RECENT ADVANCES ON THE REDUCTION AND ANALYSIS OF BIG AND HIGH-DIMENSIONAL DATA

A Dissertation
Presented to
The Academic Faculty

By

Simon Tsz Fung Mak

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology

May 2018

RECENT ADVANCES ON THE REDUCTION AND ANALYSIS OF BIG AND HIGH-DIMENSIONAL DATA

Approved by:

Dr. Roshan J. Vengazhiyil, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. C. F. Jeff Wu, Co-Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Yao Xie
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. George Lan
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Fred J. Hickernell
Department of Applied Mathematics
Illinois Institute of Technology

Date Approved: March 12, 2018

Where the world ceases to be the scene of our personal hopes and wishes, where we face it as free beings admiring, asking and observing, there we enter the realm of Art and Science.

Albert Einstein

Fantasy, abandoned by reason, produces impossible monsters; united with it, she is the mother of the arts and the origin of marvels.

Francisco Goya

Beauty in things exists in the mind which contemplates them.

David Hume

To my parents, for their love, encouragement and support.

ACKNOWLEDGEMENTS

I would like to first express my sincere gratitude to my advisor, Professor Roshan J. Vengazhiyil, for his continuous support and guidance throughout my Ph.D. study. His patience, motivation and passion for research helped me greatly in developing the necessary skills and knowledge for a career in academic research.

I also would like to express my deepest appreciation to my co-advisor, Professor C. F. Jeff Wu, for his immense support and guidance during my studies. His mentorship was instrumental in not only broadening my perspectives on research, but also developing me as a mature and critical thinker.

I am also grateful to Professor Vigor Yang for his mentorship and the opportunity to collaborate with his lab on several projects. I would like to thank Professor Yao Xie for her guidance and support on a summer project, Professor Fred Hickernell for his mentorship in the SAMSI Program on Quasi-Monte Carlo, and Professor George Lan for his helpful advice on my thesis. I am also thankful to Professor Derek Bingham for his advice and mentorship; without his help, I would not have had this wonderful opportunity to study at Georgia Tech.

I would like to thank all my past and present lab mates: Dr. Rui Tuo, Dr. Heng Su, Dr. Yuan Wang, Dr. Dianpeng Wang, Dr. Li Gu, Chih-Li Sung, Wenjia Wang, Yuanshuo Zhao, Li-Hsiang Lin and Zhehui Chen, for enlightening conversations and research discussions.

Last but definitely not least, my heartfelt thanks go to my parents. Without your love, encouragement, and most of all patience, I would not be able to achieve this important milestone in my life. This thesis is dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

In an era with remarkable advancements in computer engineering, computational algorithms, and mathematical modeling, data scientists are inevitably faced with the challenge of working with big and high-dimensional data. For many problems, data reduction is a necessary first step; such reduction allows for storage and portability of big data, and enables the computation of expensive downstream quantities. The next step then involves the analysis of big data – the use of such data for modeling, inference, and prediction. This thesis presents new methods for big data reduction and analysis, with a focus on solving real-world problems in statistics, machine learning and engineering.

Chapter 1 of my thesis introduces a data reduction method for compacting large datasets (or in the infinite sense, distributions) into a smaller, representative point set called support points (SPs). SPs can be viewed as optimal sampling points for distribution representation, integration, and functional approximation. One advantage of SPs is that it provides an efficient and parallelizable reduction of big data via difference-of-convex programming. Chapter 2 then presents a modification of SPs, called projected support points (PSPs), for compacting high-dimensional datasets into representative points. The key innovation for PSPs is the use of a sparsity-inducing kernel, which allows for reduction of low-dimensional properties in high-dimensional data. We then demonstrate the effectiveness of SPs and PSPs for (a) compacting posterior samples in Bayesian computation, (b) uncertainty propagation, and (c) kernel learning with big data.

Chapter 3 proposes a novel variable selection method for analyzing big data, using new basis functions called conditional main effects (CMEs). CMEs capture the conditional effect of a variable at a fixed level of another variable, and represent interpretable phenomena in many engineering and social science fields. We present an algorithm, called cmenet, which employs the new principles of CME coupling and CME reduction to guide variable selection. Compared to standard interaction analysis, cmenet yields more parsimonious

models and improved predictive performance, which we demonstrate using simulations and a gene association study on fly wing shape.

Chapter 4 introduces a surrogate model for efficient prediction and uncertainty quantification of turbulent flows in swirl injectors, devices commonly used in engineering systems. Here, high-fidelity simulations require weeks of computation time, and a new method is needed to efficiently survey the desired design space. We propose a new Gaussian process surrogate model, which incorporates known physical flow properties as simplifying assumptions. This allows for efficient model training with massive simulation data (100Gb in storage), which then enables quick flow predictions at new design settings in around an hour of computation time.

Chapter 5 considers construction algorithms for a type of experimental design called minimax designs. Minimax designs reduce a continuous design space to a set of design points, by minimizing the maximum distance from this space to its nearest point. We propose a new clustering-based construction of minimax designs on convex design regions, and demonstrate its effectiveness in simulations and a real-world sensor allocation problem. We then introduce a novel design called a minimax projection design, which yields improved minimax performance on projections of the design space.

Finally, Chapter 6 presents a new active sampling method for noisy matrix completion. This method implicitly makes use of uncertainty quantification (UQ) at unobserved matrix entries to guide active sampling. Using a singular matrix-variate Gaussian model, we first reveal novel insights on the role of compressive sensing and coding design on the sampling and UQ for noisy matrix completion. With these insights, we propose an efficient posterior sampler for quantifying subspace uncertainty, and an information-theoretic algorithm which uses this subspace learning to guide sampling. The effectiveness of this integrated method is then demonstrated in simulations and two collaborative filtering examples.

CHAPTER 1

SUPPORT POINTS – A NEW WAY TO COMPACT DISTRIBUTIONS

1.1 Introduction

This chapter explores a new method for compacting a continuous probability distribution F into a set of representative points (rep-points) for F, which we call *support points*. Support points have many important applications in a wide array of fields, because these point sets provide an improved representation of F compared to a random sample. One such application is to the "small-data" problem of uncertainty propagation, where the use of support points as simulation inputs can allow engineers to quantify the propagation of input uncertainty onto system output at minimum cost. Another important application is to "big-data" problems encountered in Bayesian computation, specifically as a tool for compacting large posterior sample chains from Markov chain Monte Carlo (MCMC) methods [1]. In this chapter, we demonstrate the theoretical and practical effectiveness of support points for the general problem of integration, and illustrate its usefulness for the two applications above.

We first outline two classes of existing methods for rep-points. The first class consists of the so-called *mse-rep-points* (see, e.g., Chapter 4 of [2]), which minimize the expected distance from a random point drawn from F to its closest rep-point. Also known as principal points [3], mse-rep-points have been employed in a variety of statistical and engineering applications, including quantizer design [4, 5] and optimal stratified sampling [6, 7]. In practice, these rep-points can be generated by first performing k-means clustering [8] on a large batch sample from F, then taking the converged cluster centers as rep-points. One

The paper based on this chapter will appear in *Annals of Statistics*.

weakness of mse-rep-points, however, is that they do not necessarily converge to F (see, e.g., [9, 10]). The second class of rep-points, called *energy rep-points*, aims to find a point set which minimizes some measure of statistical potential. Included here are the minimum-energy designs in [11] and the minimum Riesz energy points in [12]. While the above point sets converge in distribution to F, its convergence rate is quite slow, both theoretically and in practice [12]. Moreover, the construction of such point sets can be computationally expensive in high dimensions.

The key idea behind support points is that it optimizes a specific potential measure called the *energy distance*, which makes such point sets a type of energy rep-point. First introduced in [13], the energy distance was proposed as a computationally efficient way to evaluate goodness-of-fit (GOF), compared to the classical Kolmogorov-Smirnov (K-S) statistic [14], which is difficult to evaluate in high-dimensions. Similar to the existing energy rep-points above, we show in this chapter that support points indeed converge in distribution to F. In addition, we demonstrate the improved error rate of support points over Monte Carlo for integrating a large class of functions. The minimization of this distance can also be formulated as a difference-of-convex (d.c.) program, which allows for efficient generation of support points.

Indeed, the *reverse-engineering* of a GOF test forms the basis for state-of-the-art integration techniques called Quasi-Monte Carlo (QMC) methods (see [15] and [16] for a modern overview). To see this, first let g be a differentiable integrand, and let $\{\mathbf{x}_i\}_{i=1}^n$ be the point set (with empirical distribution, or e.d.f., F_n) used to approximate the desired integral $\int g(\mathbf{x})\ dF(\mathbf{x})$ with the sample average $\int g(\mathbf{x})\ dF_n(\mathbf{x})$. For simplicity, assume for now that $F = U[0,1]^p$ is the uniform distribution on the p-dimensional hypercube $[0,1]^p$, the typical setting for QMC. The Koksma-Hlawka inequality (see, e.g., [17]) provides the following upper bound on the integration error I:

$$I(g; F, F_n) \equiv \left| \int g(\mathbf{x})\ d[F - F_n](\mathbf{x}) \right| \leq V_q(g) D_r(F, F_n), \quad 1/q + 1/r = 1, \tag{1.1}$$

where $V_q(g) = \|\partial^p g/\partial \mathbf{x}\|_{L_q}$, and $D_r(F, F_n)$ is the L_r-*discrepancy*:

$$D_r(F, F_n) = \left(\int |F_n(\mathbf{x}) - F(\mathbf{x})|^r \ d\mathbf{x} \right)^{1/r}. \tag{1.2}$$

The discrepancy $D_r(F, F_n)$ measures how close the e.d.f. F_n is to F, with a smaller value suggesting a better fit. Setting $r = \infty$, the L_∞-discrepancy (or simply *discrepancy*) becomes the classical K-S statistic for testing GOF. In other words, a point set with good fit to F also provides reduced integration errors for a large class of integrands. A more general discussion of this connection in terms of kernel discrepancies can be found in [18].

For a general distribution F, the optimization of $D_r(F, F_n)$ can be a difficult problem. In the uniform setting $F = U[0, 1]^p$, there has been some work on directly minimizing the discrepancy $D_\infty(F, F_n)$, including the cdf-rep-points in [2] and the uniform designs in [19]. Such methods, however, are quite computationally expensive, and are applicable only for small point sets on $U[0, 1]^p$ (see [20]). Because of this computational burden, modern QMC methods typically use number-theoretic techniques to generate point sets which achieve an *asymptotically* quick decay rate for discrepancy. These include the randomly-shifted lattice rules [21] using the component-by-component implementation of [22] (see also [23]), and the randomly scrambled Sobol' sequences due to [24] and [25]. While most QMC methods consider integration on the uniform hypercube $U[0, 1]^p$, there are several ways to map point sets on $U[0, 1]^p$ to non-uniform F. One such map is the inverse Rosenblatt transformation [26]; however, it can be computed in closed-form only for a small class of distributions. When the density of F is known up to a proportional constant, the Markov chain Quasi-Monte Carlo (MCQMC) approach [27] can also be used to generate QMC points on F.

Viewed in this light, the energy distance can be seen as a kernel discrepancy [29] for non-uniform distributions, with the specific kernel choice being the negative Euclidean norm. However, in contrast with the typical number-theoretic construction of QMC point sets, support points are instead generated by optimizing the underlying d.c. formulation for the energy distance. This explicit optimization can have both advantages and disadvantages. On one hand, support points can be viewed as *optimal* sampling points of F (in

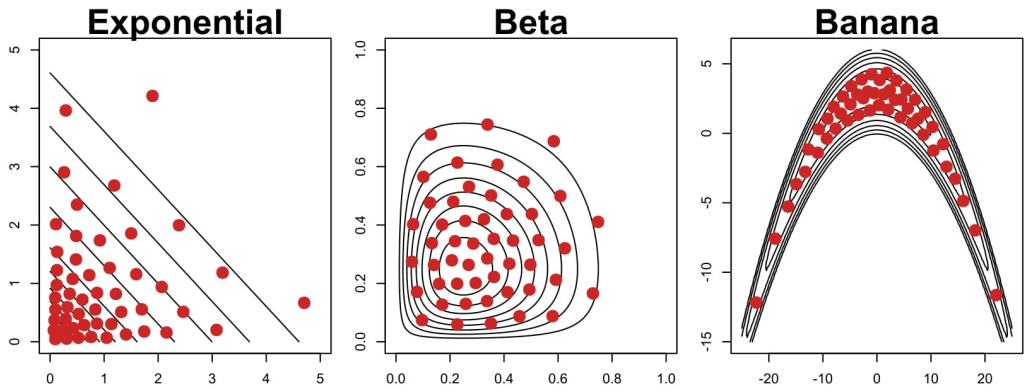

Figure 1.1: $n = 50$ support points for 2-d i.i.d. $Exp(1)$, $Beta(2, 4)$ and the banana-shaped distribution in [28]. Lines represent density contours.

the sense of minimum energy) for any desired sample size n. This optimality is evident in the three examples of support points plotted in Figure 1.1 – the points are concentrated in regions with high densities, but is sufficiently spread out to maximize the representativeness of each point. Such a "space-filling" property can allow for improved integration performance over existing QMC techniques, which we demonstrate in Section 1.4. On the other hand, the computational work for optimization can grow quickly when the desired sample size or dimension increases. To this end, we propose two algorithms which exploit the appealing d.c. formulation to efficiently generate point sets as large as 10,000 points in dimensions as large as 500.

This chapter is organized as follows. Section 1.2 proves several important theoretical properties of support points. Section 1.3 proposes two algorithms for efficiently generating support points. Section 1.4 outlines several simulations comparing the integration performance of support points with MC and an existing QMC method. Section 1.5 gives two important applications of support points in uncertainty propagation and Bayesian computation. Section 1.6 concludes with directions for future research.

1.2 Support points

1.2.1 Definition

Let us first define the energy distance between two distributions F and G:

Definition 1 (Energy distance; Def. 1 of [30]). *Let F and G be two distribution functions (d.f.s) on $\varnothing \neq \mathcal{X} \subseteq \mathbb{R}^p$ with finite means, and let $\mathbf{X}, \mathbf{X}' \overset{i.i.d.}{\sim} G$ and $\mathbf{Y}, \mathbf{Y}' \overset{i.i.d.}{\sim} F$. The* energy distance *between F and G is defined as:*

$$E(F, G) \equiv 2\mathbb{E}\|\mathbf{X} - \mathbf{Y}\|_2 - \mathbb{E}\|\mathbf{X} - \mathbf{X}'\|_2 - \mathbb{E}\|\mathbf{Y} - \mathbf{Y}'\|_2. \tag{1.3}$$

When $G = F_n$ is the e.d.f. for $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathcal{X}$, this energy distance becomes:

$$E(F, F_n) = \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}\|\mathbf{x}_i - \mathbf{Y}\|_2 - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{x}_i - \mathbf{x}_j\|_2 - \mathbb{E}\|\mathbf{Y} - \mathbf{Y}'\|_2. \tag{1.4}$$

For brevity, F is assumed to be a continuous d.f. on $\varnothing \neq \mathcal{X} \subseteq \mathbb{R}^p$ with finite mean for the remainder of the chapter.

The energy distance $E(F, F_n)$ was originally proposed in [13] as an efficient GOF test for high-dimensional data. In this light, support points are defined as the point set with best GOF under $E(F, F_n)$:

Definition 2 (Support points). *Let $\mathbf{Y} \sim F$. For a fixed point set size $n \in \mathbb{N}$, the* support points *of F are defined as:*

$$\{\boldsymbol{\xi}_i\}_{i=1}^n \in \underset{\mathbf{x}_1,\cdots,\mathbf{x}_n}{\text{Argmin}}\ E(F, F_n) = \underset{\mathbf{x}_1,\cdots,\mathbf{x}_n}{\text{Argmin}} \left\{ \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}\|\mathbf{x}_i - \mathbf{Y}\|_2 - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{x}_i - \mathbf{x}_j\|_2 \right\}. \quad \text{(O)}$$

The minimization of $E(F, F_n)$ is justified by the following metric property:

Theorem 1 (Energy distance, Prop. 2 of [30]). *$E(F, G) \geq 0$, with equality holding if and only if F=G.*

This theorem shows that the energy between two distributions is always non-negative, and equals zero if and only if these distributions are the same. In this sense, $E(F, G)$ can be viewed as a metric on the space of distribution functions. Support points, being the point set which minimizes such a metric, can then be interpreted as optimal sampling points which best represent F.

The choice of the energy distance $E(F, F_n)$ as an optimization objective is similar to its appeal in GOF testing. As mentioned in the Introduction, $E(F, F_n)$ was originally proposed as an efficient alternative to classical K-S statistic. However, not only is $E(F, F_n)$ easy-to-evaluate, it also has a desirable formulation as a d.c. program. We present in Section 1.3 two algorithms which exploits this structure to efficiently generate support points.

In the univariate setting of $p = 1$, an interesting equivalence can be established between support points and optimal L_2-discrepancy points:

Proposition 1 (Optimal L_2-discrepancy)**.** *For a univariate d.f. F, the support points of F are equal to the point set with minimal L_2-discrepancy.*

Unfortunately, such an equivalence fails to hold for $p > 1$, since the L_2-discrepancy is not rotation-invariant. Support points and optimal L_2-discrepancy points can therefore behave quite differently in the multivariate setting.

1.2.2 Theoretical properties

While the notion of reverse engineering the energy distance is intuitively appealing, some theory is needed to demonstrate why the resulting points are appropriate for (a) representing the desired distribution F, and (b) integrating under F. To this end, we provide three theorems: the first proves the distributional convergence of support points to F, the second establishes a Koksma-Hlawka-like bound connecting integration error with $E(F, F_n)$, and the last provides an existence result for the resulting error convergence rate. The proofs of these results rely on the important property that, for generalized functions, the Fourier transform of the Euclidean norm $\|\cdot\|_2$ is proportional to the same norm raised to some power

(see pg. 173-174 in [31]). We refer to various forms of this *duality* property throughout the proofs.

Convergence in distribution

We first address the distributional convergence of support points to the desired distribution F:

Theorem 2 (Distributional convergence)**.** *Let* $\mathbf{X} \sim F$ *and* $\mathbf{X}_n \sim F_n$*, where* F_n *is the e.d.f. of the support points in* (O)*. Then* $\mathbf{X}_n \xrightarrow{d} \mathbf{X}$*.*

In words, this theorem shows that support points are indeed representative of the desired distribution F when the number of points n grows large. From this, the *consistency* of support points can be established:

Corollary 1 (Consistency)**.** *Let* $\mathbf{X} \sim F$ *and* $\mathbf{X}_n \sim F_n$*, with* F_n *as in Theorem 2. (a) If* $g : \mathcal{X} \to \mathbb{R}$ *is continuous, then* $g(\mathbf{X}_n) \xrightarrow{d} g(\mathbf{X})$*. (b) If* g *is continuous and bounded, then* $\lim_{n\to\infty} \mathbb{E}[g(\mathbf{X}_n)] = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} g(\boldsymbol{\xi}_i) = \mathbb{E}[g(\mathbf{X})]$.

The purpose of this corollary is two-fold: it demonstrates the consistency of support points for integration, and justifies the use of these point sets for a variety of other applications. Specifically, part (a) shows that support points are appropriate for performing uncertainty propagation in stochastic simulations, an application further explored in Section 1.4.2. Part (b) shows that any continuous and bounded integrand g can be consistently estimated using support points, i.e., its sample average converges to the desired integral.

A Koksma-Hlawka-like bound

Next, we present a theorem which upper bounds the squared integration error $I^2(g; F, F_n)$ by a term proportional to $E(F, F_n)$ for a large class of integrands. Such a result provides some justification on why the energy distance may be a *good* criterion for integration. Here, we first provide a brief review of conditionally positive definite (c.p.d.) kernels, its native

spaces, and their corresponding reproducing kernels, three ingredients which will be used for proving the desired theorem.

Consider the following definition of a *conditionally positive definite kernel*:

Definition 3 (c.p.d. kernel; Def. 8.1 of [32])**.** *A continuous function* $\Phi : \mathbb{R}^p \to \mathbb{R}$ *is a* c.p.d. kernel *of order* m *if, for all pairwise distinct* $\mathbf{x}_1, \cdots, \mathbf{x}_N \in \mathbb{R}^p$ *and all* $\boldsymbol{\zeta} \in \mathbb{R}^N \setminus \{0\}$ *satisfying* $\sum_{j=1}^N \zeta_j p(\mathbf{x}_j) = 0$ *for all polynomials of degree less than* m*, the quadratic form* $\sum_{j=1}^N \sum_{k=1}^N \zeta_j \zeta_k \Phi(\mathbf{x}_j - \mathbf{x}_k)$ *is positive.*

Similar to the theory of positive definite kernels (see, e.g., Section 10.1 and 10.2 of [32]), one can use a c.p.d. kernel Φ to construct a reproducing kernel Hilbert space (RKHS) along with its reproducing kernel. This is achieved using the so-called *native space* of Φ:

Definition 4 (Native space; Def. 10.16 of [32])**.** *Let* $\Phi : \mathbb{R}^p \to \mathbb{R}$ *be a c.p.d. kernel of order* $m \geq 1$*, and let* $\mathcal{P} = \pi_{m-1}(\mathbb{R}^p)$ *be the space of polynomials with degree less than* m*. Define the linear space:*

$$\mathcal{F}_\Phi(\mathbb{R}^p) = \left\{ f(\cdot) = \sum_{j=1}^N \zeta_j \Phi(\mathbf{x}_j - \cdot) \ : \begin{array}{l} N \in \mathbb{N};\ \boldsymbol{\zeta} \in \mathbb{R}^N;\ \mathbf{x}_1, \cdots, \mathbf{x}_N \in \mathbb{R}^p, \\ \sum_{j=1}^N \zeta_j p(\mathbf{x}_j) = 0 \text{ for all } p \in \mathcal{P} \end{array} \right\},$$

endowed with the inner product:

$$\left\langle \sum_{j=1}^N \zeta_j \Phi(\mathbf{x}_j - \cdot), \sum_{k=1}^M \zeta'_k \Phi(\mathbf{y}_k - \cdot) \right\rangle_\Phi = \sum_{j=1}^N \sum_{k=1}^M \zeta_j \zeta'_k \Phi(\mathbf{x}_j - \mathbf{y}_k).$$

Let $\{\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_m\} \subseteq \mathbb{R}^p, m = \dim(\mathcal{P})$ *be a* $\mathcal{P}$*-unisolvent subset*[1]*, and let* $\{p_1, \cdots, p_m\} \subseteq \mathcal{P}$ *be a Lagrange basis of* $\mathcal{P}$ *for such a subset. Furthermore, define the projective map* $\Pi_{\mathcal{P}} : C(\mathbb{R}^p)$[2] $\to \mathcal{P}$ *as* $\Pi_{\mathcal{P}}(f) = \sum_{k=1}^m f(\boldsymbol{\psi}_k) p_k$*, and the map* $\mathcal{R} : \mathcal{F}_\Phi(\mathbb{R}^p) \to C(\mathbb{R}^p)$ *as*

[1] See Definition 2.6 of [32].

[2] $C(\mathbb{R}^p)$ is the space of continuous functions on $\mathbb{R}^p$.

$\mathscr{R}f(\mathbf{x}) = f(\mathbf{x}) - \Pi_{\mathscr{P}} f(\mathbf{x})$. *The* native space *for* Φ *is then defined as:*

$$\mathcal{N}_\Phi(\mathbb{R}^p) = \mathscr{R}(\mathscr{F}_\Phi(\mathbb{R}^p)) + \mathscr{P},$$

and is equipped with the semi-inner product:

$$\langle f, g\rangle_{\mathcal{N}_\Phi(\mathbb{R}^p)} = \langle \mathscr{R}^{-1}(f - \Pi_{\mathscr{P}} f), \mathscr{R}^{-1}(g - \Pi_{\mathscr{P}} g)\rangle_\Phi.$$

After obtaining the native space $\mathcal{N}_\Phi(\mathbb{R}^p)$, one can then define an appropriate inner product on $\mathcal{N}_\Phi(\mathbb{R}^p)$ to transform it into a RKHS:

Theorem 3 (Native space to RKHS; Thm. 10.20 of [32])**.** *The native space* $\mathcal{N}_\Phi(\mathbb{R}^p)$ *for a c.p.d. kernel* Φ *carries the inner product* $\langle f, g\rangle = \langle f, g\rangle_{\mathcal{N}_\Phi(\mathbb{R}^p)} + \sum_{k=1}^m f(\boldsymbol{\psi}_k) g(\boldsymbol{\psi}_k)$*. With this inner product,* $\mathcal{N}_\Phi(\mathbb{R}^p)$ *becomes a reproducing kernel Hilbert space with reproducing kernel:*

$$\begin{aligned} k(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x} - \mathbf{y}) &- \sum_{k=1}^m p_k(\mathbf{x})\Phi(\boldsymbol{\psi}_k - \mathbf{y}) - \sum_{l=1}^m p_l(\mathbf{y})\Phi(\mathbf{x} - \boldsymbol{\psi}_l) \\ &+ \sum_{k=1}^m \sum_{l=1}^m p_k(\mathbf{x}) p_l(\mathbf{y})\Phi(\boldsymbol{\psi}_k - \boldsymbol{\psi}_l) + \sum_{k=1}^m p_k(\mathbf{x}) p_k(\mathbf{y}). \end{aligned}$$

The following *generalized Fourier transform* (GFT) will also be useful:

Definition 5 (GFT; Defs. 8.8, 8.9 of [32])**.** *Suppose* $f : \mathbb{R}^p \to \mathbb{C}$ *is continuous and slowly increasing. A measurable function* $\hat{f} \in {}^3L_2^{loc}(\mathbb{R}^p \setminus \{0\})$ *is called the* generalized Fourier transform *of* f *if* $\exists m \in \mathbb{N}_0/2$ *such that* $\int_{\mathbb{R}^p} f(\mathbf{x})\hat{\gamma}(\mathbf{x})\, d\mathbf{x} = \int_{\mathbb{R}^p} \hat{f}(\omega)\gamma(\omega)\, d\omega$ *is satisfied for all* $\gamma \in \mathcal{S}_{2m}$*, where* $\hat{\gamma}$ *denotes the standard Fourier transform of* γ*. Here,* $\mathcal{S}_{2m} = \{\gamma \in \mathcal{S} : \gamma(\omega) = \mathcal{O}(\|\omega\|_2^{2m})$ *for* $\|\omega\|_2 \to 0\}$*, where* $\mathcal{S}$ *is the Schwartz space.*

[3] L_2^{loc} denotes the space of locally L_2-integrable functions.

Specific definitions for slowly increasing functions and Schwartz spaces can be found in Definitions 5.19 and 5.17 of [32]. Here, the order of the GFT $\hat{f}$ refers to the value m in Definition 5, which can reside on the half-integers $\mathbb{N}_0/2$ since the index of the underlying space $\mathcal{S}_{2m}$ will still be an integer.

With these concepts in hand, we now present the Koksma-Hlawka-like bound. As demonstrated below, the choice of the negative distance kernel $\Phi = -\|\cdot\|_2$ is important for connecting integration error with the distance-based energy distance $E(F, F_n)$.

