \dots \ x_N^{(n_N)}]^\top$ and $\mathbf{y} = \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}^\top$ prevents from readily applying (3.1) to estimate the CMO and thus infer the deconditional posterior. There is, however, a straightforward approach to alleviate this: simply replicate bag-level covariates y_j to match bags sizes n_j. Although simple, this method incurs a $\mathcal{O}((\sum_{j=1}^N n_j)^3)$ computational cost due to matrix inversion in (3.1). Alternatively, since bags ${}^b\boldsymbol{x}_j$ are sampled from conditional distribution $\mathbb{P}_{X|Y=y_j}$, unbiased Monte Carlo estimators of CMEs are given by $\hat{\mu}_{X|Y=y_j} = \frac{1}{n_j}\sum_{i=1}^{n_j} k_{x_j^{(i)}}$. Let $\hat{\mathbf{M}}_{\mathbf{y}} = [\hat{\mu}_{X|Y=y_1} \dots \hat{\mu}_{X|Y=y_N}]^\top$ denote their concatenation along columns. We can then rewrite the cross-covariance operator as $C_{XY} = \mathbb{E}_Y[\mathbb{E}_{X|Y}[k_X] \otimes \ell_Y]$ and hence take $\frac{1}{N}\hat{\mathbf{M}}_{\mathbf{y}}\mathbf{\Psi}_{\mathbf{y}}^\top$ as an estimate for C_{XY}. Substituting empirical forms into $C_{X|Y} = C_{X|Y}C_{YY}^{-1}$ and applying Woodbury identity, we obtain an alternative CMO estimator that only requires an inversion of a $N \times N$ matrix. We call it *Conditional Mean Shrinkage Operator* and define it as

$${}^S\hat{C}_{X|Y} := \hat{\mathbf{M}}_{\mathbf{y}}(\mathbf{L}_{\mathbf{yy}} + \lambda N \mathbf{I}_N)^{-1}\mathbf{\Psi}_{\mathbf{y}}^\top. \tag{3.9}$$

This estimator can be seen as a generalisation of the Kernel Mean Shrinkage Estimator [Muandet et al., 2016b] to the conditional case. We provide in Appendix A.4 modifications of (3.7) and (3.8) including this estimator for the computation of the deconditional posterior.

3.4 Deconditioning as a regression

In Section 3.3.2, we obtain a DMO-based estimate for the posterior mean of $f|\tilde{\mathbf{z}}$. When the estimate gets closer to the exact operator, the uncertainty collapses and the Bayesian view meets the frequentist. It is however unclear how the empirical operators effectively converge in finite data size. Adopting an alternative perspective, we now demonstrate that the DMO estimate can be obtained as the minimiser of a two-staged vector-valued regression. This frequentist turn enables us to leverage rich theory of vector-valued regression and establish under mild assumptions a convergence rate on the DMO estimator, with conditions to fulfill minimax optimality in terms of statistical-computational efficiency. In the following, we briefly review CMO's vector-valued regression viewpoint and construct an analogous regression problem for DMO. We refer the reader to Paulsen and Raghupathi [2016] for a comprehensive overview of vector-valued RKHS theory.

Stage 1: Regressing the Conditional Mean Operator As first introduced by Grünewälder et al. [2012] and generalized to infinite dimensional spaces by Singh et al. [2019], estimating $C_{X|Y}^\top$ is equivalent to solving a vector-valued kernel ridge regression problem in the hypothesis space of Hilbert-Schmidt operators from $\mathcal{H}_k$ to $\mathcal{H}_\ell$, denoted as $\mathrm{HS}(\mathcal{H}_k, \mathcal{H}_\ell)$. Specifically, we may consider the operator-valued kernel defined over $\mathcal{H}_k$ as $\Gamma(f, f') := \langle f, f\rangle_k \,\mathrm{Id}_{\mathcal{H}_\ell}$. We denote $\mathcal{H}_\Gamma$ the $\mathcal{H}_\ell$-valued RKHS spanned by Γ with norm $|\cdot|_\Gamma$, which can be identified to $\mathrm{HS}(\mathcal{H}_k, \mathcal{H}_\ell)$. Singh et al. [2019] frame CMO regression as the minimisation surrogate discrepancy $\mathcal{E}_{\mathrm{c}}(C) := \mathbb{E}\left[|k_X - C^\top \ell_Y|_k^2\right]$, to which they substitute an empirical regularised version restricted to $\mathcal{H}_\Gamma$ given by $\hat{\mathcal{E}}_{\mathrm{c}}(C) := \frac{1}{N}\sum_{i=1}^N |k_{x_i} - C^\top \ell_{y_i}|_k^2 + \lambda |C|_\Gamma^2$. This $\mathcal{H}_k$-valued kernel ridge regression problem admits a closed-form minimiser which shares the same empirical form as the CMO, i.e. $\hat{C}_{X|Y}^\top = \arg\min_{C\in\mathcal{H}_\Gamma} \hat{\mathcal{E}}_{\mathrm{c}}(C)$ [Grünewälder et al., 2012, Singh et al., 2019].

Stage 2 : Regressing the Deconditional Mean Operator The DMO on the other hand is defined as the operator $D_{X|Y} : \mathcal{H}_k \to \mathcal{H}_\ell$ such that $\forall f \in \mathcal{H}_k$, $D_{X|Y}^\top C_{X|Y}^\top f = f$. Since deconditioning corresponds to finding a pseudo-inverse to the CMO, it is natural to consider a reconstruction objective $\mathcal{E}_{\mathrm{d}}(D) := \mathbb{E}\left[|\ell_Y - D C_{X|Y} \ell_Y|_\ell^2\right]$. Introducing a novel characterization of the DMO, we propose to minimise this objective in the hypothesis space of Hilbert-Schmidt operators $\mathrm{HS}(\mathcal{H}_k, \mathcal{H}_\ell)$ which identifies to $\mathcal{H}_\Gamma$. As per above, we denote $\hat{C}_{X|Y}$ the empirical CMO learnt in Stage 1, and we substitute the loss with an empirical regularised formulation on $\mathcal{H}_\Gamma$

$$\hat{\mathcal{E}}_{\mathrm{d}}(D) := \frac{1}{M}\sum_{j=1}^M |\ell_{\tilde{y}_j} - D\hat{C}_{X|Y}\ell_{\tilde{y}_j}|_\ell^2 + \epsilon |D|_\Gamma^2 \qquad D \in \mathcal{H}_\Gamma \qquad \epsilon > 0 \tag{3.10}$$

Proposition 3.4.1 (Empirical DMO as vector-valued regressor)**.** *The minimiser of the empirical reconstruction risk is the empirical DMO, i.e.* $\hat{D}_{X|Y} = \arg\min_{D \in \mathcal{H}_\Gamma} \hat{\mathcal{E}}_{\mathrm{d}}(D)$

Since it requires estimating the CMO first, minimising (3.10) can be viewed as a two-staged vector value regression problem.

Convergence results Following the footsteps of Szabó et al. [2016] and Singh et al. [2019], this perspective enables us to state the performance of the DMO estimate in terms of asymptotic convergence of the objective $\mathcal{E}_{\mathrm{d}}$. As in Caponnetto and De Vito [2007], we must restrict the class of probability measure for $\mathbb{P}_{XY}$ and $\mathbb{P}_Y$ to ensure uniform convergence even when $\mathcal{H}_k$ is infinite-dimensional. The family of distribution considered is a general class of priors that do not assume parametric distributions and is parametrized by two variables: $b > 1$ controls the effective input dimension and $c \in]1, 2]$ controls functional smoothness. Mild regularity assumptions are also placed on the original spaces $\mathcal{X}, \mathcal{Y}$, their corresponding RKHS $\mathcal{H}_k, \mathcal{H}_\ell$ and the vector-valued RKHS $\mathcal{H}_\Gamma$. We discuss these assumptions in detail in Appendix A.5. Importantly, while $\mathcal{H}_k$ can be infinite-dimensional, we nonetheless have to assume the RKHS $\mathcal{H}_\ell$ is finite-dimensional. This is to ensure during the derivation of the bounds, the trace of some operators in $\mathcal{H}_\ell$ does not blow up to infinity. In further research, we hope to relax this assumption.

Theorem 3.4.2 (Empirical DMO Convergence Rate)**.** *Denote* $D_{\mathbb{P}_Y} = \arg\min_{D \in \mathcal{H}_\Gamma} \mathcal{E}_{\mathrm{d}}(D)$*. Assume assumptions stated in Appendix A.5 are satisfied. In particular, let* (b, c) *and* $(0, c')$ *be the parameters of the restricted class of distribution for* $\mathbb{P}_Y$ *and* $\mathbb{P}_{XY}$ *respectively and let* $\iota \in]0, 1]$ *be the Hölder continuity exponent in* $\mathcal{H}_\Gamma$*. Then, if we choose* $\lambda = N^{-\frac{1}{c'+1}}$, $N = M^{\frac{a(c'+1)}{\iota(c'-1)}}$ *where* $a > 0$*, we have the following result,*

- *If* $a \leq \frac{b(c+1)}{bc+1}$*, then* $\mathcal{E}_{\mathrm{d}}(\hat{D}_{X|Y}) - \mathcal{E}_{\mathrm{d}}(D_{\mathbb{P}_Y}) = \mathcal{O}(M^{\frac{-ac}{c+1}})$ *with* $\epsilon = M^{\frac{-a}{c+1}}$
- *If* $a \geq \frac{b(c+1)}{bc+1}$*, then* $\mathcal{E}_{\mathrm{d}}(\hat{D}_{X|Y}) - \mathcal{E}_{\mathrm{d}}(D_{\mathbb{P}_Y}) = \mathcal{O}(M^{\frac{-bc}{bc+1}})$ *with* $\epsilon = M^{\frac{-b}{bc+1}}$

This theorem underlines a trade-off between the computational and statistical efficiency with respect to the datasets cardinalities $N = |\mathcal{D}_1|$, $M = |\mathcal{D}_2|$ and the problem difficulty b, c, c'. For $a \leq \frac{b(c+1)}{bc+1}$, smaller a means less samples from $\mathcal{D}_1$ at fixed M and thus computational savings. But it also hampers convergence, resulting in reduced statistical efficiency. At $a = \frac{b(c+1)}{bc+1} < 2$, the convergence rate is a minimax computational-statistical efficiency optimal, i.e. convergence rate is optimal with smallest possible M. We note that at this optimal, $N > M$ and which means fewer samples are required from $\mathcal{D}_2$. $a \geq \frac{b(c+1)}{bc+1}$ does not improve the convergence rate but only increases the size of $\mathcal{D}_1$ and hence the computational cost it bears.

3.5 Deconditional Downscaling Experiments

We demonstrate and evaluate our CMP-based downscaling approaches on both synthetic experiments and a challenging atmospheric temperature field downscaling problem with unmatched multi-resolution data. We denote the exact CMP deconditional posterior as CMP, the CMP using with efficient shrinkage CMO estimation as S-CMP and the variational formulation as VARCMP. They are compared against VBAGG [Law et al., 2018c] — which we describe below — and a GP regression [Rasmussen and Williams, 2005b] baseline (GPR) modified to take bags centroids as the input. Experiments are implemented in *PyTorch* [Paszke et al., 2019, Gardner et al., 2018], all code and datasets are made available here and computational details are provided in Appendix A.6.

Variational Aggregate Learning VBAGG is introduced by Law et al. [2018c] as a variational aggregate learning framework to disaggregate exponential family models, with emphasis on the Poisson family. We consider its application to the Gaussian family, which models the relationship between aggregate targets z_j and bag covariates $\{x_j^{(i)}\}_i$ by bag-wise averaging of a GP prior on the function of interest. In fact, the Gaussian VBAGG corresponds exactly to a special case of CMP on matched data, where the bag covariates are simply one hot encoded indices with kernel $\ell(j, j') = \delta(j, j')$ where δ is the Kronecker delta. However, VBAGG cannot handle unmatched data as bag indices do not instill the smoothness that is used for mediation. For fair analysis, we compare variational methods VARCMP and VBAGG together, and exact methods CMP/S-CMP to an exact version of VBAGG, which we implement and refer to as BAGG-GP.

3.5.1 Swiss Roll

The *scikit-learn* [Pedregosa et al., 2011] swiss roll manifold sampling function allows to generate a 3D manifold of points $x \in \mathbb{R}^3$ mapped with their position along the manifold $t \in \mathbb{R}$. Our objective will be to recover t for each point x by only observing t at an aggregate level. In the first experiment, we compare our model to existing weakly supervised spatial disaggregation methods when all high-resolution covariates are matched with a coarse-level aggregate target. We then proceed to withdraw this requirement in a companion experiment.

3.5.1.1 Direct matching

Experimental Setup Inspired by the experimental setup from Law et al. [2018c], we regularly split space along height $B - 1$ times as depicted in Figure 3.2 and group together manifold points within each height level, hence mixing together points with very different positions on the manifold. We obtain bags of samples $\{({}^b\boldsymbol{x}_j, \boldsymbol{t}_j)\}_{j=1}^B$ where the j^{th} bag contains n_j points ${}^b\boldsymbol{x}_j = \{x_j^{(i)}\}_{i=1}^{n_j}$ and their corresponding

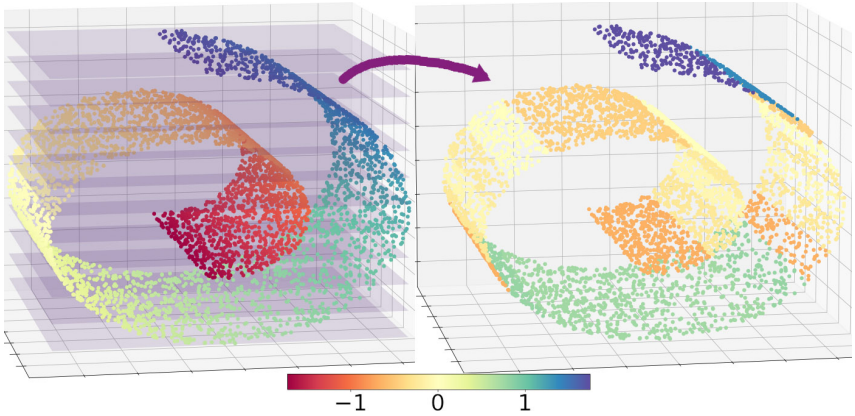


Figure 3.2: **Step 1:** Split space regularly along height. **Step 2:** Group points into height-level bags. **Step 3:** Average points targets into bag-level aggregate targets.

Table 3.1: RMSE of the swissroll experiment for models trained over directly and indirectly matched datasets ; scores averaged over 20 seeds and 1 standard deviation is reported ; * indicates our proposed methods.

Matching	CMP*	S-CMP*	VARCMP*	BAGG-GP	VBAGG	GPR
Direct	$0.33_{\pm 0.06}$	$0.25_{\pm 0.04}$	$0.18_{\pm 0.04}$	$0.60_{\pm 0.01}$	$0.22_{\pm 0.04}$	$0.70_{\pm 0.05}$
Indirect	$0.80_{\pm 0.14}$	$1.05_{\pm 0.04}$	$0.87_{\pm 0.07}$	$1.13_{\pm 0.11}$	$1.46_{\pm 0.34}$	$1.04_{\pm 0.05}$

targets $\boldsymbol{t}_j = \{t_j^{(i)}\}_{i=1}^{n_j}$. We then construct bag aggregate targets by taking noisy bag targets average $z_j := \frac{1}{n_j}\sum_{i=1}^{n_j} t_j^{(i)} + \varepsilon_j$, where $\varepsilon_j \sim \mathcal{N}(0, \sigma^2)$. We thus obtain matched weakly supervised bag learning dataset $\mathcal{D}^\circ = \{({}^b\boldsymbol{x}_j, z_j)\}_{j=1}^B$. Since each bag corresponds to a height-level, the center altitude of each height split $y_j \in \mathbb{R}$ is a natural candidate bag-level covariate that informs on relative positions of the bags. We can augment the above dataset as $\mathcal{D} = \{({}^b\boldsymbol{x}_j, y_j, z_j)\}_{j=1}^B$. Using these bag datasets, we now wish to downscale aggregate targets z_j to recover the unobserved manifold locations $\{\boldsymbol{t}_j\}_{j=1}^B$ and be able to query the target at any previously unseen input x.

Models We use a zero-mean prior on f and choose a Gaussian kernel for k and ℓ. Inducing points location is initialized with K-means++ procedure for VARCMP and VBAGG such that they spread evenly across the manifold. For exact methods, kernel hyperparameters and noise variance σ^2 are learnt on $\mathcal{D}$ by optimising the marginal likelihood. For VARCMP, they are learnt jointly with variational distribution parameters by maximising an evidence lower bound objective. While CMP-based methods can leverage bag-level covariates y_j, baselines are restricted to learn from $\mathcal{D}^\circ$. Adam optimiser [Kingma and Ba, 2015] is used in all experiments.

Results We test models against unobserved groundtruth $\{\boldsymbol{t}_j\}_{j=1}^B$ by evaluating the root mean square error (RMSE) to the posterior mean. Table 3.1 shows that CMP, S-CMP and VARCMP outperform

their corresponding counterparts i.e. BAGG-GP and VBAGG, with statistical significance confirmed by a Wilcoxon signed-rank test in Appendix A.6. Most notably, this shows that the additional knowledge on bag-level dependence instilled by ℓ is reflected even in a setting where each bag is matched with an aggregate target.

3.5.1.2 Indirect matching

Experimental Setup We now impose indirect matching through mediating variable y_j. We randomly select $N = \lfloor \frac{B}{2} \rfloor$ bags which we consider to be the N first ones without loss of generality and split $\mathcal{D}$ into $\mathcal{D}_1 = \{({}^b\boldsymbol{x}_j, y_j)\}_{j=1}^N$ and $\mathcal{D}_2 = \{(\tilde{y}_j, \tilde{z}_j)\}_{j=1}^{B-N} = \{(y_{N+j}, z_{N+j})\}_{j=1}^{B-N}$, such that no pair of covariates bag ${}^b\boldsymbol{x}_j$ and aggregate target $\tilde{z}_j$ are jointly observed.

Models CMP-based methods are naturally able to learn from this setting and are trained by independently drawing samples from $\mathcal{D}_1$ and $\mathcal{D}_2$. Baseline methods however require bags of covariates to be matched with an aggregate bag target. To remedy this, we place a separate prior $g \sim \mathcal{GP}(0, \ell)$ and fit regression model $\tilde{z}_j = g(\tilde{y}_j) + \varepsilon_j$ over $\mathcal{D}_2$. We then use the predictive posterior mean to augment the first dataset as $\mathcal{D}_1' = \left\{ \left({}^b\boldsymbol{x}_j, \mathbb{E}[g(y_j)|\mathcal{D}_2] \right) \right\}_{j=1}^N$. This dataset can then be used to train BAGG-GP, VBAGG and GPR.

Results For comparison, we use the same evaluation as in the direct matching experiment. Table 3.1 underlines an anticipated drop in RMSE for all models, but we observe that BAGG-GP and VBAGG suffer most from the mediated matching of the dataset while CMP and VARCMP report best scores by a substantial margin. This highlights how using a separate prior on g to mediate $\mathcal{D}_1$ and $\mathcal{D}_2$ turns out to be suboptimal in contrast to using the prior naturally implied by CMP. While it is more computationally efficient than CMP, we observe a relative drop in performance for S-CMP.

3.5.2 Mediated downscaling of atmospheric temperature

Given the large diversity of sources and formats of remote sensing and model data, expecting directly matched observations is often unrealistic [Watson-Parris et al., 2016]. For example, two distinct satellite products will often provide low and high resolution imagery that can be matched neither spatially nor temporally [Remer et al., 2005, Platnick et al., 2003, Stephens et al., 2002, Barnes et al., 1998]. Climate simulations [Flato, 2011, Scholze et al., 2012, Eyring et al., 2016] on the other hand provide a comprehensive coarse resolution coverage of meteorological variables that can serve as a mediating dataset.

For the purpose of demonstration, we create an experimental setup inspired by this problem using Coupled Model Intercomparison Project Phase 6 (CMIP6) [Eyring et al., 2016] simulation data. This grants us

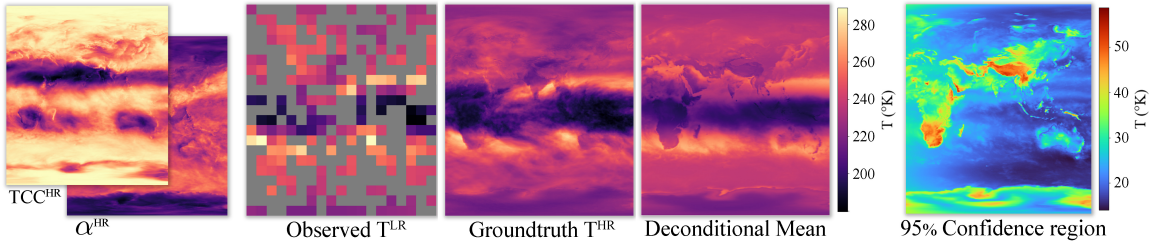


Figure 3.3: **Left:** High-resolution atmoshperic covariates used for prediction; **Center-Left:** Observed low-resolution temperature field, grey pixels are unobserved; **Center** Unobserved high-resolution groundtruth temperature field; **Center-Right:** VARCMP deconditional posterior mean; **Right** 95% confidence region size on prediction; temperature values are in Kelvin.

access to groundtruth high-resolution covariates to facilitate model evaluation.

Experimental Setup We collect monthly mean 2D atmospheric fields simulation from CMIP6 data [Roberts, 2018 v20180730, Voldoire, 2019 v20190221] for the following variables: air temperature at cloud top (T), mean cloud top pressure (P), total cloud cover (TCC) and cloud albedo (α). First, we collocate TCC and α onto an HR latitude-longitude grid of size 360×720 to obtain fine-grained fields (latitude$^{\text{HR}}$, longitude$^{\text{HR}}$, altitude$^{\text{HR}}$, TCC$^{\text{HR}}$, α^{HR}) augmented with a static HR surface altitude field. Then we collocate P and T onto an LR grid of size 21×42 to obtain coarse-grained fields (latitude$^{\text{LR}}$, longitude$^{\text{LR}}$, P$^{\text{LR}}$, T$^{\text{LR}}$). We denote by B the number of low-resolution pixels.

Our objective is to disaggregate T$^{\text{LR}}$ to the HR fields granularity. We assimilate the j^{th} coarse temperature pixel to an aggregate target $z_j := \text{T}_j^{\text{LR}}$ corresponding to bag j. Each bag includes HR covariates ${}^b\boldsymbol{x}_j = \{x_j^{(i)}\}_{i=1}^{n_j} := \{(\text{latitude}_j^{\text{HR}(i)}, \text{longitude}_j^{\text{HR}(i)}, \text{altitude}_j^{\text{HR}(i)}, \text{TCC}_j^{\text{HR}(i)}, \alpha_j^{\text{HR}(i)})\}_{i=1}^{n_j}$. To emulate unmatched observations, we randomly select $N = \lfloor \frac{B}{2} \rfloor$ of the bags $\{{}^b\boldsymbol{x}_j\}_{j=1}^{N}$ and keep the opposite half of LR observations $\{z_{N+j}\}_{j=1}^{B-N}$, such that there is no single aggregate bag target that corresponds to one of the available bags. Finally, we choose the pressure field P$^{\text{LR}}$ as the mediating variable. We hence compose a third party low resolution field of bag-level covariates $y_j := (\text{latitude}_j^{\text{LR}}, \text{longitude}_j^{\text{LR}}, \text{P}_j^{\text{LR}})$ which can separately be matched with both above sets to obtain datasets $\mathcal{D}_1 = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{N}$ and $\mathcal{D}_2 = \{(\tilde{y}_j, \tilde{z}_j)\}_{j=1}^{B-N} = \{(y_{N+j}, z_{N+j})\}_{j=1}^{B-N}$.

Models Setup We only consider variational methods to scale to a large number of pixels. VARCMP is naturally able to learn from indirectly matched data. We use a Matérn-1.5 kernel for rough spatial covariates (latitude, longitude) and a Gaussian kernel for atmospheric covariates (P, TCC, α) and surface altitude. k and ℓ are both taken as sums of Matérn and Gaussian kernels, and their hyperparameters are learnt along with noise variance during training. A high-resolution noise term is also introduced, with details provided in Appendix A.6. Inducing points locations are uniformly initialized across the HR grid.

Table 3.2: Downscaling similarity scores of posterior mean against groundtruth high resolution cloud top temperature field ; averaged over 10 seeds; we report 1 standard deviation ; "↓": lower is better ; "↑": higher is better.

Model	RMSE ↓	MAE ↓	Corr. ↑	SSIM ↑
VARGPR	8.02±0.28	5.55±0.17	0.831±0.012	0.212±0.011
VBAGG	8.25±0.15	5.82±0.11	0.821±0.006	0.182±0.004
VARCMP	7.40±0.25	5.34±0.22	0.848±0.011	0.212±0.013

We replace GPR with an inducing point variational counterpart VARGPR [Titsias, 2009]. Since baseline methods require a matched dataset, we proceed as with the unmatched swiss roll experiment and fit a GP regression model g with kernel ℓ on $\mathcal{D}_2$ and then use its predictive posterior mean to obtain pseudo-targets for the bags of HR covariates from $\mathcal{D}_1$.

Results Performance is evaluated by comparing downscaling deconditional posterior mean against original high-resolution field T^{HR} available in CMIP6 data [Voldoire, 2019 v20190221], which we emphasise is never observed. We use random Fourier features [Rahimi et al., 2007] approximation of kernel k to scale kernel evaluation to the HR covariates grid during testing. As reported in Table 3.2, VARCMP substantially outperforms both baselines with statistical significance provided in Appendix A.6. Figure 3.3 shows the reconstructed image with VARCMP, plots for other methods are included in the Appendix A.6. The model resolves statistical patterns from HR covariates into coarse resolution temperature pixels, henceforth reconstructing a faithful HR version of the temperature field.

3.6 Discussion

We introduced a scalable Bayesian solution to the mediated statistical downscaling problem, which handles unmatched multi-resolution data. The proposed approach combines Gaussian Processes with the framework of deconditioning using RKHSs and recovers previous approaches as its special cases. We provided convergence rates for the associated deconditioning operator. Finally, we demonstrated the advantages over spatial disaggregation baselines in synthetic data and in a challenging atmospheric field downscaling problem.

In future work, exploring theoretical guarantees of the computationally efficient shrinkage formulation in a multi-resolution setting and relaxing finite dimensionality assumptions for the convergence rate will have fruitful practical and theoretical implications. Further directions also open up to quantify uncertainty over the deconditional posterior since it is computed using empirical estimates of the CMP covariance. This may be problematic if the mediating variable undergoes covariate shift between the two datasets.

4 | Uncertainty Quantification for Causal Data Fusion

This chapter is based on the following paper:

> **Siu Lun Chau***, Jean-Francois Ton*, Javier Gonzalez, Yee Whye Teh, and Dino Sejdinovic. "BayesIMP: Uncertainty Quantification for Causal Data Fusion." Advances in Neural Information Processing Systems (NeurIPS), 2021

Abstract

While causal models are becoming one of the mainstays of machine learning, the problem of uncertainty quantification in causal inference remains challenging. In this chapter, we study the causal data fusion problem, where datasets pertaining to multiple causal graphs are combined to estimate the average treatment effect of a target variable. As data arises from multiple sources and can vary in quality and quantity, principled uncertainty quantification becomes essential. To that end, we introduce Bayesian Interventional Mean Processes, a framework that combines ideas from probabilistic integration and kernel mean embeddings to represent interventional distributions in the reproducing kernel Hilbert space while taking into account the uncertainty within each causal graph. To demonstrate the utility of our uncertainty estimation, we apply our method to the Causal Bayesian Optimisation task and show improvements over state-of-the-art methods.

4.1 Introduction

Causal inference has seen a significant surge of research interest in areas such as healthcare [Thompson, 2019], ecology [Courtney et al., 2017], and optimisation [Aglietti et al., 2020a]. However, data fusion, the problem of merging information from multiple data sources, has received limited attention in the context of causal modelling, yet presents significant potential benefits for practical situations [Meng et al., 2020, Singh et al., 2019]. In this work, we consider a causal data fusion problem where two causal graphs are combined for the purposes of inference of a target variable (see Fig. 4.1). In particular, our goal is to quantify the uncertainty under such a setup and determine the level of confidence in our treatment effect estimates.

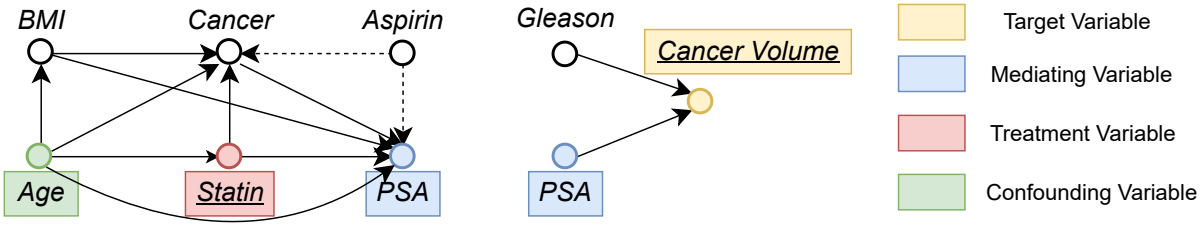


Figure 4.1: Example problem setup: Causal graphs collected in two separate medical studies i.e. [Ferro et al., 2015] and [Stamey et al., 1989]. Arrows are pointed from cause to effect variables, and dotted arrows denoted unobserved confounding effect. (Left) $\mathcal{D}_1$: Data describing the causal relationships between statin level and Prostate Specific Antigen (PSA). (Right) $\mathcal{D}_2$: Data from a prostate cancer study for patients about to receive a radical prostatectomy. Goal: **Model** $\mathbb{E}[\textbf{\textit{Cancer Volume}}|\textbf{\textit{do(Statin)}}]$ while also quantifying its uncertainty.

Let us consider the motivating example in Fig. 4.1, where a medical practitioner is investigating how *prostate cancer volume* is affected by a *statin* drug dosage. We consider the case where the doctor only has access to two separate medical studies describing the quantities of interest. On one hand, we have observational data, from one medical study $\mathcal{D}_1$ [Thompson, 2019], describing the causal relationship between *statin* level and *prostate specific antigen (PSA)*, and on the other hand, we have observational data, from a second study $\mathcal{D}_2$ [Stamey et al., 1989], that looked into the link between *PSA* level and *prostate cancer volume*. The goal is to model the **interventional effect** between our target variable (*cancer volume*) and the treatment variable (*statin*). This problem setting is different from the standard observational scenario as it comes with the following challenges:

- **Unmatched data:** Our goal is to estimate $\mathbb{E}[cancer\ volume|do(statin)]$ but the observed *cancer volume* is not paired with *statin* dosage. Instead, they are related via a mediating variable *PSA*.
- **Uncertainty quantification:** The two studies may be of different data quantity/quality. Furthermore, a covariate shift in the mediating variable, i.e. a difference between its distributions in two datasets,

may cause inaccurate extrapolation. Hence, we need to account for uncertainty in both datasets.

Formally, let X be the treatment (*Statin*), Y be the mediating variable (*PSA*) and T our target (*cancer volume*), and our aim is to estimate $\mathbb{E}[T|do(X)]$. The problem of unmatched data in a similar context has been previously considered by [Singh et al., 2019] using a two-staged regression approach ($X \to Y$ and $Y \to T$). However, uncertainty quantification, despite being essential if our estimates of interventional effects will guide decision-making, has not been previously explored. In particular, it is crucial to quantify the uncertainty in both stages as this takes into account the lack of data in specific parts of the space. Given that we are using different datasets for each stage, there are also two sources of epistemic uncertainties (due to lack of data) as well as two sources of aleatoric uncertainties (due to inherent randomness in Y and T) [Hüllermeier and Waegeman, 2021] . It is thus natural to consider regression models based on Gaussian Processes (GP) [Rasmussen and Williams, 2005a], as they are able to model both types of uncertainties. However, as GPs, or any other standard regression models, are designed to model conditional expectations only and will fail to capture the underlying distributions of interest (e.g. if there is multimodality in Y as discussed in Ton et al. [2021a]). This is undesirable since, as we will see, interventional effect estimation requires accurate estimates of distributions. While one could in principle resort to density estimation methods, this becomes challenging since we typically deal with a number of conditional/ interventional densities.

In this chapter, we introduce the framework of *Bayesian Interventional Mean Processes* (BAYESIMP) to circumvent the challenges in the causal data fusion setting described above. BAYESIMP considers kernel mean embeddings [Muandet et al., 2017] for representing distributions in a reproducing kernel Hilbert space (RKHS), in which the whole arsenal of kernel methods can be extended to probabilistic inference (e.g. kernel Bayes rule [Fukumizu et al., 2010], hypothesis testing [Zhang et al., 2018], distribution regression [Law et al., 2018b]). Specifically, BAYESIMP uses kernel mean embeddings to represent the interventional distributions and to analytically marginalise out Y, hence accounting for aleatoric uncertainties. Further, BAYESIMP uses GPs to estimate the required kernel mean embeddings from data in a Bayesian manner, which allows to quantify the epistemic uncertainties when representing the interventional distributions. To illustrate the quality of our uncertainty estimates, we apply BAYESIMP to Causal Bayesian Optimisation [Aglietti et al., 2020b], an efficient heuristic to optimise objective functions of the form $x^* = \arg\min_{x \in \mathcal{X}} \mathbb{E}[T|do(X) = x]$. Our contributions are summarised below:

1. We propose a novel *Bayesian Learning of Conditional Mean Embedding* (BAYESCME) that allows us to estimate conditional mean embeddings in a Bayesian framework.

2. Using BAYESCME, we propose a novel *Bayesian Interventional Mean Process* (BAYESIMP) that

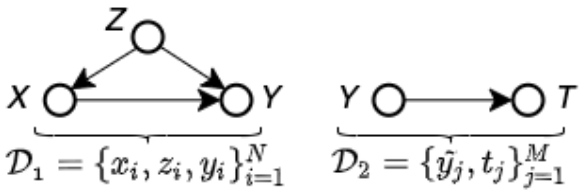


Figure 4.2: A general two stage causal learning setup.

allows us to model interventional effect across causal graphs without explicit density estimation, while obtaining uncertainty estimates for $\mathbb{E}[T|do(X) = x]$.

3. We apply BAYESIMP to Causal Bayesian Optimisation, a problem introduced in Aglietti et al. [2020b] and show significant improvements over existing state-of-the-art methods.

Note that Bareinboim and Pearl [2016] also considered a causal fusion problem but with a different objective. They focused on extrapolating experimental findings across treatment domains, i.e. inferring $\mathbb{E}[Y \mid do(X)]$ when only data from $p(Y \mid do(S))$ is observed, where S is some other treatment variable. In contrast, we focus on modelling combined causal graphs, with a strong emphasis on uncertainty quantification. While Singh et al. [2020] considered mapping interventional distributions in the RKHS to model quantities such as $\mathbb{E}[T \mid do(X)]$, they only considered a frequentist approach, which does not account for epistemic uncertainties.

Notations. We denote X, Y, Z as random variables taking values in the non-empty sets $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z}$ respectively. Let $k_x : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be positive definite kernels on X with an associated RKHS $\mathcal{H}_{k_x}$. The corresponding canonical feature map $k_x(x', \cdot)$ is denoted as $\phi_x(x')$. Analogously for Y and Z.

In the simplest setting, we observe i.i.d samples $\mathcal{D}_1 = \{x_i, y_i, z_i\}_{i=1}^N$ from joint distribution $\mathbb{P}_{XYZ}$ which we concatenate into vectors $\mathbf{x} := [x_1, ..., x_N]^\top$. Similarly for $\mathbf{y}$ and $\mathbf{z}$. For this work, X is referred as *treatment variable*, Y as *mediating variable* and Z as *adjustment variables* accounting for confounding effects. With an abuse of notation[1], features matrices are defined by stacking feature maps along the columns, i.e $\Phi_\mathbf{x} := [\phi_x(x_1), ..., \phi_x(x_N)]$. We denote the Gram matrix as $K_{\mathbf{xx}} := \Phi_\mathbf{x}^\top \Phi_\mathbf{x}$ and the vector of evaluations $k_{x\mathbf{x}}$ as $[k_x(x, x_1), ..., k_x(x, x_N)]$. We define $\Phi_\mathbf{y}, \Phi_\mathbf{z}$ analogously for $\mathbf{y}$ and $\mathbf{z}$.

Lastly, we denote $T = f(Y) + \epsilon$ as our *target variable*, which is modelled as some noisy evaluation of a function $f : \mathcal{Y} \to \mathcal{T}$ on Y while ϵ being some random noise. For our problem setup we observe a second dataset of i.i.d realisations $\mathcal{D}_2 = \{\tilde{y}_j, t_j\}_{j=1}^M$ from the joint $\mathbb{P}_{YT}$ independent of $\mathcal{D}_1$. Again, we define $\tilde{\mathbf{y}} := [_1, ..., _M]^\top$ and $\mathbf{t} := [t_1, ..., t_M]^\top$ just like for $\mathcal{D}_1$. See Fig. 4.2 for illustration.

[1] because we are stacking infinite dimensinoal vectors and representing them as matrices.

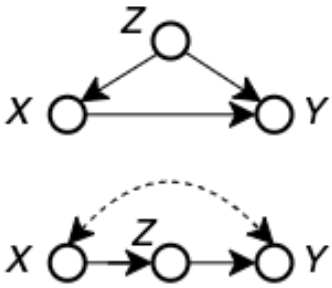


Figure 4.3: (Top) Backdoor adjustment (Bottom) Front-door adjustment, dashed edges denote unobserved confounders.

4.2 Background

Representing interventional distributions in an RKHS[2] has been explored in different contexts [Muandet et al., 2018, Singh et al., 2020, Mitrovic et al., 2018]. In particular, when the treatment is continuous, Singh et al. [2020] introduced the *Interventional Mean Embeddings* (IMEs) to model densities in an RKHS by utilising smoothness across treatments. Given that IME is an important building block to our contribution, we give it a detailed review by first introducing the key concepts of *do*-calculus [Pearl, 1995] and conditional mean embeddings [Song et al., 2013].

4.2.1 Interventional distribution and *do*-calculus

In this work, we consider the structural causal model [Pearl, 1995] (SCM) framework, where a causal directed acyclic graph (DAG) $\mathcal{G}$ is given and encodes knowledge of existing causal mechanisms amongst the variables in terms of conditional independencies. Given random variables X and Y, a central question in interventional inference [Pearl, 1995] is to estimate the distribution $p(Y|do(X) = x)$, where $\{do(X) = x\}$ represents an intervention on X whose value is set to x. Note that this quantity is not directly observed given that we are usually only given observational data, i.e, data sampled from the conditional $p(Y|X)$ but not from the interventional density $p(Y|do(X))$. However, Pearl developed *do*-calculus which allows us to estimate interventional distributions from purely observational distributions under the identifiability assumption. Here we present the backdoor and front-door adjustments, which are the fundamental components of DAG-based causal inference.

The backdoor adjustment is applicable when there are observed confounding variables Z between the cause X and the effect Y (see Fig. 4.3 (Top)). In order to correct for this confounding bias we can use the following equation, adjusting for Z as $p(Y|do(X) = x) = \int_{\mathcal{Z}} p(Y|X = x, z)p(z)dz$.

The front-door adjustment applies to cases when confounders are unobserved (see Fig. 4.3 (Bottom)).

[2]We refer the reader to a detailed review of RKHS methods provided in the Appendix of Singh et al. [2019]

Given a set of front-door adjustment variables Z, we can again correct the estimate for the causal effect from X to Y with $p(Y|do(X) = x) = \int_{\mathcal{Z}} \int_{\mathcal{X}} p(Y|x', z)p(z|X = x)p(x')dx'dz$.

We rewrite the above formulae in a more general form as we show below.

$$p(Y|do(X) = x) = \mathbb{E}_{\Omega_x}[p(Y|\Omega_x)] = \int p(Y|\Omega_x)p(\Omega_x)d\Omega_x \tag{4.1}$$

For backdoor we have $\Omega_x = \{X = x, Z\}$ and $p(\Omega_x) = \delta_x p(Z)$ where δ_x is the Dirac measure at $X = x$. For front-door, $\Omega_x = \{X', Z\}$ and $p(\Omega_x) = p(X')p(Z|X = x)$.

4.2.2 Conditional Mean Embeddings

Kernel mean embeddings of distributions provide a powerful framework for representing probability distributions [Muandet et al., 2017, Song et al., 2013] in an RKHS. In particular, we work with conditional mean embeddings (CMEs) in this chapter. Given random variables X, Y with joint distribution $\mathbb{P}_{XY}$, the conditional mean embedding with respect to the conditional density $p(Y|X = x)$, is defined as:

$$\mu_{Y|X=x} := \mathbb{E}_{Y|X=x}[\phi_y(Y)] = \int_{\mathcal{Y}} \phi_y(y)p(y|X = x)dy \tag{4.2}$$

CMEs allow us to represent the distribution $p(Y|X = x)$ as an element $\mu_{Y|X=x}$ in the RKHS $\mathcal{H}_{k_y}$ without having to model the densities explicitly. Following Song et al. [2013], CMEs can be associated with a Hilbert-Schmidt operator $\mathcal{C}_{Y|X} : \mathcal{H}_{k_x} \to \mathcal{H}_{k_y}$, known as the conditional mean embedding operator, which satisfies $\mu_{Y|X=x} = \mathcal{C}_{Y|X}\phi_x(x)$ where $\mathcal{C}_{Y|X} := \mathcal{C}_{YX}\mathcal{C}_{XX}^{-1}$ with $\mathcal{C}_{YX} := \mathbb{E}_{Y,X}[\phi_y(Y) \otimes \phi_x(X)]$ and $\mathcal{C}_{XX} := \mathbb{E}_{X,X}[\phi_x(X) \otimes \phi_x(X)]$ being the covariance operators. As a result, the finite sample estimator of $\mathcal{C}_{Y|X}$ based on the dataset $\{\mathbf{x}, \mathbf{y}\}$ can be written as:

$$\hat{\mathcal{C}}_{Y|X} = \Phi_{\mathbf{y}}(K_{\mathbf{xx}} + \lambda I)^{-1}\Phi_{\mathbf{x}}^T \tag{4.3}$$

where $\lambda > 0$ is a regularization parameter. Note that from Eq. 4.3, Grünewälder et al. [2012] showed that the CME can be interpreted as a vector-valued kernel ridge regressor (V-KRR) i.e. $\phi_x(x)$ is regressed to an element in $\mathcal{H}_{k_y}$. This is crucial as CMEs allow us to turn the integration, in Eq. 4.2, into a regression task and hence remove the need for explicit density estimation. This insight is important as it allows us to derive analytic forms for our algorithms. Furthermore, the regression formalism of CMEs motivated us to derive a Bayesian version of CME using vector-valued Gaussian Processes (V-GP), see Sec.4.3.

4.2.3 Interventional Mean Embeddings

Interventional Mean Embeddings (IME) [Singh et al., 2020] combine the above ideas to represent interventional distributions in RKHSs. We derive the front-door adjustment embedding here, but the backdoor adjustment follows analogously. Denote $\mu_{Y|do(X)=x}$ as the IME corresponding to the interventional distribution $p(Y|do(X)=x)$, which can be written as:

$$\mu_{Y|do(X)=x} := \int_{\mathcal{Y}} \phi_y(y) p(y|do(X)=x)dy = \int_{\mathcal{X}}\int_{\mathcal{Z}} \underbrace{\Big(\int_{\mathcal{Y}} \phi_y(y)p(y|x',z)dy\Big)}_{\text{CME } \mu_{Y|X=x,Z=z}} p(z|x)p(x')dzdx'$$

using the front-door formula with adjustment variable Z, and rearranging the integrals. By definition of CME $\int \phi_y(y)p(y|x',z)dy = C_{Y|X,Z}(\phi_x(x') \otimes \phi_z(z))$ and linearity of integration, we have

$$= C_{Y|X,Z}\Big(\underbrace{\int_{\mathcal{X}} \phi_x(x')p(x')dx'}_{=\mu_X} \otimes \underbrace{\int_{\mathcal{Z}} \phi_z(z)p(z|x)dz}_{=\mu_{Z|X=x}}\Big) = C_{Y|X,Z}\big(\mu_X \otimes \mu_{Z|X=x}\big)$$

Using notations from Sec.4.2.1, embedding interventional distributions into an RKHS is as follows.

Proposition 4.2.1. *Given an identifiable do-density of the form* $p(Y|do(X)=x) = \mathbb{E}_{\Omega_x}[p(Y|\Omega_x)]$, *the general form of the empirical interventional mean embedding is given by,*

$$\hat{\mu}_{Y|do(X)=x} = \Phi_Y(K_{\Omega_x} + \lambda I)^{-1}\Phi_{\Omega_x}(x)^\top \tag{4.4}$$

where $K_{\Omega_x} = K_{XX} \odot K_{ZZ}$ *and* $\Phi_{\Omega_x}(x)$ *is derived depending on* $p(\Omega_x)$. *In particular, for backdoor adjustments,* $\Phi^{(bd)}_{\Omega_x}(x) = \Phi_X^\top k_X(x,\cdot) \odot \Phi_Z^\top \hat{\mu}_z$ *and for front-door* $\Phi^{(fd)}_{\Omega_x}(x) = \Phi_X^\top \hat{\mu}_X \odot \Phi_Z^\top \hat{\mu}_{Z|X=x}$.

4.3 Our Proposed Method

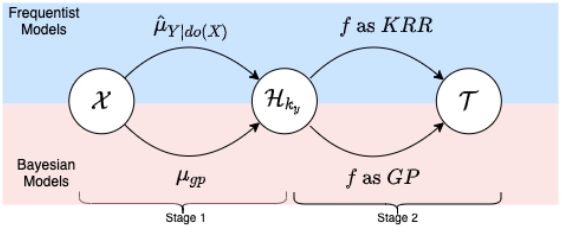


Figure 4.4: Two-stage causal learning problem

METHODS	Stage 1	Stage 2
IME [Singh et al., 2020]	KRR	KRR
IMP (Ours)	KRR	GP
BAYESIME (Ours)	GP	KRR
BAYESIMP (Ours)	GP	GP

Table 4.1: Summary of our proposed methods

Two-stage Causal Learning. Given two independent datasets $\mathcal{D}_1 = \{(x_i, z_i, y_i)\}_{i=1}^N$ and $\mathcal{D}_2 = \{(\tilde{y}_j, t_j)\}_{j=1}^M$, our goal is to model the average treatment effect in T when intervening on variable X, i.e. model $g(x) = \mathbb{E}[T|do(X) = x]$. Note that the target variable T and the treatment variable X are never jointly observed. Rather, they are linked via a mediating variable Y observed in both datasets. In our problem setting, we make the following two assumptions:

A1 The treatment only affects the target through the mediating variable, i.e $T \perp\!\!\!\perp do(X)|Y \rightarrow P(T|do(X), Y) = p(T|Y)$, in other words, that in the true data generating model P(X, Y, Z, T), all causal paths from X to T are mediated through Y.

A2 Function f given by $f(y) = \mathbb{E}[T|Y = y]$ belongs to an RKHS $\mathcal{H}_{k_y}$.[3]

We can thus express the average treatment effect as:

$$g(x) = \mathbb{E}[T|do(X) = x] = \int_{\mathcal{Y}} \underbrace{\mathbb{E}[T|do(X) = x, Y = y]}_{=\mathbb{E}[T|Y=y], \text{ since } T \perp\!\!\!\perp do(X)|Y} p(y|do(X) = x)dy \tag{4.5}$$

$$= \int_{\mathcal{Y}} f(y)p(y|do(X) = x)dy = \langle f, \mu_{Y|do(X)=x} \rangle_{\mathcal{H}_{k_y}}. \tag{4.6}$$

The final expression decomposes the problem of estimating g into that of estimating the IME $\mu_{Y|do(X)}$ (which can be done using $\mathcal{D}_1$) and that of estimating the integrand $f : \mathcal{Y} \rightarrow \mathcal{T}$ (which can be done using $\mathcal{D}_2$). Each of these two components can either be estimated using a GP or KRR approach (See Table 4.1). Furthermore, the reformulation as an RKHS inner product is crucial, as it circumvents the need for density estimation as well as the need for subsequent integration in Eq. 4.6. Rather, the main parts of the task can now be viewed as two instances of regression (recall that mean embeddings can be viewed as vector-valued regression).

To model g and quantify its uncertainty, we propose 3 GP-based approaches. While the first 2 methods, *Interventional Mean Process* (IMP) and *Bayesian Interventional Mean Embedding* (BAYESIME) are novel derivations that allow us to quantify uncertainty from either one of the datasets, we treat them as intermediate yet necessary steps to derive our main algorithm, *Bayesian Interventional Mean Process* (BAYESIMP), which allows us to quantify uncertainty from both sources in a principled way. For a summary of the methods, see Fig. 4.4 and Table 4.1. All derivations are included in the appendix.

Interventional Mean Process: Firstly, we train f as a GP using $\mathcal{D}_2$ and model $\mu_{Y|do(X)=x}$ as V-KRR using $\mathcal{D}_1$. By drawing parallels to Bayesian quadrature [Briol et al., 2019] and conditional mean process

[3] We note that this two-stage setup resembles Instrumental Variable [Muandet et al., 2019] (IV) regression. However, our general setup allows the IV and the treatment to be confounded, which is not the case in standard IV regression.

introduced in Chau et al. [2021a], the integral of interest $g(x) = \int f(y)p(y|do(X) = x)dy$ will be a GP indexed by the treatment variable X. We can then use the empirical embedding $\hat{\mu}_{Y|do(X)}$ learnt in $\mathcal{D}_1$ to obtain an analytic mean and covariance of g.

Bayesian Interventional Mean Embedding: Next, to account for the uncertainty from $\mathcal{D}_1$, we model f as a KRR and $\mu_{Y|do(X)=x}$ using a V-GP. We introduce our novel *Bayesian Learning of Conditional Mean Embeddings* (BAYESCME), which uses a *nuclear dominant kernel* [Lukić and Beder, 2001] construction, similar to Flaxman et al. [2016], to ensure that the inner product $\langle f, \mu_{Y|do(X)=x} \rangle$ is well-defined. As the embedding is a GP, the resulting inner product is also a GP and hence takes into account the uncertainty in $\mathcal{D}_1$.(See Prop. 4.3.3).

Bayesian Interventional Mean Process: Lastly, in order to account for uncertainties coming from both $\mathcal{D}_1$ and $\mathcal{D}_2$, we combine ideas from the above IMP and BAYESIME. We place GPs on both f and $\mu_{Y|do(X)}$ and use their inner product to model $\mathbb{E}[T \mid do(X)]$. Interestingly, the resulting uncertainty can be interpreted as the sum of uncertainties coming from IMP and BAYESIME with an additional interaction term (See Prop. 4.3.4).

4.3.1 Interventional Mean Process

Firstly, we consider the case where f is modelled using a GP and $\mu_{Y|do(X)=x}$ using a V-KRR. This allows us to take into account the uncertainty from $\mathcal{D}_2$ by modelling the relationship between Y and T using a GP. Drawing parallels to Bayesian quadrature [Briol et al., 2019] where integrating f with respect to a marginal measure results in a Gaussian random variable, we integrate f with respect to a conditional measure, thus resulting in a GP indexed by the conditioning variable. Note that Chau et al. [2021a] studied this GP in a non-causal setting, for a very specific downscaling problem. In this work, we extend their approach to model uncertainty in the causal setting. The resulting mean and covariance are then estimated analytically, i.e. without integrals, using the empirical IME $\hat{\mu}_{Y|do(X)}$ learnt from $\mathcal{D}_1$, see Prop. 4.3.1.

Proposition 4.3.1 (IMP). *Given dataset* $D_1 = \{(x_i, y_i, z_i)\}_{i=1}^N$ *and* $D_2 = \{(\tilde{y}_j, t_j)\}_{j=1}^M$*, if* f *is the posterior* GP *learnt from* $\mathcal{D}_2$*, then* $g = \int f(y)p(y|do(X))dy$ *is a* GP $\mathcal{GP}(m_1, \kappa_1)$ *defined on the treatment variable* X *with the following mean and covariance estimated using* $\hat{\mu}_{Y|do(X)}$ *,*

$$m_1(x) = \langle \hat{\mu}_{Y|do(x)}, m_f \rangle_{\mathcal{H}_{k_y}} = \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} K_{\mathbf{y}\tilde{\mathbf{y}}} (K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1} \mathbf{t} \tag{4.7}$$

$$\kappa_1(x, x') = \hat{\mu}_{Y|do(x)}^\top \hat{\mu}_{Y|do(x')} - \hat{\mu}_{Y|do(x)}^\top \Phi_{\tilde{\mathbf{y}}} (K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda I)^{-1} \Phi_{\tilde{\mathbf{y}}}^\top \hat{\mu}_{Y|do(x')} \tag{4.8}$$

$$= \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} \tilde{K}_{\mathbf{yy}} (K_{\Omega_x} + \lambda I)^{-1} \Phi_{\Omega_x}(x') \tag{4.9}$$

where $\hat{\mu}_{Y|do(x)} = \hat{\mu}_{Y|do(X)=x}$*,* $K_{\tilde{\mathbf{y}}\mathbf{y}} = \Phi_{\tilde{\mathbf{y}}}^\top \Phi_{\mathbf{y}}$*,* m_f *and* $\tilde{K}_{\mathbf{yy}}$ *are the posterior mean function and covariance of* f *evaluated at* $\mathbf{y}$ *respectively.* $\lambda > 0$ *is the regularisation of the* CME*.* $\lambda_f > 0$ *is the noise term for GP* f*.* Ω_x *is the set of variables as specified in Prop. 4.2.1.*

Summary: The posterior covariance between x and x' in IMP can be interpreted as the similarity between their corresponding empirical IMEs $\hat{\mu}_{Y|do(X)=x}$ and $\hat{\mu}_{Y|do(X)=x'}$ weighted by the posterior covariance $\tilde{K}_{\mathbf{yy}}$, where the latter corresponds to the uncertainty when modelling f as a GP in $\mathcal{D}_2$. However, since f only considers uncertainty in $\mathcal{D}_2$, we need to develop a method that allows us to quantify uncertainty when learning the IME from $\mathcal{D}_1$. In the next section, we introduce a Bayesian version of CME, which then lead to BAYESIME, a remedy to this problem.

4.3.2 Bayesian Interventional Mean Embedding

To account for the uncertainty in $\mathcal{D}_1$ when estimating $\mu_{Y|do(X)}$, we consider a GP model for CME, and later extend to the interventional embedding IME. We note that Bayesian formulation of CMEs has also been considered in Hsu et al. [2018], but with a specific focus on discrete target spaces.

Bayesian learning of conditional mean embeddings with V-GP. As mentioned in Sec. 4.2, CMEs have a clear "feature-to-feature" regression perspective, i.e. $\mathbb{E}[\phi_y(Y)|X = x]$ is the result of regressing $\phi_y(Y)$ onto $\phi_x(X)$. Hence, we consider a vector-valued GP construction to estimate the CME.

Let $\mu_{gp}(x, y)$ be a GP that models $\mu_{Y|X=x}(y)$. Given that $f \in \mathcal{H}_{k_y}$, for $\langle f, \mu_{gp}(x, \cdot) \rangle_{\mathcal{H}_{k_y}}$ to be well-defined, we need to ensure $\mu_{gp}(x, \cdot)$ is also restricted to $\mathcal{H}_{k_y}$ for any fixed x. Consequently, we cannot define a $\mathcal{GP}(0, k_x \otimes k_y)$ prior on μ_{gp} as usual, as draws from such prior will almost surely fall outside $\mathcal{H}_{k_x} \otimes \mathcal{H}_{k_y}$ [Lukić and Beder, 2001]. Instead, we define a prior over $\mu_{gp} \sim \mathcal{GP}(0, k_x \otimes r_y)$, where r_y is a *nuclear dominant kernel* [Lukić and Beder, 2001] over k_y, which ensures that samples paths of $\mu_{gp}(x, \cdot)$ (as a function of y) live in $\mathcal{H}_{k_y}$ almost surely[4]. In particular, we follow a similar construction as

[4]It doesn't matter if the sample paths drop outside of $\mathcal{H}_{k_x}$ as we are not defining inner product on $\mathcal{H}_{k_x}$

in Flaxman et al. [2016] and model r_y as $r_y(y_i, y_j) = \int k_y(y_i, u)k_y(u, y_j)\nu(du)$ where ν is some finite measure on Y. Hence, we can now set up a vector-valued regression in $\mathcal{H}_{k_y}$ as follows:

$$\phi_y(y_i) = \mu_{gp}(x_i, \cdot) + \lambda^{\frac{1}{2}}\epsilon_i \tag{4.10}$$

where $\epsilon_i \sim \mathcal{GP}(0, r)$ are independent noise functions. By taking the inner product with $\phi_y(y')$ on both sides, we then obtain $k_y(y_i, y') = \mu_{gp}(x_i, y') + \lambda^{\frac{1}{2}}\epsilon_i(y')$. Hence, we can treat $k(y_i, y_j)$ as noisy evaluations of $\mu_{gp}(x_i, y_j)$ and obtain the following posterior mean and covariance for μ_{gp}.

Proposition 4.3.2 (BAYESCME). *The posterior* GP *of μ_{gp} given observations $\{\mathbf{x}, \mathbf{y}\}$ has the following mean and covariance:*

$$m_\mu((x, y)) = k_{x\mathbf{x}}(K_{\mathbf{xx}} + \lambda I)^{-1}K_{\mathbf{yy}}R_{\mathbf{yy}}^{-1}r_{\mathbf{y}y} \tag{4.11}$$

$$\kappa_\mu((x, y), (x', y')) = k_{xx'}r_{y,y'} - k_{x\mathbf{x}}(K_{\mathbf{xx}} + \lambda I)^{-1}k_{\mathbf{x}x'}r_{\mathbf{yy}}R_{\mathbf{yy}}^{-1}r_{\mathbf{y}y'} \tag{4.12}$$

In addition, the following marginal likelihood can be used for hyperparameter optimisation,

$$-\frac{N}{2}\Big(\log|K_{\mathbf{xx}} + \lambda I| + \log|R|\Big) - \frac{1}{2}\operatorname{Tr}\Big((K_{\mathbf{xx}} + \lambda I)^{-1}K_{\mathbf{yy}}R_{\mathbf{yy}}^{-1}K_{\mathbf{yy}}\Big) \tag{4.13}$$

Note that in practice we fix the lengthscale of k_y and r_y when optimising the above likelihood. This is to avoid trivial solutions for the vector-valued regression problem as discussed in Ton et al. [2021a]. The Bayesian version of the IME is derived analogously, and we refer the reader to the appendix due to limited space.

Finally, with V-GPs on embeddings defined, we can model $g(x)$ as $\langle f, \mu_{gp}(x, \cdot)\rangle_{H_{k_y}}$, which due to the linearity of the inner product, is itself a GP. Here, we first considered the case where f is a KRR learnt from $\mathcal{D}_2$ and call the model BAYESIME.

Proposition 4.3.3 (BAYESIME). *Given dataset $D_1 = \{(x_i, y_i, z_i)\}_{i=1}^N$ and $D_2 = \{(\tilde{y}_j, t_j)\}_{j=1}^M$, if f is a* KRR *learnt from $\mathcal{D}_2$ and $\mu_{Y|do(X)}$ modelled as a* V-GP *using $\mathcal{D}_1$, then $g = \langle f, \mu_{Y|do(X)}\rangle \sim \mathcal{GP}(m_2, \kappa_2)$ where,*

$$m_2(x) = \Phi_{\Omega_x}(x)^\top(K_{\Omega_x} + \lambda I)^{-1}K_{\mathbf{yy}}R_{\mathbf{yy}}^{-1}R_{\mathbf{y}\tilde{\mathbf{y}}}A \tag{4.14}$$

$$\kappa_2(x, x') = B\Phi_{\Omega_x}(x)^\top\Phi_{\Omega_x}(x) - C\Phi_{\Omega_x}(x)^\top(K_{\Omega_x} + \lambda I)^{-1}\Phi_{\Omega_x}(x') \tag{4.15}$$

where $A = (K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1}\mathbf{t}$, $B = A^\top R_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}A$ and $C = A^\top R_{\tilde{\mathbf{y}}\mathbf{y}}R_{\mathbf{yy}}^{-1}R_{\mathbf{y}\tilde{\mathbf{y}}}A$

Summary: Constants B and C in κ_2 can be interpreted as different estimation of $||f||_{\mathcal{H}_{k_y}}$, i.e. the RKHS norm of f. As a result, problems that are "harder" to learn in $\mathcal{D}_2$, i.e. corresponding to the larger magnitude of $||f||_{\mathcal{H}_{k_y}}$, will result in larger values of B and C. Therefore, the covariance κ_2 can be interpreted as uncertainty in $\mathcal{D}_1$ scaled by the difficulty of the problem to learn in $\mathcal{D}_2$.

4.3.3 Bayesian Interventional Mean Process

To incorporate both uncertainties in $\mathcal{D}_1$ and $\mathcal{D}_2$, we combine ideas from IMP and BAYESIME to estimate $g = \langle f, \mu_{Y|do(X)} \rangle$ by placing GPs on both f and $\mu_{Y|do(X)}$. Again as before, a nuclear dominant kernel r_y was used to ensure the GP f is supported on $\mathcal{H}_{k_y}$. For ease of computation, we consider a finite-dimensional approximation of the GPs f and $\mu_{Y|do(X)}$ and estimate g as the RKHS inner product between them. In the following, we collate $\mathbf{y}$ and $\tilde{\mathbf{y}}$ into a single set of points $\hat{\mathbf{y}}$, which can be seen as landmark points for the finite approximation [Trecate et al., 1999]. We justify this in the Appendix.

Proposition 4.3.4 (BAYESIMP). *Let f and $\mu_{Y|do(X)}$ be* GP*s learnt as above. Denote $\tilde{f}$ and $\tilde{\mu}_{Y|do(X)}$ as the finite dimensional approximation of f and $\mu_{Y|do(X)}$ respectively. Then $\tilde{g} = \langle \tilde{f}, \tilde{\mu}_{Y|do(X)} \rangle$ has the following mean and covariance:*

$$m_3(x) = E_x K_{\mathbf{y}\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} R_{\hat{\mathbf{y}}\tilde{\mathbf{y}}} (R_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1} \mathbf{t} \tag{4.16}$$

$$\kappa_3(x, x') = \underbrace{E_x \Theta_1^\top \tilde{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}} \Theta_1 E_{x'}^\top}_{\text{Uncertainty from } \mathcal{D}_1} + \underbrace{\Theta_2^{(a)} F_{xx'} - \Theta_2^{(b)} G_{xx'}}_{\text{Uncertainty from } \mathcal{D}_2} + \underbrace{\Theta_3^{(a)} F_{xx'} - \Theta_3^{(b)} G_{xx'}}_{\text{Uncertainty from Interaction}} \tag{4.17}$$

where $E_x = \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1}, F_{xx'} = \Phi_{\Omega_x}(x)^\top \Phi_{\Omega_x}(x'), G_{xx'} = \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} \Phi_{\Omega_x}(x')$, and $\Theta_1 = K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} R_{\hat{\mathbf{y}}\mathbf{y}} R_{\mathbf{y}\mathbf{y}}^{-1} K_{\mathbf{y}\mathbf{y}}$, $\Theta_2^{(a)} = \Theta_4^\top R_{\hat{\mathbf{y}}\hat{\mathbf{y}}} \Theta_4, \Theta_2^{(b)} = \Theta_4^\top R_{\hat{\mathbf{y}}\mathbf{y}} R_{\mathbf{y}\mathbf{y}}^{-1} R_{\mathbf{y}\hat{\mathbf{y}}} \Theta_4$ and $\Theta_3^{(a)} = tr(K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} R_{\hat{\mathbf{y}}\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} \bar{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}), \Theta_3^{(b)} = tr(R_{\hat{\mathbf{y}}\mathbf{y}} R_{\mathbf{y}\mathbf{y}}^{-1} R_{\mathbf{y}\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} \bar{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1})$ and $\Theta_4 = K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} R_{\hat{\mathbf{y}}\tilde{\mathbf{y}}} (K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f)^{-1} \mathbf{t}$. $\bar{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}$ is the posterior covariance of f evaluated at $\hat{\mathbf{y}}$

Summary: While the first two terms in κ_3 resemble the uncertainty estimates from IMP and BAYESIME, the last term acts as an extra interaction between the two uncertainties from $\mathcal{D}_1$ and $\mathcal{D}_2$. We note that unlike IMP and BAYESIME, $\tilde{g}$ from Prop. 4.3.4 is not a GP as inner products between Gaussian vectors are not Gaussian. Nonetheless, the mean and covariance can be estimated.

4.4 Experiments

In this section, we first present an ablation study on how our methods would perform under settings where we have missing data parts at different regions of the two datasets. We then demonstrate BAYESIMP's

proficiency in the Causal Bayesian Optimisation setting.

In particular, we compare our methods against the sampling approach considered in Aglietti et al. [2020b]. Aglietti et al. [2020b] start by modelling $f : Y \to T$ as GP and estimate the density $p(Y|do(X))$ using a GP along with *do*-calculus. Then given a treatment x, we obtain L samples of y_l and R samples of f_r from their posterior GPs. The empirical mean and standard deviation of the samples $\{f_r(y_l)\}_{l=1,r=1}^{L,R}$ can now be taken to estimate $\mathbb{E}[T|do(X) = x]$ as well as the correspondingly uncertainty. We emphasize that this point estimation requires repeated sampling and is thus inefficient compared to our approaches, where we explicitly model the uncertainty as a covariance function.

Ablation study. In order to get a better intuition into our methods, we will start off with a preliminary example, where we investigate the uncertainty estimates in a toy case. We assume two simple causal graphs $X \to Y$ for $\mathcal{D}_1$ and $Y \to T$ for $\mathcal{D}_2$ and the goal is to estimate $\mathbb{E}[T|do(X) = x]$ (generating process given in the appendix). We compare our methods from Sec.4.3 with the sampling-based uncertainty estimation approach described above. In Fig. 4.5 we plot the mean and the 95% credible interval of the resulting GP models for $\mathbb{E}[T|do(X) = x]$. On the x-axis we also plotted a histogram of the treatment variable x to illustrate its density.

From Fig. 4.5(a), we see that the uncertainty for sampling is rather uniform across the ranges of x despite the fact we have more data around $x = 0$. This is contrary to our methods, which show a reduction of uncertainty at high x density regions. In particular, $x = -5$ corresponds to an extrapolation of data, where x gets mapped to a region of y where there is no data in $\mathcal{D}_2$. This fact is nicely captured by the spike of credible interval in Fig. 4.5(c) since IMP utilises uncertainty from $\mathcal{D}_2$ directly. Nonetheless, IMP failed to capture the uncertainty stemming from $\mathcal{D}_1$, as seen from the fact that the credible interval did not increase as we have fewer data in the region $|x| > 5$. In contrast, BAYESIME (Fig. 4.5(d)) gives higher uncertainty around low x density regions but failed to capture the **extrapolation** phenomenon. Finally, BAYESIMP Fig. 4.5(e) seems to inherit the desirable characteristics from both IMP and BAYESIME, due to taking into account uncertainties from both $\mathcal{D}_1$, $\mathcal{D}_2$. Hence, in our experiments, we focus on BAYESIMP and refer the reader to the appendix for the remaining methods.

BayesIMP for Bayesian Optimisation (BO). We now demonstrate, on both synthetic and real-world data, the usefulness of the uncertainty estimates obtained using our methods in BO tasks. Our goal is to utilise the uncertainty estimates to direct the search for the optimal value of $\mathbb{E}[T|do(X) = x]$ by querying as few values of the treatment variable X as possible, i.e. we want to optimize for $x^* = \arg\min_{x \in \mathcal{X}} \mathbb{E}[T|do(X) = x]$. For the first synthetic experiment (see Fig. 4.6 (Top)), we will use the following two datasets: $\mathcal{D}_1 = \{x_i, u_i, z_i, y_i\}_{i=1}^N$ and $\mathcal{D}_2 = \{\tilde{y}_j, t_j\}_{j=1}^M$. Note that BAYESIMP from

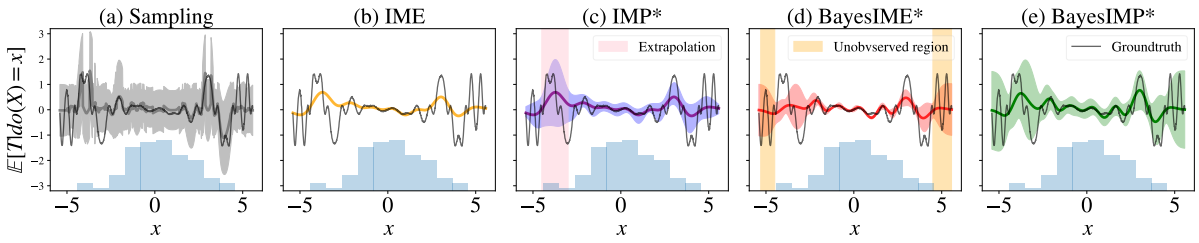


Figure 4.5: Ablation studies of various methods in estimating uncertainties for an illustrative experiment. $*$ indicates our methods. $N = M = 100$ data points are used. The histogram (blue bars) lemme for the treatment variable x is shown as well. Uncertainty from sampling gives a uniform estimate of uncertainty and IME does not come with uncertainty estimates. We see IMP and BAYESIME covering different regions of uncertainty while BAYESIMP takes the best of both worlds.

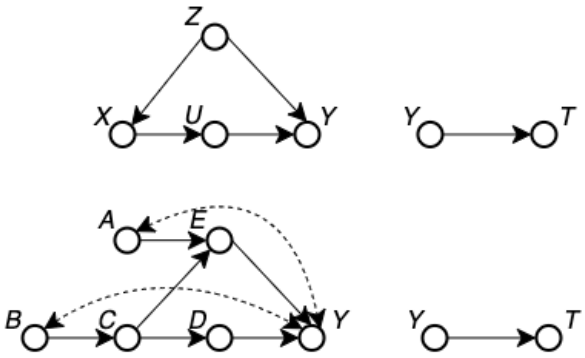


Figure 4.6: Illustration of synthetic data experiments.

Prop. 4.3.4 is not a GP as inner products between Gaussian vectors are not Gaussian. Nonetheless, with the mean and covariance estimated, we will use moment matching to construct a GP out of BAYESIMP for posterior inference. At the start, we are given $\mathcal{D}_1$ and $\mathcal{D}_2$, where these observations are used to construct a GP prior for the interventional effect of X on T, i.e $\mathbb{E}[T|do(X) = x]$, to provide a "warm" start for the BO.

Again we compare BAYESIMP with the sampling-based estimation of $\mathbb{E}[T|do(X)]$ and its uncertainty, which is exactly the formulation used in the Aglietti et al. [2020b]. In order to demonstrate how BAYESIMP performs in the multimodal setting, we will be considering the case where we have the following distribution on Y i.e. $p(y|u, z) = \pi p_1(y|u, z) + (1 - \pi)p_2(y|u, z)$ where Y is a mixture and $\pi \in [0, 1]$. These scenarios might arise when there is an unobserved binary variable that induces a switching between two regimes on how Y depends on (U, Z). In this case, the sampling-based GP model of Aglietti et al. [2020b] would only capture the conditional expectation of Y with an inflated variance, leading to slower convergence and higher variance in the estimates of the prior as we will show in our experiments. Throughout our experiments, similarly to Aglietti et al. [2020b], we will be using the expected improvement (EI) acquisition function to select the next point to query.

Synthetic data experiments. We compare BAYESIMP to Aglietti et al. [2020b]'s sampling approach as well as to a simple GP with no learned prior as the baseline. We will be using $N = 100$ data points for $\mathcal{D}_1$ and $M = 50$ data points $\mathcal{D}_2$. We ran each method 10 times and plot the resulting standard deviation for each iteration in the figures below. The data generation and details were added in the Appendix. We see from Fig .4.7 that BAYESIMP is able to find the maxima much faster and with smaller

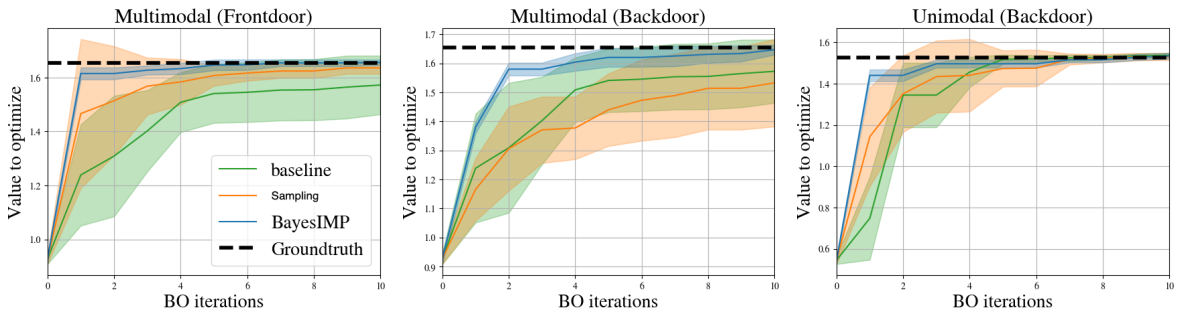


Figure 4.7: We are interested in finding the maximal value of $\mathbb{E}[T|do(X) = x]$ with as few BO iterations as possible. We ran experiments with **multimodality** in Y. (Left) Using front-door adjustment (Middle) Using backdoor adjustment (Right) Using backdoor adjustment (**unimodal** Y)

standard deviations, than the current state-of-the-art sampling method, using both front-door and backdoor adjustments (Fig. 4.7(Right, Middle)). Given that our method uses more flexible representations of conditional distributions, we are able to circumvent the multimodality problem in Y. In addition, we also consider the unimodal version, i.e. $\pi = 0$ (see right Fig. 4.7). We see that the performance of sampling improves in the unimodal setting, however BAYESIMP still converges faster than sampling even in this scenario.

Next, we consider a harder causal graph (see Fig. 4.6 (Bottom)), previously considered in Aglietti et al. [2020b]. We again introduce multimodality in the Y variable in order to explore the case of more challenging conditional densities. We see from Fig. 4.8 (Left, Middle), that BAYESIMP again converges much faster to the true optima than sampling and the standard GP prior baseline. We note that the fast convergence of BAYESIMP throughout our experiments is not due to the simplicity of the underlying BO problems. Indeed, the BO with a standard GP prior requires significantly more iterations. It is rather the availability of the observational data, that allows us to construct a more appropriate prior, which leads to a "warm" start of the BO procedure.

Healthcare experiments. We conclude with a healthcare dataset corresponding to our motivating medical example in Fig. 4.1. The causal mechanism graph, also considered in the Aglietti et al. [2020b], studies the effect of certain drugs (Aspirin/Statin) on Prostate-Specific Antigen (PSA) levels Ferro et al. [2015]. In our case, we modify *statin* to be continuous, in order to optimize for the correct drug dosage.

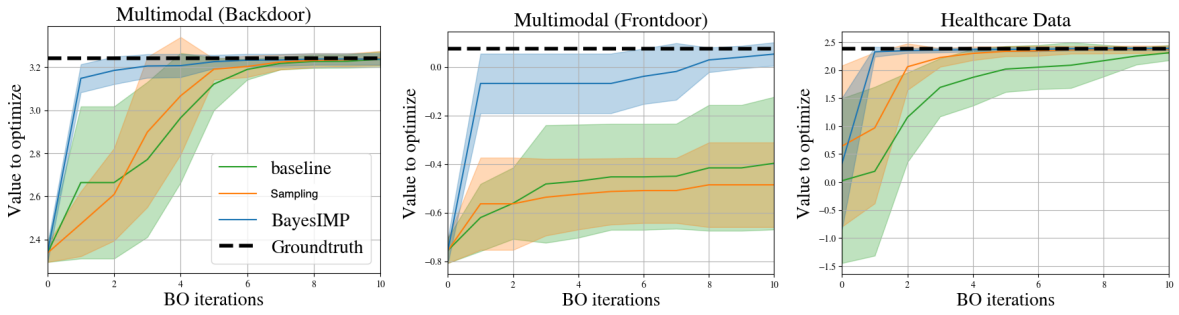


Figure 4.8: (Left) Experiments where we are interested in $\mathbb{E}[T|do(D) = d]$ with **multimodal** Y, (Middle) Experiments where we are interested in $\mathbb{E}[T|do(E) = e]$ with **multimodal** Y, (Right) Experiments on **healthcare data** where we are interested in $\mathbb{E}[\textit{Cancer Volume}|\textit{do(Statin)}]$.

However, in contrast to Aglietti et al. [2020b], we consider a second experimental dataset, arising from a different medical study, which looks into the connection between *PSA* levels and *cancer volume* amount in patients [Stamey et al., 1989]. Similar to Aglietti et al. [2020b], given that interventional data is hard to obtain, we construct data generators based on the true data collected in Stamey et al. [1989]. This is done by first fitting a GP on the data and then sampling from the posterior (see Appendix for more details). Hence, this is the perfect test bed for our model where we are interested in $\mathbb{E}[\textit{Cancer Volume}|\textit{do(Statin)}]$. We see from Fig. 4.8 (Right) that BAYESIMP again converges to the true optima faster than sampling hence allowing us to find the fastest ways of optimizing *cancer volume* by requesting much less interventional data. This could be critical as interventional data in real-life situations can be very expensive to obtain.

4.5 Discussion and Conclusion

In this chapter we propose BAYESIMP for quantifying uncertainty in the setting of causal data fusion. In particular, our proposed method BAYESIMP allows us to represent interventional densities in the RKHS without explicit density estimation, while still accounting for epistemic and aleatoric uncertainties. We demonstrated the quality of the uncertainty estimates in a variety of Bayesian optimization experiments, in both synthetic and real-world healthcare datasets, and achieve significant improvement over current SOTA in terms of convergence speed. However, we emphasize that BAYESIMP is not designed to replace CBO but rather an alternative model for interventional effects.

In the future, we would like to improve BAYESIMP over several limitations. As in Aglietti et al. [2020b], we assumed full knowledge of the underlying causal graph, which might be limited in practice. Furthermore, as the current formulation of BAYESIMP only allows the combination of two causal graphs, we hope to generalise the algorithm into an arbitrary number of graphs in the future. Causal graphs with recurrent structures will be an interesting direction to explore. Lastly, we would also like to include a cost

function as in Aglietti et al. [2020b] to constrain the search space to the most sensible solutions.

5 | RKHS-SHAP: Shapley Values for Kernel Methods

This chapter is based on the following publication

> **Siu Lun Chau**, Robert Hu, Javier Gonzalez, and Dino Sejdinovic. "RKHS-SHAP: Shapley Values for Kernel Methods" Advances in Neural Information Processing Systems (NeurIPS), 2022

Abstract

Feature attribution for kernel methods is often heuristic and not individualised for each prediction. To address this, we turn to the concept of Shapley values (SV), a coalition game theoretical framework that has previously been applied to different machine learning model interpretation tasks, such as linear models, tree ensembles, and deep networks. By analysing SVs from a functional perspective, we propose RKHS-SHAP, an attribution method for kernel machines that can efficiently compute both *Interventional* and *Observational Shapley values* using kernel mean embeddings of distributions. We show theoretically that our method is robust with respect to local perturbations - a key yet often overlooked desideratum for consistent model interpretation. Further, we propose *Shapley regulariser*, applicable to a general empirical risk minimisation framework, allowing learning while controlling the level of specific features' contributions to the model. We demonstrate that the Shapley regulariser enables learning which is robust to covariate shift of a given feature and fair learning which controls the SVs of sensitive features.

5.1 Introduction

Machine learning model interpretability is critical for researchers, data scientists, and developers to explain, debug and trust their models and understand the value of their findings. A typical way to understand model performance is to attribute importance scores to each input feature [Carvalho et al., 2019]. These scores can be computed either for an entire dataset to explain the model's overall behaviour (global) or compute individually for every single prediction (local).

Understanding feature importances in reproducing kernel Hilbert space (RKHS) methods such as kernel ridge regression and support vector machines often require the study of kernel lengthscales across dimensions [Williams and Rasmussen, 2006, Chapter 5]. The larger the value, the less relevant the feature is to the model. Albeit straightforward, this approach comes with three shortcomings:

1. It only provides global feature importance and cannot be individualised to every single prediction. This explanation is limited as global importance does not necessarily imply local importance [Ribeiro et al., 2016]). In safety-critical domain such as medicine, understanding individual prediction is arguably more important than capturing the general model performance. See Fig 5.1 for an example of local explanation.

2. The tuning of lengthscales often requires a user-specified grid of possible configurations and is selected using cross-validations. This pre-specification thus injects a substantial amount of human bias to the explanation task.

3. Lengthscales across kernels acting on different data types, such as binary and continuous variables, are difficult to compare and interpret.

To address this problem we turn to the Shapley value (SV) [Shapley, 1953] literature, which has become central to many model explanation methods in recent years. The Shapley value was originally a concept used in game theory that involves fairly distributing credits to players working in coalition. Štrumbelj and Kononenko [2014] were one of the first to connect SV with machine learning explanations by casting predictions as coalition games, and features as players. Since then, a variety of SV based explanation models were proposed. For example, LINEARSHAP [Štrumbelj and Kononenko, 2014] for linear models, TREESHAP [Lundberg et al., 2018] for tree ensembles and DEEPSHAP [Lundberg and Lee, 2017] for deep networks. Model agnostic methods such as DATA-SHAPLEY [Ghorbani and Zou, 2019], SAGE [Covert et al., 2020] and KERNELSHAP [1] [Lundberg and Lee, 2017] were also proposed. However, to the best of our knowledge, an SV-based local feature attribution framework suited for kernel

[1] The kernel in KERNELSHAP refers to the estimation procedure is not related to RKHS kernel methods.

methods has not been proposed.

While one could still apply model-agnostic KERNELSHAP on kernel machines, we show that by representing distributions as elements in the RKHS through kernel mean embeddings [Song et al., 2013, Muandet et al., 2016a], we can compute Shapley values more efficiently by circumventing the need to sample and estimate an exponential amount of densities required to compute the value functions, an essential component for Shapley value computation. We call this approach RKHS-SHAP to distinguish it from KERNELSHAP. Through the lens of RKHS, we study Shapley values from a functional perspective and prove that our method is robust with respect to local perturbations under mild assumptions, which is an important yet often neglected criterion for explanation models as discussed in Hancox-Li [2020]. In addition, a *Shapley regulariser* based on RKHS-SHAP is proposed for the empirical risk minimisation framework, allowing the modeller to control the degree of feature contribution during the learning. We also discuss its application to robust learning to covariate shift of a given feature and fair learning while controlling contributions from sensitive features. We summarise our contributions below:

1. We propose RKHS-SHAP, a model-specific algorithm to compute Shapley values efficiently for kernel methods by circumventing the need to sample and fit from an exponential number of densities.
2. We prove that the corresponding Shapley values are robust to local perturbations under mild assumptions, thus providing consistent explanations for the kernel model.
3. We propose a *Shapley regulariser* for the empirical risk minimisation framework, allowing the modeller to control the degree of feature contribution during the learning.

5.2 Background Materials

Notation. We denote X, Y as random variables (rv) with distribution $p(X, Y)$ taking values in the d-dimensional instance space $\mathcal{X} \subseteq \mathbb{R}^d$ and the label space $\mathcal{Y}$ (could be in $\mathbb{R}$ or discrete) respectively. We use $D = \{1, ..., d\}$ to denote the feature index set of X and $S \subseteq D$ to denote the subset of features of interests. Lowercase letters are used to denote observations from corresponding rvs.

5.2.1 The Shapley Value

The Shapley value was first proposed by Shapley [1953] to allocate performance credit across coalition game players in the following sense: Let $\nu : \{0, 1\}^d \to \mathbb{R}$ be a *coalition game* that returns a score for each coalition $S \subseteq D_g$, where $D_g = \{1, ..., d\}$ represents a set of players. Assuming the grand coalition D_g is participating and one wishes to provide the i^{th} player with a fair allocation of the total profit $\nu(D_g)$, how

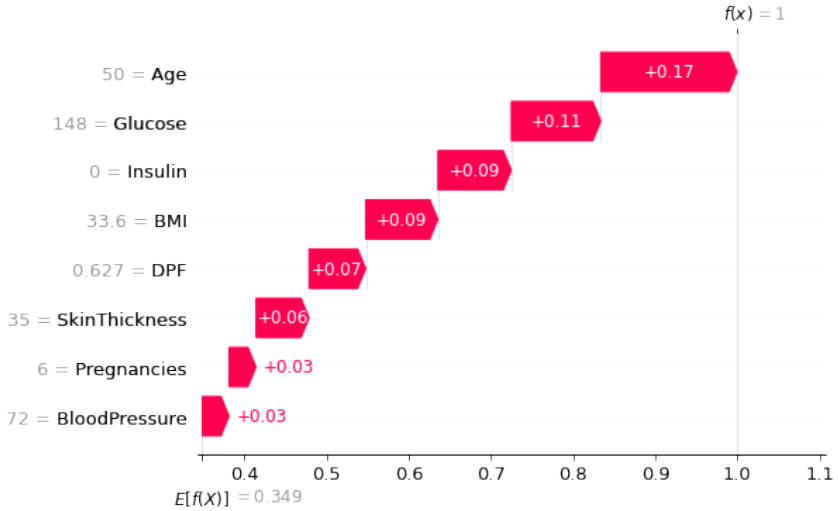


Figure 5.1: An example of RKHS-SHAP providing local explanations to why a kernel logistic model predicts this patient to be diabetic [Kaggle, 2022]. RKHS-SHAP provides a more granular level of explanation than studying lengthscales across dimensions.

should one do it? Surely this is related to each player's *marginal contribution* to the profit with respect to a coalition S, i.e. $\nu(S \cup i) - \nu(S)$. Shapley [1953] proved that there exists a *unique* combination of marginal contributions that satisfy a set of favourable and fair game theoretical axioms, commonly known as *efficiency, null player, symmetry* and *additivity*. This unique combination of contributions is later denoted as the *Shapley value*. Formally, given a coalition game ν, the Shapley value for player i is computed as the following,

$$\phi_i(\nu) = \frac{1}{d} \sum_{S \subseteq D_g \setminus \{i\}} \binom{d-1}{|S|}^{-1} \Big(\nu(S \cup i) - \nu(S) \Big). \tag{5.1}$$

Choosing ν for ML explanation In recent years, the Shapley value concept has become popular for feature attribution in machine learning. SHAP [Lundberg and Lee, 2017], SHAPLEY EFFECT [Song et al., 2016], DATA-SHAPLEY [Ghorbani and Zou, 2019] and SAGE [Covert et al., 2020] are all examples that cast model explanations as coalition games by choosing problem-specific value functions ν. Denote $f : \mathcal{X} \to \mathcal{Y}$ as the machine learning model of interest. Value functions for local attribution on observation x often take the form of the expectation of f with respect to some reference distribution $r(X_{S^c} \mid X_S = x_S)$, where $S \subseteq D$ is some coalition of features in analogous to the game theory setting, such that:

$$\nu_{x,S}(f) = \mathbb{E}_{r(X_{S^c}|X_S=x_S)}[f(\{x_S, X_{S^c}\}], \tag{5.2}$$

where $\{x_S, X_{S^c}\}$ denotes the concatenation of the arguments. We wrote f as the main argument of ν to highlight its interpretation as a functional indexed by local observation x and coalition S. When r is set to be marginal distribution, i.e $r(X_{S^c} \mid X_S = x_S) = p(X_{S^c})$, the value function is denoted as the *Interventional value function* by Janzing et al. [2020]. *Observational value function* [Frye et al., 2020], on the other hand, set the reference distribution to be a conditional distribution $p(X_{S^c} \mid X_S = x_S)$. Other choices of reference distributions will lead to Shapley values with specific properties, e.g., better locality of explanations [Ghalebikesabi et al., 2021] or incorporating causal knowledge [Heskes et al., 2020]. In this work, we shall restrict our attention to marginal and conditional cases as they are the two most commonly adopted choices in the literature.

Definition 5.2.1 (Value functions)**.** *Given model f, local observation x and a coalition set $S \subseteq D$, the Interventional and Observational value functions are denoted by,*

$$\nu^{(I)}_{x,S}(f) := \mathbb{E}[(f(x_S, X_{S^c})], \tag{5.3}$$

$$\nu^{(O)}_{x,S}(f) := \mathbb{E}[f(x_S, X_{S^c})|X_S = x_S]. \tag{5.4}$$

The right choice of ν has been a long-standing debate in the community. While Janzing et al. [2020] argued from a causal perspective that $\nu^{(I)}_{x,f}$ is the correct notion to represent the missingness of features in an explanation task, Frye et al. [2020] argued that computing marginal expectation ignores feature correlation and leads to unrealistic results since one would be evaluating the value function outside the data-manifold. This controversy was further investigated by Chen et al. [2020], where they argued that the choice of ν is *application dependent* and the two approaches each lead to an explanation that is either *true to the model* (marginal expectation) or *true to the data* (conditional expectation). When the context is clear, we denote the Shapley value of the i^{th} feature of observation x at f as $\phi_{x,i}(f)$ and use a superscript to indicate whether it is *Interventional* $\phi^{(I)}_{x,i}(f)$ or *Observational* $\phi^{(O)}_{x,i}(f)$.

Computing Shapley values. While Shapley values can be estimated directly from Eq. (5.1) using a sampling approach [Štrumbelj and Kononenko, 2014], Lundberg and Lee [2017] proposed KERNELSHAP, a more efficient algorithm for estimating Shapley values in high dimensional feature spaces by casting Eq. (5.1) as a weighted least square problem. Similar to LIME [Ribeiro et al., 2016], for each data x, model f, and feature coalition S, KERNELSHAP places a linear model $u_x(S) = \beta_{x,0} + \sum_{i \in S} \beta_{x,i}$ to explain the value function $\nu_{x,S}(f)$, which corresponds to solving the following regression problem:

$$\min_{\beta_{x,0},\ldots,\beta_{x,d}} \sum_{S \subseteq D} w(S)(u_x(S) - \nu_{x,S}(f))^2$$

, where $w(S) = \frac{d-1}{\binom{d}{|S|}|S|(d-|S|)}$ is a carefully chosen weighting such that the regression coefficients recover Shapley values. In particular, we typically set $w(\varnothing) = w(D) = \infty$ to effectively enforce constraints $\beta_{x,0} = \nu_{x,\varnothing}(f)$ and $\sum_{i \in D} \beta_{x,i} = \nu_{x,D}(f) - \nu_{x,\varnothing}(f)$. Denoting each subset $S \subseteq D$ using the corresponding binary vector $\mathbf{z} \in \{0,1\}^d$, and with an abuse of notation by setting $\nu_{\cdot,\mathbf{z}} := \nu_{\cdot,S}$ and $w(\mathbf{z}) := w(S)$ for $S = \{j : \mathbf{z}[j] = 1\}$, we can express the Shapley values $\boldsymbol{\beta}_x := [\beta_{x,0}, ..., \beta_{x,d}]$ as $\boldsymbol{\beta}_x = (Z^\top W Z)^{-1} Z^\top W \mathbf{v}_x$ where $Z \in \mathbb{R}^{2^d \times d}$ is the binary matrices with columns $\{\mathbf{z}_i\}_{i=1}^{2^d}$, W is the diagonal matrix with entries $w_{ii} = w(\mathbf{z}_i)$ and $\mathbf{v}_x := \{\nu_{x,\mathbf{z}_i}(f)\}_{i=1}^{2^d} \in \mathbb{R}^{2^d \times 1}$ the vector of evaluated value functions, which is often estimated using sampling and data imputations. We shall explain the pathology of this approach in detail later in Section 5.3. In practice, instead of evaluating at all 2^d combinations, one would subsample the coalitions $z \sim w(z)$ for computational efficiency [Covert and Lee, 2021].

Model-specific Shapley methods. KERNELSHAP provides efficient model-agnostic estimations of Shapley values. However, by leveraging additional structural knowledge about specific models, one could further improve computational performance. This leads to a variety of model-specific approximations, most of which rely on utilising their specific structure to speed up the computation of value functions. For example, LINEARSHAP [Štrumbelj and Kononenko, 2014] explain linear models using model coefficients directly. TREESHAP [Lundberg et al., 2018] provides an exponential reduction in complexity compared to KERNELSHAP by exploiting the tree structure. DEEPSHAP [Lundberg and Lee, 2017], on the other hand, combines DEEPLIFT [Shrikumar et al., 2017] with Shapley values and uses the compositional nature of deep networks to improve efficiencies. However, to the best of our knowledge, a kernel method specific Shapley value approximation has not been studied. Later in Section 5.3, we show that kernel methods can be used to speed up the computation in KERNELSHAP for RKHS functions by estimating value functions analytically, thus circumventing the need for estimating and sampling from an exponential number of densities.

Related work on kernel-based Shapley methods. Da Veiga [2021]'s work on tackling global sensitive analysis by proposing the kernel-based maximum mean discrepancy as value function, is conceptually most similar to ours. However, there are multiple key differences in our contributions. Firstly, their method is designed for global explanation, while ours is for local. Secondly, similar to interventional SV, they do not consider any conditional distributions, thus leading to completely different estimation procedures and thus novelty. Lastly, their method is on understanding the input/outputs relationship of a numerical simulation model, while ours focuses on understanding specific RKHS models learnt from a machine learning task, e.g. kernel ridge regression and kernel logistic regression.

5.2.2 Kernel Methods

Kernel methods are one of the pillars of machine learning, as they provide flexible yet principled ways to model complex functional relationships and come with well-established statistical properties and theoretical guarantees.

Empirical Risk Minimisation. Recall in the supervised learning framework, we are learning a function $f : \mathcal{X} \to \mathcal{Y}$ from a hypothesis space $\mathcal{H}$, such that given training set $(\mathbf{x}, \mathbf{y}) = \{(x_i, y_i)\}_{i=1}^n$ sampled identically and independently from p, the following empirical risk is minimised:

$$f^* = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)) + \lambda_f \Omega(f)$$

, where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is the loss function, $\Omega : \mathcal{H} \to \mathbb{R}$ a regularisation function and λ_f a scalar controlling the level of regularisation. Denote $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a positive definite kernel with feature map ψ_x for input $x \in$ and $\mathcal{H}_k$ the corresponding RKHS. If we pick $\mathcal{H}_k$ as our hypothesis space, then the *Representer theorem* [Steinwart and Christmann, 2008a] tells us that the optimal solution takes the form of $f^* = \sum_{i=1}^n \alpha_i k(\cdot, x_i) = \Psi_{\mathbf{x}} \boldsymbol{\alpha}$, where $\Psi_{\mathbf{x}} = [\psi_{x_1} \dots \psi_{x_n}]$ is the feature matrix defined by stacking feature maps along columns. If ℓ is the squared loss then the above optimisation is known as kernel ridge regression and $\boldsymbol{\alpha}$ can be recovered in the closed form $\boldsymbol{\alpha} = (\mathbf{K}_{\mathbf{xx}} + \lambda_f I)^{-1}\mathbf{y}$, where $\mathbf{K}_{\mathbf{xx}} = \Psi_{\mathbf{x}}^\top \Psi_{\mathbf{x}}$ is the kernel matrix. If ℓ is the logistic loss, then the problem is known as kernel logistic regression, and $\boldsymbol{\alpha}$ can be obtained using gradient descent.

Kernel embedding of distributions. An essential component for RKHS-SHAP is the embedding of both marginal and conditional distribution of features into the RKHS [Song et al., 2013, Muandet et al., 2016a], thus allowing one to estimate the value function analytically. Formally, the kernel mean embedding (KME) of a marginal distribution P_X is defined as $\mu_X := \mathbb{E}_X[\psi_X] = \int_{\mathcal{X}} \psi_x dP_X(x)$ and the empirical estimate can be obtained as $\hat{\mu} := \frac{1}{n}\sum_{i=1}^n \psi_{x_i}$. Furthermore, given another kernel $g : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ with feature map ψ_Y of RKHS $\mathcal{H}_g$, the conditional mean embedding (CME) of the conditional distribution $P_{Y|X=x}$ is defined as $\mu_{Y|X=x} := \mathbb{E}[\psi_Y | X = x] = \int_{\mathcal{Y}} \psi_y dP_{Y|X=x}(y)$.

One way to understand CME is to view it as an evaluation of a vector-valued(VV) function $\mu_{Y|X} : \mathcal{X} \to \mathcal{H}_g$ such that $\mu_{Y|X}(x) = \mu_{Y|X=x}$, which minimises the following risk function $\mathbb{E}_{p(X,Y)}[||\psi_Y - \mu_{Y|X}(X)||^2_{\mathcal{H}_g}]$ [Grünewälder et al., 2012]. Let $\mathcal{L}(\mathcal{H}_g)$ be the space of bounded linear operators from $\mathcal{H}_g$ to itself. Denote $\Gamma_x : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{H}_g)$ as the operator-valued kernel such that $\Gamma_x(x, x') = k(x, x')\mathbf{1}$ with $\mathbf{1}$ the identity operator on $\mathcal{H}_g$. We denote $\mathcal{H}_{\Gamma_x}$ as the corresponding vector-valued RKHS. By utilising the

VV-Representer theorem [Micchelli and Pontil, 2005], we could minimises the following empirical risk:

$$\hat{\mu}_{Y|X} = \underset{\mu_{Y|X} \in \mathcal{H}_{\Gamma_x}}{\arg\min} \sum_{i=1}^{n} ||\psi_{y_i} - \mu_{Y|X}(x_i)||^2_{\mathcal{H}_g} + n\eta ||\mu_{Y|X}||^2_{\Gamma_x}$$

where $\eta > 0$ is a regularisation parameter. This leads to the following empirical estimate of the CME, i.e., $\hat{\mu}_{Y|X} = \Psi_{\mathbf{y}}(\mathbf{K}_{\mathbf{xx}} + n\eta I)^{-1}\Psi_{\mathbf{x}}^{\top}$, where $\Psi_{\mathbf{y}} := [\psi_{y_1}...\psi_{y_n}]$ and $\Psi_{\mathbf{x}} := [\psi_{x_1}...\psi_{x_n}]$ are feature matrices. Intuitively, this essential turns CME estimation to a regression problem from $\mathcal{X}$ to the vector-valued labels ψ_Y. Please see Micchelli and Pontil [2005] and Grünewälder et al. [2012] for further discussions on vector-valued RKHSs and CMEs. In fact, when using finite-dimensional feature maps, such as in the case with running Random Fourier Features [Rahimi et al., 2007] and Nyström methods [Yang et al., 2012] for scalability, one could reduce the computational complexity of evaluating empirical CME from $\mathcal{O}(n^3)$ to $\mathcal{O}(b^3) + \mathcal{O}(b^2 n)$ [Muandet et al., 2016a] where b is the dimension of the feature map and often can be chosen much smaller than n [Li et al., 2019a].

5.3 RKHS-SHAP

While KernelSHAP is model agnostic, by restricting our attention to the class of kernel methods, faster Shapley value estimation can be derived. We assume our RKHS takes a tensor product structure, i.e, $\mathcal{H}_k = \bigotimes_{i=1}^{d} \mathcal{H}_{k^{(i)}}$, where $k^{(i)}$ is the kernel for each dimension $i \in D$. This structural assumption allows us to decompose the value functionals into tensor products of embeddings and feature maps, thus we can estimate them analytically, as later shown in Prop. 5.3.1. Tensor product RKHSs are commonly used in practice, as they preserve universalities of kernels from individual dimension [Szabó and Sriperumbudur, 2017], thus providing a rich function space. Note that this assumption is not essential within our framework. Namely, for a non-product kernel, one can still evaluate the value functions using tools from conditional mean embeddings and utilise our interpretability pipeline without conditional density estimation. We show this in Appendix C.3. In the following, we will lay out the disadvantage of existing sampling and data imputation approach and show that by estimating the value functionals as elements in the RKHS, we can circumvent the need for learning and sampling from an exponential number of conditionals densities – thus improving the computational efficiency in the estimation.

Estimating value functions by sampling. Estimating the Observational value function $\nu^{(O)}_{x,S}(f)$ is typically much harder than the Interventional value function $\nu^{(I)}_{x,S}(f)$ as it requires integration with respect to the unknown conditional density $p(X_{S^c} \mid X_S)$. Therefore, estimating OSVs often boils down to a two-stage approach: (1) Conditional density estimation and (2) Monte Carlo averaging over imputed data, as shown in Aas et al. [2019], where they considered using multivariate Gaussian and Gaussian Copula

for density estimation. Recently, an alternate way to estimate observational value functions is proposed by Frye et al. [2020], where they formulate the estimation as a regression problem and compute the value function using a masked neural network directly without making any distributional assumption. This method shares conceptual similarities to ours but uses very different tools for the estimation. We highlight such differences in the appendix C.2.

Once the conditional density function $p(X_{S^c} \mid X_S)$ for each $S \subseteq D$ is estimated, the observational value function at the i^{th} observation x_i can then be computed by taking averages of m Monte Carlo samples from the estimated conditional density, i.e. $\frac{1}{m}\sum_{j=1}^{m} f(\{x_{iS}, x_{j S^c}\})$ where $\{x_{iS}, x_{j S^c}\}$ is the concatenation of x_{iS} with the j^{th} sample $x_{j S^c}$ from $p(X_{S^c}|X_S = x_{iS})$. Note further that the Monte Carlo samples cannot be reused for another observation x_k as their conditional densities are different. In other words, $n \times m$ Monte Carlo samples are required for each coalition S if one wishes to compute Shapley values for all n observations. This is clearly not desirable. In the spirit of Vapnik's principle[2], as our goal is to estimate conditional expectations that lead to Shapley values, we are not going to solve a harder and more general problem of conditional density estimation as an intermediate step, but instead utilise the arsenal of kernel methods to estimate the conditional expectations directly. Further discussion on comparing complexity of RKHS-SHAP with density estimation methods can be found in Appendix C.1.

Estimating value functions using mean embeddings. If our model f lives in $\mathcal{H}_k$, both the marginal and conditional expectation can be estimated analytically without any sampling or density estimation. We first show that the Riesz representations [Paulsen and Raghupathi, 2016] of both *Interventional* and *Observational value functionals* exist and are well-defined in $\mathcal{H}_k$. In the following, for simplicity, we will denote the functional and its corresponding Riesz representer using the same notation. For example, we will write $\nu_{x,S}(f) = \langle f, \nu_{x,S}\rangle_{\mathcal{H}_k}$ when the context is clear. Given a vector of n instances $\mathbf{x}$, we denote the corresponding vector of value functions as $\nu_{\mathbf{x},S}(f) = [\nu_{x_i,S}(f)]_{i=1}^{n}$. All proofs of this chapter can be found in Appendix C.4.

Proposition 5.3.1 (Riesz representations of value functionals)**.** *Denote k as the product kernel of d bounded kernels $k^{(i)} : \mathcal{X}^{(i)} \times \mathcal{X}^{(i)} \to \mathbb{R}$, where $\mathcal{X}^{(i)}$ is the domain of the i^{th} feature for $i \in D$. Riesz representations of the Interventional and Observational value functionals then exist and can be written as $\nu^{(I)}_{x,S} = \psi_{x_S} \otimes \mu_{X_{S^c}}$ and $\nu^{(O)}_{x,S} = \psi_{x_S} \otimes \mu_{X_{S^c}|X_S=x_S}$, where $\psi_{x_S} := \bigotimes_{i\in S} \psi_{x^{(i)}}$, $\mu_{X_{S^c}} := \mathbb{E}[\bigotimes_{i\in S^c} \psi_{x^{(i)}}]$ and $\mu_{X_{S^c}|X_S=x_S} := \mathbb{E}[\bigotimes_{i\in S^c} \psi_{x^{(i)}}|X_S = x_S]$.*

The corresponding finite sample estimators $\hat{\nu}^{(I)}_{x,S}$ and $\hat{\nu}^{(O)}_{x,S}$ are then obtained by replacing the corresponding

[2] *When solving a problem, try to avoid solving a more general one as an intermediate step.* [Vapnik, 1995, Section 1.9]

KME and CME components with their empirical estimators. As a result, given $f^* = \Psi_{\mathbf{x}}\boldsymbol{\alpha}$ trained on dataset $(\mathbf{x}, \mathbf{y})$, Prop. 5.3.1 allows us to estimate the value functionals analytically since $\hat{\nu}^{(I)}_{x,S}(f^*) = \langle f^*, \psi_{x_S} \otimes \hat{\mu}_{X_{S^c}} \rangle$ and $\hat{\nu}^{(O)}_{x,S}(f^*) = \langle f^*, \psi_{x_S} \otimes \hat{\mu}_{X^{S^c}|X_S=x_S} \rangle$. This corresponds to the direct non-parametric estimators of value functions given in the following proposition, which circumvent the need for sampling or density estimation.

Proposition 5.3.2 (Estimation). *Given $\mathbf{x}' \in \mathbb{R}^{n'}$ a vector of instances and $f = \Psi_{\mathbf{x}}\boldsymbol{\alpha}$, the empirical estimates of the functionals can be computed as, $\hat{\nu}^{(I)}_{\mathbf{x}',S}(f) = \boldsymbol{\alpha}^\top \mathcal{K}^{(I)}_{\mathbf{x}',S}$, $\hat{\nu}^{(O)}_{\mathbf{x}',S}(f) = \boldsymbol{\alpha}^\top \mathcal{K}^{(O)}_{\mathbf{x}',S}$, respectively, where $\mathcal{K}^{(I)}_{\mathbf{x}',S} = \mathbf{K}_{\mathbf{x}_S\mathbf{x}'_S} \odot \frac{1}{n} \operatorname{diag}(\mathbf{K}^\top_{\mathbf{x}_{S^c}\mathbf{x}_{S^c}}\mathbf{1}_{\mathbf{n}})\mathbf{1}_{\mathbf{n}}\mathbf{1}_{\mathbf{n}'}{}^\top$ and $\mathcal{K}^{(O)}_{\mathbf{x}',S} = \mathbf{K}_{\mathbf{x}_S\mathbf{x}'_S} \odot \Xi_S \mathbf{K}_{\mathbf{x}_S\mathbf{x}'_S}$, $\mathbf{1}_{\mathbf{n}}$ is the all-one vector with length n, $\odot$ the Hadamard product and $\Xi_S = \mathbf{K}_{\mathbf{x}_{S^c}\mathbf{x}_{S^c}}(\mathbf{K}_{\mathbf{x}_S\mathbf{x}_S} + n\eta I)^{-1}$.*

Finally, to obtain the Shapley values with these value functions, we deploy the same least square approach as KERNELSHAP.

Proposition 5.3.3 (RKHS-SHAP). *Given $f \in \mathcal{H}_k$ and ν, Shapley values $\mathbf{B} \in \mathbb{R}^{d \times n}$ for all d features and all n input $\mathbf{x}$ can be computed as $\mathbf{B} = (Z^\top W Z)^{-1} Z^\top W \hat{\mathbf{V}}$ where $\hat{\mathbf{V}}_{i,:} = \hat{\nu}_{\mathbf{x},S_i}(f)$.*

Estimating value functions with specific models. To the best of our knowledge, TreeSHAP [Lundberg et al., 2018] was the only machine learning model-specific SV algorithm computing conditional expectations using the properties of the model (tree in this case) directly, rather than relying on some sort of sampling procedure and density estimation. However, it is unclear how to validate the assumptions about feature distribution in TreeSHAP, which are specified as "the distribution generated by the tree", as discussed by Sundararajan and Najmi [2020]. In comparison, RKHS-SHAP does not pose assumptions on the underlying feature distribution and computes the corresponding conditional expectations via mean embeddings analytically. However, one should note that each of these model specific algorithm are only designed to explain specific models, therefore it is not informative to compare, e.g. TreeSHAP values with RKHS-SHAP values, as they are explaining different models.

5.3.1 Robustness of RKHS-SHAP

Robustness of interpretability methods is important from both an epistemic and ethical perspective, as discussed in Hancox-Li [2020]. On the other hand, Alvarez-Melis and Jaakkola [2018] showed empirically that Shapley methods when used with complex non-linear black-box models such as neural networks, yield explanations that vary considerably for some neighbouring inputs, even if the deep network gives

similar predictions at those neighbourhoods. In light of this, we analyse the Shapley values obtained from our proposed RKHS-SHAP and show that they are robust. To illustrate this, we first formally define the *Shapley functional*,

Proposition 5.3.4 (Shapley functional)**.** *Given a value functional ν indexed by input x and coalition S, i.e. $\nu_{x,S}$, the Shapley functional $\phi_{x,i} : \mathcal{H}_k \to \mathbb{R}$ such that $\phi_{x,i}(f)$ gives the i^{th} Shapley values of x on f, has the following Riesz representation in the RKHS:*

$$\phi_{x,i} = \frac{1}{d} \sum_{S \subseteq D \setminus \{i\}} \binom{d-1}{|S|}^{-1} \left(\nu_{x,S \cup i} - \nu_{x,S}\right)$$

Analogously, we denote $\phi_{x,i}^{(I)}$ and $\phi_{x,i}^{(O)}$ as the *Interventional Shapley functional* (ISF) and *Observational Shapley functional* respectively (OSF). Using the functional formalism, we now show that given $f \in \mathcal{H}_k$, when $||x - x'||^2 \leq \delta$ for $\delta > 0$, the difference in Shapley values at x and x' will be arbitrarily small for all features i.e. $|\phi_{x,i}(f) - \phi_{x',i}(f)|$ is small $\forall i \in D$. This corresponds to the following,

$$|\phi_{x,i}(f) - \phi_{x',i}(f)|^2 = |\langle f, \phi_{x,i} - \phi_{x',i} \rangle|^2 \leq ||f||_{\mathcal{H}_k}^2 ||\phi_{x,i} - \phi_{x',i}||_{\mathcal{H}_k}^2 \tag{5.5}$$

where we use Cauchy-Schwarz for the last line. Therefore, for a given f with fix RKHS norm, the key to show robustness lies into bounding the Shapley functionals. In the following theorem, we make two assumptions: (1) the base kernels $k^{(i)}$ for each dimension $i \in D$ are bounded, and (2) the (population) conditional mean embedding functions $\mu_{X_{S^c}|X_S}$ belong to the vector-valued RKHSs $\mathcal{H}_{\Gamma_{X_S}}$ for all coalitions $S \subseteq D$, therefore have finite norms. This assumption is also adopted in Park and Muandet [2020, Theorem 4.5].

Theorem 5.3.5 (Bounding Shapley functionals). *Let k be a product kernel with d bounded kernels $|k^{(i)}(x,x)| \leq M$ for all $i \in D$. Denote $M_\mu := \sup_{S \subseteq D} M^{|S|}$, $M_\Gamma := \sup_{S \subseteq D} ||\mu_{X_{S^c}|X_S}||^2_{\Gamma_{X_S}}$ and $L_\delta = \sup_{S \subseteq D} ||\psi_{x_S} - \psi_{x'_S}||^2_{\mathcal{H}_k}$. Let $\delta > 0$, assume $|x^{(i)} - x^{(i)'}|^2 \leq \delta$ for all features $i \in D$, then differences of the Interventional and Observational Shapley functionals for feature i at observation x, x' can be bounded as $||\phi^{(I)}_{x,i} - \phi^{(I)}_{x',i}||^2_{\mathcal{H}_k} \leq 2M_\mu L_\delta$ and $||\phi^{(O)}_{x,i} - \phi^{(O)}_{x',i}||^2_{\mathcal{H}_k} \leq 4M_\Gamma M_\mu L_\delta$. If k is the RBF kernel with lengthscale l, then*

$$||\phi^{(I)}_{x,i} - \phi^{(I)}_{x',i}||^2_{\mathcal{H}_k} \leq 4(1 - \exp(-d\delta/2l^2)), \qquad ||\phi^{(O)}_{x,i} - \phi^{(O)}_{x',i}||^2_{\mathcal{H}_k} \leq 8M_\Gamma(1 - \exp(-d\delta/2l^2))$$

Therefore, as long as $||f||_{\mathcal{H}_k}$ is small, RKHS-SHAP will return robust Shapley values with respect to small perturbations. Notice the Shapley functionals do not depend on f and can be estimated separately purely based on data. We will show in the next section how this key property allows us to use the functional itself to aid in learning of f. This enables us to enforce particular structural constraints on f via an additional regularisation term.

5.4 Shapley regularisation

Regularisation is popular in machine learning because it allows inductive bias to be injected to learn functions with specific properties. For example, classical L_1 and L_2 regularisers are used to control the sparsity and smoothness of model parameters. Manifold regularisation [Belkin et al., 2006], on the other hand, exploits the geometry of the distribution of unlabelled data to improve learning in a semi-supervised setting, whereas Pérez-Suay et al. [2017] and Li et al. [2019b] adopted a kernel dependence regulariser to learn functions for fair regression and fair dimensionality reduction. In the following, we propose a new *Shapley regulariser* based on the Shapley functionals, which allows learning while controlling the level of specific features' contributions to the model.

Formulation Let A be a specific feature whose contribution we wish to regularise, f the function we wish to learn, and $\phi_{x_i,A}(f)$ the Shapley value of A at a given observation x_i. Our goal is to penalise the mean squared magnitude of $\{\phi_{x_i,A}(f)\}_{i=1}^n$ in the ERM framework, which corresponds to $\min_{f \in \mathcal{H}_k} \sum_{i=1}^n \ell(y_i, f(x_i)) + \lambda_f ||f||^2_{\mathcal{H}_k} + \frac{\lambda_S}{n} \sum_{i=1}^n |\phi_{x_i,A}(f)|^2$, where ℓ is some loss function and λ_f and λ_S control the level of regularisations. By penalising the mean squared magnitude of the Shapley values correspond to feature A, we hope to learn a function that uses less information from A. If we replace the population Shapley functional with the finite sample estimate from Prop. 5.3.1, and utilise the

Representer theorem, we can rewrite the optimisation in terms of $\boldsymbol{\alpha}$,

Proposition 5.4.1 (Shapley regularisation with ERM)**.** *The above optimisation can be rewritten as,*

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \sum_{i=1}^{n} \ell(y_i, \mathbf{K}_{x_i\mathbf{x}}\boldsymbol{\alpha}) + \lambda_f \boldsymbol{\alpha}^\top \mathbf{K}_{\mathbf{xx}}\boldsymbol{\alpha} + \frac{\lambda_S}{n}\boldsymbol{\alpha}^\top \zeta_A \zeta_A^\top \boldsymbol{\alpha}$$

*. To regularise the Interventional SVs (*ISV-REG*) of A, we set $\zeta_A = \frac{1}{J}\sum_{j=1}^{J} \mathcal{K}^{(I)}_{\mathbf{x},S_j\cup A} - \mathcal{K}^{(I)}_{\mathbf{x},S_j}$ where S_j's are coalitions sampled from $p_{SV}(S) = \frac{1}{d}\binom{d-1}{|S|}^{-1}$. For regularising Observational SVs (*OSV-REG*), we set $\zeta_A = \frac{1}{J}\sum_{j=1}^{J} \mathcal{K}^{(O)}_{\mathbf{x},S_j\cup A} - \mathcal{K}^{(O)}_{\mathbf{x},S_j}$.*

In particular, closed form optimal dual weights $\boldsymbol{\alpha} = (\mathbf{K}^2_{\mathbf{xx}} + \lambda_f \mathbf{K}_{\mathbf{xx}} + \frac{\lambda_S}{n}\zeta_A\zeta_A^\top)^{-1}\mathbf{K}_{\mathbf{xx}}\mathbf{y}$ can be recovered when ℓ is the squared loss.

Choice of regularisation. Similar to the feature attribution problem, *the choice of regularising against ISVs or OSVs is application dependent* and boils down to whether one wants to take the correlation of A with other features into account or not.

ISV-REG ISV-REG can be used to protect the model when covariate shift of variable A is expected to happen at test time and one wishes to downscale A's contribution during training instead of completely removing this (potentially useful) feature. Such situation may arise if, e.g., a different measurement equipment or process is used for collecting observations of A during test time. ISV is well suited for this problem as dependencies across features will be broken by the covariate shift at test time.

OSV-REG On the other hand, OSV-REG can find its application in fair learning – learning a function that is fair with respect to some sensitive feature A. There exist a variety of fairness notions one could consider, such as, e.g. *Statistical Parity*, *Equality of Opportunity* and *Equalised Odds* [Corbett-Davies and Goel, 2018]. In particular, we consider the fairness notion recently explored in the literature [Jain et al., 2020, Mase et al., 2021] that uses Shapley values, which are becoming a bridge between Explainable AI and fairness, given that they can detect biased explanations from biased models. In particular, Jain et al. [2020] illustrated that if a model is fair against a sensitive feature A, A should have neither a positive nor negative contribution towards the prediction. This corresponds to A having SVs with negligible magnitudes. Simply removing A from the training doesn't make the model fair, as contributions of A might enter the model via correlated features, therefore it is important to take feature correlations into account while regularising. Hence, it is natural to deploy OSV-REG for fair learning.

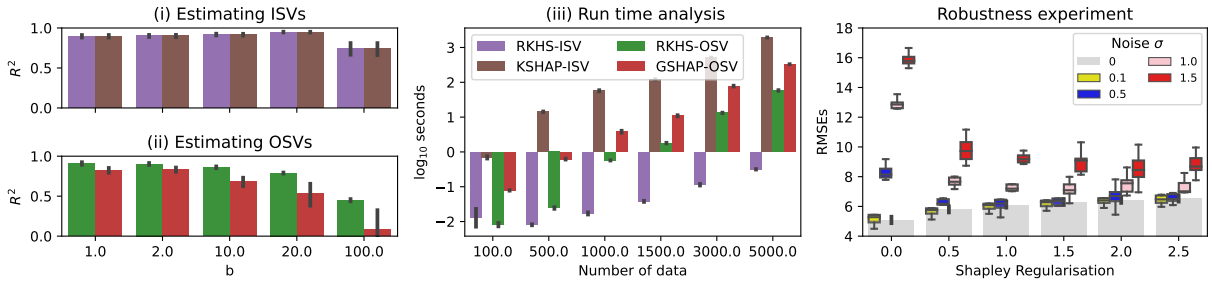


Figure 5.2: (Left) Estimation of Shapley values using data from the Banana distribution. (Mid) Run time analysis in log scale is reported. (Right) For ISV-REG, RMSEs of f_{reg} on noisy test data at different noise level σ'. All scores are averaged over 10 runs and 1sd is reported.

5.5 Experiments

We demonstrate specific properties of RKHS-SHAP and Shapley regularisers using four synthetic experiments, because these properties are best illustrated under a fully controlled environment. For example, to highlight the merit of distributional-assumption-free value function estimation in RKHS-SHAP, we need groundtruth conditional expectations of value functions for verification, but they are not available in real-world data because we do not observe the true data generating distribution. Nonetheless, as model interpretability is a practical problem, we have also ran several larger scales ($n = 50000, 1.8 \times 10^6$) real-world explanation tasks using RKHS-SHAP and reported our findings in Appendix C.5 for a complete empirical demonstration. All code and implementations are made publicly available here.

In the first two experiments, we evaluate RKHS-SHAP methods against benchmarks on estimating Interventional and Observational SVs on a Banana-shaped distribution with nonlinear dependencies [Sejdinovic et al., 2014]. The setup allows us to obtain closed-form expressions for the ground truth ISVs and OSVs, yet the conditional distributions among features are challenging to estimate using any standard parametric density estimation methods. We also present a run time analysis to demonstrate empirically that mean embedding-based approaches are significantly more efficient than sampling-based approaches. Finally, the last two experiments are applications of Shapley regularisers in robust modeling to covariate shifts and fair learning with respect to a sensitive feature.

In the following, we denote RKHS-OSV and RKHS-ISV as the OSV and ISV obtained from RKHS-SHAP. As a benchmark, we implement the model agnostic sampling-based algorithm KERNELSHAP from the Python package **shap** [Lundberg and Lee, 2017]. We denote the ISV obtained from KERNELSHAP as KSHAP-ISV. As **shap** does not offer a model-agnostic OSV algorithm, we implement the approach from Aas et al. [2019], where OSVs are estimated using Monte Carlo samples from fitted multivariate Gaussians. We denote this approach as GSHAP-OSV. We fit a kernel ridge regression on each of our experiments. Lengthscales of the kernel are selected using median heuristic and regularisation parameters

are selected using cross-validation. Further implementation details and real-world data illustrations are included in Appendix C.5.

5.5.1 RKHS-SHAP experiments

Experiment 1: Estimating Shapley values from Banana data. We consider the following 2d-Banana distribution $\mathcal{B}(b^{-1}, v)$ from Sejdinovic et al. [2014]: Sample $Z \sim N(0, \text{diag}(v, 1))$ and transform the data by setting $X_1 = Z_1$ and $X_2 = b^{-1}(Z_1^2 - v) + Z_2$. Regression labels are obtained from $f_{\text{truth}}(X) = b^{-1}(X_1^2 - v) + X_2$. This formulation allows us to compute the true ISVs and OSVs in closed forms, i.e $\phi^{(I)}_{X,1}(f_{\text{truth}}) = b^{-1}(X_1^2 - v)$, $\phi^{(I)}_{X,2}(f_{\text{truth}}) = X_2$, $\phi^{(O)}_{X,1}(f_{\text{truth}}) = \frac{1}{2}(3b^{-1}(X_1^2 - v) - X_2)$ and $\phi^{(O)}_{X,2}(f_{\text{truth}}) = \frac{1}{2}(3X_2 - b^{-1}(X_1^2 - v))$. In the following we will simulate 3000 data points from $\mathcal{B}(b^{-1}, 10)$ with $b \in [1, 10, 20, 50, 100]$, where smaller values of b correspond to more nonlinearly elongated distributions. We choose R^2 as our metric since the true Shapley values for each experiment are scaled according to b. Figure 5.2 (left) demonstrate the R^2 scores of estimated ISVs and OSVs in contrast with groundtruths SVs across different configurations. We see that RKHS-ISV and KSHAP-ISV give exactly the same R^2 scores across configurations. This is not surprising as the two methods are mathematically equivalent. While in KSHAP-ISV one averages over evaluated $\{f(x'_j)\}$ with x'_j being the imputed data, RKHS-ISV aggregated feature maps of the imputed data first before evaluating at f, i.e $\sum_{j=1} f(x'_j) = \langle f, \sum_{j=1} \phi(x'_j)\rangle_{\mathcal{H}_k} = \langle f, \hat{\mu}_X\rangle_{\mathcal{H}_k}$. However, it is this subtle difference in the order of operations contributes to a significant computational speed difference as we later show in Experiment 2. In the case of estimating OSVs, we see RKHS-OSV is consistently better than GSHAP-OSV at all configurations. This highlights the merit of RKHS-OSV as no density estimation is needed, thus avoiding any potential distribution model misspecification which happens in GSHAP-OSV.

Experiment 2: Run time analysis. In this experiment we sample n data points from $\mathcal{B}(1, 10)$ where $n \in [100, 500, 1000, 1500, 3000, 5000]$ and record the $\log_{10}$ seconds required to complete each algorithm. In practice, as the software documentation of **shap** suggests, one is encouraged to subsample their data before passing to the KERNELSHAP algorithm as the background sampling distribution to avoid slow run time. As this approach speeds up computation at the expense of estimation accuracy since fewer data is used, for a fair comparison with our RKHS-SHAP method which utilises all data, we pass the whole training set to the KERNELSHAP algorithm. Figure 5.2 (mid) illustrates the run time across methods. We note that the difference in runtime between the two sampling based methods KSHAP-ISV and GSHAP-OSV can be attributed to a different software implementation, but we observe that they are both significantly slower than RKHS-ISV and RKHS-OSV. RKHS-OSV is slower than RKHS-ISV as it involves matrix inversion when computing the empirical CME. In practice, one can trivially subsample data for

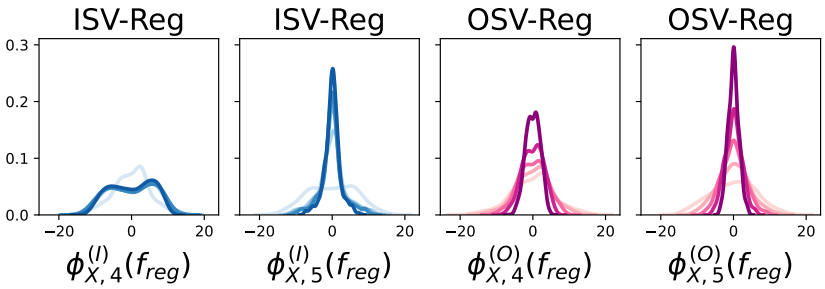


Figure 5.3: Distributions of SVs of sensitive feature X_5 and correlated feature X_4 obtained from ISV-REG and OSV-REG at different regularisation parameters. Colour intensity represents the strength of regularisation.

RKHS-SHAP to achieve further speedups like in the **shap** package, but one can also deploy a range of kernel approximation techniques as discussed in Section 5.2.2.

5.5.2 Shapley regularisation experiments

For the last two experiments we will simulate 3000 samples from $X \sim N(0, \Sigma)$ with $\text{diag}(\Sigma) = \mathbf{1_5}$ and $\Sigma_{4,5} = \Sigma_{5,4} = 0.9$, 0 otherwise, therefore feature X_4 and X_5 will be highly correlated. We set our regression labels as $f_{\text{true}}(x) = x^\top \beta$ with $\beta = [1, 2, 3, 4, 10]$, enforcing X_5 to be the most influential feature. We use 70% of our data for training and 30% for testing.

Experiment 3: Protection against covariate shift using ISV-REG. For this experiment, we inject extra mean zero Gaussian noise to the most influential feature X_5 in the testing set, i.e. $X_5' = X_5 + \sigma' N(0, 1)$ for $\sigma' \in [0, 0.1, 0.5, 1, 1.5]$. We assume that there is an expectation for covariate shift in X_5 to occur at test time, due to e.g. a change in the measurement precision – hence, we train our model f_{reg} using ISV-REG at different regularisation level λ_s for $\lambda_s \in [0, 0.5, 1, 1.5, 2, 2.5]$. We then compare RMSEs when no covariate shift is present ($\sigma' = 0$) against RMSEs at different noise levels. The results are shown in Figure 5.2 (right). We see that when no regularisation is applied, RMSEs increase rapidly as σ' increases, indicating our standard unprotected kernel ridge regressor is sensitive to noises from X_5'. As the Shapley regularisation parameter increases, the RMSE of the noiseless case gradually increases too, but RMSEs of the noisy data are much closer to the noiseless case, exhibiting robustness to the covariate shift.

Experiment 4: Fair learning with OSV-REG At last, we demonstrate the use of Shapley regulariser to enable fair learning. In this context, as we will see, OSV-REG is the appropriate regulariser. Consider X_5 as some sensitive feature which we would like to minimise its contribution during the learning of f. Recall X_4 is highly correlated to X_5 so it contains sensitive information from X_5 as well. Figure 5.3 demonstrates how distributions of ISVs and OSVs of X_4 and X_5 changes as λ_s increases. As regularisation increases,

the SVs of X_5 becomes more centered at 0, indicating lesser contribution to the model f_{reg}. Similar behavior can be seen from the distribution of $\phi^{(O)}_{X,4}(f_{\text{reg}})$ but not from $\phi^{(I)}_{X,4}$. This illustrates how ISV-REG will propagate unfairness through correlated feature X_4 while OSV-REG can take them into account by minimising the contribution of sensitive information during learning.

5.6 Conclusion, limitations, and future directions

In this work, we proposed a more accurate and more efficient algorithm to compute Shapley values for kernel methods, termed RKHS-SHAP. We proved that the corresponding local attributions are robust to local perturbations under mild assumptions, a desirable property for consistent model interpretation. Furthermore, we proposed the Shapley regulariser which allows learning while controlling specific feature contribution to the model. We suggested two applications of this regulariser and concluded our work with synthetic experiments demonstrating specific aspects of our contributions. Extensive real-world data explanations are provided in Appendix C.5.2 for empirical demonstration.

While our methods currently only are applicable to functions arising from kernel methods, a fruitful direction would be to extend the applicability to more general models using the same paradigm. It would also be interesting to extend our formulation to kernel-based hypothesis testing, and for example, to interpret results from two-sample tests.

6 | Learning Inconsistent Preferences with Gaussian Processes

This chapter is based on the following published paper:

> **Siu Lun Chau***, Javier Gonzalez, and Dino Sejdinovic. "Learning inconsistent preferences with Gaussian Processes" *The 25th International Conference on Artificial Intelligence and Statistics (AISTATS 2022)*

Abstract

We revisit widely used *preferential Gaussian processes* (PGP) by Chu and Ghahramani [2005] and challenge their modelling assumption that imposes rankability of data items via latent utility function values. We propose a generalisation of PGP which can capture more expressive latent preferential structures in the data and thus be used to model inconsistent preferences, i.e. where transitivity is violated, or to discover clusters of comparable items via spectral decomposition of the learned preference functions. We also consider the properties of associated covariance kernel functions and its reproducing kernel Hilbert Space (RKHS), giving a simple construction that satisfies universality in the space of preference functions. Finally, we provide an extensive set of numerical experiments on simulated and real-world datasets showcasing the competitiveness of our proposed method with state-of-the-art. Our experimental findings support the conjecture that violations of rankability are ubiquitous in real-world preferential data.