Theorem 4 (Koksma-Hlawka)**.** *Let $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathfrak{X} \subseteq \mathbb{R}^p$ be a point set with e.d.f. F_n, and let $\Phi(\mathbf{x}) = -\|\mathbf{x}\|_2$. Then Φ is a c.p.d. kernel of order 1. Moreover:*

(a) The native space of Φ, $\mathcal{N}_\Phi(\mathbb{R}^p)$, can be explicitly written as:

$$\mathcal{N}_\Phi(\mathbb{R}^p) = \left\{ f \in C(\mathbb{R}^p) : \begin{array}{l} \text{(G1) } \exists m \in \mathbb{N}_0 \text{ s.t. } f(\mathbf{x}) = \mathcal{O}(\|\mathbf{x}\|_2^m) \text{ for } \|\mathbf{x}\|_2 \to \infty \\ \text{(G2) } f \text{ has a GFT } \hat{f} \text{ of order } 1/2 \\ \text{(G3) } \int \|\omega\|_2^{p+1} |\hat{f}(\omega)|^2 \, d\omega < \infty \end{array} \right\}, \tag{1.5}$$

with semi-inner product given by:

$$\langle f, g \rangle_{\mathcal{N}_\Phi(\mathbb{R}^p)} = \left\{ \Gamma((p+1)/2) 2^p \pi^{(p-1)/2} \right\}^{-1} \int \hat{f}(\omega) \overline{\hat{g}(\omega)} \|\omega\|_2^{p+1} \, d\omega, \tag{1.6}$$

(b) Consider the function space $\mathcal{G}_p = \mathcal{N}_\Phi(\mathbb{R}^p)$, equipped with inner product $\langle f, g \rangle_{\mathcal{G}_p} = \langle f, g \rangle_{\mathcal{N}_\Phi(\mathbb{R}^p)} + f(\boldsymbol{\psi}) g(\boldsymbol{\psi})$ for a fixed choice of $\boldsymbol{\psi} \in \mathfrak{X}$. Then $(\mathcal{G}_p, \langle \cdot, \cdot \rangle_{\mathcal{G}_p})$ is a RKHS, and for any integrand $g \in \mathcal{G}_p$, the integration error in (1.1) *is bounded by:*

$$I(g; F, F_n) \leq \|g\|_{\mathcal{G}_p} \sqrt{E(F, F_n)}, \quad \|g\|_{\mathcal{G}_p}^2 \equiv \langle g, g \rangle_{\mathcal{G}_p}. \tag{1.7}$$

The appeal of Theorem 4 is that it connects the integration error $I(g; F, F_n)$ with the

energy distance $E(F, F_n)$ for all integrands g in the function space $\mathscr{G}_p$. Similar to the usual Koksma-Hlawka inequality, such a theorem justifies the use of support points for integration, because the integration error for all functions in $\mathscr{G}_p$ can be sufficiently bounded by minimizing $E(F, F_n)$.

A natural question to ask is how large $\mathscr{G}_p$ is compared with the commonly-used Sobolev space $W_{s,2}$, i.e., the set of functions whose s-th order differentials have finite L_2 norm. Such a comparison is particularly important in light of the fact that an anchored variant of the Sobolev space is typically employed in QMC analysis (see, e.g., [16]). Recall that s can be extended to the non-negative real numbers using fractional calculus, in which case $W_{s,2}$ becomes the fractional Sobolev space. By comparing the definition of the fractional Sobolev space in the Fourier domain (see (3.7) in [33]), one can show that $W_{(p+1)/2,2}$ is contained within $\mathscr{G}_p$. Moreover, using the fact that $W_{s,2}$ is a decreasing family as $s > 0$ increases (see paragraph prior to Prop. 1.52 in [34]), it follows that $W_{\lceil (p+1)/2 \rceil,2} \subseteq W_{(p+1)/2,2} \subseteq \mathscr{G}_p$. In fact, for odd dimensions p, Theorem 10.43 of [32] shows that $\mathscr{G}_p$ is indeed equal to the Sobolev space $W_{\lceil (p+1)/2 \rceil,2} = W_{(p+1)/2,2}$, so the embedding result becomes an equality.

Viewing this embedding now in terms of Theorem 4, it follows that all integrands g with square-integrable $\lceil (p+1)/2 \rceil$-th order differentials enjoy the upper bound in (1.7). Hence, as dimension p grows, an increasing order of smoothness is required for integration using support points, which appears to be a necessary trade-off for the appealing d.c. formulation in (O). This is similar to the anchored Sobolev spaces employed in QMC, which requires integrands to have square-integrable mixed first derivatives.

Error convergence rate

Next, we investigate the convergence rate of $I(g; F, F_n)$ under support points. Under eigenvalue decay conditions, the following theorem establishes an *existence* result, which demonstrates the existence of a point set sequence achieving a particular error rate. An additional theorem then clarifies when such decay conditions are satisfied in practice. The

main purpose of these results is to demonstrate the quicker *theoretical* convergence of support points over Monte Carlo. From the simulations in Section 1.4, the rate below does not appear to be tight, and a quicker convergence rate is conjectured in Appendix A.3 of the supplemental article [35].

Theorem 5 (Error rate)**.** *Let F_n be the e.d.f. for support points $\{\boldsymbol{\xi}\}_{i=1}^n$, and let $g \in \mathscr{G}_p$. Define the kernel $k(\mathbf{x},\mathbf{y}) = \mathbb{E}\|\mathbf{x}-\mathbf{Y}\|_2 + \mathbb{E}\|\mathbf{y}-\mathbf{Y}\|_2 - \mathbb{E}\|\mathbf{Y}-\mathbf{Y}'\|_2 - \|\mathbf{x}-\mathbf{y}\|_2$, $\mathbf{Y},\mathbf{Y}' \overset{i.i.d.}{\sim} F$. If (a) $\mathbb{E}[\|\mathbf{Y}\|_2^3] < \infty$, and (b) the weighted eigenvalues of k under F satisfy $\sum_{k=1}^\infty \lambda_k^{1/\alpha} < \infty$ for some $\alpha > 1$, then:*

$$I(g; F, F_n) = \mathcal{O}\{\|g\|_{\mathscr{G}_p} n^{-1/2} (\log n)^{-(\alpha-1)/2}\}, \tag{1.8}$$

with constant terms depending on α and p.

Here, the weighted eigenvalue sequence of k under F is the decreasing sequence $(\lambda_k)_{k=1}^\infty$ satisfying $\lambda_k \phi_k(\mathbf{x}) = \mathbb{E}[k(\mathbf{x},\mathbf{Y})\phi_k(\mathbf{Y})]$, $\mathbb{E}[\phi_k^2(\mathbf{Y})] = 1$.

The following theorem provides some insight on when the eigenvalue decay condition $\sum_{k=1}^\infty \lambda_k^{1/\alpha} < \infty$ in Theorem 5 is satisfied.

Theorem 6 (Eigenvalue conditions)**.** *Let F_n and F be as in Theorem 5, and let $g \in \mathscr{G}_p$.*

(a) If $\mathcal{X} \subseteq \mathbb{R}^p$ is a bounded Borel set with non-empty interior, then $I(g; F, F_n) = \mathcal{O}\{\|g\|_{\mathscr{G}_p} n^{-1/2} (\log n)^{-(1-\nu)/(2p)}\}$ for any $\nu \in (0,1)$,

(b) If $\mathcal{X} \subseteq \mathbb{R}^p$ is measurable with positive Lebesgue measure, and there exists some $\beta > 0$ and $C \geq 0$ such that:

$$\limsup_{r\to\infty} r^\beta \int_{\mathcal{X}\setminus B_r(\mathbf{y})} \mathbb{E}\|\mathbf{x}-\mathbf{Y}\|_2 \, dF(\mathbf{x}) \leq C \text{ for all } \mathbf{y} \in \mathcal{X}, \tag{1.9}$$

then $I(g; F, F_n) = \mathcal{O}\{\|g\|_{\mathscr{G}_p} n^{-1/2} (\log n)^{-(\gamma-\nu)/(2p)}\}$ for any $\nu \in (0,\gamma)$, where $\gamma = \beta/(\beta+1)$ and $B_r(\mathbf{y})$ denotes an r-ball around $\mathbf{y}$.

Here, constant terms may depend on ν, p or β.

In words, Theorem 6 demonstrates the improvement of support points over MC under certain conditions on the sample space $\mathcal{X}$ or the desired distribution F. Specifically, part (a) requires the sample space $\mathcal{X}$ to be bounded with non-empty interior, whereas part (b) relaxes this boundedness restriction on $\mathcal{X}$ at the cost of the mild moment condition (2.15) on F. This condition holds for a large class of distributions which are not too heavy-tailed.

For illustration, consider the standard normal distribution for F, with sample space $\mathcal{X} = \mathbb{R}^p$. Note that, when $\|\mathbf{x}\|_2$ becomes large, $\mathbb{E}\|\mathbf{x} - \mathbf{Y}\|_2 \approx \|\mathbf{x}\|_2$. Hence, the condition in (2.15) becomes:

$$\limsup_{r\to\infty} r^\beta P(r), \quad P(r) \equiv (2\pi)^{-p/2} \int_{\mathbb{R}^p \setminus B_r(\mathbf{0})} \|\mathbf{x}\|_2 \exp\{-\|\mathbf{x}\|_2^2/2\}\, d\mathbf{x}.$$

Since $P'(r) \propto -r^p \exp\{-r^2/2\}$, it follows that $P(r) = \mathcal{O}(r^{p-1} \exp\{-r^2/2\})$, so $\limsup_{r\to\infty} r^\beta P(r) = 0$ for all $\beta > 0$. Applying part (b) of Theorem 6, support points enjoy a convergence rate of $\mathcal{O}\{n^{-1/2}(\log n)^{-(1-\nu)/(2p)}\}$ for any $\nu \in (0, 1)$ in this case. An analogous argument shows a similar rate holds for any spherically symmetric distribution (see, e.g., [2]) with an exponentially decaying density in its radius.

1.2.3 Comparison with MC and existing QMC methods

We first discuss the implications of Theorems 5 and 6 in comparison to Monte Carlo. Using the law of iterated logarithms [36], one can show that the error convergence rate for MC is bounded a.s. by $\mathcal{O}(n^{-1/2}\sqrt{\log\log n})$ for any distribution F. Comparing this with (1.8), the error rate of support points is asymptotically quicker than MC by at least some log-factor when dimension p is fixed. This improvement is reflected in the simulations in Section 1.4, where support points enjoy a considerable improvement over MC for all point set sizes n. When dimension p is allowed to vary (and assuming $\|g\|_{\mathcal{G}_p}$ and $\mathrm{Var}\{g(\mathbf{X})\}$, $\mathbf{X} \sim F$, do not depend on p), note that the MC rate is independent of p, while the rate in (1.8) can

have constants which depend on p. From a theoretical perspective, this suggests support points may be inferior to MC for high-dimensional integration problems. Such a *curse-of-dimensionality*, however, is not observed in our numerical experiments, where support points enjoy a sizable error reduction over MC for p as large as 500.

Compared to existing QMC techniques, the existence rate in Theorem 5 falls short in the uniform setting of $F = U[0,1]^p$. For fixed dimension p, [2] showed that for any integrand g with bounded variation (in the sense of Hardy and Krause), the error rate for classical QMC point sets is $\mathcal{O}\{n^{-1}(\log n)^p\}$, which is faster than (1.8). Moreover, when p is allowed to vary, it can be shown (see [37, 16]) that certain randomized QMC (RQMC) methods, such as the randomly-shifted lattice rules in [21], enjoy a root-mean-squared error rate of $\mathcal{O}(n^{-1+\delta})$ with $\delta \in (0, 1/2)$, where constant terms do not depend on dimension p. On the other hand, support points provide *optimal* integration points (in the sense of minimum energy) for *non-uniform* distributions at fixed sample size n. Because of this optimality, support points can enjoy reduced errors to existing QMC methods in practice, which we demonstrate later for a specific RQMC method called randomly-scrambled Sobol' sequences [24, 25]. This suggests the rate in Theorem 5 may not be tight, and further theoretical work is needed (we outline one possible proof approach in Appendix A.3 of the supplemental article [35]).

1.3 Generating support points

The primary appeal of support points is the efficiency by which these point sets can be optimized, made possible by exploiting the d.c. structure of the energy distance. Here, we present two algorithms, `sp.ccp` and `sp.sccp`, which employ a combination of the convex-concave procedure (CCP) with resampling to quickly optimize support points. `sp.ccp` should be used when sample batches are computationally expensive to obtain from F, whereas `sp.sccp` should be used when samples can be easily obtained. We prove the convergence of both algorithms to a stationary point set, and briefly discuss their

running times.

1.3.1 Algorithm statements

We first present the steps for `sp.ccp`, then introduce `sp.sccp` as an improvement on `sp.ccp` when multiple sample batches from F can be efficiently obtained. Suppose a single sample batch $\{\mathbf{y}_m\}_{m=1}^N$ is obtained from F. Using this, `sp.ccp` optimizes the following Monte Carlo approximation of the support points formulation (O):

$$\underset{\mathbf{x}_1,\cdots,\mathbf{x}_n}{\operatorname{argmin}} \hat{E}(\{\mathbf{x}_i\};\{\mathbf{y}_m\}) \equiv \frac{2}{nN}\sum_{i=1}^{n}\sum_{m=1}^{N}\|\mathbf{y}_m-\mathbf{x}_i\|_2 - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{x}_i-\mathbf{x}_j\|_2. \tag{MC}$$

The approximated objective $\hat{E}$ was originally proposed by [13] as a two-sample GOF statistic for testing whether $\{\mathbf{y}_m\}_{m=1}^N$ and $\{\mathbf{x}_i\}_{i=1}^n$ are generated from the same distribution. Posed as an optimization problem, however, the goal in (MC) is to recover the point set which best represents the random sample $\{\mathbf{y}_m\}_{m=1}^N$ from F in terms of goodness-of-fit.

The key observation here is that the objective function $\hat{E}$ can be written as a difference of convex functions in $\mathbf{x} = (\mathbf{x}_1,\cdots,\mathbf{x}_n)$, namely, the two terms in (MC). This structure allows for efficient optimization using d.c. programming methods, which enjoy a well-established theoretical and numerical framework [38, 39]. While global optimization algorithms have been proposed for d.c. programs (e.g., [40]), such methods are typically quite slow in practice [41], and may not be appropriate for the large-scale problem at hand. Instead, we employ a d.c. algorithm called the convex-concave procedure (CCP, see [42]) which, in conjunction with the distance-based property of the energy distance, allows for efficient optimization of (MC).

The main idea in CCP is to first replace the concave term in the d.c. objective with a convex upper bound, then solve the resulting "surrogate" formulation (which is convex) using convex programming techniques. This procedure is then repeated until the solution iterates converge. CCP can be seen as a specific case of majorization-minimization (MM, see [43]), a popular optimization technique in statistics. The key to computational efficiency lies in

Algorithm 1 `sp.ccp`: Support points using one sample batch

- Sample $\mathscr{D}^{[0]} = \{\mathbf{x}_i^{[0]}\}_{i=1}^n$ i.i.d. from $\{\mathbf{y}_m\}_{m=1}^N$.
- Set $l = 0$, and **repeat** until convergence of $\mathscr{D}^{[l]}$:
  - **For** $i = 1, \cdots, n$ **do parallel**:
    - Set $\mathbf{x}_i^{[l+1]} \leftarrow M_i(\mathscr{D}^{[l]}; \{\mathbf{y}_m\}_{m=1}^N)$, with M_i defined in (1.11).
  - Update $\mathscr{D}^{[l+1]} \leftarrow \{\mathbf{x}_i^{[l+1]}\}_{i=1}^n$, and set $l \leftarrow l + 1$.
- Return the converged point set $\mathscr{D}^{[\infty]}$.

finding a convex surrogate formulation which can be minimized in closed-form. Here, such a formulation can be obtained by exploiting the distance-based structure of (MC), with its closed-form minimizer given by the iterative map $\mathbf{x}_i^{[l+1]} \leftarrow M_i(\{\mathbf{x}_j^{[l]}\}_{j=1}^n; \{\mathbf{y}_m\}_{m=1}^N)$, $i = 1, \cdots, n$, where M_i is given in (1.11). The appeal of CCP here is two-fold. First, the evaluation of the iterative maps $M_i, i = 1, \cdots, n$ requires $\mathcal{O}(n^2 p)$ work, thereby allowing for the efficient generation of moderately-sized point sets in moderately-high dimensions. Second, the computation of these maps can be greatly sped up using parallel computing, a point further discussed in Section 1.3.3.

Algorithm 1 outlines the detailed steps for `sp.ccp` following the above discussion. One caveat for `sp.ccp` is that it uses only one sample batch from F, even when multiple sample batches can be generated efficiently. This motivates the second algorithm, `sp.sccp`, whose steps are outlined in Algorithm 2. The main difference for `sp.sccp` is that $\{\mathbf{y}_m\}_{m=1}^N$ is *resampled* within each CCP iteration (a procedure known as stochastic MM). This resampling scheme allows `sp.sccp` to converge to a stationary point set for the desired problem (O), which we demonstrate next.

1.3.2 Algorithmic convergence

For completeness, a brief overview of MM is provided, following [43].

Definition 6 (Majorization function)**.** *Let $f : \mathbb{R}^s \to \mathbb{R}$ be the objective function to be minimized. A function $h(\mathbf{z}|\mathbf{z}')$* majorizes *$f(\mathbf{z})$ at a point $\mathbf{z}' \in \mathbb{R}^s$ if $h(\mathbf{z}|\mathbf{z}') \geq f(\mathbf{z})$, with equality holding when $\mathbf{z} = \mathbf{z}'$.*

Algorithm 2 `sp.sccp`: Support points using multiple sample batches

- Sample $\mathscr{D}^{[0]} = \{\mathbf{x}_i^{[0]}\}_{i=1}^n \overset{i.i.d.}{\sim} F$.
- Set $l = 0$, and **repeat** until convergence of $\mathscr{D}^{[l]}$:
  - Resample $\{\mathbf{y}_m^{[l]}\}_{m=1}^N \overset{i.i.d.}{\sim} F$.
  - **For** $i = 1, \cdots, n$ **do parallel**:
    - Set $\mathbf{x}_i^{[l+1]} \leftarrow M_i(\mathscr{D}^{[l]}; \{\mathbf{y}_m^{[l]}\}_{m=1}^N)$, with M_i defined in (1.11).
  - Update $\mathscr{D}^{[l+1]} \leftarrow \{\mathbf{x}_i^{[l+1]}\}_{i=1}^n$, and set $l \leftarrow l + 1$.
- Return the converged point set $\mathscr{D}^{[\infty]}$.

Starting at an initial point $\mathbf{z}^{[0]}$, the goal in MM is to minimize the majorizing function h as a surrogate for the true objective f, and iterate the updates $\mathbf{z}^{[l+1]} \leftarrow \operatorname{argmin}_{\mathbf{z}} h(\mathbf{z}|\mathbf{z}^{[l]})$ until convergence. This iterative procedure has the so-called descent property $f(\mathbf{x}^{[l+1]}) \leq f(\mathbf{x}^{[l]})$, which ensures solution iterates are always decreasing in f. The key for efficiency is to find a majorizing function g with a closed-form minimizer which is easy to compute.

Consider now the Monte Carlo approximation in (MC), which has a d.c. formulation in $\{\mathbf{x}_i\}_{i=1}^n$, with concave term $-n^{-2}\sum_{i=1}^n\sum_{j=1}^n \|\mathbf{x}_i - \mathbf{x}_j\|_2$. Following CCP, we first majorize this term using a first-order Taylor expansion at the current iterate $\{\mathbf{x}'_j\}_{j=1}^n$, yielding the surrogate convex program:

$$\begin{aligned}
&\operatorname*{argmin}_{\mathbf{x}_1,\cdots,\mathbf{x}_n} h(\{\mathbf{x}_i\}_{i=1}^n; \{\mathbf{x}'_j\}_{j=1}^n) \\
&\equiv \frac{2}{nN}\sum_{i=1}^n\sum_{m=1}^N \|\mathbf{y}_m - \mathbf{x}_i\|_2 - \frac{1}{n^2}\left[\sum_{i=1}^n\sum_{j=1}^n\left(\|\mathbf{x}'_i - \mathbf{x}'_j\|_2 + \frac{2(\mathbf{x}_i - \mathbf{x}'_i)^T(\mathbf{x}'_i - \mathbf{x}'_j)}{\|\mathbf{x}'_i - \mathbf{x}'_j\|_2}\right)\right].
\end{aligned} \tag{1.10}$$

Implicit here is the assumption that the current point set is pairwise distinct, i.e., $\mathbf{x}'_i \neq \mathbf{x}'_j$ for all $i, j = 1, \cdots, n$. From simulations, this appears to be always satisfied by initializing the algorithm with a pairwise distinct point set, because the random sampling of $\{\mathbf{y}_m\}$ and the "almost-random" round-off errors [44] in the evaluation of M_i force subsequent point sets to be pairwise distinct. Such an assumption can also be easily checked after each iteration.

While (1.10) can be solved using gradient-based convex programming techniques, this

can be computationally burdensome when n or p becomes large, because such methods may require many evaluations of h and its subgradient. Instead, the following lemma allows us to perform a slight "convexification" of the convex term in (1.10), which then yields a efficient closed-form minimizer.

Lemma 1 (Convexification)**.** $Q(\mathbf{x}|\mathbf{x}') = \frac{\|\mathbf{x}\|_2^2}{2\|\mathbf{x}'\|_2} + \frac{\|\mathbf{x}'\|_2}{2}$ *majorizes* $\|\mathbf{x}\|_2$ *at* $\mathbf{x}'$ *for any* $\mathbf{x}' \in \mathbb{R}^p$.

Lemma 1 has an appealing geometric interpretation. Viewing $\|\mathbf{x}\|_2$ as a second-order cone centered at $\mathbf{0}$, $Q(\mathbf{x}|\mathbf{x}')$ can be interpreted as the tightest convex paraboloid intersecting this cone at $\mathbf{x}'$. Note that the quadratic nature of the majorizer Q, which is crucial for deriving a closed-form minimizer, is made possible by the distance-based structure of the energy distance.

From this, the following lemma provides a quadratic majorizer for (1.10), along with its corresponding closed-form minimizer:

Lemma 2 (Closed-form iterations)**.** *Define the function* h^Q *as:*

$$h^Q(\{\mathbf{x}_i\}_{i=1}^n; \{\mathbf{x}'_j\}_{j=1}^n) \equiv \frac{2}{nN} \sum_{i=1}^n \sum_{m=1}^N \left\{ \frac{\|\mathbf{y}_m - \mathbf{x}_i\|_2^2}{2\|\mathbf{y}_m - \mathbf{x}'_i\|_2} + \frac{\|\mathbf{y}_m - \mathbf{x}'_i\|_2}{2} \right\} - \frac{1}{n^2} \left[\sum_{i=1}^n \sum_{j=1}^n \left(\|\mathbf{x}'_i - \mathbf{x}'_j\|_2 + \frac{2(\mathbf{x}_i - \mathbf{x}'_i)^T(\mathbf{x}'_i - \mathbf{x}'_j)}{\|\mathbf{x}'_i - \mathbf{x}'_j\|_2} \right) \right],$$

Then $h^Q(\cdot; \{\mathbf{x}'_j\}_{j=1}^n)$ *majorizes* $\hat{E}$ *at* $\{\mathbf{x}'_j\}_{j=1}^n$*. Moreover, the global minimizer of* $h^Q(\cdot; \{\mathbf{x}'_j\}_{j=1}^n)$ *is given by:*

$$\begin{aligned} \mathbf{x}_i &= M_i(\{\mathbf{x}'_j\}_{j=1}^n; \{\mathbf{y}_m\}_{m=1}^N) \\ &\equiv \left(\sum_{m=1}^N \|\mathbf{x}'_i - \mathbf{y}_m\|_2^{-1} \right)^{-1} \left(\frac{N}{n} \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\mathbf{x}'_i - \mathbf{x}'_j}{\|\mathbf{x}'_i - \mathbf{x}'_j\|_2} + \sum_{m=1}^N \frac{\mathbf{y}_m}{\|\mathbf{x}'_i - \mathbf{y}_m\|_2} \right), \ i = 1, \cdots, n. \end{aligned} \tag{1.11}$$

One can now prove the convergence of `sp.ccp` and `sp.sccp`.

Theorem 7. (Convergence - `sp.ccp`) *Assume $\mathfrak{X}$ is closed and convex. For any pairwise distinct $\mathfrak{D}^{[0]} \subseteq \mathfrak{X}$ and fixed sample batch $\{\mathbf{y}_m\}_{m=1}^N \subseteq \mathfrak{X}$, the sequence $(\mathfrak{D}^{[l]})_{l=1}^\infty$ in Algorithm 1 converges to a limiting point set $\mathfrak{D}^{[\infty]}$ which is stationary for $\hat{E}$.*

Theorem 8. (Convergence - `sp.sccp`) *Assume $\mathfrak{X}$ is compact and convex. For any pairwise distinct $\mathfrak{D}^{[0]} \subseteq \mathfrak{X}$, the sequence $(\mathfrak{D}^{[l]})_{l=1}^\infty$ in Algorithm 2 converges a.s. to a limiting point set $\mathfrak{D}^{[\infty]}$ which is stationary for E.*

(Recall that $\mathbf{z} \in D$ is a stationary solution for a function $f : D \subseteq \mathbb{R}^s \to \mathbb{R}$ if:

$$f'(\mathbf{z}, \mathbf{d}) \geq 0 \quad \text{for all } \mathbf{d} \in \mathbb{R}^s \text{ s.t. } \mathbf{z} + \mathbf{d} \in D,$$

where $f'(\mathbf{z}, \mathbf{d})$ is the directional derivative of f at $\mathbf{z}$ in direction $\mathbf{d}$.) Note that the compactness condition on $\mathfrak{X}$ in Theorem 8 is needed to prove the convergence of stochastic MM algorithms, since it allows for an application of the law of large numbers (see [45] for details).

1.3.3 Running time and parallelization

Regarding the running time of `sp.ccp`, it is well known that MM algorithms enjoy a linear error convergence rate [46]. This means $L = \mathcal{O}(\log \delta^{-1})$ iterations of (1.11) are sufficient for achieving an objective gap of $\delta > 0$ from the stationary solution. Since the maps in (1.11) require $\mathcal{O}\{n(n+N)p\}$ work to compute, the running time of `sp.ccp` is $\mathcal{O}\{n(n+N)p \log \delta^{-1}\}$. Assuming the batch sample size N does not increase with n or p, this time reduces to $\mathcal{O}(n^2 p \log \delta^{-1})$, which suggests the proposed algorithm can efficiently generate moderately-sized point sets in moderately-high dimensions, but may be computationally burdensome for large point sets. While a similar linear error convergence is difficult to establish for `sp.sccp` due to its stochastic nature (see [47, 48]), its running time is quite similar to `sp.ccp` from simulations.

The separable form of (1.11) also allows for further computational speed ups using parallel processing. As outlined in Algorithms 1 and 2, the iterative map for each point $\mathbf{x}_i$ can be computed in parallel using separate processing cores. Letting P be the total number of computation cores available, such a parallelization scheme reduces the running time of `sp.ccp` and `sp.sccp` to $\mathcal{O}(\lceil n/P \rceil np \log \delta^{-1})$, thereby allowing for quicker optimization of large point sets. This feature is particularly valuable given the increasing availability of multi-core processors in personal laptops and computing clusters.

1.4 Simulations

Several simulations are presented here which demonstrate the effectiveness of support points in practice. We first discuss the space-filling property of support points, then comment on its computation time using `sp.sccp`. Finally, we compare the integration performance of support points with MC and a RQMC method called IT-RSS (defined later).