6.1 Introduction

Data concerning user preferences for items or services is ubiquitous and is often used to detect patterns in user behaviour and to make recommendations. Moreover, these user preferences are often relative (i.e. based on recording choices between a pair of competing items) and may involve an abundance of ranking inconsistencies, e.g. preference of A over B, B over C, but C over A – sometimes called a *rock-paper-scissors relation*, and reported, e.g. in mating strategies of certain species [Sinervo and Lively, 1996]. Situation like this arises in many domains and is an example of the Condorcet Paradox extensively investigated in social choice theory [Gehrlein, 1983]. Such inconsistencies may arise due to latent structures determining the criteria for preferences, where different item features may be relevant for making each of these three choices. As an example, consider the case where a cue is present in an item description for C, which may be relevant for its comparison to A but not for its comparison to B, and that this cue changes the user's criterion when making the choice. Motivated by such inconsistent preferences, we will propose a Gaussian process (GP) model which can capture such latent structures by seamlessly incorporating all the available context information, i.e. sets of item covariates.

Our main contributions can be summarised as follows:

1. We propose a simple generalisation of PGP by Chu and Ghahramani [2005], allowing to model preferences that do not conform to a consistent ranking. Our method can be integrated directly into many existing probabilistic preference learning algorithms in fields such as rank aggregation [Simpson and Gurevych, 2020], Bayesian optimisation [González et al., 2017], duelling bandits [Zoghi et al., 2015], recommender systems [Nguyen et al., 2014] and reinforcement learning [Zintgraf et al., 2018].

2. The proposed *Generalised Preferential Gaussian Processes* (GPGP) use *Generalised Preferential Kernels* – we give a simple construction of these kernels which we prove to satisfy the appropriate notion of universality, i.e. the corresponding RKHS is rich enough to approximate any bounded continuous skew-symmetric function arbitrarily well. While a weaker form of this result has previously appeared in Waegeman et al. [2012], our proof uses different techniques, building on c_0-universality notions as developed by Sriperumbudur et al. [2011], allowing for more general domains like $\mathbb{R}^d$.

3. We extend ideas from partial ranking [Cheng et al., 2012] and propose a spectral decomposition method to extract *clusters of comparable items* from preferential data using GPGP. This allows us to extract interpretable substructures from a complex network of preferential relationships.

The chapter is outlined as follows: in section 6.2, we outline the problem and overview related work. In section 3, we introduce GPGP, describe universality of the corresponding kernel function and how GPGP can be used to uncover clusters of comparable items. Section 4 provides extensive experiments on synthetic and real-world data. Our results improve performance over PGP on all real-world datasets, giving further evidence for the ubiquity of inconsistent preferences. We conclude in section 5.

6.2 Background

Let $\mathcal{X}$ be the data domain where we choose data to compare. The well-established paradigm in this context is *preference learning* (PL), which is concerned with predicting and modeling an order relation on a collection of data items [Fürnkranz and Hüllermeier, 2010]. Typical PL models [Chu and Ghahramani, 2005, d'Aspremont et al., 2019, González et al., 2017, Houlsby et al., 2012] assume that there is a latent *utility function* $f : \mathcal{X} \to \mathbb{R}$ to be optimised. We may observe noisy evaluations of f in forms such as item ratings or rankings, but in many cases, explicit direct feedback from f is scarce or expensive and the quantity of implicit feedback data typically far outweighs the explicit data. Moreover, when the feedback comes from human users, they are better at evaluating relative differences than absolute quantities [Kahneman and Tversky, 1979], and in the absence of a reference point explicit feedback may be unreliable and its scale may be ambiguous or difficult to determine. This motivates us to consider the situation where the feedback consists of *binary preferences*, which we denote as *duelling* data. Formally, a pair of items $(x, x') \in \mathcal{X} \times \mathcal{X}$ is presented to the user and we observe a binary outcome that tells us whether x or x' won the duel. For simplicity, we will assume here that no draws are allowed.

Binary preference data are often represented as Directed Aclycic Graphs (DAGs), where items are denoted as nodes and an edge from node $x \to x'$ implies that x won the duel over x' [Pahikkala et al., 2009]. As a result, preference learning can often be seen as learning on DAGs. For example, PageRank [Page et al., 1998] can be seen as an Eigenvector centrality measure on a preference graph. For the rest of the chapter, we will use the term preference graph and preferential data interchangeably.

One simple model for the duelling feedback is given by

$$p(y|(x, x')) = \sigma(yg(x, x')), \quad y \in \{-1, +1\} \tag{6.1}$$

for some $g : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, logistic function $\sigma(t) = \frac{1}{1+e^{-t}}$ and $y = +1$ denoting that x is preferred over x'. We note that g must be skew-symmetric, i.e. $g(x, x') = -g(x', x)$ to satisfy the natural condition $p\left(y|(x, x')\right) + p\left(y|(x', x)\right) = 1$ since there are only two outcomes allowed: a win for x or a win for x'. Considering general relations on pairs of items in $\mathcal{X} \times \mathcal{X}$, Pahikkala et al. [2010] term relations which

satisfy skew-symmetry *reciprocal*.

An instance of model (6.1) is *Preferential Gaussian Process* (PGP) introduced by Chu and Ghahramani [2005]. It is assumed therein that g imposes *rankability* on $\mathcal{X}$. If we define $x \preceq x' \iff g(x, x') \leq 0$, then $\preceq$ is a total order on all of $\mathcal{X}$. This corresponds to writing $g(x, x') = f(x) - f(x')$, where f is the utility function which is determined up to a global shift. Pahikkala et al. [2010] consider a similar notion of a reciprocal relation and term it *weakly ranking representable* when such f exists. In the PGP model, a GP prior is imposed on latent f and the likelihood for a given observation $(x_i, x_j, y_{i,j})$ now becomes

$$p(y_{i,j}|(x_i, x_j)) = \sigma\left(\left(f(x_i) - f(x_j)\right) y_{i,j}\right). \tag{6.2}$$

Inference on f can then proceed similarly as in GP classification, using methods such as Laplace approximation [Williams and Rasmussen, 2006, Section 3.4] or variational methods [Hensman et al., 2015].

A multitude of probabilistic PL algorithms are developed based on PGP. An extension of the model to predict crowd preferences is introduced by Simpson and Gurevych [2020], where a low-rank structure is imposed on the crowd preference matrix and each component is modelled using a GP. On the other hand, González et al. [2017] developed preferential Bayesian optimisation to optimise black-box functions where queries only come in the form of duels. Houlsby et al. [2012] incorporated PGP with unsupervised dimensionality reduction for multi-user recommendation systems. Under a similar setting, Nguyen et al. [2014] applied PGP into a GP factorisation machines to model context-aware recommendations. PGP is also used in the field of reinforcement learning to provide preference elicitation strategies for supporting multi-objective decision making [Zintgraf et al., 2018]. Finally, one can directly incorporate the learned preference function into learning to rank problems [Ailon and Mohri, 2010]. All models mentioned above assume the data to be perfectly rankable and this is the assumption we challenge in this chapter.

Other preference learning models also typically assume data to be rankable and that a well defined utility function exists. Classical examples are *random utility model* [Thurstone, 1994], Bradley-Terry-Luce models [Bradley and Terry, 1952, Luce, 1959], the Thurstone-Mosteller model [Mosteller and Nogee, 1951] and many of their variants. Non-probabilistic preference models such as SVM-Rank [Joachims, 2009], Serial-Rank [Fogel et al., 2016], Sync-Rank [Cucuringu, 2016] and SVD-Rank [d'Aspremont et al., 2019] also typically assume rankability in their formulations.

In practice however, total rankability is often too strong of an assumption. There might be many reasons why some "noisy" preferences do not conform to a single overall ranking. For example, it is well studied that cognitive biases often lead to inconsistent human preferences in behavioral economics [Tversky

and Kahneman, 1992]. In fact, not until very recently did the ranking community start to challenge this assumption by proposing quantitative metrics on measuring rankability of duelling data: Anderson et al. [2019], Cameron et al. [2020] considered rankability as a metric measuring the difference between the observed preference graph and a perfectly rankable complete dominance graph. This motivates the need to consider a general preference modelling methods without assuming total rankability.

To relax rankability assumptions and thus capture more complex latent structures in preferential data, we will consider a Gaussian process formulation for a general case where no single order can be formed and it is, in particular, possible that transitivity is violated, i.e. $x \preceq x', x' \preceq x''$ but $x'' \preceq x$. We believe that in many cases, such inconsistent relationships are fundamental to the data generating process. In fact, this conjecture is supported by the findings of Zoghi et al. [2015] who consider discrete choice (duelling bandits) problem with the application in ranker evaluation for information retrieval. They concluded that the instances where the Condorcet winner (an item which beats all the others with probability larger than $\frac{1}{2}$) does not exist far outweigh those where it does. Since the existence of a single objective function f with a unique global maximum would imply the existence of the Condorcet winner, we see that inconsistent preferences may, in fact, be prevalent in practice.

A thread of important related work arises in the inference of general (i.e. not necessarily preferential) relations between pairs of data objects [Pahikkala et al., 2010, Waegeman et al., 2012] using *frequentist kernel methods*. In particular, Pahikkala et al. [2010] similarly emphasise the importance of being able to model *intransitive* reciprocal relationships, motivating it using sports games examples. They also introduce the same kernel function we will consider in this work. Waegeman et al. [2012] take this work further, consider more general graded relations, reiterating importance of intransitivity, and study the connections to fuzzy set theory. Waegeman et al. [2012] also prove the theoretical result which is a slightly weaker form of our Theorem 6.3.2 on universality. As such, we emphasise that the generalised preferential kernels we will consider are not new, but to the best of our knowledge they have not been used in Gaussian process modelling, nor in discovering richer latent structure behind preferential data, which we propose in this work. There is also work that considers intransitive relations using different types of statistical models – without using item covariates and operating only on the matrix of match outcomes. For example, Causeur and Husson [2005] extend the classical Bradley-Terry model, while Chen and Joachims [2016] introduce so called Blade-Chest model and discover that substantial intransitivity exists in contexts such as online video gaming data.

We will in this chapter deliberately adopt both Bayesian and frequentist viewpoints to kernel methods. We consider and implement a new Gaussian process framework, generalising PGP of Chu and Ghahramani [2005] which can hence be integrated in many probabilistic preference learning algorithms that build on

PGP. But we also study the properties of the RKHSs associated to the corresponding kernel functions, arriving at conclusions essentially equivalent to those in Pahikkala et al. [2010], Waegeman et al. [2012], although we use different proof techniques which are more grounded in the notions of RKHS universality developed by Sriperumbudur et al. [2011], allowing us to consider more general spaces $\mathcal{X}$ of item covariates. We note that GPs and RKHSs have deep connections, as described in Kanagawa et al. [2018].

6.3 Methodology

6.3.1 Generalised preferential kernels

Recall that in PGP we express the preference function $g(x, x')$ as $f(x) - f(x')$ and place a GP prior on f. In fact, one can recast the inference solely in terms of g as f directly induces a GP prior on g by linearity. The corresponding covariance kernel k_E^0 is then given by

$$\begin{aligned} k_E^0((u,u'),(v,v')) &= \operatorname{cov}(f(u) - f(u'), f(v) - f(v')) \\ &= k(u,v) + k(u',v') - k(u,v') - k(u',v), \end{aligned} \tag{6.3}$$

where the base kernel k is the covariance structure on f. Houlsby et al. [2012] called k_E^0 the *preference kernel*. This reformulation allows us to directly apply many state-of-the-art GP classification methods.

Now consider a more general case where $g : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ corresponds to any skew-symmetric function. We will consider the following skew-symmetric kernel:

$$k_E((u,u'),(v,v')) = k(u,v)k(u',v') - k(u,v')k(u',v), \tag{6.4}$$

termed *Generalised Preferential Kernel* and the corresponding GP will be called the *Generalised Preferential Gaussian Processes* (GPGP).

The kernel (6.4) is not new and was previously studied by Pahikkala et al. [2010], Waegeman et al. [2012] in their work on intransitive relations, as well as in persistent homology analysis to enforce appropriate symmetry conditions [Kwitt et al., 2015, Reininghaus et al., 2015]. In particular, Pahikkala et al. [2010] take a feature mapping ψ on $\mathcal{X} \times \mathcal{X}$ and "skew-symmetrise" it in the following way: $\varphi(x, x') = \psi(x, x') - \psi(x', x)$. Now φ and the corresponding kernel can be used to model skew-symmetric functions and, thus, reciprocal relations. In case where ψ corresponds to the Kronecker product kernel $k \otimes k$, this results exactly in (6.4). We give some further details of the feature map view of these kernels in the Appendix.

One can interpret both k_E^0 and k_E as kernels between edges in a preference graph. k_E can be extended further to tackle more complex preferential data settings such as *learning from crowd preferences* and *preference learning from distributional data*. We will keep the exposition here simple and a further description of these extensions is included in the Appendix.

For any kernel function κ, denote its RKHS by $\mathcal{H}_\kappa$. $\mathcal{H}_{k_E}$ is clearly more expressive than $\mathcal{H}_{k_E^0}$ as it imposes no rankability assumption on its elements. We next consider how expressive $\mathcal{H}_{k_E}$ is, given suitable regularity conditions on $\mathcal{X}$ and k. In particular, for *any* skew-symmetric bounded continuous function g on $\mathcal{X} \times \mathcal{X}$, can one find a function in $\mathcal{H}_{k_E}$ that arbitrarily well approximates g? We define a suitable notion of ss-c_0-universality below which allows for a very general domain $\mathcal{X}$. There are different notions of universality for kernels and we refer the reader to Micchelli et al. [2006], Sriperumbudur et al. [2011] and references therein for further details.

Definition 6.3.1 (ss-c_0-universality)**.** *Let $\mathcal{X}$ be a locally compact Hausdorff space and let $C_{0,ss}(\mathcal{X} \times \mathcal{X})$ be the space of functions $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which are continuous, bounded, skew-symmetric and vanish at infinity. A kernel k is said to be ss-c_0-universal on $\mathcal{X} \times \mathcal{X}$ if and only if $\mathcal{H}_k$ is dense in $C_{0,ss}(\mathcal{X} \times \mathcal{X})$ w.r.t. the uniform norm.*

We next prove a theorem which allows us to easily construct ss-c_0-universal kernels k_E by simply selecting k to be c_0-universal [Sriperumbudur et al., 2011]. We note that a weaker form of this result was first proved in [Waegeman et al., 2012, Theorem III.4] using different techniques. Our proof (included in Appendix) builds on the notion of c_0-universality and its relationship with integrally strictly positive definite kernels developed by Sriperumbudur et al. [2011], making the construction applicable to any *locally compact Hausdorff space* $\mathcal{X}$, whereas Waegeman et al. [2012] require *compact metric spaces*, thereby excluding interesting domains such as $\mathbb{R}^d$ or infinite discrete spaces.

Theorem 6.3.2 (ss-c_0-universality of k_E)**.** *Assume that the base kernel k is c_0-universal on the locally compact Hausdorff space $\mathcal{X}$. Then the generalised preferential kernel $k_E((u,u'),(v,v')) = k(u,v)k(u',v') - k(u,v')k(u',v)$ is ss-c_0-universal on $\mathcal{X} \times \mathcal{X}$.*

6.3.2 Clusters of comparable items

Clustering is a popular method to consider latent structures behind preferential data. Many existing methods [Cao et al., 2012, Li et al., 2018, Grbovic et al., 2013, Fogel et al., 2016] cluster items based on their similarity devised from the outcomes of matches. For example, in Fogel et al. [2016] the authors used a two-hop aggregation method on the preference graph to compute the similarity between two items,

i.e. $S_{i,j} = \sum_{k=1}^{n} y_{i,k} y_{j,k}$. In this work, we consider a different notion of clustering for preferential data, which we term *clusters of comparable items*. In particular, we are interested in discovering groups of items that are comparable and thus rankable within clusters but not across. Cases like this might arise when the pairwise comparison is defined indirectly. For example, product preferences are often deduced using product search histories in e-commerce [Karmaker Santu et al., 2017] and products may not always belong to the same categories. A related problem is studied in partial rankings [Cheng et al., 2012], where certain pairs of items can be declared as incomparable by thresholding the probabilities of pairwise preferences between items. In contrast to partial rankings though, we do not need to consider individual probabilities, and by clustering the items, all pairings across clusters are declared as incomparable.

Consider a latent preference function g and assume that it belongs to $\mathcal{H}_{k_E}$. We can associate to g a skew-symmetric Hilbert-Schmidt operator $S_g : \mathcal{H}_k \to \mathcal{H}_k$ which satisfies

$$\left\langle k\left(\cdot, x\right), S_g k\left(\cdot, x'\right)\right\rangle_{\mathcal{H}_k} = g\left(x, x'\right). \tag{6.5}$$

For example, if $g(x, x') = f(x) - f(x')$ then S_g is a rank two operator given by $S_g = f \otimes e - e \otimes f$ and $e(x) = 1$ is the constant function. Conversely, if S_g has rank two and one of its top singular functions is constant, a total order can be imposed on $\mathcal{X}$ by the non-constant top singular function. Similar reasoning can also be applied to the match outcomes matrix directly and is the core idea behind SVD-based approaches to ranking [d'Aspremont et al., 2019, Chau et al., 2020].

In general, however, S_g may have a higher rank. Specifically, in the case of the existence of L clusters of comparable items, S_g can be written as an operator of rank $2L$ given by

$$S_g = \sum_{l=1}^{L} (f_l \otimes e_l - e_l \otimes f_l) \tag{6.6}$$

where f_l is the utility function of the l-th cluster and e_l is the l-th cluster indicator function, i.e. it equals to 1 if item x belongs to cluster l, and 0 otherwise.

We are now interested in extracting clusters of comparable items from a fitted function g. Assuming (6.6), the true complete preference matrix G with $G_{i,j} = g(x_i, x_j)$ satisfies

$$G = \sum_{l=1}^{L} (\mathbf{f}_l \mathbf{1}_l^\top - \mathbf{1}_l \mathbf{f}_l^\top). \tag{6.7}$$

$\mathbf{f}_l$ is the vector of evaluations of the l-th cluster utility function f_l and $\mathbf{1}_l$ is the l-th cluster indicator vector,

i.e. its j-th entry equals to 1 if item x_j belongs to cluster l, and 0 otherwise.

To recover the clusters, we first estimate the preference matrix $\hat{G}$ using GPGP and treat it as a noisy version of the true low rank matrix G. The clusters can then be recovered by applying standard clustering algorithms (e.g. K-means) to the data representation given by the top $2L$ singular vectors from $\hat{G}$, analogously to classical spectral clustering.

6.3.3 Data augmentation baseline

It is simple to extend any classification algorithm to model skew-symmetric duelling preferences using data augmentation, without assuming rankability. One example is to take an observation (x_i, x_j, y_{ij}) of the match between x_i and x_j, and concatenate the two sets of item covariates in two different orders, as $x_{i,j} = [x_i, x_j]$ and $x_{j,i} = [x_j, x_i]$ and pass them to a classification model with both $x_{i,j}$, $x_{j,i}$ as inputs and $y_{i,j}$ and $y_{j,i} = -y_{i,j}$ as their respective targets. While such data augmentation does encourage skew-symmetry, the resulting function is not guaranteed to be skew-symmetric on all inputs. Skew-symmetry can then be enforced by averaging the model outputs:

$$\hat{p}(y_{ij} = 1|(x_i, x_j)) \quad = \quad \frac{1}{2}\hat{p}_{\text{cat}}(y_{ij} = 1|x_{i,j}) + \frac{1}{2}\hat{p}_{\text{cat}}(y_{ji} = -1|x_{j,i}), \tag{6.8}$$

where $\hat{p}_{\text{cat}}$ are the probabilities fitted on the concatenated item covariates. Although this ad-hoc augmentation allows us to relax the rankability assumption in preference learning and is applicable to any models, including GPs, its theoretical justification is questionable, and the additional computational cost due to doubling the data size may be problematic.

We note that another approach applicable to linear models would be to impose skew-symmetry via model coefficients directly, but it is not clear how one might extend it to nonparametric methods such as GPs. We provide further discussion of this line of reasoning with its connection to the feature maps of PGP and GPGP in the Appendix.

6.3.4 Scalability

Since GPGP is formulated on the joint item space $\mathcal{X} \times \mathcal{X}$ of pairs of items, computational considerations need to be taken into account. In the worst case scenario, we may be storing and inverting a $\binom{n}{2} \times \binom{n}{2}$ kernel matrix for n items, if a match is played between every pair of items. This seldom happens in practice, however. In fact, most real-world comparison data is highly sparse, especially if the number of items n is large. Nonetheless, there are a large number of well established ways to scale up GPs that can be readily applied to GPGP, e.g. variational inducing points [Hensman et al., 2015] or conjugate gradient

methods [Filippone and Engler, 2015]. In addition, Gardner et al. [2018] proposed techniques to reduce the asymptotic complexity of exact GP inference from cubic to quadratic. One can also use methods such as KISS-GP [Wilson and Nickisch, 2015] exploiting Kronecker and Toeplitz algebra for further speedups. Kronecker structure of kernel matrices, as well as conjugate gradient methods were also exploited by Pahikkala et al. [2013] in the context of regularized least squares with generalised preferential kernel.

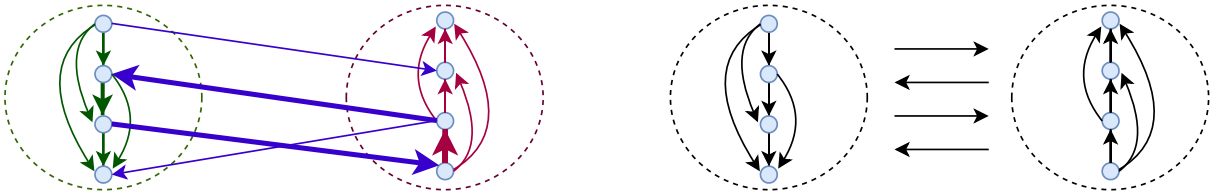
6.4 Experiments

Our experiments demonstrate the key aspect of GPGP: the ability to model cyclic and inconsistent preferences from duelling data. In section 6.4.1, we study the robustness of GPGP using simulated preferences with different levels of sparsity and inconsistencies. Section 6.4.2 studies the problem of clusters of comparable items using simulation to further showcase how GPGP can learn complex preferential structures. Finally, we conclude the experiments by testing GPGP against alternative preference prediction methods using 4 real-world datasets with a total of 22 examples. As baselines, we compare GPGP with *Preferential GP* (PGP), *GP with data augmentation* (PAIR-GP) and *Logistic Regression with data augmentation* (PAIR-LOGREG). The latter two baselines use a scheme described in 6.3.3. For all methods involving kernels, we use the Gaussian radial basis function kernel (RBF) $k(x, x') = \exp\big(-\frac{||x-x'||^2}{2\gamma^2}\big)$ and obtain lengthscale γ by optimising the evidence lower bound. We use Laplace approximation and conjugate gradient methods for inference in GPGP, PGP and PAIR-GP.

6.4.1 Simulation: Cyclic and inconsistent preferences

Data generation Consider a comparison network with n items and a covariate matrix $X \in \mathbb{R}^{n\times p}$. We assign to each node a latent state $z \in \{1, .., L\}$ and generate a set of utility functions $\{r_{z,z'}\}_{z,z'=1}^{L}$, i.e. there is a different utility function for each pair (z, z') of latent states. We let $r_{z,z'}(x) = \sum_{j=1}^{n} \alpha_j^{z,z'} k(x, x_j)$ with each vector $\alpha^{z,z'} \overset{i.i.d}{\sim} N(0, I_n)$. Comparison between node i and j is then conducted based on the utility selected by their latent states, i.e. $i \preceq j \iff r_{z_i,z_j}(x_i) < r_{z_i,z_j}(x_j)$. This setup brings in cyclic and inconsistent preferences to the overall preference graph. Figure 6.1a provides a visual illustration of the experiment with $L = 2$ with a cycle indicated in bold. Different colour of the edges indicates that a different criterion, i.e. utility function, is used in pairwise comparisons.

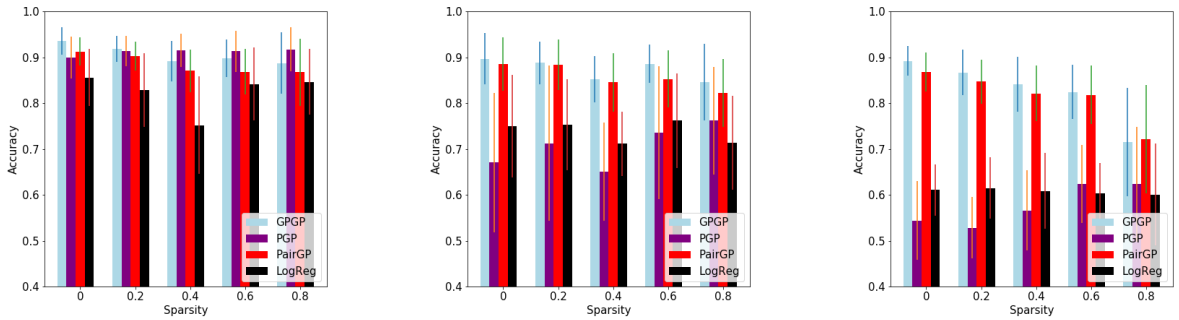
We simulate a preference graph with $n = 30$ players each containing $p = 5$ covariates with different level of graph sparsity and number of latent states ($L = 1, 2, 5$). Latent states are simulated uniformly. Item features are generated conditionally on latent states with $x|z \sim N(z\mathbf{1}, I_5)$, thus allowing the features to encapsulate information about the latent states. We do a $70 - 30$ train-test-split on the data and repeat the experiments 20 times.



(a) Cyclic and Inconsistent Preference illustration

(b) Clusters of comparable items

Figure 6.1: (left): Items belongs to different groups and preference between items corresponds to the utility function determined by their latent states (different colors indicate that different utility function is used). Overall preference exhibit cycles (indicated in bold). (right) Items belongs to different groups and items are rankable within the groups, but preferences across groups are random.



(a) L = 1 (Perfectly Rankable)

(b) L = 2

(c) L = 5

Figure 6.2: Comparisons of algorithms for simulations at different sparsity and inconsistency level. Accuracies are averaged over 20 runs and error bars of 1 standard deviation are provided.

Results Figure 6.2 gives the accuracy of GPGP when predicting preferences on held-out data in comparison with baselines. As L increases, we see a significant decrease in accuracy for PGP and PAIR-LOGREG whereas GPGP and PAIR-GP performed relatively stable. On average GPGP outperforms the other methods, except in the high sparsity regime with $L = 1$, where PGP performed better. In fact, this is not surprising as $L = 1$ corresponds to a perfectly rankable duelling problem since there is only one utility function.

6.4.2 Simulation: Clusters of comparable items

Data generation Similar to the setup from section 6.4.1, we assign to each data a latent state and match outcomes follow utility functions dependant on these states. However, when comparisons are made across latent groups, the outcome is a Bernoulli$(1/2)$, independent of all else, due to items being non-comparable. See Figure 6.1b for a visual illustration. We simulate matches between 30 players each containing 5 features with different level of sparsity and number of latent clusters $L = 2, 3$.

We give three possible approaches of finding the clusters of comparable items,

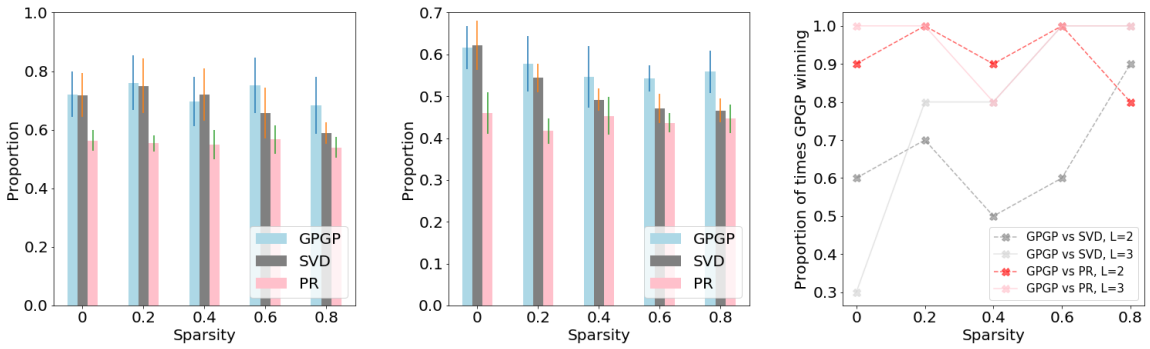
1. GPGP-CLUS: First recover the latent preference matrix G using GPGP, then run KMeans on the top

$2L$ corresponding singular vectors of G.

2. PR-CLUS: First apply the partial ranking with abstention method from Cheng et al. [2012] to remove non-comparable matches. SVD and KMeans are then applied to the trimmed comparison matrix.

3. SVD-CLUS: Apply KMeans to the data representation given by the top 2L singular vectors from the comparison graph directly.

We report the proportion of items which are correctly clustered as a metric of performance. We do not include PGP-CLUS here because PGP performs poorly when there are multiple ranking signals.



(a) L = 2 (b) L = 3 (c) Comparing GPGP with baselines

Figure 6.3: (a, b) Comparisons of algorithms for simulations with different number of clusters and sparsity level. Proportion of items correctly clustered are averaged over 20 runs and error bars of 1 standard deviation are provided. (c) Proportion of times GPGP performed better than baselines.

Results Figure 6.3 gives the performance of the methods in recovering clusters of comparable items, comparing the proportion of the items each method clustered correctly. On average GPGP-CLUS performed better than the rest, except at low sparsity, i.e. dense graphs, where it performed similarly to SVD-CLUS. This is expected as for a highly dense preference graph, modelling with GPGP will not gain further additional information about the overall preference structure. On the other hand, PR-CLUS performed consistently poorly because it assumes rankability of the data. In other words, it only removes matches that agree with the sole ranking signal the algorithm recovered.

6.4.3 Predicting preferences on real data

We apply GPGP and baselines to a variety of real-world comparison graphs, and measure outcome by their accuracy in predicting preferences on the test set. A 70-30 train-test split is applied to the data over 20 trials. Table 6.1 summarises the test results on 4 datasets for preference learning. We report the average network clustering coefficient C_{avg} [Saramäki et al., 2007] as a proxy to illustrate how non-rankable the problem is.

Table 6.1: Test results on the 4 datasets for preference learning. Accuracy averaged over 20 trials is reported along with its standard deviation. C_{avg} is the average clustering coefficient of a comparison graph. The symbol * indicates when the algorithm's accuracy is significantly worse than that of GPGP. Wilcoxon rank-sum test with level 0.05 was used to determine the statistical significance.

				Accuracy (%)			
DATASET	# Item	# Edge	C_{avg}	GPGP	PGP	PAIRGP	PAIRLOGREG
Chameleon	35	104	0.33	0.78 ± 0.06	$0.51 \pm 0.09^*$	$0.72 \pm 0.08^*$	$0.71 \pm 0.08^*$
Flatlizard	77	100	0.07	0.83 ± 0.06	0.80 ± 0.09	$0.78 \pm 0.07^*$	$0.77 \pm 0.09^*$
NFL 2000-18	32	213x19 yrs	0.54	0.59 ± 0.03	$0.51 \pm 0.02^*$	0.58 ± 0.03	0.65 ± 0.03
ArXiv Graph	1025	1000	0.11	0.74 ± 0.02	$0.66 \pm 0.03^*$	$0.70 \pm 0.02^*$	$0.62 \pm 0.02^*$

Male Cape Dwarf Chameleons Contest This data is used in the study by Stuart-Fox et al. [2006b]. Physical measurements are made on 35 male Cape dwarf chameleons, and the results of 104 contests are recorded. From Table 6.1, we see that GPGP statistically outperformed all baselines. In particular, PGP was the worst performer due to the moderately high clustering coefficient.

Flatlizard Competition The data is collected at Augrabies Falls National Park (South Africa) in September-October 2002 [Whiting et al., 2009], on the contest performance and background attributes of 77 male flat lizards (*Platysaurus Broadleyi*). The results of 100 contests were recorded, along with 18 physical measurements made on each lizard, such as *weight* and *head size*. This comparison graph has the lowest average clustering coefficient thus is the most rankable compared to the rest. On average GPGP still performed better than PGP but the difference is not statistically significant.

NFL Football 2000-2018 The data contains the outcome of National Football League (NFL) matches during the regular season, for the years 2000 - 2018 [1]. In addition, 256 matches per year between 32 teams, along with 18 performance metrics, such as *yards per game* and *number of fumbles* are recorded. We pick the top 5 informative features by applying the BAHSIC feature selection algorithm [Song et al., 2012] and run the algorithm on each year's comparison graph separately and average the results. In this highly non-rankable ($C_{avg} = 0.54$) problem, PAIR-LOGREG outperformed the rest. This is not surprising as the features (e.g. *yards per game*) are expected to be linearly related to the match outcome and a linear model may thus better capture these relationships. Nonetheless, GPGP still outperformed PGP.

ArXiv Citation Network The last dataset we use is from the Open Graph Benchmark [Hu et al., 2020] arXiv Computer Science papers citation network. Each paper represents a node and an edge from node $i \to j$ means paper i cited paper j. We pick an induced subgraph with 1025 nodes and 1000 edges from the full network. Each node contains a 128-dimensional feature vector obtained by averaging the embedding of words in its title and abstract. Again, we see GPGP performed significantly better than the

[1] data collected from nfl.com

other algorithms. It is interesting to note that PAIR-LOGREG was the worst performing method, indicating that word-embedding features, in contrast to the features from the NFL problem, have a highly non-linear relationship with the match outcome.

6.5 Conclusion and Discussion

We proposed *Generalised Preferential Gaussian Processes* (GPGP), a new probabilistic model for preferential data. GPGP relaxed the rankability assumption and comes with a strong theoretical justification in terms of universality of the corresponding kernel function. It can be readily integrated into many existing preference learning algorithms that are based on PGP. Experimental results on simulations and real-world datasets show superior performance in comparison to PGP, the latter demonstrating the prevalence of inconsistent preferences and the need for relaxing the rankability assumptions in practice. We demonstrated how GPGP can be used to solve a specific problem that goes beyond rankability, i.e. recovering clusters of comparable items. A number of other problems which similarly involve more complex preferential structures can be studied based on the proposed framework.

Relaxing rankability allows to investigate latent structures influencing preferences, including the case where preferences are inconsistent, cyclical or when many items are simply not comparable to each other. Building on the existing preferential Gaussian Process (PGP) model, our approach introduces additional flexibility but preserves the advantages of having a Bayesian probabilistic model and faithful uncertainty quantification. The algorithms we proposed may enable more robust and customised recommendations to users in recommender systems and information retrieval. It is also envisaged that our work will find applications in A/B testing, gaming systems, and Bayesian optimisation with implicit or relative feedback.

Digital trails such as web searches and purchase patterns are often collected for targeted recommendations. It is worth noting that these features might include sensitive personal information and utilising them without careful consideration might be unethical. Therefore, an important practical research direction will be to consider combining GPGP with algorithmic fairness approaches applicable to kernel methods and GPs [Li et al., 2019b], or to use differentially private mechanisms for GPs [Smith et al., 2018].

7 | Explaining Preference Models with Shapley Values

This chapter is based on the following paper,

> Robert Hu*, **Siu Lun Chau***, Jaime Ferrando Huertas, and Dino Sejdinovic. "Explaining Preferences with Shapley Values" Advances in Neural Information Processing Systems (NeurIPS), 2022

Abstract

While preference modelling is becoming one of the pillars of machine learning, the problem of preference explanation remains challenging and underexplored. In this chapter, we propose PREF-SHAP, a Shapley value-based model explanation framework for pairwise comparison data. We derive the appropriate value functions for preference models and further extend the framework to model and explain *context specific* information, such as the surface type in a tennis game. To demonstrate the utility of PREF-SHAP, we apply our method to a variety of synthetic and real-world datasets and show that richer and more insightful explanations can be obtained over the baseline.

7.1 Introduction

Preference learning [Fürnkranz and Hüllermeier, 2003] is a classical problem in machine learning, where one is interested in learning the order relations on a collection of data items. Preference learning algorithms [Bradley and Terry, 1952, Thurstone, 1994, Chu and Ghahramani, 2005, González et al., 2017] often assume that there is a latent utility function $f : \mathcal{X} \mapsto \mathbb{R}$ dictating the outcome of preferences, where $\mathcal{X}$ denotes the domain of item covariates. An explicit feedback such as item ratings or rankings from recommender systems can be treated as noisy evaluations of f, whereas pairwise comparison data (also known as duelling data) arising from, e.g., sports match outcomes [Cattelan et al., 2013, Chau et al., 2020] can be used to implicitly infer f, i.e. item $\mathbf{x}^{(\ell)}$ is preferred over (beats) item $\mathbf{x}^{(r)}$ when $f(\mathbf{x}^{(\ell)}) > f(\mathbf{x}^{(r)})$. As shown by Kahneman and Tversky [1979], humans often struggle with evaluating absolute quantities when it comes to eliciting preferences, but are broadly capable of evaluating relative differences, a core observation often exploited in preference learning. Motivated by such, this work will focus on explaining preferences inferred using duelling data.

Explaining preference models is crucial when they are applied in areas such as recommendation systems [Houlsby et al., 2012], finance [Bennett et al., 2022], and sports science [Stuart-Fox et al., 2006b] for the practitioner to trust, debug and understand the value of their findings [Chau et al., 2021b]. However, despite its importance, no prior work has studied this problem to the best of our knowledge. While one may suggest applying existing explainability tools such as LIME [Ribeiro et al., 2016], or SHAP [Lundberg and Lee, 2017] to a learned utility function f, we reason that this approach only explains the utility but not the mechanism of eliciting preferences itself. We highlight the important differences between these two viewpoints in our numerical experiments. Moreover, the utility-based model places a strong *rankability* assumption on the underlying preferences, meaning that if we define $\mathbf{x}^{(\ell)} \preceq \mathbf{x}^{(r)} \iff f(\mathbf{x}^{(\ell)}) \leq f(\mathbf{x}^{(r)})$, then $\preceq$ is a total order on all the items. However, as Pahikkala et al. [2010] and Chau et al. [2022] have discussed, there are many departures from rankability in practice, e.g. we might easily see a preference of A over B, B over C, but C over A – conforming to the *rock-paper-scissors* relation. Such inconsistent preferences are under frequent study in social choice theory [List, 2022, Gehrlein, 1983], and are of wider interest in both healthcare Tsopra et al. [2018] and retail Feng et al. [2021] where data are both large and noisy.

To move beyond the rankability assumption, we will utilise the *Generalised Preferential Kernel* from Chau et al. [2022] to model the underlying preferences, and develop PREF-SHAP, a novel Shapley value [Shapley, 1953]-based explainability toolbox, to explain the inferred preferences. Our contributions can be summarised as follows:

1. We propose PREF-SHAP, a novel Shapley value-based explainability algorithm, to explain preferences based on duelling data.
2. We empirically demonstrate that PREF-SHAP gives more informative explanations compared to the naive approach of applying SHAP to the inferred utility function f.
3. We release a high-performant implementation of PREF-SHAP at Hu and Chau.

7.2 Background materials

We will first give a brief overview of preference learning and Shapley Additive Explanations (SHAP) [Lundberg and Lee, 2017], which are the two core concepts of our contribution, PREF-SHAP, described in Section 7.3.

Notation. Scalars are denoted by lower case letters, while vectors and matrices are denoted by bold lower case and upper case letters, respectively. Random variables are denoted by upper case letters. $\mathcal{X} \subseteq \mathbb{R}^d$ denotes the item space with d features and $\mathcal{Y} = \{-1, 1\}$ is the binary preference outcome space[1]. We let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function and $\mathcal{H}_k$ the corresponding reproducing kernel Hilbert space (RKHS).

7.2.1 Preference Learning

In this section, we will introduce the two approaches to model preferences from duelling data, namely the *utility based approach* and the more general approach from Chau et al. [2022]. Formally, a preference feedback is denoted as *duelling*, when a pair of items $(\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}) \in \mathcal{X} \times \mathcal{X}$ is given to a user, and a binary outcome $y \in \mathcal{Y}$ telling us whether $\mathbf{x}^{(\ell)}$ or $\mathbf{x}^{(r)}$ won the duel, is observed. In general, we observe m binary preferences among n items, giving the data $D = \left(\mathbf{y}, \mathbf{X}^{(\ell)}, \mathbf{X}^{(r)}\right) = \left\{\left(y_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}\right)\right\}_{j=1}^{m}$. We also use $\mathbf{X} \in \mathbb{R}^{n \times d}$ to denote the full item covariate matrix.

Utility-based Preference model (UPM) The following likelihood model is often used [Bradley and Terry, 1952, Thurstone, 1994, Chu and Ghahramani, 2005, González et al., 2017, Chau et al., 2020] to model duelling feedback using a latent utility function f:

$$p\left(y \mid \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) = \sigma\left(y\left(f\left(\mathbf{x}^{(\ell)}\right) - f\left(\mathbf{x}^{(r)}\right)\right)\right), \tag{7.1}$$

[1] Thus, we do not model 'draws' in match outcomes, but the model can be straightforwardly extended to include them by specifying the appropriate likelihood function.

where σ is the logistic CDF, i.e. $\sigma(z) = (1 + \exp(-z))^{-1}$. Maximum likelihood approaches are then deployed to learn the latent utility function f. Consequently, preferences between items can be inferred accordingly from $\mathbf{f} = \{f(\mathbf{x}_i)\}_{i=1}^n$, i.e. $\mathbf{x}_i$ is on average preferred over $\mathbf{x}_j$ if $\mathbf{f}_i \geq \mathbf{f}_j$.

Albeit elegant, there are several drawbacks to this approach in modelling preferences. As mentioned, using a one-dimensional vector $\mathbf{f}$ to derive preferences assumes that the items $\{\mathbf{x}_i\}_{i=1}^n$ are perfectly rankable, i.e. there is a total ordering on $\mathcal{X}$ which the true preferences are consistent with. This is a strong assumption that often does not hold in practice. For example, it is well studied that cognitive biases often lead to inconsistent human preferences in behavioural economics [Kahneman and Tversky, 1979]. Moreover, the ranking community has also challenged this assumption by devising rankability metrics [Anderson et al., 2019, Cameron et al., 2020] to test this restrictive assumption in practice.

Generalised Preference Model (GPM) Chau et al. [2022] proposed to model preference directly using a more general $g : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that captures the preference within any pair of items, using the likelihood

$$p\left(y \mid \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) = \sigma\left(yg\left(\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right)\right). \tag{7.2}$$

We note that g has to be a skew-symmetric function to ensure the natural property $p(y \mid \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}) = 1 - p(y \mid \mathbf{x}^{(r)}, \mathbf{x}^{(\ell)})$. The utility based approach can be obtained as a special case of this model, i.e. by setting $g(\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}) = f(\mathbf{x}^{(\ell)}) - f(\mathbf{x}^{(r)})$. We propose that when one is interested in modelling (and thus explaining) pairwise preferences, we should consider the preference function g directly instead of explaining preferences based on a restrictive utility model f.

We follow Chau et al. [2022]'s approach to model g non-parametrically using kernel methods [Paulsen and Raghupathi, 2016]. We assume g as a function lives in the following RKHS of skew-symmetric functions: given kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined on the item space $\mathcal{X}$, the *generalised preferential kernel* k_E on $\mathcal{X} \times \mathcal{X}$ is constructed as follows:

$$k_E\left(\left(\mathbf{x}_i^{(\ell)}, \mathbf{x}_i^{(r)}\right), \left(\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}\right)\right) = k\left(\mathbf{x}_i^{(\ell)}, \mathbf{x}_j^{(\ell)}\right) k\left(\mathbf{x}_i^{(r)}, \mathbf{x}_j^{(r)}\right) - k\left(\mathbf{x}_i^{(\ell)}, \mathbf{x}_j^{(r)}\right) k\left(\mathbf{x}_i^{(r)}, \mathbf{x}_j^{(\ell)}\right).$$

This kernel allows us to model the similarity across pairs of items. Moreover, if k is a universal kernel Sriperumbudur et al. [2011], then k_E also satisfies the corresponding notion of universality, meaning that the corresponding RKHS $\mathcal{H}_{k_E}$ is rich enough to approximate any bounded continuous skew-symmetric function arbitrarily well [Chau et al., 2022, Theorem. 1]. To infer $g \in \mathcal{H}_{k_E}$ using likelihood (7.2), one simply runs kernel logistic regression with data $\mathbf{y}$ as labels and $\left(\mathbf{X}^{(\ell)}, \mathbf{X}^{(r)}\right)$ as inputs. We will refer to this approach as the *Generalised Preference Model* (GPM).

We emphasize that *explaining* GPM allows us to specifically explain *inconsistent preferences*, which in contrast to *explaining rank* allows us to infer preferences even when transitivity is violated. Such insights can be of great importance in broader contexts such as decision theory [Anand, 1987] and utility theory [Aumann, 1964] where transitivity does not hold.

Incorporating context variables. Besides item-level covariates $\mathbf{x} \in \mathcal{X}$, when there exist additional *context covariates* $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^{d'}$ that describe the context in which a specific pairwise comparison is made, they can be incorporated into the kernel design as discussed in Chau et al. [2022, Appendix. B]. Examples of such context covariates could be court type when a tennis match is conducted, or where a different user compares two clothing items in e-commerce. Considering the enriched dataset $D = \left\{\left(y_j, \mathbf{u}_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}\right)\right\}_{j=1}^{m}$, we can now model the preference incorporating the context as: $p(y \mid \mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}) = \sigma\left(g_U\left(\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right)\right)$. Now, given a kernel k_U defined on the context space $\mathcal{U}$, the context-specific preference function $g_U : \mathcal{U} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ can be learnt non-parametrically with the following kernel,

$$k_E^{(U)}\left(\left(\mathbf{u}_i, \mathbf{x}_i^{(\ell)}, \mathbf{x}_i^{(r)}\right), \left(\mathbf{u}_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}\right)\right) = k_U\left(\mathbf{u}_i, \mathbf{u}_j\right) k_E\left(\left(\mathbf{x}_i^{(\ell)}, \mathbf{x}_i^{(r)}\right), \left(\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}\right)\right).$$

We refer to this approach as the *Context-specific Generalised Preference Model* (C-GPM).

7.2.2 Shapley Additive Explanations (SHAP)

To explain preferences, we will utilise the popular SHAP (SHapley Additive exPlanations) paradigm, which is based on the concept of Shapley values (SV). SV [Shapley, 1953] were originally proposed as a credit allocation scheme for a group of d players in the context of cooperative games, which are characterised by a value function $\nu : [0,1]^d \to \mathbb{R}$ that measures *utility* of subsets of players. Formally, the Shapley value for player j in game ν is defined as:

$$\phi_j(\nu) = \sum_{S \subseteq \Omega \setminus \{j\}} \left(|S|!(d - |S| - 1)!/d!\right)\left(\nu(S \cup j) - \nu(S)\right), \tag{7.3}$$

where $\Omega = \{1, ..., d\}$ is the set of players of the game. Given a value function ν, the Shapley values are proven to be the only credit allocation scheme that satisfies a particular set of favourable and fair game theoretical axioms, commonly known as *efficiency*, *null player property*, *symmetry* and *additivity* [Shapley, 1953]. Štrumbelj and Kononenko [2014] later connect Shapley values to the field of *explainable machine learning* by drawing an analogy between model fitting and cooperative game. Given a specific data point,

by considering its *features* as *players* participating in a game that measures features' utilities, the Shapley values obtained can be treated as *local feature importance scores*. Such games are typically defined through the value functions defined below.

Definition 7.2.1 (Value functions)**.** *Let X be a random variable on $\mathcal{X} \subseteq \mathbb{R}^d$ and $f : \mathcal{X} \to \mathbb{R}$ a model from hypothesis space $\mathcal{H}$. The value function $\nu : \mathcal{X} \times [0,1]^d \times \mathcal{H}$ is given by*

$$\nu_{\mathbf{x},S}(f) = \mathbb{E}_{r(X_{S^c}|X_S=\mathbf{x}_S)}\left[f(\{X_S, X_{S^c}\}) \mid X_S = \mathbf{x}_S\right] \tag{7.4}$$

where r is an appropriate reference distribution, X_S is the subvector of X corresponding to the feature set S, S^c is the complement of the feature set S and $\{X_S, X_{S^c}\} = X$ denotes the concatenation of X_S and X_{S^c}.

In other words, given a data point $\mathbf{x}$, the utility of the feature subset S is defined as the impact on the model prediction, after "removing" the contribution from S^c via integration with respect to the reference distribution r. These "removal-based" strategies are common in the explainability literature [Covert et al., 2021]. Nonetheless, the correct choice of the reference distribution has been a long-standing debate [Chen et al., 2020]. Janzing et al. [2020] argued from a causality perspective that the feature marginal distribution should be used as the reference distribution, i.e. $r(X_{S^c} \mid X_S = x_S) = p(X_{S^c})$ where p is the data distribution. On the other hand, Frye et al. [2020] disagreed by pointing out these "marginal" value functions ignore feature correlations and lead to unintelligible explanations in higher-dimensional data, and they instead advocate the use of conditional distribution as reference, i.e. $r(X_{S^c} \mid X_S = x_S) = p(X_{S^c} \mid X_S = x_S)$. Thus, there is no consensus and in fact, Chen et al. [2020] took a neutral stand and argued the choice depends on the application at hand. This also leads to design of value functions for specific problems, e.g. improving local estimation [Ghalebikesabi et al., 2021], incorporating causal knowledge [Frye et al., 2019, Heskes et al., 2020] and modelling structured data [Duval and Malliaros, 2021]. In this chapter, we will design an appropriate value function for preference learning and show that naive application of the existing value function to preference learning will lead to unintuitive results.

Shapley value estimation. Given a data point $\mathbf{x}$ and a model f, estimating Shapley values consist of two main steps: Firstly, for each feature subset $S \subseteq \Omega$, estimate the value function $\nu_{\mathbf{x},f}(S)$ either by Monte Carlo sampling from the reference distributions r, or by utilising a model specific structure to speed up the estimation such as in LINEARSHAP [Štrumbelj and Kononenko, 2014], DEEPSHAP [Lundberg and Lee, 2017], TREESHAP [Lundberg et al., 2018], and RKHS-SHAP [Chau et al., 2021b]. The former

sampling procedure is straightforward when r is the marginal distribution, but computationally heavy and difficult when r is the conditional distribution, as it involves estimating an exponential number of conditional densities Yeh et al. [2022]. Finally, after estimating the value functions, one can compute the Shapley values based on Eq. 7.3 or by utilising the efficient weighted least square approach proposed by Lundberg and Lee [2017].

Estimating value functions when $f \in \mathcal{H}_k$. We give a review to the recently introduced RKHS-SHAP algorithm proposed by Chau et al. [2021b] as it is another core component for PREF-SHAP. RKHS-SHAP is a SV estimation method for functions in a given RKHS. It circumvents the need for any density estimation and utilises the arsenal of kernel mean embeddings [Muandet et al., 2016a] to estimate the value functions non-parametrically. Assume k takes a product kernel structure across dimensions, then for any $f \in \mathcal{H}_k$, by applying the *reproducing property* [Paulsen and Raghupathi, 2016], the value function can be decomposed as:

$$\nu_{\mathbf{x},S}(f) = \left\langle f,\ \mathbb{E}_{r(X_{S^c}|X_S=\mathbf{x}_S)}\left[k\left(\{X_S, X_{S^c}\}, \cdot\right) \mid X_S = \mathbf{x}_S\right]\right\rangle_{\mathcal{H}_k} \tag{7.5}$$

$$= \left\langle f, k_{X_S} \otimes \mu_{r(X_{S^c}|X_S=\mathbf{x}_S)}\right\rangle_{\mathcal{H}_k}, \tag{7.6}$$

where k_{X_S} is the product of kernels belonging to the feature set S, and $\mu_{r(X_{S^c}|X_S=\mathbf{x}_S)} := \int k_{X_{S^c}} r(X_{S^c} \mid X_S = \mathbf{x}_S) dX_{S^c}$ is the kernel mean embedding [Muandet et al., 2016a] of the reference distribution r. Depending on the choice of the reference distribution, one recovers either the standard kernel mean embedding or the conditional mean embedding. This allows us to arrive at a closed form expression of the value function and circumvents the need for fitting an exponential number of conditional densities.

7.3 Proposed method: PREF-SHAP

In this section, we will present PREF-SHAP, a new Shapley explainability toolbox designed to explain preferences by attributing contribution scores over item-level and context-level covariates for our preference models. Recall the likelihood model for C-GPM from Chapter. 6:

$$p\left(y \mid \mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) = \sigma\left(y g_U\left(\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right)\right), \tag{7.7}$$

where g_U is the context-included preference function that denotes the strength of preference of item $\mathbf{x}^{(\ell)}$ over item $\mathbf{x}^{(r)}$ under context $\mathbf{u}$. As there are two distinct sets of covariates present, we will propose two different value functions to capture the influences from items and context variables respectively, and show how they could be estimated non-parametrically using tools from the kernel methods literature, as in

RKHS-SHAP.

7.3.1 Preferential value function for items

To explain a general preference model $g : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, we propose the following *preferential value function for items*.

Definition 7.3.1 (Preferential value function for items)**.** *Given a preference function $g \in \mathcal{H}$, a pair of items $\left(\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) \in \mathcal{X} \times \mathcal{X}$ to compare, we define the preferential value function for items as $\nu^{(p_I)} : \mathcal{X} \times \mathcal{X} \times [0,1]^d \times \mathcal{H} \to \mathbb{R}$ such that:*

$$\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(g) = \mathbb{E}_q\left[g(\{X_S^{(\ell)}, X_{S^c}^{(\ell)}\}, \{X_S^{(r)}, X_{S^c}^{(r)}\}) \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)}, X_S^{(r)} = \mathbf{x}_S^{(r)}\right] \tag{7.8}$$

where expectation is taken over the reference $q\left(X_{S^c}^{(\ell)}, X_{S^c}^{(r)} \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)}, X_S^{(r)} = \mathbf{x}_S^{(r)}\right)$.

We note that $\nu^{(p_I)}$ is also applicable to the context-specific preference models. For example, applying $\nu^{(p_I)}$ to $g_{\mathbf{u}} := g_U(\mathbf{u}, \cdot, \cdot)$ allows one to quantify the item covariate's influences under a specific context $\mathbf{u}$, while applying $\nu^{(p_I)}$ to $\bar{g} := \mathbb{E}_{p(U)}[g_U(U, \cdot, \cdot)]$ quantifies the average influence from each of the item covariates instead.

Similar to standard value functions, the influence of a feature set S shared by the items $\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$ is measured as the impact on the preference model after "removing" contributions from features in S^c, via integration with respect to some reference distribution r. Similar to g, this value function is skew-symmetric in its first two arguments, i.e. $\nu^{(p_I)}(\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}, S, g) = -\nu^{(p_I)}(\mathbf{x}^{(r)}, \mathbf{x}^{(\ell)}, S, g)$. This is justified, since features that "encourage" preference of $\mathbf{x}^{(\ell)}$ over $\mathbf{x}^{(r)}$ should naturally be the ones that "discourage" preference of $\mathbf{x}^{(r)}$ over $\mathbf{x}^{(\ell)}$ to ensure consistency. In this chapter, we assume the items are i.i.d sampled from some distribution p, and we utilise the observational data distribution as reference as in Frye et al. [2020], i.e. we take $r\left(X_{S^c}^{(\ell)}, X_{S^c}^{(r)} \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)}, X_S^{(r)} = \mathbf{x}_S^{(r)}\right)$ to be $p\left(X_{S^c}^{(\ell)} \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)}\right) p\left(X_{S^c}^{(r)} \mid X_S^{(r)} = \mathbf{x}_S^{(r)}\right)$. Although we decide here to use the observational distribution as the reference, the corresponding estimation procedure follows analogously if one instead uses the marginal distribution approach in Janzing et al. [2020].

Problems with direct application of SHAP to preference model g A naive way of explaining with SHAP a general preference model g which assumes no rankability would require concatenation of the items' covariates. Namely, we would set $\mathbf{z} = (\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}) \in \mathbb{R}^{2d}$ and then apply SHAP to the function $g(\mathbf{z})$ directly, now giving $2d$ Shapley values for each observed preference, i.e. two Shapley values for

each feature. Not only does this approach require us to consider a larger number of feature coalitions during computation (squaring the original amount), but it also ignores that $\mathbf{x}^{(\ell)}$ and $\mathbf{x}^{(r)}$ in fact consist of the same features, leading to inconsistent explanations, i.e. that the same feature in $\mathbf{x}^{(\ell)}$ and $\mathbf{x}^{(r)}$ has a different influence, hence giving different explanations simply due to the ordering of items. We illustrate these pitfalls of such a naive approach in Appendix E.2.

Empirical estimation of the preferential value function $\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(g)$. While the *preferential value function* is general in the sense that it could be applied to any preference function g, we divert our attention to functions in $\mathcal{H}_{k_E}$, where k_E is the *generalised preferential kernel* introduced in Chapter. 6. This allows us to adapt the recently introduced RKHS-SHAP to our settings, and we can thus circumvent learning an exponential number of conditional densities as in Frye et al. [2020]. In the following segment, we prove the existence of the Riesz representation of the *preferential value functional*, a necessary step to adapt the RKHS-SHAP framework to our setting.

Proposition 7.3.2 (Preferential value functional for items)**.** *Let k be a product kernel on $\mathcal{X}$, i.e. $k(\mathbf{x}^{(\ell)},\mathbf{x}^{(r)}) = \prod_{j=1}^{d} k^{(j)}(x^{(j)},x'^{(j)})$. Assume $k^{(j)}$ are bounded for all j, then the Riesz representation of the functional $\nu^{(p)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}$ exists and takes the form:*

$$\nu^{(p)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S} = \frac{1}{\sqrt{2}}\Big(\mathcal{K}(\mathbf{x}^{(\ell)},S)\otimes\mathcal{K}(\mathbf{x}^{(r)},S) - \mathcal{K}(\mathbf{x}^{(r)},S)\otimes\mathcal{K}(\mathbf{x}^{(\ell)},S)\Big)$$

where $\mathcal{K}(\mathbf{x},S) = k_S(\cdot,\mathbf{x}_S)\otimes\mu_{X_{S^c}|X_S=\mathbf{x}_S}$ and $k_S(\cdot,\mathbf{x}_S) = \bigotimes_{j\in S} k^{(j)}(\cdot,x^{(j)})$ is the sub-product kernel defined analogously as X_S.

All proofs are included in the appendix. By representing the functionals as elements in the corresponding RKHS, we can now estimate the value function non-parametrically using kernel mean embeddings.

Proposition 7.3.3 (Non-parametric Estimation)**.** *Given $\hat{g} = \sum_{j=1}^{m}\alpha_j k_E((\mathbf{x}_j^{(\ell)},\mathbf{x}_j^{(r)}),\cdot)$, datasets $\mathbf{X}^{(\ell)},\mathbf{X}^{(r)}$, test items $\mathbf{x}^{(\ell)},\mathbf{x}^{(r)}$, the preferential value function at test items $\mathbf{x}^{(\ell)},\mathbf{x}^{(r)}$ for coalition S and preference function $\hat{g}$ can be estimated as*

$$\hat{\nu}^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(\hat{g}) = \boldsymbol{\alpha}^\top\Big(\Gamma(\mathbf{X}_S^{(\ell)},\mathbf{x}_S^{(\ell)})\odot\Gamma(\mathbf{X}_S^{(r)},\mathbf{x}_S^{(r)}) - \Gamma(\mathbf{X}_S^{(\ell)},\mathbf{x}_S^{(r)})\odot\Gamma(\mathbf{X}_S^{(r)},\mathbf{x}_S^{(\ell)})\Big),$$

where $\Gamma(\mathbf{X}_S^{(\ell)},\mathbf{x}_S^{(\ell)}) = \mathbf{K}_{\mathbf{X}_S^{(\ell)},\mathbf{x}_S^{(\ell)}}\odot\mathbf{K}_{\mathbf{X}_{S^c}^{(\ell)},\mathbf{X}_{S^c}}\mathbf{K}^{-1}_{\mathbf{X}_S,\lambda}\mathbf{K}_{\mathbf{X}_S,\mathbf{x}_S^{(\ell)}}$, $\mathbf{K}_{\mathbf{X}_S,\lambda} = \mathbf{K}_{\mathbf{X}_S,\mathbf{X}_S} + n\lambda I$, $\boldsymbol{\alpha} = \{\alpha_j\}_{j=1}^m$ and $\lambda > 0$ is a regularisation parameter.

Table 7.1: A summary of how our preference value functions can tackle different explanation tasks

Candidate	Explanation of interest	Value function	Preference function
$\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$	Which item features contributed most to this duel?	$\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}$	$g, \mathbb{E}_U[g_U(U,\cdot,\cdot)]$
$\mathbf{x}^{(\ell)}$	Which item features contributed most to $\mathbf{x}^{(\ell)}$'s matches?	$\frac{1}{n}\sum_{i=1}^{n} \nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}_i,S}$	$g, \mathbb{E}_U[g_U(U,\cdot,\cdot)]$
$\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$	Which context features contributed most to this duel?	$\nu^{(p_U)}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}$	g_U
$\mathbf{u}$	Which context features contributed most on average?	$\frac{1}{m}\sum_{j=1}^{m} \nu^{(p_U)}_{\mathbf{u},\mathbf{x}_j^{(\ell)},\mathbf{x}_j^{(r)},S'}$	g_U

7.3.2 Preferential value function for contexts

The influence an individual context feature in U has on a C-GPM function g_U can be measured by the following value function.

Proposition 7.3.4 (Preferential value function for contexts)**.** *Given a preference function $g_U \in \mathcal{H}_{k_E^U}$, denote $\Omega' = \{1, ..., d'\}$, then the utility of context features $S' \subseteq \Omega'$ on $\{\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\}$ is measured by $\nu^{(p_U)}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S'}(g_U) = \mathbb{E}[g_U\left(\{\mathbf{u}_S, U_{S^c}\}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) \mid U_S = \mathbf{u}_S]$ where the expectation is taken over the observational distribution of U. Now, given a test triplet $(\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)})$, if $\hat{g}_U = \sum_{j=1}^{m} \alpha_j k_E^U\left((\mathbf{u}_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot\right)$, the non-parametric estimator is:*

$$\hat{\nu}^{(p_U)}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S'}(\hat{g}_U) = \boldsymbol{\alpha}^\top \left(\left(\mathbf{K}_{\mathbf{U}_{S'},\mathbf{u}_{S'}} \odot \mathbf{K}_{\mathbf{U}_{S'^c},\mathbf{U}_{S'^c}} \left(\mathbf{K}_{\mathbf{U}_{S'},\mathbf{U}_{S'}} + m\lambda' I\right)^{-1} \mathbf{K}_{\mathbf{U}_{S'},\mathbf{u}_{S'}}\right) \odot \Xi_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)}}\right)$$

where $\Xi_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)}} = \left(\mathbf{K}_{\mathbf{X}^{(\ell)},\mathbf{x}^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}^{(r)},\mathbf{x}^{(r)}} - \mathbf{K}_{\mathbf{X}^{(r)},\mathbf{x}^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}^{(\ell)},\mathbf{x}^{(r)}}\right)$.

Analogously, the average influence of a specific context feature can be computed by taking an average over all pairs of matches, i.e. by using a modified value function $\frac{1}{m}\sum_{j=1}^{m} \nu^{(p_U)}_{\mathbf{u},\mathbf{x}_j^{(\ell)},\mathbf{x}_j^{(r)},S'}(\hat{g}_U)$. We summarise different ways to modify the proposed preferential value functions to interrogate the preference models in Table 7.1.

Computational complexity of PREF-SHAP When computing GPM, it is fundamentally a *kernel ridge regression* (KRR), which naively has complexity $\mathcal{O}(n^3)$. There exists a multitude of approximation techniques for KRR, the most common type being the Nystrom approximation [Drineas and Mahoney, 2005]. For all our experiments, we use FALKON [Meanti et al., 2020], a large-scale library for solving kernel logistic regression using preconditioned conjugate gradient descent and Nyström approximations. FALKON has a computational complexity of $\mathcal{O}(n\sqrt{n})$, which effectively becomes the complexity for GPM when using FALKON. As the value function for GPM requires estimating conditional mean embeddings, which in turn also are KRR's, one can appeal to FALKON again to reduce complexity to $\mathcal{O}(n\sqrt{n})$. We summarize the procedure of PREF-SHAP in Algorithm 1. We further detail computational

Algorithm 1 PREF-SHAP

Input: Solution $\boldsymbol{\alpha}$, datasets $\mathbf{X}^{(\ell)}, \mathbf{X}^{(r)}, \mathbf{X}, \mathbf{U}$, test items $\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}, \mathbf{u}$, batch size n_b, number of coalition samples n_S, context-specific flag `cflg`

Compute effective dimension d_{eff} := Number of features with variance greater than 0.

Compute coalitions $\mathcal{S} = \{S_1, \ldots, S_{n_S}\}$, form binary matrix $\mathbf{Z} \in \{0,1\}^{n_S, d_{\text{eff}}}$ from $\mathcal{S}$, and compute weights $\mathbf{W} = [w_1, \ldots, w_{n_S}]$ with $w_i = \frac{d-1}{\binom{d}{|S_i|}|S_i|(d-|S_i|)}$.

for batch $\mathcal{S}_b$ in $\mathcal{S}$ **do**

  if `cflg` Take S, S^c of $\mathbf{X}^{(\ell)}, \mathbf{X}^{(r)}, \mathbf{X}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$ **else** Take S', S'^c of $\mathbf{U}, \mathbf{u}$

  if `cflg` Compute $\mathbf{K}^{-1}_{\mathbf{X}_S,\lambda}[\mathbf{K}_{\mathbf{X}_S,\mathbf{x}_S^{(\ell)}}, \mathbf{K}_{\mathbf{X}_S,\mathbf{x}_S^{(r)}}]$ **else** $\left(\mathbf{K}_{\mathbf{U}_{S'},\mathbf{U}_{S'}} + m\lambda' I\right)^{-1} \mathbf{K}_{\mathbf{U}_{S'},\mathbf{u}_{S'}}$ using `BatchedCGD`

  if `cflg` Compute $\hat{\nu}^{(p)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},\mathcal{S}_b}(\hat{g})$ **else** $\hat{\nu}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},\mathcal{S}'_b}(\hat{g}_U)$

end for

if `cflg` Set $\mathbf{v}_x = \{\hat{\nu}^{(p)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},\mathcal{S}_b}(\hat{g})\}_{b=1}^B$ **else** $\mathbf{v}_x = \{\hat{\nu}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},\mathcal{S}'_b}(\hat{g}_U)\}_{b=1}^B$

Calculate Shapley values $\boldsymbol{\beta}_x = \left(\mathbf{Z}^\top \mathbf{W} \mathbf{Z}\right)^{-1} \mathbf{Z}^\top \mathbf{W} \mathbf{v}_x$

return $\boldsymbol{\beta}_x$

details pertaining to computing coalitions S and batched conjugate gradient descent (`BatchedCGD`) in Appendix E.1.

7.4 Experiments

The main focus of our experiments is to illustrate the difference between explaining GPM (PREF-SHAP) and applying SHAP to UPM, thus highlighting the difference in explaining the mechanism of eliciting preferences and explaining the utility. When we explain UPM, we first explain how items $\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$ affect their utilities $f(\mathbf{x}^{(\ell)}), f(\mathbf{x}^{(r)})$. Explaining the utility corresponds to calculating the value functions of the utilities $\nu_{\mathbf{x}^{(\ell)},S}(f)$ and $\nu_{\mathbf{x}^{(r)},S}(f)$. By linearity of SHAP values [Lundberg and Lee, 2017] and the simple structure relating preference and utilities in UPM, we can explain UPM by subtracting the Shapley values of $\mathbf{x}^{(\ell)}$ with $\mathbf{x}^{(r)}$. However, this type of explanation is only correct when data is rankable, which seldom happens in practice, thus motivating PREF-SHAP.

We apply PREF-SHAP to unrankable synthetic and real-world datasets to connect theory with practice. We split data, i.e. matches with their outcomes, into train (80%), validation (10%), and test (10%) and explain the model on a random subset of the data. The hyperparameters for the kernels are selected using gradient descent, based on the proposed method in Meanti et al. [2022]. We first generate synthetic duelling data where performance can be compared against ground truth, to demonstrate that PREF-SHAP is capable of identifying the relevant features.

Synthetic data We first consider a synthetic experiment with unrankable duelling data. We generate the items by first sampling 1000 item covariates $[x_i^{[0]}, x_i^{AB}, x_i^{AC}, x_i^{BC}] =: \mathbf{p}_i \in \mathbb{R}^4 \sim \mathcal{N}(0, \mathbf{I}_4)$. We

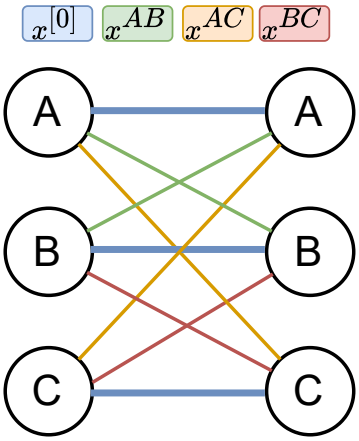


Figure 7.1: An illustration of our simulation: each edge corresponds to the variable that dictates the comparison based on the colour.

associate each item with a cluster membership $c_i \in \{A, B, C\}$, where the assignment is randomly chosen for each item with equal probability. We then form the full item covariate by concatenating $\mathbf{p}_i$ with one-hot encoded c_i as $\mathbf{x}_i = [\mathbf{p}_i, \texttt{one\_hot}(c_i)]$. 40000 matches between randomly chosen pairs of items are conducted by the following mechanism: match outcomes are decided based on the underlying cluster membership of the items. For example, if an item from cluster A competes against an item from cluster B, the winner is decided by their inter-cluster covariate x^{AB}, i.e. $i \preceq j$ if $x_j^{AB} \geq x_i^{AB}$. When the match is between members of the same cluster, it is dictated by the maximum among the within-cluster variable, i.e. $\max(x_i^{[0]}, x_j^{[0]})$. See Fig. 7.1 for an illustration. As no clusters have any advantage over the others, the data is not rankable, and we expect the inter-cluster covariates x^{AB}, x^{AC}, x^{BC} to have similar explanations on average, but significantly different from each other when we examine local explanations.

We consider both global and *grouped-local* explanations of the synthetic dataset in Fig. 7.2 and Fig. 7.3 respectively. In the global explanations, we explain all matches regardless of the cluster membership, while in the grouped-local explanations we only explain matches between items from A against items from B. For more grouped-local explanations on different cluster pairs, we refer to appendix E.2.

Interpreting the simulation explanations. The beehive plots showcase the recovered PREF-SHAP values, where the bar plots demonstrated the average PREF-SHAP values for each feature. The colour in the beehive plots indicates the magnitude of the difference between the corresponding features of the winner and of the loser in that match. For example, a red point in a beehive plot for feature d indicates that the difference $x_{winner}^{(d)} - x_{loser}^{(d)}$ is large.

Fig. 7.2 illustrates the explanation results for the global synthetic experiments. We see that PREF-SHAP identified the within-cluster variable $x^{[0]}$ as the most important, which is a consequence of the fact that the

largest number of matches are played between the items of the same cluster (cf. Fig. 7.1 where there are three blue lines and two lines of each of the other colours). The three inter-cluster variables contributed similarly according to PREF-SHAP, which, by symmetry, should be the case. Furthermore, the correct battle mechanism is captured by PREF-SHAP but not UPM, as we see that the large PREF-SHAP values for each feature are red in the beehive plot. This indicates that items with larger value are more likely to win against items with lower value in the corresponding features. In contrast, SHAP for UPM does not recover this insight.

The explanations for the matches between items from A against items from B, are shown in Fig 7.3. Here, x^{AB} is correctly picked as the relevant feature in these matches with PREF-SHAP, but not with SHAP for UPM. We see again that there is a clear tendency that large PREF-SHAP values are red for feature x^{AB}, showing that PREF-SHAP once again captures the designed gaming mechanism – which is not the case in SHAP for UPM. Intuitively, even though SHAP for UPM allows local explanations, it does so based on a *global utility*, which fails completely in a non-rankable case.

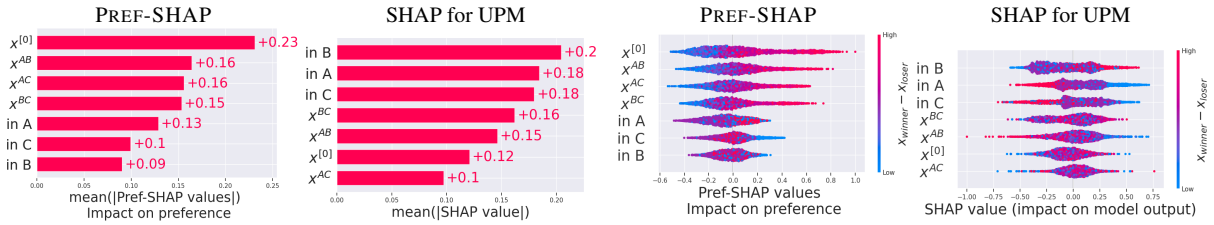


Figure 7.2: Bar and Beehive plots for global explanations on the synthetic dataset.

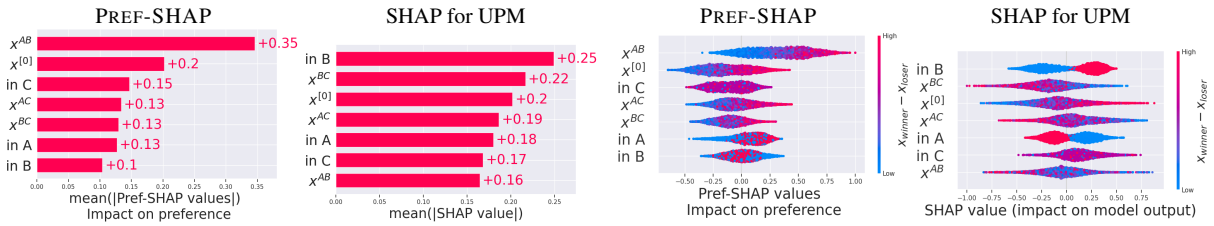


Figure 7.3: Bar and Beehive plots for grouped local explanations on the synthetic dataset (Cluster A vs B).

Real-world explanations For our real-world datasets, we consider publicly available datasets *Chameleon*, *Pokémon* and *Tennis*. We provide descriptive statistics of these datasets in Table E.2 and give their brief descriptions below. Appendix E.2 contains further large scale experiments on an additional dataset consisting of user-item interactions on a fashion retail website.

The Chameleon dataset [Stuart-Fox et al., 2006a] considers 106 contests between 35 male dwarf chameleons. Physical traits of the chameleons are measured such as the *height of their casque*, *length of their jaw*, *body mass* etc. According to Stuart-Fox et al. [2006a], they fitted a linear Bradley Terry model and examined the coefficients to deduce that *casque height* (ch.res) and *relative area of the flank*

Table 7.2: Dataset summary

Dataset	N_{Matches}	N_{items}	N_{Context}	$D_{\text{continuous}}$	D_{binary}	$D^{\text{Context}}_{\text{continuous}}$	$D^{\text{Context}}_{\text{binary}}$
Synthetic	40000	1000	-	4	3	-	-
Chameleon	106	35	-	7	19	-	-
Pokémon	60000	800	-	7	0	-	-
Tennis	95359	3483	4114 (tournaments)	4	7	0	6

Table 7.3: GPM vs UPM. Mean and standard deviations of performance averaged over 5 runs.

	Synthetic		Chameleon		Pokémon		Tennis	
	GPM	UPM	GPM	UPM	GPM	UPM	C-GPM	UPM
Test AUC	$0.98_{\pm 0.00}$	$0.71_{\pm}0.01$	$0.92_{\pm 0.07}$	$0.80_{\pm 0.07}$	$0.86_{\pm 0.00}$	$0.82_{\pm 0.00}$	$0.58_{\pm 0.02}$	$0.52_{\pm 0.02}$
SpecR	0.09		0.24		0.20		$0.13_{\pm 0.07}$	

patch (prop.patch) positively affected the fighting ability the most. *The Pokémon* dataset considers 60000 Pokémon battles among 800 Pokémon. Pokémon have different characteristics such as *attack power*, *speed*, *health* etc. The Pokémon further has at least one different *type* such as *Electric, Water, Fire*, etc. Certain types have advantages and disadvantages against each other, for instance, fire Pokémon are weak to water-based attacks (receiving twice the damage) and as a result have a disadvantage against water Pokémon.

The Tennis dataset considers professional tennis matches between 1991 and 2017 in all major tournaments each year. The data is provided publicly by ATP World Tour [dataset, 2022]. Features such as *birthyear*, *weight*, *height* etc are included about each tennis player together with context details of the match such as the court being indoor or outdoor and what surface the match is being played on.

The above datasets are not rankable, and we validate this claim by comparing GPM performance against UPM in Table E.1, together with the estimated rankability measure *SpecR* proposed in Anderson et al. [2019] for each dataset. *SpecR* measures the similarity of the data to a complete dominance graph (i.e. rankable data). It takes values between 0 and 1 with values close to 1 being evidence in support of rankability. For the Tennis data where there are additional relationships with the context (tournaments), we estimate the average *SpecR* of each tournament. Both the superior performance of GPM over UPM and the low *SpecR* measures suggest that the datasets are generally not rankable, which points to limitations in explaining preferences via utility-based modeling.

Explaining Pokémon battles. We first consider standard dueling data for explaining preferences. We explain the learned preferences and learned differences in utilities on the Pokémon dataset in Figure 7.4. In this dataset, we have summed the Shapley values for *Type* features.

We see that explaining general preferences provides further insight than just explaining the difference

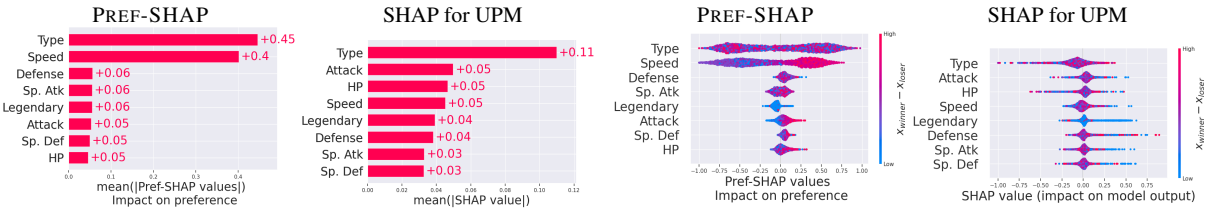


Figure 7.4: Bar and Beehive plots for the Pokémon dataset. PREF-SHAP captures that both speed and type matter, while SHAP for UPM only captures the type importance.

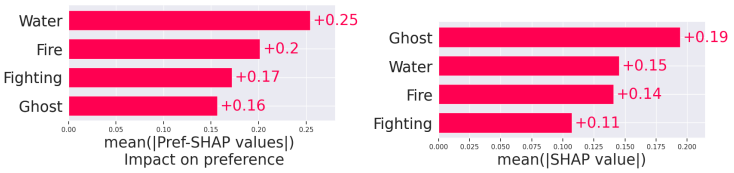


Figure 7.5: Explaining matches between 4 types of Pokémon, among them only fire and water has a type disadvantage/advantage against each other. PREF-SHAP (top) correctly identifies that fire and water are the most important, while water and fire are not deemed most important by SHAP for UPM.

in utility functions. In particular, SHAP for UPM does not capture the additional importance of *Speed* in winning battles. As higher (more red) values of differences in speed $x^{\text{speed}}_{\text{winner}} - x^{\text{speed}}_{\text{loser}}$ have a positive impact on the outcome, we conclude that having higher speed than your opponent is advantageous besides a type advantage. This insight is aligned with the "Sweeper" strategy, where one would employ a leading Pokémon with very high speed and attack to attempt downing the opponent before they can strike back. In Figure 7.5, we see PREF-SHAP can also capture the correct type advantage/disadvantages among the Pokémon, but not SHAP for UPM.

Explaining Chameleon contests. We find that UPM's explanations are more aligned with Stuart-Fox et al. [2006a]'s findings (*prop.path* and *ch.res* are the most important features), which is unsurprising since the Bradley Terry model used in Stuart-Fox et al. [2006a] is also a utility based model. However, since GPM gives a much better predictive performance than UPM (Test AUC 0.92 v.s. 0.80), we believe PREF-SHAP's explanations are also insightful. In fact, PREF-SHAP discovers that having larger *jaw sizes* (jl.res) than your opponent have a significant negative effect on match outcome, a previously undiscovered mechanism from Stuart-Fox et al. [2006a]. We verify this finding in Appendix E.2 by applying PREF-SHAP to GPM trained on multiple folds of the Chameleon dataset and consistently find that high values of the *jaw size* (jl.res) variable have a negative impact on the outcome.

Explaining Tennis matches. We now consider preference learning with context covariates and explain both item characteristics and context covariates in Figure 7.7. In terms of item-based inference, PREF-SHAP finds that being older than your opponent ($x^{\text{yob}}_{\text{winner}} - x^{\text{yob}}_{\text{loser}} < 0 \rightarrow$ Blue), physically heavier, and taller than your opponent positively impacts the chances of winning. We also find that debuting earlier

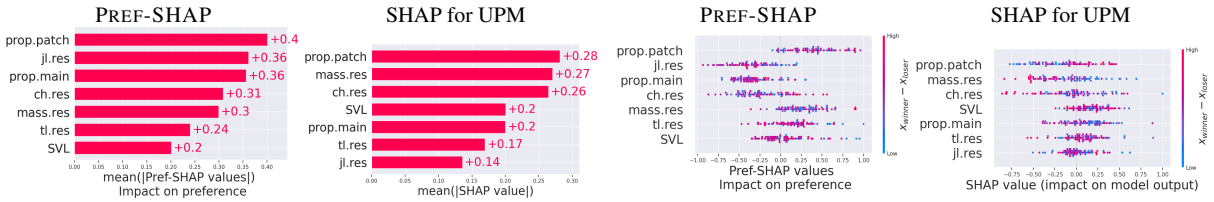


Figure 7.6: Bar and Beehive plots for the Chameleon dataset

as a professional tennis player than your opponent positively impacts your chances of winning. This is not surprising as debuting earlier may be indicative of a promising young talent. Across all competitions, there appear to be no significant patterns in environment effects.

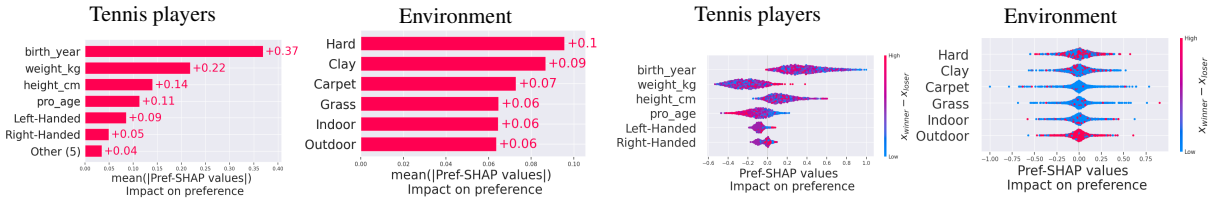


Figure 7.7: Item and context-specific Pref-SHAP values for the Tennis dataset

Explaining Djokovic's losses In plot Figure 7.8, we locally explain all Novak Djokovic's losses in his professional career. Novak Djokovic is regarded as one of the greatest tennis players of all time, so understanding his weakness could serve as a practical demonstration of the utility of PREF-SHAP.

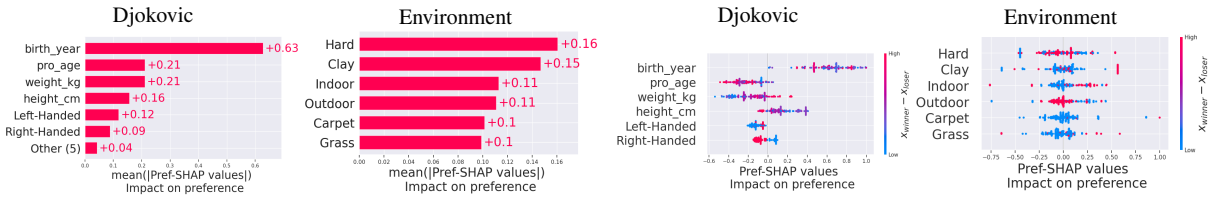


Figure 7.8: Local explanations of Djokovic losses

While the results take a similar shape to the global explanations, Djokovic remarkably seems to be weaker to players shorter than him, contrary to the general advantage of being taller. Besides this, Djokovic seems to be weaker on clay courts and when playing indoors.

7.5 Conclusion

In this work, we proposed PREF-SHAP to explain preference learning for pairwise comparison data. We proposed the appropriate value function for preference explanations and demonstrated the pathologies of the naive concatenation approach in Appendix E.2. Experiments demonstrated that PREF-SHAP recovers richer explanations than utility-based approaches, showcasing the ability of PREF-SHAP in interpreting

the mechanism of preference elicitation.

8 | Discussion, Limitation, and Future work

In the following, I provide a summary of the thesis, and proceed to further summarise each chapter, discuss their limitations, and highlight what the future directions could be to conclude the thesis.

Summary. This thesis addresses a range of problems concerning the safety and transparency of machine learning using kernels. Specifically, we focus on utilizing kernel mean embeddings to represent complex high-dimensional distributions that are challenging to model parametrically. For instance, we use conditional mean embeddings to capture distributions of high-resolution covariates given their coarse-resolution covariates in Chapter 3, causal distributions in Chapter 4, and covariate distributions for explainability in Chapter 5 and 7. Additionally, we employ Gaussian processes to effectively capture uncertainties while modelling problems such as statistical downscaling in Chapter 3, two-staged conditional treatment effects in Chapter 4, and preferential learning in Chapter 6. We demonstrated how these tools can be leveraged together to develop models that have superior performance on various tasks for uncertainty modelling and explainability.

Deconditional downscaling with Gaussian processes

The thesis begins by demonstrating how kernel mean embeddings and Gaussian processes can be used in tandem to tackle a statistical downscaling problem in Chapter 3. In particular, when low-resolution targets are observed along with high-resolution covariates, a mediating variable can be used to instil details from the latter to the former. We propose to treat the low-resolution targets as conditional expectations of the high-resolution target and reverse the conditioning procedure to learn the functional relationship between the high-resolution covariate and high-resolution target.

We proposed a deconditional GP model to tackle this problem. Our approach is made tractable and scalable using variational inference and a novel estimation of conditional mean operator for aggregated data. Furthermore, we bridged the gap in the convergence rate analysis of deconditional mean operator by framing it as the solution to a two-staged vector-valued reconstruction problem. We applied our approach to a challenging atmospheric field downscaling problem, demonstrating its promising empirical results.

In terms of future directions, one limitation of the deconditional GP approach is its inability to handle non-Gaussian output. Therefore, a natural next step would be to incorporate methods from Law et al. [2018a] to model exponential family outputs.

Conditional mean processes can also be applied to problems besides deconditioning. For example, Bayesian optimisation application is considered in Chapter 4, whereas Bayesian quadrature [Briol et al.,

2019] would be another interesting avenue to explore to estimate intractable integration problems with respect to conditional distribution.

Moreover, the key idea behind conditional mean process, which involves incorporating representations of linear functionals from the RKHS to a GP, is likely to be extended to functionals beyond the conditional expectations. For example, one could incorporate the kernel Bayes rule [Fukumizu et al., 2013] formulation with a GP and apply to kernel-based state-space modelling such as the work in Song et al. [2009] while quantifying estimation uncertainty.

Uncertainty Quantification for Causal Data Fusion

In Chapter 4, we address the issue of uncertainty quantification in estimating a two-stage conditional average treatment effect. We first developed a Bayesian conditional mean embedding model that allows us to capture uncertainty while estimating the conditional mean embedding. We then combine this with conditional mean process from Chapter 3 to develop BayesIMP, an algorithm that enables us to quantify different sources of uncertainty in data and propagate them in a principled way into the decision-making pipeline, here in the form of causal Bayesian optimisation.

Our method's limitation arises from the need for complete knowledge of causal graphs and specific assumptions that permit us to "chain" two graphs to estimate the causal effect. It would be intriguing to investigate similar multi-dataset settings while estimating the causal parameter under the potential outcome framework of causal inference. Although assumptions such as strong ignorability [Rubin, 2005] must be made, one could argue that they are less restrictive than trying to construct a complete causal graph.

In terms of future directions, we believe that creating a general framework for combining multiple datasets for inference could have significant practical implications. Learning from multiple datasets could be seen as a missing data problem as collecting data from the entire joint distribution could be expensive, while gathering smaller subsets of data might be more manageable. Having a principled procedure to incorporate uncertainty at the data imputation level is crucial for model safety. Furthermore, we believe our proposed Bayesian conditional mean embedding could allow for the revisiting of conditional mean embedding applications, but incorporating uncertainty quantification and a more principled hyperparameter optimisation process based on the marginal likelihood. An example of recent work adopting this approach is Martinez-Taboada and Sejdinovic [2022].

RKHS-SHAP: Shapley values for Kernel Methods

In Chapter 5, we shift our focus to another important aspect of trustworthy ML, namely explainability tools and, in particular, how they can be applied in a principled and effective way to kernel methods. We adopted the popular paradigm of framing local explanation as a coalition game between features, with a predefined value function that measures some notion of feature contribution. We demonstrated how one could utilise conditional mean embedding to effectively estimate the (conditional) value function without the need for density estimation, thus resulting in RKHS-SHAP. We further contributed by proposing a novel Shapley value based regulariser that can be used to control the amount of feature contribution during learning. This line of research is specific to kernel methods, as Shapley value functionals on RKHSs can be derived without access to a labelled dataset. It is an interesting direction to consider whether similar ideas could be explored within the context of model-agnostic Shapley methods more broadly.

One limitation of RKHS-SHAP would be its computational cost in evaluating the Shapley values, a problem shared across most Shapley value based algorithms. However, there are recent work on improving the computation via approximation scheme, such as FastSHAP [Jethani et al., 2021], which could be incorporated into our procedure as well.

Given the connection between RKHS and GP, it is natural to wonder how RKHS-SHAP can be extended to GP models. In fact, in our latest paper [Chau et al., 2023] I used a variant of Shapley values to explain GPs, and used similar techniques as in RKHS-SHAP to estimate non-parametrically the corresponding cooperative game.[1] In addition, we can propagate the predictive uncertainty to the explanations utilising methods introduced in Chapter 3 and Chapter 4.

However, it is also important to ask whether Shapley values should be the default solution to model explanation methods. Firstly, despite being a unique solution to a set of sensible axioms, the appropriate choice of the game (value function) [Chen et al., 2020], is still not well understood. It would be really fruitful to the community if one could restrict the solution space of value functions by proposing another set of axioms akin to what Shapley proposed.

Another interesting direction would be to continue the work on attribution prior. This would allow a practitioner to instill their inductive bias in ways that are arguably easier to interpret than Bayesian priors over certain hyperparameters.

[1] This paper is completed after the initial submission of the thesis and is included during the minor correction period for completeness.

Learning Inconsistent Preference with Gaussian processes

In Chapter 6, we studied the problem of modeling inconsistent preferences, a phenomenon that is prevalent in practice. We adopted the kernel introduced by Pahikkala et al. [2010], Waegeman et al. [2012] and proposed the generalised preferential Gaussian processes. We also demonstrated a stronger universality result than [Waegeman et al., 2012] using c_0 universality notions developed by [Sriperumbudur et al., 2011], allowing for more general domains like $\mathbb{R}^d$.

Although we focused on modeling pairwise preferences, other datasets can be used to model preferences as well. Choice models [Salvatore et al., 2008] that take in an item set and return a smaller set of preferred items can be seen as a generalisation of pairwise comparison methods, as for any pairwise preference relation one can generate a choice rule but not vice versa. As such, it would be interesting to extend our non-parametric formulation to model choice behaviour as well as challlenging similar rationality assumptions for choice models. An extension of our work considering rational choice functions is considered in Benavoli et al. [2023].

Pref-SHAP: Explaining preference with Shapley values

Lastly, we integrate the work from Chapter 5 with Chapter 7 to study a preference explanation problem. We showed that a naïve application of the existing explanation algorithm fails to distinguish symmetries within the data, in this case, it is the fact that players share the same set of features in the preference model. We proposed an appropriate modification to the value function and resulting in the method we term Pref-SHAP. We proposed how the value function should be modified under different scenarios and demonstrated the effectiveness of our Pref-SHAP through extensive experiments.

One promising future direction could be to generalise the framework of designing value functions that respects different classes of symmetries within the data. For example, to explain persistent homology models discussed in Kwitt et al. [2015].

References

Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *arXiv preprint arXiv:1903.10464*, 2019.

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

Virginia Aglietti, Theodoros Damoulas, Mauricio Álvarez, and Javier González. Multi-task causal learning with gaussian processes. *Advances in Neural Information Processing Systems*, 33, 2020a.

Virginia Aglietti, Xiaoyu Lu, Andrei Paleyes, and Javier González. Causal bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 3155–3164. PMLR, 2020b.

Nir Ailon and Mehryar Mohri. Preference-based learning to rank. *Machine Learning*, 80(2-3):189–211, 2010.

Ahmed M Alaa, Jinsung Yoon, Scott Hu, and Mihaela Van der Schaar. Personalized risk scoring for critical care prognosis using mixtures of gaussian processes. *IEEE Transactions on Biomedical Engineering*, 65(1):207–218, 2017.

David Alvarez-Melis and Tommi S Jaakkola. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*, 2018.

Paul Anand. Are the preference axioms really rational? *Theory and Decision*, 23(2):189–214, Sep 1987. ISSN 1573-7187. 10.1007/BF00126305. URL https://doi.org/10.1007/BF00126305.

Paul Anderson, Timothy Chartier, and Amy Langville. The rankability of data. *SIAM Journal on Mathematics of Data Science*, 1(1):121–143, 2019.

Robert J. Aumann. Utility theory without the completeness axiom: A correction. *Econometrica*, 32 (1/2):210–212, 1964. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1913746.

Ilke Aydogan, Loïc Berger, Valentina Bosetti, et al. Three layers of uncertainty and the role of model misspecification. Technical report, 2020.

Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016.

William L. Barnes, Thomas S. Pagano, and Vincent V. Salomonson. Prelaunch characteristics of the

Moderate Resolution Imaging Spectroradiometer (MODIS) on EOS-AM1. *IEEE Transactions on Geoscience and Remote Sensing*, 1998.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(11), 2006.

Alessio Benavoli, Dario Azzimonti, and Dario Piga. Learning choice functions with gaussian processes. *arXiv preprint arXiv:2302.00406*, 2023.

Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert. Lead-lag detection and network clustering for multivariate time series with an application to the us equity market. *arXiv preprint arXiv:2201.08283*, 2022.

Alain Berlinet and Christine Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer, 2004.

Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

Ioana Bica and Mihaela van der Schaar. Transfer learning on heterogeneous feature spaces for treatment effects estimation. *arXiv preprint arXiv:2210.06183*, 2022.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

François-Xavier Briol, Chris J Oates, Mark Girolami, Michael A Osborne, Dino Sejdinovic, et al. Probabilistic integration: A role in statistical computation? *Statistical Science*, 2019.

Thomas R Cameron, Amy N Langville, and Heather C Smith. On the graph laplacian and the rankability of data. *Linear Algebra and its Applications*, 588:81–100, 2020.

Liangliang Cao, Xin Jin, Zhijun Yin, Andrey Del Pozo, Jiebo Luo, Jiawei Han, and Thomas S Huang. Rankcompete: Simultaneous ranking and clustering of information networks. *Neurocomputing*, 95: 98–104, 2012.

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 2007.

Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.

Manuela Cattelan, Cristiano Varin, and David Firth. Dynamic bradley–terry modelling of sports tournaments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(1):135–150, 2013.

David Causeur and François Husson. A 2-dimensional extension of the Bradley–Terry model for paired comparisons. *Journal of Statistical Planning and Inference*, 135(2):245–259, 2005.

Siu Lun Chau, Mihai Cucuringu, and Dino Sejdinovic. Spectral ranking with covariates, 2020.

Siu Lun Chau, Shahine Bouabid, and Dino Sejdinovic. Deconditional downscaling with gaussian processes. *arXiv preprint arXiv:2105.12909*, 2021a.

Siu Lun Chau, Javier Gonzalez, and Dino Sejdinovic. RKHS-SHAP: Shapley values for kernel methods. *arXiv preprint arXiv:2110.09167*, 2021b.

Siu Lun Chau, Jean-Francois Ton, Javier González, Yee Teh, and Dino Sejdinovic. Bayesimp: Uncertainty quantification for causal data fusion. *Advances in Neural Information Processing Systems*, 34:3466–3477, 2021c.

Siu Lun Chau, Javier Gonzalez, and Dino Sejdinovic. Learning inconsistent preferences with gaussian processes. *International Conference on Artificial Intelligence and Statistics*, 2022.

Siu Lun Chau, Krikamol Muandet, and Dino Sejdinovic. Explaining the uncertain: Stochastic shapley values for gaussian process models. *arXiv preprint arXiv:2305.15167*, 2023.

Hugh Chen, Joseph D Janizek, Scott Lundberg, and Su-In Lee. True to the model or true to the data? *arXiv preprint arXiv:2006.16234*, 2020.

Shuo Chen and Thorsten Joachims. Modeling intransitivity in matchup and comparison data. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, page 227–236, 2016.

Weiwei Cheng, Eyke Hüllermeier, Willem Waegeman, and Volkmar Welker. Label ranking with partial abstention based on thresholded probabilistic models. In *Advances in neural information processing systems*, pages 2501–2509, 2012.

Wei Chu and Zoubin Ghahramani. Preference learning with gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 137–144, 2005.

Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*, 2018.

Travis A Courtney, Mario Lebrato, Nicholas R Bates, Andrew Collins, Samantha J De Putron, Rebecca

Garley, Rod Johnson, Juan-Carlos Molinero, Timothy J Noyes, Christopher L Sabine, et al. Environmental controls on modern scleractinian coral and reef-scale calcification. *Science advances*, 3(11): e1701356, 2017.

Ian Covert and Su-In Lee. Improving kernelshap: Practical shapley value estimation using linear regression. In *International Conference on Artificial Intelligence and Statistics*, pages 3457–3465. PMLR, 2021.

Ian Covert, Scott Lundberg, and Su-In Lee. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, 33, 2020.

Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.

Mihai Cucuringu. Sync-rank: Robust ranking, constrained ranking and rank aggregation via eigenvector and sdp synchronization. *IEEE Transactions on Network Science and Engineering*, 3(1):58–79, 2016.

Sébastien Da Veiga. Kernel-based anova decomposition and shapley effects–application to global sensitivity analysis. *arXiv preprint arXiv:2101.05487*, 2021.

Andreas C. Damianou and Neil D. Lawrence. Deep Gaussian Processes. *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2013.

Alexandre d'Aspremont, Mihai Cucuringu, and Hemant Tyagi. Ranking and synchronization from pairwise measurements via svd. *arXiv preprint arXiv:1906.02746*, 2019.

Tennis dataset. https://datahub.io/sports-data/atp-world-tour-tennis-data, 2022.

Petros Drineas and Michael W. Mahoney. On the nystrom method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6(72):2153–2175, 2005. URL http://jmlr.org/papers/v6/drineas05a.html.

R. M. Dudley. *Real Analysis and Probability*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2 edition, 2002. 10.1017/CBO9780511755347.

Alexandre Duval and Fragkiskos D Malliaros. Graphsvx: Shapley value explanations for graph neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 302–318. Springer, 2021.

Veronika Eyring, Sandrine Bony, Gerald A Meehl, Catherine A Senior, Bjorn Stevens, Ronald J Stouffer, and Karl E Taylor. Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development*, 2016.

Yifan Feng, René Caldentey, and Christopher Ryan. Robust learning of consumer preferences. *Operations Research*, 12 2021. 10.1287/opre.2021.2157.

Ana Ferro, Francisco Pina, Milton Severo, Pedro Dias, Francisco Botelho, and Nuno Lunet. Use of statins and serum levels of prostate specific antigen. *Acta Urológica Portuguesa*, 32(2):71–77, 2015.

Maurizio Filippone and Raphael Engler. Enabling scalable stochastic gradient-based inference for gaussian processes by employing the unbiased linear system solver (ulisse). *arXiv preprint arXiv:1501.05427*, 2015.

Gregory M. Flato. Earth system models: an overview. *WIREs Climate Change*, 2011.

Seth Flaxman, Dino Sejdinovic, John P Cunningham, and Sarah Filippi. Bayesian learning of kernel embeddings. *arXiv preprint arXiv:1603.02160*, 2016.

Fajwel Fogel, Alexandre d'Aspremont, and Milan Vojnovic. Spectral ranking using seriation. *The Journal of Machine Learning Research*, 17(1):3013–3057, 2016.

Gerald B Folland. *Real analysis: modern techniques and their applications*, volume 40. John Wiley & Sons, 1999.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Christopher Frye, Ilya Feige, and Colin Rowat. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. *arXiv preprint arXiv:1910.06358*, 2019.

Christopher Frye, Damien de Mijolla, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley-based explainability on the data manifold. *arXiv preprint arXiv:2006.01272*, 2020.

Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 2004.

Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems*, 2008.

Kenji Fukumizu, Le Song, and Arthur Gretton. Kernel bayes' rule. *arXiv preprint arXiv:1009.5736*, 2010.

Kenji Fukumizu, Le Song, and Arthur Gretton. Kernel bayes' rule: Bayesian inference with positive definite kernels. *The Journal of Machine Learning Research*, 14(1):3753–3783, 2013.

Johannes Fürnkranz and Eyke Hüllermeier. Pairwise preference learning and ranking. In Nada Lavrač,

Dragan Gamberger, Hendrik Blockeel, and Ljupčo Todorovski, editors, *Machine Learning: ECML 2003*, pages 145–156, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-39857-8.

Johannes Fürnkranz and Eyke Hüllermeier. Preference learning and ranking by pairwise comparison. In *Preference learning*, pages 65–82. Springer, 2010.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, 2018.

William V Gehrlein. Condorcet's paradox. *Theory and Decision*, 15(2):161–197, 1983.

Sahra Ghalebikesabi, Lucile Ter-Minassian, Karla DiazOrdaz, and Chris C Holmes. On locality of local explanation models. *Advances in Neural Information Processing Systems*, 34, 2021.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019.

Periklis Gogas and Theophilos Papadimitriou. Machine learning in economics and finance. *Computational Economics*, 57:1–4, 2021.

Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Javier González, Zhenwen Dai, Andreas Damianou, and Neil D Lawrence. Preferential bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1282–1291. JMLR. org, 2017.

Mihajlo Grbovic, Nemanja Djuric, Shengbo Guo, and Slobodan Vucetic. Supervised clustering of label ranking data using label preference information. *Machine learning*, 93(2-3):191–225, 2013.

Brandon M Greenwell, Bradley C Boehmke, and Andrew J McCarthy. A simple and effective model-based variable importance measure. *arXiv preprint arXiv:1805.04755*, 2018.

Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. *Advances in neural information processing systems*, 20, 2007.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

Steffen Grünewälder, Guy Lever, Luca Baldassarre, Sam Patterson, Arthur Gretton, and Massimilano

Pontil. Conditional Mean Embeddings as Regressors. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 2012.

Oliver Hamelijnck, Theodoros Damoulas, Kangrui Wang, and Mark A. Girolami. Multi-resolution multi-task Gaussian processes. In *Advances in Neural Information Processing Systems*, 2019.

Leif Hancox-Li. Robustness in machine learning explanations: does it matter? In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 640–647, 2020.

Dan He, Xingquan Zhu, and Xindong Wu. Error detection and uncertainty modeling for imprecise data. In *2009 21st IEEE International Conference on Tools with Artificial Intelligence*, pages 792–795. IEEE, 2009.

James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable Variational Gaussian Process Classification. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38, pages 351–360, 2015.

Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. *Advances in neural information processing systems*, 33:4778–4789, 2020.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. A review of kernel methods in machine learning. *Mac-Planck-Institute Technical Report*, 156, 2006.

Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Jose M Hernández-Lobato. Collaborative gaussian processes for preference learning. In *Advances in neural information processing systems*, pages 2096–2104, 2012.

Kelvin Hsu and Fabio Ramos. Bayesian Deconditional Kernel Mean Embeddings. Proceedings of Machine Learning Research. PMLR, 2019.

Kelvin Hsu, Richard Nock, and Fabio Ramos. Hyperparameter learning for conditional kernel mean embeddings with rademacher complexity bounds. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 227–242. Springer, 2018.

Robert Hu and year = howpublished = https://github.com/MrHuff/PREF-SHAP Chau, Siu Lun.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

Aditya Jain, Manish Ravula, and Joydeep Ghosh. Biased models have biased explanations. *arXiv preprint arXiv:2012.10986*, 2020.

Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. Feature relevance quantification in explainable ai: A causal problem. In *International Conference on Artificial Intelligence and Statistics*, pages 2907–2916, 2020.

Neil Jethani, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath. Fastshap: Real-time shapley value estimation. In *International Conference on Learning Representations*, 2021.

Thorsten Joachims. Svm-rank: Support vector machine for ranking. *Cornell University*, 2009.

Kaggle. https://www.kaggle.com/datasets/mathchi/diabetes-data-set?resource=download, 2022.

Daniel Kahneman and Amos Tversky. On the interpretation of intuitive probability: A reply to jonathan cohen. 1979.

Olav Kallenberg. *Foundations of Modern Probability*. Springer, 2002.

Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences, 2018.

Shubhra Kanti Karmaker Santu, Parikshit Sondhi, and ChengXiang Zhai. On application of learning to rank for e-commerce search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 475–484, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015.

Noemi Kreif and Karla DiazOrdaz. Machine learning in policy evaluation: new tools for causal inference. *arXiv preprint arXiv:1903.00402*, 2019.

Nils M Kriege. Weisfeiler and leman go walking: Random walk kernels revisited. *arXiv preprint arXiv:2205.10914*, 2022.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

Roland Kwitt, Stefan Huber, Marc Niethammer, Weili Lin, and Ulrich Bauer. Statistical topological data

analysis - a kernel perspective. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3070–3078. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5887-statistical-topological-data-analysis-a-kernel-perspective.pdf.

FM Larkin. Gaussian measure in Hilbert space and applications in numerical analysis. *The Rocky Mountain Journal of Mathematics*, 1972.

Ho Chung Law, Dino Sejdinovic, Ewan Cameron, Tim Lucas, Seth Flaxman, Katherine Battle, and Kenji Fukumizu. Variational learning on aggregate outputs with gaussian processes. In *Advances in Neural Information Processing Systems*, pages 6081–6091, 2018a.

Ho Chung Law, Peilin Zhao, Leung Sing Chan, Junzhou Huang, and Dino Sejdinovic. Hyperparameter learning via distributional transfer. In *Advances in Neural Information Processing Systems*, 2019.

Ho Chung Leon Law, Dougal Sutherland, Dino Sejdinovic, and Seth Flaxman. Bayesian approaches to distribution regression. In *International Conference on Artificial Intelligence and Statistics*, pages 1167–1176. PMLR, 2018b.

Leon Ho Chung Law, Dino Sejdinovic, Ewan Cameron, Tim C.D. Lucas, Seth Flaxman, Katherine Battle, and Kenji Fukumizu. Variational learning on aggregate outputs with Gaussian processes. In *Advances in Neural Information Processing Systems*, 2018c.

Guy Lever, John Shawe-Taylor, Ronnie Stafford, and Csaba Szepesvári. Compressed conditional mean embeddings for model-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. MMMD GAN: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pages 2203–2213, 2017.

Jiyi Li, Yukino Baba, and Hisashi Kashima. Simultaneous clustering and ranking from pairwise comparisons. In *IJCAI*, pages 1554–1560, 2018.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

Z. Li, J.-F. Ton, D. Oglic, and D. Sejdinovic. Towards A Unified Analysis of Random Fourier Features. In *International Conference on Machine Learning (ICML)*, pages PMLR 97:3905–3914, 2019a.

Zhu Li, Adrian Perez-Suay, Gustau Camps-Valls, and Dino Sejdinovic. Kernel dependence regularizers

and gaussian processes with applications to algorithmic fairness. *arXiv preprint arXiv:1911.04322*, 2019b.

Christian List. Social Choice Theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2022 edition, 2022.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *Journal of machine learning research*, 2(Feb):419–444, 2002.

R Duncan Luce. On the possible psychophysical laws. *Psychological review*, 66(2):81, 1959.

Milan Lukić and Jay Beder. Stochastic processes with sample paths in reproducing kernel hilbert spaces. *Transactions of the American Mathematical Society*, 353(10):3945–3969, 2001.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

Julien Mairal. End-to-end kernel learning with supervised convolutional kernel networks. *Advances in neural information processing systems*, 29, 2016.

Diego Martinez-Taboada and Dino Sejdinovic. Bayesian counterfactual mean embeddings and off-policy evaluation. *arXiv preprint arXiv:2211.01518*, 2022.

Masayoshi Mase, Art B Owen, and Benjamin B Seiler. Cohort shapley value for algorithmic fairness. *arXiv preprint arXiv:2105.07168*, 2021.

Giacomo Meanti, Luigi Carratino, Lorenzo Rosasco, and Alessandro Rudi. Kernel methods through the roof: handling billions of points efficiently. *ArXiv*, abs/2006.10350, 2020.

Giacomo Meanti, Luigi Carratino, Ernesto De Vito, and Lorenzo Rosasco. Efficient hyperparameter tuning for large scale kernel ridge regression, 2022. URL https://arxiv.org/abs/2201.06314.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

Tong Meng, Xuyang Jing, Zheng Yan, and Witold Pedrycz. A survey on machine learning for data fusion. *Information Fusion*, 57:115–129, 2020.

Charles A Micchelli and Massimiliano Pontil. On learning vector-valued functions. *Neural computation*, 2005.

Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(Dec):2651–2667, 2006.

Jovana Mitrovic, Dino Sejdinovic, and Yee Whye Teh. Causal inference via kernel deviance measures. *arXiv preprint arXiv:1804.04622*, 2018.

Frederick Mosteller and Philip Nogee. An experimental measurement of utility. *Journal of Political Economy*, 59(5):371–404, 1951.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *arXiv preprint arXiv:1605.09522*, 2016a.

Krikamol Muandet, Bharath Sriperumbudur, Kenji Fukumizu, Arthur Gretton, and Bernhard Schölkopf. Kernel mean shrinkage estimators. *Journal of Machine Learning Research*, 2016b.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10 (1-2):1–141, 2017. 10.1561/2200000060. URL http://dx.doi.org/10.1561/2200000060.

Krikamol Muandet, Motonobu Kanagawa, Sorawit Saengkyongam, and Sanparith Marukatat. Counterfactual mean embeddings. *arXiv preprint arXiv:1805.08845*, 2018.

Krikamol Muandet, Arash Mehrjou, Si Kai Lee, and Anant Raj. Dual instrumental variable regression. *arXiv preprint arXiv:1910.12358*, 2019.

Trung V Nguyen, Alexandros Karatzoglou, and Linas Baltrunas. Gaussian process factorization machines for context-aware recommendations. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 63–72, 2014.

League of Legends Interpretability Demonstration. https://slundberg.github.io/shap/notebooks/League%20of%20Legends%20Win%20Prediction%20with%20XGBoost.html, 2022.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the Web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, 1998.

Tapio Pahikkala, Evgeni Tsivtsivadze, Antti Airola, Jouni Järvinen, and Jorma Boberg. An efficient algorithm for learning to rank from preference graphs. *Machine Learning*, 75(1):129–165, 2009.

Tapio Pahikkala, Willem Waegeman, Evgeni Tsivtsivadze, Tapio Salakoski, and Bernard De Baets.

Learning intransitive reciprocal relations with kernel methods. *European Journal of Operational Research*, 206(3):676–685, 2010.

Tapio Pahikkala, Antti Airola, Michiel Stock, Bernard De Baets, and Willem Waegeman. Efficient regularized least-squares algorithms for conditional ranking on relational data. *Machine Learning*, 93 (2-3):321–356, 2013.

Junhyung Park and Krikamol Muandet. A measure-theoretic approach to kernel conditional mean embeddings. *arXiv preprint arXiv:2002.03689*, 2020.

Junhyung Park, Uri Shalit, Bernhard Schölkopf, and Krikamol Muandet. Conditional distributional treatment effect with kernel conditional mean embeddings and u-statistic regression. In *International Conference on Machine Learning*, pages 8401–8412. PMLR, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. 2019.

Vern Paulsen and Mrinal Raghupathi. *An introduction to the theory of reproducing kernel Hilbert spaces*. Cambridge university press, 2016.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

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

Adrián Pérez-Suay, Valero Laparra, Gonzalo Mateo-García, Jordi Muñoz-Marí, Luis Gómez-Chova, and Gustau Camps-Valls. Fair kernel learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 339–355. Springer, 2017.

S. Platnick, M.D. King, S.A. Ackerman, W.P. Menzel, B.A. Baum, J.C. Riedi, and R.A. Frey. The MODIS cloud products: algorithms and examples from Terra. *IEEE Transactions on Geoscience and Remote Sensing*, 2003.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2008.

Ali Rahimi, Benjamin Recht, et al. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2007.

C Rasmussen and C Williams. Gaussian Processes for Machine Learning, 2005a.

C Rasmussen and C Williams. Gaussian Processes for Machine Learning, 2005b.

Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4741–4748, 2015.

L. A. Remer, Y. J. Kaufman, D. Tanré, S. Mattoo, D. A. Chu, J. V. Martins, R.-R. Li, C. Ichoku, R. C. Levy, R. G. Kleidman, T. F. Eck, E. Vermote, and B. N. Holben. The MODIS Aerosol Algorithm, Products, and Validation. *Journal of the Atmospheric Sciences*, 2005.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

Livio Robaldo, Serena Villata, Adam Wyner, and Matthias Grabmair. Introduction for artificial intelligence and law: special issue “natural language processing for legal texts”, 2019.

Malcolm Roberts. MOHC HadGEM3-GC31-HM model output prepared for CMIP6 HighResMIP hist-1950. *Earth System Grid Federation*, 2018 v20180730. 10.22033/ESGF/CMIP6.6040.

Ribana Roscher, Bastian Bohn, Marco F Duarte, and Jochen Garcke. Explainable machine learning for scientific insights and discoveries. *Ieee Access*, 8:42200–42216, 2020.

Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005.

Walter Rudin. Functional analysis, mcgrawhill. *Inc, New York*, 1991.

Tim GJ Rudner, Dino Sejdinovic, and Yarin Gal. Inter-domain deep Gaussian processes. In *International Conference on Machine Learning*, 2020.

S. Saitoh and Yoshihiro Sawano. *Theory of Reproducing Kernels and Applications*. Springer, 2016.

Dominick Salvatore et al. Microeconomics: theory and applications. *OUP Catalogue*, 2008.

Jari Saramäki, Mikko Kivelä, Jukka-Pekka Onnela, Kimmo Kaski, and Janos Kertesz. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 75(2):027105, 2007.

Anna Schmitz, Maram Akila, Dirk Hecker, Maximilian Poretschkin, and Stefan Wrobel. The why and how of trustworthy ai. *at-Automatisierungstechnik*, 70(9):793–804, 2022.

Marko Scholze, J. Icarus Allen, William J. Collins, Sarah E. Cornell, Chris Huntingford, Manoj M. Joshi, Jason A. Lowe, Robin S. Smith, and Oliver Wild. *Earth system models*. Cambridge University Press, 2012.

Dino Sejdinovic. Advanced topics in machine learning. 2019.

Dino Sejdinovic, Heiko Strathmann, Maria Lomeli Garcia, Christophe Andrieu, and Arthur Gretton. Kernel adaptive metropolis-hastings. In *International conference on machine learning*, pages 1665–1673. PMLR, 2014.

Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

Suresh Shirgave, Chetan Awati, Rashmi More, and Sonam Patil. A review on credit card fraud detection using machine learning. *International Journal of Scientific & technology research*, 8(10):1217–1220, 2019.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.

Edwin Simpson and Iryna Gurevych. Scalable bayesian preference learning for crowds. *Machine Learning*, pages 1–30, 2020.

B. Sinervo and C.M. Lively. The rock-paper-scissors game and the evolution of alternative male strategies. *Nature*, 380(6571):240–243, 1996.

Rahul Singh, Maneesh Sahani, and Arthur Gretton. Kernel instrumental variable regression. In *Advances in Neural Information Processing Systems*, 2019.

Rahul Singh, Liyuan Xu, and Arthur Gretton. Kernel methods for policy evaluation: Treatment effects, mediation analysis, and off-policy planning. *arXiv preprint arXiv:2010.04855*, 2020.

Arvind Kumar Sinha, Md Amir Khusru Akhtar, and Ashwani Kumar. Resume screening using natural language processing and machine learning: A systematic review. *Machine Learning and Information Processing: Proceedings of ICMLIP 2020*, pages 207–214, 2021.

Michael T. Smith, Mauricio A. Álvarez, Max Zwiessele, and Neil D. Lawrence. Differentially private

regression with Gaussian processes. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 1195–1203, 2018.

Eunhye Song, Barry L Nelson, and Jeremy Staum. Shapley effects for global sensitivity analysis: Theory and computation. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1060–1083, 2016.

Le Song, Jonathan Huang, Alex Smola, and Kenji Fukumizu. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(May):1393–1434, 2012.

Le Song, Kenji Fukumizu, and Arthur Gretton. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 2013.

Bharath K Sriperumbudur, Kenji Fukumizu, and Gert RG Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *Journal of Machine Learning Research*, 2011.

Bharath K. Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert R. G. Lanckriet. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 2012.

Thomas A Stamey, John N Kabalin, John E McNeal, Iain M Johnstone, Fuad Freiha, Elise A Redwine, and Norman Yang. Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. ii. radical prostatectomy treated patients. *The Journal of urology*, 141(5):1076–1083, 1989.

Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008a.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Publishing Company, Incorporated, 2008b.

Graeme L Stephens, Deborah G Vane, Ronald J Boain, Gerald G Mace, Kenneth Sassen, Zhien Wang, Anthony J Illingworth, Ewan J O'connor, William B Rossow, Stephen L Durden, et al. The CloudSat mission and the A-train: A new dimension of space-based observations of clouds and precipitation. *Bulletin of the American Meteorological Society*, 2002.

Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.

Devi Stuart-Fox, David Firth, Adnan Moussalli, and Martin Whiting. Multiple signals in chameleon contests: Designing and analysing animal contests as a tournament. *Animal Behaviour*, 71:1263–1271, 06 2006a. 10.1016/j.anbehav.2005.07.028.

Devi M Stuart-Fox, David Firth, Adnan Moussalli, and Martin J Whiting. Multiple signals in chameleon contests: designing and analysing animal contests as a tournament. *Animal Behaviour*, 71(6):1263–1271, 2006b.

Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International Conference on Machine Learning*, pages 9269–9278. PMLR, 2020.

Zoltán Szabó and Bharath K Sriperumbudur. Characteristic and universal tensor product kernels. *The Journal of Machine Learning Research*, 2017.

Zoltán Szabó, Bharath K. Sriperumbudur, Barnabás Póczos, and Arthur Gretton. Learning theory for distribution regression. *Journal of Machine Learning Research*, 2016.

Yusuke Tanaka, Toshiyuki Tanaka, Tomoharu Iwata, Takeshi Kurashima, Maya Okawa, Yasunori Akagi, and Hiroyuki Toda. Spatially aggregated Gaussian processes with multivariate areal outputs. *Advances in Neural Information Processing Systems*, 2019.

Clay Thompson. Causal graph analysis with the causalgraph procedure. In *Proceedings of SAS Global Forum*, 2019.

Louis L Thurstone. A law of comparative judgment. *Psychological review*, 101(2):266, 1994.

Michalis K. Titsias. Variational learning of inducing variables in sparse gaussian processes. In *AISTATS*, 2009.

Jean-Francois Ton, Lucian Chan, Yee Whye Teh, and Dino Sejdinovic. Noise contrastive meta-learning for conditional density estimation using kernel mean embeddings. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 1099–1107. PMLR, 13–15 Apr 2021a. URL http://proceedings.mlr.press/v130/ton21a.html.

Jean-Francois Ton, CHAN Lucian, Yee Whye Teh, and Dino Sejdinovic. Noise contrastive meta-learning for conditional density estimation using kernel mean embeddings. In *International Conference on Artificial Intelligence and Statistics*, pages 1099–1107. PMLR, 2021b.

Giancarlo Ferrari Trecate, Christopher KI Williams, and Manfred Opper. Finite-dimensional approxima-

tion of gaussian processes. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 218–224, 1999.

Rosy Tsopra, Jean-Baptiste Lamy, and Karima Sedki. Using preference learning for detecting inconsistencies in clinical practice guidelines: Methods and application to antibiotherapy. *Artificial Intelligence in Medicine*, 89, 04 2018. 10.1016/j.artmed.2018.04.013.

Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5(4):297–323, 1992.

Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

Arto Klam Ville Tanskanen , Krista Longi. Non-Linearities in Gaussian Processes with Integral Observations. *IEEE international Workshop on Machine Learning for Signal*, 2020.

Aurore Voldoire. CNRM-CERFACS CNRM-CM6-1-HR model output prepared for CMIP6 HighResMIP hist-1950. *Earth System Grid Federation*, 2019 v20190221. 10.22033/ESGF/CMIP6.4040.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

Willem Waegeman, Tapio Pahikkala, Antti Airola, Tapio Salakoski, Michiel Stock, and Bernard De Baets. A kernel-based framework for learning graded relations from data. *IEEE Transactions on Fuzzy Systems*, 20(6):1090–1101, 2012.

D. Watson-Parris, N. Schutgens, N. Cook, Z. Kipling, P. Kershaw, E. Gryspeerdt, B. Lawrence, and P. Stier. Community Intercomparison Suite (CIS) v1.4.0: a tool for intercomparing models and observations. *Geoscientific Model Development*, 2016.

Junfeng Wen, Chun-Nam Yu, and Russell Greiner. Robust learning under uncertain test distributions: Relating covariate shift to model misspecification. In *International Conference on Machine Learning*, pages 631–639. PMLR, 2014.

Martin J Whiting, Jonathan K Webb, and J Scott Keogh. Flat lizard female mimics use sexual deception in visual but not chemical signals. *Proceedings of the Royal Society B: Biological Sciences*, 276(1662): 1585–1591, 2009.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784, 2015.

Andrew Gordon Wilson, David A. Knowles, and Zoubin Ghahramani. Gaussian process regression networks. *Proceedings of the 29th International Conference on Machine Learning, ICML*, 2012.

Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. *Advances in neural information processing systems*, 25:476–484, 2012.

Chih-Kuan Yeh, Kuan-Yun Lee, Frederick Liu, and Pradeep Ravikumar. Threading the needle of on and off-manifold value functions for shapley explanations. *arXiv preprint arXiv:2202.11919*, 2022.

Fariba Yousefi, Michael Thomas Smith, and Mauricio A. Álvarez. Multi-task learning for aggregated data using Gaussian processes. In *Advances in Neural Information Processing Systems*, 2019.

Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020.

Qinyi Zhang, Sarah Filippi, Arthur Gretton, and Dino Sejdinovic. Large-scale kernel methods for independence testing. *Statistics and Computing*, 28(1):113–130, 2018.

Yivan Zhang, Nontawat Charoenphakdee, Zhenguo Wu, and Masashi Sugiyama. Learning from aggregate observations. In *Advances in Neural Information Processing Systems*, 2020.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International conference on machine learning*, pages 7404–7413. PMLR, 2019.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 2017.

Luisa M Zintgraf, Diederik M Roijers, Sjoerd Linders, Catholijn M Jonker, and Ann Nowé. Ordered preference elicitation strategies for supporting multi-objective decision making. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1477–1485. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Masrour Zoghi, Zohar S Karnin, Shimon Whiteson, and Maarten De Rijke. Copeland dueling bandits. In *Advances in Neural Information Processing Systems*, pages 307–315, 2015.

Appendices

A | Chapter 3 Appendix

A.1 Proofs of Section 3.3

Proposition A.1.1. *Let $h : \mathcal{X} \to \mathbb{R}$ such that $\mathbb{E}[|h(X)|] < \infty$. Then, $\{y \in \mathcal{Y} \mid \mathbb{E}[|h(x)||Y = y] < \infty\}$ is a full measure set with respect to $\mathbb{P}_Y$.*

Proof. Since $\mathcal{X}$ is a Borel space and $\mathcal{Y}$ is measurable, the existence of a $\mathbb{P}_Y$-a.e. regular conditional probability distribution is guaranteed by [Kallenberg, 2002, Theorem 6.3]. Now suppose $\mathbb{E}[|h(X)|] < \infty$ and let $\mathcal{Y}^o = \{y \in \mathcal{Y} \mid \mathbb{E}[|h(X)||Y = y] < \infty\}$. Since $\mathbb{E}[|h(X)|] = \mathbb{E}\left[\mathbb{E}[|h(X)| \mid Y]\right]$, the conditional expectation $\mathbb{E}[|h(X)| \mid Y]$ must have finite expectation almost everywhere, i.e. $\mathbb{P}_Y(\mathcal{Y}^o) = 1$. □

Proposition 3.2. *Suppose $\mathbb{E}[|m(X)|] < \infty$ and $\mathbb{E}[|k_X|_k] < \infty$ and let $(X', Y') \sim \mathbb{P}_{XY}$. Then g is a Gaussian process $g \sim \mathcal{GP}(\nu, q)$ a.s. , specified by*

$$\nu(y) = \mathbb{E}[m(X)|Y = y](y, y') = \mathbb{E}[k(X, X')|Y = y, Y' = y'] \tag{A.1}$$

$\forall y, y' \in \mathcal{Y}$. Furthermore, $q(y, y') = \langle \mu_{X|Y=y}, \mu_{X|Y=y'} \rangle_k$ a.s.

Proof of Proposition 3.3.2. We will assume for the sake of simplicity that $m = 0$ in the following derivations and will return to the case of an uncentered GP at the end of the proof.

Show that $g(y)$ is in a space of Gaussian random variables Let $(\Omega, \mathcal{F}, \mathbb{P})$ denote a probability space and $L^2(\Omega, \mathbb{P})$ the space of square integrable random variables endowed with standard inner product. $\forall x \in \mathcal{X}$, since $f(x)$ is Gaussian, then $f(x) \in L^2(\Omega, \mathbb{P})$. We can hence define $\mathcal{S}(f)$ as the closure in $L^2(\Omega, \mathbb{P})$ of the vector space spanned by f, i.e. $\mathcal{S}(f) := \overline{\text{Span}}\{f(x) : x \in \mathcal{X}\}$.

Elements of $\mathcal{S}(f)$ write as limits of centered Gaussian random variables, hence when their covariance sequence converge, they are normally distributed. Let $T \in \mathcal{S}(f)^\perp$, then we have $\mathbb{E}[Tf(x)] = 0$. Let $y \in \mathcal{Y}$, we also have

$$\mathbb{E}[Tg(y)] = \mathbb{E}\left[\int_{\mathcal{X}} Tf(x)\, \mathrm{d}\mathbb{P}_{X|Y=y}\right] \tag{A.2}$$

In order to switch orders of integration, we need to show that the double integral satisfies absolute

convergence.

$$\int_{\mathcal{X}} \mathbb{E}[|Tf(x)|] \, \mathrm{d}\mathbb{P}_{X|Y=y}(x) \leq \int_{\mathcal{X}} \sqrt{\mathbb{E}[T^2]\mathbb{E}[f(x)^2]} \, \mathrm{d}\mathbb{P}_{X|Y=y}(x) \tag{A.3}$$

$$= \sqrt{\mathbb{E}[T^2]} \int_{\mathcal{X}} |k_x|_k \, \mathrm{d}\mathbb{P}_{X|Y=y}(x) \tag{A.4}$$

$$= \sqrt{\mathbb{E}[T^2]}\mathbb{E}[|k_X|_k|Y = y] \tag{A.5}$$

Since $T \in L^2(\Omega, \mathbb{P})$, $\mathbb{E}[T^2] < \infty$. Plus, as we assume that $\mathbb{E}[|k_X|_k] < \infty$, Proposition A.1.1 gives that $\mathbb{E}[|k_X|_k|Y = y] < \infty$ a.s. We can thus apply Fubini's theorem and obtain

$$\mathbb{E}[Tg(y)] = \int_{\mathcal{X}} \mathbb{E}[Tf(x)] \, \mathrm{d}\mathbb{P}_{X|Y=y}(x) = 0 \text{ a.s.} \tag{A.6}$$

As this holds for any $T \in \mathcal{S}(f)^\perp$, we conclude that $g(y) \in \left(\mathcal{S}(f)^\perp\right)^\perp$ a.s. $\Rightarrow g(y) \in \mathcal{S}(f)$ a.s.. We cannot claim yet though that $g(y)$ is Gaussian since we do not know whether it results from a sequence of Gaussian variables with converging variance sequence. We now have to prove that $g(y)$ has a finite variance.