1.4.1 Visualization and timing

For visualization, Figure 1.2 shows the $n = 128$-point point sets for the i.i.d. $N(0, 1)$ and $Exp(1)$ distributions in $p = 2$ dimensions, with lines outlining density contours (additional visualizations provided in Appendix B of the supplemental article [35]). Support points are plotted on the left, Monte Carlo samples in the middle and inverse Sobol' points on the right. The latter is generated by choosing the Sobol' points on $U[0, 1]^2$ which maximize the minimum interpoint distance over 10,000 random scramblings (see next section for details), then performing an inverse-transform of F on such a point set. From this figure, support points appear to be slightly more visually representative of the underlying distribution F than the inverse Sobol' points, and much more representative than MC. Specifically, the proposed point set is concentrated in regions with high density, but each point is sufficiently spaced out from one another to maximize their representative power. Borrowing a term from design-of-experiments literature [49], we call point sets with these two properties to

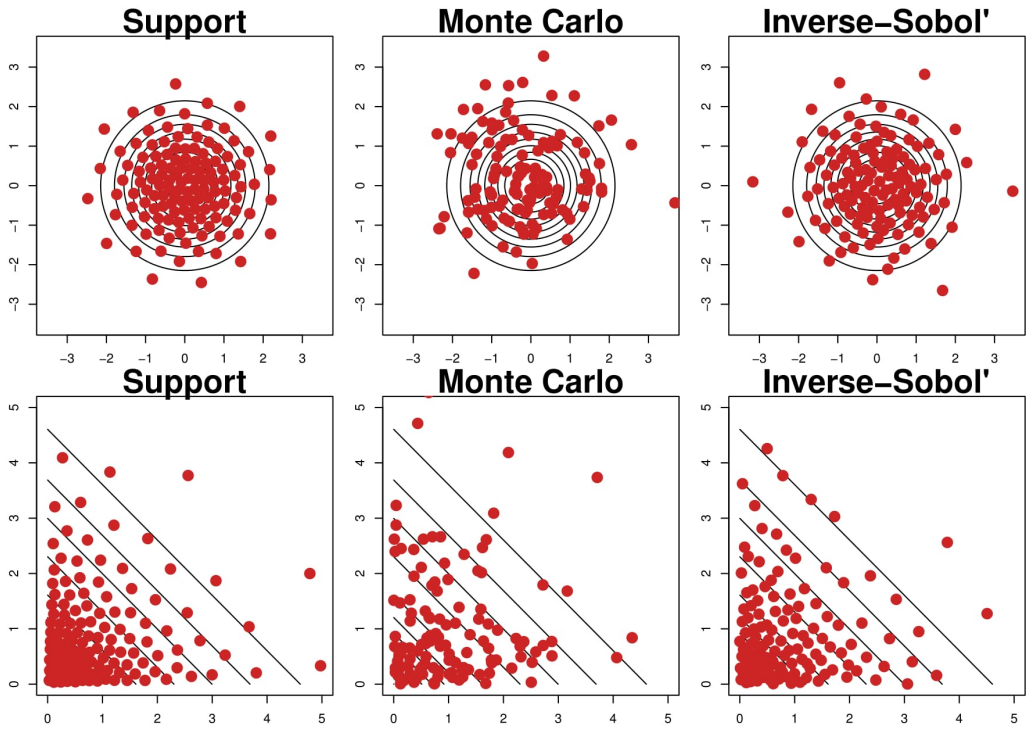

Figure 1.2: $n = 128$ support points, MC points and inverse Sobol' points for i.i.d. $N(0, 1)$ and Exp(1) in $p = 2$ dimensions. Lines represent density contours.

be *space-filling* on F. A key reason for this space-fillingness is the distance-based property of the energy distance: the two terms for $E(F, F_n)$ in (1.4) force support points to not only mimic the desired distribution F, but also ensure no two points are too close together. This allows for a more appealing visual representation of F, and can provide more robust integration performance.

Regarding computation time, Figure 1.3 shows the times (in seconds) needed for `sp.sccp` to generate support points for the i.i.d. $Beta(2, 4)$ distribution, first as a function of point set size n with fixed dimension p, then as a function of p with fixed n. The resampling size is fixed at $N = 10,000$ for all choices of n and p. Similar times are reported for other distributions, and are not reported for brevity. All computations are performed on a 12-core Intel Xeon 3.50 Ghz processor. From this figure, two interesting observations can be made. First, for fixed n, these plots show that the empirical running times grow quite linearly in p, whereas for fixed p, these running times exhibit a slow quadratic (but almost

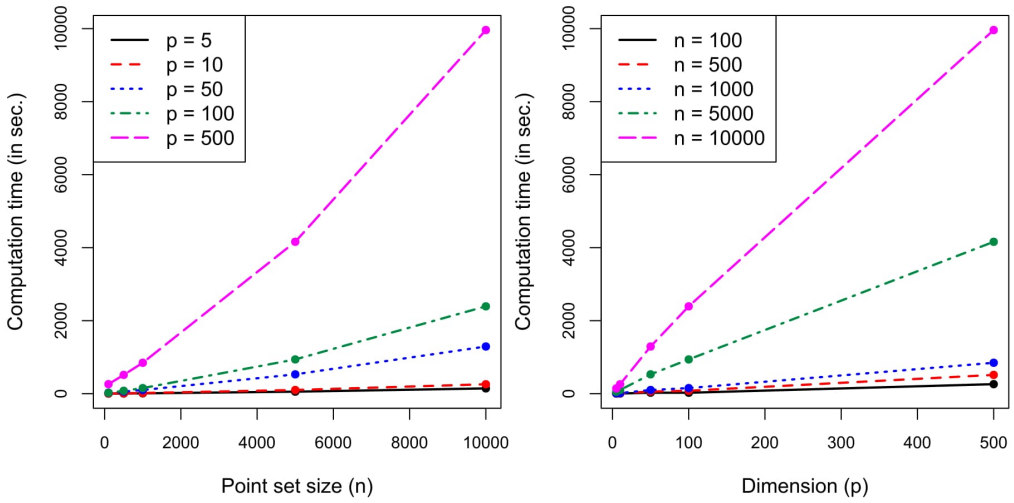

Figure 1.3: Computation time (in seconds) of `sp.sccp` as a function of point set size (n) and dimension (p) for the i.i.d. $Beta(2, 4)$ distribution.

linear) growth in n. This provides evidence for the $\mathcal{O}(n^2p)$ running time asserted in Section 1.3.3. Second, as a result of this running time, support points can be generated efficiently for moderate-sized point sets in moderately-high dimensions. For $p = 2$, the required times for generating $n = 50 - 10,000$ points range from 3 seconds to 2 minutes; for $p = 50$, 27 seconds to 20 minutes; and for $p = 500$, 4 minutes to 2.5 hours. While these times are quite fast from an optimization perspective, they are still slower than number-theoretic QMC methods, which can generate, say, $n = 10^6$ points in $p = 10^3$ dimensions in a matter of seconds. The appeal for support points is that, by exploiting the d.c. structure of the energy distance in [13], one obtains for any distribution (locally) minimum energy sampling points which can outperform number-theoretic QMC methods.

1.4.2 Numerical integration

We now investigate the integration performance of support points in comparison with Monte Carlo and an RQMC method called the inverse-transformed randomized Sobol' sequences (IT-RSS). The former is implemented using the Mersenne twister [50], the default pseudo-random number generator in the software R [51]. The latter is obtained by (a) gen-

erating a randomized Sobol' sequence using the R package `randtoolbox` [52] (which employs Owen-style scrambling [25] with Sobol' sequences generated in the implementation of [53]), and (b) performing the inverse-transform of F on the resulting point set. As mentioned in Section 1.2, IT-RSS performs well in the uniform setting $F = U[0,1]^p$, and provides a good benchmark for comparing support points with existing QMC methods.

The simulation set-up is as follows. Support points are generated using `sp.sccp`, with point set sizes ranging from $n = 50$ to $10,000$ and resampling size N fixed at $10,000$. Since MC and IT-RSS are randomized methods, we replicate both for 100 trials to provide an estimate of error variability, with replications seeded for reproducibility. Three distributions are considered for F: the i.i.d. $N(0,1)$, the i.i.d. $Exp(1)$ and the i.i.d. $Beta(2,4)$ distributions, with p ranging from 5 to 500. For the integrand g, two (modified) test functions are taken from [54]: the Gaussian peak function (GAPK): $g(\mathbf{x}) = \exp\{-\sum_{l=1}^{p} \alpha_l^2 (x_l - u_l)^2\}$ and the (modified) oscillatory function (OSC): $g(\mathbf{x}) = \exp\{-\sum_{l=1}^{p} \beta_l x_l^2\} \cos(2\pi u_1 + \sum_{l=1}^{p} \beta_l x_l)$. Here, $\mathbf{x} = (x_l)_{l=1}^{p}$, u_l is the marginal mean for the l-th dimension of F, and the scale parameters α_l and β_l are set as $20/p$ and $5/p$, respectively.

Figure 1.4 shows the resulting log-absolute errors in $p = 5$, 50 and 200 dimensions for GAPK under the i.i.d. $Exp(1)$ distribution, and for OSC under the i.i.d. $N(0,1)$ distribution (results are similar for other settings, and are omitted for brevity). For MC and IT-RSS, the dotted lines indicate average error decay, and the shaded bands mark the area between the 25-th and 75-th error quantiles. Two observations can be made here. First, for all choices of n, support points enjoy considerably reduced errors compared to the averages of both MC and IT-RSS, with the proposed method providing an improvement to the 25-th quantiles of IT-RSS for most settings. Second, this advantage over MC and IT-RSS persists in both low and moderate dimensions. In view of the relief from dimensionality enjoyed by IT-RSS, this gives some evidence that support points *may* enjoy a similar property as well, a stronger assertion than is provided in Theorem 5 or 6. Exploring the theoretical

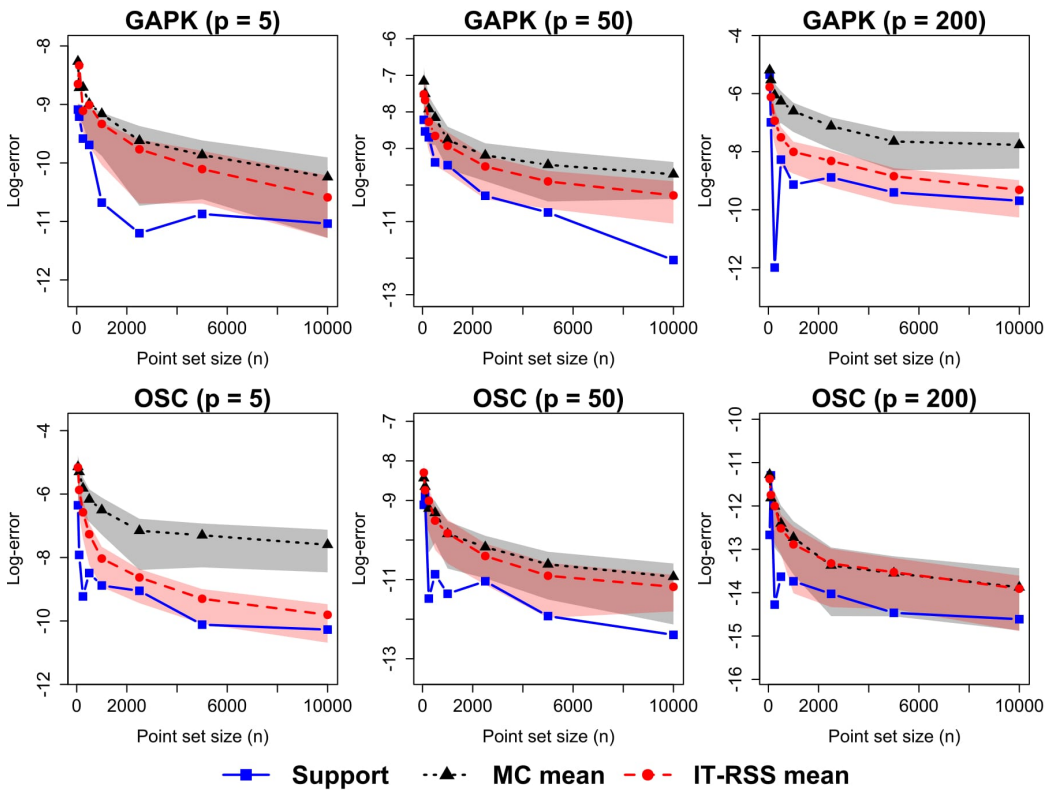

Figure 1.4: Log-absolute errors for GAPK under the i.i.d. $Exp(1)$ distribution (top) and for OSC under the i.i.d. $N(0, 1)$ distribution (bottom). Lines denote log average-errors, and shaded bands mark the 25-th and 75-th quantiles.

performance of support points in high dimensions will be an interesting direction for future work.

In summary, for point set sizes as large as $10,000$ points in dimensions as large as 500, simulations show that support points can be efficiently generated and enjoy improved performance over MC and IT-RSS. This opens up a wide range of important applications for support points in both small-data and big-data problems, two of which we describe next.

1.5 Applications of support points

1.5.1 Uncertainty propagation in expensive simulations

We first highlight an important small-data application of support points in simulation. With the development of powerful computational tools, computer simulations are becoming the

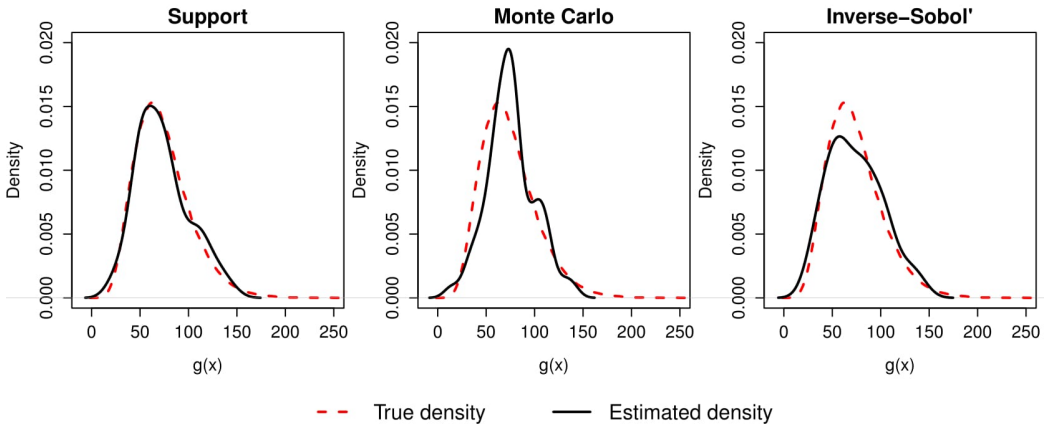

Figure 1.5: True and estimated density functions for $g(\mathbf{X})$ using $n = 60$ points.

de-facto method for conducting engineering experiments. For such simulations, a key point of interest is *uncertainty propagation*, or how uncertainty in input variables (resulting from, say, manufacturing tolerances) propagate and affect output variability. Mathematically, let $g(\mathbf{x})$ be the observed output at input setting $\mathbf{x}$, and let $\mathbf{X} \sim F$ denote input uncertainties. The distribution $g(\mathbf{X})$ can then be seen as the resulting uncertainty on system output. For engineers, the estimation of $g(\mathbf{X})$ using as few simulation runs as possible is of great importance, because each run can be computationally and monetarily expensive.

To demonstrate the effectiveness of support points for this problem, we use the *borehole physical model* [55], which simulates water flow rate through a borehole. The 8 input variables for this model, along with their corresponding uncertainty distributions (assumed to be mutually independent), are summarized in Appendix C of the supplemental article [35]. To reflect the expensive cost of simulations, we test only small point set sizes ranging from $n = 20$ to $n = 100$ runs. Support points are generated using `sp.sccp` with the same settings as before, with the randomized MC and IT-RSS methods replicated for 100 trials.

Consider now the estimation of the output distribution $g(\mathbf{X})$, which quantifies the uncertainty in water flow rate. Figure 1.5 compares the estimated density function of $g(\mathbf{X})$ using $n = 60$ points with its true density, where the latter estimated using a large Monte Carlo sample. Visually, support points provide the best density approximation for $g(\mathbf{X})$,

Table 1.1: Prior specification for the tree growth model (left), and the ratios of thinning over support point error for posterior quantities (right). $R_\mu(n)$ and $R_{\sigma^2}(n)$ denote the error ratios for posterior means and variances using n points, respectively.

Parameter	*Prior*	$R_\mu(375)$	$R_\mu(750)$	$R_{\sigma^2}(375)$	$R_{\sigma^2}(750)$
ϕ_{i1}	$\log \phi_{i1} \overset{indep.}{\sim} N(\mu_1, \sigma_1^2)$	2.27	2.75	15.89	6.37
ϕ_{i2}	$\log(\phi_{i2}+1) \overset{indep.}{\sim} N(\mu_2, \sigma_2^2)$	2.10	3.58	18.01	2.47
ϕ_{i3}	$\log(-\phi_{i3}) \overset{indep.}{\sim} N(\mu_3, \sigma_3^2)$	1.59	2.23	11.90	102.49
σ_C^2	$\sigma_C^2 \sim$ Inv-Gamma$(0.001, 0.001)$	0.98	2.80	6.15	7.69
$r(1600)$	$r(t) = \frac{1}{5}\sum_{i=1}^{5} \frac{\partial}{\partial s}\eta_i(s)\big\vert_{s=t}$	1.95	3.17	-	-
$r(1625)$		2.30	3.28	-	-
$r(1650)$		2.51	3.04	-	-
μ_j	$\mu_j \overset{i.i.d.}{\sim} N(0, 100)$	-	-	-	-
σ_j^2	$\sigma_j^2 \overset{i.i.d.}{\sim}$ Inv-Gamma$(0.01, 0.01)$	-	-	-	-

capturing well both the peak and tails of the desired output distribution. This suggests support points are not only asymptotically consistent for density estimation, but may also be optimal in some sense. A similar conclusion holds in the estimation of the expected flow rate $\mathbb{E}[g(\mathbf{X})]$ (see Appendix C of the supplemental article [35]).

1.5.2 Optimal MCMC reduction

The second application of support points is as an improved alternative to MCMC thinning for Bayesian computation. Thinning here refers to the discarding of all but every k-th sample for an MCMC sample chain obtained from the posterior distribution. This is performed for several reasons (see [56]): it reduces high autocorrelations in the MCMC chain, saves computer storage space, and reduces processing time for computing derived posterior quantities. However, by carelessly throwing away samples, a glaring fault of thinning is that samples from thinned chains are inherently less accurate than that from the full chain. To this end, the proposed algorithm `sp.ccp` can provide considerable improvements to thinning by optimizing for a point set which best captures the distribution of the full MCMC chain.

We illustrate this improvement using the orange tree growth model in [57]. The data

here consists of trunk circumference measurements $\{Y_i(t_j)\}_{i=1}^{5}{}_{j=1}^{7}$, where $Y_i(t_j)$ denotes the measurement taken on day t_j from tree i. To model these measurements, the growth model $Y_i(t_j) \overset{indep.}{\sim} N(\eta_i(t_j), \sigma_C^2), \eta_i(t_j) = \phi_{i1}/(1 + \phi_{i2}\exp\{\phi_{i3}t_j\})$ was assumed in [57], where ϕ_{i1}, ϕ_{i2} and ϕ_{i3} control the growth behavior of tree i. There are 16 parameters in total, which we denote by the set $\Theta = (\phi_{11}, \phi_{12}, \cdots, \phi_{53}, \sigma^2)$. Since no prior information is available on Θ, vague priors are assigned, with the full specification provided in the left part of Table 1.1. MCMC sampling is then performed for the posterior distribution using the R package `STAN` [58], with the chain run for 150,000 iterations and the first 75,000 of these discarded as burn-in. The remaining $N = 75,000$ samples are then thinned at a rate of 200 and 100, giving $n = 375$ and $n = 750$ thinned samples, respectively. Support points are generated using `sp.ccp` for the same choices of n, using the full MCMC chain as the approximating sample $\{\mathbf{y}_m\}_{m=1}^{N}$. Since posterior variances vary greatly between parameters, we first rescale each parameter in the MCMC chain to unit variance before performing `sp.ccp`, then scale back the resulting support points after.

These two methods are then compared on how well they estimate two quantities: (a) marginal posterior means and standard deviations of each parameter, and (b) the averaged instantaneous growth rate $r(t)$ (see Table 1.1) at three future times. True posterior quantities are estimated by running a longer MCMC chain with 600,000 iterations. This comparison is summarized in the right part of Table 1.1, which reports the ratios of thinning over support point error for each parameter. Keeping in mind that a ratio exceeding 1 indicates lower errors for support points, one can see that `sp.ccp` provides a sizable improvement over thinning for nearly all posterior quantities. Such a result should not be surprising, because `sp.ccp` compacts the *full* MCMC chain into a set of optimal representative points, whereas thinning wastes valuable information by discarding a majority of this chain.

1.6 Conclusion and future work

In this chapter, a new method is proposed for compacting a continuous distribution F into a set of representative points called support points, which are defined as the minimizer of the energy distance in [30]. Three theorems are proven here which justify the use of these point sets for integration. First, we showed that support points are indeed representative of the desired distribution, in that these point sets converge in distribution to F. Second, we provided a Koksma-Hlawka-like bound which connects integration error with the energy distance for a large class of integrands. Lastly, using an existence result, we demonstrated the theoretical error improvement of support points over Monte Carlo. A key appeal of support points is its formulation as a difference-of-convex optimization problem. The two proposed algorithms, `sp.ccp` and `sp.sccp`, exploit this structure to efficiently generate moderate-sized point sets ($n \leq 10,000$) in moderately-high dimensions ($p \leq 500$). Simulations confirm the improved performance of support points to MC and a specific QMC method, and the practical applicability of the proposed point set is illustrated using two real-world applications, one for small-data and the other for big-data. An efficient C++ implementation of `sp.ccp` and `sp.sccp` is made available in the R package `support` [59].

While the current chapter establishes some interesting results for support points, there are still many exciting avenues for future research. First, we are interested in exploring a tighter convergence rate for support points which reflects its empirical performance from simulations, particularly for high-dimensional problems. Next, the d.c. formulation of the energy distance can potentially be further exploited for the global optimization of support points. Moreover, by minimizing the *distance-based* energy distance, support points also have an inherent link to the *distance-based* designs used in computer experiments [49, 60, 61], and exploring this connection may reveal interesting insights between the two fields, and open up new approaches for uncertainty quantification in engineering [62] and

machine-learning [63] problems. Lastly, motivated by [29] and [60], rep-points in high-dimensions should not only provide a good representation of the full distribution F, but also for *marginal* distributions of F. Such a projective property is enjoyed by most QMC point sets in the literature [16], and new methodology is needed to incorporate this within the support points framework.

CHAPTER 2
PROJECTED SUPPORT POINTS – A NEW METHOD FOR HIGH-DIMENSIONAL DATA REDUCTION

2.1 Introduction

This chapter explores a new way to compact a continuous distribution F into a set of representative points with good projective properties, which we call *projected support points* (PSPs). Representative point sets have important applications in statistics and engineering, because they provide an improved representation of F compared to a random sample. However, in many practical problems, two additional concerns need to be addressed: (a) the sample space of F (call this $\mathfrak{X}$) is often high-dimensional, and (b) the underlying problem typically focuses on a low-dimensional subspace of $\mathfrak{X}$. Such a scenario is commonly encountered in Bayesian analysis, where a modeler considers many parameters, but may only be interested in posterior quantities involving a handful of these parameters. We present here a flexible framework for generating point sets which not only enjoys excellent goodness-of-fit (GOF) of F, but also provides good fit of the marginal distributions of F. We refer to the latter property as *projected goodness-of-fit* (PGOF) for the remainder of the chapter.

The motivating idea for PGOF – namely, low-dimensional structure in high-dimensional functions – has been studied in both deterministic sampling (Quasi-Monte Carlo, or QMC) and experimental design. One of the earliest mentions of this in QMC is in [2] and [29], who advocated for a discrepancy measure which does not increase under projections of a point set. A related concept called effective dimension was then proposed in [64] and [65],

The paper based on this chapter is under revision in *Journal of the American Statistical Association*.

quantifying the belief that certain dimensions in an integral are more important than others. These works have culminated in a recent thrust in QMC on establishing dimension-free error rates for integration on the uniform hypercube $U[0,1]^p$ (see, e.g., [37, 16]). This focus on low-dimensional structure is mirrored in experimental design literature, specifically in the principles of effect sparsity, hierarchy and heredity [66, 67, 68], which serve as guiding rules for analyzing experimental data. Recently, these principles were further developed in the maximum projection (MaxPro) designs [69], which enjoy good space-filling properties on projections of the design space. The PSPs proposed here provides a unifying framework which connects these developments within the context of integration under non-uniform distributions.

The framework for PSPs can be seen as an extension of two recent developments in deterministic sampling: kernel herding and support points. The first, kernel herding [70], generates a point set sequence by sequentially minimizing some kernel-based discrepancy measure between the desired distribution F and the empirical distribution of the approximating point set. It can be shown [71] that herding points have a theoretical integration error rate which is at least comparable to Monte Carlo, and enjoy considerably improved performance in practice. One disadvantage of kernel herding is that it can only be performed for specific kernel-distribution pairs [71]. The second development, support points (SPs, [72]), aims to find a point set which minimizes a statistical potential measure called the *energy distance* [73]. The appeal of SPs lies in the difference-of-convex formulation of the energy distance, which allows for efficient generation of *optimal* representative points for any distribution F. However, by considering *only* goodness-of-fit on the full space $\mathcal{X}$, both herding and SPs suffer from poor projected goodness-of-fit. To this end, we present a new method for generating points which are representative of both F and its marginal distributions.

The chapter is organized as follows. Section 2 reviews kernel herding and SPs, and presents the new idea of PSPs. Section 3 presents a unifying framework for PSPs which

connects PGOF, the three effect principles in experimental design and a dimension-free error rate. Section 4 provides a Bayesian framework for PSPs, and reveals an interesting connection with MaxPro designs. Section 5 introduces two algorithms for efficiently generating PSPs, and proves their convergence to a stationary point set. Section 6 demonstrates the effectiveness of PSPs in several simulations, and Section 7 illustrates an important application of PSPs in optimally reducing Markov-chain Monte Carlo (MCMC) chains. Finally, Section 8 concludes with some directions for future research. All proofs of technical results are deferred to the Appendix for brevity.

2.2 Background and definition

2.2.1 Kernel herding and support points

We first review kernel herding, following [70] and [71]. Let $\gamma : \mathfrak{X} \times \mathfrak{X} \to \mathbb{R}$ be a symmetric, positive-definite (p.d.) kernel, with H_γ its reproducing kernel Hilbert space (RKHS). Herding generates the following sequential sampling scheme:

$$\mathbf{x}_{n+1} = \underset{\mathbf{x}\in\mathfrak{X}}{\text{Argmax}} \left\{ \mathbb{E}[\gamma(\mathbf{x}, \mathbf{Y})] - \frac{1}{n+1} \sum_{i=1}^{n} \gamma(\mathbf{x}, \mathbf{x}_i) \right\}, \quad \mathbf{Y} \sim F. \tag{2.1}$$

[71] provides a beautiful interpretation of (2.1) as the Frank-Wolfe steps for solving a corresponding convex program in the function space H_k. Borrowing results from convex programming, it can be proved [71] that for *finite-dimensional* kernels, the sequence of herding points $(\mathbf{x}_i)_{i=1}^{\infty}$ enjoy an improved integration error rate over the $\mathcal{O}(n^{-1/2})$ rate for Monte Carlo. While herding appears to provide better performance over MC in the more useful setting of *infinite-dimensional* kernels [70, 71, 74], this has yet to be shown theoretically. One caveat for herding is that a closed-form expression is needed for $\mathbb{E}[\gamma(\mathbf{x}, \mathbf{Y})]$, which is only possible for specific choices of k and F. However, given a sample batch from the desired distribution F, herding points can be generated using a Monte Carlo approximation of (2.1).

Unlike herding, the support points in [72] are instead motivated by statistical potentials and its uses in GOF testing. Define first the *energy distance* between two distribution functions (d.f.s) F and G:

Definition 7. *[73] Let* $\mathbf{X}, \mathbf{X}' \overset{i.i.d.}{\sim} G$ *and* $\mathbf{Y}, \mathbf{Y}' \overset{i.i.d.}{\sim} F$*, where* G *and* F *are d.f.s on* $\mathcal{X} \subseteq \mathbb{R}^p$ *with* $\mathbb{E}\|\mathbf{X}\|_2, \mathbb{E}\|\mathbf{Y}\|_2 < \infty$*. The energy distance between* F *and* G *is defined as:*

$$E(F, G) \equiv 2\mathbb{E}\|\mathbf{X} - \mathbf{Y}\|_2 - \mathbb{E}\|\mathbf{Y} - \mathbf{Y}'\|_2 - \mathbb{E}\|\mathbf{X} - \mathbf{X}'\|_2. \tag{2.2}$$

One appealing property of $E(F, G)$ is the so-called *metric property* (Prop. 1, [30]), which states that $E(F, G) \geq 0$, with equality holding if and only if F and G are the same distribution. Such a property is important for SPs (and the PSPs introduced later), because it ensures that point sets with low energy also provides a good representation of the desired distribution F.