Show that $g(y)$ has finite variance We proceed by computing the expression of the covariance between $g(y)$ and $g(y')$ which is more general and yields the variance.

Let $y, y' \in \mathcal{Y}$, the covariance of $g(y)$ and $g(y')$ is given by

$$q(y, y') = \mathbb{E}[g(y)g(y')] - \mathbb{E}[g(y)]\mathbb{E}[g(y')] \tag{A.7}$$

$$= \mathbb{E}\left[\int_{\mathcal{X}}\int_{\mathcal{X}} f(x)f(x') \, \mathrm{d}\mathbb{P}_{X|Y=y}(x) \, \mathrm{d}\mathbb{P}_{X|Y=y'}(x')\right] \tag{A.8}$$

$$- \mathbb{E}\left[\int_{\mathcal{X}} f(x) \, \mathrm{d}\mathbb{P}_{X|Y=y}(x)\right] \mathbb{E}\left[\int_{\mathcal{X}} f(x') \, \mathrm{d}\mathbb{P}_{X|Y=y'}(x')\right] \tag{A.9}$$

Choosing T as a constant random variable in the above, we can show that $\int_{\mathcal{X}} \mathbb{E}[|f(x)|] \, \mathrm{d}\mathbb{P}_{X|Y=y}(x) < \infty$ a.s. We can hence apply Fubini's theorem to switch integration order in the mean terms (A.9) and obtain that $\mathbb{E}[g(y)] = 0$ since f is centered.

To apply Fubini's theorem to (A.8), we need to show that the triple integration absolutely converges. Let $x, x' \in \mathcal{X}$, we know that $\mathbb{E}[|f(x)f(x')|] \leq \sqrt{\mathbb{E}[f(x)^2]\mathbb{E}[f(x')^2]} = |k_x|_k|k_{x'}|_k$. Using similar arguments

as above, we obtain

$$\int_{\mathcal{X}}\int_{\mathcal{X}} \mathbb{E}[|f(x)f(x')|]\,\mathrm{d}\mathbb{P}_{X|Y=y}(x)\,\mathrm{d}\mathbb{P}_{X|Y=y'}(x') \leq \mathbb{E}[|k_X|_k|Y=y]\mathbb{E}[|k_X|_k|Y=y'] < \infty \text{ a.s.} \tag{A.10}$$

We can thus apply Fubini's theorem which yields

$$\begin{aligned} q(y,y') &= \int_{\mathcal{X}}\int_{\mathcal{X}} \mathbb{E}[f(x)f(x')]\,\mathrm{d}\mathbb{P}_{X|Y=y}(x)\,\mathrm{d}\mathbb{P}_{X|Y=y'}(x') && \text{(A.11)}\\ &= \int_{\mathcal{X}}\int_{\mathcal{X}} \underbrace{\mathrm{Cov}(f(x),f(x'))}_{k(x,x')}\,\mathrm{d}\mathbb{P}_{X|Y=y}(x)\,\mathrm{d}\mathbb{P}_{X|Y=y'}(x') && \text{(A.12)}\\ &= \mathbb{E}[k(X,X')|Y=y,Y'=y'] && \text{(A.13)}\\ &\leq \mathbb{E}[|k_X|_k|Y=y]\mathbb{E}[|k_X|_k|Y=y'] < \infty \text{ a.s.} && \text{(A.14)} \end{aligned}$$

where (X',Y') denote random variables with same joint distribution than (X,Y) as defined in the proposition.

$g(y) \in \mathcal{S}(f)$ and has finite variance $q(y,y)$ a.s., it is thus a centered Gaussian random variable a.s. Furthermore, as this holds for any $y \in \mathcal{Y}$, then any finite subset of $\{g(y) : y \in \mathcal{Y}\}$ follows a multivariate normal distribution which shows that g is a centered Gaussian process on $\mathcal{Y}$ and its covariance function is specified by q.

Uncentered case $m \neq 0$ We now return to an uncentered GP prior on f with assumption that $\mathbb{E}[|m(X)|] < \infty$. By Proposition A.1.1, we get that $\mathbb{E}[|m(X)|\,|Y=y] < \infty$ a.s. for $y \in \mathcal{Y}$.

Let $\nu : y \mapsto \mathbb{E}[m(X)|Y=y]$. We can clearly rewrite g as the sum of ν and a centered GP on $\mathcal{Y}$

$$g(y) = \nu(y) + \int_{\mathcal{X}} (f(x) - m(x))\,\mathrm{d}\mathbb{P}_{X|Y=y}(x), \qquad \forall y \in \mathcal{Y} \tag{A.15}$$

which is well-defined almost surely.

It hence comes $\mathbb{E}[g(y)] = \mathbb{E}[\nu(y)] + 0 = \nu(y)$. Plus since $\nu(y)$ is a constant shift, the covariance is not affected and has the same expression than for the centered GP. Since this holds for any $y \in \mathcal{Y}$, we conclude that $g \sim \mathcal{GP}(\nu, q)$ a.s.

Show that $q(y,y') = \langle \mu_{X|Y=y}, \mu_{X|Y=y'} \rangle_k$ First, we know by Proposition A.1.1 that $\mathbb{E}[|k_X|_k|Y=y] < \infty$ $\mathbb{P}_Y$-a.e. .By triangular inequality, we obtain $|\mu_{X|Y=y}|_k = |\mathbb{E}[k_X|Y=y]|_k \leq \mathbb{E}[|k_X|_k|Y=y] < \infty$ $\mathbb{P}_Y$-a.e. and hence $\mu_{X|Y=y}$ is well-defined up to a set of measure zero with respect to $\mathbb{P}_Y$.

With notations from Proposition 3.3.2, we can proceed for any $y, y' \in \mathcal{Y}$ as

$$
\begin{aligned}
q(y, y') &= \mathbb{E}[k(X, X')|Y = y', Y' = y'] && \text{(A.16)}\\
&= \int_{\mathcal{X}} \int_{\mathcal{X}} k(x, x') \, \mathrm{d}\mathbb{P}_{X|Y=y}(x) \, \mathrm{d}\mathbb{P}_{X|Y=y'}(x') && \text{(A.17)}\\
&= \int_{\mathcal{X}} \int_{\mathcal{X}} \langle k_x, k_{x'} \rangle_k \, \mathrm{d}\mathbb{P}_{X|Y=y}(x) \, \mathrm{d}\mathbb{P}_{X|Y=y'}(x') && \text{(A.18)}\\
&= \left\langle \int_{\mathcal{X}} k_x \, \mathrm{d}\mathbb{P}_{X|Y=y}(x), \int_{\mathcal{X}} k_{x'} \, \mathrm{d}\mathbb{P}_{X|Y=y'}(x') \right\rangle_k \quad \text{a.s.} && \text{(A.19)}\\
&= \langle \mu_{X|Y=y}, \mu_{X|Y=y'} \rangle_k \quad \text{a.s.} && \text{(A.20)}
\end{aligned}
$$

□

Proof of Proposition 3.3.2. Let $y \in \mathcal{Y}$. We first show that $\mu_{X|Y=y}$ is well defined.

First, we know by Proposition A.1.1 that $\mathbb{E}[|k_X|_k|Y = y] < \infty$ $\mathbb{P}_Y$-a.e. .By triangular inequality, we obtain $|\mu_{X|Y=y}|_k = |\mathbb{E}[k_X|Y = y]|_k \leq \mathbb{E}[|k_X|_k|Y = y] < \infty$ $\mathbb{P}_Y$-a.e. and hence $\mu_{X|Y=y}$ is well-defined up to a set of measure zero with respect to $\mathbb{P}_Y$.

If we now further suppose that ℓ is continuous, then $\mathcal{H}_\ell$ is a space of continuous functions [Saitoh and Sawano, 2016, Theorem 2.3]. Under this assumption, points in $\mathcal{H}_\ell$ are separated by almost everywhere equality. Indeed, let $h \in \mathcal{H}_\ell$ such that $h = 0$ $\mathbb{P}_Y$-a.e. and consider $\varepsilon > 0$. Suppose there exists $y \in \mathcal{Y}$ such that $|h(y)| > \varepsilon$. Since h is continuous, y admits an open neighbourhood in which $|h| > \varepsilon/2$ which contradicts that $h = 0$ $\mathbb{P}_Y$-a.e.

Hence, if ℓ is continuous, $\mu_{X|Y=y}$ is unique and well-defined and, with notations from Proposition 3.3.2, we can proceed for any $y, y' \in \mathcal{Y}$ as

$$
\begin{aligned}
q(y, y') &= \mathbb{E}[k(X, X')|Y = y', Y' = y'] && \text{(A.21)}\\
&= \int_{\mathcal{X}} \int_{\mathcal{X}} k(x, x') \, \mathrm{d}\mathbb{P}_{X|Y=y}(x) \, \mathrm{d}\mathbb{P}_{X|Y=y'}(x') && \text{(A.22)}\\
&= \int_{\mathcal{X}} \int_{\mathcal{X}} \langle k_x, k_{x'} \rangle_k \, \mathrm{d}\mathbb{P}_{X|Y=y}(x) \, \mathrm{d}\mathbb{P}_{X|Y=y'}(x') && \text{(A.23)}\\
&= \left\langle \int_{\mathcal{X}} k_x \, \mathrm{d}\mathbb{P}_{X|Y=y}(x), \int_{\mathcal{X}} k_{x'} \, \mathrm{d}\mathbb{P}_{X|Y=y'}(x') \right\rangle_k && \text{(A.24)}\\
&= \langle \mu_{X|Y=y}, \mu_{X|Y=y'} \rangle_k && \text{(A.25)}
\end{aligned}
$$

□

Proposition 3.3.3. *Given aggregate observations $\tilde{\mathbf{z}}$ with homoscedastic noise σ^2, the deconditional posterior of f is defined as the Gaussian process $f|\tilde{\mathbf{z}} \sim \mathcal{GP}(m_{\mathrm{d}}, k_{\mathrm{d}})$ where*

$$m_{\mathrm{d}}(x) = m(x) + (C_{X|Y}^{\top} k_x)^{\top} \mathbf{\Psi}_{\tilde{\mathbf{y}}} (\mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2 \mathbf{I}_M)^{-1} (\tilde{\mathbf{z}} - \nu(\tilde{\mathbf{y}})), \tag{A.26}$$

$$k_{\mathrm{d}}(x, x') = k(x, x') - (C_{X|Y}^{\top} k_x)^{\top} \mathbf{\Psi}_{\tilde{\mathbf{y}}} (\mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2 \mathbf{I}_M)^{-1} \mathbf{\Psi}_{\tilde{\mathbf{y}}}^{\top} C_{X|Y}^{\top} k_{x'}. \tag{A.27}$$

Proof of Proposition 3.3.3. Recall that

$$\begin{bmatrix} f(\mathbf{x}) \\ \tilde{\mathbf{z}} \end{bmatrix} \mid \mathbf{y}, \tilde{\mathbf{y}} \sim \mathcal{N} \left(\begin{bmatrix} m(\mathbf{x}) \\ \nu(\tilde{\mathbf{y}}) \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{\mathbf{xx}} & \mathbf{\Upsilon} \\ \mathbf{\Upsilon}^{\top} & \mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2 \mathbf{I}_M \end{bmatrix} \right). \tag{A.28}$$

where $\mathbf{\Upsilon} = \mathrm{Cov}(f(\mathbf{x}), \tilde{\mathbf{z}}) = \mathbf{\Phi}_{\mathbf{x}}^{\top} C_{X|Y} \mathbf{\Psi}_{\tilde{\mathbf{y}}}$.

Applying Gaussian conditioning, we obtain that

$$f(\mathbf{x}) \mid \tilde{\mathbf{z}}, \mathbf{y}, \tilde{\mathbf{y}} \sim \mathcal{N}(m(\mathbf{x}) + \mathbf{\Upsilon} (\mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2 \mathbf{I}_M)^{-1} (\tilde{\mathbf{z}} - \nu(\tilde{\mathbf{y}})), \tag{A.29}$$

$$\mathbf{K}_{\mathbf{xx}} - \mathbf{\Upsilon} (\mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2 \mathbf{I}_M)^{-1} \mathbf{\Upsilon}^{\top}) \tag{A.30}$$

Since the latter holds for any input $\mathbf{x} \in \mathcal{X}^N$, by Kolmogorov extension theorem this implies that f conditioned on the data $\tilde{\mathbf{z}}, \tilde{\mathbf{y}}$ is a draw from a GP. We denote it $f|\tilde{\mathbf{z}} \sim \mathcal{GP}(m_{\mathrm{d}}, k_{\mathrm{d}})$ and it is specified by

$$m_{\mathrm{d}}(x) = m(x) + k_x^{\top} C_{X|Y} \mathbf{\Psi}_{\tilde{\mathbf{y}}} (\mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2 \mathbf{I}_M)^{-1} (\tilde{\mathbf{z}} - \nu(\tilde{\mathbf{y}})), \tag{A.31}$$

$$k_{\mathrm{d}}(x, x') = k(x, x') - k_x^{\top} C_{X|Y} \mathbf{\Psi}_{\tilde{\mathbf{y}}} (\mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2 \mathbf{I}_M)^{-1} \mathbf{\Psi}_{\tilde{\mathbf{y}}}^{\top} C_{X|Y}^{\top} k_{x'}. \tag{A.32}$$

Note that we abuse notation

$$k_x^{\top} C_{X|Y} \mathbf{\Psi}_{\tilde{\mathbf{y}}} = \begin{bmatrix} \langle k_x, C_{X|Y} \ell_{\tilde{y}_1} \rangle_k & \dots & \langle k_x, C_{X|Y} \ell_{\tilde{y}_M} \rangle_k \end{bmatrix} \tag{A.33}$$

$$= \begin{bmatrix} \langle k_x, \mu_{X|Y=\tilde{y}_1} \rangle_k & \dots & \langle k_x, \mu_{X|Y=\tilde{y}_M} \rangle_k \end{bmatrix} \tag{A.34}$$

$$= \begin{bmatrix} \mathrm{Cov}(f(x), g(\tilde{y}_1)) & \dots & \mathrm{Cov}(f(x), g(\tilde{y}_M)) \end{bmatrix}. \tag{A.35}$$

□

A.2 Proofs of Section 3.4

Proposition 3.4.1 (Empirical DMO as vector-valued regressor)**.** *The minimiser of the empirical reconstruction risk is the empirical DMO, i.e.* $\hat{D}_{X|Y} = \arg\min_{D\in\mathcal{H}_\Gamma} \hat{\mathcal{E}}_{\mathrm{d}}(D)$

Proof of Proposition 3.4.1. Let $D \in \mathcal{H}_\Gamma$, we recall the form of the regularised empirical objective

$$\hat{\mathcal{E}}_{\mathrm{d}}(D) = \frac{1}{M}\sum_{j=1}^{M} |\ell_{\tilde{y}_j} - D\hat{C}_{X|Y}\ell_{\tilde{y}_j}|_\ell^2 + \epsilon|D|_\Gamma^2 \tag{A.36}$$

By [Micchelli and Pontil, 2005, Theorem 4.1], if $\hat{D} \in \underset{D\in\mathcal{H}_\Gamma}{\arg\min}\, \hat{\mathcal{E}}_{\mathrm{d}}(D)$, then it is unique and has form

$$\hat{D} = \sum_{j=1}^{M} \Gamma_{\hat{C}_{X|Y}\ell_{\tilde{y}_j}} c_j \tag{A.37}$$

where $\Gamma_{\hat{C}_{X|Y}\ell_{\tilde{y}_j}} : \mathcal{H}_\ell \to \mathcal{H}_\Gamma$ is the vector-valued kernel Γ's feature map indexed by $\hat{C}_{X|Y}\ell_{\tilde{y}_j}$, such that for any $h \in \mathcal{H}_\Gamma$ and $g \in \mathcal{H}_\ell$, we have $\langle h, \Gamma_{\hat{C}_{X|Y}\ell_{\tilde{y}_j}} g\rangle_\Gamma = \langle h(\hat{C}_{X|Y}\ell_{\tilde{y}_j}), g\rangle_\ell$. (see Paulsen and Raghupathi [2016] for a detailed review of vector-valued RKHS). Furthermore, coefficients $c_1, \ldots, c_M \in \mathcal{H}_\ell$ are the unique solutions to

$$\sum_{i=1}^{M} \left(\Gamma(\hat{C}_{X|Y}\ell_{\tilde{y}_i}, \hat{C}_{X|Y}\ell_{\tilde{y}_j}) + M\epsilon\delta_{ij}\right) c_i = \ell_{\tilde{y}_j} \tag{A.38}$$

Since

$$\Gamma(\hat{C}_{X|Y}\ell_{\tilde{y}_i}, \hat{C}_{X|Y}\ell_{\tilde{y}_j}) = \langle \hat{C}_{X|Y}\ell_{\tilde{y}_i}, \hat{C}_{X|Y}\ell_{\tilde{y}_j}\rangle_k \,\mathrm{Id}_{\mathcal{H}_\ell} = \hat{q}(\tilde{y}_i, \tilde{y}_j)\,\mathrm{Id}_{\mathcal{H}_\ell} \tag{A.39}$$

where $\mathrm{Id}_{\mathcal{H}_\ell}$ denotes the identity operator on $\mathcal{H}_\ell$. The above simplifies as

$$\sum_{i=1}^{M} \left(\hat{q}(\tilde{y}_i, \tilde{y}_j) + M\epsilon\delta_{ij}\right) c_i = \ell_{\tilde{y}_j} \quad \forall 1 \le j \le M \tag{A.40}$$

$$\Leftrightarrow \left(\hat{\mathbf{Q}}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + M\epsilon\mathbf{I}_M\right)\mathbf{c} = \mathbf{\Psi}_{\tilde{\mathbf{y}}} \tag{A.41}$$

$$\Leftrightarrow \mathbf{c} = \left(\hat{\mathbf{Q}}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + M\epsilon\mathbf{I}_M\right)^{-1}\mathbf{\Psi}_{\tilde{\mathbf{y}}} \tag{A.42}$$

where $\mathbf{c} = \begin{bmatrix} c_1 & \ldots & c_M \end{bmatrix}$.

Since for any $f \in \mathcal{H}_k$ and $g \in \mathcal{H}_\ell$, our choice of kernel gives $\Gamma_f g = g \otimes f$, plugging (A.40) into (A.37)

we obtain

$$\hat{D} = \left[\mathbf{\Psi}_{\tilde{\mathbf{y}}}\left(\hat{\mathbf{Q}}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + M\epsilon\mathbf{I}_M\right)^{-1}\right] \otimes \left[\hat{C}_{X|Y}\mathbf{\Psi}_{\tilde{\mathbf{y}}}\right] \tag{A.43}$$

$$= \mathbf{\Psi}_{\tilde{\mathbf{y}}}\left(\hat{\mathbf{Q}}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + M\epsilon\mathbf{I}_M\right)^{-1}\mathbf{\Psi}_{\tilde{\mathbf{y}}}^{\top}\hat{C}_{X|Y}^{\top} \tag{A.44}$$

$$= \mathbf{\Psi}_{\tilde{\mathbf{y}}}\left(\hat{\mathbf{Q}}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + M\epsilon\mathbf{I}_M\right)^{-1}\mathbf{\Psi}_{\tilde{\mathbf{y}}}^{\top}\mathbf{\Psi}_{\mathbf{y}}\left(\mathbf{L}_{\mathbf{yy}} + N\lambda\mathbf{I}_N\right)^{-1}\mathbf{\Phi}_{\mathbf{x}} \tag{A.45}$$

$$= \mathbf{\Psi}_{\tilde{\mathbf{y}}}\left(\hat{\mathbf{Q}}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + M\epsilon\mathbf{I}_M\right)^{-1}\mathbf{A}\mathbf{\Phi}_{\mathbf{x}} \tag{A.46}$$

$$= \hat{D}_{X|Y} \tag{A.47}$$

which concludes the proof. □

Theorem 3.4.2 (Empirical DMO Convergence Rate)**.** *Denote $D_{\mathbb{P}_Y} = \arg\min_{D\in\mathcal{H}_\Gamma}\mathcal{E}_{\mathrm{d}}(D)$. Assume assumptions stated in Appendix A.5 are satisfied. In particular, let (b, c) and $(0, c')$ be the parameters of the restricted class of distribution for $\mathbb{P}_Y$ and $\mathbb{P}_{XY}$ respectively and let $\iota \in]0, 1]$ be the Hölder continuity exponent in $\mathcal{H}_\Gamma$. Then, if we choose $\lambda = N^{-\frac{1}{c'+1}}$, $N = M^{\frac{a(c'+1)}{\iota(c'-1)}}$ where $a > 0$, we have the following result,*

- *If $a \leq \frac{b(c+1)}{bc+1}$, then $\mathcal{E}_{\mathrm{d}}(\hat{D}_{X|Y}) - \mathcal{E}_{\mathrm{d}}(D_{\mathbb{P}_Y}) = \mathcal{O}(M^{\frac{-ac}{c+1}})$ with $\epsilon = M^{\frac{-a}{c+1}}$*
- *If $a \geq \frac{b(c+1)}{bc+1}$, then $\mathcal{E}_{\mathrm{d}}(\hat{D}_{X|Y}) - \mathcal{E}_{\mathrm{d}}(D_{\mathbb{P}_Y}) = \mathcal{O}(M^{\frac{-bc}{bc+1}})$ with $\epsilon = M^{\frac{-b}{bc+1}}$*

Proof of Theorem 3.4.2. In Appendix A.5, we present Theorem A.5.4 which is a detailed version of this result with all assumptions explicitly stated. The proof of Theorem A.5.4 constitutes the proof of this result. □

A.3 Variational formulation of the deconditional posterior

Inference computational complexity is $\mathcal{O}(M^3)$ for the posterior mean and $\mathcal{O}(N^3 + M^3)$ for the posterior covariance. To scale to large datasets, we introduce in the following a variational formulation as a scalable approximation to the deconditional posterior $f(\mathbf{x})|\tilde{\mathbf{z}}$. Without loss of generality, we assume in the following that f is centered, i.e. $m = 0$.

A.3.1 Variational formulation

We consider a set of d inducing locations $\mathbf{w} = \begin{bmatrix} w_1 & \dots & w_d \end{bmatrix}^{\top} \in \mathcal{X}^d$ and define inducing points as the gaussian vector $\mathbf{u} := f(\mathbf{w}) \sim \mathcal{N}(0, \mathbf{K}_{\mathbf{ww}})$, where $\mathbf{K}_{\mathbf{ww}} := k(\mathbf{w}, \mathbf{w})$. We set d-dimensional variational distribution $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\boldsymbol{\eta}, \mathbf{\Sigma})$ over inducing points and define $q(\mathbf{f}) := \int p(\mathbf{f}|\mathbf{u})q(\mathbf{u})\,\mathrm{d}\mathbf{u}$ as an approximation of the deconditional posterior $p(\mathbf{f}|\mathbf{z})$. The estimation of the deconditional posterior can

thus be approximated by optimising the variational distribution parameters $\boldsymbol{\eta}$, $\boldsymbol{\Sigma}$ to maximise the *evidence lower bound* (ELBO) objective given by

$$\mathrm{ELBO}(q) = \mathbb{E}_{q(\mathbf{f})}[\log p(\tilde{\mathbf{z}}|\mathbf{f})] + \mathrm{KL}(q(\mathbf{u})|p(\mathbf{u})). \tag{A.48}$$

As both q and p are Gaussians, the Kullback-Leibler divergence admits closed-form. The expected log likelihood term decomposes as

$$\mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{z}|\mathbf{f})] = -\frac{M}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\left(\mathrm{tr}\left(\mathbf{A}^\top\bar{\boldsymbol{\Sigma}}\mathbf{A}\right) + \left|\tilde{\mathbf{z}} - A^\top\bar{\boldsymbol{\eta}}\right|_2^2\right) \tag{A.49}$$

where $\bar{\boldsymbol{\eta}}$ and $\bar{\boldsymbol{\Sigma}}$ are the parameters of the posterior variational distribution $q(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\bar{\boldsymbol{\eta}}, \bar{\boldsymbol{\Sigma}})$ given by

$$\bar{\boldsymbol{\eta}} = \mathbf{K_{xw}}\mathbf{K_{ww}^{-1}}\boldsymbol{\eta} \qquad \bar{\boldsymbol{\Sigma}} = \mathbf{K_{xx}} - \mathbf{K_{xw}}\left[\mathbf{K_{ww}^{-1}} - \mathbf{K_{ww}^{-1}}\boldsymbol{\Sigma}\mathbf{K_{ww}^{-1}}\right]\mathbf{K_{wx}} \tag{A.50}$$

Given this objective, we can optimise this lower bound with respect to variational parameters $\boldsymbol{\eta}$, $\boldsymbol{\Sigma}$, noise σ^2 and parameters of kernels k and ℓ, with an option to parametrize these kernels using feature maps given by deep neural network Law et al. [2019], using a stochastic gradient approach for example. We might also want to learn the inducing locations $\mathbf{w}$.

A.3.2 Details on evidence lower bound derivation

For completeness, we provide here the derivation of the evidence lower bound objective. Let us remind its expression as stated in (A.48)

$$\mathrm{ELBO}(q) = \mathbb{E}_{q(\mathbf{f})}[\log p(\tilde{\mathbf{z}}|\mathbf{f})] - \mathrm{KL}(q(\mathbf{u})|p(\mathbf{u})) \tag{A.51}$$

The second term here is the Kullback-Leibler divergence of two gaussian densities which has a known and tractable closed-form expression.

$$\mathrm{KL}(q(\mathbf{u})|p(\mathbf{u})) = \frac{1}{2}\left[\mathrm{tr}\left(\mathbf{K_{ww}^{-1}}\boldsymbol{\Sigma}\right) + \boldsymbol{\eta}^\top\mathbf{K_{ww}^{-1}}\boldsymbol{\eta} - d + \log\frac{\det\mathbf{K_{ww}}}{\det\boldsymbol{\Sigma}}\right] \tag{A.52}$$

The first term is the expected log likelihood and needs to be derived. Using properties of integrals of

gaussian densities, we can start by showing that $q(\mathbf{f})$ also corresponds to a gaussian density which comes

$$q(\mathbf{f}) = \int p(\mathbf{f}|\mathbf{u})q(\mathbf{u})\,\mathrm{d}\mathbf{u} \tag{A.53}$$

$$= \int \mathcal{N}(\mathbf{f}|\mathbf{K_{xw}}\mathbf{K_{ww}^{-1}}\mathbf{u}, \mathbf{K_{xx}} - \mathbf{K_{xw}}\mathbf{K_{ww}^{-1}}\mathbf{K_{wx}}) \times \mathcal{N}(\mathbf{u}|\boldsymbol{\eta}, \boldsymbol{\Sigma})\,\mathrm{d}\mathbf{u} \tag{A.54}$$

$$= \mathcal{N}(\mathbf{f}|\bar{\boldsymbol{\eta}}, \bar{\boldsymbol{\Sigma}}) \tag{A.55}$$

where

$$\bar{\boldsymbol{\eta}} = \mathbf{K_{xw}}\mathbf{K_{ww}^{-1}}\boldsymbol{\eta} \tag{A.56}$$

$$\bar{\boldsymbol{\Sigma}} = \mathbf{K_{xx}} - \mathbf{K_{xw}}\left[\mathbf{K_{ww}^{-1}} - \mathbf{K_{ww}^{-1}}\boldsymbol{\Sigma}\mathbf{K_{ww}^{-1}}\right]\mathbf{K_{wx}} \tag{A.57}$$

Let's try now to obtain a closed-form expression of $\mathbb{E}_{q(\mathbf{f})}[\log p(\tilde{\mathbf{z}}|\mathbf{f})]$ on which we will be able to perform a gradient-based optimization routine. Using Gaussian conditioning on (3.4), we obtain

$$p(\tilde{\mathbf{z}}|\mathbf{f}) = \mathcal{N}(\tilde{\mathbf{z}}|\boldsymbol{\Upsilon}^\top\mathbf{K_{xx}^{-1}}\mathbf{f}\ ,\ \mathbf{Q_{\tilde{y}\tilde{y}}} + \sigma^2\mathbf{I}_M - \boldsymbol{\Upsilon}^\top\mathbf{K_{xx}}^{-1}\boldsymbol{\Upsilon}) \tag{A.58}$$

We notice that $\boldsymbol{\Upsilon}^\top\mathbf{K_{xx}^{-1}} = \ell(\tilde{\mathbf{y}}, \mathbf{y})(\mathbf{L_{yy}} + \lambda N\mathbf{I}_N)^{-1}\mathbf{K_{xx}}\mathbf{K_{xx}^{-1}} = \ell(\tilde{\mathbf{y}}, \mathbf{y})(\mathbf{L_{yy}} + \lambda N\mathbf{I}_N)^{-1} = \mathbf{A}$.

Hence we also have $\boldsymbol{\Upsilon}^\top\mathbf{K_{xx}^{-1}}\boldsymbol{\Upsilon} = \mathbf{A}^\top\mathbf{K_{xx}}\mathbf{A} = \mathbf{Q_{\tilde{y}\tilde{y}}}$.

We can thus simplify (A.58) as

$$p(\tilde{\mathbf{z}}|\mathbf{f}) = \mathcal{N}(\tilde{\mathbf{z}}|\mathbf{A}^\top\mathbf{f}, \sigma^2\mathbf{I}_n) \tag{A.59}$$

Then,

$$\log p(\tilde{\mathbf{z}}|\mathbf{f}) = -\frac{M}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left|\tilde{\mathbf{z}} - \mathbf{A}^\top\mathbf{f}\right|_2^2 \tag{A.60}$$

$$\Rightarrow \mathbb{E}_{q(\mathbf{f})}[\log p(\tilde{\mathbf{z}}|\mathbf{f})] = -\frac{M}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\mathbb{E}_{q(\mathbf{f})}\left[\left|\tilde{\mathbf{z}} - \mathbf{A}^\top\mathbf{f}\right|_2^2\right] \tag{A.61}$$

Using the trace trick to express the expectation with respect to the posterior variational parameters $\bar{\boldsymbol{\eta}}, \bar{\boldsymbol{\Sigma}}$,

we have

$$\mathbb{E}_{q(\mathbf{f})}\left[\left|\tilde{\mathbf{z}}-\mathbf{A}^\top\mathbf{f}\right|_2^2\right] = \mathbb{E}_{q(\mathbf{f})}\left[\mathrm{tr}\left(\left(\tilde{\mathbf{z}}-\mathbf{A}^\top\mathbf{f}\right)^\top\left(\tilde{\mathbf{z}}-\mathbf{A}^\top\mathbf{f}\right)\right)\right] \tag{A.62}$$

$$= \mathbb{E}_{q(\mathbf{f})}\left[\mathrm{tr}\left(\left(\tilde{\mathbf{z}}-\mathbf{A}^\top\mathbf{f}\right)\left(\tilde{\mathbf{z}}-\mathbf{A}^\top\mathbf{f}\right)^\top\right)\right] \tag{A.63}$$

$$= \mathrm{tr}\left(\mathbb{E}_{q(\mathbf{f})}\left[\left(\tilde{\mathbf{z}}-\mathbf{A}^\top\mathbf{f}\right)\left(\tilde{\mathbf{z}}-\mathbf{A}^\top\mathbf{f}\right)^\top\right]\right) \tag{A.64}$$

$$\tag{A.65}$$

And

$$\mathbb{E}_{q(\mathbf{f})}\left[\left(\tilde{\mathbf{z}}-\mathbf{A}^\top\mathbf{f}\right)\left(\tilde{\mathbf{z}}-\mathbf{A}^\top\mathbf{f}\right)^\top\right] = \mathrm{Cov}(\tilde{\mathbf{z}}-\mathbf{A}^\top\mathbf{f}) + \mathbb{E}_{q(\mathbf{f})}\left[\tilde{\mathbf{z}}-\mathbf{A}^\top\mathbf{f}\right]\mathbb{E}_{q(\mathbf{f})}\left[\tilde{\mathbf{z}}-\mathbf{A}^\top\mathbf{f}\right]^\top \tag{A.66}$$

$$= \mathbf{A}^\top\bar{\boldsymbol{\Sigma}}\mathbf{A} + \left(\tilde{\mathbf{z}}-\mathbf{A}^\top\bar{\boldsymbol{\eta}}\right)\left(\tilde{\mathbf{z}}-\mathbf{A}^\top\bar{\boldsymbol{\eta}}\right)^\top \tag{A.67}$$

Hence, it comes that

$$\mathbb{E}_{q(\mathbf{f})}\left[\left|\tilde{\mathbf{z}}-\mathbf{A}^\top\mathbf{f}\right|_2^2\right] = \mathrm{tr}\left(\mathbf{A}^\top\bar{\boldsymbol{\Sigma}}\mathbf{A}\right) + \mathrm{tr}\left(\left(\tilde{\mathbf{z}}-\mathbf{A}^\top\bar{\boldsymbol{\eta}}\right)\left(\tilde{\mathbf{z}}-\mathbf{A}^\top\bar{\boldsymbol{\eta}}\right)^\top\right) \tag{A.68}$$

$$= \mathrm{tr}\left(\mathbf{A}^\top\bar{\boldsymbol{\Sigma}}\mathbf{A}\right) + \left|\tilde{\mathbf{z}}-\mathbf{A}^\top\bar{\boldsymbol{\eta}}\right|_2^2 \tag{A.69}$$

which can be efficiently computed as it only requires diagonal terms.

Wrapping up, we obtain that

$$\mathrm{ELBO}(q) = -\frac{M}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(\mathrm{tr}\left(\mathbf{A}^\top\bar{\boldsymbol{\Sigma}}\mathbf{A}\right) + \left|\tilde{\mathbf{z}}-\mathbf{A}^\top\bar{\boldsymbol{\eta}}\right|_2^2\right) - \mathrm{KL}(q(\mathbf{u})|p(\mathbf{u})) \tag{A.70}$$

A.4 Details on Conditional Mean Shrinkage Operator

A.4.1 Deconditional posterior with Conditional Mean Shrinkage Operator

We recall from Proposition 3.3.3 that the deconditional posterior is a GP specifed by mean and covariance functions

$$m_{\mathrm{d}}(x) = m(x) + k_x^\top C_{X|Y}\boldsymbol{\Psi}_{\tilde{\mathbf{y}}}(\mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2\mathbf{I}_M)^{-1}(\tilde{\mathbf{z}} - \nu(\tilde{\mathbf{y}})), \tag{A.71}$$

$$k_{\mathrm{d}}(x,x') = k(x,x') - k_x^\top C_{X|Y}\boldsymbol{\Psi}_{\tilde{\mathbf{y}}}(\mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2\mathbf{I}_M)^{-1}\boldsymbol{\Psi}_{\tilde{\mathbf{y}}}^\top C_{X|Y}^\top k_{x'} \tag{A.72}$$

for any $x, x' \in \mathcal{X}$, where we abuse notation for the cross-covariance term

$$k_x^\top C_{X|Y} \mathbf{\Psi}_{\tilde{\mathbf{y}}} = \begin{bmatrix} \langle k_x, C_{X|Y} \ell_{\tilde{y}_1} \rangle_k & \dots & \langle k_x, C_{X|Y} \ell_{\tilde{y}_M} \rangle_k \end{bmatrix}. \tag{A.73}$$

The CMO appears in the cross-covariance term $k_x^\top C_{X|Y} \mathbf{\Psi}_{\tilde{\mathbf{y}}}$ and in the CMP covariance matrix $\mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} = \mathbf{\Psi}_{\tilde{\mathbf{y}}}^\top C_{X|Y}^\top C_{X|Y} \mathbf{\Psi}_{\tilde{\mathbf{y}}}$. To derive empirical versions using the Conditional Mean Shrinkage Operator we replace it by ${}^S\hat{C}_{X|Y} = \hat{\mathbf{M}}_{\mathbf{y}} (\mathbf{L}_{\mathbf{yy}} + \lambda N \mathbf{I}_N)^{-1} \mathbf{\Psi}_{\mathbf{y}}^\top$.

The empirical cross-covariance operator with shrinkage CMO estimate is given by

$$\begin{aligned} k_x^\top {}^S\hat{C}_{X|Y} \mathbf{\Psi}_{\tilde{\mathbf{y}}} &= k_x^\top \hat{\mathbf{M}}_{\mathbf{y}} (\mathbf{L}_{\mathbf{yy}} + \lambda N \mathbf{I}_N)^{-1} \mathbf{\Psi}_{\mathbf{y}}^\top \mathbf{\Psi}_{\tilde{\mathbf{y}}} && \text{(A.74)} \\ &= k_x^\top \hat{\mathbf{M}}_{\mathbf{y}} (\mathbf{L}_{\mathbf{yy}} + \lambda N \mathbf{I}_N)^{-1} \mathbf{L}_{\mathbf{y}\tilde{\mathbf{y}}} && \text{(A.75)} \\ &= k_x^\top \hat{\mathbf{M}}_{\mathbf{y}} \mathbf{A} && \text{(A.76)} \end{aligned}$$

where we abuse notation

$$\begin{aligned} k_x^\top \hat{\mathbf{M}}_{\mathbf{y}} &:= \begin{bmatrix} \langle k_x, \hat{\mu}_{X|Y=y_1} \rangle_k & \dots & \langle k_x, \hat{\mu}_{X|Y=y_N} \rangle_k \end{bmatrix} && \text{(A.77)} \\ &= \begin{bmatrix} \frac{1}{n_1} \sum_{i=1}^{n_1} k(x_1^{(i)}, x) & \dots & \frac{1}{n_N} \sum_{i=1}^{n_N} k(x_N^{(i)}, x) \end{bmatrix}. && \text{(A.78)} \end{aligned}$$

The empirical shrinkage CMP covariance matrix is given by

$$\begin{aligned} {}^S\hat{\mathbf{Q}}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} &:= \mathbf{\Psi}_{\tilde{\mathbf{y}}}^\top {}^S\hat{C}_{X|Y}^\top {}^S\hat{C}_{X|Y} \mathbf{\Psi}_{\tilde{\mathbf{y}}} && \text{(A.79)} \\ &= \mathbf{\Psi}_{\tilde{\mathbf{y}}}^\top \mathbf{\Psi}_{\mathbf{y}} (\mathbf{L}_{\mathbf{yy}} + \lambda N \mathbf{I}_N)^{-1} \hat{\mathbf{M}}_{\mathbf{y}}^\top \hat{\mathbf{M}}_{\mathbf{y}} (\mathbf{L}_{\mathbf{yy}} + \lambda N \mathbf{I}_N)^{-1} \mathbf{\Psi}_{\mathbf{y}}^\top \mathbf{\Psi}_{\tilde{\mathbf{y}}} && \text{(A.80)} \\ &= \mathbf{A}^\top \hat{\mathbf{M}}_{\mathbf{y}}^\top \hat{\mathbf{M}}_{\mathbf{y}} \mathbf{A} && \text{(A.81)} \end{aligned}$$

where with similar notation abuse

$$\hat{\mathbf{M}}_{\mathbf{y}}^\top \hat{\mathbf{M}}_{\mathbf{y}} = \begin{bmatrix} \langle \hat{\mu}_{X|Y=y_i}, \hat{\mu}_{X|Y=y_j} \rangle_k \end{bmatrix}_{1 \le i,j \le N} = \left[\frac{1}{n_i n_j} \sum_{l=1}^{n_i} \sum_{r=1}^{n_j} k(x_i^{(l)}, x_j^{(r)}) \right]_{1 \le i,j \le N} \tag{A.82}$$

Substituting the latters into (A.71) and (A.72), we obtain empirical estimates of the deconditional posterior with shrinkage CMO estimator defined as

$$\begin{aligned} {}^S\hat{m}_{\mathrm{d}}(x) &:= m(x) + k_x^\top \hat{\mathbf{M}}_{\mathbf{y}} \mathbf{A} (\mathbf{A}^\top \hat{\mathbf{M}}_{\mathbf{y}}^\top \hat{\mathbf{M}}_{\mathbf{y}} \mathbf{A} + \sigma^2 \mathbf{I}_M)^{-1} (\tilde{\mathbf{z}} - \hat{\mu}(\tilde{\mathbf{y}})), && \text{(A.83)} \\ {}^S\hat{k}_{\mathrm{d}}(x, x') &:= k(x, x') - k_x^\top \hat{\mathbf{M}}_{\mathbf{y}} \mathbf{A} (\mathbf{A}^\top \hat{\mathbf{M}}_{\mathbf{y}}^\top \hat{\mathbf{M}}_{\mathbf{y}} \mathbf{A} + \sigma^2 \mathbf{I}_M)^{-1} \mathbf{A}^\top \hat{\mathbf{M}}_{\mathbf{y}}^\top k_{x'} && \text{(A.84)} \end{aligned}$$

for any $x, x' \in \mathcal{X}$.

Note that as the number of bags increases, it is possible to derive a variational formulation similar to the one proposed in Section A.3 that leverages the shrinkage estimator to further speed up the overall computation.

A.4.2 Ablation Study

In this section we will present an ablation study on the shrinkage CMO estimator. The key is to illustrate that the Shrinkage CMO performs on par with the standard CMO estimator but is much faster to compute.

In the following, we will sample bag data of the form ${}^b\mathcal{D} = \{{}^b\boldsymbol{x}_j, y_j\}_{j=1}^N$ and ${}^b\boldsymbol{x}_j = \{x_j^{(i)}\}_{i=1}^n$, i.e there are N bags with n elements inside each. We first sample N bag labels $y_j \sim \mathcal{N}(0, 2)$ and for each bag y_j, we sample n observations $x_j^{(i)}|y_j \sim \mathcal{N}(y_j \sin(y_j), 0.5^2)$.

Recall in standard CME one would need to repeat the number of bag labels to match the cardinality of $x_j^{(i)}$, i.e estimating CME using data $\{x_j^{(i)}, y_j\}_{j=1,i=1}^{N,n}$.

Denote $\hat{C}_{X|Y}$ as the standard CMO estimator and ${}^S\hat{C}_{X|Y}$ as the shrinkage CMO estimator. We will compare the RMSE between the two estimator when tested on a grid of test points $\{x_i^*, y_i^*\}_{i=1}^{N^*}$, i.e comparing the RMSE of the values between $\hat{\mu}_{X|Y=y_i^*}(x_i^*) := \langle \hat{C}_{X|Y}\ell_{y_i^*}, k_{x_i^*}\rangle_k$ and ${}^S\hat{\mu}_{X|Y=y_i^*}(x_i^*) := \langle {}^S\hat{C}_{X|Y}\ell_{y_i^*}, k_{x_i^*}\rangle_k$ for each i. We also report the time in seconds needed to compute the estimator. The following results are ran on a CPU. Kernel hyperparameters are chosen using the median heuristic. The regularisation for both estimator is set to 0.1.

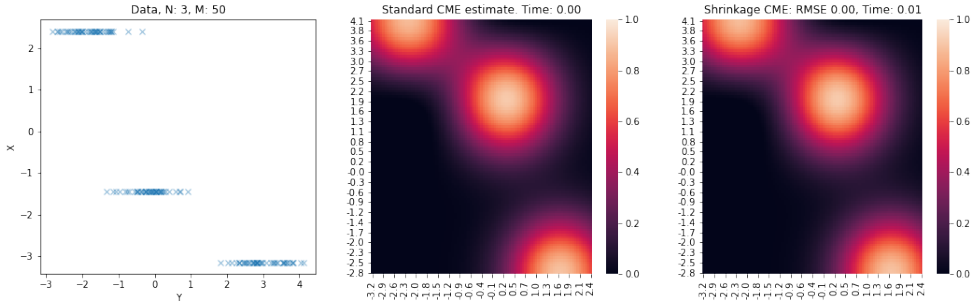


Figure A.1: 3 bags with 50 samples each. (left) Data, (middle) $\hat{\mu}_{X|Y=y_i^*}(x_i^*)$ Standard CME. (right) ${}^S\hat{\mu}_{X|Y=y_i^*}(x_i^*)$ Shrinkage CME. We see both algorithms require very little time to train, (~ 0.01second) with a negligible difference in values as shown by the RMSE.

Figures A.1 and A.2 show how shrinkage CMO performed compared to the standard CMO in a small data regime. Now when we increase the data size, we will start to see the major computational differences. (See Figures A.3 and A.4)

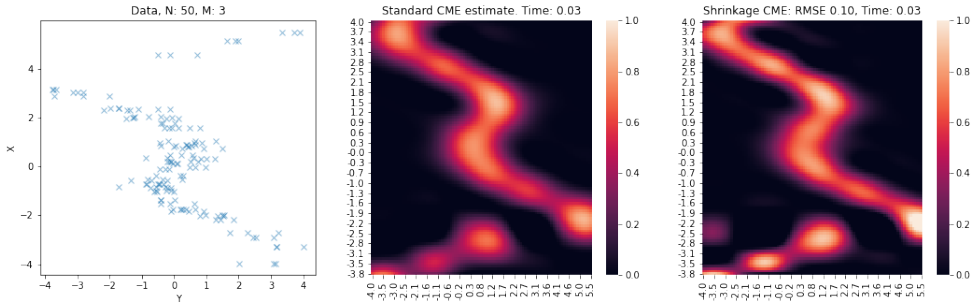


Figure A.2: 50 bags with 3 samples each. (left) Data, (middle) $\hat{\mu}_{X|Y=y_i^*}(x_i^*)$ Standard CME. (right) ${}^S\hat{\mu}_{X|Y=y_i^*}(x_i^*)$ Shrinkage CME. Again, we see both algorithms require very little time to train, (~ 0.03 second). However, there is an increase in RMSE for the shrinkage estimator because there are much less samples for each bag, thus the empirical CME estimate $\hat{\mu}_{X|Y=y_j}$ might not be accurate. Nonetheless, it is still a small difference.

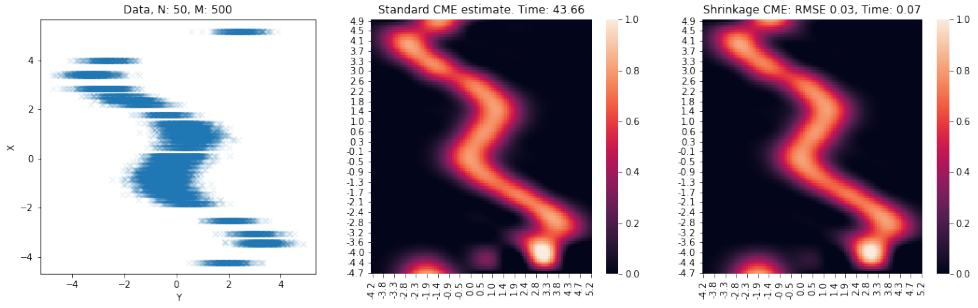


Figure A.3: 50 bags with 500 samples each. (left) Data, (middle) $\hat{\mu}_{X|Y=y_i^*}(x_i^*)$ Standard CME. (right) ${}^S\hat{\mu}_{X|Y=y_i^*}(x_i^*)$ Shrinkage CME. With a small RMSE of 0.03, the Shrinkage CME is approximately 600 times quicker than the standard version.

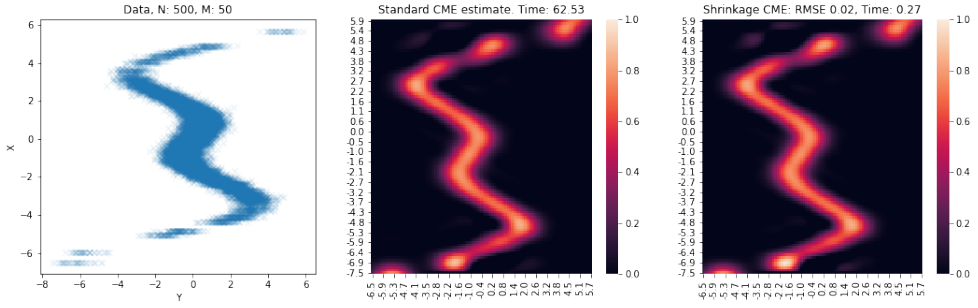


Figure A.4: 500 bags with 50 samples each. (left) Data, (middle) $\hat{\mu}_{X|Y=y_i^*}(x_i^*)$ Standard CME. (right) ${}^S\hat{\mu}_{X|Y=y_i^*}(x_i^*)$ Shrinkage CME. Again, with a small RMSE of 0.02, Shrinakge CME is approximately 200 times quicker than the standard CME.

A.5 Details on Convergence Result

In this section, we provide insights about the convergence results stated in Section 3.4. These results are largely based on the impactful work of Caponnetto and De Vito [2007], Szabó et al. [2016] and Singh et al. [2019] which we modify to fit our problem setup. Each assumption that we make is adapted from a similar assumption made in those works, for which we provide intuition and a detailed justification. We start by redefining the mathematical tools introduced in these works that are necessary to state our result.

A.5.1 Definitions and $\mathcal{P}_K(b, c)$ spaces

We start by providing a general definition of covariance operators over vector-valued RKHS, which will allow us to specify a class of probability distributions for our convergence result.

Definition A.5.1 (Covariance operator)**.** *Let $\mathcal{W}$ a Polish space endowed with measure ρ, $\mathcal{G}$ a real separable Hilbert space and $K : \mathcal{W}^2 \to \mathcal{L}(\mathcal{G})$ an operator-valued kernel spanning a $\mathcal{G}$-valued RKHS $\mathcal{H}_K$.*

The covariance operator of K is defined as the positive trace class operator given by

$$T_K := \int_{\mathcal{Z}} K_w K_w^* d\rho(w) \in \mathcal{L}(\mathcal{H}_K) \tag{A.85}$$

where $\mathcal{L}(\mathcal{H}_K)$ denotes the space of bounded linear operators over $\mathcal{H}_k$.

Definition A.5.2 (Power of self-adjoint Hilbert operator)**.** *Let T a compact self-adjoint Hilbert space operator with spectral decomposition $T = \sum_{n=1}^{\infty} \lambda_n e_n \otimes e_n$ on $(e_n)_{n\in\mathbb{N}}$ basis of $\mathrm{Ker}(T)^{\perp}$. The r^{th} power of T is defined as $T^r = \sum_{n=1}^{\infty} \lambda_n^r e_n \otimes e_n$.*

Using the covariance operator, we now introduce a general class of priors that does not assume parametric distributions, by adapting to our setup a definition originally introduced by Caponnetto and De Vito [2007]. This class captures the difficulty of a regression problem in terms of two simple parameters, b and c [Szabó et al., 2016].

Definition A.5.3 ($\mathcal{P}_K(b,c)$ class)**.** *Let $\mathcal{E}_\rho : \mathcal{G}^{\mathcal{Z}} \to [0,\infty[$ an expected risk function over ρ and $E_\rho = \arg\min \mathcal{E}_\rho$. Then given $b > 1$ and $c \in]1,2]$, we say that ρ is a $\mathcal{P}_K(b,c)$ class probability measure w.r.t. $\mathcal{E}_\rho$ if*

1. Range assumption: $\exists G \in \mathcal{H}_K$ *such that* $E_\rho = T_K^{\frac{c-1}{2}} \circ G$ *with* $|G|_K^2 \leq R$ *for some* $R \geq 0$
2. Spectral assumption: *the eigenvalues* $(\lambda_n)_{n\in\mathbb{N}}$ *of* T_K *satisfy* $\alpha \leq n^b \lambda_n \leq \beta\,, \forall n \in \mathbb{N}$ *for some* $\beta \geq \alpha \geq 0$

The range assumption controls the functional smoothness of E_ρ as larger c corresponds to increased smoothness. Specifically, elements of $\mathrm{Range}(T_K^{\frac{c-1}{2}})$ admit Fourier coefficients $(\gamma_n)_{n\in\mathbb{N}}$ such that $\sum_{n=1}^{\infty} \gamma_n^2 \lambda_n^{-(c+1)} < \infty$. In the limit $c \to 1$, we obtain $\mathrm{Range}(T_K^0) = \mathrm{Range}(\mathrm{Id}_{\mathcal{H}_K}) = \mathcal{H}_K$. Since ranked eigenvalues are positive and $\lambda_n \to 0$, greater power of the covariance operator T_K give rise to faster decay of the Fourier coefficients and hence smoother operators.

The spectral assumptions can be read as a polynomial decay over the eigenvalues of T_K. Thus, larger b leads to enhanced decay $\lambda_n = \Theta(n^{-b})$ and concretely in a smaller effective input dimension.

A.5.2 Complete statement of the convergence result

The following result corresponds to a detailed version of Theorem 3.4.2 where all the assumptions are explicitly stated. As such, its proof also constitutes the proof for Theorem 3.4.2.

Theorem A.5.4 (Empirical DMO Convergence Rate)**.** *Assume that*

1. $\mathcal{X}$ *and* $\mathcal{Y}$ *are Polish spaces, i.e. separable and completely metrizable topoligical spaces*
2. k *and* ℓ *are continuous, bounded, their canonical feature maps* k_x *and* ℓ_y *are measurable and* k *is characteristic*
3. $\mathcal{H}_\ell$ *is finite dimensional*
4. $\arg\min \mathcal{E}_\mathrm{c} \in \mathcal{H}_\Gamma$ *and* $\arg\min \mathcal{E}_\mathrm{d} \in \mathcal{H}_\Gamma$
5. *The operator family* $\{\Gamma_{\mu_{X|Y=y}}\}_{y\in\mathcal{Y}}$ *is Hölder continuous with exponent* $\iota \in]0,1]$
6. $\mathbb{P}_{XY}$ *is a* $\mathcal{P}_\Gamma(0,c')$ *class probability measure w.r.t.* $\mathcal{E}_\mathrm{c}$ *and* $\mathbb{P}_Y$ *is a* $\mathcal{P}_\Gamma(b,c)$ *class probability measure w.r.t.* $\mathcal{E}_\mathrm{d}$
7. $\forall g \in \mathcal{H}_\ell$, $|g|_\ell < \infty$ *almost surely*

Let $D_{\mathbb{P}_Y} = \underset{D\in\mathcal{H}_\Gamma}{\arg\min}\, \mathcal{E}_\mathrm{d}(D)$. *Then, if we choose* $\lambda = N^{-\frac{1}{c'+1}}$ *and* $N = M^{\frac{a(c'+1)}{\iota(c'-1)}}$ *where* $a > 0$, *we have*

- *If* $a \leq \frac{b(c+1)}{bc+1}$, *then* $\mathcal{E}_\mathrm{d}(\hat{D}_{X|Y}) - \mathcal{E}_\mathrm{d}(D_{\mathbb{P}_Y}) = \mathcal{O}(M^{\frac{-ac}{c+1}})$ *with* $\epsilon = M^{\frac{-a}{c+1}}$
- *If* $a \geq \frac{b(c+1)}{bc+1}$, *then* $\mathcal{E}_\mathrm{d}(\hat{D}_{X|Y}) - \mathcal{E}_\mathrm{d}(D_{\mathbb{P}_Y}) = \mathcal{O}(M^{\frac{-bc}{bc+1}})$ *with* $\epsilon = M^{\frac{-b}{bc+1}}$

Proof of Theorem 3.4.2. The main objective here will be to rigorously verify that within our setup, the conditions in Theorem 4 from Singh et al. [2019] are met. We reformulate from our problem perspective each of the assumptions stated by Singh et al. [2019] and verify they are satisfied.

Assumption 1 *Assume observation model $\tilde{Z} = f(X) + \tilde{\varepsilon}$, with $\mathbb{E}[\tilde{\varepsilon}|Y] = 0$ and suppose $\mathbb{P}_{X|Y=y}$ is not constant in y.*

In this work, the observation model considered is $Z = \mathbb{E}[f(X)|Y] + \varepsilon$ and the objective is to recover the underlying random variable $f(X)$ which noisy conditional expectation is observed. The latter presumes that we could bring Z to X's resolution. We can model it by introducing "pre-aggregation" observation model $\tilde{Z} = f(X) + \tilde{\varepsilon}$ such that $Z = \mathbb{E}[\tilde{Z}|Y]$ and $\tilde{\varepsilon}$ is a noise term at individual level satisfying $\mathbb{E}[\tilde{\varepsilon}|Y] = 0$.

Assumption 2 *$\mathcal{X}$ and $\mathcal{Y}$ are Polish spaces.*

We also make this assumption.

Assumption 3 *k and ℓ are continuous and bounded, their canonical feature maps are measurable and k is characteristic.*

We make the same assumptions. The separability of $\mathcal{X}$ and $\mathcal{Y}$ along with continuity assumptions on kernels allow to propagate separability to their associated RKHS $\mathcal{H}_k$ and $\mathcal{H}_\ell$ and to the vector-valued RKHS $\mathcal{H}_\Gamma$. Boundedness and continuity on kernels ensure the measurability of the CMO and hence that measures on $\mathcal{X}$ and cY can be extended to $\mathcal{H}_k$ and $\mathcal{H}_\ell$. The assumption on k being characteristic ensures that conditional mean embeddings $\mu_{X|Y=y}$ uniquely embed conditional distributions $\mathbb{P}_{X|Y=y}$ and henceforth operators over $\mathcal{H}_\ell$ are identified.

Assumption 4 $\arg\min \mathcal{E}_{\mathrm{c}} \in \mathcal{H}_\Gamma$.

This property stronger is than what the actual conditional mean operator needs to satisfy, but it is necessary to make sure the problem is well-defined. We also make this assumption.

Assumption 5 *$\mathbb{P}_{XY}$ is a $\mathcal{P}_\Gamma(0, c')$ class probability measure, with $c' \in]1, 2]$*

As explained by Singh et al. [2019], this is further required to bound the approximation error which we also make. Through the definition of the $\mathcal{P}_\Gamma(0, c')$ class, this hypothesis assumes the existence of a probability measure over $\mathcal{H}_k$ we denote $\mathbb{P}_{\mathcal{H}_k}$. Since $\mathcal{H}_k$ is Polish (proof below), the latter can be constructed as an extension of $\mathbb{P}_X$ over the Borel σ-algebra associated to $\mathcal{H}_k$ [Steinwart and Christmann, 2008b, Lemma A.3.16].

Assumption 6 *$\mathcal{H}_k$ is a Polish space*

Since k is continuous and $\mathcal{X}$ is separable, $\mathcal{H}_k$ is a separable Hilbert space which makes it Polish.

Assumption 7 *The $\{\Gamma_{\mu_{X|Y=y}}\}_{y\in\mathcal{Y}}$ operator family is*

- Uniformly bounded in Hilbert-Schmidt norm, i.e. $\exists B > 0$ such that $\forall y \in \mathcal{Y}$, $|\Gamma_{\mu_{X|Y=y}}|^2_{\mathrm{HS}(\mathcal{H}_\ell,\mathcal{H}_\Gamma)} \leq B$
- Hölder continuous in operator norm, i.e. $\exists L > 0, \iota \in]0, 1]$ such that $\forall y, y' \in \mathcal{Y}$, $|\Gamma_{\mu_{X|Y=y}} - \Gamma_{\mu_{X|Y=y'}}|_{\mathcal{L}(\mathcal{H}_\ell,\mathcal{H}_\Gamma)} \leq L|\mu_{X|Y=y} - \mu_{X|Y=y'}|^\iota_k$

where $\mathcal{L}(\mathcal{H}_\ell, \mathcal{H}_\Gamma)$ denotes the space of bounded linear operator between $\mathcal{H}_\ell$ and $\mathcal{H}_\Gamma$.

Since we assume finite dimensionality of $\mathcal{H}_\ell$, we make a stronger assumption than the boundedness in Hilbert-Schmidt norm which we obtain as

$$|\Gamma_{\mu_{X|Y=y}}|^2_{\mathrm{HS}(\mathcal{H}_\ell,\mathcal{H}_\Gamma)} = \mathrm{tr}\left(\Gamma(\mu_{X|Y=y}, \mu_{X|Y=y})\right) \tag{A.86}$$

$$= \mathrm{tr}\left(\langle\mu_{X|Y=y}, \mu_{X|Y=y}\rangle_k \,\mathrm{Id}_{\mathcal{H}_\ell}\right) \tag{A.87}$$

$$= q(y, y)\,\mathrm{tr}\left(\mathrm{Id}_{\mathcal{H}_\ell}\right) < \infty. \tag{A.88}$$

Hölder continuity is a mild assumption commonly satisfied as stated in Szabó et al. [2016].

Assumption 8 $\arg\min \mathcal{E}_\mathrm{d} \in \mathcal{H}_\Gamma$ *and $\mathcal{H}_\ell$ is a space of bounded functions almost surely*

We assume that the true minimiser of $\mathcal{E}_\mathrm{d}$ is in $\mathcal{H}_\Gamma$ to have a well-defined problem. The second assumption here is expressed in terms of probability measure $\mathbb{P}_{\mathcal{H}_\ell}$ over $\mathcal{H}_\ell$. We do also assume that there exists $B > 0$ such that $\forall g \in \mathcal{H}_\ell$, $|g|_\ell < B$ $\mathbb{P}_{\mathcal{H}_\ell}-$ almost surely.

Assumption 9 $\mathbb{P}_Y$ *is a $\mathcal{P}_\Gamma(b, c)$ class probability measure, with $b > 1$ and $c \in]1, 2]$*

This last hypothesis is not required per se to obtain a bound on the excess error of regularized estimate $\hat{D}_{X|Y}$. However, it allows to simplify the bounds and state them in terms of parameters b and c which characterize efficient input size and functional smoothness respectively.

Furthermore, a premise to this assumption is the existence of a probability measure over $\mathcal{H}_\ell$ that we denote $\mathbb{P}_{\mathcal{H}_\ell}$. Since ℓ is continuous and $\mathcal{Y}$ separable, it makes $\mathcal{H}_\ell$ a separable and thus Polish. We can then construct $\mathbb{P}_{\mathcal{H}_\ell}$ by extension of $\mathbb{P}_Y$ [Steinwart and Christmann, 2008b, Lemma A.3.16] □

This theorem underlines a trade-off between the computational and statistical efficiency w.r.t. the datasets cardinalities $N = |\mathcal{D}_1|$ and $M = |\mathcal{D}_2|$ and the problem difficulty (b, c, c').

For $a \leq \frac{b(c+1)}{bc+1}$, smaller a means less samples from $\mathcal{D}_1$ at fixed M and thus computational savings. But it also hampers convergence, resulting in reduced statistical efficiency. At $a = \frac{b(c+1)}{bc+1} < 2$, convergence rate is a minimax computational-statistical efficiency optimal, i.e. convergence rate is optimal with smallest possible M. We note that at this optimal, $N > M$ and hence we require less samples from $\mathcal{D}_2$. $a \geq \frac{b(c+1)}{bc+1}$ does not improve the convergence rate but increases the size of $\mathcal{D}_1$ and hence the computational cost it bears.

We also note that larger Hölder exponents ι, which translates in smoother kernels, leads to reduced N. Similarly, since $c' \mapsto \frac{c'+1}{c-1}$ and $c \mapsto \frac{b(c+1)}{bc+1}$ are strictly decreasing functions over $]1, 2]$, stronger range assumptions regularity which means smoother operators reduces the number of sample needed from $\mathcal{D}_1$ to achieve minimax optimality. Smoother problems do hence require fewer samples.

Larger spectral decay exponent b translate here in requiring more samples to reach minimax optimality and undermines optimal convergence rate. Hence problems with smaller effective input dimension are harder to solve and require more samples and iterations.

A.6 Additional Experimental Results

A.6.1 Swiss Roll Experiment

A.6.1.1 Statistical significance table

A.6.1.2 Compute and Resources Specifications

Computations for all experiments were carried out on an internal cluster. We used a single GeForce GTX 1080 Ti GPU to speed up computations and conduct each experiment with multiple initialisation seeds. We underline however that the experiment does not require GPU acceleration and can be performed on CPU in a timely manner.

A.6.2 CMP with high-resolution noise observation model

Deconditional posterior with high-resolution noise

Beyond observation noise on the aggregate observations $\tilde{\mathbf{z}}$ as introduced in Section 3.3.2, it is natural to also consider observing noise at the high-resolution level, i.e. noises placed on f level directly in addition to the one g at aggregate level. Let $\xi \sim \mathcal{GP}(0, \delta)$ the zero-mean Gaussian process with covariance

Table A.1: p-values from a two-tailed Wilcoxon signed-rank test between all pairs of methods for the test RMSE of the swiss-roll experiment with a direct and indirect matching setup. The null hypothesis is that scores samples come from the same distribution. We only present the lower triangular matrix of the table for clarity of reading.

Matching		CMP	BAGG-GP	VARCMP	VBAGG	GPR	S-CMP
Direct	CMP	-	-	-	-	-	-
	BAGG-GP	0.00006	-	-	-	-	-
	VARCMP	0.00008	0.00006	-	-	-	-
	VBAGG	0.00006	0.00006	0.005723	-	-	-
	GPR	0.00006	0.00006	0.00006	0.00006	-	-
	S-CMP	0.00006	0.00006	0.000477	0.014269	0.00006	-
Indirect	CMP	-	-	-	-	-	-
	BAGG-GP	0.011129	-	-	-	-	-
	VARCMP	0.001944	0.015240	-	-	-	-
	VBAGG	0.000089	0.047858	0.000089	-	-	-
	GPR	0.025094	0.047858	0.047858	0.851925	-	-
	S-CMP	0.000089	0.002821	0.000089	0.000140	0.052222	-

function

$$\delta : \left| \begin{array}{ccc} \mathcal{X} \times \mathcal{X} & \longrightarrow & \mathbb{R} \\ (x, x') & \longmapsto & \begin{cases} 1 \text{ if } x = x' \\ 0 \text{ else} \end{cases} \end{array} \right. . \tag{A.89}$$

By incorporating this gaussian noise process in the integrand, we can replace the definition of the CMP by

$$g(y) = \int_{\mathcal{X}} (f(x) + \varsigma\xi(x)) \, \mathrm{d}\mathbb{P}_{X|Y=y} \,, \quad \forall y \in \mathcal{Y}, \tag{A.90}$$

where $\varsigma > 0$ is the high-resolution noise standard deviation parameter. Essentially, this amounts to consider a contaminated covariance for the HR observation process. This covariance is defined as

$$k^{\varsigma} : \left| \begin{array}{ccc} \mathcal{X} \times \mathcal{X} & \longrightarrow & \mathbb{R} \\ (x, x') & \longmapsto & k(x, x') + \varsigma^2 \delta(x, x') \end{array} \right. . \tag{A.91}$$

Provided the same regularity assumptions as in Proposition 3.3.2, the covariance of the CMP becomes $q(y, y') = \mathbb{E}[k^{\varsigma}(X, X')|Y = y, Y' = y']$ — the mean and cross-covariance terms are not affected. Similarly be written in terms of conditional mean embeddings, but using as an integrand for the CMEs the canonical feature maps induced by k^{ς}, i.e. $\mu^{\varsigma}_{X|Y=y} := \mathbb{E}[k^{\varsigma}(X, \cdot)|Y = y]$ for any $y \in \mathcal{Y}$. Critically, this is reflected in the expression of the empirical CMP covariance which writes

$$\hat{q}(y, y') = \ell(y, \mathbf{y})(\mathbf{L_{yy}} + N\lambda\mathbf{I}_N)^{-1}(\mathbf{K_{xx}} + \varsigma^2\mathbf{I}_N)(\mathbf{L_{yy}} + N\lambda\mathbf{I}_N)^{-1}\ell(\mathbf{y}, y') \tag{A.92}$$

thus, yielding matrix form

$$\hat{\mathbf{Q}}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} := \hat{q}(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}) \tag{A.93}$$

$$= \mathbf{L}_{\tilde{\mathbf{y}}\mathbf{y}}(\mathbf{L}_{\mathbf{yy}} + N\lambda\mathbf{I}_N)^{-1}(\mathbf{K}_{\mathbf{xx}} + \varsigma^2\mathbf{I}_N)(\mathbf{L}_{\mathbf{yy}} + N\lambda\mathbf{I}_N)^{-1}\mathbf{L}_{\mathbf{y}\tilde{\mathbf{y}}} \tag{A.94}$$

$$= \mathbf{A}^\top(\mathbf{K}_{\mathbf{xx}} + \varsigma^2\mathbf{I}_N)\mathbf{A}. \tag{A.95}$$

which can readily be used in (3.7) and (3.8) to compute the deconditional posterior.

This high-resolution noise term introduces an additional regularization to the model that helps preventing degeneracy of the deconditional posterior covariance. Indeed, we have

$$\hat{k}_{\mathrm{d}}(\mathbf{x}, \mathbf{x}) = \mathbf{K}_{\mathbf{xx}} - \mathbf{K}_{\mathbf{xx}}\mathbf{A}(\hat{\mathbf{Q}}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2\mathbf{I}_M)^{-1}\mathbf{A}^\top\mathbf{K}_{\mathbf{xx}} \tag{A.96}$$

$$= \mathbf{K}_{\mathbf{xx}} - \mathbf{K}_{\mathbf{xx}}\mathbf{A}(\mathbf{A}^\top(\mathbf{K}_{\mathbf{xx}} + \varsigma^2\mathbf{I}_N)\mathbf{A} + \sigma^2\mathbf{I}_M)^{-1}\mathbf{A}^\top\mathbf{K}_{\mathbf{xx}} \tag{A.97}$$

$$= \mathbf{K}_{\mathbf{xx}} - \mathbf{K}_{\mathbf{xx}}(\mathbf{A}\mathbf{A}^\top(\mathbf{K}_{\mathbf{xx}} + \varsigma^2\mathbf{I}_N) + \sigma^2\mathbf{I}_M)^{-1}(\mathbf{A}\mathbf{A}^\top\mathbf{K}_{\mathbf{xx}}). \tag{A.98}$$

where on the last line we have used the Woodburry identity. We can see that when $\sigma = \varsigma = 0$, (A.98) degenerates to 0. The aggregate observation model noise σ provides a first layer of regularization at low-resolution. The high-resolution noise ς supplements it, making for a more stable numerical compuation for the empirical covariance matrix.

Variational deconditional posterior with high-resolution noise

The high-resolution noise observation process can also be incorporated into the variational derivation to obtain a slightly different ELBO objective. We have

$$p(\tilde{\mathbf{z}}|\mathbf{f}) = \mathcal{N}(\tilde{\mathbf{z}}|\boldsymbol{\Upsilon}^\top\mathbf{K}_{\mathbf{xx}}^{-1}\mathbf{f}\ ,\ \mathbf{Q}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \sigma^2\mathbf{I}_M - \boldsymbol{\Upsilon}^\top{\mathbf{K}_{\mathbf{xx}}}^{-1}\boldsymbol{\Upsilon}) \tag{A.99}$$

$$= \mathcal{N}(\tilde{\mathbf{z}}|\mathbf{A}\mathbf{f}\ ,\ \mathbf{A}^\top(\mathbf{K}_{\mathbf{xx}} + \varsigma^2\mathbf{I}_N)\mathbf{A} + \sigma^2\mathbf{I}_M - \mathbf{A}^\top\mathbf{K}_{\mathbf{xx}}\mathbf{A}) \tag{A.100}$$

$$= \mathcal{N}(\tilde{\mathbf{z}}|\mathbf{A}\mathbf{f}\ ,\ \varsigma^2\mathbf{A}^\top\mathbf{A} + \sigma^2\mathbf{I}_M) \tag{A.101}$$

The expected loglikelihood with respect to the variational posterior hence writes

$$\mathbb{E}_{q(\mathbf{f})}[p(\tilde{\mathbf{z}}|\mathbf{f})] = -\frac{M}{2}\log(2\pi) - \frac{1}{2}\log\det(\varsigma^2\mathbf{A}^\top\mathbf{A} + \sigma^2\mathbf{I}_M) \tag{A.102}$$

$$-\frac{1}{2}\mathbb{E}_{q(\mathbf{f})}\left[(\tilde{\mathbf{z}} - \mathbf{A}^\top\mathbf{f})^\top(\varsigma^2\mathbf{A}^\top\mathbf{A} + \sigma^2\mathbf{I}_M)^{-1}(\tilde{\mathbf{z}} - \mathbf{A}^\top\mathbf{f})\right] \tag{A.103}$$

With a derivation similar to the one proposed in Appendix A.3, the expected loglikelihood can be expressed

in terms of the posterior variational parameters as

$$\mathbb{E}_{q(\mathbf{f})}[p(\tilde{\mathbf{z}}|\mathbf{f})] = -\frac{M}{2}\log(2\pi) - \frac{1}{2}\log\det(\varsigma^2\mathbf{A}^\top\mathbf{A} + \sigma^2\mathbf{I}_M) \tag{A.104}$$

$$-\frac{1}{2}(\tilde{\mathbf{z}} - \mathbf{A}^\top\bar{\boldsymbol{\eta}})^\top(\varsigma^2\mathbf{A}^\top\mathbf{A} + \sigma^2\mathbf{I}_M)^{-1}(\tilde{\mathbf{z}} - \mathbf{A}^\top\bar{\boldsymbol{\eta}}) \tag{A.105}$$

$$-\frac{1}{2}\operatorname{tr}\left((\varsigma^2\mathbf{A}^\top\mathbf{A} + \sigma^2\mathbf{I}_M)^{-1}\mathbf{A}^\top\bar{\boldsymbol{\Sigma}}\mathbf{A}\right) \tag{A.106}$$

In particular, the last term can be rearranged into $\operatorname{tr}\left(\bar{\boldsymbol{\Sigma}}^{1/2}\mathbf{A}(\varsigma^2\mathbf{A}^\top\mathbf{A} + \sigma^2\mathbf{I}_M)^{-1}\mathbf{A}^\top\bar{\boldsymbol{\Sigma}}^{1/2}\right)$ which can efficiently be computed as an inverse quadratic form Gardner et al. [2018].

A.6.3 Mediated downscaling of atmospheric temperature

A.6.3.1 Map visualization of atmospheric fields dataset

Downscaling prediction maps

A.6.3.2 Statistical significance table

Table A.2: p-values from a two-tailed Wilcoxon signed-rank test between all pairs of methods for the evaluation scores on the mediated statistical downscaling experiment. The null hypothesis is that scores samples come from the same distribution. As before, we only present the lower-traingular table for clarity of reading.

Metric		VARCMP	VBAGG	VARGPR
RMSE	VARCMP	-	-	-
	VBAGG	0.005062	-	-
	VARGPR	0.006910	0.046853	-
MAE	VARCMP	-	-	-
	VBAGG	0.005062	-	-
	VARGPR	0.059336	0.006910	-
CORR	VARCMP	-	-	-
	VBAGG	0.005062	-	-
	VARGPR	0.016605	0.028417	-
SSIM	VARCMP	-	-	-
	VBAGG	0.005062	-	-
	VARGPR	0.959354	0.005062	-

Compute and Resources Specifications

Computations for all experiments were carried out on an internal cluster. We used a single GeForce GTX 1080 Ti GPU to speed up computations and conduct each experiment with multiple initialisation seeds.

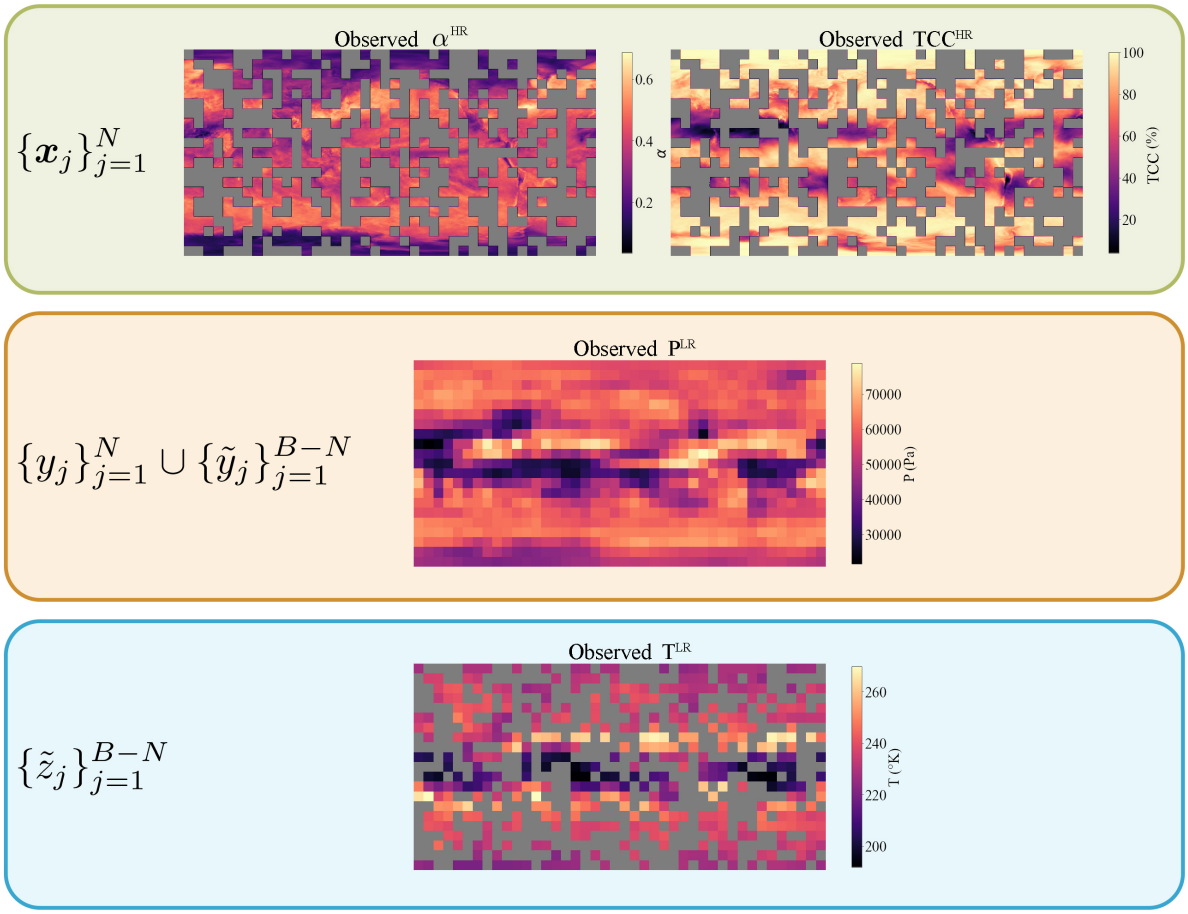


Figure A.5: Map visualization of the dataset used in the mediated downscaling experiment (for one random seed); **Top:** Bags of high-resolution albedo α^{HR} and total cloud cover TCC^{HR} pixels which are observed in $\mathcal{D}_1$ — each "coarse pixel" delineates a bag of HR pixels; **Middle:** Low-resolution pressure field P^{LR} which is observed everywhere and plays the role of mediating variable; **Bottom:** Low-resolution temperature field T^{LR} pixels which are observed in $\mathcal{D}_2$ and that we want to downscale; grey pixels are unobserved; the grey layer on HR covariates maps (top) is the exact complementary of the grey layer on the observed T^{LR} map (bottom).

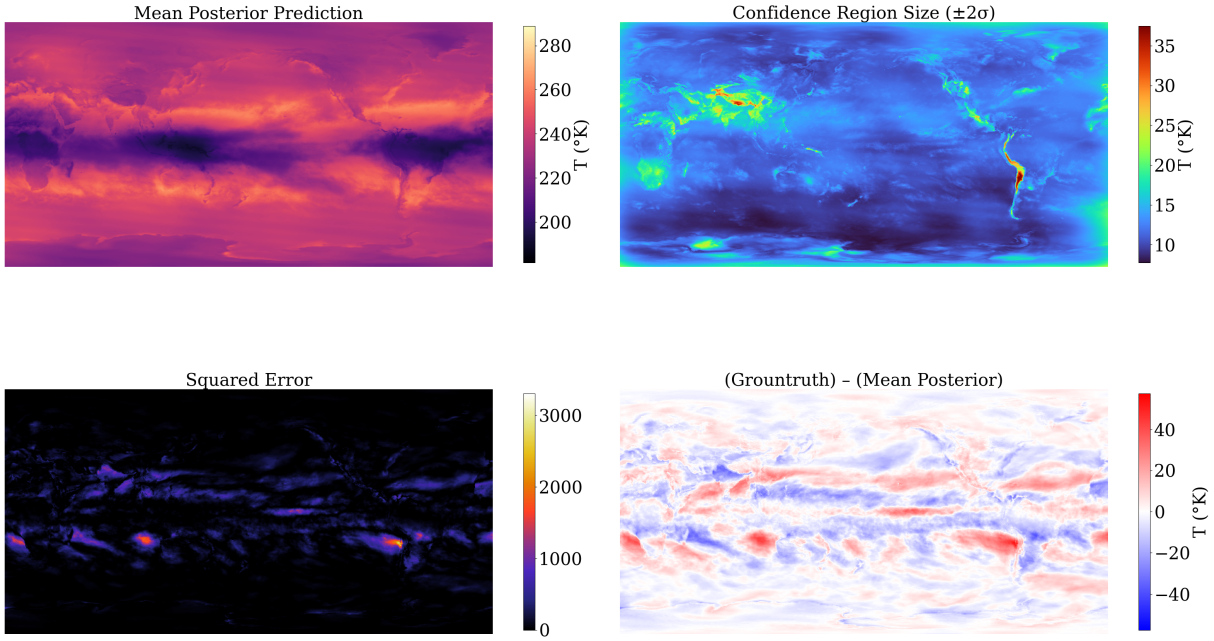


Figure A.6: Predicted downscaled atmospheric temperature field with VARGPR; **Top-Left:** Posterior mean; **Top-Right:** 95% confidence region size, i.e. 2 standard deviation of the posterior; **Bottom-Left:** Squared difference with unobserved groundtruth T^{HR}; **Bottom-Right:** Difference between unobserved groundtruth T^{HR} and the posterior mean.

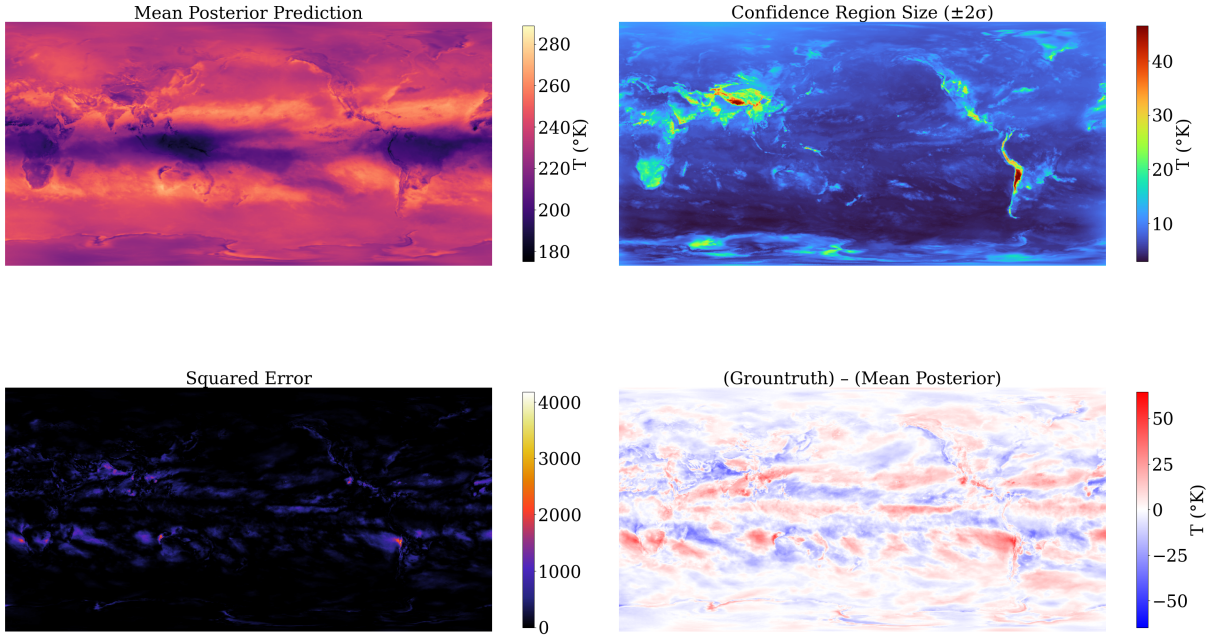


Figure A.7: Predicted downscaled atmospheric temperature field with VBAGG; **Top-Left:** Posterior mean; **Top-Right:** 95% confidence region size, i.e. 2 standard deviation of the posterior; **Bottom-Left:** Squared difference with unobserved groundtruth T^{HR}; **Bottom-Right:** Difference between unobserved groundtruth T^{HR} and the posterior mean.

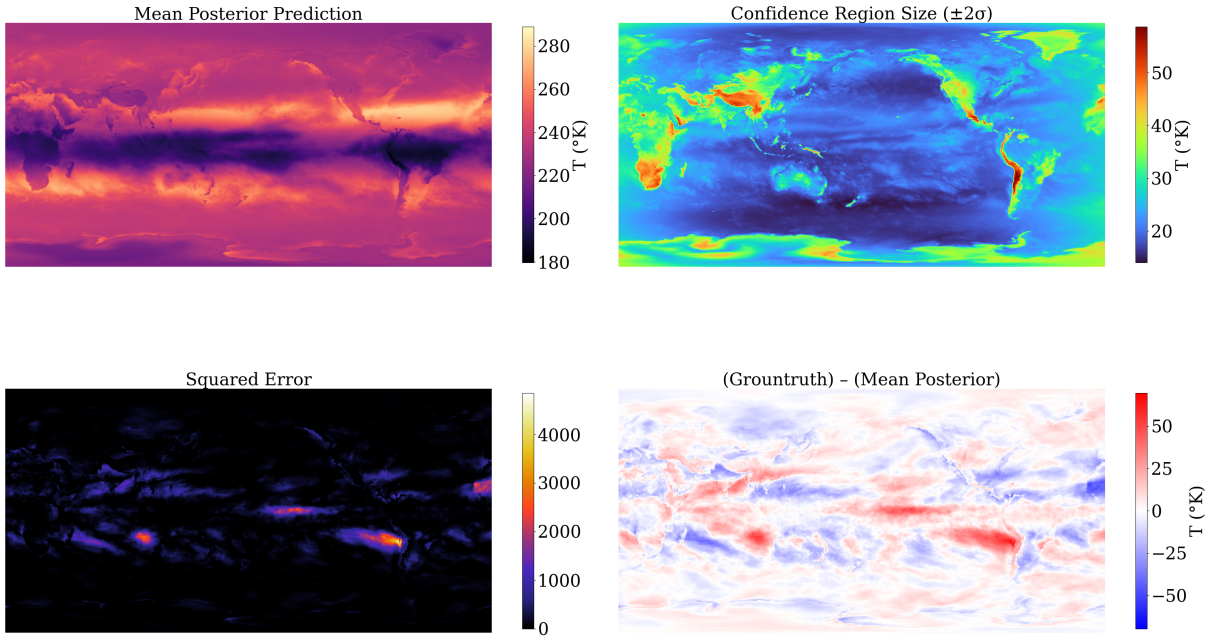


Figure A.8: Predicted downscaled atmospheric temperature field with VARCMP; **Top-Left:** Posterior mean; **Top-Right:** 95% confidence region size, i.e. 2 standard deviation of the posterior; **Bottom-Left:** Squared difference with unobserved groundtruth T^{HR}; **Bottom-Right:** Difference between unobserved groundtruth T^{HR} and the posterior mean.

B | Chapter 4 Appendix

B.1 Additional background on backdoor/front-door adjustments

In causal inference, we are often times interested in the interventional distributions i.e $p(y|do(x))$ rather than $p(y|x)$, as the former allows us to account for confounding effects. In order to obtain the interventional density $p(y|do(x))$, we resort to *do*-calculus [Pearl, 1995]. Here below we write out the definition for the 2 most crucial formulae; the front-door and backdoor adjustments, with which we are able to recover the interventional density using only the conditional ones.

B.1.1 Back-door Adjustment

The key intuition of back-door adjustments is to find/adjust a set of confounders that are unaffected by the treatment. We can then study the effect the treatment has on the target.

Definition B.1.1 (Back-Door). *A set of variables Z satisfies the backdoor criterion relative to an ordered pair of variables X_i, X_j in a DAG G if:*

1. *no node in Z is a descendant of X_i; and*
2. *Z blocks every path between X_i and X_j that contains an arrow into X_i*

Similarly, if X and Y are two disjoint subsets of nodes in G, then Z is said satisfy the back-door criterion relative to (X, Y) if it satisfies the criterion relative to any pair (X_i, X_j) such that $X_i \in X$ and $X_j \in Y$

Now with a given set Z that satisfies the back-door criterion, we apply the backdoor adjustment,

Theorem B.1.2 (Back-Door Adjustment). *If a set of variables Z satisfies the back-door criterion relative to (X, Y), then the causal effect of X on Y is identifiable and is given by the formula*

$$P(y|do(X) = x) = \int_z p(y|x, z)p(z)dz \tag{B.1}$$

B.2 Front-door Adjustment

Front-door adjustment deals with the case where confounders are unobserved and hence the backdoor adjustment is not applicable.

Definition B.2.1 (Front-door). *A set of variables Z is said to satisfy the front-door criterion relative to an ordered pair of variables (X, Y) if:*

1. *Z intercepts all directed paths from X to Y;*
2. *there is no back-door path from X to Z; and*
3. *all back-door paths from Z to Y are blocked by X*

Again, with an appropriate front-door adjustment set Z, we can identify the do density using the front-door adjustment formula.

Theorem B.2.2 (Front-Door Adjustment). *If Z satisfies the front-door criterion relative to (X, Y) and if $P(x, z) > 0$, then the causal effect of X on Y is identifiable and is given by the formula:*

$$p(y|do(X) = x) = \int_z p(z|x) \int_{x'} p(y|x', z)p(x')dx'dz \tag{B.2}$$

B.3 Derivations

B.3.1 CMP Derivation

Proposition 4.3.1. *Given dataset $D_1 = \{(x_i, y_i, z_i)\}_{i=1}^N$ and $D_2 = \{(\tilde{y}_j, t_j)\}_{j=1}^M$, if f is the posterior* GP *learnt from $\mathcal{D}_2$, then $g = \int f(y)p(y|do(X))dy$ is a* GP *$\mathcal{GP}(m_1, \kappa_1)$ defined on the treatment variable X with the following mean and covariance estimated using $\hat{\mu}_{Y|do(X)}$,*

$$m_1(x) = \langle \hat{\mu}_{Y|do(x)}, m_f \rangle_{\mathcal{H}_{k_Y}} = \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} K_{\mathbf{y}\tilde{\mathbf{y}}} (K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1} \mathbf{t} \tag{B.3}$$

$$\kappa_1(x, x') = \hat{\mu}^\top_{Y|do(x)} \hat{\mu}_{Y|do(x')} - \hat{\mu}^\top_{Y|do(x)} \Phi_{\tilde{\mathbf{y}}} (K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda I)^{-1} \Phi^\top_{\tilde{\mathbf{y}}} \hat{\mu}_{Y|do(x')} \tag{B.4}$$

$$= \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} \tilde{K}_{\mathbf{yy}} (K_{\Omega_x} + \lambda I)^{-1} \Phi_{\Omega_x}(x') \tag{B.5}$$

where $\hat{\mu}_{Y|do(x)} = \hat{\mu}_{Y|do(X)=x}$, $K_{\tilde{\mathbf{y}}\mathbf{y}} = \Phi^\top_{\tilde{\mathbf{y}}} \Phi_{\mathbf{y}}$, m_f and $\tilde{K}_{\mathbf{yy}}$ are the posterior mean function and covariance of f evaluated at $\mathbf{y}$ respectively. $\lambda > 0$ is the regularisation of the CME. *$\lambda_f > 0$ is the noise term for GP f. Ω_x is the set of variables as specified in Prop. 4.2.1.*

Proof for Proposition 4.3.1. The integral operator preserves Gaussianity under mild conditions (see conditions Chau et al. [2021a]), therefore

$$g(x) = \int f(y) dP(y|do(X) = x) \tag{B.6}$$

is also a Gaussian. For a standard GP prior $f \sim GP(0, k_y)$ and data $D_E = \{(\tilde{y}_j, t_j)\}_{j=1}^M$, standard conjugacy results for GPs lead to the posterior GP with mean $\bar{m}(y) = k_{y\tilde{\mathbf{y}}}(K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1}\mathbf{t}$ and covariance $\bar{k_y}(y, y') = k_y(y, y') - k_{y\tilde{\mathbf{y}}}(K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1}k_{\tilde{\mathbf{y}}y}$. Similar to Briol et al. [2019], repeated application of Fubini's theorem yields:

$$\mathbb{E}_f[g(x)] = \mathbb{E}_f\left[\int f(y)dP(y|do(X) = x\right] = \int \mathbb{E}_f[f(y)]dP(y|do(X) = x) \tag{B.7}$$

$$= \int \bar{m}(y)dP(y|do(X) = x) = \langle \bar{m}, \hat{\mu}_{Y|do(X)=x}\rangle \tag{B.8}$$

$$cov(g(x), g(x')) = \int\int cov(f(y), f(y'))dP(y|do(X) = x)dP(y'|do(X) = x) \tag{B.9}$$

$$= \int\int \bar{k}_y(y, y')dP(y|do(x))dP(y'|do(x')) \tag{B.10}$$

$$= \langle \mu_{Y|do(x)}, \mu_{Y|do(x')}\rangle - \hat{\mu}^\top_{Y|do(x)}\Phi_{\tilde{\mathbf{y}}}(K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda I)^{-1}\Phi^\top_{\tilde{\mathbf{y}}}\hat{\mu}_{Y|do(x')} \tag{B.11}$$

$$= \Phi_{\Omega_x}(x)^\top(K_{\Omega_x} + \lambda I)^{-1}\tilde{K}_{\mathbf{yy}}(K_{\Omega_x} + \lambda I)^{-1}\Phi_{\Omega_x}(x') \tag{B.12}$$

□

B.3.2 Choice of Nuclear Dominant Kernel

Recall in section 4.3.2, we introduced the nuclear dominant kernel r_y to ensure samples of $\mu_{gp} \sim GP(0, k_x \otimes r_y)$ are supported in $\mathcal{H}_{k_x} \otimes \mathcal{H}_{k_y}$ with probability 1. In the following, we will present the analytic form of the nuclear dominant kernel we used in this paper, which is the same as the formulation introduced in Appendix A.2 and A.3 of Flaxman et al. [2016]. Pick k_y as the RBF kernel, i.e.

$$k_y(y, y') = \exp\Big(-\frac{1}{2}(y - y')^\top\Sigma_\theta(y - y')\Big) \tag{B.13}$$

where Σ_θ is the covariance matrix for the kernel k_y. The nuclear dominant kernel construction from Flaxman et al. [2016] then yields the following expression:

$$r_y(y, y') = \int k_y(y, u)k_y(u, y')\nu(du) \tag{B.14}$$

where ν is some finite measure. If we pick $\nu(du) = \exp(\frac{||u||_2^2}{2\eta^2})du$, then we have

$$r_y(y, y') = (2\pi)^{D/2}|2\Sigma_\theta^{-1} + \eta^{-2}I|^{-1/2}\exp\Big(-\frac{1}{2}(y - y')^\top(2\Sigma_\theta)^{-1}(y - y')\Big) \tag{B.15}$$

$$\times \exp\Big(-\frac{1}{2}\Big(\frac{y + y'}{2}\Big)^\top\Big(\frac{1}{2}\Sigma_\theta + \eta^2 I\Big)^{-1}\Big(\frac{y + y'}{2}\Big)\Big) \tag{B.16}$$

B.3.3 BayesCME derivations

Proposition 4.3.2. *The posterior* GP *of* μ_{gp} *given observations* $\{\mathbf{x}, \mathbf{y}\}$ *has the following mean and covariance:*

$$m_\mu((x,y)) = k_{x\mathbf{x}}(K_{\mathbf{xx}} + \lambda I)^{-1} K_{\mathbf{yy}} R_{\mathbf{yy}}^{-1} r_{\mathbf{y}y} \tag{B.17}$$

$$\kappa_\mu((x,y),(x',y')) = k_{xx'} r_{y,y'} - k_{x\mathbf{x}}(K_{\mathbf{xx}} + \lambda I)^{-1} k_{\mathbf{x}x'} r_{y\mathbf{y}} R_{\mathbf{yy}}^{-1} r_{\mathbf{y}y'} \tag{B.18}$$

In addition, the following marginal likelihood can be used for hyperparameter optimisation,

$$-\frac{N}{2}\Big(\log|K_{\mathbf{xx}} + \lambda I| + \log|R|\Big) - \frac{1}{2}\operatorname{tr}\Big((K_{\mathbf{xx}} + \lambda I)^{-1} K_{\mathbf{yy}} R_{\mathbf{yy}}^{-1} K_{\mathbf{yy}}\Big) \tag{B.19}$$

Proof of Proposition 4.3.2. Recall the Bayesian formulation of CME corresponds to the following model,

$$\begin{aligned} \mu_{gp} &\sim GP\left(0, k_x \otimes r_y\right), \\ k_y\left(y_i, y'\right) &= \mu_{gp}\left(x_i, y'\right) + \lambda^{1/2}\epsilon_i\left(y'\right) \end{aligned}$$

with $\epsilon_i \sim GP(0, r_y)$ independently across i. Now consider $k_y(y_i, y_j)$ as noisy evaluations of $\mu_{gp}(x_i, y_j)$, we have the predictive posterior mean as

$$\begin{aligned} \operatorname{vec}\left(r_{\mathbf{y}y} k_{x\mathbf{x}}\right)^\top \left(K_{\mathbf{xx}} \otimes R_{\mathbf{yy}} + \lambda I \otimes R_{\mathbf{yy}}\right)^{-1} \operatorname{vec}(K_{\mathbf{yy}}) &= \operatorname{vec}\left(r_{\mathbf{y}y} k_{x\mathbf{x}}\right)^\top \Big(\left(K_{\mathbf{xx}} + \lambda I\right)^{-1} \otimes R_{\mathbf{yy}}^{-1}\Big) \operatorname{vec}(K_{\mathbf{yy}}) \\ &= \operatorname{vec}\left(r_{\mathbf{y}y} k_{x\mathbf{x}}\right)^\top \operatorname{vec}\Big(R_{\mathbf{yy}}^{-1} K_{\mathbf{yy}} \left(K_{\mathbf{xx}} + \lambda I\right)^{-1}\Big) \\ &= \operatorname{tr}\Big(r_{\mathbf{y}y} k_{x\mathbf{x}} \left(K_{\mathbf{xx}} + \lambda I\right)^{-1} K_{\mathbf{yy}} R_{\mathbf{yy}}^{-1}\Big) \\ &= k_{x\mathbf{x}} \left(K_{\mathbf{xx}} + \lambda I\right)^{-1} K_{\mathbf{yy}} R_{\mathbf{yy}}^{-1} r_{\mathbf{y}y}. \end{aligned}$$

And the covariance is,

$$
\begin{aligned}
\kappa\left((x,y),\left(x',y'\right)\right) &= k(x,x')r(y,y') - \text{vec}\left(r_{\mathbf{y}y}k_{x\mathbf{x}}\right)^\top \left(K_{\mathbf{xx}}\otimes R_{\mathbf{yy}} + \lambda I\otimes R_{\mathbf{yy}}\right)^{-1}\text{vec}\left(r_{\mathbf{y}y'}k_{x'\mathbf{x}}\right)\\
&= k(x,x')r(y,y') - \text{vec}\left(r_{\mathbf{y}y}k_{x\mathbf{x}}\right)^\top \left(\left(K_{\mathbf{xx}}+\lambda I\right)^{-1}\otimes R_{\mathbf{yy}}^{-1}\right)\text{vec}\left(r_{\mathbf{y}y'}k_{x'\mathbf{x}}\right)\\
&= k(x,x')r(y,y') - \text{vec}\left(r_{\mathbf{y}y}k_{x\mathbf{x}}\right)^\top \text{vec}\left(R_{\mathbf{yy}}^{-1}r_{\mathbf{y}y'}k_{x'\mathbf{x}}\left(K_{\mathbf{xx}}+\lambda I\right)^{-1}\right)\\
&= k(x,x')r(y,y') - \text{tr}\left(r_{\mathbf{y}y}k_{x\mathbf{x}}\left(K_{\mathbf{xx}}+\lambda I\right)^{-1}k_{\mathbf{x}x'}r_{y'\mathbf{y}}R_{\mathbf{yy}}^{-1}\right)\\
&= k(x,x')r(y,y') - k_{x\mathbf{x}}\left(K_{\mathbf{xx}}+\lambda I\right)^{-1}k_{\mathbf{x}x'}r_{y'\mathbf{y}}R_{\mathbf{yy}}^{-1}r_{\mathbf{y}y}.
\end{aligned}
$$

To compute the log likelihood, note that it contains the following two terms:

$$
\begin{aligned}
\text{vec}(K_{\mathbf{yy}})^\top\left(K_{\mathbf{xx}}\otimes R_{\mathbf{yy}}+\lambda I\otimes R_{\mathbf{yy}}\right)^{-1}\text{vec}(K_{\mathbf{yy}}) &= \text{vec}(K_{\mathbf{yy}})^\top\left(\left(K_{\mathbf{xx}}+\lambda I\right)^{-1}\otimes R_{\mathbf{yy}}^{-1}\right)\text{vec}(K_{\mathbf{yy}})\\
&= \text{vec}(K_{\mathbf{yy}})^\top\text{vec}\left(R_{\mathbf{yy}}^{-1}K_{\mathbf{yy}}\left(K_{\mathbf{xx}}+\lambda I\right)^{-1}\right)\\
&= \text{tr}\left(K_{\mathbf{yy}}\left(K_{\mathbf{xx}}+\lambda I\right)^{-1}K_{\mathbf{yy}}R_{\mathbf{yy}}^{-1}\right)
\end{aligned}
$$

and

$$
\begin{aligned}
-\frac{1}{2}\Big(\log\left|\left(K_{\mathbf{xx}}+\lambda I\right)\otimes R_{\mathbf{yy}}\right|\Big) &= -\frac{1}{2}\log\Big(\left|\left(K_{\mathbf{xx}}+\lambda I\right)\right|^N|R|^N\Big)\\
&= -\frac{N}{2}\Big(\log\left|K_{\mathbf{xx}}+\lambda I\right|+\log|R|\Big)
\end{aligned}
$$

where we used the fact that determinant of Kronecker product of two $N\times N$ matrices A, B is: $|A\otimes B| = |A|^N|B|^N$.

Therefore the log likelihood can be expressed as

$$
-\frac{N}{2}\Big(\log\left|K_{\mathbf{xx}}+\lambda I\right|+\log|R|\Big) - \frac{1}{2}\,\text{tr}\left(\left(K_{\mathbf{xx}}+\lambda I\right)^{-1}K_{\mathbf{yy}}R_{\mathbf{yy}}^{-1}K_{\mathbf{yy}}\right) \tag{B.20}
$$

□

B.3.4 Causal BayesCME derivations

The following proposition extend BAYESCME to the causal setting.

Proposition C.1 (Causal BayesCME). *Denote μ^{do}_{gp} as the GP modelling $\mu_{Y|do(X)}$. Then using the Ω notations introduced in proposition 4.2.1, the posterior* GP *of μ^{do}_{gp} given observations $\{\mathbf{x}, \mathbf{z}, \mathbf{y}\}$ has the following mean and covariance:*

$$m^{do}_{\mu}((x,y)) = \Phi_{\Omega_x}(x)^\top \Big(K_{\Omega_x} + \lambda I\Big)^{-1} K_{\mathbf{yy}} R^{-1}_{\mathbf{yy}} r_{\mathbf{y}y} \tag{B.21}$$

$$\kappa^{do}_{\mu}((x,y),(x',y')) = \Phi_{\Omega_x}(x)^\top \Phi_{\Omega_x}(x') r_{y,y'} - \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} \Phi_{\Omega_x}(x') r_{y\mathbf{y}} R^{-1}_{\mathbf{yy}} r_{\mathbf{y}y'} \tag{B.22}$$

In addition, the following marginal likelihood can be used for hyperparameter optimisation,

$$-\frac{N}{2}\Big(\log|K_{\Omega_x} + \lambda I| + \log|R|\Big) - \frac{1}{2}\operatorname{tr}\Big((K_{\Omega_x} + \lambda I)^{-1} K_{\mathbf{yy}} R^{-1}_{\mathbf{yy}} K_{\mathbf{yy}}\Big) \tag{B.23}$$

Proof of Proposition C.1. In the following we will assume Z is the backdoor adjustment variable. Front-door and general cases follow analogously. Denote $\mu_{gp}((x,z),y)$ as the BAYESCME model for $\mu_{Y|X=x,Z=z}(y)$. As we have

$$\mu_{Y|do(X)=x} = \int\int \phi_y(y) p(y|x,z) p(z) dz dy \tag{B.24}$$

$$= \int \mu_{Y|X=x,Z=z} p(z) dz \tag{B.25}$$

$$= \mathbb{E}_Z[\mu_{Y|X=x,Z}] \tag{B.26}$$

It is thus natural to define μ^{do}_{gp} as the induced GP when we replace $\mu_{Y|X=x,Z=z}$ with $\mu_{gp}((x,z),\cdot)$,

$$\mu^{do}_{gp}(x,\cdot) = \mathbb{E}_Z[\mu_{gp}((x,Z),\cdot)] \tag{B.27}$$

Now we can compute the mean of μ^{do}_{gp},

$$m^{do}_{\mu}(x,y) = \mathbb{E}_{\mu_{gp}}\mathbb{E}_Z[\mu_{gp}(x,Z,y)] \tag{B.28}$$

$$= \mathbb{E}_Z\Big((k_{x\mathbf{x}} \odot k_z(Z,\mathbf{z}))(K_{\mathbf{xx}} \odot K_{\mathbf{zz}} + \lambda I)^{-1} K_{\mathbf{yy}} R^{-1}_{\mathbf{yy}} r_{\mathbf{y}y}\Big) \tag{B.29}$$

$$= \Big((k_{x\mathbf{x}} \odot \mu_z^\top \Phi_{\mathbf{z}})(K_{\mathbf{xx}} \odot K_{\mathbf{zz}} + \lambda I)^{-1} K_{\mathbf{yy}} R^{-1}_{\mathbf{yy}} r_{\mathbf{y}y}\Big) \tag{B.30}$$

$$= \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} K_{\mathbf{yy}} R^{-1}_{\mathbf{yy}} r_{\mathbf{y}y} \tag{B.31}$$

Similarly for covariance, we have,

$$\kappa^{do}_{\mu}((x,y),(x',y')) = \mathbb{E}_{Z,Z'}[cov\big(\mu_{gp}((x,Z),y), \mu_{gp}((x',Z'),y')\big)] \tag{B.32}$$

and the rest is just algebra,

$$= \Phi_{\Omega_x}(x)^\top \Phi_{\Omega_x}(x') r_{y,y'} - \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} \Phi_{\Omega_x}(x') r_{y\mathbf{y}} R_{\mathbf{yy}}^{-1} r_{\mathbf{y}y'} \quad \text{(B.33)}$$

□

B.3.5 BayesIME derivation

Now we have derived the Causal BAYESCME, it is time to compute $\langle f, \mu_{gp}^{do}(x, \cdot)\rangle$ where $f \in \mathcal{H}_{k_y}$. This requires us to be able to compute $\langle f, r_y(\cdot, y)\rangle$ which corresponds to the following:

$$\langle f, r_y(\cdot, y)\rangle_{\mathcal{H}_{k_y}} = \left\langle f, \int k_y(\cdot, u) k_y(u, y) \nu(du) \right\rangle \quad \text{(B.34)}$$

$$= \int f(u) k_y(u, y) \nu(du) \quad \text{(B.35)}$$

when f is a KRR learnt from $\mathcal{D}_2$, i.e $f(y) = k_{y\tilde{\mathbf{y}}}(K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1}\mathbf{t}$, we have

$$= \mathbf{t}^\top (K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1} \int k_{\tilde{\mathbf{y}}u} k_y(u, y) \nu(du) \quad \text{(B.36)}$$

$$= \mathbf{t}^\top (K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1} r_{\tilde{\mathbf{y}}y} \quad \text{(B.37)}$$

Now we are ready to derive BAYESIME.

Proposition 4.3.3. *Given dataset $D_1 = \{(x_i, y_i, z_i)\}_{i=1}^N$ and $D_2 = \{(\tilde{y}_j, t_j)\}_{j=1}^M$, if f is a* KRR *learnt from $\mathcal{D}_2$ and $\mu_{Y|do(X)}$ modelled as a* V-GP *using $\mathcal{D}_1$, then $g = \langle f, \mu_{Y|do(X)}\rangle \sim \mathcal{GP}(m_2, \kappa_2)$ where,*

$$m_2(x) = \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} K_{\mathbf{yy}} R_{\mathbf{yy}}^{-1} R_{\mathbf{y}\tilde{\mathbf{y}}} A \quad \text{(B.38)}$$

$$\kappa_2(x, x') = B\Phi_{\Omega_x}(x)^\top \Phi_{\Omega_x}(x) - C\Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} \Phi_{\Omega_x}(x') \quad \text{(B.39)}$$

where $A = (K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1}\mathbf{t}$, $B = A^\top R_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} A$ and $C = A^\top R_{\tilde{\mathbf{y}}\mathbf{y}} R_{\mathbf{yy}}^{-1} R_{\mathbf{y}\tilde{\mathbf{y}}} A$

Proof of Proposition 4.3.3. Using the μ_{gp}^{do} notation from Proposition $C.1$, we can write the inner product

as $\langle \mu^{do}_{gp}(x,\cdot), f\rangle$, where the mean is,

$$m_2(x) = \mathbb{E}[\mu^{do}_{gp}(x,\cdot)]^\top f \tag{B.40}$$

$$= \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} K_{\mathbf{yy}} R^{-1}_{\mathbf{yy}} R(\mathbf{y},\cdot)^\top f \tag{B.41}$$

$$= \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} K_{\mathbf{yy}} R^{-1}_{\mathbf{yy}} R_{\mathbf{y}\tilde{\mathbf{y}}} (K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1} \mathbf{t} \tag{B.42}$$

where we used the fact f is a KRR learnt from $\mathcal{D}_2$. The covariance can then be computed by realising $cov(f^\top \mu^{do}_{gp}(x,\cdot), f^\top \mu^{do}_{gp}(x',\cdot)) = f^\top cov(\mu^{do}_{gp}(x,\cdot), \mu^{do}_{gp}(x',\cdot)) f$. □

B.3.6 BayesIMP Derivations

BAYESIMP can be understood as a model characterising the RKHS inner product of Gaussian Processes. In the following, we will first introduce some general theory of inner product of GPs, and introduce a finite dimensional scheme later on. Finally, we will show how BAYESIMP can be derived right away from this general framework.

Before that, we will showcase the following identity for computing variance of inner products of independent multivariate Gaussians,

Proposition C.2. *Let $\mu_X := \mathbb{E}[X]$ and $\Sigma_X := Var(X)$ be the mean and variance of a multivariate Gaussian rv, similarly μ_Y, Σ_Y for Gaussian rv Y. If X and Y are independent, then the variance of their inner product is given by the following expression,*

$$Var(X^\top Y) = \mu_X^\top \Sigma_Y \mu_X + \mu_Y^\top \Sigma_X \mu_Y + tr\Big(\Sigma_Y \Sigma_X\Big) \tag{B.43}$$

Moreover, the covariance between $X^\top Y_1$, $X^\top Y_2$ follows a similar form,

$$cov(X^\top Y_1, X^\top Y_2) = \mu_X^\top \Sigma_{Y_1 Y_2} \mu_X + \mu_{Y_1}^\top \Sigma_X \mu_{Y_2} + \mathrm{tr}(\Sigma_X \Sigma_{Y_1 Y_2}) \tag{B.44}$$

Proof.

$$\begin{aligned}
\mathrm{Var}\left[X^\top Y\right] &= \mathbb{E}\left[\left(X^\top Y\right)^2\right] - \mathbb{E}\left[X^\top Y\right]^2 \\
&= \mathbb{E}\left[X^\top Y Y^\top X\right] - \left(\mathbb{E}[X]^\top \mathbb{E}[Y]\right)^2 \\
&= \mathbb{E}\left[\mathrm{tr}\left(XX^\top YY^\top\right)\right] - \left(\mu_X^\top \mu_Y\right)^2 \\
&= \mathrm{tr}\left(\mathbb{E}\left[XX^\top\right]\mathbb{E}\left[YY^\top\right]\right) - \left(\mu_X^\top \mu_Y\right)^2 \\
&= \mathrm{tr}\left(\left(\mu_X\mu_X^\top + \Sigma_X\right)\left(\mu_Y\mu_Y^\top + \Sigma_Y\right)\right) - \left(\mu_X^\top \mu_Y\right)^2 \\
&= \mathrm{tr}\left(\mu_X\mu_X^\top\mu_Y\mu_Y^\top\right) + \mathrm{tr}\left(\mu_X\mu_X^\top\Sigma_Y\right) + \mathrm{tr}\left(\Sigma_X\mu_Y\mu_Y^\top\right) + \mathrm{tr}\left(\Sigma_X\Sigma_Y\right) - \left(\mu_X^\top \mu_Y\right)^2 \\
&= \left(\mu_X^\top\mu_Y\right)^2 + \mathrm{tr}\left(\mu_X^\top\Sigma_Y\mu_X\right) + \mathrm{tr}\left(\mu_Y^\top\Sigma_X\mu_Y\right) + \mathrm{tr}\left(\Sigma_X\Sigma_Y\right) - \left(\mu_X^\top \mu_Y\right)^2 \\
&= \mu_X^\top\Sigma_Y\mu_X + \mu_Y^\top\Sigma_X\mu_Y + \mathrm{tr}\left(\Sigma_X\Sigma_Y\right)
\end{aligned} \tag{B.45}$$

Generalising to the case for covariance is straight forward. □

RKHS inner product of Gaussian Processes

Let $f_1 \sim GP(m_1, \kappa_1)$ and $f_2 \sim GP(m_2, \kappa_2)$. We assume that f and g are both supported within the RKHS $\mathcal{H}_k$. Can we characterise the distribution of $\langle f_1, f_2\rangle_{\mathcal{H}_k}$?

This situation would arise if f_1 and f_2 arise as GP posteriors in a regression model corresponding to the priors $f_1 \sim GP(0, r_1)$, $f_2 \sim GP(0, r_2)$ where r_1, r_2 satisfy the nuclear dominance property. In particular, we could choose

$$\begin{aligned}
r_1(u,v) &= \int k(u,z)\,k(z,v)\,\nu_1(dz), \\
r_2(u,v) &= \int k(u,z)\,k(z,v)\,\nu_2(dz).
\end{aligned}$$

Posterior means in that case can be expanded as

$$m_1 = \sum \alpha_i r_1(\cdot, x_i), \qquad m_2 = \sum \beta_j r_2(\cdot, y_j).$$

We assume that f_1 and f_2 are independent, i.e. they correspond to posteriors computed on independent

data. Then

$$\begin{aligned}\mathbb{E}\left\langle f_1, f_2\right\rangle_{\mathcal{H}_k} &= \left\langle m_1, m_2\right\rangle_{\mathcal{H}_k} \\ &= \left\langle \sum \alpha_i r_1\left(\cdot, x_i\right), \sum \beta_j r_2\left(\cdot, y_j\right)\right\rangle_{\mathcal{H}_k} \\ &= \alpha^{\top} Q \beta,\end{aligned}$$

where

$$\begin{aligned}Q_{ij} = q\left(x_i, y_j\right) &:= \left\langle r_1\left(\cdot, x_i\right), r_2\left(\cdot, y_j\right)\right\rangle_{\mathcal{H}_k} \\ &= \left\langle \int k\left(\cdot, z\right) k\left(z, x_i\right) \nu_1\left(dz\right), \int k\left(\cdot, z'\right) k\left(z', y_j\right) \nu_2\left(dz'\right)\right\rangle_{\mathcal{H}_k} \\ &= \int\int \left\langle k\left(\cdot, z\right), k\left(\cdot, z'\right)\right\rangle_{\mathcal{H}_k} k\left(z, x_i\right) k\left(z', y_j\right) \nu_1\left(dz\right) \nu_2\left(dz'\right) \\ &= \int\int k\left(z, z'\right) k\left(z, x_i\right) k\left(z', y_j\right) \nu_1\left(dz\right) \nu_2\left(dz'\right).\end{aligned}$$

The variance would be given, in analogy to the finite dimensional case, by

$$\text{var}\left\langle f_1, f_2\right\rangle_{\mathcal{H}_k} = \left\langle m_1, \Sigma_2 m_1\right\rangle_{\mathcal{H}_k} + \left\langle m_2, \Sigma_1 m_2\right\rangle_{\mathcal{H}_k} + \text{tr}\left(\Sigma_1 \Sigma_2\right),$$

with $\Sigma_1 f = \int \kappa_1\left(\cdot, u\right) f(u) du$ and similarly for Σ_2. Thus

$$\begin{aligned}\left\langle m_1, \Sigma_2 m_1\right\rangle_{\mathcal{H}_k} &= \left\langle \sum \alpha_i r_1\left(\cdot, x_i\right), \sum \alpha_j \int \kappa_2\left(\cdot, u\right) r_1\left(u, x_j\right) du\right\rangle_{\mathcal{H}_k} \\ &= \sum\sum \alpha_i \alpha_j \int \left\langle r_1\left(\cdot, x_i\right), \kappa_2\left(\cdot, u\right)\right\rangle_{\mathcal{H}_k} r_1\left(u, x_j\right) du.\end{aligned}$$

Now, given that kernel κ_2 depends on r_2 in a simple way, it should be possible to write down the full expression similarly as for Q_{ij} above. In particular

$$\kappa_2\left(\cdot, u\right) = r_2\left(\cdot, u\right) - r_2\left(\cdot, \mathbf{y}\right)\left(R_{2,\mathbf{yy}} + \sigma_2^2 I\right)^{-1} r_2\left(\mathbf{y}, u\right).$$

Hence

$$\left\langle r_1\left(\cdot, x_i\right), \kappa_2\left(\cdot, u\right)\right\rangle_{\mathcal{H}_k} = q\left(x_i, u\right) - q\left(x_i, \mathbf{y}\right)\left(R_{2,\mathbf{yy}} + \sigma_2^2 I\right)^{-1} r_2\left(\mathbf{y}, u\right).$$

However, this further requires approximating integrals of the type

$$\int q(x_i, u)\, r_1(u, x_j)\, du = \iiiint k(z, z')\, k(z, x_i)\, k(z', u)\, k(u, z'')\, k(z'', x_j)\, \nu_1(dz)\, \nu_2(dz')\, \nu_1(dz'')\, du,$$

etc. Thus, while possible in principle, this approach to compute the variance is cumbersome.

A finite dimensional approximation

To approximate the variance, hence, it is simpler to consider finite-dimensional approximations to f_1 and f_2. Namely, collate $\{x_i\}$ and $\{y_j\}$ into a single set of points ξ (note that we could here take an arbitrary set of points), and consider finite-dimensional GPs given by

$$\tilde{f}_1 = \sum a_j k(\cdot, \xi_j), \quad \tilde{f}_2 = \sum b_j k(\cdot, \xi_j),$$

where we selects distribution of a and b such that evaluations of $\tilde{f}_1$ and $\tilde{f}_2$ on ξ, $K_{\xi\xi}a$ and $K_{\xi\xi}b$ respectively, have the same distributions as evaluations of f_1 and f_2 on ξ. In particular, we take

$$a \sim \mathcal{N}\left(K_{\xi\xi}^{-1} m_1(\xi), K_{\xi\xi}^{-1} \mathcal{K}_{1,\xi\xi} K_{\xi\xi}^{-1}\right), \quad b \sim \mathcal{N}\left(K_{\xi\xi}^{-1} m_2(\xi), K_{\xi\xi}^{-1} \mathcal{K}_{2,\xi\xi} K_{\xi\xi}^{-1}\right),$$

where we denoted by $m_1(\xi)$ a vector such that $[m_1(\xi)]_i = m_1(\xi_i)$ and by $\mathcal{K}_{1,\xi\xi}$ a matrix such that $[\mathcal{K}_{1,\xi\xi}]_{ij} = \kappa_1(\xi_i, \xi_j)$.

Then, clearly

$$\begin{aligned} \left\langle \tilde{f}_1, \tilde{f}_2 \right\rangle_{\mathcal{H}_k} &= a^\top K_{\xi\xi} b \\ &= \left(K_{\xi\xi}^{1/2} a\right)^\top \left(K_{\xi\xi}^{1/2} b\right), \end{aligned}$$

and now we are left with the problem of computing the mean and the variance of inner product between two independent Gaussian vectors, as given in Proposition $C.2$. We have

$$\begin{aligned} \mathbb{E}\left\langle \tilde{f}_1, \tilde{f}_2 \right\rangle_{\mathcal{H}_k} &= \left(K_{\xi\xi}^{1/2} K_{\xi\xi}^{-1} m_1(\xi)\right)^\top \left(K_{\xi\xi}^{1/2} K_{\xi\xi}^{-1} m_2(\xi)\right) \\ &= m_1(\xi)^\top K_{\xi\xi}^{-1} K_{\xi\xi} K_{\xi\xi}^{-1} m_2(\xi) \\ &= m_1(\xi)^\top K_{\xi\xi}^{-1} m_2(\xi), \end{aligned}$$

and

$$\begin{aligned}
\text{var}\left\langle \tilde{f}_1, \tilde{f}_2 \right\rangle_{\mathcal{H}_k} &= \left(K_{\xi\xi}^{1/2} K_{\xi\xi}^{-1} m_1(\xi)\right)^\top K_{\xi\xi}^{-1/2} \mathcal{K}_{2,\xi\xi} K_{\xi\xi}^{-1/2} \left(K_{\xi\xi}^{1/2} K_{\xi\xi}^{-1} m_1(\xi)\right) \\
&+ \left(K_{\xi\xi}^{1/2} K_{\xi\xi}^{-1} m_2(\xi)\right)^\top K_{\xi\xi}^{-1/2} \mathcal{K}_{1,\xi\xi} K_{\xi\xi}^{-1/2} \left(K_{\xi\xi}^{1/2} K_{\xi\xi}^{-1} m_2(\xi)\right) \\
&+ \text{tr}\left(K_{\xi\xi}^{-1/2} \mathcal{K}_{1,\xi\xi} K_{\xi\xi}^{-1/2} K_{\xi\xi}^{-1/2} \mathcal{K}_{2,\xi\xi} K_{\xi\xi}^{-1/2}\right) \\
&= m_1(\xi)^\top K_{\xi\xi}^{-1} \mathcal{K}_{2,\xi\xi} K_{\xi\xi}^{-1} m_1(\xi) \\
&+ m_2(\xi)^\top K_{\xi\xi}^{-1} \mathcal{K}_{1,\xi\xi} K_{\xi\xi}^{-1} m_2(\xi) \\
&+ \text{tr}\left(\mathcal{K}_{1,\xi\xi} K_{\xi\xi}^{-1} \mathcal{K}_{2,\xi\xi} K_{\xi\xi}^{-1}\right).
\end{aligned}$$

Coming back to BayesIMP

Now coming back to the derivation of BayesIMP. We will first provide two finite approximation of f and $\mu_{gp}^{do}(x, \cdot)$ in the following two propositions. Recall these finite approximations are set up such that they match the distributions of evaluations of f and μ_{gp}^{do} at $\hat{\mathbf{y}} = [\mathbf{y}^\top \ \tilde{\mathbf{y}}^\top]^\top$. The latter thus act as landmark points for the finite dimensional approximations.

Proposition C.3 (Finite dimensional approximation of f)**.** *Let $\hat{\mathbf{y}} = [\mathbf{y}^\top \ \tilde{\mathbf{y}}^\top]^\top$ be the concatenation of $\mathbf{y}$ and $\tilde{\mathbf{y}}$. We can approximate f with ,*

$$\tilde{f}|\mathbf{t} \sim N(m_{\tilde{f}}, \Sigma_{\tilde{f}}) \tag{B.46}$$

where,

$$m_{\tilde{f}} = \Phi_{\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} R_{\hat{\mathbf{y}}\tilde{\mathbf{y}}} (R_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1} \mathbf{t} \tag{B.47}$$

$$\Sigma_{\tilde{f}} = \Phi_{\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} \bar{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} \Phi_{\hat{\mathbf{y}}}^\top \tag{B.48}$$

and $\bar{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}} = R_{\hat{\mathbf{y}}\hat{\mathbf{y}}} - R_{\hat{\mathbf{y}}\tilde{\mathbf{y}}} (R_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1} R_{\tilde{\mathbf{y}}\hat{\mathbf{y}}}$.

Similarly for $\mu_{gp}^{do}(x, \cdot)$, we have the following,

Proposition C.4 (Finite dimensional approximation of $\mu_{gp}^{do}(x,\cdot)$)**.** *Let* $\hat{\mathbf{y}} = [\mathbf{y}^\top \ \tilde{\mathbf{y}}^\top]^\top$ *be the concatenation of* $\mathbf{y}$ *and* $\tilde{\mathbf{y}}$*. We can approximate* $\mu_{gp}^{do}(x,\cdot)$ *with ,*

$$\tilde{\mu}_{gp}^{do}(x,\cdot)|\,\mathrm{vec}(K_{\mathbf{yy}}) \sim N(m_{\tilde{\mu}}, \Sigma_{\tilde{\mu}}) \tag{B.49}$$

where,

$$m_{\tilde{\mu}} = \Phi_{\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} R_{\hat{\mathbf{y}}\mathbf{y}} R_{\mathbf{yy}}^{-1} K_{\mathbf{yy}} (K_{\Omega_x} + \lambda I)^{-1} \Phi_{\Omega_x}(x) \tag{B.50}$$

$$\Sigma_{\tilde{\mu}} = \Phi_{\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{\mu} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} \Phi_{\hat{\mathbf{y}}}^\top \tag{B.51}$$

where $K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{\mu} = \Phi_{\Omega_x}(x)^\top \Phi_{\Omega_x}(x) R_{\hat{\mathbf{y}}\hat{\mathbf{y}}} - \big(\Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} \Phi_{\Omega_x}(x)\big) R_{\hat{\mathbf{y}}\mathbf{y}} R_{\mathbf{yy}}^{-1} R_{\mathbf{y}\hat{\mathbf{y}}}$

Now we have everything we need to derive the main algorithm in our paper, the BAYESIMP. Note that we did not introduce the μ_{gp}^{do} notation in the main text to avoid confusion as we did not have space to properly define μ_{gp}^{do}.

Proposition 4.3.4 (BAYESIMP). *Let* f *and* $\mu_{Y|do(X)}$ *be* GP*s learnt as above. Denote* $\tilde{f}$ *and* $\tilde{\mu}_{Y|do(X)}$ *as the finite dimensional approximation of* f *and* $\mu_{Y|do(X)}$ *respectively. Then* $\tilde{g} = \langle \tilde{f}, \tilde{\mu}_{Y|do(X)} \rangle$ *has the following mean and covariance:*

$$m_3(x) = E_x K_{\mathbf{y}\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} R_{\hat{\mathbf{y}}\tilde{\mathbf{y}}} (R_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f I)^{-1} \mathbf{t} \tag{B.52}$$

$$\kappa_3(x,x') = \underbrace{E_x \Theta_1^\top \tilde{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}} \Theta_1 E_{x'}^\top}_{\text{Uncertainty from } \mathcal{D}_1} + \underbrace{\Theta_2^{(a)} F_{xx'} - \Theta_2^{(b)} G_{xx'}}_{\text{Uncertainty from } \mathcal{D}_2} + \underbrace{\Theta_3^{(a)} F_{xx'} - \Theta_3^{(b)} G_{xx'}}_{\text{Uncertainty from Interaction}} \tag{B.53}$$

where $E_x = \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1}, F_{xx'} = \Phi_{\Omega_x}(x)^\top \Phi_{\Omega_x}(x'), G_{xx'} = \Phi_{\Omega_x}(x)^\top (K_{\Omega_x} + \lambda I)^{-1} \Phi_{\Omega_x}(x')$*, and* $\Theta_1 = K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} R_{\hat{\mathbf{y}}\mathbf{y}} R_{\mathbf{yy}}^{-1} K_{\mathbf{yy}}$*,* $\Theta_2^{(a)} = \Theta_4^\top R_{\hat{\mathbf{y}}\hat{\mathbf{y}}} \Theta_4, \Theta_2^{(b)} = \Theta_4^\top R_{\hat{\mathbf{y}}\mathbf{y}} R_{\mathbf{yy}}^{-1} R_{\mathbf{y}\hat{\mathbf{y}}} \Theta_4$ *and* $\Theta_3^{(a)} = tr(K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} R_{\hat{\mathbf{y}}\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} \bar{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}), \Theta_3^{(b)} = tr(R_{\hat{\mathbf{y}}\mathbf{y}} R_{\mathbf{yy}}^{-1} R_{\mathbf{y}\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} \bar{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1})$ *and* $\Theta_4 = K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} R_{\hat{\mathbf{y}}\tilde{\mathbf{y}}} (K_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} + \lambda_f)^{-1} \mathbf{t}$*.* $\bar{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}$ *is the posterior covariance of* f *evaluated at* $\hat{\mathbf{y}}$

Proof of Proposition 4.3.4. Since $\tilde{g} = \langle \tilde{f}, \tilde{\mu}_{gp}^{do} \rangle$ is an inner product between two finite dimensional GPs, we know the variance (as given by Proposition $C.2$) is characterised by,

$$var(g) = m_{\tilde{\mu}}^\top \Sigma_{\tilde{f}} m_{\tilde{\mu}} + m_{\tilde{f}}^\top \Sigma_{\tilde{\mu}} m_{\tilde{f}} + \mathrm{tr}(\Sigma_{\tilde{f}} \Sigma_{\tilde{\mu}}) \tag{B.54}$$

Expanding out each term we get Proposition 4.3.4:

$$m^\top \Sigma m = E_x \Theta_1^\top \tilde{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}} \Theta_1 E_{x'}^\top \tag{B.55}$$

$$m_{\tilde{f}}^\top \Sigma_{\tilde{\mu}} m_{\tilde{f}} = \Theta_2^{(a)} F_{xx'} - \Theta_2^{(b)} G_{xx'} \tag{B.56}$$

$$\tag{B.57}$$

while the first two terms resembles the uncertainty obtained from IMP and BAYESIME, the trace term is new and we will expand it out here,

$$\begin{aligned}
\mathrm{tr}(\Sigma\Sigma) &= \mathrm{tr}\left(\Phi_{\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{\mu} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} \Phi_{\hat{\mathbf{y}}}^\top \Phi_{\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} \bar{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} \Phi_{\hat{\mathbf{y}}}^\top\right) && \text{(B.58)} \\
&= \mathrm{tr}\left(K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{\mu} K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} \bar{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}\right) && \text{(B.59)} \\
&= \mathrm{tr}\left(K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} \left(F_{xx'} R_{\hat{\mathbf{y}}\hat{\mathbf{y}}} - G_{xx'} R_{\hat{\mathbf{y}}\mathbf{y}} R_{\mathbf{y}\mathbf{y}}^{-1} R_{\mathbf{y}\hat{\mathbf{y}}}\right) K_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1} \bar{R}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}\right) && \text{(B.60)} \\
&= \Theta_3^{(a)} F_{xx'} - \Theta_3^{(b)} G_{xx'} && \text{(B.61)}
\end{aligned}$$

□

B.4 Details on Experimental setup

B.4.1 Details on Ablation Study

B.4.1.1 Data Generating Process

We use the following causal graphs, $X \rightarrow Y$ and $Y \rightarrow T$, to demonstrate a simple scenario for our data fusion setting. As linking functions, we used for $\mathcal{D}_1$, $Y = xcos(\pi x) + \epsilon_1$ and for $\mathcal{D}_2$, $T = 0.5 * y * cos(y) + \epsilon_2$. where $\epsilon_i \sim \mathcal{N}(0, \sigma_i)$. Here below we plotted the data for illustration purposes.