Following [72], SPs are defined as the point set whose empirical distribution function (e.d.f.) F_n has minimal energy to F:

Definition 8. *[72] Let* $\mathbf{Y} \sim F$*, with* $\mathbb{E}\|\mathbf{Y}\|_2 < \infty$*. For fixed point set size* $n \in \mathbb{N}$*, the support points (SPs) of* F *are defined as:*

$$\underset{\mathbf{x}_1, \cdots, \mathbf{x}_n}{\text{Argmin}}\, E(F, F_n) = \underset{\mathbf{x}_1, \cdots, \mathbf{x}_n}{\text{Argmin}} \left\{ \frac{2}{n} \sum_{i=1}^{n} \mathbb{E}\|\mathbf{x}_i - \mathbf{Y}\|_2 - \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \|\mathbf{x}_i - \mathbf{x}_j\|_2 \right\}. \tag{2.3}$$

The formulation in (2.3) has several interesting connections to kernel herding. To see their similarities, set $\gamma(\mathbf{x}, \mathbf{y})$ as the negative L_2-norm $-\|\mathbf{x} - \mathbf{y}\|_2$, despite the latter being only conditionally positive-definite (c.p.d.). The updates in (2.1) can then be viewed as the sequential optimization of the $(n+1)$-th point $\mathbf{x}_{n+1}$ in (2.3), after fixing the first n points $\{\mathbf{x}_i\}_{i=1}^n$. In this sense, SPs exploit the underlying difference-of-convex (d.c.) structure in

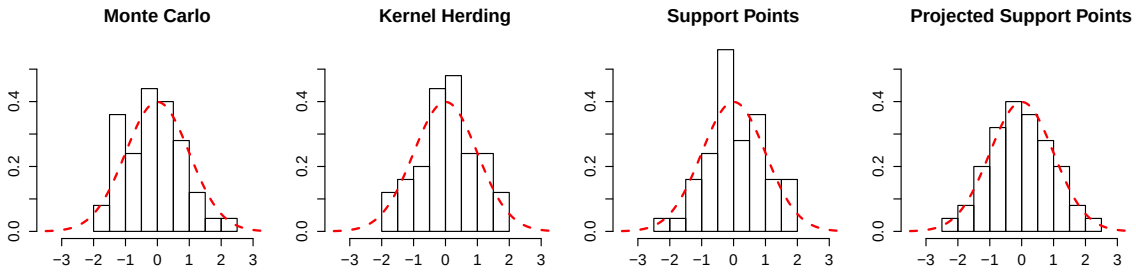

Figure 2.1: One-dimensional projections of $n = 50$ point sets for i.i.d. $N(0, 1)$ in $p = 10$ dimensions.

$E(F, F_n)$ to efficiently generate *optimal* sampling points for any distribution F, while herding can be viewed as a *greedy*, sequential minimization of the kernel analogue for $E(F, F_n)$ which may lead to suboptimal solutions. SPs also enjoy a theoretical improvement over Monte Carlo in integrating a large class of integrands [72].

However, by focusing only on the full sample space $\mathcal{X}$, both herding points and SPs can have poor goodness-of-fit for *marginal* distributions of F. To see this, Figure 2.1 shows the histograms for the 1-d projections of $n = 50$ points from Monte Carlo, herding, SPs and PSPs for the 10-d standard normal distribution, with the true marginal densities plotted in red. Herding points are generated using the isotropic Gaussian kernel $\gamma(\mathbf{x}, \mathbf{y}) = \exp\{-\theta\|\mathbf{x} - \mathbf{y}\|^2\}$ following [70], with $\theta = 1$. One surprising observation is that, after projection, both herding points and SPs provide a poorer fit of the 1-d marginal distribution compared to Monte Carlo! On the other hand, the proposed PSPs balance GOF for the full distribution F with GOF for its marginal distributions, thereby providing a much better projected fit. We formally introduce this trade-off below.

2.2.2 Projected support points

The key idea for PSPs is to use a flexible kernel family which can quantify the desired GOF trade-off between F and its marginal distributions. To this end, we assume the general

Gaussian kernel:

$$\gamma_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y}) \equiv \exp\left\{-\sum_{\varnothing\neq\mathbf{u}\subseteq[p]}\theta_{\mathbf{u}}\|\mathbf{x}_{\mathbf{u}}-\mathbf{y}_{\mathbf{u}}\|_2^2\right\}, \quad \theta_{\mathbf{u}}\geq 0, \quad [p]\equiv\{1,\cdots,p\} \tag{2.4}$$

for the remainder of this chapter. Assuming that a larger value of $\theta_{\mathbf{u}}$ encodes a greater importance on the GOF in projection $\mathbf{u}$ (a justification for this is provided later in Section 2.3), the kernel in (2.4) provides a general framework for quantifying the importance of each projected subspace of $\mathcal{X}$. For example, by setting $\theta_{\{1,2\}}=10$ and $\theta_{\mathbf{u}}=1$, $\mathbf{u}\neq\{1,2\}$, one places greater importance on the GOF for the marginal distribution in dimensions 1 and 2, and smaller (but equal) importance for all other projections. It is worth noting that, while the choice of a Gaussian kernel is made to facilitate theoretical analysis in Sections 2.3 and 6.2.2, the proposed methodology can easily be extended for any scale-parametrized kernel.

From (2.4), the *$\boldsymbol{\theta}$-weighted* and *π-expected discrepancies* can then be defined:

Definition 9. *Let $\mathbf{X},\mathbf{X}'\overset{i.i.d.}{\sim}F$ and $\mathbf{Y},\mathbf{Y}'\overset{i.i.d.}{\sim}G$, where F and G are d.f.s on $\mathcal{X}\subseteq\mathbb{R}^p$. For $\boldsymbol{\theta}=(\theta_{\mathbf{u}})_{\varnothing\neq\mathbf{u}\subseteq[p]}$, $\theta_{\mathbf{u}}\geq 0$, the $\boldsymbol{\theta}$-weighted discrepancy of F and G is:*

$$E_{\boldsymbol{\theta}}(F,G)\equiv\mathbb{E}\left\{\gamma_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X}')\right\}-2\mathbb{E}\left\{\gamma_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{Y})\right\}+\mathbb{E}\left\{\gamma_{\boldsymbol{\theta}}(\mathbf{Y},\mathbf{Y}')\right\}. \tag{2.5}$$

Letting $\boldsymbol{\theta}$ follow some proper prior π, the π-expected discrepancy is $E_{\boldsymbol{\theta}\sim\pi}(F,G)\equiv\mathbb{E}_{\boldsymbol{\theta}\sim\pi}\left[E_{\boldsymbol{\theta}}(F,G)\right]$.

The following proposition shows that the aforementioned metric property also holds for $E_{\boldsymbol{\theta}}(F,G)$ and $E_{\boldsymbol{\theta}\sim\pi}(F,G)$, which justifies both as valid goodness-of-fit criteria:

Theorem 9. *$E_{\boldsymbol{\theta}}(F,G)\geq 0$, with equality holding if and only if F=G. The same holds for $E_{\boldsymbol{\theta}\sim\pi}(F,G)$ under any proper prior π.*

The *projected support points* (PSPs) of F are then defined as follows:

Definition 10. *For fixed $\boldsymbol{\theta} = (\theta_{\mathbf{u}})_{\varnothing \neq \mathbf{u} \subseteq [p]}$, the $\boldsymbol{\theta}$-weighted PSPs are defined as:*

$$\underset{\mathbf{x}_1, \cdots, \mathbf{x}_n}{\text{Argmin}} \, E_{\boldsymbol{\theta}}(F, F_n) = \underset{\mathbf{x}_1, \cdots, \mathbf{x}_n}{\text{Argmin}} \left[-\frac{2}{n} \sum_{i=1}^{n} \mathbb{E}_{\mathbf{Y}} \left\{ \gamma_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{Y}) \right\} + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j) \right]. \tag{2.6}$$

If $\boldsymbol{\theta} \sim \pi$, the π-expected PSPs are defined as:

$$\underset{\mathbf{x}_1, \cdots, \mathbf{x}_n}{\text{Argmin}} \, E_{\boldsymbol{\theta} \sim \pi}(F, F_n) = \underset{\mathbf{x}_1, \cdots, \mathbf{x}_n}{\text{Argmin}} \left[-\frac{2}{n} \sum_{i=1}^{n} \mathbb{E}_{\mathbf{Y}, \boldsymbol{\theta} \sim \pi} \left\{ \gamma_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{Y}) \right\} + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbb{E}_{\boldsymbol{\theta} \sim \pi} \left\{ \gamma_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j) \right\} \right]. \tag{2.7}$$

Unfortunately, by parametrizing the importance of all possible projections, the kernel in (2.4) becomes too general to use for both theoretical analysis and practical implementation. We therefore consider the following two simplifications on $(\theta_{\mathbf{u}})_{\varnothing \neq \mathbf{u} \subseteq [p]}$:

- *Anisotropic:*

$$\theta_{\mathbf{u}} = \theta_l \text{ for } \mathbf{u} = \{l\}, \; l = 1, \cdots, p, \quad \text{and} \quad \theta_{\mathbf{u}} = 0 \text{ otherwise.} \tag{2.8}$$

  Under this setting, (2.4) reduces to the anisotropic Gaussian kernel.

- *Product-and-order (POD):*

$$\theta_{\mathbf{u}} = \Gamma_{|\mathbf{u}|} \prod_{l \in \mathbf{u}} \theta_l, \tag{2.9}$$

  where $(\theta_l)_{l=1}^{p}$ and $(\Gamma_{|\mathbf{u}|})_{|\mathbf{u}|=1}^{\infty}$ are known as *product* and *order* weights, respectively. POD weights were first introduced by [75] for analyzing partial differential equations, and we show later that such weights provide a concise quantification of the three effect principles in experimental design. Note that the earlier anisotropic setting can be recovered by setting $\Gamma_{|\mathbf{u}|} = 0$ for $|\mathbf{u}| > 1$.

We make use of the simpler anisotropic setting for theoretical analysis in Sections 2.3 and 6.2.2, and the POD setting for practical implementation from Section 2.4.2 onwards.

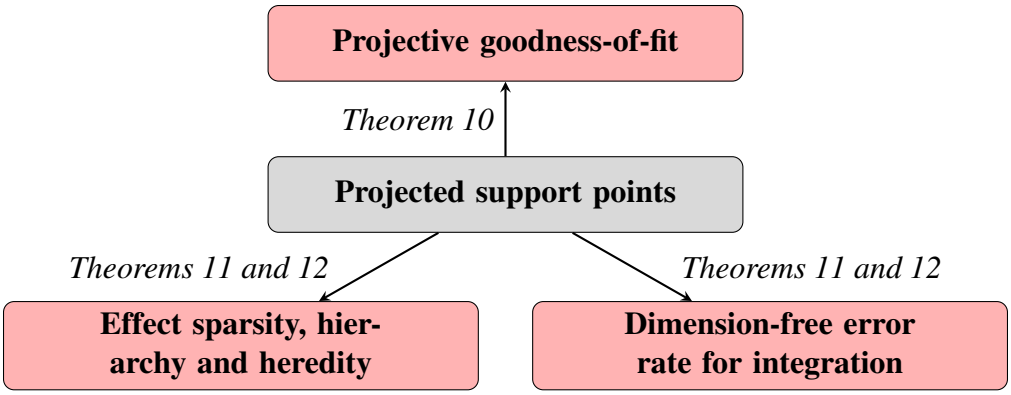

Figure 2.2: The triangle connection for PSPs.

2.3 Theoretical framework

Using the anisotropic setting, a theoretical framework for PSPs is presented here, connecting the ideas of (a) projective goodness-of-fit, (b) the three effect principles in experimental design and (c) a dimension-free integration error rate. This unifies recent developments in experimental design and QMC, and extends them in the context of integration under non-uniform distributions. For reference, Figure 2.2 shows a visualization of this "triangle" connection, along with their corresponding theorems. This section concludes with a result demonstrating the theoretical improvement of PSPs over MC for fixed dimension p.

2.3.1 Triangle connection

We first provide some insight on the effect of $\boldsymbol{\theta}$ on projected goodness-of-fit:

Theorem 10. *Fix $\boldsymbol{\theta} = (\theta_l)_{l=1}^p \in \mathbb{R}_+^p$, and let $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$ be the SPs of F, with F_n its e.d.f. and $\mathbf{x}^{(l)} = (x_{il})_{i=1}^n$ the l-th dimensional points of $\mathcal{D}$. Under two approximations:*

1. $\gamma_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}) \approx \bar{\gamma}$ *for any* $\mathbf{x} \in \mathfrak{X} \setminus \mathbf{x}_i$,

2. $\int_{\mathfrak{X}} \int_{\mathfrak{X}} |x_l - x_l'| \, dF_n(\mathbf{x}) dF_n(\mathbf{x}') \approx \int_{\mathfrak{X}} \int_{\mathfrak{X}} |x_l - x_l'| \, dF(\mathbf{x}) dF(\mathbf{x}')$,

and letting $E(F_l, F_{l,n})$ denote the energy distance between the l-th dimensional marginal

distributions of F and F_n, it follows that:

$$\|\nabla_{\mathbf{x}^{(l)}} E_{\boldsymbol{\theta}}(F, F_n)\|_1 \lesssim 4\bar{\gamma}\theta_l E(F_l, F_{l,n}). \tag{2.10}$$

The first approximation is justifiable when $\theta_l \to \infty$ (i.e., greater importance is placed on PGOF in the l-th dimension), and the second assumption is justifiable when $n \to \infty$, because SPs converge in distribution to F [72].

Theorem 10 reveals an important connection between $\boldsymbol{\theta}$ and PGOF. Recall that the negative gradient of an objective function indicates the direction of greatest descent, with the norm of this gradient representing the magnitude of this descent. In this light, Theorem 10 shows that, for the current point set $\{\mathbf{x}_i\}_{i=1}^n$, the reduction in $E_{\boldsymbol{\theta}}(F, F_n)$ achievable by adjusting the l-th dimensional points $\mathbf{x}^{(l)}$ is largely dominated by (a) the scale parameter θ_l and (b) the l-th dimensional energy $E(F_l, F_{l,n})$. This can be interpreted in two ways. First, assuming equal energies $E(F_l, F_{l,n})$ over all dimensions l, a larger value of θ_l encourages greater movement for the l-th dimensional points $\mathbf{x}^{(l)}$ in the minimization of $E_{\boldsymbol{\theta}}(F, F_n)$. Second, because the goal in optimization is to obtain an optimal point set with gradient norm $\|\nabla E_{\boldsymbol{\theta}}(F, F_n)\|_1$ equal to 0, a key ingredient for reducing this norm is to reduce the l-th dimensional energy $E(F_l, F_{l,n})$, which corresponds to improving PGOF in the l-th dimension. In other words, from an optimization perspective, *a larger value of θ_l imposes a greater emphasis on the l-dimensional PGOF for PSPs*.

This interpretation also sheds light on the poor PGOF of herding points and SPs in Figure 2.1. By assuming a priori the same scale parameters for kernel k, the resulting formulation assumes all dimensions are equally important with certainty. This then encourages GOF only for the full distribution F, and ignores PGOF for its marginal distributions. To foreshadow, we address this problem by assigning an appropriate prior distribution on $\boldsymbol{\theta}$.

Next, we explore the connection between the three design principles in experimental

design and recent QMC work on dimension-free error rates. For clarity, a brief overview is provided below on both topics. In experimental design, the principles of effect sparsity, hierarchy and (strong) heredity quantify, respectively, the prior beliefs that a response surface is dominated by a small number of effects, with lower-order effects accounting for most of the response variability, and higher-order effects active only when all its lower-order effects are present. These principles are highly useful for selecting appropriate models from experimental data, because the number of runs is often limited and the effects of interest fully-aliased. As we show below, this is inherently related to the idea of a dimension-free rate in QMC, where the integration error rate does not grow in dimension p. Such a rate provides the theoretical basis for applying QMC methods to high-dimensional integration problems, and recent results (see [65], [37] and [16]) show that under tractability conditions on the integrand, certain *randomized* QMC methods (e.g., the randomly shifted lattice rules in [21, 22]) can indeed achieve this rate for $F = U[0,1]^p$. In the current work, the RKHS for $\gamma_{\boldsymbol{\theta}}$ reveals an insightful connection between these conditions and the three effect principles, which can then be used to demonstrate a dimension-free rate for PSPs on non-uniform distributions.

We first provide an explicit construction of the RKHS for $\gamma_{\boldsymbol{\theta}}$:

Theorem 11. *Let $H_{\gamma,\boldsymbol{\theta}}$ be the RKHS for the kernel $\gamma_{\boldsymbol{\theta}}$. Then:*

$$H_{\gamma,\boldsymbol{\theta}} = \left\{ g : \mathbb{R}^p \to \mathbb{R} \;\middle|\; \exists \{w_{\boldsymbol{\alpha}}\}_{|\boldsymbol{\alpha}|=0}^{\infty} \text{ s.t. } g(\mathbf{x}) = \exp(-\|\mathbf{x}\|_{\boldsymbol{\theta}}^2) \sum_{|\boldsymbol{\alpha}|=0}^{\infty} w_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}},\ \|g\|_{\gamma,\boldsymbol{\theta}} < \infty \right\}, \tag{2.11}$$

with inner product given by:

$$\langle f, g \rangle_{\gamma,\boldsymbol{\theta}} = \sum_{k=0}^{\infty} \frac{k!}{2^k} \sum_{|\boldsymbol{\alpha}|=k} \frac{v_{\boldsymbol{\alpha}} w_{\boldsymbol{\alpha}}}{C_{\boldsymbol{\alpha}}^k \boldsymbol{\theta}^{\boldsymbol{\alpha}}}, \quad f(\mathbf{x}) = \exp(-\|\mathbf{x}\|_{\theta}^2) \sum_{|\boldsymbol{\alpha}|=0}^{\infty} v_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}. \tag{2.12}$$

Here, $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_p)$ with $|\boldsymbol{\alpha}| = \sum_{l=1}^p \alpha_l$, $\{w_{\boldsymbol{\alpha}}\}_{|\boldsymbol{\alpha}|=0}^{\infty}, \{v_{\boldsymbol{\alpha}}\}_{|\boldsymbol{\alpha}|=0}^{\infty} \subseteq \mathbb{R}$ are coefficients, $\mathbf{x}^{\boldsymbol{\alpha}} = \prod_{l=1}^p x_l^{\alpha_l}$ (similarly for $\boldsymbol{\theta}^{\boldsymbol{\alpha}}$) and $C_{\boldsymbol{\alpha}}^k = k!/(\alpha_1! \cdots \alpha_p!)$ is the multinomial coefficient.

In words, the RKHS $H_{\gamma,\boldsymbol{\theta}}$ consists of all integrands spanned by $\{\exp(-\|\mathbf{x}\|^2_{\boldsymbol{\theta}})\mathbf{x}^{\boldsymbol{\alpha}}\}_{|\boldsymbol{\alpha}|=0}^{\infty}$, with corresponding coefficients $\{w_{\boldsymbol{\alpha}}\}_{|\boldsymbol{\alpha}|=0}^{\infty}$ (we call these ANOVA coefficients from here on). The explicit construction in (2.11) is quite appealing, because it gives an interpretable, ANOVA-like decomposition of the function space $H_{\gamma,\boldsymbol{\theta}}$. For example, for a function $g \in H_{\gamma,\boldsymbol{\theta}}$, a larger ANOVA coefficient $w_{\boldsymbol{\alpha}}$ indicates a greater importance of the basis term $\exp(-\|\mathbf{x}\|^2_{\boldsymbol{\theta}})\mathbf{x}^{\boldsymbol{\alpha}}$ in g.

Using this RKHS along with a simple application of Cauchy-Schwarz, an upper bound can be obtained which connects integration error with the $\boldsymbol{\theta}$-weighted discrepancy:

Lemma 3. *Let F_n be the e.d.f. of an approximating point set for F. For any integrand $g \in H_{\gamma,\boldsymbol{\theta}}$, its integration error can be bounded by:*

$$I(g; F, F_n) \equiv \left| \int_{\mathfrak{X}} g(\mathbf{x})\, d[F - F_n](\mathbf{x}) \right| \leq \|g\|_{\gamma,\boldsymbol{\theta}} \sqrt{E_\theta(F, F_n)}. \tag{2.13}$$

The following theorem then establishes a dimension-free convergence rate for PSPs:

Theorem 12. *Let F_n be the e.d.f. of the $\boldsymbol{\theta}$-weighted PSPs, and let $g \in H_{\gamma,\boldsymbol{\theta}}$. Further assume the ANOVA coefficients for g are of the POD-like form $w_{\boldsymbol{\alpha}} = \mathrm{T}_{|\boldsymbol{\alpha}|} \prod_{l=1}^{p} w_l^{\alpha_l}$. If:*

$$\mathrm{T}_{|\boldsymbol{\alpha}|} = \mathcal{O}\left\{ p^{-1/4} \left(|\boldsymbol{\alpha}|!\right)^{-1/2} \right\} \quad \textit{and} \quad \sum_{l=1}^{\infty} w_l^4 / \theta_l^2 < 4, \tag{2.14}$$

then $I(g; F, F_n) \leq C/\sqrt{n}$ for some constant $C > 0$ not depending on p.

The two conditions in (6.5) reveal important insights on the connection between $\boldsymbol{\theta}$ and a dimension-free rate. Consider first the POD-like form of $w_{\boldsymbol{\alpha}}$ (a similar framework also arises in experimental design, see [76]). The claim is that (a) the *product* weights $(w_l)_{l=1}^{p}$ control *effect heredity*, and (b) the *order* weights $(\mathrm{T}_{|\mathbf{u}|})_{|\mathbf{u}|=1}^{\infty}$ dictate *effect hierarchy*. To see this, suppose the ANOVA coefficient $w_{\boldsymbol{\alpha}}$ is large, thereby indicating a significant interaction

effect for g in the subspace of order $\boldsymbol{\alpha}$. From the product structure $\prod_{l=1}^{p} w_l^{\alpha_l}$ in $w_{\boldsymbol{\alpha}}$, this suggests all first-order ANOVA coefficients $\{w_l : \alpha_l > 0\}$ are also significant, which is precisely the principle of (strong) effect heredity. Moreover, when $(\mathrm{T}_{|\boldsymbol{\alpha}|})_{|\boldsymbol{\alpha}|=0}^{\infty}$ forms a strictly decreasing sequence, the order structure in $w_{\boldsymbol{\alpha}}$ forces all higher-order ANOVA effects to be less significant than lower-order effects, which is precisely effect hierarchy.

In this light, the two conditions in Theorem 12 can be interpreted in terms of the effect principles. The first condition $\mathrm{T}_{|\boldsymbol{\alpha}|} = \mathcal{O}\left\{p^{-1/4}\left(|\boldsymbol{\alpha}|!\right)^{-1/2}\right\}$ suggests an *effect hierarchy decay* of $\mathcal{O}\left\{p^{-1/4}\left(|\boldsymbol{\alpha}|!\right)^{-1/2}\right\}$ is required for a dimension-free convergence rate. This rate is quite appealing intuitively, because by effect sparsity and hierarchy, one expects the order weights to decay rapidly in dimension p and order $|\mathbf{u}|$, respectively. Indeed, a similar factorial decay of order weights also arises when proving the dimension-free convergence rate of component-by-component lattice rules (pg. 76 of [16]), which is quite fascinating and draws a parallel between the standard $U[0,1]^p$ setting of QMC and the non-uniform setting here. The second condition $\sum_{l=1}^{\infty} w_l^4/\theta_l^2 < 4$ can be viewed as an expression of *effect sparsity*. To see this, suppose the simple case where $\theta_l = 1$ for all dimensions l. The resulting constraint $\sum_{l=1}^{\infty} w_l^4 < 4$ limits the number of active dimensions in the high-dimensional setting of $p \to \infty$, which is precisely effect sparsity. Moreover, when too many factors are active and this condition is violated, the scale parameters θ_l can be used to force $\sum_{l=1}^{\infty} w_l^4/\theta_l^2 < 4$. Put another way, a highly influential dimension l for integrand g can be counteracted by setting a sufficiently large value of θ_l. As mentioned earlier in the context of optimization, this places greater emphasis on PGOF in the l-th dimension, which is as expected.

2.3.2 Convergence rate for fixed p

While Theorem 12 addresses the conditions for avoiding the curse-of-dimensionality, the asserted $\mathcal{O}(n^{-1/2})$ rate only makes PSPs comparable to MC for fixed dimension p. To this end, the following theorem borrows techniques from [72] to demonstrate the theoretical

improvement of PSPs over Monte Carlo for fixed p.

Theorem 13. *For some active set $\mathscr{A} \subseteq [p]$, suppose $g \in H_{\gamma,\boldsymbol{\theta}}$, with $\theta_l > 0$ for $l \in \mathscr{A}$ and $\theta_l = 0$ otherwise. Let $\mathfrak{X} \subseteq \mathbb{R}^p$ be measurable with positive Lebesgue measure, let F_n be the e.d.f. for the $\boldsymbol{\theta}$-weighted PSPs of F, and suppose F satisfies the mild moment condition:*

$$\exists \beta > 0, C \geq 0 \text{ s.t. } \limsup_{r\to\infty} r^{\beta} \int_{\mathfrak{X}\setminus B_r(\mathbf{y})} \mathbb{E}[\gamma_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{Y})]\, dF(\mathbf{x}) \leq C,\ \mathbf{Y} \sim F,\ \text{for all } y \in \mathfrak{X}. \tag{2.15}$$

Then, with $\gamma = \beta/(\beta+1)$, it follows that for any $\nu \in (0, \gamma)$:

$$I(g; F, F_n) \leq \mathcal{O}\left\{ \|g\|_{\gamma,\boldsymbol{\theta}}\ n^{-1/2} (\log n)^{-(\gamma-\nu)/(2|\mathscr{A}|)} \right\}, \tag{2.16}$$

where constants may depend on p and ν.

Two important insights can be made from this theorem. First, when the integrand g is active in all dimensions (i.e., $\mathscr{A} = [p]$) and F is not too heavy-tailed, PSPs enjoy a faster error convergence to MC by at least the log-factor $(\log n)^{-1/(2p)}$. While this is indeed an improvement, simulation studies in Section 2.6 suggest a quicker rate for PSPs in both low and high dimensions, and more work is needed to establish this theoretically. Second, when g is active only in a subset of dimensions (i.e., $\mathscr{A} \subsetneq [p]$) and the proposed PSPs are constructed on these active dimensions, this log-factor improves to $(\log n)^{-1/(2|\mathscr{A}|)}$, where $|\mathscr{A}| < p$. This generalizes the result in [72] for low-dimensional integrands in high-dimensional spaces. In practice, the active dimensions for g are not known in advance, and need to be learned using some form of adaptive sampling. A Bayesian formulation for $\boldsymbol{\theta}$ is well-suited for such a task, and we briefly mention in Section 6.5.2 a posterior update scheme for $\boldsymbol{\theta}$ which can iteratively select active dimensions and generate integration points.

2.4 Prior specification on POD weights

Having established the theoretical framework of PSPs, we now examine several specifications for the product weights $(\theta_l)_{l=1}^p$ and order weights $(\Gamma_{\mathbf{u}})_{|\mathbf{u}|=1}^{\infty}$ under the POD setting (2.9). We first study the choice of prior π for product weights, revealing a connection to recent developments in experimental design, then conclude with a brief discussion on order weights.