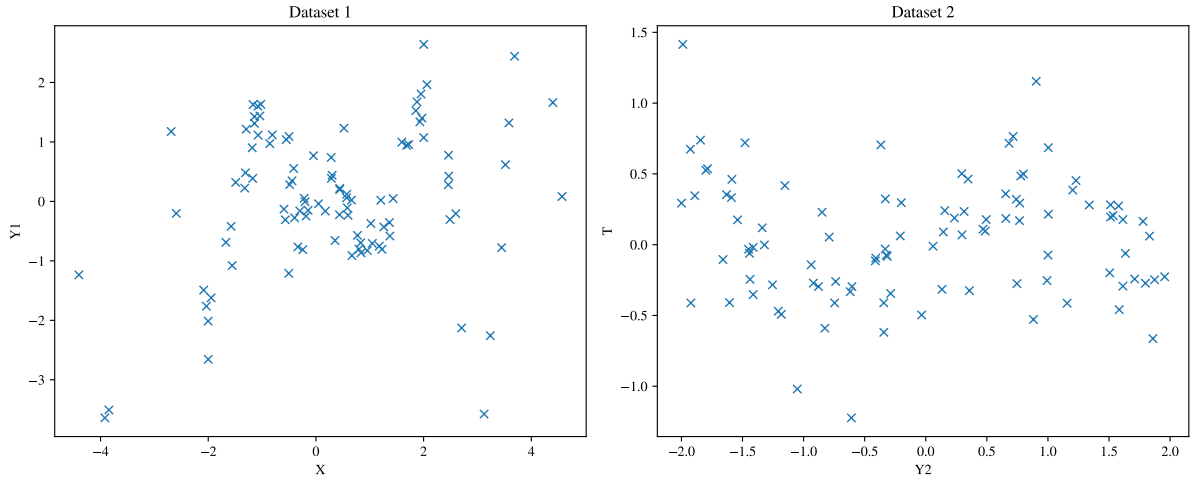


Figure B.1: (Left) Illustration of $\mathcal{D}_1$ (Right) Illustration of $\mathcal{D}_2$

B.4.1.2 Explanation on the extrapolation effect

In the main text, we referred to the case where IMP is better than BAYESIME as **extrapolation effect**. We note from the figure above that in $\mathcal{D}_1$ we have x around -4 being mapped onto y values around -3. Note however, that in $\mathcal{D}_2$, we do not observe any values $\tilde{Y}$ below -2. Hence, because IMP uses a GP model for $\mathcal{D}_2$ we are able to account for this mismatch in support and hence attribute more uncertainty to this region, i.e. we see the spike in uncertainty in Fig. 4.5 for IMP.

B.4.1.3 Calibration Plots

To investigate the accuracy of the uncertainty quantification in the proposed methods, we perform a (frequentist) calibration analysis of the credible intervals stemming from each method. Fig. B.2 gives the calibration plots of the sampling methods (sampling-based method of Aglietti et al. [2020b]) as well as the three proposed methods. On the x-axis is the portion of the posterior mass, corresponding to the width of the credible interval. We will interpret that as a nominal coverage probability of the true function values.

On the y-axis is the true coverage probability estimated using the percentage of the times true function values do lie within the corresponding credible intervals. A perfectly calibrated method should have a nominal coverage probability equal to the true coverage probability, i.e. being closer to the diagonal line is better.

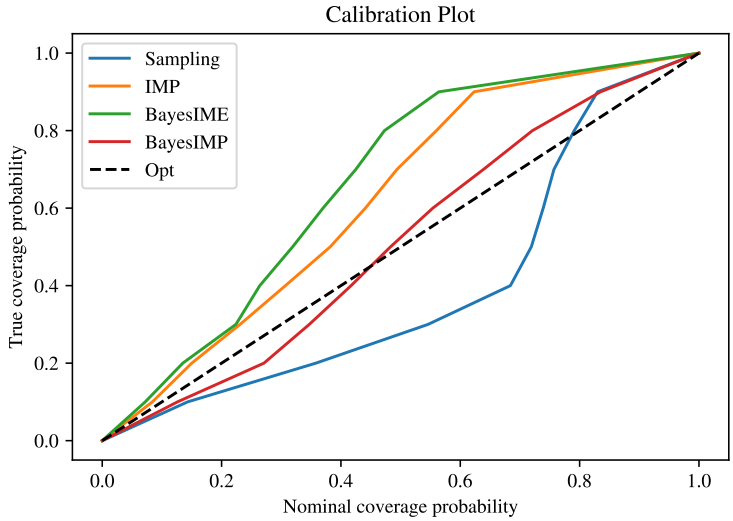


Figure B.2: Calibration plots of Sampling method as well as our 3 proposed methods. We clearly see that BAYESIMP is the best-calibrated method amongst all other methods.

B.4.2 Details on Synthetic Data experiments

B.4.2.1 Data Generating Process for simple synthetic dataset

For the first simple synthetic dataset (See Fig. 4.6 (Top)) we used the following data-generating graph is defined as.

- $X \to U : U = 2 * X + \epsilon$
- $Z \to X : X = 3 * \cos(Z) + \epsilon$
- $\{Z, U\} \to Y : Y = U + \exp(-Z) + \epsilon$
- $Y \to T : T = \cos(Y) - \exp(-y/20) + \epsilon$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and $Z \sim \mathcal{U}[-4, 4]$, where for $\mathcal{D}_2$ we have that $\tilde{Y} \sim \mathcal{U}[-10, 10]$. In addition, with probability $\pi = 1/2$ we shift U by $+1$ horizontally and -3 vertically to thus create the multimodality in the data. In order to generate from the interventional distribution, we simply remove the edge from $Z \to X$ and fix the value of x.

B.4.2.2 Data Generating Process for harder synthetic dataset from Aglietti et al. [2020b]

For the first simple synthetic dataset (Fig. 4.6(Bottom)) we used the same data generating format as in Aglietti et al. [2020b].

- $U_1 = \epsilon_1$
- $U_2 = \epsilon_2$
- $F = \epsilon_3$
- $A = F^2 + U_1 + \epsilon_A$
- $B = U_2 + \epsilon_B$
- $C = \exp(-B) + \epsilon_C$
- $D = \exp(-C)/10 + \epsilon_D$
- $E = \cos(A) + C/10\epsilon_E$
- $Y_1 = \cos(D) + \sin(E) + U_1 + U_2$
- $Y_2 = \cos(D) + \sin(E) + U_1 + U_2 + 2\pi$
- $T = 6 * \sin(3 * Y) + \epsilon$

where the noise is fixed to be $\mathcal{N}(0, 1)$ and where we switch with $\pi = 1/2$ from mode Y_1 and Y_2, where $\tilde{Y} \sim \mathcal{U}[-2, 9]$ for $\mathcal{D}_2$.

B.4.3 Details on Healthcare Data experiments

B.4.3.1 Data Generating Process

For the healthcare dataset, $\mathcal{D}_1$, (Fig. 4.1) we used the same data generating format as in Aglietti et al. [2020b] with the difference that we make *statin* continuous and increased the age range.

- $age = \mathcal{U}[15, 75]$
- $bmi = \mathcal{N}(27 - 0.01 * age, 0.7)$
- $aspirin = \sigma(-8.0 + 0.1 * age + 0.03 * bmi)$
- $statin = -13 + 0.1 * age + 0.2 * bmi$
- $cancer = \sigma(2.2 - 0.05 * age + 0.01 * bmi - 0.04 * statin + 0.02 * aspirin)$
- $PSA = \mathcal{N}(6.8 + 0.04 * age - 0.15 * bmi - 0.6 * statin + 0.55 * aspirin + cancer, 0.4)$

As for the second dataset, $\mathcal{D}_2$ we firstly fit a GP on the data collected from Stamey et al. [1989]. Once we have the posterior GP, we can then use it as a generator for the $\mathcal{D}_2$ as it takes as input PSA. This generator hence acts as a link between $\mathcal{D}_1$ and $\mathcal{D}_2$. This way we are able to create a simulator that allows us to obtain samples from $\mathbb{E}[\textit{Cancer volume}|do(Statin)]$ for our causal BO setup.

B.4.4 Bayesian Optimisation experiments with IMP and BAYESIME

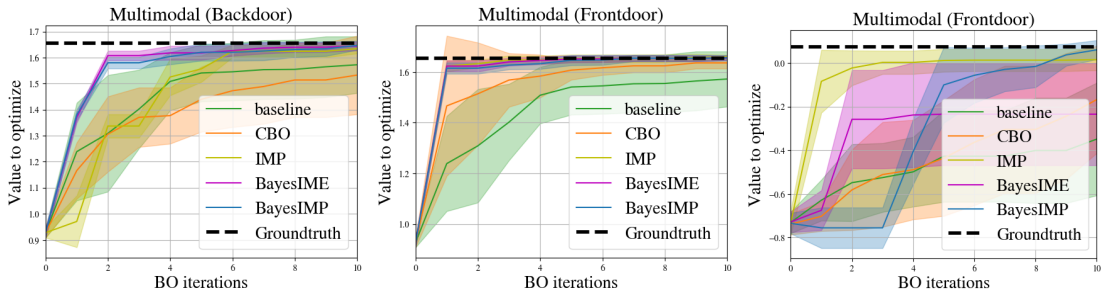


Figure B.3: (Left) Simple graph using backdoor adjustment (Middle) Simple graph using front-door adjustment (Right) Harder graph using front-door adjustment. BAYESIMP strikes the right balance between IMP and BAYESIME and all three perform better than CBO and the GP baseline.

The main text compares BAYESIMP to CBO and the baseline GP with no learned prior in the Bayesian Optimisation experiments. Here, we include IMP and BAYESIME (i.e. simplified versions of BAYESIMP that account for only one source of uncertainty each) in those comparisons. We see from Fig.B.3 that BAYESIMP is comparable to IMP and BAYESIME in most cases. While BAYESIMP is not the best-performing method in every scenario, it does hit a good middle ground between the first two proposed methods. For Fig. B.3 (Left, Middle) we used $N = 100$ and $M = 50$. In the left figure, BAYESIME and BAYESIMP are very similar, whereas IMP is considerably worst. In the middle figure, all methods seems to perform well without much difference. In the right figure, we have $N = 500$ and $M = 50$ and this is a case where IMP is best, while BAYESIME appears to get stuck in a local optimum (recall that BAYESIME does not take into account uncertainty in $\mathcal{D}_2$ where there is little data). We note that all three methods converge faster than the current SOTA CBO.

C | Chapter 5 Appendix

C.1 Computational complexity

The gains in speed-up and accuracy in RKHS-SHAP come from estimating $\nu^{(O)}_{\mathbf{x},S}$ using Conditional Mean Embeddings (CMEs). To compare with alternative approaches, it is sufficient to look at the complexity of estimating $\nu^{(O)}_{\mathbf{x},S}(f)$. For RKHS-SHAP this is $\mathcal{O}(Nd^2m) + \mathcal{O}(N^2d^2)$ where N is the number of data, d is the number of Fourier features which could be taken much smaller than N [Li et al., 2019a] and m is the number of conjugate gradient solver steps. Previous approaches would require some form of density estimation and Monte Carlo sampling, for which there are many methods, so we present a generic decomposition of complexity here: assuming we take L Monte Carlo samples for each x_{i_S} from $p(X_{S^c}|X_S = x_{i_S})$ to estimate $\nu^{(O)}_{\mathbf{x},S}(f)$, we have

$$\mathcal{O}(L^2N^2) + \mathcal{O}(\text{sampling } NL \text{ data from estimated densities}) + \mathcal{O}(\text{estimating } N \text{ conditional densities}).$$

It is not clear how to select L nor how fast it should grow with N. Aas et al. [2019] considered $L = N$ recovering a standard Nadaraya-Waston estimator for their empirical conditional mean estimator. In practice, for nonparametric methods, the computational cost is dominated by density estimation and sampling, both of which are not needed in our approach.

C.2 Comparison with Frye et al. [2020]

As mentioned in the main text, Frye et al. [2020]'s approach and ours share a similar regression-like intuition, thus we believe it is important to emphasize the technical difference between the methods.

- **Difference in regression target** Frye's approach regression onto scaler values $f(X)$ for specific f while RKHS-SHAP regresses onto an infinite dimensional feature map instead. Our model is aiming to capture representation of the full conditional distribution via the RKHS embedding, rather than the conditional expectation for a specific f.
- **Difference in dependency on** f CME estimation depends on the function space that f belongs to and not on the specific f. This subtle but crucial point allows onto apply Shapley functionals as attribution priors during the learning of f itself, in order to regularise it.
- **Difference in hypothesis space:** Frye's approach uses a scaler-valued parametric neural network model, while our approach uses an RKHS-valued non-parametric kernel ridge regression with a

ridge penalty to promote smoothness.

C.3 RKHS-SHAP for non-product kernels

When k is not a product kernel, such as the polynomial kernel and Matérn kernel, we can still proceed with estimating the value function using tools from conditional mean embeddings, and utilise our interpretability pipeline without the need for solving conditional density estimation tasks. To do so, we notice that for any $f \in \mathcal{H}_k$, we have

$$\nu_{x,S}(f) := \mathbb{E}_X[f(X) \mid X_S = x_S] \tag{C.1}$$

$$= \langle f, \mathbb{E}_X[\psi_X \mid X_S = x_S] \rangle_{\mathcal{H}_k} \tag{C.2}$$

$$= \left\langle f, \mu_{X|X_S=x_S} \right\rangle_{\mathcal{H}_k}. \tag{C.3}$$

Thus, we can proceed with the following estimator of $\mathbb{E}_X[\psi_X \mid X_S = x_S]$ using the standard conditional mean embedding estimator (with the conditioning variable being the subset of features): Denote $k_S : \mathcal{X}_S \times \mathcal{X}_S \to \mathbb{R}$ as a kernel defined on $\mathcal{X}_S$, where $\mathcal{X}_S$ is the subspace of the instance space of $\mathcal{X}$ according to S. Note that in principle, this kernel k_S need not be of the same form as the kernel k defined on the full feature space,

$$\hat{\mu}_{X|X_S=x_S} = \mathbf{K}_{x_S,X_S} \left(\mathbf{K}_{X_S,X_S} + n\lambda I\right)^{-1} \Psi_X^\top. \tag{C.4}$$

As a result, for $f = \sum_{i=1}^n \alpha_i k(\cdot, x_i)$, the corresponding non-parametric estimator of the value function $\nu_{x,S}(f)$ will be,

$$\bar{\nu}_{x,S}(f) = \mathbf{K}_{x_S,X_S} \left(\mathbf{K}_{X_S,X_S} + n\lambda I\right)^{-1} \mathbf{K}_{X,X}\boldsymbol{\alpha}. \tag{C.5}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top$.

Empirical demonstration In the following, we will demonstrate the above estimation procedure to explain a kernel ridge regression learnt using Matérn kernel, given by

$$k(x, x') = \frac{1}{\Gamma(v)2^{v-1}} \left(\frac{\sqrt{2v}}{l}\|x - x'\|\right)^v K_v \left(\frac{\sqrt{2v}}{l}\|x - x'\|\right) \tag{C.6}$$

where $v = 0.5$, K_v is the modified Bessel function of the second kind, and Γ is the gamma function. Kernel ridge regression is fitted on the diabetes and housing regression datasets from Appendix C.5.

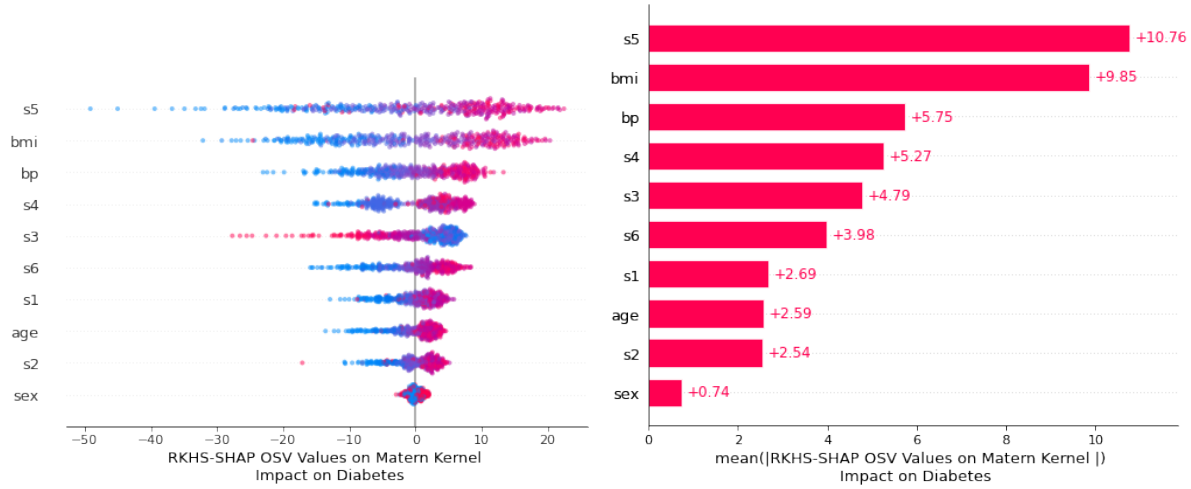


Figure C.1: Explaining a Kernel Ridge regression learnt using a Matérn kernel on the Diabetes regression dataset. In comparison to Figure C.5, where the KRR uses a Gaussian kernel, we see both models treat feature *s5*, *bp*, and *bmi* as top predictors, but having different emphasises on features *s3* and *s4*.

Figure C.1 and C.2 illustrated the explanation results coming from the kernel ridge regression with a Matérn kernel. We refer the reader to Appendix C.5 for a guide to interpret results from the beeswarm and bar plots.

In summary, the product kernel assumption is not required for the benefits of RKHS-SHAP to be brought to bear. Our proposed framework can thus be applied to essentially any kernel appropriate for the problem at hand. It is however, required to specify the form of the said kernel for any subset of features in the case of a non-product kernel, e.g. whether it again takes a Matérn form like the original kernel, or something else. Kernel hyperparameter learning will be more challenging than the product case as well, since e.g. lengthscale parameters typically vary with dimension and one would essentially require one lengthscale per subset of the features we are conditioning on, in contrast to the product case, where one lengthscale per feature dimension suffices. We might incur extra estimation error compared to the product kernel case as well. This is because one must fit the conditional mean embedding for any subset of features individually by regressing to the original RKHS defined on a higher-dimensional space (on all features d rather than on the subset $|S^c|$). As an example, if $d = 100$, in the non-product case we always perform estimation on the space of functions of 100 arguments, whereas in the product case, if one is conditioning on a $|S| = 99$ dimensional subset, this simplifies to estimation on the space of functions of a scaler argument. Not only is the learning problem harder, the non-product approach has to ignore the fact that the conditioning variable here is simply the subset of features – i.e. standard CME proceeds with regressing from features of X_S to features of X, while in the product case it is possible to simply isolate the features we condition on, and set them to the values of interests. As a result, the product kernel assumption allows us to circumvent potential statistical errors, and thus we chose to focus on the product kernel in the main text.

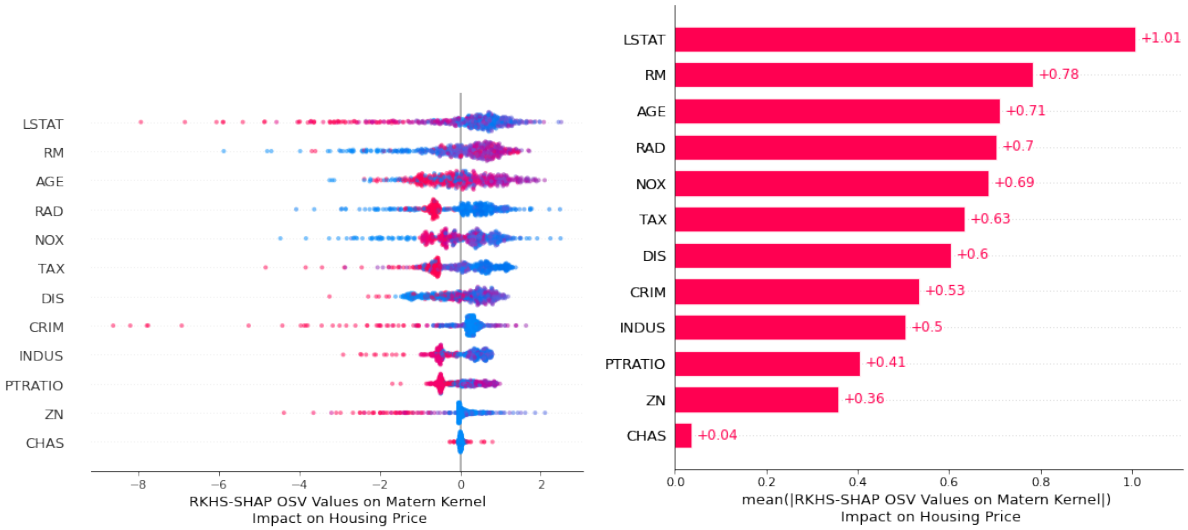


Figure C.2: Explaining a Kernel Ridge regression learnt using a Matérn kernel on the House price regression dataset. In comparison to Figure C.3, we see that *ZN* is no longer the top predictor. This illustrated that the models emphasised the feature *ZN* very differently.

C.4 Proofs

Proposition 5.3.1 (Riesz representations of value functionals)**.** *Denote k as the product kernel of D bounded kernels $k_d : \mathcal{X}^{(d)} \times \mathcal{X}^{(d)} \to \mathbb{R}$, where $\mathcal{X}^{(d)}$ is the d^{th} feature space. The Riesz representations of the Interventional value functional and Observational value functional exist and have the following forms in $\mathcal{H}_k$,*

$$\nu^{(I)}_{x,S} = \psi_{x_S} \otimes \mu_{X_{S^c}} \tag{C.7}$$

$$\nu^{(O)}_{x,S} = \psi_{x_S} \otimes \mu_{X_{S^c}|X_S=x_S} \tag{C.8}$$

where $\psi_{x_S} := \bigotimes_{i\in S} \psi_{x^{(i)}}$, $\mu_{X_{S^c}} := \mathbb{E}[\bigotimes_{i\in S^c} \psi_{x^{(i)}}]$ and $\mu_{X_{S^c}|X_S=x_S} := \mathbb{E}[\bigotimes_{i\in S^c} \psi_{x^{(i)}}|X_S = x_S]$.

Proof. Since $\nu^{(I)}_{x,S}$ and $\nu^{(O)}_{x,S}$ are bounded linear functionals on all $f \in \mathcal{H}_k$ with $||f||_{\mathcal{H}_k}$ bounded, Riesz representation theorem [Paulsen and Raghupathi, 2016] tells us there exist $r_{\nu^{(I)}_{x,S}}$ and $r_{\nu^{(O)}_{x,S}}$ living in $\mathcal{H}_k$ such that $\nu^{(I)}_{x,S}(f) = \langle f, r_{\nu^{(I)}_{x,S}} \rangle$ and $\nu^{(O)}_{x,S}(f) = \langle f, r_{\nu^{(O)}_{x,S}} \rangle$. If fact, if we set $r_{\nu^{(I)}_{x,S}}$ to be $\psi_{x_S} \otimes \mu_{X_{S^c}}$ and $r_{\nu^{(O)}_{x,S}}$ to be $\psi_{x_S} \otimes \mu_{X_{S^c}|X_S=x_S}$, then for the former, we have,

$$\langle f, \psi_{x_S} \otimes \mu_{X_{S^c}} \rangle = \mathbb{E}[\langle f, \psi_{x_S} \otimes \psi_{X_{S^c}} \rangle] \tag{C.9}$$

$$= \mathbb{E}[f(\{x_S, X_{S^c}\}] \tag{C.10}$$

Similarly,

$$\langle f, \psi_{x_S} \otimes \mu_{X_{S^c}|X_S=x_S} \rangle = \mathbb{E}[\langle f, \psi_{x_S} \otimes \psi_{X_{S^c}} \rangle | X_S = x_S] \tag{C.11}$$

$$= \mathbb{E}[f(\{x_S, X_{S^c}\}|X_S = x_S] \tag{C.12}$$

□

Proposition 5.3.2. *Given $\mathbf{x}' \in \mathbb{R}^{n'}$ a vector of instances and $f = \Psi_{\mathbf{x}}\boldsymbol{\alpha}$, the empirical estimates of $\nu^{(I)}_{\mathbf{x}',S}(f)$ and $\nu^{(O)}_{\mathbf{x}',S}(f)$ can be computed as,*

$$\hat{\nu}^{(I)}_{\mathbf{x}',S}(f) = \boldsymbol{\alpha}^\top \mathcal{K}^{(I)}_{\mathbf{x}',S} \qquad \hat{\nu}^{(O)}_{\mathbf{x}',S}(f) = \boldsymbol{\alpha}^\top \mathcal{K}^{(O)}_{\mathbf{x}',S} \tag{C.13}$$

where $\mathcal{K}^{(I)}_{\mathbf{x}',S} = \left(\mathbf{K}_{\mathbf{x}_S\mathbf{x}'_S} \odot \frac{1}{n}\operatorname{diag}(\mathbf{K}^\top_{\mathbf{x}_{S^c}\mathbf{x}'_{S^c}}\mathbf{1_n})\mathbf{1_n}\mathbf{1_n}^\top\right)$ and $\mathcal{K}^{(O)}_{\mathbf{x}',S} = \left(\mathbf{K}_{\mathbf{x}_S\mathbf{x}'_S} \odot \Xi_S\mathbf{K}_{\mathbf{x}_S\mathbf{x}'_S}\right)$, $\mathbf{1_n}$ is the all-one vector with length n, $\odot$ the Hadamard product and $\Xi_S = \mathbf{K}_{\mathbf{x}_{S^c}\mathbf{x}_{S^c}}(\mathbf{K}_{\mathbf{x}_S\mathbf{x}_S} + n\eta I)^{-1}$

Proof. Consider x a single observation. Recall $f = \Psi_{\mathbf{x}}\boldsymbol{\alpha}$ and $\Psi_{\mathbf{x}} = [\psi_{x_1}...\psi_{x_n}] = [\psi_{x_{1S}} \otimes \psi_{x_{1S^c}}...\psi_{x_{nS}} \otimes \psi_{x_{nS^c}}]$. To compute $\hat{\nu}^{(I)}_{x,S}(f)$, we have:

$$\hat{\nu}^{(I)}_{x,S}(f) = \langle f, \psi_{x_S} \otimes \hat{\mu}_{X_{S^c}} \rangle \tag{C.14}$$

$$= \langle \Psi_{\mathbf{x}}\boldsymbol{\alpha}, \psi_{x_S} \otimes \frac{1}{n}\sum_{i=1}^{n} \psi_{x_{iS^c}} \rangle \tag{C.15}$$

$$= \boldsymbol{\alpha}^\top \left(\mathbf{K}_{\mathbf{x}_S x_S} \times \frac{1}{n}\mathbf{K}^\top_{\mathbf{x}_{S^c} x_{S^c}}\mathbf{1}_n\right) \tag{C.16}$$

Similarly, for $\nu^{(O)}_{x,S}(f)$,

$$\hat{\nu}^{(O)}_{x,S}(f) = \langle f, \psi_{x_S} \otimes \hat{\mu}_{X_{S^c}|X_S=x_S} \rangle \tag{C.17}$$

$$= \langle \Psi_{\mathbf{x}}\boldsymbol{\alpha}, \psi_{x_S} \otimes \Psi_{\mathbf{x}_{S^c}}(\mathbf{K}_{\mathbf{x}_S\mathbf{x}_S} + \eta I)^{-1}\mathbf{K}_{\mathbf{x}_S x_S} \rangle \tag{C.18}$$

$$= \boldsymbol{\alpha}^\top \left(\mathbf{K}_{\mathbf{x}_S x_S} \odot \mathbf{K}_{\mathbf{x}_{S^c}\mathbf{x}_{S^c}}(\mathbf{K}_{\mathbf{x}_S\mathbf{x}_S} + n\eta I)^{-1}\mathbf{K}_{\mathbf{x}_S x_S}\right) \tag{C.19}$$

Extension to a vector of instance $\mathbf{x}'$ is then straightforward. □

Proposition 5.3.3 (RKHS-SHAP). *Given $f \in \mathcal{H}_k$ and a value functional ν, Shapley values for all d features and all input $\mathbf{x}$ can be computed as follows:*

$$\mathbf{B} = (Z^\top W Z)^{-1} Z^\top W \hat{\mathbf{V}} \tag{C.20}$$

where $\hat{\mathbf{V}}_{i,:} = \langle f, \hat{\nu}_{\mathbf{x},S_i} \rangle$.

Proof. Since we now have a compact way to estimate the conditional estimations for a vector of observations using mean embeddings, we can restate the KernelSHAP objective, which essentially is a weighted least regression, into a multi-output weighted least square formulation. □

Proposition 5.3.4 (Shapley functional). *Given a value functional ν indexed by input x and coalition S, the Shapley functional $\phi_{x,i} : \mathcal{H}_k \to \mathbb{R}$ such that $\phi_{x,i}(f)$ is the i^{th} Shapley values for model f on input x, has the following Reisz representation in the RKHS,*

$$\phi_{x,i} = \frac{1}{d} \sum_{S \subseteq D \setminus \{i\}} \binom{d-1}{|S|}^{-1} \left(\nu_{x,S \cup i} - \nu_{x,S} \right) \tag{C.21}$$

Proof. Since the Shapley functional is a linear combination of bounded linear functionals (value functionals), it admits a Riesz representer in the RKHS.

□

Theorem 5.3.5 (Bounding Shapley functionals). *Let k be a product kernel with d bounded kernels $|k^{(i)}(x,x)| \leq M$ for all $i \in D$. Denote $M_\mu := \sup_{S \subseteq D} M^{|S|}$, $M_\Gamma := \sup_{S \subseteq D} ||\mu_{X_{S^c}|X_S}||^2_{\Gamma_{X_S}}$ and $L_\delta = \sup_{S \subseteq D} ||\psi_{x_S} - \psi_{x'_S}||^2_{\mathcal{H}_k}$. Let $\delta > 0$, assume $|x^{(i)} - x^{(i)\prime}|^2 \leq \delta$ for all features $i \in D$, then differences of the Interventional and Observational Shapley functionals for feature i at observation x, x' can be bounded as $||\phi^{(I)}_{x,i} - \phi^{(I)}_{x',i}||^2_{\mathcal{H}_k} \leq 2 M_\mu L_\delta$ and $||\phi^{(O)}_{x,i} - \phi^{(O)}_{x',i}||^2_{\mathcal{H}_k} \leq 4 M_\Gamma M_\mu L_\delta$. If k is the RBF kernel with lengthscale l, then*

$$||\phi^{(I)}_{x,i} - \phi^{(I)}_{x',i}||^2_{\mathcal{H}_k} \leq 4\left(1 - \exp\left(\frac{-d\delta}{2l^2}\right)\right) \tag{C.22}$$

$$||\phi^{(O)}_{x,i} - \phi^{(O)}_{x',i}||^2_{\mathcal{H}_k} \leq 8 M_\Gamma \left(1 - \exp\left(\frac{-d\delta}{2l^2}\right)\right) \tag{C.23}$$

Proof. To prove that Shapley functionals between two observations x and x' are δ close when the two points are closed, we proceed as follows: (1) We show that when one pick the usual product RBF kernel,

we can bound the distance of the feature maps as a function of δ. (2) We then upper bound the value functionals and show that this bound can be relaxed so that it is independent with the choice of coalition. (3) Since Shapley values is an expectation of differences of value functions, by devising a coalition independent bound for the difference in value functionals, the expectation disappears in our bound. □

Proposition A.1 (Bounding feature maps)**.** *For the simplest 1 dimensional case with $|x - x'|^2 \leq \delta$, if we pick k the standard RBF kernel with lengthscale l, we have,*

$$||\psi_x - \psi_{x'}||^2_{\mathcal{H}_k} \leq 2 - 2\exp\Big(-\frac{\delta}{2l^2}\Big) \tag{C.24}$$

When we pick $x, x' \in \mathbb{R}^d$ and with a product RBF kernel i.e $k(x, x') = \prod_{j=1}^{d} k^j(x^{(j)}, x'^{(j)})$, where $k^{(j)}$ themselves RBF kernels. For simplicity, we assume they all share the same lengthscale l. If $|x^{(j)} - x'^{(j)}| \leq \delta$ for all $j \in D$, then we can bound the difference in feature maps as follows,

$$||\psi_x - \psi_{x'}||^2_{\mathcal{H}_k} \leq 2 - 2\exp\Big(-\frac{d\delta}{2l^2}\Big) \tag{C.25}$$

Proof. Since $||\psi_x - \psi_{x'}||^2_{\mathcal{H}_k} = k(x, x) + k(x, x') - 2k(x, x')$. Therefore the first 2 terms are 1 and we can bound the last term since,

$$k(x, x') = \exp\Big(-\frac{|x - x'|^2}{2l^2}\Big) \geq \exp\Big(-\frac{\delta}{2l^2}\Big) \tag{C.26}$$

Multiply this lower bound d times to obtain the bound for the d dimensional case. □

Proposition A.1 tells us how the distance in feature maps $||k_x - k_{x'}||_{\mathcal{H}_k}$ can be expressed by the distance between x and x' in the RBF kernel. Different bounds can be derived for different kernels and we only show the special RBF case for illustration purpose.

Now we shall prove a bound for the value functionals. We shall first proceed with the interventional case and move on to observational afterwards.

Proposition A.2 (Bounding Interventional value functionals)**.** *For a fix coalition S, denote $D_S^{(I)} = ||\nu_{x,S}^{(I)} - \nu_{x',S}^{(I)}||_{\mathcal{H}_k}^2$. Then $D_S^{(I)} \leq ||\psi_{x_S} - \psi_{x'_S}||_{\mathcal{H}_{k_S}}^2 ||\mu_{X_{S^c}}||_{\mathcal{H}_{k_{S^c}}}^2$. Let $L_\delta := \sup_{S \subseteq D} ||\psi_{x_S} - \psi_{x'_S}||_{\mathcal{H}_{k_S}}^2$ and assume kernels are all bounded per dimension by M, i.e $k^{(j)}(x, x') \leq M$ for all $j \in D$. Denote $M_\mu := \sup_{S \subseteq D} M^{|S|}$. Then the bound can be further loosen up,*

$$D_S^{(I)} \leq M_\mu L_\delta \tag{C.27}$$

Proof.

$$D_S^{(I)} = ||\nu_{x,S}^{(I)} - \nu_{x',S}^{(I)}||_{\mathcal{H}_k}^2 \tag{C.28}$$

$$= ||\psi_{x_S} \otimes \mu_{X_{S^c}} - \psi_{x'_S} \otimes \mu_{X_{S^c}}||_{\mathcal{H}_k}^2 \tag{C.29}$$

$$\leq ||\psi_{x_S} - \psi_{x'_S}||_{\mathcal{H}_{k_S}}^2 ||\mu_{X_{S^c}}||_{\mathcal{H}_{k_{S^c}}}^2 \tag{C.30}$$

Note that $||\mu_{X_{S^c}}||^2 = ||\mathbb{E}[k(X_{S^c}, X'_{S^c})]||^2 \leq M^{2|S^c|}$, therefore,

$$\leq L_\delta M_\mu \tag{C.31}$$

□

Before we prove the main theorem, we will illustrate the following bounds for conditional mean embeddings, which will be used to bound the observational Shapley functionals.

Proposition A.3 (Bounding conditional mean embeddings)**.** *If we take on the vector-valued function perspective of conditional mean embeddings as in Grünewälder et al. [2012], then we could assume, in general for random variables Y and X, there exists a function $\mu_{Y|X} \in \mathcal{H}_{\Gamma_x}$ where $\Gamma_x : \mathcal{X} \times \mathcal{X} \mapsto \mathcal{L}(\mathcal{H}_\ell)$ with $\mathcal{L}(\mathcal{H}_\ell)$ being the space of self-adjoint operators from the RKHS $\mathcal{H}_\ell$ to itself, is the vector-valued kernel $\Gamma_x(x, x') = k(x, x')\mathbf{1}$, such that $\mu_{Y|X}(x) = \mu_{Y|X=x}$. If we assume such a function exists, then by definition of vector-valued RKHSs as in Park and Muandet [2020], $||\mu_{Y|X}||_{\mathcal{H}_{\Gamma_x}}$ has finite norm. Therefore the following is defined if the base kernel k is bounded,*

$$||\mu_{Y|X=x}||_{\mathcal{H}_\ell} \leq ||\mu_{Y|X}||_{\mathcal{H}_{\Gamma_x}} ||\psi_x||_{\mathcal{H}_k} \tag{C.32}$$

and correspondingly,

$$||\mu_{Y|X=x} - \mu_{Y|X=x'}||_{\mathcal{H}_\ell} \leq ||\mu_{Y|X}||_{\mathcal{H}_{\Gamma_x}} ||\psi_x - \psi_{x'}|| \tag{C.33}$$

Proof. For the first claim, using the result from Micchelli and Pontil [2005, Prop.1], we have,

$$||\mu_{Y|X}(x)||_{\mathcal{H}_\ell} \leq ||\mu_{Y|X}||_{\mathcal{H}_{\Gamma_x}} ||\Gamma_x(x, x)||_{op}^{\frac{1}{2}} \tag{C.34}$$

however, we have,

$$||\Gamma_x(x, x)||_{op} = \sup_{g \in \mathcal{H}_\ell} \frac{||k(x, x)g||_{\mathcal{H}_\ell}}{||g||_{\mathcal{H}_\ell}} = |k(x, x)| = ||\psi_x||^2 \tag{C.35}$$

For the second part, we start with,

$$||\mu_{Y|X=x} - \mu_{Y|X=x'}||_{\mathcal{H}_\ell} \leq ||\mu_{Y|X}||_{\mathcal{H}_{\Gamma_x}} ||\Gamma_x(\cdot, x) - \Gamma_x(\cdot, x')||_{op} \tag{C.36}$$

Using the result from Micchelli and Pontil [2005, Prop.1] again, we have

$$||\Gamma_x(\cdot, x) - \Gamma_x(\cdot, x')||_{op} = ||\big(\Gamma_x(\cdot, x) - \Gamma_x(\cdot, x')\big)^* \big(\Gamma_x(\cdot, x) - \Gamma_x(\cdot, x')\big)||_{op}^{\frac{1}{2}} \tag{C.37}$$

where the $*$ denotes the adjoint of the operator,

$$= ||\Gamma_x(\cdot,x)^*\Gamma_x(\cdot,x) - 2\Gamma_x(\cdot,x)^*\Gamma_x(\cdot,x') + \Gamma_x(\cdot,x')^*\Gamma_x(\cdot,x')||_{op}^{\frac{1}{2}} \quad \text{(C.38)}$$

$$= ||\Gamma_x(x,x) - 2\Gamma_x(x,x') + \Gamma_x(x',x')||_{op}^{\frac{1}{2}} \quad \text{(C.39)}$$

$$= ||\big(k(x,x) - 2k(x,x') + k(x',x')\big)\mathbf{1}||_{op}^{\frac{1}{2}} \quad \text{(C.40)}$$

$$= ||\psi_x - \psi_{x'}||_{\mathcal{H}_k} \quad \text{(C.41)}$$

therefore we have as a result,

$$||\mu_{Y|X=x} - \mu_{Y|X=x'}||_{\mathcal{H}_\ell} \leq ||\mu_{Y|X}||_{\mathcal{H}_{\Gamma_x}}||\psi_x - \psi_{x'}||_{\mathcal{H}_k} \quad \text{(C.42)}$$

□

Proposition A.4 (Bounding Observational value functionals via vector-valued function perspective of CME)**.** *For a fix coalition S, denote $D_S^{(O)} = ||\nu_{x,S}^{(I)} - \nu_{x',S}^{(I)}||_{\mathcal{H}_k}^2$. Then*

$$D_S^{(O)} \leq ||\psi_{x_S} - \psi_{x'_S}||_{\mathcal{H}_{k_S}}^2 ||\mu_{X_{S^c}|X_S}||_{\mathcal{H}_{\Gamma_{X_S}}}^2 \left(||\psi_{x_S}||_{\mathcal{H}_{k_S}}^2 + ||\psi_{x'_S}||_{\mathcal{H}_{k_S}}^2\right)$$

, where $\mathcal{H}_{\Gamma_{X_S}}$ is the $\mathcal{H}_{k_{S^c}}$-valued RKHS. If we denote $L_\delta = \sup_{S\subseteq D} ||\psi_{x_S} - \psi_{x'_S}||_{\mathcal{H}_{k_S}}^2$ and $M_\mu = \sup_{S\subseteq D} M^{|S|}$ and $M_\Gamma = \sup_{S\subseteq D} ||\mu_{X_{S^c}|X_S}||_{\mathcal{H}_{\Gamma_{X_S}}}^2$. Then $D_S^{(O)} \leq 2M_\Gamma M_\mu L_\delta$ for all coalition S.

Proof.

$$D_S^{(O)} = ||\nu_{x,S}^{(O)} - \nu_{x',S}^{(O)}||_{\mathcal{H}_k}^2 \tag{C.43}$$

$$= ||\psi_{x_S} \otimes \mu_{X_{S^c}|X_S=x_S} - \psi_{x'_S} \otimes \mu_{X_{S^c}|X_S=x'_S}||_{\mathcal{H}_k}^2 \tag{C.44}$$

$$= ||\psi_{x_S} \otimes \mu_{X_{S^c}|X_S=x_S} - \psi_{x'_S} \otimes \mu_{X_{S^c}|X_S=x_S} + \psi_{x'_S} \otimes \mu_{X_{S^c}|X_S=x_S} - \psi_{x'_S} \otimes \mu_{X_{S^c}|X_S=x'_S}||_{\mathcal{H}_k}^2 \tag{C.45}$$

$$\leq ||\psi_{x_S} - \psi_{x'_S}||_{\mathcal{H}_{k_S}}^2 ||\mu_{X_{S^c}|X_S=x_s}||_{\mathcal{H}_{k_{S^c}}}^2 + ||\psi_{x'_S}||_{\mathcal{H}_{k_S}}^2 ||\mu_{X_{S^c}|X_S=x_S} - \mu_{X_{S^c}|X_S=x'_S}||_{\mathcal{H}_{k_{S^c}}}^2 \tag{C.46}$$

$$\leq ||\psi_{x_S} - \psi_{x'_S}||_{\mathcal{H}_{k_S}}^2 ||\mu_{X_{S^c}|X_S}||_{\mathcal{H}_{\Gamma_{X_S}}}^2 ||\psi_{x_S}||_{\mathcal{H}_{k_S}}^2 + ||\psi_{x'_S}||_{\mathcal{H}_{k_S}}^2 ||\mu_{X_{S^c}|X_S}||_{\mathcal{H}_{\Gamma_{X_S}}}^2 ||\psi_{x_S} - \psi_{x'_S}||_{\mathcal{H}_{k_S}}^2 \tag{C.47}$$

$$= ||\psi_{x_S} - \psi_{x'_S}||_{\mathcal{H}_{k_S}}^2 ||\mu_{X_{S^c}|X_S}||_{\mathcal{H}_{\Gamma_{X_S}}}^2 \left(||\psi_{x_S}||_{\mathcal{H}_{k_S}}^2 + ||\psi_{x'_S}||_{\mathcal{H}_{k_S}}^2\right) \tag{C.48}$$

$$\leq 2M_\Gamma M_\mu L_\delta \tag{C.49}$$

Finally, we note that,

$$||\phi_{x,i} - \phi_{x',i}||_{\mathcal{H}_k}^2 = ||\frac{1}{d} \sum_{S \subseteq D \setminus \{i\}} \binom{d-1}{|S|}^{-1} \nu_{x,S\cup i} - \nu_{x,S} - (\nu_{x',S\cup i} - \nu_{x',S})||_{\mathcal{H}_k}^2 \tag{C.50}$$

$$\leq \frac{1}{d} \sum_{S \subseteq D \setminus \{i\}} \binom{d-1}{|S|}^{-1} D_S + D_{S\cup i} \tag{C.51}$$

$$= \mathbb{E}_S[D_S + D_{S\cup i}] \tag{C.52}$$

Since we have proven bounds for $D_S^{(O)}$ and $D_S^{(I)}$ that is coalition independent, we can directly substitute the bound inside the expectation. Therefore

$$||\phi_{x,i}^{(I)} - \phi_{x',i}^{(I)}||_{\mathcal{H}_k}^2 \leq 2L_\delta M_\mu \tag{C.53}$$

$$||\phi_{x,i}^{(O)} - \phi_{x',i}^{(O)}||_{\mathcal{H}_k}^2 \leq 4M_\Gamma L_\delta M_\mu \tag{C.54}$$

In the case when we pick k as a product RBF kernel, we have $L_\delta = 2 - 2\exp\left(\frac{d\delta}{2l^2}\right)$ and $M_\mu = 1$, therefore,

$$||\phi_{x,i}^{(I)} - \phi_{x',i}^{(I)}||_{\mathcal{H}_k}^2 \leq 4\left(1 - \exp\left(\frac{-d\delta}{2l^2}\right)\right) \tag{C.55}$$

$$||\phi_{x,i}^{(O)} - \phi_{x',i}^{(O)}||_{\mathcal{H}_k}^2 \leq 8M_\Gamma\left(1 - \exp\left(\frac{-d\delta}{2l^2}\right)\right) \tag{C.56}$$

□

Proposition 5.4.1. *The above optimisation can be rewritten as,* $\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \sum_{i=1}^n \ell(y_i, \mathbf{K}_{x_i\mathbf{x}}\boldsymbol{\alpha}) + \lambda_f \boldsymbol{\alpha}^\top \mathbf{K}_{\mathbf{xx}}\boldsymbol{\alpha} + \frac{\lambda_S}{n}\boldsymbol{\alpha}^\top \zeta_A \zeta_A^\top \boldsymbol{\alpha}$. *To regularise the Interventional SVs (*ISV-REG*) of* A*, we set* $\zeta_A = \frac{1}{J}\sum_{j=1}^J \mathcal{K}^{(I)}_{\mathbf{x},S_j\cup A} - \mathcal{K}^{(I)}_{\mathbf{x},S_j}$ *where* S_j*'s are coalitions sampled from* $p_{SV}(S) = \frac{1}{d}\binom{d-1}{|S|}^{-1}$. *For regularising Observational SVs (*OSV-REG*), we set* $\zeta_A = \frac{1}{J}\sum_{j=1}^J \mathcal{K}^{(O)}_{\mathbf{x},S_j\cup A} - \mathcal{K}^{(O)}_{\mathbf{x},S_j}$.

Sketch proof. To express

$$\min_{f\in\mathcal{H}_k} \sum_{i=1}^n \ell(y_i, f(x_i)) + \lambda_f ||f||^2_{\mathcal{H}_k} + \frac{\lambda_S}{n}\sum_{i=1}^n |\phi_{x_i,A}(f)|^2$$

as

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \sum_{i=1}^n \ell(y_i, \mathbf{K}_{x_i\mathbf{x}}\boldsymbol{\alpha}) + \lambda_f \boldsymbol{\alpha}^\top \mathbf{K}_{\mathbf{xx}}\boldsymbol{\alpha} + \frac{\lambda_S}{n}\boldsymbol{\alpha}^\top \zeta_A \zeta_A^\top \boldsymbol{\alpha},$$

it suffices to show that $\frac{\lambda_S}{n}\sum_{i=1}^n |\phi_{x_i,A}(f)|^2 = \frac{\lambda_S}{n}\boldsymbol{\alpha}^\top \zeta_A\zeta_A^\top\boldsymbol{\alpha}$. However, note that

$$\sum_{i=1}^n |\phi_{x_i,A}(f)|^2 = \phi_{\mathbf{x},A}(f)^\top \phi_{\mathbf{x},A}(f) \tag{C.57}$$

$$= f^\top \phi_{\mathbf{x},A}\phi^\top_{\mathbf{x},A} f \tag{C.58}$$

Now we can estimate the Shapley functional $\phi_{\mathbf{x},A}$ defined in Proposition 5, by applying the finite sample estimator of the value functions from Proposition 2, we can compute the finite sample estimate of $\phi^\top_{\mathbf{x},A} f$ as $\zeta_A^\top \boldsymbol{\alpha}$. □

C.5 Further experiment details

C.5.1 Banana Distribution $\mathcal{B}(b^{-1}, v)$

Recall the Banana distribution $\mathcal{B}(b^{-1}, v)$ is defined as follows: Let $Z \sim N(0, \text{diag}(v, 1))$and set $X_1 = Z_1$ and $X_2 = b^{-1}(Z_1^2 - v) + Z_2$. We define $f(x) = b^{-1}(x_1^2 - v) + x_2$. Now then we have,

$$\mathbb{E}[f(X)] = 0 \tag{C.59}$$

$$\mathbb{E}[f(X)|X_1 = x_1] = 2b^{-1}(x_1^2 - v) \tag{C.60}$$

$$\mathbb{E}[f(X)|X_2 = x_2] = 2x_2 \tag{C.61}$$

$$\mathbb{E}[f(X)|do(X_1) = x_1] = b^{-1}(x_1^2 - v) \tag{C.62}$$

$$\mathbb{E}[f(X)|do(X_2) = x_2] = x_2 \tag{C.63}$$

This corresponds to the following Observational Shapley values,

$$\begin{aligned}
\phi_{x,1}^{(O)}(f) &= \frac{1}{2}\left[\binom{1}{0}^{-1}\left(\mathbb{E}f\left(\mathbf{X}|X_1 = x_1\right) - \mathbb{E}f\left(\mathbf{X}\right)\right) + \binom{1}{1}^{-1}\left(\mathbb{E}f\left(\mathbf{X}|X_1 = x_1, X_2 = x_2\right) - \mathbb{E}f\left(\mathbf{X}|X_2 = x_2\right)\right)\right] \\
&= \frac{1}{2}\left(3b^{-1}\left(x_1^2 - v\right) - x_2\right). \\
\phi_{x,2}^{(O)}(f) &= \frac{1}{2}\left[\binom{1}{0}^{-1}\left(\mathbb{E}f\left(\mathbf{X}|X_2 = x_2\right) - \mathbb{E}f\left(\mathbf{X}\right)\right) + \binom{1}{1}^{-1}\left(\mathbb{E}f\left(\mathbf{X}|X_1 = x_1, X_2 = x_2\right) - \mathbb{E}f\left(\mathbf{X}|X_1 = x_1\right)\right)\right] \\
&= \frac{1}{2}\left(3x_2 - b^{-1}\left(x_1^2 - v\right)\right)
\end{aligned}$$

Similarly, for Interventional Shapley values we have,

$$\phi_{x,1}^{(I)}(f) = b^{-1}(x_1^2 - v)$$
$$\phi_{x,2}^{(I)}(f) = x_2$$

C.5.2 RKHS-SHAP on real-world examples

We demonstrate the result of running RKHS-SHAP on 6 real-world datasets and showcase their RKHS-SHAP Observational Shapley values in Beeswarm summary plots. Interventional SVs are omitted because we have shown in the main text that running KernelSHAP-ISV and RKHS-SHAP-ISV gives you the same SVs, and they only differ in computational run time.

These results are not included in the main text because **we do not observe the actual data distribution,**

Table C.1: Real-world explanation tasks

Dataset	$n_{instances}$	$n_{features}$	Downstream task
Boston Housing	506	12	Predict Boston House Price (Regression)
Diabetes Progression	442	10	Predict diabetes progression (Regression)
Diabetes for Pima Indian Heritage	768	8	Predict whether a patient has diabetes (Classification)
Breast Cancer	569	30	Predict whether a patient might have breast cancer or not (Classification)
Census Income	48,842	14	Predict whether an individual is making over $50k a year (Classification)
League of Legends Win Prediction	1,800,000	71	Predict the winning probability of a player (Classification)

thus there are no groundtruth observational SVs that our algorithm can be compared to measure and verify how well it is performing. In the following, all models are fitted with the Gaussian kernel. We first fit a Kernel Ridge Regression or Kernel Logistic Regression to learn the function f, and apply RKHS-SHAP to f to recover the corresponding observational Shapley values.

We present our results using Beeswarm plot and bar plot. According to the **shap** package, the beeswarm plot is designed to display an information-dense summary of how the top features in the dataset impact the model's output. Each instance the given explanation is represented by a single dot on each feature row. The x position of the dot is determined by the RKHS-SHAP value of that feature, and dots "pile up" along each feature row to show density. Colour is used to display the original value of a feature, which is scaled with red indicating high, and blue indicating low values. On the other hand, the bar plot shows the mean absolute value of the Shapley values per feature, thus providing some global summary based on recovered local importance.

We summarise our real-world explanation tasks in table C.1.

Boston Housing The Boston house price dataset[1] contains 506 instances and 12 numerical features. Below is the description of its features:

CRIM per capita crime rate by town

ZN proportion of residential land zoned for lots over 25k sq.ft

INDUS proportion of non-retail business acres per town

[1] https://archive.ics.uci.edu/ml/machine-learning-databases/housing/

CHAS	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
NOX	nitric oxides concentration (parts per 10 million)
RM	average number of rooms per dwelling
AGE	proportion of owner-occupied units built prior to 1940
DIS	weighted distances to five Boston employment centres
RAD	index of accessibility to radial highways
TAX	full-value property-tax rate per $10,000
PTRATIO	pupil-teacher ratio by town
LSTAT	% lower status of the population
MEDV	Median value of owner-occupied homes in $1000's

We fit a Kernel Ridge Regression to predict the Boston house price. The results are shown in Fig. C.3. We see that RKHS-SHAP does capture several intuitive explanations, e.g. Higher crime rate (red dots in feature CRIM) corresponds to negative impact on the house price. We also recover explanations such as lower percentage of lower status of the population (LSTAT) will increase the house price.

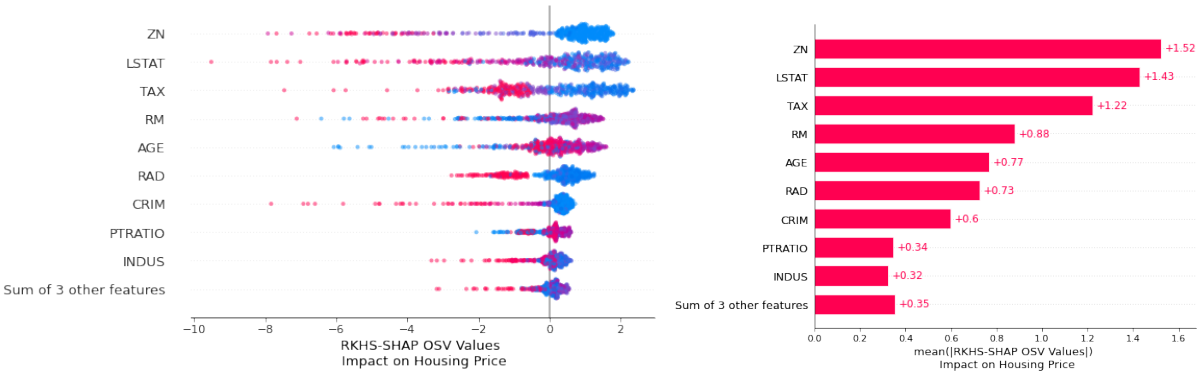


Figure C.3: Beeswarm and bar plot for the housing dataset.

We can examine specific houses and interpret why the kernel ridge regression predicts their corresponding house prices as well. See Fig C.4.

Diabetes progression regression Next we apply RKHS-SHAP to the diabetes[2] dataset with 442 samples and 10 features. The machine learning task is to model the disease progression of patients as a regression problem. We fit a kernel ridge regression for that. Figure C.5 records the results. Feature $s1$ to

[2]https://www4.stat.ncsu.edu/ boos/var.select/diabetes.html

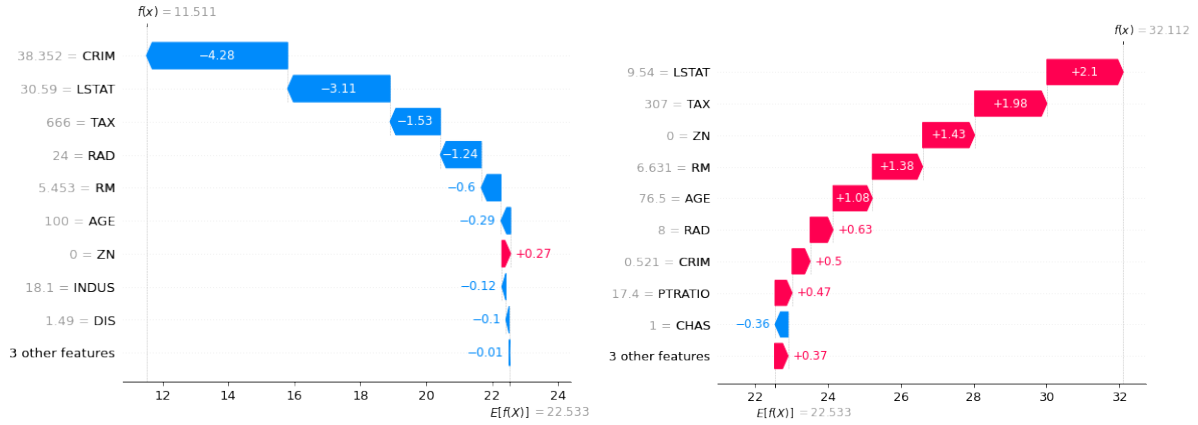


Figure C.4: (left) The algorithm believes having a high crime rate is the major reason for its low house price. (Right) Having a high LSTAT increased the house price.

$s6$ are blood serum measurements. We note that bmi is one of the most influential feature, which follows our intuition that higher value of bmi (red clusters in the bmi row) should be a strongly predictive variable to diabetes.

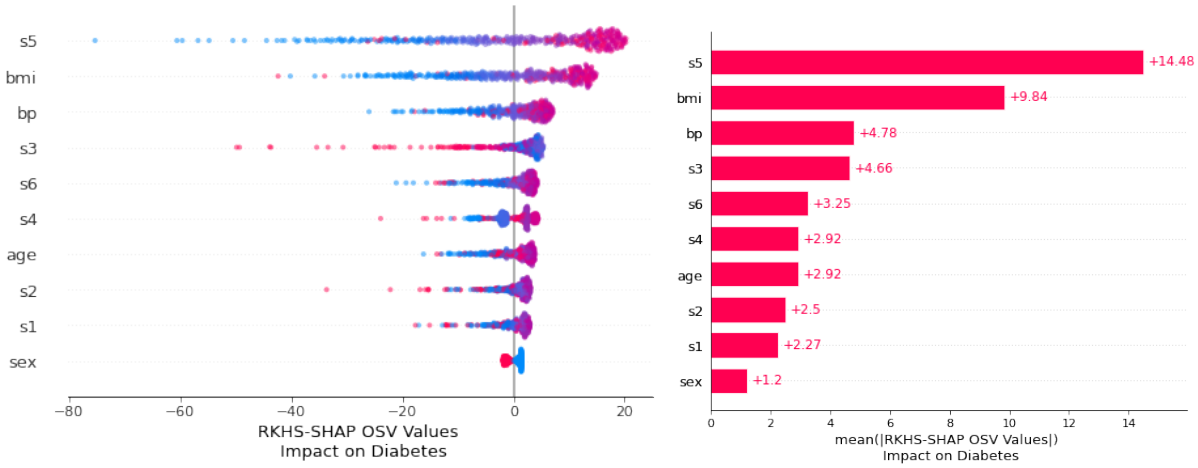


Figure C.5: Beeswarm and barplot of the RKHS-SHAP values on the Diabetes dataset

Diabetes for Pima Indian heritage Here we consider another dataset of diabetes study for Pima Indian heritage women aged 21 over. The data set is collected from here[3]. There are 768 samples with 8 features. The goal is to predict whether a patient has diabetes and fit a kernel logistic regression.

Figure C.6 demonstrated how RKHS-SHAP explains the kernel logistic regression. The top predictor, "Glucose", which measures the plasma glucose concentration 2 hours in an oral glucose tolerance test, aligns with the intuition that it should be strongly predictive to whether a person is diabetic. Also, high BMI leading to someone more likely to be diabetic is also reflected from RKHS-SHAP values.

[3]https://www.kaggle.com/datasets/mathchi/diabetes-data-set?resource=download

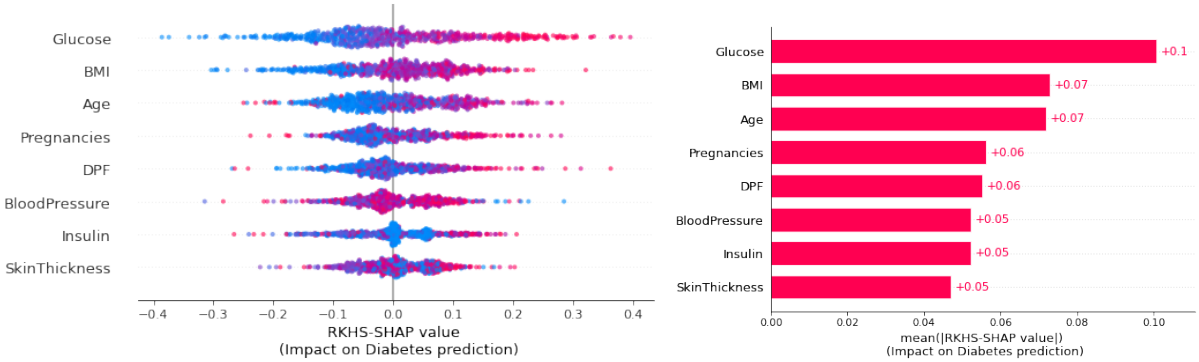


Figure C.6: Beeswarm and barplot of the RKHS-SHAP values on the Diabetes for pima indian heritage dataset

Breast Cancer Classification Next, we apply RKHS-SHAP to the breast cancer wisconsin dataset[4] to interpret the kernel logistic regression we have fitted to predict whether a patient might have breast cancer given their attributes. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the medical image. There are 569 data and 30 features. When running RKHS-SHAP, we did not use all 2^{30} coalitions but subsampled 10000 coalitions instead. Convergence analysis of such an approach is studied extensively by Covert and Lee [2021], where they empirically show that the algorithm will converge in $\mathcal{O}(n)$. Results are shown in Figure C.7. We can see that features such as "worst radius", "worst concave points", "worst perimeter" that describes the cell nuclei present in the breast mass, are most predictive to whether a patient has cancer or not.