2.4.1 Product weights and the projection kernel

Consider the following independent Gamma prior specification for $(\theta_l)_{l=1}^p$:

$$\theta_l \overset{i.i.d.}{\sim} Gamma(\nu, \lambda), \quad \text{i.e.,} \quad \tilde{\pi}(\theta) = \prod_{l=1}^{p} \left\{ \frac{\lambda^\nu}{\Gamma(\nu)} \theta_l^{\nu-1} \exp\left(-\lambda\theta_l\right) \right\}. \tag{2.17}$$

The prior $\tilde{\pi}$ provides two appealing properties for PSPs. First, the i.i.d. framework reflects the belief that no subset of dimensions is favored over another, an intuitive assumption to make a priori. Second, the choice of Gamma priors allows for a closed-form expression for the expected discrepancy $E_{\boldsymbol{\theta}\sim\tilde{\pi}}(F, F_n)$. This closed-form is valuable for two reasons: (a) it provides insight on the effect of prior hyperparameters ν and λ on PSPs, and (b) reveals a connection with the MaxPro designs in [69]. While the following discussion entertains only the i.i.d. Gamma prior, the proposed algorithm in Section 6.5.2 can be used for any prior π which can be efficiently sampled.

We first provide the closed-form expected discrepancy under $\tilde{\pi}$:

Proposition 2. *Assume the anisotropic setting in* (2.8)*. Under the prior in* (6.9)*:*

$$\frac{E_{\boldsymbol{\theta}\sim\tilde{\pi}}(F, F_n)}{\lambda^\nu} = \int_{\mathfrak{X}} \int_{\mathfrak{X}} \left\{ \prod_{l=1}^{p} \frac{1}{(x_l - y_l)^2 + \lambda} \right\}^{\nu} d[F - F_n](\mathbf{x})\, d[F - F_n](\mathbf{y}). \tag{2.18}$$

We call $\tilde{k}_{\nu,\lambda} = \{\prod_{l=1}^{p} (x_l - y_l)^2 + \lambda\}^{-\nu}$ the *projection kernel* for the rest of the chapter,

because it provides a way to quantify the projected similarities between two points. To see this, set $\nu = 1$ and $\lambda = 0.01$, and consider the three points $\mathbf{x} = (0, 0)$, $\mathbf{y} = (\sqrt{2}, 0)$ and $\mathbf{z} = (1, 1)$. While the Euclidean distance is the same between $\mathbf{x}$ and $\mathbf{y}$ and between $\mathbf{x}$ and $\mathbf{z}$, the projected kernel gives a much higher similarity measure for the first pair of points ($\tilde{k}_{\nu,\lambda}(\mathbf{x}, \mathbf{y}) = 4930$) than the second pair ($\tilde{k}_{\nu,\lambda}(\mathbf{x}, \mathbf{z}) = 0.96$). An inspection of $\tilde{k}_{\nu,\lambda}$ reveals why this is the case. Whenever two points are close in some dimension l, the denominator term $(x_l - y_l)^2 + \lambda$ becomes small, which results in a large value for $\tilde{k}_{\nu,\lambda}$. Such a kernel therefore assigns *larger* values for point pairs which are *close* in some coordinate projection. Note that the projection kernel can be viewed as the product kernel of the generalized inverse multiquadric kernel [77], the latter being a popular tool in image classification [78].

Proposition 2 also provides some insight on the effect of hyperparameters ν and γ on the resulting PSPs. Consider first the shape parameter ν. From (2.18), a larger ν places greater emphasis on the GOF in regions with large $\tilde{k}_{\nu,\lambda}$ (i.e., regions with high projected similarities), and a smaller value of ν places greater emphasis on the GOF over the full space $\mathcal{X}$. In other words, ν partially controls the trade-off between the GOF of the full distribution F and PGOF of its marginal distributions. Next, for the rate parameter λ, observe that $\lim_{\lambda \to 0^+} \lambda^{\nu} E_{\boldsymbol{\theta} \sim \tilde{\pi}}(F, F_n) \approx \int_{\mathcal{X}} \int_{\mathcal{X}} \left(\max_{l=1,\cdots,p} |x_l - y_l|^{-2\nu}\right) \, d[F - F_n](\mathbf{x}) \, d[F - F_n](\mathbf{y})$, which suggest that smaller values of λ improve the worst-case PGOF of the 1-d marginal distributions. This again comes at a trade-off, because $\lambda \to 0^+$ introduces asymptotic behavior for $\tilde{k}(\mathbf{x}, \mathbf{y})$ whenever $\mathbf{x}$ and $\mathbf{y}$ are close after projection, which in turn causes numerical instabilities in optimization. We found the choice of $(\nu, \lambda) = (0.1, 1)$ to work quite well for the simulations in Section 2.6 and the application in Section 2.7.2.

The closed-form discrepancy $E_{\boldsymbol{\theta} \sim \tilde{\pi}}(F, F_n)$ also reveals an illuminating connection between PSPs and the MaxPro designs in [69]. The latter is a popular design choice for computer experiments, because it has good space-filling properties (see, e.g., [49]) when projected onto any coordinate subspace of the unit hypercube $[0, 1]^p$. Setting $\nu = 1$ and

expanding the terms in (2.18), $E_{\boldsymbol{\theta}\sim\tilde{\pi}}(F, F_n)$ can be written as:

$$\mathbb{E}\left(\prod_{l=1}^{p}\frac{1}{|Y_l - Y_l'|^2+\lambda}\right) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\prod_{l=1}^{p}\frac{1}{|x_{il}-x_{jl}|^2+\lambda}\right) - \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}\left(\prod_{l=1}^{p}\frac{1}{|x_{il}-Y_l|^2+\lambda}\right), \tag{2.19}$$

where $Y_l, Y_l' \overset{i.i.d.}{\sim} F_l$. The criterion to minimize in MaxPro designs is precisely the limit of the middle term in (2.19) as $\lambda \to 0^+$! In this sense, PSPs generalize the MaxPro designs in two important ways. First, in the uniform setting of $F = U[0,1]^p$, the addition of an adjustment factor $-(2/n)\sum_{i=1}^{n}\mathbb{E}\left\{\prod_{l=1}^{p}(|x_{il}-Y_l|^2+\lambda)^{-1}\right\}$ to the MaxPro criterion allows the resulting point set to have good *uniformity* after projection. Second, PSPs generalize the desired projective property of MaxPro designs from the uniform hypercube $U[0,1]^p$ to non-uniform distributions F.

2.4.2 Order weights

Lastly, we provide some insight on the choice of order weights $(\Gamma_{|\mathbf{u}|})_{|\mathbf{u}|=1}^{\infty}$ in the POD framework (2.9). By effect hierarchy, lower-order terms should be more significant than higher-order terms, so $\Gamma_{|\mathbf{u}|}$ should form a decreasing sequence in $|\mathbf{u}|$. We consider two settings of order weights here: (a) the anisotropic setting: $\Gamma_{|\mathbf{u}|} = p^{-1/4}$ if $|\mathbf{u}| = 1$, $\Gamma_{|\mathbf{u}|} = 0$ otherwise, and (b) the factorial decay setting $\Gamma_{|\mathbf{u}|} = p^{-1/4}(|\mathbf{u}|!)^{-1/2}$. The first is appropriate when the desired integrand g is dominated by first-order effects, whereas the second is appropriate when g is largely composed of lower-order (but not necessarily first-order) effects. The specific decay $p^{-1/4}(|\mathbf{u}|!)^{-1/2}$ is motivated by the conditions for a dimension-free rate in Theorem 12. Both choices of weights are tested in the simulations in Section 2.6.

2.5 Algorithm

We present here two algorithms, `psp.ccp` and `psp.sccp`, which can efficiently generate PSPs using a combination of the convex-concave procedure and resampling. The first should be used when only a single batch from F and π are available, and the second should be used when multiple sample batches from F and π can be efficiently generated. These

Algorithm 3

`psp.ccp`: PSPs using one sample batch

- Warm-start the initial point set $\mathscr{D}^{[0]} = \{\mathbf{x}_i^{[0]}\}_{i=1}^n$ using SPs.
- Set $l = 0$, **repeat** until convergence of $\mathscr{D}^{[l]}$:
  - **For** $i = 1, \cdots, n$:
    - Set $\mathbf{x}_i^{[l+1]} \leftarrow M_i(\mathbf{x}_i^{[l]}; \mathscr{Y}, \vartheta, \mathscr{D}_{-i}^{[l]})$, with M_i defined in (2.24).
    - Update $\mathscr{D}_i^{[l]} \leftarrow \mathbf{x}_i^{[l+1]}$.
  - Update $\mathscr{D}^{[l+1]} \leftarrow \{\mathbf{x}_i^{[l+1]}\}_{i=1}^n$, set $l \leftarrow l + 1$.
- Return the converged point set $\mathscr{D}^{[\infty]}$.

Algorithm 4

`psp.sccp`: PSPs using multiple sample batches

- Warm-start the initial point set $\mathscr{D}^{[0]} = \{\mathbf{x}_i^{[0]}\}_{i=1}^n$ using SPs.
- Set $l = 0$, **repeat** until convergence of $\mathscr{D}^{[l]}$:
  - **For** $i = 1, \cdots, n$:
    - Resample $\mathscr{Y}^{[l]} \overset{i.i.d.}{\sim} F$ and $\vartheta^{[l]} \overset{i.i.d.}{\sim} \pi$.
    - Set $\mathbf{x}_i^{[l+1]} \leftarrow M_i(\mathbf{x}_i^{[l]}; \mathscr{Y}^{[l]}, \vartheta^{[l]}, \mathscr{D}_{-i}^{[l]})$, with M_i defined in (2.24).
    - Update $\mathscr{D}_i^{[l]} \leftarrow \mathbf{x}_i^{[l+1]}$.
  - Update $\mathscr{D}^{[l+1]} \leftarrow \{\mathbf{x}_i^{[l+1]}\}_{i=1}^n$, and set $l \leftarrow l + 1$.
- Return the converged point set $\mathscr{D}^{[\infty]}$.

can be seen as extensions of the algorithms `sp.ccp` and `sp.sccp` in [72] to the general Gaussian kernel setting (2.4) under the POD weights (2.9), with a suitable prior assigned to $\boldsymbol{\theta}$. We first provide a brief description of the two algorithms, then discuss theoretical details on convergence and running times.

2.5.1 Algorithm statement

We first provide the steps for `psp.ccp`, then introduce `psp.sccp` as an improvement on `psp.ccp` when multiple sample batches from F and π are available. Suppose the sample batches $\mathscr{Y} = \{\mathbf{y}_m\}_{m=1}^N$ and $\vartheta = \{\boldsymbol{\theta}_r\}_{r=1}^R$ are obtained from F and π, respectively. The

Monte Carlo approximation of the PSP formulation (2.7) becomes:

$$\underset{\mathbf{x}_1,\cdots,\mathbf{x}_n}{\text{Argmin}}\ \hat{E}(\{\mathbf{x}_i\};\mathscr{Y},\vartheta) \equiv -\frac{2}{nNR}\sum_{i=1}^{n}\sum_{m=1}^{N}\sum_{r=1}^{R}\gamma_{\boldsymbol{\theta}_r}(\mathbf{x}_i,\mathbf{y}_m) + \frac{1}{n^2R}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{R}\gamma_{\boldsymbol{\theta}_r}(\mathbf{x}_i,\mathbf{x}_j). \tag{2.20}$$

Note that, in the specific setting of anisotropic $\boldsymbol{\theta}$ with i.i.d. Gamma priors, one can forgo the resampling of $\boldsymbol{\theta}$ by using the closed-form projection kernel $\tilde{k}_{\nu,\lambda}$ from Proposition 2. However, even for such a setting, we found that the resampling of $\boldsymbol{\theta}$ provides a more numerically stable algorithm, because $\tilde{k}_{\nu,\lambda}$ can be ill-conditioned when design points are close in some projected subspace.

Recall that a key advantage of SPs is that the energy distance can be viewed as a d.c. program, which allows for efficient optimization. Indeed, seeing how the kernel $\gamma_{\boldsymbol{\theta}}(\cdot,\cdot)$ is concave near its origin, a similar technique can be used to efficiently optimize (6.1). By exploiting this structure, we found that the proposed algorithm performs considerably quicker than black-box gradient-descent methods (see [79]). This is not surprising, because the latter requires multiple evaluations of both objective and gradient functions to perform one update, which can be computationally expensive for the large-scale optimization at hand.

To exploit this structure, consider the blockwise optimization of $E_{\boldsymbol{\theta}\sim\pi}(F,F_n)$ for design point $\mathbf{x}_i$, fixing the remaining $n-1$ points $\mathscr{D}_{-i} \equiv \{\mathbf{x}_j\}_{j\neq i}$:

$$\underset{\mathbf{x}}{\text{Argmin}}\ \hat{E}_i(\mathbf{x};\mathscr{Y},\vartheta,\mathscr{D}_{-i}) \equiv -\frac{1}{NR}\sum_{m=1}^{N}\sum_{r=1}^{R}\gamma_{\boldsymbol{\theta}_r}(\mathbf{x},\mathbf{y}_m) + \frac{1}{nR}\sum_{\substack{j=1\\ j\neq i}}^{n}\sum_{r=1}^{R}\gamma_{\boldsymbol{\theta}_r}(\mathbf{x},\mathbf{x}_j). \tag{2.21}$$

The strategy here is to update $\mathbf{x}_i$ by solving (2.21), then repeat this optimization cyclically for each of the remaining $n-1$ points until the point set converges. This blockwise technique is known as *blockwise coordinate descent* (BCD, [80]), and is widely employed within many optimization algorithms in machine learning and statistics (see, e.g., [81] and [82]). BCD is adept at solving problems with appealing structure in its blockwise formula-

tion, as is the case here. Specifically, the problem in (2.21) can be efficiently solved using a d.c. optimization technique called the *concave-convex procedure* (CCP, see [42]), with details provided in the following section.

Following the above description, Algorithm 3 outlines the steps for `psp.ccp`. Here, the closed-form updates M_i within the for-loop perform one step of CCP, whereas the for-loop itself implements the cyclic BCD procedure. When multiple sample batches can be obtained from either F or π, `psp.ccp` can be further improved by incorporating such samples into the algorithm. This motivates the second algorithm, `psp.sccp`, with steps outlined in Algorithm 4. The key difference is that the approximating samples $\mathcal{Y}$ and ϑ are *resampled* before each iterative update in `psp.sccp`, which allows for convergence to a stationary solution of the desired problem (2.7).

Finally, recall the result in Theorem 13, which shows a quicker asymptotic rate when active dimensions can be identified through adaptive sampling. The algorithms proposed here provide an appropriate framework for such a scheme, in that the Bayesian modeling of $\boldsymbol{\theta}$ allows it to be updated via posterior sampling (say, using a Gaussian process model on g, see [83, 84]), and the blockwise optimization in `psp.ccp` allows for efficient generation of sequential points which incorporate this posterior learning of $\boldsymbol{\theta}$. Given the scope of the current chapter, we defer this topic to future work.

2.5.2 Algorithm correctedness

Before establishing the theoretical correctedness of `psp.ccp` and `psp.sccp`, we first provide a brief overview of a more general version of CCP called *majorization-minimization* (MM), following [43]. Consider first the definition of a *majorization function*:

Definition 11. *Let $f : \mathbb{R}^s \to \mathbb{R}$ be an objective to be minimized. A function $h(\mathbf{z}|\mathbf{z}')$ majorizes $f(\mathbf{z})$ at $\mathbf{z}' \in \mathbb{R}^s$ if $h(\mathbf{z}'|\mathbf{z}') = f(\mathbf{z}')$ and $h(\mathbf{z}|\mathbf{z}') \geq f(\mathbf{z})$ for all $\mathbf{z} \neq \mathbf{z}'$.*

Starting at an initial point $\mathbf{z}^{[0]} \in \mathbb{R}^s$, Mm first minimizes the majorization function $h(\cdot|\mathbf{z}^{[0]})$ in place of the true objective f, then iterates the updates $\mathbf{z}^{[l+1]} \leftarrow \operatorname{argmin}_{\mathbf{z}} h(\mathbf{z}|\mathbf{z}^{[l]})$ until

convergence. The solution sequence from such an update scheme can be shown to have the *descent property* $f(\mathbf{z}^{[l+1]}) \leq h(\mathbf{z}^{[l+1]}|\mathbf{z}^{[l]}) \leq h(\mathbf{z}^{[l]}|\mathbf{z}^{[l]}) = f(\mathbf{z}^{[l]})$, which ensures the sequence of objective values $(f(\mathbf{z}^{[l]}))_{l=1}^{\infty}$ is monotonically decreasing.

The key to computational efficiency for MM is to judiciously choose a surrogate g which not only majorizes f, but also admits an easy-to-compute closed-form minimizer. We establish such a surrogate majorizer for the blockwise objective $\hat{E}_i$ in (2.21) by showing $\gamma_{\boldsymbol{\theta}}$ can be majorized and minorized by appropriately-chosen paraboloids:

Lemma 4. *Let $\gamma_{\boldsymbol{\theta}}(\mathbf{z})$ be the shift-invariant form of the Gaussian kernel* (2.4) *under the POD weights* (2.9)*. For any $\mathbf{z}' \in \mathbb{R}^p$, $\gamma_{\boldsymbol{\theta}}(\mathbf{z})$ is majorized at $\mathbf{z}'$ by the paraboloid:*

$$\bar{Q}_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{z}') \equiv \gamma_{\boldsymbol{\theta}}(\mathbf{z}') - 2[\gamma_{\boldsymbol{\theta}}(\mathbf{z}')\Omega_{\boldsymbol{\theta}}\mathbf{z}']^T(\mathbf{z}-\mathbf{z}') + 2(\mathbf{z}-\mathbf{z}')^T\Delta_{\boldsymbol{\theta}}(\mathbf{z}-\mathbf{z}'), \tag{2.22}$$

and minorized at $\mathbf{z}'$ by the paraboloid:

$$\underline{Q}_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{z}') \equiv \gamma_{\boldsymbol{\theta}}(\mathbf{z}')\left[1 + \mathbf{z}'\Omega_{\boldsymbol{\theta}}\mathbf{z}'\right] - \gamma_{\boldsymbol{\theta}}(\mathbf{z}')\mathbf{z}^T\Omega_{\boldsymbol{\theta}}\mathbf{z}, \tag{2.23}$$

where $\Omega_{\boldsymbol{\theta}} = \underset{i=1,\cdots,p}{\text{diag}}\left\{\sum_{i\in\mathbf{u}\subseteq[p]}\Gamma_{|\mathbf{u}|}\prod_{l\in\mathbf{u}}\theta_l\right\}$ *and* $\Delta_{\boldsymbol{\theta}} = \frac{1}{e}\left(\max_l \Omega_{\boldsymbol{\theta},ll}\right)\mathbf{I}_p$.

Using $\bar{Q}$ and $\underline{Q}$, a majorizing paraboloid can then be established for $\hat{E}_i$:

Lemma 5. *$\hat{E}_i(\mathbf{x}; \mathcal{Y}, \vartheta, \mathcal{D}_{-i})$ is majorized at $\mathbf{x}' \in \mathbb{R}^p$ by:*

$$h_i(\mathbf{x}|\mathbf{x}'; \mathcal{Y}, \vartheta, \mathcal{D}_{-i}) = \frac{1}{nR}\sum_{\substack{j=1\\ j\neq i}}^{n}\sum_{r=1}^{R}\bar{Q}_{\boldsymbol{\theta}_r}(\mathbf{x}-\mathbf{x}_j|\mathbf{x}'-\mathbf{x}_j) - \frac{1}{NR}\sum_{m=1}^{N}\sum_{r=1}^{R}\underline{Q}_{\boldsymbol{\theta}_r}(\mathbf{x}-\mathbf{y}_m|\mathbf{x}'-\mathbf{y}_m),$$

which has the unique closed-form minimizer:

$$M_i(\mathbf{x}';\mathcal{Y},\vartheta,\mathcal{D}_{-i}) = \left(\frac{2}{NR}\sum_{m=1}^{N}\sum_{r=1}^{R}\gamma_{\boldsymbol{\theta}_r}(\mathbf{x}'-\mathbf{y}_m)\Omega_{\boldsymbol{\theta}_r} + \frac{4(n-1)}{nR}\sum_{r=1}^{R}\Delta_{\boldsymbol{\theta}_r}\right)^{-1}$$
$$\left[\frac{2}{NR}\sum_{m=1}^{N}\left(\sum_{r=1}^{R}\gamma_{\boldsymbol{\theta}_r}(\mathbf{x}'-\mathbf{y}_m)\Omega_{\boldsymbol{\theta}_r}\right)\mathbf{y}_m + \frac{2}{nR}\sum_{\substack{j=1\\j\neq i}}^{n}\left(\sum_{r=1}^{R}\gamma_{\boldsymbol{\theta}_r}(\mathbf{x}'-\mathbf{x}_j)\Omega_{\boldsymbol{\theta}_r}\right)(\mathbf{x}'-\mathbf{x}_j) + \right.$$
$$\left.\frac{4(n-1)}{nR}\left(\sum_{r=1}^{R}\Delta_{\boldsymbol{\theta}_r}\right)\mathbf{x}'\right]. \tag{2.24}$$

From this, one can then prove the stationary convergence of `psp.ccp` and `psp.sccp`:

Theorem 14. *Let $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}_+^p$, and suppose $\mathcal{X}$ and Θ are convex.*

(a) If $\mathcal{X}$ and Θ are closed, then for any initial point set $\mathcal{D}^{[0]} \subseteq \mathcal{X}$ and fixed sample batches $\mathcal{Y} \subseteq \mathcal{X}$ and $\vartheta \subseteq \Theta$, the sequence $(\mathcal{D}^{[l]})_{l=1}^{\infty}$ returned by `psp.ccp` *converges to a stationary limiting point set $\mathcal{D}^{[\infty]}$ for $\hat{E}$,*

(b) If $\mathcal{X}$ and Θ are compact, then for any initial point set $\mathcal{D}^{[0]} \subseteq \mathcal{X}$, the sequence $(\mathcal{D}^{[l]})_{l=1}^{\infty}$ returned by `psp.sccp` *converges a.s. to a stationary limiting point set $\mathcal{D}^{[\infty]}$ for $E_{\boldsymbol{\theta}\sim\pi}$.*

For part (b), the compactness condition on $\mathcal{X}$ and Θ is necessary for the convergence of stochastic MM algorithms [45]. In practice, this is not too restrictive, because a truncation can always be performed on $\mathcal{X}$ or Θ to capture probability sufficiently close to 1.

2.5.3 Algorithm running time

One computational bottleneck for `psp.ccp` and `psp.sccp` is the evaluation of the diagonal matrix $\Omega_{\boldsymbol{\theta}}$ in Lemma 4, a step required for calculating the closed-form iterative map M_i. Addressing this is particularly important for high-dimensions, because a brute-force

evaluation of each entry in $\Omega_{\boldsymbol{\theta}}$ requires $\mathcal{O}(2^p)$ work, which is infeasible for even moderate choices of p. Similar to the recursive component-by-component construction of POD-weighted shifted lattice rules (see Section 5.6 of [16]), the following theorem provides a recursive algorithm for efficiently computing $\Omega_{\boldsymbol{\theta}}$:

Theorem 15. *The l-th diagonal of $\Omega_{\boldsymbol{\theta}}$ can be computed as $\Omega_{\boldsymbol{\theta},ll} = \theta_l \sum_{k=1}^{p} \Gamma_k r_{p,k-1}^{(-l)}$. For each $l = 1, \cdots, p$, $r_{p,k-1}^{(-l)}$ can be computed recursively by:*

$$r_{s,k}^{(-l)} = \theta_s r_{s-1,k-1}^{(-l)} + r_{s-1,k}^{(-l)}, \quad s \in [p] \setminus \{l\}, \quad r_{l,k}^{(-l)} = r_{l-1,k}^{(-l)}, \tag{2.25}$$

with initial values $r_{s,0}^{(-l)} = 1$ and $r_{s,k}^{(-l)} = 0$, $k > s$.

The appeal of such a scheme is that each entry in $\Omega_{\boldsymbol{\theta}}$ can now be computed in $\mathcal{O}(p^2)$ work, which is much faster than the $\mathcal{O}(2^p)$ work in a brute-force evaluation. For *finite-order* order weights, i.e., $\Gamma_{|\mathbf{u}|} = 0, |\mathbf{u}| > K$ for some $K < p$, this work can be further reduced to $\mathcal{O}(Kp)$. Since our choice of order weights impose a quick factorial decay in $|\mathbf{u}|$ (see Section 2.4.2), a simple truncation can satisfy this finite-order condition without sacrificing much accuracy.

Using this procedure and assuming the subsample sizes N and R are independent of point set size n or dimension p, the running time for each update of $\mathscr{D}^{[l]}$ is $\mathcal{O}\{n(np+Kp^2)\}$. Since this time grows quadratically in n and p, the proposed algorithm can efficiently generate high-quality solutions for point set sizes as large as 10,000 in dimensions as large as 200 within a matter of hours (for $p \leq 50$, this can be reduced to a matter of minutes). While this running time is quite fast from an optimization perspective, it is considerably slower than number-theoretic QMC methods, which can generate millions of points in hundreds of dimensions within a matter of seconds. The value of the proposed method is its ability to generate *optimal* sampling points for any non-uniform distribution, with the flexibility to adjust the desired level of PGOF.

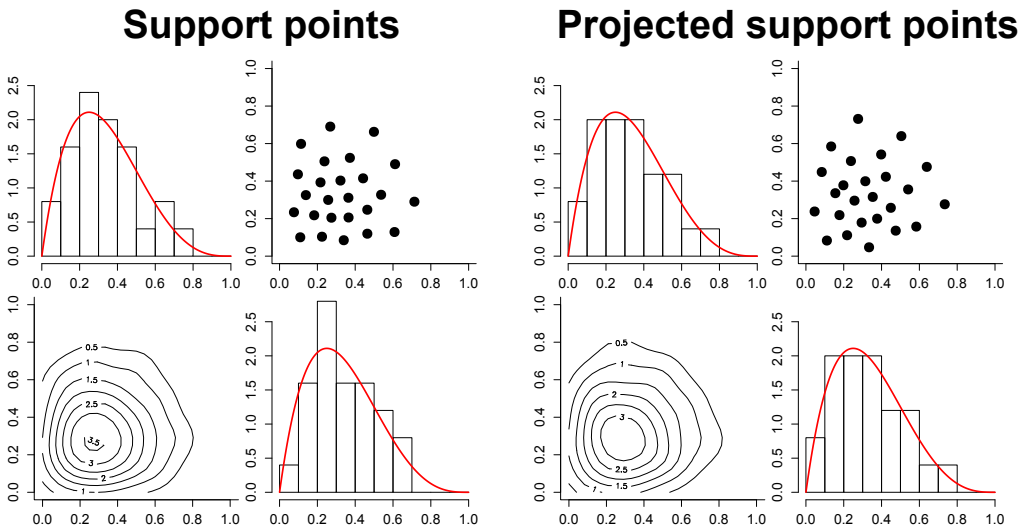

Figure 2.3: $n = 25$-point SPs and PSPs for the 2-d i.i.d. $Beta(2, 4)$ distribution. Diagonals show the marginal histograms of the point set and the true marginal densities, and off-diagonals show the scatterplot of the point set and its density contour plot.

2.6 Simulations

We now demonstrate the advantages of PSPs over existing methods in several simulations. This comparison is made in two parts: we first assess the PGOF of these point sets, then evaluate their integration performance on several integrands with low-dimensional structure.

2.6.1 Visualization and metrics

For visualization, consider Figure 2.3, which plots the $n = 25$-point SPs and PSPs for the 2-d i.i.d. $Beta(2, 4)$ distribution. We see that both SPs and PSPs provide excellent GOF in the full 2-d space, which is not surprising. However, regarding PGOF, PSPs provide a near-perfect representation of the marginal $Beta(2, 4)$ distributions, whereas the corresponding fit for SPs is quite poor. In applications where only one of the two dimensions is active, PSPs can provide considerably improved performance over SPs.

To quantify this improvement, we compare the PGOF of the proposed point sets with SPs, kernel herding, MC and the inverse-transform of a Sobol' sequence [24, 25], a pop-

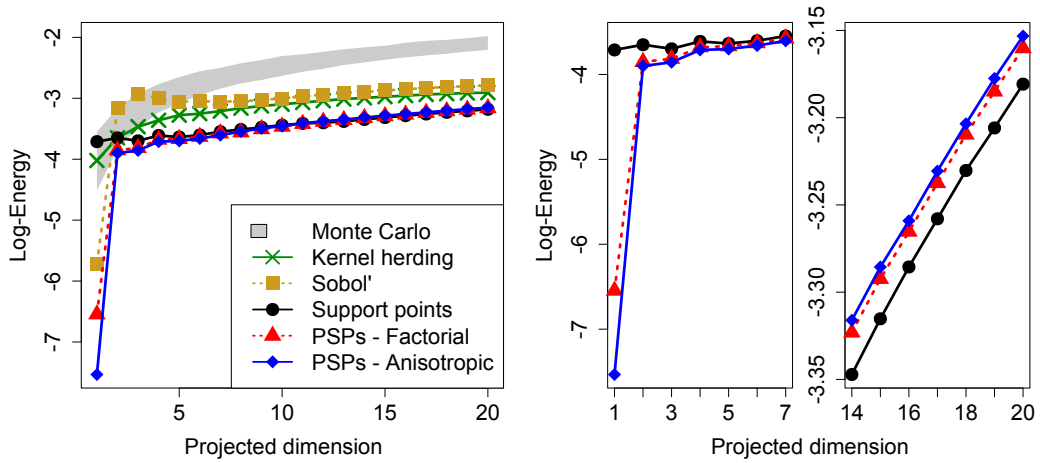

Figure 2.4: Log-energy of the worst-case PGOF for various $n = 50$ point sets on the 20-d i.i.d. $N(0, 1)$ distribution. A close-up of the first seven and last seven dimensions is shown on the right.

ular QMC method. For $n = 50$ point sets on the 20-d i.i.d. $N(0, 1)$ distribution, Figure 2.4 shows the log of the energy distance (see Definition 7) for the *worst-case* fit over all projected subspaces of dimension l, $l = 1, \cdots, 20$. Two observations can be made from these plots. First, both anisotropic and factorial PSPs provide a considerably better fit of low-dimensional marginal distributions than existing methods. Specifically, PSPs enjoy a lower log-energy to MC, Sobol' points and herding points for all projected dimensions, and offer an improvement to SPs for all subspace dimensions less than 11. Second, a trade-off can be seen between SPs, factorial PSPs and anisotropic PSPs. On one end, SPs enjoy a better fit for higher-dimensional marginal distributions than PSPs (see the right-hand plot in Figure 2.4), which is not surprising because SPs focuses solely on the full 20-d distribution. On the other end, anisotropic PSPs enjoy improved fit on lower-dimensional projections, which is again expected because the anisotropic order weights emphasize the importance of one-dimensional marginals. Factorial PSPs provide an intermediate position in this trade-off, and is therefore appropriate for integrands with moderate effect sparsity.

2.6.2 Integration

We now investigate the integration performance of PSPs in comparison to SPs, Monte Carlo, and the inverse-transform of randomly-shifted lattice rules (IT-RLR, using the CBC construction in [22]), another popular QMC method. For the special case of $F = U[0,1]^p$, IT-RLR enjoys a dimension-free convergence rate [16], and therefore provides a good benchmark for PSPs. Three choices of F are considered: the i.i.d. $N(0,1)$, the i.i.d. $Exp(1)$ and the i.i.d. $Beta(2,4)$ distributions, with p ranging from 5 to 100. Two choices of integrands are tested: the Gaussian peak function (GAPK, [54]): $g(\mathbf{x}) = \exp\{-\sum_{l=1}^{p} \alpha_l^2 (x_l - u_l)^2\}$ and the additive Gaussian function (ADD): $g(\mathbf{x}) = \exp\{-\sum_{l=1}^{p} \beta_l x_l\}$, where u_l is the marginal mean for F_l. To account for low-dimensional structure, a proportion q of the p dimensions is randomly chosen to be active, with α_l and β_l set as $0.1/(qp)$ and $0.01/(qp)$ for active dimensions, and 0 otherwise. The corresponding integrands are abbreviated by GAPK(q) and ADD(q), respectively.

Figure 2.5 plots the resulting log-absolute errors in $p = 5$, 20 and 50 dimensions for the GAPK(0.2) and ADD(0.5) integrands under the i.i.d. $Exp(1)$ distribution (results are similar for other cases, and are omitted for brevity). We make two observations here. First, both anisotropic and factorial PSPs provide reduced errors to SPs in most test cases, which demonstrates the value in minimizing PGOF for problems with low-dimensional structure. Second, both types of PSPs provide improvements over both MC and IT-RLR in nearly all test cases, which shows its effectiveness over existing methods. Such results are not too surprising given the improved PGOF for PSPs in Figure 2.4.

2.7 Applications

Next, we present two real-world applications for which PSPs can be employed as an effective data reduction technique. The first application involves the reduction of training data for efficient kernel learning, and the second involves the reduction of MCMC sample data

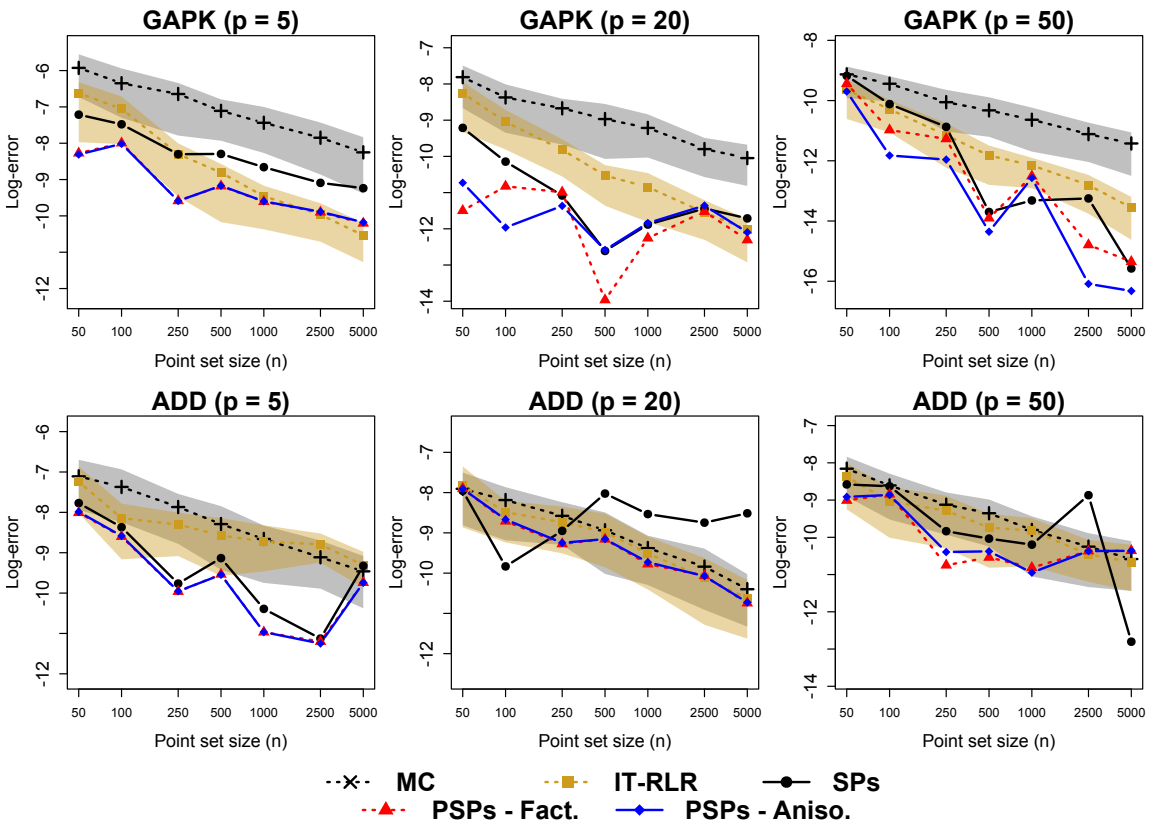

Figure 2.5: Log-absolute errors for GAPK(0.2) and ADD(0.5) under the p-dim. i.i.d. $Exp(1)$ distribution. Lines denote log-average errors, and shaded bands mark the 25-th and 75-th quantiles.

for efficient Bayesian computation.

2.7.1 Kernel ridge regression

In statistical learning, there has been a wealth of recent work on the topic of *kernel methods*. As its name suggests, kernel methods make use of a kernel function k (typically non-linear) to quantify similarities between data points; this then allows for effective, non-linear modeling in both supervised and unsupervised learning problems (e.g., support vector machines [85], kernel principal components analysis [**Mea1998**], and kernel ridge regression [86]). Letting N be the number of data points in the training dataset, one key bottleneck for kernel methods is that it can be very time-consuming to compute for large N. More specifically, kernel methods require the computation of an $N \times N$ matrix inverse, which has a run-

ning time of $\mathcal{O}(N^3)$; for $N > 5,000$, this becomes computationally infeasible to run on most desktop computers. This problem is further compounded when training data is high-dimensional (i.e., $p \gg 1$), since a larger sample size N is typically required for learning. In this context, the proposed PSPs offer a solution to this dilemma, by reducing the large, high-dimensional training dataset to retain low-dimensional structure for modeling.

We make use of a well-known machine learning dataset, the Million Song Dataset (MSD; [34]), to illustrate the effectiveness of PSPs for this problem. MSD is an open-source collection of audio features and metadata, extracted from a million contemporary music tracks released in the years 1922 – 2011. We consider here a subset of this data from the UCI Machine Learning Repository (515,345 songs), with $N = 463,715$ songs used for training and the remainder for testing (this training-testing split is recommended by the data publishers). In total, $p = 90$ song features (continuous) are extracted, including the loudness, pitch, and timbre of each song track. Here, the goal is to first fit a predictive model using the training data, then use this to predict the release year (treated as continuous) for a new song in the testing data.

To build this predictive model, we employ a kernel method called *kernel ridge regression* (KRR). Given (a) a kernel of choice k, and (b) training song features $\{\mathbf{f}_m\}_{m=1}^N$ (inputs, normalized to zero mean and unit variance) and release years $\{y_m\}_{m=1}^N$ (output, normalized to zero mean and unit variance), KRR fits the following non-linear smoother $\hat{h}$:

$$\hat{h} \leftarrow \underset{h \in \mathscr{H}_k}{\text{Argmin}} \left\{ \frac{1}{N} \sum_{m=1}^{N} (y_m - h(\mathbf{f}_m))^2 + \lambda \|h\|_{\mathscr{H}_k}^2 \right\}, \tag{2.26}$$

where $\mathscr{H}_k$ is the RKHS of kernel k with corresponding norm $\|\cdot\|_{\mathscr{H}_k}$. Here, the smoother $\hat{h}$ can be viewed as the non-linear function which best fits the training dataset, subject to a regularization penalty $\lambda \|h\|_{\mathscr{H}_k}^2$. Using this fitted function, one can then use $\hat{h}(\mathbf{f}_{new})$ to predict the release year of a new song with features $\mathbf{f}_{new}$. As typical in statistical regularization problems, the penalty λ is tuned via cross-validation [86].

Unfortunately, as mentioned previously, the computation of (2.26) requires the inverse of the $N \times N$ matrix $[k(\mathbf{f}_m, \mathbf{f}_{m'})]_{m=1}^{N}{}_{m'=1}^{N}$ (see [86] for details), which has a running time of $\mathcal{O}(N^3)$. Clearly, for MSD, the full fit (2.26) is computationally infeasible with $N = 463,715$! To this end, let $n \ll N$, and consider the following *reduced* fit:

$$\hat{h}' \leftarrow \underset{h \in \mathscr{H}_k}{\text{Argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i' - h(\mathbf{f}_i'))^2 + \lambda \|h\|_{\mathscr{H}_k}^2 \right\}, \tag{2.27}$$

where $\mathcal{T}' := \{(\mathbf{f}_i', y_i')_{i=1}^n\}$ is a reduced subset of the full training data $\mathcal{T} := \{(\mathbf{f}_m, y_m)_{m=1}^N\}$. Using (2.27), the computation time for $\hat{g}'$ reduces from $\mathcal{O}(N^3)$ to $\mathcal{O}(n^3)$. The goal then is to judiciously select a good subset of the training data, so that the objective function in (2.27) well-approximates that in (2.26). Letting F be the e.d.f. of the full training data $\mathcal{T}$, this is equivalent to finding a reduced dataset $\mathcal{T}' \subseteq \mathcal{T}$ with e.d.f. F_n such that $\mathbb{E}_{\mathbf{X} \sim F}[g(\mathbf{X})] \approx \mathbb{E}_{\mathbf{X} \sim F_n}[g(\mathbf{X})]$, where g is the integrand:

$$g(\mathbf{f}, y) = \{y - h(\mathbf{f})\}^2. \tag{2.28}$$

Moveover, it is highly unlikely that *all* $p = 90$ song features are useful for prediction, e.g., from intuition, song pitch (and certainty its interaction effects) should not be an important predictor for release year. This suggests that the true input-output function h (and hence g) is inherently low-dimensional. In this context, the PSPs of F (rounded to its closest point in $\mathcal{T}$) should provide a good choice for the reduced dataset $\mathcal{T}'$ in (2.27).

Our set-up is as follows. We compare three different methods: (a) anisotropic PSPs on F (rounded to its closest point in $\mathcal{T}$), (b) herding points on F using the standard Gaussian kernel (rounded to its closest point in $\mathcal{T}$), and (c) random subsamples from $\mathcal{T}$. All three reduce the full training dataset $\mathcal{T}$ to $n = 4,000$ points. These methods are then compared on predictive performance on 250 randomly-chosen songs in the testing set, with this randomization repeated 250 times to provide an measure of error variability.

Figure 2.6 plots the distribution of these prediction errors for the three methods. Two

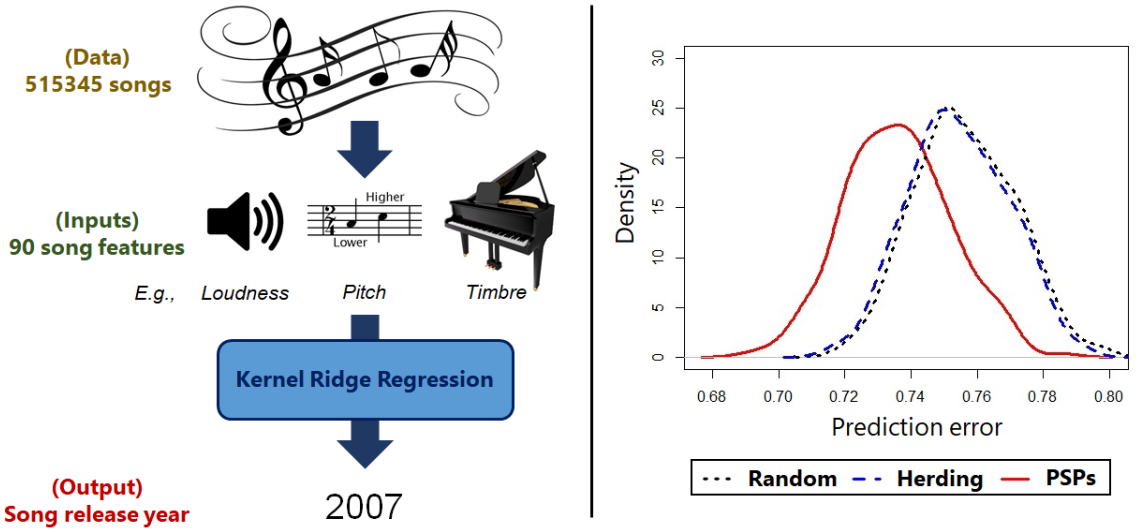

Figure 2.6: (Left) A visualization of the kernel ridge regression procedure for predicting song release year. (Right) Distribution of prediction error for 250 songs in testing set.

observations are of interest here. First, we see that herding offers no reduction in prediction error over random sampling. This is not too surprising, since we know from intuition that the integrand g in (2.28) has low-dimensional structure. Such a structure is, however, not accounted for in the kernel choice for herding, hence its comparable performance with random sampling. Second, we see that PSPs provide noticeably better predictive performance over both herding and random samping. This is again expected, because the kernel choice for PSPs accounts for low-dimensional structure when performing data reduction.

Lastly, we compare the running time of these methods for both data reduction and KRR computation in (2.27) (λ tuned via cross-validation), with the hypothetical running time of the full KRR problem in (2.26) without data reduction. Random subsampling here is the quickest method, requiring 1,583 seconds on a single-core processor (this corresponds to only the KRR fit). Kernel herding and PSPs require 2,631 and 3,057 seconds of computation time, respectively (this increase in time is due to the data reduction step). To contrast, the full KRR step in (2.26) (i.e., without data reduction) has a hypothetical running time of $1,583 \cdot N^3/n^3 = 2.47 \times 10^9$ sec. (≈ 78.2 years!), and has a memory requirement of $\mathcal{O}(N^2) \approx 1,720$ gigabytes; this problem is clearly *infeasible* on standard computing

systems. Given time and memory constraints, our approach yields the best predictive performance of all three data reduction methods.

2.7.2 MCMC thinning

Next, we present an important application of PSPs for reducing big data in Bayesian computation. For Bayesian modeling, one learns the underlying parameters of interest by simulating from a posterior distribution, typically via MCMC sampling methods. In practice, Bayesian practitioners perform a post-processing step called *thinning* – the discarding of all-but-every k-th sample from the MCMC sample chain (call this $\{\mathbf{y}_m\}_{m=1}^N$). Thinning is done for three reasons [56]: it reduces high autocorrelations in the sample chain, lowers the memory requirement for sample storage, and reduces the computation time for posterior quantities of interest. One key weakness of thinning is that it is quite wasteful, since valuable information from posterior samples is carelessly discarded. To this end, PSPs can offer an improved alternative to thinning, by employing the full MCMC chain to find a good representative point set. The proposed approach is particularly effective for large-scale Bayesian modeling problems with many parameters, where (a) the sample chain $\{\mathbf{y}_m\}_{m=1}^N$ is high-dimensional, but (b) posterior quantities of interest are typically computationally expensive and depend on a small number of parameters.

To illustrate the effectiveness of PSPs, we consider a Bayesian modeling problem involving a solid end milling process, a common cutting process for precise machining of complicated parts in the aerospace industry. Figure 2.7 visualizes this process: a cutting tool (in blue) applied at a force to the workpiece (in gray), then moved along the surface of the workpiece along the blue direction lines, stripping away material as it passes. There are six design inputs of interest: five for the cutting tool (rake angle, helix angle, relief angle, corner radius, and flute length), and one for the workpiece material (hardness); Figure 2.8 (left) summarizes the desired design region. Figure 2.8 (right) shows, for several different input settings, the peak tangential force over time ($T = 3,373$ force values in total) – a key

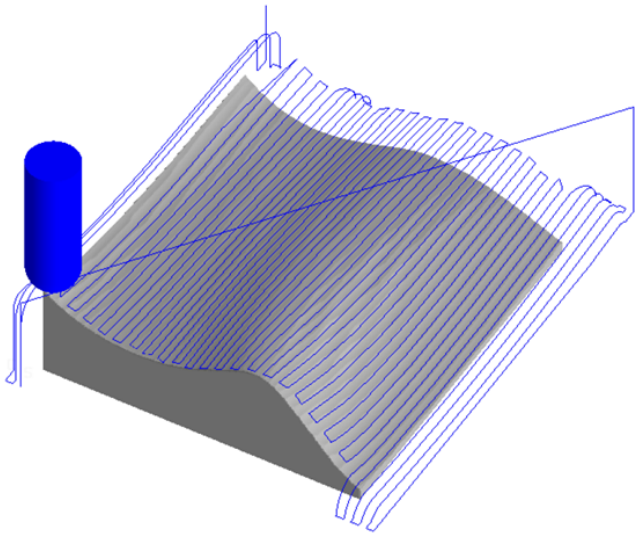

Figure 2.7: A visualization of the solid end milling procedure.

output response of the milling process. For a fixed input setting, this tangential force can be obtained via computer simulations on the standard Production Module software from Third Wave Systems[1], with each run requiring a computation time of 2 minutes. The exploration of the full design space using *solely* computer simulations can therefore be quite time-consuming; to this end, our goal is to build an efficient emulator for predicting peak tangential force (as a function of time) for an unsimulated input setting.

We employ here a standard model for emulation (see [49]): for each slice of time, the tangential force values are modeled using independent Gaussian process (GP) models with time-varying correlation parameters. More specifically, for fixed time t, let $F_t(\mathbf{c})$ be the force observed at time t using input setting $\mathbf{c} \in \mathbb{R}^6$. Our emulator model assumes:

$$f_t(\mathbf{c}) \sim \text{GP}\{\eta_t, \sigma_t^2 r(\cdot, \cdot; \boldsymbol{\tau}_t)\}, \quad F_t(\mathbf{c}) \perp F_{t'}(\mathbf{c}), \quad t \neq t'. \tag{2.29}$$

Here, η_t and σ_t^2 are the process mean and variance at time t, and $\boldsymbol{\tau}_t \in \mathbb{R}_+^6$ are the length-scale parameters for the squared-exponential correlation function r. Figure 2.9 shows a visualization of this emulator model. With this in hand, suppose computer simulations are

[1] https://www.thirdwavesys.com/production-module/

Design variables	Design range
Hardness	95 – 125
Rake angle	0.00 – 0.05
Helix angle	14 – 26
Relief angle	7 – 13
Corner radius	0.45 – 0.50
Flute length	0.7 – 1.3

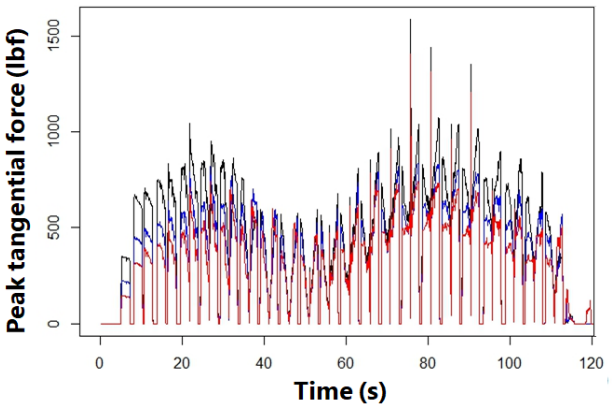

Figure 2.8: (Left) Design range of input variables for solid end milling. (Right) Peak tangential force over time for different input settings.

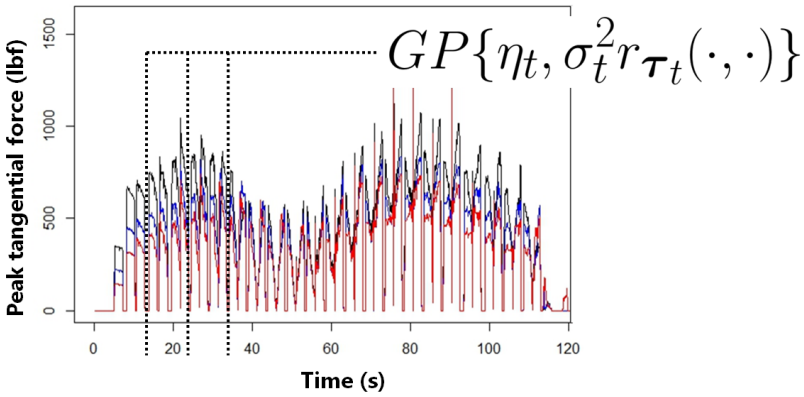

Figure 2.9: A visualization of the GP emulation model over time.

conducted at input settings $\{\mathbf{c}_d\}_{d=1}^D$, yielding observed forces over time $\{f_1(\mathbf{c}_d), \cdots, f_T(\mathbf{c}_d)\}_{d=1}^D$. Using this data with *fixed* parameters $\Theta_t = \{\eta_t, \sigma_t^2, \boldsymbol{\tau}_t\}$, the model in (2.29) offers the following closed-form predictor for tangential force at a new input setting $\mathbf{c}_{new}$:

$$\hat{f}_t(\mathbf{c}_{new}; \Theta_t) = \mathbb{E}\{f_t(\mathbf{c}_{new})|\text{Data}\} = \mu_t + \mathbf{r}_{t,new}^T \mathbf{R}_t^{-1}(\mathbf{f}_t - \mu_t), \quad t = 1, \cdots, T, \tag{2.30}$$

where $\mathbf{r}_{t,new} = [r(\mathbf{c}_{new}, \mathbf{c}_d; \boldsymbol{\tau}_t)]_{d=1}^D$, $\mathbf{R}_t = [r(\mathbf{c}_d, \mathbf{c}_{d'}; \boldsymbol{\tau}_t)]_{d=1}^D{}_{d=1}^D$, and $\mathbf{f}_t = [f_t(\mathbf{c}_d)]_{d=1}^D$. The uncertainty in this predictor can also be quantified in closed-form:

$$V_t(\mathbf{c}_{new}; \Theta_t) = \text{Var}\{f_t(\mathbf{c}_{new})|\text{Data}\} = \sigma_t^2 \mathbf{r}_{t,new}^T \mathbf{R}_t^{-1} \mathbf{r}_{t,new}, \quad t = 1, \cdots, T. \tag{2.31}$$

A detailed derivation of these two equations can be found in [49].

Of course, in practice the parameters Θ_t are never known and require estimation. From a Bayesian view, we are interested in the posterior means of these terms, namely $\mathbb{E}_{\Theta_t|\text{Data}}[\hat{f}_t(\mathbf{c}_{new})]$ and $\mathbb{E}_{\Theta_t|\text{Data}}[V_t(\mathbf{c}_{new})]$. Using posterior samples $\Theta_t^{(1)}, \cdots, \Theta_t^{(N)} \sim [\Theta_t|\text{Data}]$, these two quantities can then be estimated via the sample averages:

$$\frac{1}{N}\sum_{m=1}^{N} \hat{f}_t(\mathbf{c}_{new}; \Theta_t^{(m)}) \quad \text{and} \quad \frac{1}{N}\sum_{m=1}^{N} V_t(\mathbf{c}_{new}; \Theta_t^{(m)}), \quad t = 1, \cdots, T. \tag{2.32}$$

The computational bottleneck is now apparent: every evaluation of the integrand in (2.32) requires $\mathcal{O}(D^3)$ work (for the matrix inverse), so the prediction and uncertainty quantification (UQ) of tangential force at a new setting $\mathbf{c}_{new}$ needs $\mathcal{O}(NTD^3)$ work. With many time steps T and a moderately large design size D, the computation of (2.32) becomes time-consuming even for a small number of posterior samples. In this context, the PSPs of $\Theta_t^{(1)}, \cdots, \Theta_t^{(N)}$ can offer considerable reduction in computation time.

Our set-up is as follows. First, $D = 30$ computer simulations are conducted, with input settings allocated using a MaxPro design [69]. Next, for each $t = 1, \cdots, T$, we sample $N = 50,000$ MCMC samples from the posterior distribution $[\Theta_t|\text{Data}]$, and reduce this sample down to $n = 1,000$ points. Finally, the new input setting $\mathbf{c}_{new}$ is chosen as the center point of the design region in Figure 2.8 (left), and prediction / UQ is performed via (2.32) using the reduced MCMC sample. As before, three reduction methods are used: thinning, herding using the standard Gaussian kernel, and (anisotropic) PSPs. The resulting tangential force predictions and UQ over time are then compared (in a mean-squared, time-averaged sense) with the desired posterior quantities $\mathbb{E}_{\Theta_t|\text{Data}}[\hat{f}_t(\mathbf{c}_{new})]$ and $\mathbb{E}_{\Theta_t|\text{Data}}[V_t(\mathbf{c}_{new})]$, which we estimate via a longer MCMC validation chain with 200,000 samples.