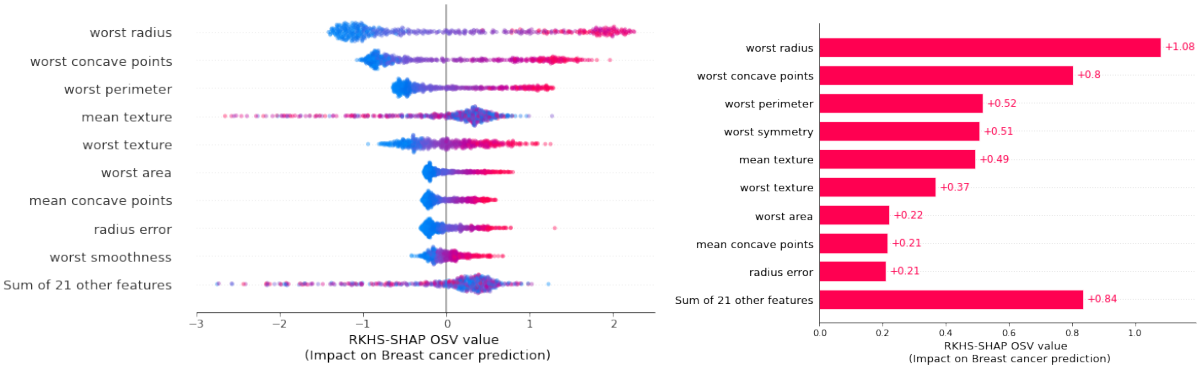


Figure C.7: Beeswarm and barplot for the breast cancer prediction problem

Census Income dataset In the following, we will explain the kernel logistic regression deployed to predict the probability of an individual making over $ 50K a year in annual income using the standard UCI adult income dataset. There are 48,842 number of instances and 14 attributes. We see that features such as relationship, education level, and capital gain are most predictive of whether a person earns more

[4] https://goo.gl/U2Uwz2

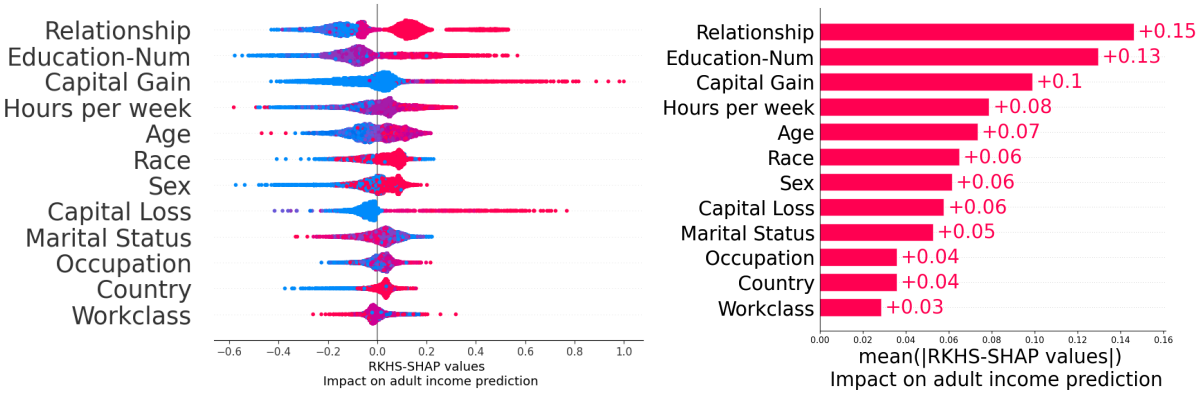


Figure C.8: Beeswarm and barplot for the census income data prediction problem

$ 50k a year. We see that as a person grows older, it is more likely to earn more, but the effect is not as impactful as, e.g. Education level or Capital gain.

League of Legends Win Prediction Finally, we use the Kaggle dataset League of Legends Ranked Matches which contains 1,800,000 players matches starting from 2014. We follow the preprocessing steps from of Legends Interpretability Demonstration [2022], and obtained 71 features at the end. We deploy RKHS-SHAP to explain the fitted Kernel Logistic regression model and obtain results in Figure C.9. We see that features such as "Deaths per min" and "Assists per min" are most influential to the match outcome. It follows the game mechanism, as a player is intuitively considered as "strong" if he doesn't die often in a round of the game. We would also like to point out we recover similar explanations from of Legends Interpretability Demonstration [2022], where they applied TreeSHAP to recover the explanations, see Fig. C.10. Interestingly, our kernel logistic regression seems to believe that "Gold earned per min" is less informative to the winning probability compared to "Deaths per min", which is different to the results obtained from the tree ensembles.

Figure C.9: Beeswarm plot for the League of Legends player winning prediction problem obtained using RKHS-SHAP.

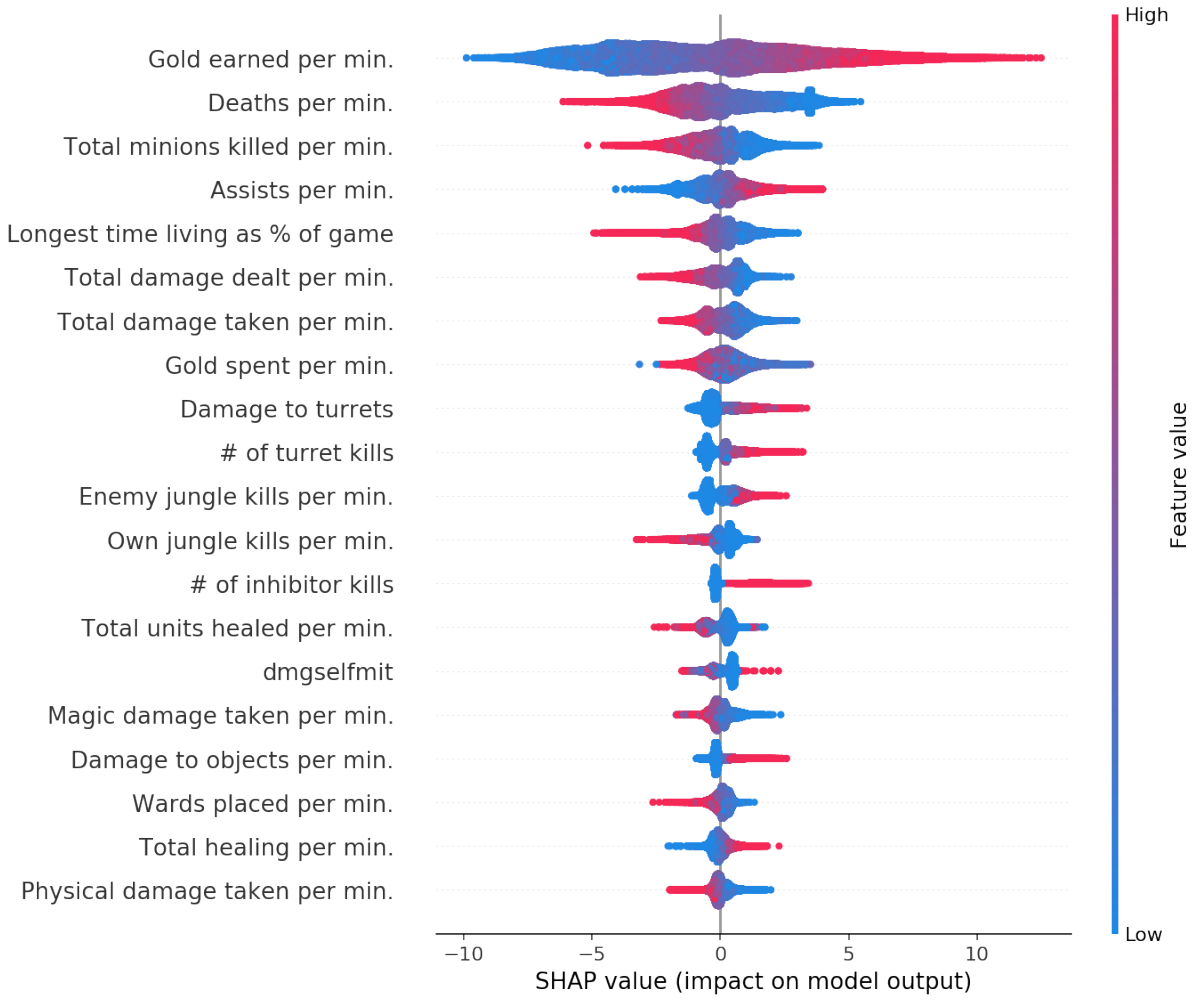


Figure C.10: Beeswarm plot for the League of Legends player winning prediction problem obtained using TreeSHAP. Similar insights are recovered compared to RKHS-SHAP. However, since the two methods are explaining different models – an RKHS function and a tree, it is not possible to tell which one gives more "correct" explanation.

D | Chapter 6 Appendix

D.1 Proof of ss-c_0-Universality

To prove Theorem 1, we present three propositions to establish the link between ss-c_0-universality and integrally strictly positive definite kernels, following closely the characterisation of c_0-universal kernels from Sriperumbudur et al. [2011, Proposition 4]. The domain $\mathcal{X}$ is assumed to be a locally compact Hausdorff space. We denote by $C_0(\mathcal{X} \times \mathcal{X})$ the space of functions $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which are continuous, bounded, and vanish at infinity. We let $C_{0,ss}(\mathcal{X} \times \mathcal{X}) \subset C_0(\mathcal{X} \times \mathcal{X})$ be the subspace of skew-symmetric functions. We denote by $M_{b,ss}(\mathcal{X} \times \mathcal{X})$ the set of finite signed Radon measures on $\mathcal{X} \times \mathcal{X}$ and similarly $M_{b,ss}(\mathcal{X} \times \mathcal{X}) \subset M_b(\mathcal{X} \times \mathcal{X})$ is the subset of skew-symmetric Radon measures.

Proposition D.1.1. *Let $\mathcal{X}$ be a locally compact Hausdorff space and $C'_{0,ss}(\mathcal{X} \times \mathcal{X})$ the topological dual of $C_{0,ss}(\mathcal{X} \times \mathcal{X})$. Then there is a bijective linear isometry $\nu \mapsto T_\nu$ from $M_{b,ss}(\mathcal{X} \times \mathcal{X})$ onto $C'_{0,ss}(\mathcal{X} \times \mathcal{X})$ given by the natural mapping, $T_\nu(f) = \int f d\nu$, $f \in C_{0,ss}(\mathcal{X} \times \mathcal{X})$. Thus we can identify $C'_{0,ss}(\mathcal{X} \times \mathcal{X}) = M_{b,ss}(\mathcal{X} \times \mathcal{X})$.*

Proof. By the Riesz representation theorem [Folland, 1999, Theorem 7.17], $C'_0(\mathcal{X} \times \mathcal{X}) = M_b(\mathcal{X} \times \mathcal{X})$ for corresponding spaces without enforcing skew-symmetry. Since $C_{0,ss}(\mathcal{X} \times \mathcal{X}) \subseteq C_0(\mathcal{X} \times \mathcal{X})$, for every linear functional $T_\nu \in C'_{0,ss}(\mathcal{X} \times \mathcal{X})$, there is a unique measure $\nu \in M_b(\mathcal{X} \times \mathcal{X})$. We will show that ν must be skew-symmetric.

Assume ν is not skew-symmetric, since $\nu \mapsto T_\nu$ is unique, we have,

$$T_\nu(f) = \int f(x, x') d\nu(x, x') = -\int f(x', x) d\nu^t(x', x) \tag{D.1}$$

where ν^t is the transpose of ν. Furthermore, we decompose $\nu = \nu^+ + \nu^-$ into a symmetric and skew-symmetric component with $\nu^+ = \frac{1}{2}(\nu + \nu^t)$ and $\nu^- = \frac{1}{2}(\nu - \nu^t)$. Thus, $T_\nu(f) = \int f d\nu = \int f d\nu^+ + \int f d\nu^-$, however,

$$\int f d\nu^+ = \frac{1}{2}\int f d\nu + \frac{1}{2}\int f d\nu^t = 0 \tag{D.2}$$

by uniqueness. Therefore ν is skew-symmetric and $C'_{0,ss}(\mathcal{X} \times \mathcal{X}) = M_{b,ss}(\mathcal{X} \times \mathcal{X})$. □

Proposition D.1.1 demonstrates the equivalence between the dual of $C_{0,ss}(\mathcal{X} \times \mathcal{X})$ and the measure space $M_{b,ss}(\mathcal{X} \times \mathcal{X})$. This fact is then used along with the Hahn-Banach theorem [Rudin, 1991, Theorem 3.5]

to prove a necessary and sufficient condition for k to be ss-c_0-universal.

Proposition D.1.2. *Suppose $\mathcal{X} \times \mathcal{X}$ is a locally compact Hausdorff space with kernel k bounded and $k(\cdot,(x,x')) \in C_{0,ss}(\mathcal{X}\times\mathcal{X}), \forall (x,x') \in \mathcal{X}\times\mathcal{X}$. Then k is ss-c_0-universal if and only if the embedding*

$$\nu \mapsto \int k\left(\cdot,(x,x')\right) d\nu(x,x') \tag{D.3}$$

is injective for all $\nu \in M_{b,ss}(\mathcal{X}\times\mathcal{X})$.

Proof. By definition, k is ss-c_0-universal if $\mathcal{H}_k$ is dense in $C_{0,ss}(\mathcal{X}\times\mathcal{X})$. This can be shown directly by applying the Hahn-Banach theorem [Rudin, 1991, Theorem 3.5], which states that $\mathcal{H}_k$ is dense in $C_{0,ss}(\mathcal{X}\times\mathcal{X})$ if and only if $\mathcal{H}_k^{\perp} := \{T \in C'_{0,ss}(\mathcal{X}\times\mathcal{X}) : \forall f \in \mathcal{H}_k, T(f) = 0\} = \{0\}$. However, $C'_{0,ss}(\mathcal{X}\times\mathcal{X}) = M_{b,ss}(\mathcal{X}\times\mathcal{X})$ by Proposition 1, therefore $\mathcal{H}_k^{\perp} = \{\nu \in M_{b,ss}(\mathcal{X}\times\mathcal{X}) : \forall f \in \mathcal{H}_k, \int f d\nu = 0\} = \{0\}$. A direct application of the Riesz representation theorem shows that $\mathcal{H}_k^{\perp} = \{\nu \in M_{b,ss}(\mathcal{X}\times\mathcal{X}) : \int k(\cdot,(x,x'))d\nu(x,x') = 0\}$ thus proving injectivity. □

Finally, we connect ss-c_0-universal kernels to integrally strictly pd kernels as follows:

Proposition D.1.3. *Let $\mathcal{X}$ be a locally compact Hausdorff metric space and k a continuous kernel on the joint space $\mathcal{X}\times\mathcal{X}$. Then, k is ss-c_0-universal if and only if $\mu_k : M_{b,ss}(\mathcal{X}\times\mathcal{X}) \to \mathcal{H}_k$ is a vector space monomorphism, that is,*

$$||\mu_k(\nu)||^2_{\mathcal{H}_k} = \int\int k((u,u'),(v,v'))d\nu((u,u'))d\nu((v,v')) > 0 \quad \forall \nu \in M_{b,ss}(\mathcal{X}\times\mathcal{X})\backslash\{0\}. \tag{D.4}$$

Proof. ($\Leftarrow$) Suppose k is not ss-c_0-universal. By Proposition 2, there exists $\nu \in M_{b,ss}(\mathcal{X}\times\mathcal{X})\backslash\{0\}$ such that $\int k(\cdot,(x,x'))d\nu(x,x') = 0$, which implies

$$\left|\left| \int k(\cdot,(x,x'))d\nu(x,x') \right|\right|^2_{\mathcal{H}_k} = \int\int k((u,u'),(v,v'))d\nu(u,u')d\nu(v,v') = 0$$

thus showing k is not integrally strictly pd. Therefore k has to be ss-c_0-universal.

($\Rightarrow$) Suppose there exists $\nu \in M_{b,ss}(\mathcal{X}\times\mathcal{X})\backslash\{0\}$ such that $\left|\left|\mu_k(\nu)\right|\right|_{\mathcal{H}_k} = 0$. This means,

$$\left|\left| \int k(\cdot,(x,x'))d\nu(x,x') \right|\right|^2_{\mathcal{H}_k} = 0 \Rightarrow \int k(\cdot,(x,x'))d\nu(x,x') = 0.$$

Therefore, the embedding is not injective, thus a contradiction by Proposition 2. Therefore, if k is ss-c_0-universal, then k satisfies (D.4). □

Now we can finish the proof for the main theorem using the above characterisations for ss-c_0-universal kernels.

Proof of main theorem. Pick any $\nu \in M_{b,ss}(\mathcal{X} \times \mathcal{X}) \backslash \{0\}$. Consider the corresponding kernel mean embedding of ν i.e $\mu_{K_E}(\nu)$, we have:

$$||\mu_{k_E}(\nu)||^2_{\mathcal{H}_{k_E}} = \int_{(u,u')} \int_{(v,v')} k_E((u,u'),(v,v')) \quad d\nu((u,u'))d\nu((v,v')) \tag{D.5}$$

$$= \int \int k(u,v)k(u',v')d\nu_{u,u'}d\nu_{v,v'} - \int \int k(u,v')k(u',v)d\nu_{u,u'}d\nu_{v,v'} \tag{D.6}$$

$$= \int \int k(u,v)k(u',v')d\nu_{u,u'}d\nu_{v,v'} + \int \int k(u,v')k(u',v)d\nu_{u',u}d\nu_{v,v'} \tag{D.7}$$

$$= 2 \int \int k(u,v)k(u',v')d\nu_{u,u'}d\nu_{v,v'} \tag{D.8}$$

$$= 2||\mu_{k\otimes k}(\nu)||^2_{\mathcal{H}_{k\otimes k}} \tag{D.9}$$

$$> 0. \tag{D.10}$$

We flip the sign in (D.7) because ν is skew-symmetric. In the last inequality we used the fact that if k is universal on $\mathcal{X}$, then the product kernel is also universal on the product space $\mathcal{X} \times \mathcal{X}$ [Szabó and Sriperumbudur, 2017] hence they are integrally strictly pd [Sriperumbudur et al., 2011, Proposition 4]. Therefore by Proposition 3, k_E is ss-c_0-universal. □

D.2 Extending the Generalised Preferential Kernel

The kernel we provided in the main paper in fact can be extended to tackle different preference learning situations.

Crowd Preferential Learning Given pairwise labels provided by a crowd, one can model the user-specific preference function $g : \mathcal{X} \times \mathcal{X} \times \mathcal{Z} \to \{0, 1\}$ by setting up a RKHS with the following kernel,

$$k^z_E\Big((u,u',z),(v,v',z')\Big) = \Big(k(u,v)k(u',v') - k(u,v')k(u',v)\Big)k_z(z,z') \tag{D.11}$$

where k, k_z are kernels defined on the item space $\mathcal{X}$ and user space $\mathcal{Z}$ respectively. Appropriate universality can be shown to hold for k^z_E as well, provided k, k_z are universal respectively. If the same set of items are

voted on by each user, one can further use tensor algebra to speed up computations since $K_E^z = K_E \otimes K_z$.

Distributional Preferential Learning We now consider situations where we would like to model preferences between groups of items while we only have access to individual level features. Football matches and e-sports tournaments are common examples of this setup. Mathematically this corresponds to the following setup: we have a dataset $\left\{\{x_i^a\}_{i=1}^{N_a}, \{x_i^b\}_{i=1}^{N_b}, y_{a,b}\right\}$ where each $B_a = \{x_i^a\}_{i=1}^{N_a}$ is assumed to be a sample coming from some distribution P_a, and $y_{a,b}$ is the preference outcome when B_a is compared to B_b. Since we only have preferences on the distributional level, we call this *Distributional Preferential Learning* and consider the following generative model

$$p(y_{a,b} = 1 | \{x_i^a\}_{i=1}^{N_a}, \{x_i^b\}_{i=1}^{N_b}) = \sigma(g(\{x_i^a\}_{i=1}^{N_a}, \{x_i^b\}_{i=1}^{N_b})). \tag{D.12}$$

Once again we consider g as a skew symmetric function corresponding to the RKHS $\mathcal{H}_{k_E^{(B)}}$ with the following kernel,

$$k_E^{(B)}\Big((B_a, B_b), (B_c, B_d)\Big) = k^{(B)}(B_a, B_c)k^{(B)}(B_b, B_d) - k^{(B)}(B_a, B_d)k^{(B)}(B_b, B_c) \tag{D.13}$$

where $k^{(B)}(B_a, B_c) = k^{(B)}(\{x_i^a\}_{i=1}^{N_a}, \{x_i^c\}_{i=1}^{N_c}) = \frac{1}{N_a N_c}\sum_{i=1}^{N_a}\sum_{j=1}^{N_c} k(x_i, x_j)$ is a linear kernel between the empirical kernel mean embeddings, which are commonly used as feature representations for probability distributions.

D.3 Feature Maps of Preferential Kernels

We here briefly describe the differences in terms of feature space representations between the preferential and generalised preferential kernels. These are very similar to the differences between the sum-kernel and the product-kernel when combining kernels on individual domains to construct a kernel on the product domain.

The feature space of the preferential kernel

$$\kappa\left(\left(u, u'\right), \left(v, v'\right)\right) = k\left(u, v\right) + k\left(u', v'\right) - k\left(u, v'\right) - k\left(u', v\right)$$

is given by the direct sum $\mathcal{H}_k \oplus \mathcal{H}_k$ and the feature map is

$$\varphi : \left(u, u'\right) \mapsto \frac{1}{\sqrt{2}}\left(k\left(\cdot, u\right) \oplus k\left(\cdot, u'\right) - k\left(\cdot, u'\right) \oplus k\left(\cdot, u\right)\right).$$

Indeed,

$$\begin{aligned}
&\left\langle \varphi\left(u,u'\right),\varphi\left(v,v'\right)\right\rangle \\
&\quad=\frac{1}{2}\left\langle k\left(\cdot,u\right)\oplus k\left(\cdot,u'\right)-k\left(\cdot,u'\right)\oplus k\left(\cdot,u\right),k\left(\cdot,v\right)\oplus k\left(\cdot,v'\right)-k\left(\cdot,v'\right)\oplus k\left(\cdot,v\right)\right\rangle \\
&\quad=\frac{1}{2}\Big\{\left\langle k\left(\cdot,u\right)\oplus k\left(\cdot,u'\right),k\left(\cdot,v\right)\oplus k\left(\cdot,v'\right)\right\rangle+\left\langle k\left(\cdot,u'\right)\oplus k\left(\cdot,u\right),k\left(\cdot,v'\right)\oplus k\left(\cdot,v\right)\right\rangle \\
&\qquad-\left\langle k\left(\cdot,u'\right)\oplus k\left(\cdot,u\right),k\left(\cdot,v\right)\oplus k\left(\cdot,v'\right)\right\rangle-\left\langle k\left(\cdot,u\right)\oplus k\left(\cdot,u'\right),k\left(\cdot,v'\right)\oplus k\left(\cdot,v\right)\right\rangle\Big\} \\
&\quad=\frac{1}{2}\Big\{\left\langle k\left(\cdot,u\right),k\left(\cdot,v\right)\right\rangle+\left\langle k\left(\cdot,u'\right),k\left(\cdot,v'\right)\right\rangle+\left\langle k\left(\cdot,u'\right),k\left(\cdot,v'\right)\right\rangle+\left\langle k\left(\cdot,u\right),k\left(\cdot,v\right)\right\rangle \\
&\qquad-\left\langle k\left(\cdot,u'\right),k\left(\cdot,v\right)\right\rangle-\left\langle k\left(\cdot,u\right),k\left(\cdot,v'\right)\right\rangle-\left\langle k\left(\cdot,u\right),k\left(\cdot,v'\right)\right\rangle-\left\langle k\left(\cdot,u'\right),k\left(\cdot,v\right)\right\rangle\Big\} \\
&\qquad\quad=k\left(u,v\right)+k\left(u',v'\right)-k\left(u,v'\right)-k\left(u',v\right).
\end{aligned}$$

On the other hand, the feature space of the generalised preferential kernel

$$\kappa\left(\left(u,u'\right),\left(v,v'\right)\right)=k\left(u,v\right)k\left(u',v'\right)-k\left(u,v'\right)k\left(u',v\right)$$

is given by the tensor product $\mathcal{H}_k\otimes\mathcal{H}_k$ and the feature map is

$$\varphi:\left(u,u'\right)\mapsto\frac{1}{\sqrt{2}}\left(k\left(\cdot,u\right)\otimes k\left(\cdot,u'\right)-k\left(\cdot,u'\right)\otimes k\left(\cdot,u\right)\right).$$

Indeed,

$$\begin{aligned}
&\left\langle \varphi\left(u,u'\right),\varphi\left(v,v'\right)\right\rangle \\
&\quad=\frac{1}{2}\left\langle k\left(\cdot,u\right)\otimes k\left(\cdot,u'\right)-k\left(\cdot,u'\right)\otimes k\left(\cdot,u\right),k\left(\cdot,v\right)\otimes k\left(\cdot,v'\right)-k\left(\cdot,v'\right)\otimes k\left(\cdot,v\right)\right\rangle \\
&\quad=\frac{1}{2}\Big\{\left\langle k\left(\cdot,u\right)\otimes k\left(\cdot,u'\right),k\left(\cdot,v\right)\otimes k\left(\cdot,v'\right)\right\rangle+\left\langle k\left(\cdot,u'\right)\otimes k\left(\cdot,u\right),k\left(\cdot,v'\right)\otimes k\left(\cdot,v\right)\right\rangle \\
&\qquad-\left\langle k\left(\cdot,u'\right)\otimes k\left(\cdot,u\right),k\left(\cdot,v\right)\otimes k\left(\cdot,v'\right)\right\rangle-\left\langle k\left(\cdot,u\right)\otimes k\left(\cdot,u'\right),k\left(\cdot,v'\right)\otimes k\left(\cdot,v\right)\right\rangle\Big\} \\
&\quad=\frac{1}{2}\Big\{\left\langle k\left(\cdot,u\right),k\left(\cdot,v\right)\right\rangle\left\langle k\left(\cdot,u'\right),k\left(\cdot,v'\right)\right\rangle+\left\langle k\left(\cdot,u'\right),k\left(\cdot,v'\right)\right\rangle\left\langle k\left(\cdot,u\right),k\left(\cdot,v\right)\right\rangle \\
&\qquad-\left\langle k\left(\cdot,u'\right),k\left(\cdot,v\right)\right\rangle\left\langle k\left(\cdot,u\right),k\left(\cdot,v'\right)\right\rangle-\left\langle k\left(\cdot,u\right),k\left(\cdot,v'\right)\right\rangle\left\langle k\left(\cdot,u'\right),k\left(\cdot,v\right)\right\rangle\Big\} \\
&\qquad\quad=k\left(u,v\right)k\left(u',v'\right)-k\left(u,v'\right)k\left(u',v\right).
\end{aligned}$$

These results of course also apply to finite-dimensional feature spaces where the direct sum operation corresponds to concatenation of the individual feature vectors of two players and the tensor product corresponds to an outer product between feature vectors. Namely, if ϕ is an explicit finite-dimensional

feature map of kernel k, with explicit feature space $\mathbb{R}^m$, then feature map φ of kernel κ can be constructed as

$$\varphi : (u, u') \mapsto \frac{1}{\sqrt{2}} \left(\begin{bmatrix} \phi(u) \\ \phi(u') \end{bmatrix} - \begin{bmatrix} \phi(u') \\ \phi(u) \end{bmatrix} \right) \in \mathbb{R}^{2m}$$

in the case of the preferential kernel, and as

$$\varphi : (u, u') \mapsto \frac{1}{\sqrt{2}} \left(\phi(u) \phi(u')^\top - \phi(u') \phi(u)^\top \right) \in \mathbb{R}^{m^2}$$

in the case of the generalised preferential kernel. These feature maps can be used to construct large scale approximations via explicit feature representations for the preferential or generalised preferential kernel, using, for example, random Fourier features [Rahimi and Recht, 2008] for the base kernel k.

Linear base kernels. Let us consider preferential models with a linear base kernel $k(u, v) = u^\top v$. Here we will assume a logistic model for concreteness, like in the main text, but this of course readily extends to other forms of observation models. Likelihood in the preferential kernel case boils down to

$$\begin{aligned} p(y_{ij} = 1 | (x_i, x_j)) &= \sigma \left(\beta^\top \left(\begin{bmatrix} x_i \\ x_j \end{bmatrix} - \begin{bmatrix} x_j \\ x_i \end{bmatrix} \right) \right) \\ &= \sigma \left(\beta_1^\top (x_i - x_j) + \beta_2^\top (x_j - x_i) \right) \\ &= \sigma \left((\beta_1 - \beta_2)^\top (x_i - x_j) \right), \end{aligned}$$

where constant $1/\sqrt{2}$ is folded into the coefficient vector and we denoted the two halves of entries in β by β_1 and β_2 respectively. Hence we recover a simple logistic model on the differences between feature vectors with a p-dimensional vector of coefficients $w := \beta_1 - \beta_2$.

In contrast, as we will see, the generalised preferential kernel model starting with a linear base kernel parametrises likelihood using a general skew-symmetric bilinear form of the individual feature vectors. Collating coefficients into a $p \times p$ matrix B, we obtain likelihood given by

$$\begin{aligned} p(y_{ij} = 1 | (x_i, x_j)) &= \sigma \left(\mathrm{tr} \left[B \left(x_i x_j^\top - x_j x_i^\top \right)^\top \right] \right) \\ &= \sigma \left(x_i^\top B x_j - x_j^\top B x_i \right). \end{aligned}$$

We note that B can decomposed into symmetric and skew-symmetric part with $B^+ = \frac{1}{2}(B + B^\top)$ and

$B^- = \frac{1}{2}\left(B - B^\top\right)$. Hence the likelihood becomes $\sigma\left(x_i^\top W x_j\right)$ where $W = 2B^- = B - B^\top$ is a skew-symmetric matrix.

Enforcing skew-symmetry via the coefficients. One can consider a more immediate way to construct a linear skew-symmetric model in the case of explicit features, by enforcing that the coefficients take particular form. These turn out to be equivalent to using feature maps described above. For example, we could have a model on the concatenation

$$\begin{aligned} p\left(y_{ij} = 1 | \left(x_i, x_j\right)\right) &= \sigma\left(\begin{bmatrix} w \\ -w \end{bmatrix}^\top \begin{bmatrix} \phi\left(x_i\right) \\ \phi\left(x_j\right) \end{bmatrix}\right) \\ &= \sigma\left(w^\top\left(\phi\left(x_i\right) - \phi\left(x_j\right)\right)\right). \end{aligned}$$

For a general bilinear model, we write

$$p\left(y_{ij} = 1 | \left(x_i, x_j\right)\right) = \sigma\left(\phi\left(x_i\right)^\top W \phi\left(x_j\right)\right),$$

and require that the matrix W is skew-symmetric, i.e. that $a^\top W b = -b^\top W a$.

Hence, we conclude that preferential and generalised preferential feature maps correspond to over-parametrised versions of such models, where we parametrise functions using β rather than $\beta_1 - \beta_2$ and using B rather than $B - B^\top$. However, while constraints such as these may be enforceable in finite-dimensional feature spaces, it is not clear whether it is possible to enforce skew-symmetry directly on the dual coefficients in the infinite-dimensional case.

E | Appendix for Chapter 7

E.1 Computation and Implementation Details

We propose several optimizations in the PREF-SHAP procedure. We consider fast sampling of coalitions S in Algorithm 2 batched conjugate gradient descent in Algorithm 3 described below.

Fast coalitions We first propose an optimized sampling scheme for finding coalitions S in Algorithm 2.

Algorithm 2 Sampling unique n, d-dimensional coalitions in $\mathcal{O}(d)$ time

Input: Number of coalitions n, number of features d.

$I = \text{SampleWithoutReplacement}(n, 0, 2^d)$ ▷ Sample n unique integers between 0 and 2^d
def base2(i): ▷ Convert integer to base-2 representation
    $S = [0, \dots, 0] \in \mathbb{R}^d$, $r = i$ ▷ Initialize d-dimensional 0 vector and the rest term r
    while $r > 0$ **do**
        $i = \lfloor \log_2(r) \rfloor$ ▷ Find which index of S to set to 1
        $S[i] = 1$ ▷ Update S
        $r = r - 2^i$ ▷ Update rest term
    end while
    return S
return $\{S_1, \dots S_n\} = \text{parallel\_apply}(I, \text{base2})$ ▷ Each integer can be independently converted

In contrast to the implementation in Lundberg and Lee [2017] which samples the weights from $p(Z)$, our method is embarrassingly parallel, which allows for an additional $\mathcal{O}(n)$ reduction. A naive algorithm that compares each sample S_i has complexity $\mathcal{O}(n^2 d^2)$ and cannot be parallelized.

Stabilizing the Shapley value estimation We remove the features which have 0 variance in the data we are explaining, similar to the implementation in SHAP. To ensure we get numerically stable Shapley Values, we calculate the inverse using Cholesky decomposition, as we found the regular inverse function provided inconsistent results.

To calculate CMEs effectively, we use *preconditioned batched* conjugate gradient descent over coalitions detailed in Algorithm 3.

Algorithm 3 Batched conjugate gradient descent

Input: Preconditioner $P = \mathbf{K}_{\mathbf{x}_D}^{-1}$, batch $\mathbf{X} = [\mathbf{K}_{\mathbf{x}_{S_i}} \dots \mathbf{K}_{\mathbf{x}_{S_n}}]$, $\mathbf{B} = [\mathbf{K}_{\mathbf{x}_{S_i}, x_{S_i}} \dots \mathbf{K}_{\mathbf{x}_{S_n}, x_{S_n}}]$, `max_its`, tolerance ε

Set $\mathbf{R} = \mathbf{B}$, $\mathbf{Z} = \text{BatchMM}(P, \mathbf{B})$, $\mathbf{p} = \mathbf{Z}$, $\mathbf{a} = \mathbf{0}$

Set $\mathbf{R}_Z = [(\mathbf{R}_i \circ \mathbf{Z}_i)_{++} \dots (\mathbf{R}_n \circ \mathbf{Z}_n)_{++}]$ ▷ Element wise product and sum

for `max_its` **do**

$\mathbf{L} = \text{BatchMM}(\mathbf{X}, \mathbf{B})$

$\boldsymbol{\alpha} = \mathbf{L} \circ \frac{1}{[(\mathbf{L}_i \circ \mathbf{p}_i)_{++} \dots (\mathbf{L}_n \circ \mathbf{p}_n)_{++}]}$

$\mathbf{a} = \mathbf{a} + \boldsymbol{\alpha} \circ \mathbf{p}$

$\mathbf{r} = \mathbf{r} - \boldsymbol{\alpha} \circ \mathbf{L}$

if $\text{Mean}([(\mathbf{r}_i \circ \mathbf{r}_i)_{++} \dots (\mathbf{r}_n \circ \mathbf{r}_n)_{++}]) < \varepsilon$ **then return** $\mathbf{a}$

end if

$\mathbf{z} = \text{BatchMM}(P, \mathbf{r})$

$\mathbf{R}_Z^{\text{new}} = [(\mathbf{R}_i \circ \mathbf{Z}_i)_{++} \dots (\mathbf{R}_n \circ \mathbf{Z}_n)_{++}]$

$\mathbf{p} = \mathbf{z} + [(\mathbf{R}_i^{\text{new}} \circ \frac{1}{\mathbf{R}_i})_{++} \dots (\mathbf{R}_n^{\text{new}} \circ \frac{1}{\mathbf{R}_n})_{++}] \circ \mathbf{p}$

$\mathbf{R}_Z = \mathbf{R}_Z^{\text{new}}$

end for

return $\mathbf{a}$

We have run all our jobs on one Nvidia V100 GPU.

E.2 Additional Experimental Results

Naive Concatenation We demonstrate the pathologies of the naive concatenation approach mentioned in Sec. 3 with our synthetic experiment. Recall that naive-concatenation approach here corresponds to first concatenating $\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$'s features together and applying SHAP to the learned function g directly, in order to obtain $2d$ Shapley values, instead of the original d, since each feature has been duplicated. This approach ignores that the items $\mathbf{x}^{(\ell)}$ and $\mathbf{x}^{(r)}$ in fact consist of the same features. Therefore, when we use the usual value function from SHAP (corresponding to the impact an individual feature has on the model when it is turned "off" by integration), we would be turning "off" the feature from the left item, while keeping "on" the feature from the right item, obtaining a difficult to interpret attribution score. This is highly problematic, as we might be inferring vastly different contributions of the same feature purely because of the item ordering when concatenating them. We note that the item ordering in all our experiments is arbitrary and carries no additional information about the match.

We can see from Fig. E.1 that when we explain the preference model applied to the synthetic experiment, we see that, for example, $x^{AC}(l)$ from $\mathbf{x}^{(\ell)}$ and $x^{AC}(r)$ from $\mathbf{x}^{(r)}$ have in fact very different average Shapley values. Even attempting to average each pair of corresponding features does not give a meaningful feature contribution ordering (x^{AB} and x^{AC} are scored higher on average than x^{BC} and $x^{[0]}$).

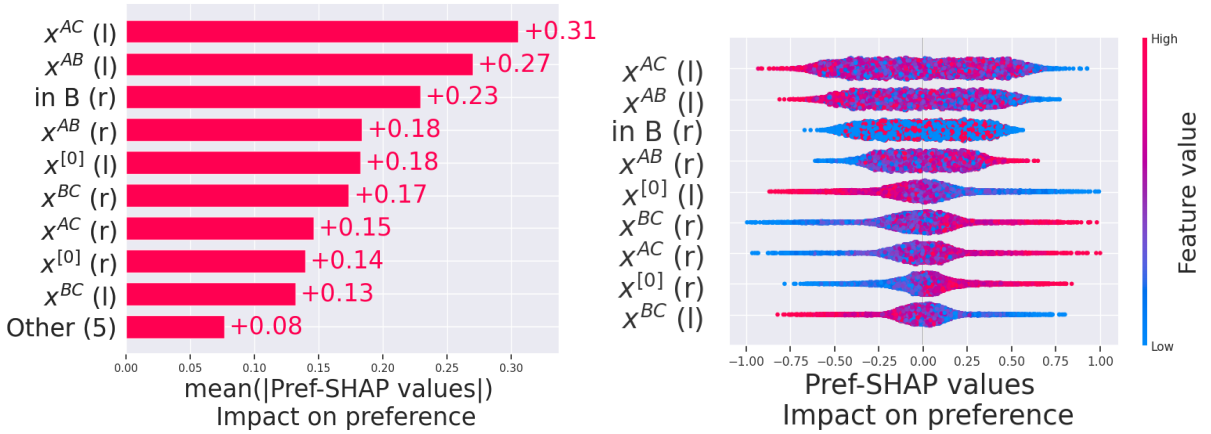


Figure E.1: Explaining a naive concatenation model

Additional synthetic data We consider an additional synthetic experiment where we generate data directly from a GPM model and one where we construct synthetic dueling data. When simulating data, we first generate player covariates as $\mathbf{x}_i \in \mathbb{R}^d \sim \mathcal{N}(0, 0.1\mathbf{I}_d)$ for each player i. When generating from the GPM model, we would set 2 covariates as important, by only keeping the 2 first entries of $\mathbf{x}_i$ and fixing the rest to be constant (equal to 0). We build a GPM model for g out of these covariates and generate match outcomes.

We consider $d = 10$, where only the two first features are set to be important in predicting the outcome.

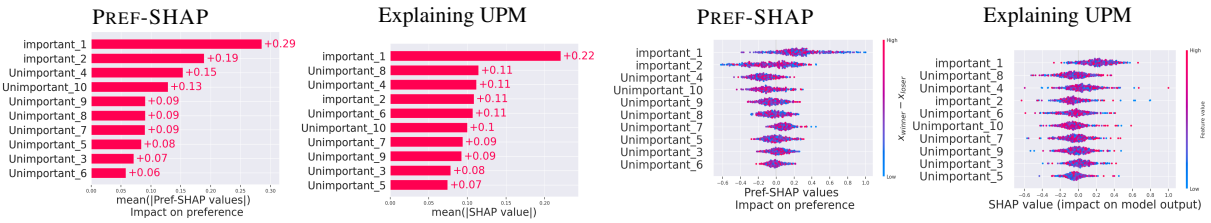


Figure E.2: Bar and Beehive plots for Simulation A. PREF-SHAP recovers the correct features (1,2), while explaining UPM does not.

Chameleon data We further provide explanations of the Chameleon dataset on several different folds in $\text{cham}_appandcham_appu_pm$.

Additional local explanations We additionally provide local explanations on the synthetic dataset in $\text{synt}_beeplot_appendix_1andsynt_beeplot_appendix_2$.

Website dataset *The Website* dataset considers anonymized visitors on a fashion retail website, where we are given what garment each visitor viewed and what each visitor clicked in a session. A user may have more than one session. In this setup, we interpret a browsing session for a visitor as multiple matches

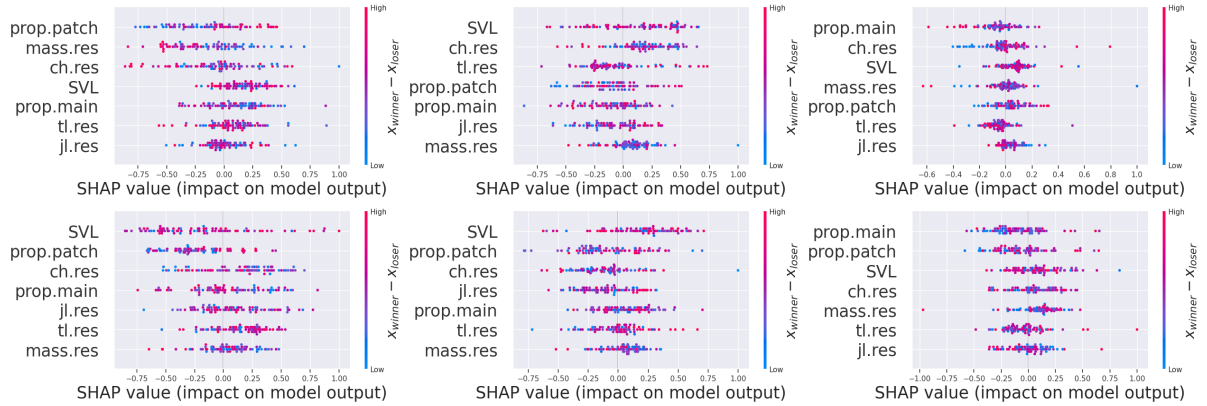


Figure E.4: UPM explanations on 6 different folds of Chameleon. UPM is unable to find a consistent pattern for the impact of *jaw length* (jl.res) on the outcome.

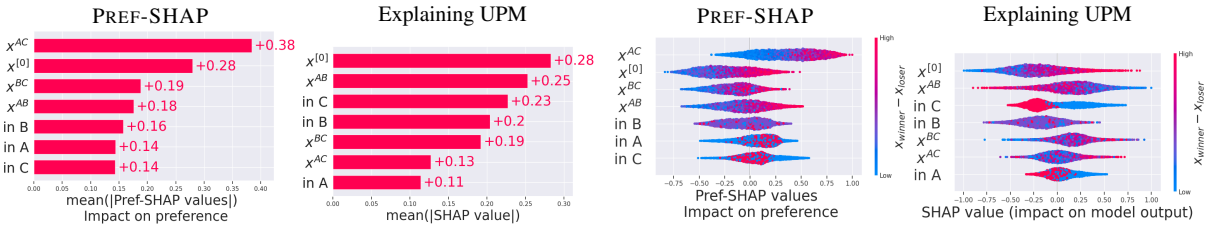


Figure E.5: Explaining matches between clusters 0 and 2 on the synthetic dataset.

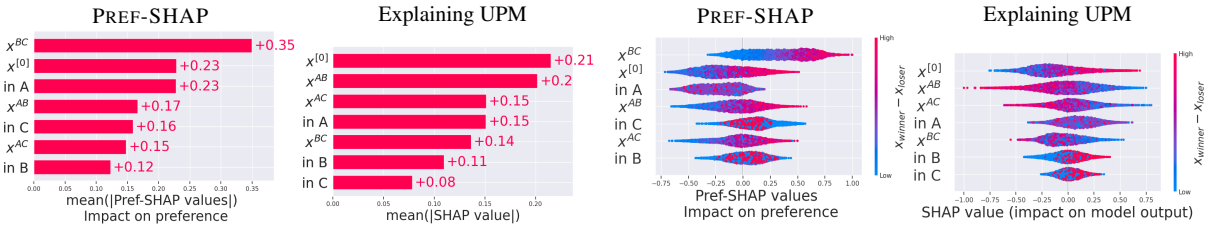


Figure E.6: Explaining matches between clusters 1 and 2 on the synthetic dataset.

between items, such that the winning item (clicked) competes against all losing items (only viewed). If several items are winners, they do not play against each other. Each item has several descriptive statistics such as *colour*, *garment type*, *assortment characteristic* etc. There are some limited descriptive statistics of the visitors, such as *year of birth* and *gender code* (i.e. Male/Female/Unspecified/Unknown).

Explaining Website For the website dataset, we explain product and user preferences in figure. E.7. We generally found that, for the period considered, cosmetic products and the "Jersey Basic category" drove clicks.

Table E.2: Dataset summary

Dataset	N_{Matches}	N_{Players}	N_{Context}	$D_{\text{continuous}}$	D_{binary}	$D^{\text{Context}}_{\text{continuous}}$	$D^{\text{Context}}_{\text{binary}}$
Website	85144	20626	129117 (users)	0	93	1	4

Figure E.7: Barplots and beeplots for the website dataset, products on the left and user variables on the right.

Table E.1: GPM vs UPM. Mean and standard deviations of performance averaged over 5 runs.

	Synthetic		Chameleon		Pokémon		Tennis		Website	
	GPM	UPM	GPM	UPM	GPM	UPM	C-GPM	UPM	C-GPM	UPM
Test AUC	$0.98_{\pm 0.00}$	$0.71_{\pm 0.01}$	$0.92_{\pm 0.07}$	$0.80_{\pm 0.07}$	$0.86_{\pm 0.00}$	$0.82_{\pm 0.00}$	$0.58_{\pm 0.02}$	$0.52_{\pm 0.02}$	$0.66_{\pm 0.01}$	$0.65_{\pm 0.01}$
SpecR	0.09		0.24		0.20		$0.13_{\pm 0.07}$		$0.53_{\pm 0.10}$	

E.3 Proofs

Proposition 3.1 (Preferential value functional for items)**.** *Let k be a product kernel on $\mathcal{X}$, i.e. $k(\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}) = \prod_{j=1}^{d} k^{(j)}(x^{(j)}, x'^{(j)})$. Assume $k^{(j)}$ are bounded for all j, then the Riesz representation of the functional $\nu^{(p)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}$ exists and takes the form:*

$$\nu^{(p)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S} = \frac{1}{\sqrt{2}}\Big(\mathcal{K}(\mathbf{x}^{(\ell)}, S) \otimes \mathcal{K}(\mathbf{x}^{(r)}, S) - \mathcal{K}(\mathbf{x}^{(r)}, S) \otimes \mathcal{K}(\mathbf{x}^{(\ell)}, S)\Big)$$

where $\mathcal{K}(\mathbf{x}, S) = k_S(\cdot, \mathbf{x}_S) \otimes \mu_{X_{S^c}|X_S=\mathbf{x}_S}$ and $k_S(\cdot, \mathbf{x}_S) = \bigotimes_{j\in S} k^{(j)}(\cdot, x^{(j)})$ is the sub-product kernel defined analogously as X_S.

Proof. From Chau et al. [2022], we know the generalised preferential kernel has the following feature map:

$$k_E((\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot) = \frac{1}{\sqrt{2}}\big(k(\cdot, \mathbf{x}^{(\ell)}) \otimes k(\cdot, \mathbf{x}^{(r)}) - k(\cdot, \mathbf{x}^{(r)}) \otimes k(\cdot, \mathbf{x}^{(\ell)})\big) \tag{E.1}$$

where $\otimes$ are the usual tensor product. Recall we defined the preferential value function for items as,

$$\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(g) = \mathbb{E}[g(X^{(\ell)}, X^{(r)}) \mid X^{(\ell)}_S = \mathbf{x}^{(\ell)}_S, X^{(r)}_S = \mathbf{x}^{(r)}_S] \tag{E.2}$$

as $\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}$ is a bounded linear functional on g where $g \in \mathcal{H}_{k_E}$ is bounded, Riesz representation

theorem Paulsen and Raghupathi [2016] tells us there exists a Riesz representation of the functional in $\mathcal{H}_{k_E}$, which for notation simplicity, we will denote it as $\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}$ as well. This corresponds to,

$$\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(g) = \mathbb{E}[g(X^{(\ell)}, X^{(r)}) \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)}, X_S^{(r)} = \mathbf{x}_S^{(r)}] \tag{E.3}$$

$$= \langle g, \mathbb{E}[k_E\left((X^{(\ell)}, X^{(r)}), \cdot\right) \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)}, X_S^{(r)} = \mathbf{x}_S^{(r)}]\rangle_{\mathcal{H}_{k_E}} \tag{E.4}$$

$$= \langle g, \nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}\rangle_{\mathcal{H}_{k_E}} \tag{E.5}$$

now we expand the expectation of the feature map as,

$$\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S} = \mathbb{E}\left[\frac{1}{\sqrt{2}}\big(k(\cdot, X^{(\ell)}) \otimes k(\cdot, X^{(r)}) - k(\cdot, X^{(r)}) \otimes k(\cdot, X^{(\ell)}) \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)}, X_S^{(r)} = \mathbf{x}_S^{(r)}\right] \tag{E.6}$$

However, we note that

$$\mathbb{E}[k(\cdot, X^{(\ell)}) \otimes k(\cdot, X^{(r)}) \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)}, X_S^{(r)} = \mathbf{x}_S^{(r)}] = \mathbb{E}[k(\cdot, X) \mid X_S = \mathbf{x}_S^{(\ell)}] \otimes \mathbb{E}[k(\cdot, X) \mid X_S = \mathbf{x}_S^{(r)}],$$

because $X^{(\ell)}$ and $X^{(r)}$ are identical copies of X and we take the reference distribution as $p(X^{(\ell)}, X^{(r)} \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)}, X_S^{(r)} = \mathbf{x}_S^{(r)}) = p(X^{(\ell)} \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)})p(X^{(r)} \mid X_S^{(r)} = \mathbf{x}_S^{(r)})$. Focusing on the duplicating component, we have,

$$\mathbb{E}[k(\cdot, X) \mid X_S = \mathbf{x}^{(\ell)}] = \mathbb{E}[k_S(\cdot, X_S) \otimes k_{S^c}(\cdot, X_{S^c}) \mid X_S = \mathbf{x}_S^{(\ell)}] \tag{E.7}$$

$$= k_S(\cdot, \mathbf{x}_S^{(\ell)}) \otimes \mathbb{E}[k_{S^c}(\cdot, X_{S^c}) \mid X_S = \mathbf{x}_S^{(\ell)}] \tag{E.8}$$

$$= k_S(\cdot, \mathbf{x}_S^{(\ell)}) \otimes \mu_{X_{S^c}|X_S=\mathbf{x}_S^{(\ell)}} \tag{E.9}$$

$$=: \mathcal{K}(\mathbf{x}^{(\ell)}, S) \tag{E.10}$$

therefore by symmetry, we can arrange the terms in Eq E.6 and conclude the proposition,

$$\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S} = \frac{1}{\sqrt{2}}\left(\mathcal{K}(\mathbf{x}^{(\ell)}, S) \otimes \mathcal{K}(\mathbf{x}^{(r)}, S) - \mathcal{K}(\mathbf{x}^{(r)}, S) \otimes \mathcal{K}(\mathbf{x}^{(\ell)}, S)\right) \tag{E.11}$$

□

To estimate the preferential value functional, we simply replace the conditional mean embeddings with the empirical versions, i.e. $\hat{\mathcal{K}}(\mathbf{x}, S) = k_S(\cdot, \mathbf{x}_S) \otimes \hat{\mu}_{X_{S^c}|X_S=\mathbf{x}_S}$, where $\hat{\mu}_{X_{S^c}|X_S=\mathbf{x}_S} = \mathbf{K}_{\mathbf{x}_S,\mathbf{x}_S}(\mathbf{K}_{\mathbf{x}_S,\mathbf{x}_S} + n\lambda I)^{-1}\mathbf{\Phi}_{X_{S^c}}^\top$ is the standard conditional mean embedding estimator ($\mathbf{\Phi}_{X_{S^c}}$ is the feature map matrix of

rv X_{S^c}).

Now we proceed to estimate the preferential value function given a function g from the RKHS,

Proposition 3.2 (Non-parametric Estimation)**.** *Given $\hat{g} = \sum_{j=1}^{m} \alpha_j k_E((\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot)$, datasets $\mathbf{X}^{(\ell)}, \mathbf{X}^{(r)}$, test items $\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$, the preferential value function at test items $\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$ for coalition S and preference function $\hat{g}$ can be estimated as*

$$\hat{\nu}^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(\hat{g}) = \boldsymbol{\alpha}^\top \left(\Gamma(\mathbf{X}_S^{(\ell)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{X}_S^{(r)}, \mathbf{x}_S^{(r)}) - \Gamma(\mathbf{X}_S^{(\ell)}, \mathbf{x}_S^{(r)}) \odot \Gamma(\mathbf{X}_S^{(r)}, \mathbf{x}_S^{(\ell)}) \right),$$

where $\Gamma(\mathbf{X}_S^{(\ell)}, \mathbf{x}_S^{(\ell)}) = \mathbf{K}_{\mathbf{X}_S^{(\ell)}, \mathbf{x}_S^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}_{S^c}^{(\ell)}, \mathbf{X}_{S^c}} \mathbf{K}^{-1}_{\mathbf{X}_S, \lambda} \mathbf{K}_{\mathbf{X}_S, \mathbf{x}_S^{(\ell)}}$, $\mathbf{K}_{\mathbf{X}_S, \lambda} = \mathbf{K}_{\mathbf{X}_S, \mathbf{X}_S} + n\lambda I$, $\boldsymbol{\alpha} = \{\alpha_j\}_{j=1}^m$ and $\lambda > 0$ is a regularisation parameter.

Proof. Given $\hat{g}$, the preferential value function evaluated at $\hat{g}$ can be written as,

$$\hat{\nu}^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(\hat{g}) = \langle \hat{g}, \hat{\nu}^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S} \rangle_{\mathcal{H}_{k_E}} \tag{E.12}$$

$$= \langle \sum_{j=1}^{m} \alpha_j k_E((\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot), \frac{1}{\sqrt{2}} \left(\hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) - \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \right) \rangle_{\mathcal{H}_{k_E}} \tag{E.13}$$

$$= \frac{1}{\sqrt{2}} \left\langle \sum_{j=1}^{m} \alpha_j k_E((\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle \tag{E.14}$$

$$- \frac{1}{\sqrt{2}} \left\langle \sum_{j=1}^{m} \alpha_j k_E((\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot), \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \right\rangle \tag{E.15}$$

Now we focus on the first component, and rewrite:

$$\frac{1}{\sqrt{2}} \left\langle \sum_{j=1}^{m} \alpha_j k_E((\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle = \sum_{j=1}^{m} A_j^{(1)} \tag{E.16}$$

and we continue to expand the terms,

$$A_j^{(1)} := \frac{1}{\sqrt{2}} \left\langle \alpha_j k_E((\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle \tag{E.17}$$

$$= \frac{1}{\sqrt{2}} \left\langle \frac{\alpha_j}{\sqrt{2}} \big(k(\cdot, \mathbf{x}_j^{(\ell)}) \otimes k(\cdot, \mathbf{x}_j^{(r)}) - k(\cdot, \mathbf{x}_j^{(r)}) \otimes k(\cdot, \mathbf{x}_j^{(\ell)}) \big), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle \tag{E.18}$$

$$= \frac{\alpha_j}{2} \left(\left\langle k(\cdot, \mathbf{x}_j^{(\ell)}) \otimes k(\cdot, \mathbf{x}_j^{(r)}), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle - \left\langle k(\cdot, \mathbf{x}_j^{(r)}) \otimes k(\cdot, \mathbf{x}_j^{(\ell)}), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle \right) \tag{E.19}$$

$$= \frac{\alpha_j}{2} \left(A_j^{(1,\ell)} - A_j^{(1,r)} \right) \tag{E.20}$$

We then note that

$$A_j^{(1,\ell)} := \left\langle k(\cdot, \mathbf{x}_j^{(\ell)})) \otimes k(\cdot, \mathbf{x}_j^{(r)}), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle \tag{E.21}$$

$$= \left\langle k(\cdot, \mathbf{x}_j^{(\ell)}), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \right\rangle \left\langle k(\cdot, \mathbf{x}_j^{(r)}), \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle \tag{E.22}$$

$$= k_S(\mathbf{x}_j^{(\ell)}, \mathbf{x}^{(\ell)}) \mathbf{K}_{\mathbf{x}_{j\ S}^{(\ell)}, \mathbf{X}_S} (\mathbf{K}_{\mathbf{X}_S, \mathbf{X}_S} + n\lambda I)^{-1} \mathbf{K}_{\mathbf{X}_{S^c}, \mathbf{x}_{S^c}^{(\ell)}} \tag{E.23}$$

$$\times\, k_S(\mathbf{x}_j^{(r)}, \mathbf{x}^{(r)}) \mathbf{K}_{\mathbf{x}_{j\ S}^{(r)}, \mathbf{X}_S} (\mathbf{K}_{\mathbf{X}_S, \mathbf{X}_S} + n\lambda I)^{-1} \mathbf{K}_{\mathbf{X}_{S^c}, \mathbf{x}_{S^c}^{(r)}} \tag{E.24}$$

$$= \Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(r)}) \tag{E.25}$$

To go from the second equation to the third equation in this paragraph, realise $k(\cdot, \mathbf{x}^{(\ell)}) = k_S(\cdot, \mathbf{x}_S^{(\ell)}) \otimes k_{S^c}(\cdot, \mathbf{x}_{S^c}^{(\ell)})$ by product kernel assumption. In this case, we can rewrite $A_j^{(1)}$ as,

$$A_j^{(1)} = \frac{\alpha_j}{2} \left(\Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(r)}) - \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(r)}) \right) \tag{E.26}$$

Analogously, define $\sum A_j^{(2)}$ as the second component after the subtraction sign, by symmetry, we know

$$A_j^{(2)} = \frac{\alpha_j}{2} \left(\Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(r)}) \odot \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(\ell)}) - \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(r)}) \odot \Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(\ell)}) \right) \tag{E.27}$$

by subtracting $A_j^{(1)}$ and $A_j^{(1)}$, we get the following:

$$\hat{\nu}^{(p_I)}_{\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}, S}(\hat{g}) = \sum_{j=1}^{m} \alpha_j \left(\Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(r)}) - \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(r)}) \right) \tag{E.28}$$

writing it in compact form, we arrive to our result,

$$= \boldsymbol{\alpha}^\top \left(\Gamma(\mathbf{X}_S^{(\ell)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{X}_S^{(r)}, \mathbf{x}_S^{(r)}) - \Gamma(\mathbf{X}_S^{(\ell)}, \mathbf{x}_S^{(r)}) \odot \Gamma(\mathbf{X}_S^{(r)}, \mathbf{x}_S^{(\ell)}) \right) \tag{E.29}$$

□

Proposition 3.3 (Preferential value function for contexts)**.** *Given a preference function $g_U \in \mathcal{H}_{k_E^U}$, denote $\Omega' = \{1, ..., d'\}$, then the utility of context features $S' \subseteq \Omega'$ on $\{\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\}$ is measured by $\nu^{(p_U)}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S'}(g_U) = \mathbb{E}[g_U\left(\{\mathbf{u}_{S'}, U_{S'^c}\}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) \mid U_{S'} = \mathbf{u}_{S'}]$ where the expectation is taken over the observational distribution of U. Now, given a test triplet $(\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)})$, if $\hat{g}_U = \sum_{j=1}^m \alpha_j k_E^U\left((\mathbf{u}_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot\right)$, the non-parametric estimator is:*

$$\hat{\nu}^{(p_U)}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S'}(\hat{g}_U) = \boldsymbol{\alpha}^\top \left(\left(\mathbf{K}_{\mathbf{U}_{S'},\mathbf{u}_{S'}} \odot \mathbf{K}_{\mathbf{U}_{S'^c},\mathbf{U}_{S'^c}}\left(\mathbf{K}_{\mathbf{U}_{S'},\mathbf{U}_{S'}} + m\lambda' I\right)^{-1} \mathbf{K}_{\mathbf{U}_{S'},\mathbf{u}_{S'}}\right) \odot \Xi_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)}}\right)$$

where $\Xi_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)}} = \left(\mathbf{K}_{\mathbf{X}^{(\ell)},\mathbf{x}^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}^{(r)},\mathbf{x}^{(r)}} - \mathbf{K}_{\mathbf{X}^{(r)},\mathbf{x}^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}^{(\ell)},\mathbf{x}^{(r)}}\right)$.

Proof. Recall the feature map of the kernel k_E^U takes the following form,

$$k_E^U\left((\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot\right) = k_u(\mathbf{u}, \cdot) \otimes k_E((\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot) \tag{E.30}$$

Therefore we can express the preferential value function for context as,

$$\begin{aligned}
\nu^{(p_U)}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S'}(g_U) &= \mathbb{E}\left[g_U\left(\{\mathbf{u}_{S'}, U_{S^c}\}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) \mid U_S = \mathbf{u}_{S'}\right] && \text{(E.31)}\\
&= \left\langle g_U, \mathbb{E}\left[k_E^U\left((\{\mathbf{u}_{S'}, U_{S'^c}\}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot\right) \mid U_{S'} = \mathbf{u}_{S'}\right]\right\rangle && \text{(E.32)}\\
&= \left\langle g_U, \mathbb{E}\left[k_u(\{\mathbf{u}_{S'}, U_{S^c}\} \mid U_{S'} = \mathbf{u}_{S'}] \otimes k_E((\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot)\right\rangle && \text{(E.33)}\\
&= \left\langle g_U, k_{u_{S'}}(\mathbf{u}_{S'}) \otimes \mu_{U_{S'^c}|U_{S'}=\mathbf{u}_{S'}} \otimes k_E((\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot)\right\rangle && \text{(E.34)}
\end{aligned}$$

The remaining steps are analogous to [Chau et al., 2021b, Prop.2]. To obtain the empirical estimation, we first replace the conditional mean embedding $\mu_{U_{S'^c}|U_{S'}=\mathbf{u}_{S'}}$ with its empirical estimate and replace g_U with $\hat{g}_U = \sum_{j=1}^m \alpha_j k_E^U\left((\mathbf{u}_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot\right)$. Now the empirical estimator has the following form,

$$\hat{\nu}^{(p_U)}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S'}(\hat{g}_U) = \left\langle \sum_{j=1}^m \alpha_j k_E^U\left((\mathbf{u}_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot\right), k_{u_{S'}}(\mathbf{u}_{S'}) \otimes \hat{\mu}_{U_{S'^c}|U_{S'}=\mathbf{u}_{S'}} \otimes k_E((\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot)\right\rangle \tag{E.35}$$

$$= \sum_{j=1}^m \alpha_j \left\langle k_u(\mathbf{u}_j, \cdot) \otimes k_E((\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot), k_{u_{S'}}(\mathbf{u}_{S'}) \otimes \hat{\mu}_{U_{S'^c}|U_{S'}=\mathbf{u}_{S'}} \otimes k_E((\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot)\right\rangle \tag{E.36}$$

$$= \sum_{j=1}^m \alpha_j \left\langle k_u(\mathbf{u}_j, \cdot), k_{u_S}(\mathbf{u}_S) \otimes \hat{\mu}_{U_{S'^c}|U_S=\mathbf{u}_S}\right\rangle k_E\left((\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), (\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)})\right) \tag{E.37}$$

Now write everything in terms of matrices,

$$= \boldsymbol{\alpha}^\top \left(\left(\mathbf{K}_{\mathbf{U}_{S'}, \mathbf{u}_{S'}} \odot \mathbf{K}_{\mathbf{U}_{S'^c}, \mathbf{U}_{S'^c}} \left(\mathbf{K}_{\mathbf{U}_{S'}, \mathbf{U}_{S'}} + m\lambda' I \right)^{-1} \mathbf{K}_{\mathbf{U}_{S'}, \mathbf{u}_{S'}} \right) \odot \Xi_{\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}} \right) \tag{E.38}$$

where $\Xi_{\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}} = \left(\mathbf{K}_{\mathbf{X}^{(\ell)}, \mathbf{x}^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}^{(r)}, \mathbf{x}^{(r)}} - \mathbf{K}_{\mathbf{X}^{(r)}, \mathbf{x}^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}^{(\ell)}, \mathbf{x}^{(r)}} \right)$. □