Table 2.1 summarizes the time-averaged errors for prediction and UQ for the three data reduction methods. There are two observations of interest here. First, PSPs offer

	Thinning	Herding	PSPs
Prediction	0.073	0.080	**0.068**
UQ	0.214	1.680	**0.201**

Table 2.1: Mean-squared-time-averaged errors for prediction $\mathbb{E}_{\Theta_t|\text{Data}}[\hat{f}_t(\mathbf{c}_{new})]$ and UQ $\mathbb{E}_{\Theta_t|\text{Data}}[V_t(\mathbf{c}_{new})]$ for the three MCMC reduction methods.

a noticeable improvement over random sampling, which is to be expected, since PSPs make use of information from the full MCMC chain for data reduction. More interestingly, herding performs significantly worse than thinning, for both prediction and UQ! This is surprising to us at first, because one would expect the deterministic reduction from herding to yield an *improvement* over thinning (which can be thought of as random sampling). From an engineering perspective, one likely reason is that not all design inputs (and certainty not all interactions effects) are influential for affecting tangential force output. Given this context, our results show that PSPs (which accounts for this underlying low-dimensional structure) provides more effective MCMC reduction than thinning, while herding (which does not incorporate this structure) yields poorer reduction to thinning.

Lastly, we compare the running time of these methods for both data reduction and resulting prediction / UQ via (2.32), with the running time of (2.32) without data reduction. Similar to before, thinning is the quickest method, requiring 15,972 seconds (≈ 4.5 hours) of computation time on a single-core processor (this corresponds to only the prediction and UQ). Herding and PSPs require 18,576 and 20,192 seconds of computation time, respectively (this increase in time is due to the data reduction step). To contrast, the prediction / UQ in (2.32) using the full $N = 50,000$ MCMC samples requires $798,600$ seconds (≈ 9.2 days). Note that this timing is only for predicting tangential force over time for *one* new input setting; to predict for multiple input settings, the computation time of (2.32) using the full MCMC sample would be infeasible. Given this computation constraint, PSPs offer the best prediction and UQ performance of the three data reduction methods tested.

2.8 Conclusion

In this chapter, a new type of point set called projected support points is introduced, which provides excellent goodness-of-fit for a desired distribution F as well as its marginal distributions. Using the generalized Gaussian kernel discrepancy, a theoretical framework is provided for PSPs, connecting the desired idea of projected goodness-of-fit with key principles in experimental design and the notion of a dimension-free error rate in QMC. The idea of a projection kernel is then motivated by assigning a specific prior on kernel scale parameters, through which an interesting link is revealed between PSPs and the maximum projection designs in [69]. The two algorithms `psp.ccp` and `psp.sccp` are proposed, both exploiting a recursive structure in product-and-order weights to efficiently generate PSPs. The effectiveness of PSPs for integration is then demonstrated using numerical simulations and two data reduction applications for kernel learning and MCMC reduction.

While the results and connections presented here are quite interesting, there are still many avenues for future work. First, although the function space $H_{\gamma,\boldsymbol{\theta}}$ provides valuable insight on effect sparsity, hierarchy and heredity, it is of interest to us to expand this connection to a larger family of integrands. Second, given that `psp.ccp` and `psp.sccp` have a running time of $\mathcal{O}\{n(np + Kp^2)\}$, the generation of large point sets in high-dimensions can be computationally burdensome. Since both proposed algorithms rely on subsampling from F, incorporating an additional layer of subsampling for the underlying kernel k (following the doubly-stochastic approach in [87]) may provide a quicker algorithm, and we look forward to studying this further. Finally, it will be useful to explore an adaptive sampling method which incorporates the posterior updating of $\boldsymbol{\theta}$ to learn which dimensions are active in a high-dimensional integrand.

CHAPTER 3

`CMENET` – A NEW METHOD FOR BI-LEVEL VARIABLE SELECTION OF CONDITIONAL MAIN EFFECTS

3.1 Introduction

This chapter proposes a new method for selecting *main effects* (MEs) and a set of reparametrized effects called *conditional main effects* (CMEs) from observational data. A CME can be described as follows. Let A and B denote two binary factors with levels $+$ and $-$. The CME $A|B+$ is then defined as the effect A when effect B is at the $+$ level, and 0 when B is at the $-$ level. In words, such an effect quantifies the influence of A *only* when B is at the level $+$. The CME $A|B-$ can be defined analogously.

The appeal for CMEs as basis functions for variable selection comes from its interpretability in a wide range of applications, including genomics and the social sciences. For example, in gene association studies, where the goal is to identify important genetic contributions for a trait or disease, the CME $A|B+$ quantifies the significance of gene A *only when* gene B is present. Such conditional effects are biologically interpretable and meaningful, as noted in [88]: "[the examination] of how one mutation behaves when in the presence of a second mutation forms the basis of our understanding of genetic interactions, and is part of the fundamental toolbox of genetic analysis." Viewed this way, the selection of CMEs can therefore serve as an effective tool for investigating the activation and inhibition behavior of gene-gene interactions, namely, which genes are *conditionally* active, and which are important in *activating* or *inhibiting* other genes. CMEs also arise naturally in many engineering applications. For example, in an injection molding experiment with two

The paper based on this chapter will appear in *Journal of the American Statistical Association*.

settings for mold temperature A and holding pressure B (pg. 352 of [89]), the CME $A|B+$ measures the effectiveness of mold temperature *only at* a high level of holding pressure. This conditional effect may be a result of material properties for the molding liquid, and the discovery of such effects can provide valuable insight on the injection process.

The idea of CMEs was first introduced in [90] as a way to disentangle effects which are fully-aliased (i.e., perfectly correlated) in a *designed* experiment. Ever since the pioneering work of [91], it has been widely accepted in the design community that fully-aliased effects in a regular, two-level design cannot be "de-aliased" without adding more experimental runs. Such a belief was shown to be false in [90], where the author employed a reparametrization of these fully-aliased effects into CMEs, and allowed for the selection of the resulting conditional effects. A variable selection method for designed experiments is further developed in [92], making use of the natural groupings of CMEs into so-called twin, sibling and family effects. In this chapter, we generalize this CME selection framework to *observational data*, by exploiting the implicit structure of CMEs to form new effect groups and to motivate a novel penalized selection criterion.

For penalized variable selection methods, the usual procedure for two-level factors is to first normalize each factor to zero mean and unit variance [93]. Treating these rescaled factors as continuous variables, standard variable selection techniques using the l_1-penalty in LASSO [93] or non-convex penalties (e.g., [94, 95, 96]) can then be used to identify significant effects. For the problem at hand, however, such methods are inappropriate, because they do not account for the implicit group structure present in CMEs. Grouped selection techniques, such as the group LASSO [97] or the overlapping group LASSO [98], are also not suitable here, because such methods select *all* effects from an active group, whereas only a handful of effects may be active within a CME group.

In this light, a *bi-level* selection strategy is needed to select both *active CME groups* and *active effects within CME groups*. In recent years, there have been important developments on bi-level variable selection, including the sparse group LASSO [99, 49] and the group

exponential LASSO [100, 101]. We extend the latter framework here, because it allows us to encode within the penalization criterion two important selection principles called *CME coupling* and *CME reduction*. These two principles guide the search for good CME models, and can be seen as an extension of effect heredity and effect hierarchy [68], two guiding principles used for model selection in designed experiments.

The chapter is organized as follows. Section 2 provides some motivation for the problem at hand, including the implicit collinearity structure of CME groups and its effect on selection inconsistency. Section 3 proposes a new penalization criterion for CME selection, and illustrates two appealing selection principles (CME coupling and CME reduction) encoded within this criterion. Section 4 introduces a coordinate descent optimization algorithm using threshold operators, and presents an efficient tuning procedure for penalty parameters. Section 5 outlines several simulations comparing the CME selection performance of `cmenet` to existing variable selection methods. Section 6 then demonstrates the usefulness of the proposed method in a gene association study, and Section 7 concludes with directions for future research.

3.2 Background and motivation

3.2.1 CME and CME groups

We first define some notation. Let $\mathbf{y} \in \mathbb{R}^n$ be a vector of n observations, and suppose p main effects are considered. For effect J, let $\tilde{\mathbf{x}}_j = (x_{1,j}, \cdots, x_{n,j}) \in \{-1, +1\}^n$ be its binary covariate vector, $j = 1, \cdots, p$. The tilde on $\tilde{\mathbf{x}}_j$ distinguishes the binary covariate from its normalized analogue $\mathbf{x}_j$, which is introduced later. A CME can then be defined as follows:

Definition 12 (Conditional main effect)**.** *The* conditional main effect (CME) *of J given K at level +, denoted as $J|K+$, quantifies the effect of covariate vector $\tilde{\mathbf{x}}_{j|k+} = (\tilde{x}_{1,j|k+}, \cdots, \tilde{x}_{n,j|k+})$,*

Table 3.1: Model matrix for the two MEs A and B, and its four CMEs.

A	B	$A\|B+$	$A\|B-$	$B\|A+$	$B\|A-$
+1	+1	+1	0	+1	0
+1	-1	0	+1	-1	0
-1	+1	-1	0	0	+1
-1	-1	0	-1	0	-1

where:

$$\tilde{x}_{i,j|k+} = \begin{cases} \tilde{x}_{i,j}, & \textit{if } \tilde{x}_{i,k} = +1 \\ 0, & \textit{if } \tilde{x}_{i,k} = -1 \end{cases}, \quad \textit{for } i = 1, \cdots, n.$$

The CME $J|K-$ can be defined in a similar manner.

Throughout this chapter, the effects J and K are respectively referred to as the *parent effect* and the *conditioned effect* of $J|K+$. Using this terminology, $J|K+$ quantifies the effect of parent J, given its conditioned effect K is at level +. For illustration, Table 3.1 shows the four possible CMEs constructed from two main effects A and B.

Restricted to two-level, fractional factorial designed experiments, [92] identified three important CME groups for selecting an *orthogonal* model, in which active effects are orthogonal to each other. These three groups are: (a) *sibling* CMEs: CMEs with the same parent effect, (b) *twin* CMEs: CME pairs with the same parent and conditioned effect, but with the sign for the latter flipped, (c) *family* CMEs: CMEs with fully-aliased interaction effects. Leveraging this group structure, [92] proposed three rules for selecting a parsimonious and orthogonal model. Rule 1 (the most important selection rule) relies on the two simple mathematical identities:

$$\tilde{\mathbf{x}}_{j|k+} = \frac{1}{2}\left(\tilde{\mathbf{x}}_j + \tilde{\mathbf{x}}_{j*k}\right) \quad \text{and} \quad \tilde{\mathbf{x}}_{j|k-} = \frac{1}{2}\left(\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_{j*k}\right). \tag{3.1}$$

Here, $\tilde{\mathbf{x}}_{j*k} = \tilde{\mathbf{x}}_j \circ \tilde{\mathbf{x}}_k$ is the covariate vector for the traditional two-factor interaction (2FI) $J * K$, where $\circ$ is the Hadamard (entry-wise) product. From (3.1), the CME $J|K+$ can then be viewed as an average of the main effect for J and the interaction effect for $J * K$; a

similar interpretation holds for the CME $J|K-$. Motivated by this interpretation, Rule 1 of [92] replaces a selected ME J and 2FI $J*K$ with either (a) the CME $J|K+$, if the signs for J and $J*K$ are identical and their effect magnitudes are similar, or (b) the CME $J|K-$, if the signs for J and $J*K$ are different and their effect magnitudes are similar. Such a rule (along with Rules 2 and 3) allows for the disentangling of fully-aliased interaction effects in a designed experiment.

The above CME groupings, however, are not suitable for analyzing observational data, because an orthogonal model is most likely not attainable for this more general setting. Instead, by exploring the correlation structure of CMEs, the following new groupings can be derived:

1. *Sibling* CMEs: CMEs which share the same parent effect, e.g., $\{A|B+, A|B-, A|C+, A|C-, A|D+, A|D-, \cdots\}$. This is the same as in [92].

2. *Parent-child* pairs: An effect pair consisting of a CME and its parent ME, e.g., $\{A|B+, A\}, \{A|C+, A\}, \cdots$.

3. *Cousin* CMEs[1]: CMEs which share the same conditioned effect, e.g., $\{B|A+, B|A-, C|A+, C|A-, D|A+, D|A-, \cdots\}$.

We first outline the justification for these groups in terms of collinearity, then discuss why such groupings are appealing from a selection consistency perspective.

3.2.2 Group structure for collinearity

To explore the group structure of CMEs, consider the following latent model for the main effects $\{\tilde{\mathbf{x}}_j\}_{j=1}^p \subseteq \{-1, +1\}^n$. Define the latent matrix $\mathbf{Z} = (z_{i,j})_{i=1}^n{}_{j=1}^p \in \mathbb{R}^{n \times p}$, where each row of $\mathbf{Z}$ is drawn independently from the equicorrelated normal distribution $\mathcal{N}\{\mathbf{0}, \rho\mathbf{J}_p+$

[1]From a purely linguistic point-of-view, these effects are not cousins, because their parent effects are unrelated. However, the notion of cousin nicely encapsulates a weaker form of a sibling relationship, which is the intended meaning here.

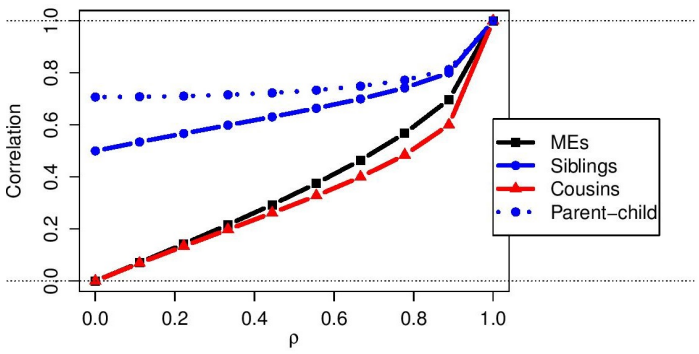

Figure 3.1: Pairwise correlations within the four effect groups as a function of latent correlation ρ.

$(1-\rho)\mathbf{I}_p\}$. Here, $\mathbf{I}_p$ is the $p\times p$ identity matrix, $\mathbf{J}_p$ is the $p\times p$ matrix of ones, and $\rho \in [0,1]$. We then assume the following form for the binary covariates $\{\tilde{\mathbf{x}}_j\}_{j=1}^p$:

$$\tilde{x}_{i,j} = \mathbf{1}\{z_{i,j} > 0\} - \mathbf{1}\{z_{i,j} \leq 0\}, \quad i = 1, \cdots, n, \ j = 1, \cdots, p. \tag{3.2}$$

Note that a larger value of ρ induces a higher correlation between the binary main effects.

Without loss of generality, assume here that the conditioned effects are set at the + level for all CMEs. With the above model, the following theorem reveals an interesting group structure for CMEs. For brevity, proofs of all technical results are deferred to the Appendix.

Theorem 16 (Pairwise correlation within groups)**.** *Under the latent model* (3.2) *for main effects, the four effect groups have the following pairwise correlations:*

Group	*Pairwise correlation*	*Group*	*Pairwise correlation*
Main effects	$\frac{2\sin^{-1}\rho}{\pi}$	Parent-child	$\frac{1}{2\sigma_c}$
Siblings	$\frac{1}{\sigma_c^2}\left\{\frac{1}{4} + \frac{\sin^{-1}\rho}{2\pi} - \left(\frac{\sin^{-1}\rho}{\pi}\right)^2\right\}$	Cousins	$\frac{1}{\sigma_c^2}\left\{\frac{\sin^{-1}\rho}{\pi} - \left(\frac{\sin^{-1}\rho}{\pi}\right)^2\right\}$

where $\sigma_c^2 = 1/2 - (\sin^{-1}\rho/\pi)^2$.

Figure 3.1 plots the pairwise correlations in Theorem 16 as a function of the latent correlation parameter ρ. Two key observations can be made. First, the magnitudes of these correlations impose a natural hierarchy on the effect groups. For all values of $\rho \in (0,1)$,

parent-child pairs have the largest correlations, followed by sibling pairs, then main effect and cousin pairs. Second, the correlation group structure can vary considerably for different choices of ρ. In the independent setting of $\rho = 0$, sibling and parent-child pairs exhibit high correlations of 0.5 and $1/\sqrt{2}$ (≈ 0.71), respectively, whereas the remaining two groups have zero correlation. For moderately large choices of ρ, say, $\rho = 1/\sqrt{2}$ (≈ 0.71), these correlations become larger and more distinct between different groups, thereby amplifying the underlying CME group structure.

In light of this complex collinearity structure, one may suspect that standard variable selection techniques, such as the LASSO, would perform poorly for CME selection, because such methods impose the same regularization penalty over all variables, and ignore the implicit grouped correlation structure. This is indeed the case, and we demonstrate its poor selection performance in the following section and in the simulations of Section 3.5.

3.2.3 Selection inconsistency

An important property of a selection method is its *consistency* in choosing the correct model. Put mathematically, a method is (sign-)*selection consistent* if $\lim_{n\to\infty} \mathbb{P}(\hat{\boldsymbol{\beta}}_n =_s \boldsymbol{\beta}) = 1$, where $\boldsymbol{\beta} \in \mathbb{R}^p$ is the true coefficient vector, $\hat{\boldsymbol{\beta}}_n$ is the estimated vector from n observations, and $=_s$ denotes equality in sign (see [102] for a precise definition). The following theorem shows that LASSO is indeed inconsistent for simple CME models:

Theorem 17 (Selection inconsistency of LASSO)**.** *Under the latent model* (3.2)*, the LASSO is selection inconsistent in the following situations: (a) for $\rho \geq 0$, a model with $q \geq 3$ active siblings, (b) for $\rho \geq 0.27$, a model with $q = 2$ active main effects, and (c) for $\rho \geq 0.29$, a model with $q \geq 6$ active cousins.*

Theorem 17 demonstrates the poor selection of LASSO for simple CME models, even when little-to-no latent correlation is present. Part (a) says that, even in the uncorrelated setting of $\rho = 0$, LASSO yields poor selection whenever three (or more) siblings are present; part (b) says that, for mild correlations as low as 0.27, the same poor selection arises for two active

MEs; part (c) says that, for correlations lower than 0.29, LASSO enjoys good selection even when many cousins (up to 5) are active – this is not too surprising, because cousins experience the lowest pairwise correlations of the four groups. The proof of this theorem relies on the *irrepresentability condition* [102], which shows that the LASSO is selection inconsistent when active variables are highly correlated with non-active ones.

3.3 `cmenet`: Penalization framework

To address these selection concerns, we propose a novel bi-level variable selection method called `cmenet`, which can identify both active CME groups and active effects within such groups. Similar to popular selection methods such as the elastic net [103] and `SparseNet` [104], the name `cmenet` draws an analogy between the proposed method's ability to select active variables amongst non-active ones, and a fishing net's ability to catch larger fish amongst smaller ones. The penalization scheme for `cmenet` encodes two important principles, called *CME coupling* and *CME reduction*, which, as we show in this section, help guide the selection procedure for CMEs.

3.3.1 Selection criterion

We first introduce the selection criterion. Let $\mathbf{x}_j \in \mathbb{R}^n$ be the normalized vector for the binary main effect covariate $\tilde{\mathbf{x}}_j$, with $\mathbf{x}_j^T \mathbf{1}_n = 0$ and $n^{-1}\|\mathbf{x}_j\|_2^2 = 1$, along with a similar notation for CME covariates. Further let $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_{p'}) \in \mathbb{R}^{n \times p'}$ be the full model matrix consisting of these normalized ME and CME effects, where $p' = p + 4\binom{p}{2}$ is the total number of effects considered. For simplicity, assume all considered effects are MEs and CMEs for the following exposition; Section 3.4.1 gives a simple extension for selecting these effects along with *other* covariate factors. Let $\boldsymbol{\beta} \in \mathbb{R}^{p'}$ be the coefficient vector, with β_j and $\beta_{j|k+}$ its corresponding coefficients for ME J and CME $J|K+$. Finally, assume that $\mathbf{y}$ is centered, i.e., $\mathbf{y}^T \mathbf{1}_n = 0$.

For effect groups, define $\mathcal{S}(j) = \{J, J|A+, J|A-, J|B+, J|B-, \cdots\}$ as the *sibling*

group for parent effect j, and $\mathscr{C}(j) = \{J, A|J+, A|J-, B|J+, B|J-, \cdots\}$ as the *cousin group* for conditioned effect j, $j = 1, \cdots, p$. We propose the following selection criterion, which can be viewed as an extension of the hierarchical framework in [100]:

$$\begin{aligned} \min_{\boldsymbol{\beta}} Q(\boldsymbol{\beta}) &\equiv \min_{\boldsymbol{\beta}} \left\{ \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + P_{\mathcal{S}}(\boldsymbol{\beta}) + P_{\mathscr{C}}(\boldsymbol{\beta}) \right\}, \\ P_{\mathcal{S}}(\boldsymbol{\beta}) &\equiv \sum_{j=1}^{p} f_{o,\mathcal{S}} \left\{ \sum_{k \in \mathcal{S}(j)} f_{i,\mathcal{S}}(\beta_k) \right\}, \quad P_{\mathscr{C}}(\boldsymbol{\beta}) \equiv \sum_{j=1}^{p} f_{o,\mathscr{C}} \left\{ \sum_{k \in \mathscr{C}(j)} f_{i,\mathscr{C}}(\beta_k) \right\}. \end{aligned} \tag{3.3}$$

Here, $f_{o,\mathcal{S}}$ and $f_{i,\mathcal{S}}$ (similarly, $f_{o,\mathscr{C}}$ and $f_{i,\mathscr{C}}$) are *outer* and *inner* penalties which control the *between-group* and *within-group* selection for sibling (similarly, cousin) groups, respectively. While the specific penalty functions are left arbitrary in (3.3), we will introduce `cmenet` for the specific choice of the exponential penalty in [101] for outer penalty, and the (scaled) minimax concave-plus penalty (MC+) in [96] for inner penalty:

$$\begin{aligned} &\textit{Outer: } f_{o,\mathcal{S}}(\theta) = \eta_{\lambda_s,\tau}(\theta),\ f_{o,\mathscr{C}}(\theta) = \eta_{\lambda_c,\tau}(\theta),\ \text{where } \eta_{\lambda,\tau}(\theta) = \frac{\lambda^2}{\tau}\left\{1 - \exp\left(-\frac{\tau\theta}{\lambda}\right)\right\}, \\ &\textit{Inner: } f_{i,\mathcal{S}}(\beta) = g_{\lambda_s,\gamma}(\beta),\ f_{i,\mathscr{C}}(\beta) = g_{\lambda_c,\gamma}(\beta),\ \text{where } g_{\lambda,\gamma}(\beta) = \int_0^{|\beta|} \left(1 - \frac{x}{\lambda\gamma}\right)_+ dx. \end{aligned} \tag{3.4}$$

This inner penalty is a scaled version of the MC+ penalty $\lambda g_{\lambda,\gamma}(\beta)$ in [96] without the scaling factor λ; such a factor is accounted for in the outer exponential penalty $\eta_{\lambda,\tau}(\theta)$.

The appeal for the "exponential-MC+" framework in (3.4) is that it provides a concise parametrization of the grouped collinearity structure in Section 3.2. First, the penalty parameters $\lambda_s > 0$ and $\lambda_c > 0$ allow for differing regularization within sibling and cousin groups, respectively, with larger penalty values reducing the number of selected effects in each group. Assuming such parameters are tuned via cross-validation, a smaller tuned value of λ_s suggests many sibling effects are present in the data, while a smaller λ_c suggests the same for cousin effects. Second, the parameter $\gamma > 1$ controls the non-convexity of the inner MC+ penalty, and provides a "bridge" between the l_0-penalty (obtained when

$\gamma \to 1^+$) and the l_1-penalty in LASSO (obtained when $\gamma \to \infty$). In view of the selection problems for LASSO (see Theorem 17), such a parameter allows for improved selection of the highly correlated CMEs, say, within a sibling group. Lastly, the parameter τ provides two appealing principles called CME coupling and reduction, which we introduce below.

3.3.2 CME coupling and reduction

Consider first a CME $J|K+$ which has yet to be selected, and assume without loss of generality that $\mathbf{x}_{j|k+}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})/n > 0$. Taking the derivative of $Q(\boldsymbol{\beta})$ with respect to $\beta_{j|k+}$, and setting $\beta_{j|k+} = 0$ (as $J|K+$ is not in the model), we get:

$$\frac{\partial}{\partial \beta_{j|k+}} Q(\boldsymbol{\beta})\Big|_{\beta_{j|k+}=0} = -\frac{1}{n}\mathbf{x}_{j|k+}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \Delta_{\mathcal{S}(j)} + \Delta_{\mathcal{C}(k)},$$
$$\text{where } \Delta_{\mathcal{S}(j)} = \lambda_s \exp\left\{-\frac{\tau\|\boldsymbol{\beta}_{\mathcal{S}(j)}\|_{\lambda_s,\gamma}}{\lambda_s}\right\} \text{ and } \Delta_{\mathcal{C}(k)} = \lambda_c \exp\left\{-\frac{\tau\|\boldsymbol{\beta}_{\mathcal{C}(k)}\|_{\lambda_c,\gamma}}{\lambda_c}\right\}. \tag{3.5}$$

Here, $\boldsymbol{\beta}_g \in \mathbb{R}^{|g|}$ denotes the coefficient vector for an effect subset $g \subseteq \{1, \cdots, p'\}$, and $\|\boldsymbol{\beta}_g\|_{\lambda,\gamma} \equiv \sum_{l\in g} g_{\lambda,\gamma}(\beta_l)$ denotes its "norm" under the inner MC+ penalty. (For completeness, a full derivation of the subgradient for $Q(\boldsymbol{\beta})$ – which is quite technical and requires several applications of the chain rule – is found in equation (C.4) of the Appendix.)

Equation (3.5) reveals an appealing selection property of `cmenet` called CME coupling, which we describe below. Note that, when more effects have been selected in the sibling group $\mathcal{S}(j)$ (or cousin group $\mathcal{C}(k)$), the effect norms $\|\boldsymbol{\beta}_{\mathcal{S}(j)}\|$ (or $\|\boldsymbol{\beta}_{\mathcal{C}(k)}\|$) become larger. This then results in a smaller linearized slope $\Delta_{\mathcal{S}(j)}$ (or $\Delta_{\mathcal{C}(k)}$), which generates a decrease in the derivative $\frac{\partial}{\partial \beta_{j|k+}} Q(\boldsymbol{\beta})$ in (3.5). Since the goal is to minimize the selection criterion $Q(\boldsymbol{\beta})$, a smaller derivative allows for greater decrease in $Q(\boldsymbol{\beta})$ when $\beta_{j|k+}$ enters the model. In other words, the CME $J|K+$ has a greater chance of entering the model when other effects in its sibling group $\mathcal{S}(j)$ or its cousin group $\mathcal{C}(k)$ have already been selected; the selection of sibling or cousin effects can *couple* in the selection of the CME

$J|K+$. We call this property ***CME coupling***, following the idea of effect coupling in [101].

Consider next a ME J which has yet to be selected, and assume again that $\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})/n > 0$. Taking the derivative of $Q(\boldsymbol{\beta})$ with respect to β_j, and setting $\beta_j = 0$ (as J is not in the model), we get:

$$\frac{\partial}{\partial \beta_j} Q(\boldsymbol{\beta})\Big|_{\beta_j=0} = -\frac{1}{n}\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \Delta_{\mathcal{S}(j)} + \Delta_{\mathcal{C}(j)}. \tag{3.6}$$

The interpretation of equation (3.6) is similar to that for (3.5). When more effects have already been selected in the sibling group $\mathcal{S}(j)$ (or the cousin group $\mathcal{C}(j)$), the linearized slopes $\Delta_{\mathcal{S}(j)}$ (or $\Delta_{\mathcal{C}(j)}$) become smaller, which then decreases the derivative $\frac{\partial}{\partial \beta_j} Q(\boldsymbol{\beta})$ in (3.6). Hence, the ME J enters the model more easily when effects in its sibling group $\mathcal{S}(j)$ or its cousin group $\mathcal{C}(j)$ have already been selected; the selection of many sibling or cousin effects can then *reduce* to its underlying main effect. We refer to this phenomenon as ***CME reduction***.

The notions of CME coupling and reduction are quite intuitive to expect in many CME applications. Consider the gene expression example in the Introduction, where the selection of the CME $A|B+$ indicates the effectiveness of gene A only when gene B is present. When several sibling CMEs of A, say, $A|B+$ and $A|C+$, are already selected in the model, one naturally expects gene A to be conditionally active under more genes as well. In other words, conditional effects with parent A are more likely to be active compared to conditional effects with no selected siblings – this is precisely the principle of CME coupling. However, when many sibling effects of gene A have already been selected, one may suspect that the underlying parent effect for gene A is active instead of these selected siblings – this is precisely the principle of CME reduction. A similar intuition holds for cousin effects.

An interesting parallel can also be made connecting CME coupling and reduction with the two guiding principles for model selection in designed experiments [68]. The first principle, called (weak) *effect heredity*, states that higher-order interactions can be selected

only when either of its parent main effects are in the model. This idea is quite similar to CME coupling, which allows for easier selection of a CME when effects with either the same parent or conditioned ME have been selected. Furthermore, note that a CME can be interpreted as a component of an interaction effect, because the difference of the two CMEs $A|B+$ and $A|B-$ is precisely the two-factor interaction $A * B$ [92]. Coupling can therefore be seen as an *extension* of effect heredity, after breaking an interaction effect (which is often difficult to interpret) into more interpretable conditional effects. The second principle, called *effect hierarchy*, states that lower-order interactions are more likely active than higher-order ones. This is akin to CME reduction, which encourages the reduction of selected sibling (or cousin) CMEs to its parent (conditioned) effect when too many siblings (cousins) are in the model.

3.4 `cmenet`: Optimization framework

With the proposed penalty $Q(\boldsymbol{\beta})$ in hand, we now present an optimization framework for `cmenet` in three parts. We first introduce the optimization algorithm for minimizing $Q(\boldsymbol{\beta})$, then describe several computational techniques for tuning penalty parameters, and finally conclude with several novel CME screening rules for speeding up the tuning procedure.

3.4.1 Optimization algorithm

Coordinate descent and threshold operators

We first develop the algorithmic framework for minimizing the selection criterion $Q(\boldsymbol{\beta})$. A key tool in this optimization algorithm is *coordinate descent*, which can be explained as follows. Viewing $Q(\boldsymbol{\beta})$ as a function of only the first coefficient β_1 (call this $Q_1(\beta_1)$), we first update β_1 as the minimizer of $Q_1(\cdot)$, keeping the remaining $p' - 1$ coefficients fixed. The same procedure is then applied cyclically over $\beta_2, \cdots, \beta_{p'}$, and repeated until the full coefficient vector $\boldsymbol{\beta}$ converges. In recent years, coordinate descent has become widely used in the variable selection literature (see, e.g., [105, 106, 107]), due to its simplicity and

efficiency for high-dimensional problems. The key to efficiency lies in the existence of a *closed-form* minimizer for the coordinate-wise objective $Q_j(\cdot)$, also known as a *threshold function* from signal processing [108]. We derive below such a threshold function for $Q(\boldsymbol{\beta})$.

Before delving into details, we first investigate the convexity properties of $Q(\boldsymbol{\beta})$:

Proposition 3 (Strict convexity)**.** *$Q(\boldsymbol{\beta})$ is strictly convex whenever $\tau+1/\gamma < \lambda_{min}(\mathbf{X}^T\mathbf{X})/(2n)$, where $\lambda_{min}(\cdot)$ returns the minimum eigenvalue. Also, assuming each column $\mathbf{x}_j$ of $\mathbf{X}$ is normalized (i.e., $\mathbf{x}_j^T\mathbf{1} = 0$ and $n^{-1}\|\mathbf{x}_j\|_2^2 = 1$ for any $j = 1, \cdots, p'$), it follows that $Q_j(\beta_j)$ is strictly convex for any $j = 1, \cdots, p'$, whenever $\tau + 1/\gamma < 1/2$.*

In words, this shows that a sufficiently small choice of $\tau + 1/\gamma$ is needed to ensure some form of convexity for the objective $Q(\boldsymbol{\beta})$. The first part of this proposition shows a unique global minimum exists for $Q(\boldsymbol{\beta})$ when $\tau + 1/\gamma < \lambda_{min}(\mathbf{X}^T\mathbf{X})/(2n)$. Such a result is quite restrictive, because it applies only to the low-dimensional setting of $n \leq p'$, where $\lambda_{min}(\mathbf{X}^T\mathbf{X})$ is strictly positive. The second part guarantees the coordinate-wise objective $Q_j(\beta_j)$ is strictly convex whenever $\tau + 1/\gamma < 1/2$, a result which holds in the high-dimensional setting of $n > p'$. This coordinate-wise convexity is important for deriving the threshold function below.

For a main effect J, consider now its coordinate-wise minimization:

$$\min_{\beta_j} Q_j(\beta_j) = \min_{\beta_j} \left[\frac{1}{2n}\|\mathbf{r}_{-j} - \mathbf{x}_j\beta_j\|_2^2 + \eta_{\lambda_s,\tau}\left\{\|\boldsymbol{\beta}_{\mathcal{S}(j)}\|_{\lambda_s,\gamma}\right\} + \eta_{\lambda_c,\tau}\left\{\|\boldsymbol{\beta}_{\mathcal{C}(j)}\|_{\lambda_c,\gamma}\right\}\right], \tag{3.7}$$

where $\mathbf{r}_{-j} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} + \mathbf{x}_j\beta_j$ is the residual vector fitted without $\mathbf{x}_j$. Similarly, for a CME $J|K+$, its coordinate-wise minimization becomes:

$$\min_{\beta_{j|k+}} Q_{j|k+}(\beta_{j|k+}) = \min_{\beta_{j|k+}} \left[\frac{1}{2n}\|\mathbf{r}_{-(j|k+)} - \mathbf{x}_{j|k+}\beta_{j|k+}\|_2^2 + \eta_{\lambda_s,\tau}\left\{\|\boldsymbol{\beta}_{\mathcal{S}(j)}\|_{\lambda_s,\gamma}\right\} + \eta_{\lambda_c,\tau}\left\{\|\boldsymbol{\beta}_{\mathcal{C}(k)}\|_{\lambda_c,\gamma}\right\}\right]. \tag{3.8}$$

An optimization technique called *majorization-minimization* (MM, see Chapter 12 of [109]) can now be used to derive a threshold function. The main idea of MM is as follows. In-

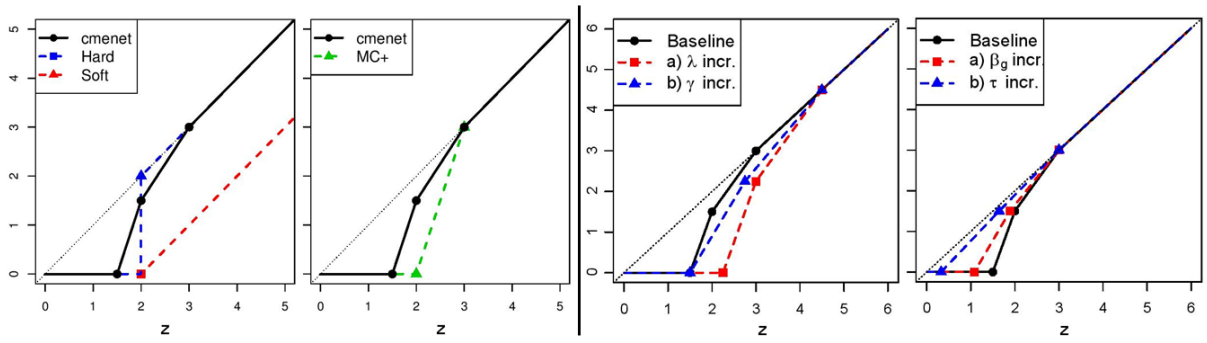

Figure 3.2: (1st and 2nd plots) A comparison of the baseline threshold function S_{λ_1,λ_2} (baseline setting: $(\lambda_1, \lambda_2, \gamma, \tau) = (1, 0.5, 3, 0.05)$ with no selected group effects) with soft-, hard- and MC+ thresholding. (3rd plot) A comparison of the baseline threshold function with two new settings $(1.5, 0.75, 3.0.05)$ and $(1, 0.5, 4.5, 0.05)$, all with no selected group effects. (Last) A comparison of the baseline threshold with two new settings $(1, 0.5, 3, 0.05)$ and $(1, 0.5, 3, 0.25)$, the latter with grouped norms $\|\beta_g\|_{\lambda_1,\gamma} = \|\beta_g\|_{\lambda_2,\gamma} = 5$.

stead of minimizing the original objective function, one first obtains a majorizing surrogate function which lies above the desired objective. This surrogate is then minimized in place of the original objective. Under certain conditions, the solution iterates generated by repeating this procedure converge to a minimizer for the original problem [109]. For Q_j and $Q_{j|k+}$, a simple first-order expansion yields a nice majorizing surrogate function which can be minimized in closed form, as the following theorem demonstrates:

Theorem 18 (Threshold function)**.** *Suppose* $\tau + 1/\gamma < 1/2$. *For fixed* $\tilde{\boldsymbol{\beta}} \in \mathbb{R}^{p'}$, *define* $\bar{Q}_j(\cdot|\tilde{\boldsymbol{\beta}})$ *and* $\bar{Q}_{j|k+}(\cdot|\tilde{\boldsymbol{\beta}})$ *as:*

$$\begin{aligned}\bar{Q}_j(\beta_j|\tilde{\boldsymbol{\beta}}) &= \frac{1}{2n}\|\mathbf{r}_{-j} - \mathbf{x}_j\beta_j\|_2^2 + \eta_{\lambda_s,\tau}\left\{\|\tilde{\boldsymbol{\beta}}_{\mathcal{S}(j)}\|_{\lambda_s,\gamma}\right\} + \eta_{\lambda_c,\tau}\left\{\|\tilde{\boldsymbol{\beta}}_{\mathcal{C}(j)}\|_{\lambda_c,\gamma}\right\} \\ &\quad + \tilde{\Delta}_{\mathcal{S}(j)}\left\{g_{\lambda_s,\gamma}(\beta_j) - g_{\lambda_s,\gamma}(\tilde{\beta}_j)\right\} + \tilde{\Delta}_{\mathcal{C}(j)}\left\{g_{\lambda_c,\gamma}(\beta_j) - g_{\lambda_c,\gamma}(\tilde{\beta}_j)\right\}, \text{ and} \\ \bar{Q}_{j|k+}(\beta_{j|k+}|\tilde{\boldsymbol{\beta}}) &= \frac{1}{2n}\|\mathbf{r}_{-(j|k+)} - \mathbf{x}_{j|k+}\beta_{j|k+}\|_2^2 + \eta_{\lambda_s,\tau}\left\{\|\tilde{\boldsymbol{\beta}}_{\mathcal{S}(j)}\|_{\lambda_s,\gamma}\right\} + \eta_{\lambda_c,\tau}\left\{\|\tilde{\boldsymbol{\beta}}_{\mathcal{C}(k)}\|_{\lambda_c,\gamma}\right\} \\ &\quad + \tilde{\Delta}_{\mathcal{S}(j)}\left\{g_{\lambda_s,\gamma}(\beta_{j|k+}) - g_{\lambda_s,\gamma}(\tilde{\beta}_{j|k+})\right\} + \tilde{\Delta}_{\mathcal{C}(k)}\left\{g_{\lambda_c,\gamma}(\beta_{j|k+}) - g_{\lambda_c,\gamma}(\tilde{\beta}_{j|k+})\right\},\end{aligned}$$

where $\tilde{\cdot}$ *indicates the quantity is computed with* $\tilde{\boldsymbol{\beta}}$ *instead of* $\boldsymbol{\beta}$. *Then:*

(a) $\bar{Q}_j(\cdot|\tilde{\boldsymbol{\beta}})$ *and* $\bar{Q}_{j|k+}(\cdot|\tilde{\boldsymbol{\beta}})$ *are majorization functions for* $Q_j(\cdot)$ *and* $Q_{j|k+}(\cdot)$, *respectively,*

(b) The unique minimizers of $\bar{Q}_j(\cdot|\tilde{\boldsymbol{\beta}})$ *and* $\bar{Q}_{j|k+}(\cdot|\tilde{\boldsymbol{\beta}})$ *are given by* $S_{\lambda_s,\lambda_c}(\mathbf{x}_j^T\mathbf{r}_{-j}/n; \tilde{\Delta}_{\mathcal{S}(j)}, \tilde{\Delta}_{\mathcal{C}(j)})$ *and* $S_{\lambda_s,\lambda_c}(\mathbf{x}_{j|k+}^T\mathbf{r}_{-j|k+}/n; \tilde{\Delta}_{\mathcal{S}(j)}, \tilde{\Delta}_{\mathcal{C}(k)})$*, respectively. Here,* $S_{\lambda_1,\lambda_2}(\cdot; \Delta_1, \Delta_2)$ *is the* threshold function*:*

$$S_{\lambda_1,\lambda_2}(z;\Delta_1,\Delta_2) = \begin{cases} z & \text{if } z \in [\lambda_{(1)}\gamma, \infty), \\ \text{sgn}(z)\left(|z| - \Delta_{(1)}\right) / \left(1 - \dfrac{\Delta_{(1)}}{\lambda_{(1)}\gamma}\right) & \\ & \text{if } z \in \left[\lambda_{(2)}\gamma + \Delta_{(1)}\left(1 - \dfrac{\lambda_{(2)}}{\lambda_{(1)}}\right), \lambda_{(1)}\gamma\right), \\ \text{sgn}(z)\left(|z| - \Delta_{(1)} - \Delta_{(2)}\right) / \left(1 - \dfrac{\Delta_{(1)}}{\lambda_{(1)}\gamma} - \dfrac{\Delta_{(2)}}{\lambda_{(2)}\gamma}\right) & \\ & \text{if } z \in \left[\Delta_{(1)} + \Delta_{(2)}, \lambda_{(2)}\gamma + \Delta_{(1)}\left(1 - \dfrac{\lambda_{(2)}}{\lambda_{(1)}}\right)\right), \\ 0, & \text{otherwise.} \end{cases} \tag{3.9}$$

where $\lambda_{(1)} = \max(\lambda_1, \lambda_2)$ *and* $\lambda_{(2)} = \min(\lambda_1, \lambda_2)$*, with* $\Delta_{(1)}$ *and* $\Delta_{(2)}$ *its corresponding slopes.*

To better understand the shrinkage behavior of this new threshold function, the left two plots in Figure 3.2 show a baseline setting of the `cmenet` threshold $S_{\lambda_1,\lambda_2}(z;\Delta_1,\Delta_2)$, compared with the soft-threshold function (corresponding to the shrinkage operator in LASSO), the hard-threshold function (corresponding to best-subset selection; see [86]), and the MC+ threshold function [107]. The baseline setting for the proposed threshold $S_{\lambda_1,\lambda_2}(z;\Delta_1,\Delta_2)$ is set as $(\lambda_1, \lambda_2, \gamma, \tau) = (1, 0.5, 3, 0.05)$, with $\|\boldsymbol{\beta}_g\|_{\lambda_1,\gamma} = \|\boldsymbol{\beta}_g\|_{\lambda_2,\gamma} = 0$ (i.e., no selected grouped effects). We see that the proposed threshold function is continuous and piecewise linear in four segments. Beginning from the left, the first segment is a horizontal line at zero, and represents the inner-product values for which a coefficient is shrunk to zero after thresholding. The last segment, which matches the identity line, represents the values for which the full coefficient signal is retained without any shrinkage. The middle two segments provide a two-step transition between these two extremes, with slopes controlled by the sibling and cousin penalties. Similar to the MC+ threshold, the `cmenet` threshold bridges the gap between the two extremes of full shrinkage and no

Algorithm 5 `cmenet`: An algorithm for bi-level CME selection

function CMENET($\mathbf{X}, \mathbf{y}, \lambda_s, \lambda_c, \gamma, \tau, \boldsymbol{\beta} = \mathbf{0}_{p'}$) ▷ Assume columns of $\mathbf{X}$ are normalized

- Initialize $\mathbf{r} \leftarrow \mathbf{y} - \bar{\mathbf{y}}$, $\Delta_{\mathcal{S}(j)} = \lambda_s$, $\Delta_{\mathscr{C}(j)} = \lambda_c$ for $j = 1, \cdots, p$

repeat

  for $j = 1, \cdots, p$ **do** ▷ For all main effects...

  - $\beta_0 \leftarrow \beta_j$, $\beta_j \leftarrow S_{\lambda_s,\lambda_c}\{\mathbf{x}_j^T\mathbf{r}/n + \beta_0; \Delta_{\mathcal{S}(j)}, \Delta_{\mathscr{C}(j)}\}$ ▷ Shrinkage
  - $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{x}_j(\beta_0 - \beta_j)$ ▷ Update residual
  - $\Delta_{\mathcal{S}(j)} \leftarrow \Delta_{\mathcal{S}(j)} \exp\{-\tau/\lambda_s \left[g_{\lambda_s,\gamma}(\beta_j) - g_{\lambda_s,\gamma}(\beta_0)\right]\}$ ▷ Update slopes
  - $\Delta_{\mathscr{C}(j)} \leftarrow \Delta_{\mathscr{C}(j)} \exp\{-\tau/\lambda_c \left[g_{\lambda_c,\gamma}(\beta_j) - g_{\lambda_c,\gamma}(\beta_0)\right]\}$

  for $j = 1, \cdots, p$ and $k = 1, \cdots, p$ **do** ▷ For all CMEs (both $J|K+$ and $J|K-$) ...

  - $\beta_0 \leftarrow \beta_{j|k+}$, $\beta_{j|k+} \leftarrow S_{\lambda_s,\lambda_c}\{\mathbf{x}_{j|k+}^T\mathbf{r}/n + \beta_0; \Delta_{\mathcal{S}(j)}, \Delta_{\mathscr{C}(k)}\}$ ▷ Shrinkage
  - $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{x}_{j|k+}(\beta_0 - \beta_{j|k+})$ ▷ Update residual
  - $\Delta_{\mathcal{S}(j)} \leftarrow \Delta_{\mathcal{S}(j)} \exp\{-\tau/\lambda_s \left[g_{\lambda_s,\gamma}(\beta_{j|k+}) - g_{\lambda_s,\gamma}(\beta_0)\right]\}$ ▷ Update slopes
  - $\Delta_{\mathscr{C}(k)} \leftarrow \Delta_{\mathscr{C}(k)} \exp\{-\tau/\lambda_c \left[g_{\lambda_c,\gamma}(\beta_{j|k+}) - g_{\lambda_c,\gamma}(\beta_0)\right]\}$

until $\boldsymbol{\beta}$ converges

return the converged coefficient vector $\boldsymbol{\beta}$

shrinkage; however, the former threshold accomplishes this transition in one step, while the latter achieves this in two steps. This two-step transition for `cmenet` is a consequence of the two-tiered coupling effect from sibling and cousin groups.

Consider next the right two plots of Figure 3.2, which investigate the sensitivity of the proposed threshold $S_{\lambda_1,\lambda_2}(z; \Delta_1, \Delta_2)$ to changes in penalty parameters. From the first plot, an increase in λ_1, λ_2 or γ appears to yield greater shrinkage of the coefficient signal. This is expected, because a larger choices of λ_1 and λ_2 induce greater regularization, and a larger γ generates a "more convex" penalty (see [107]). From the second plot, an increase in the coupling parameter τ in the presence of selected group effects appears to greatly reduce signal shrinkage. This observation nicely demonstrates the earlier CME coupling principle in Section 3.3.2, where the selection of sibling or cousin effects increases the chances of a CME entering the model.

Algorithm statement

Putting all the pieces together, Algorithm 5 summarizes the detailed steps for `cmenet`, which minimizes the selection criterion $Q(\boldsymbol{\beta})$ given fixed parameters λ_s, λ_c, γ and τ. Starting with an initial solution of $\boldsymbol{\beta} = \mathbf{0}_{p'}$, the threshold function in (3.9) is applied cyclically over each element in $\boldsymbol{\beta}$. This iterative procedure is then repeated until $\boldsymbol{\beta}$ converges. Using the majorization function in Theorem 18, one can prove the convergence of `cmenet` to a stationary solution.

Corollary 2 (Convergence of `cmenet`)**.** *When $\tau + 1/\gamma < 1/2$,* `cmenet` *converges to a stationary solution $\hat{\boldsymbol{\beta}}$ satisfying $\nabla Q(\hat{\boldsymbol{\beta}}) = 0$.*

As for its running time, one can show that one coordinate descent cycle in `cmenet` over all p' ME and CME coefficients requires $\mathcal{O}(np')$ work, because each coordinate descent step requires $\mathcal{O}(n)$ work. The linear running time in both sample size n and total effects p' is crucial for the computational efficiency of `cmenet`, particularly when a large number of main effects $p \gg 1$ is considered.

We mention here several extensions for `cmenet`. First, while Algorithm 5 considers only the selection and estimation of CMEs, the proposed algorithm can easily be extended for the selection of both CMEs and *other* covariate factors (whether continuous or discrete). For example, if the l_1-penalty were imposed on these latter factors, one can simply modify the coordinate descent loop in Algorithm 5 by incorporating soft-threshold updates [108] to the coefficients of such factors. The algorithmic convergence for this extension is analogous to Corollary 2, and is not included for brevity. Second, we note that `cmenet`, as stated in Algorithm 5, is suitable for selecting binary CMEs – CMEs which quantify the effect of a *binary* factor at fixed levels of another factor, but not continuous CMEs – CMEs which quantify the effect of a *continuous* factor at fixed levels of another factor. One way to extend `cmenet` for the latter problem is to first (a) discretize the underlying continuous factor into two levels, then (b) perform `cmenet` on the resulting binary CMEs, and finally

(c) quantify the continuous component of these continuous CMEs using the *residuals* from `cmenet` as a new response vector. However, this extension requires further developments, and given the length of the current chapter, we defer such an extension to future work.

3.4.2 Parameter tuning, warm starts and active set optimization

While Algorithm 5 provides an efficient method for minimizing the selection criterion $Q(\boldsymbol{\beta})$ given *fixed* penalty parameters λ_s, λ_c, γ and τ, such parameters are typically not known in practice, and therefore require tuning. We present below a method for performing this tuning procedure, as well as two computational tools – warm starts and active set optimization – which greatly speed up this tuning in practice.

For parameter tuning, we adopt the relatively standard procedure (see, e.g., [86, 107]) of finding the optimal penalty setting whose corresponding model (fitted using `cmenet`) minimizes some estimate of prediction error. In our implementation, called `cv.cmenet`[2], this prediction error is estimated using a technique called K-fold cross validation (or K-fold CV; see [86]), which randomly splits the observed data into K parts, and uses one part of the data to validate the model fitted with the remaining $K-1$ parts. After obtaining this optimal penalty setting, the corresponding fitted model is then used for variable selection and prediction. For brevity, the specific details for `cv.cmenet` are summarized in Appendix C.6.

One practical challenge for this tuning procedure is that there are four parameters $(\lambda_s, \lambda_c, \gamma, \tau)$ to tune for in `cv.cmenet`. Some guiding rules are therefore needed to efficiently explore this 4-d parameter space. The proposition below provides one such rule for (λ_s, λ_c):

Proposition 4 (Search rule for (λ_s, λ_c))**.** *Suppose* $\lambda_s + \lambda_c \geq \max_{j=1,\cdots,p'} |\mathbf{x}_j^T \mathbf{y}|/n$. *When* $Q(\boldsymbol{\beta})$ *is strictly convex, the unique minimizer of* $Q(\boldsymbol{\beta})$ *is the zero solution* $\boldsymbol{\beta} = \mathbf{0}_{p'}$.

It should be noted that, in the high-dimensional setting of $n > p'$, $Q(\boldsymbol{\beta})$ cannot be strictly

[2]In later sections, the tuning procedure `cv.cmenet` is often referred to as simply `cmenet` for brevity.

convex (see discussion for Proposition 3), so $\boldsymbol{\beta} = \mathbf{0}_{p'}$ is only a stationary solution. Nonetheless, the restriction of $\lambda_s + \lambda_c < \max_{j=1,\cdots,p'} |\mathbf{x}_j^T \mathbf{y}|/n$ allows for considerable reduction in the search for interesting choices of λ_s and λ_c. From Proposition 3, another rule is $\tau + 1/\gamma < 1/2$, which ensures the strict convexity of the coodinate-wise problem and therefore the numerical stability of the optimization procedure. For brevity, the incorporation of these rules in `cv.cmenet` is outlined in Appendix C.6.

Two computational tools can be used to greatly speed up the tuning procedure `cv.cmenet` in high-dimensions. The first tool, called *warm starts*, makes use the converged solution from a previous parameter setting to initialize the optimization problem for the current setting. The use of warm starts in variable selection was popularized in [106] for efficiently fitting multiple models along the full LASSO path, and we found such a tool to be equally effective for efficiently fitting multiple models over a grid of penalty parameters for `cmenet`. The second tool, called *active set optimization* (see, e.g., [110, 82]), performs coordinate descent updates over a small subset of *active* variables, instead of over the full set of p' variables. This technique is most effective when there are only a small number of active effects present, because one can avoid performing redundant coordinate descent updates on coefficients of inactive effects. Appendix C.6 provides specific details on how these two tools can be incorporated into `cv.cmenet`.

3.4.3 CME screening rules

When the number of main effects p grows large, performing even one full coordinate descent over all $p' = p + 4\binom{p}{2}$ total effects can be computationally cumbersome. One effective way of reducing computation time in such a situation is the use of screening rules, or *strong rules*, which screen out a large number of inactive variables from consideration using previously-solved coefficient solutions. The term "strong rules" is first coined in [111], where the authors used previously-solved solutions along the LASSO path to screen out inactive effects for subsequent optimizations. We derive below similar strong rules for

screening out inactive effects for `cmenet`, and reveal an interesting connection between these screening rules and CME coupling.

Suppose the parameters γ and τ are fixed, and let j index a variable of interest (ME or CME), with $\mathcal{S}$ and $\mathscr{C}$ its corresponding sibling and cousin group. Furthermore, let $\hat{\boldsymbol{\beta}}(\lambda_s, \lambda_c)$ be an optimal solution of the selection criterion $Q(\boldsymbol{\beta})$ under penalties λ_s and λ_c, and let $c_j(\lambda_s, \lambda_c) = \mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_s, \lambda_c))/n$ denote the inner-product of effect j with the current residual vector. Denoting $\lambda_s^1 > \lambda_s^2 > \cdots > \lambda_s^L$ and $\lambda_c^1 > \lambda_c^2 > \cdots > \lambda_c^M$ as the desired (decreasing) penalty sequences for λ_s and λ_c, the screening procedure can be summarized by the following three strong rules:

1. Suppose there are no active effects in $\mathcal{S}$ and $\mathscr{C}$ for penalty settings $(\lambda_s^{l-1}, \lambda_c^m)$ or $(\lambda_s^l, \lambda_c^{m-1})$. Then effect j is marked as *inactive* for penalty setting $(\lambda_s^l, \lambda_c^m)$ if:

$$|c_j(\lambda_s^{l-1}, \lambda_c^m)| < \lambda_s^l + \lambda_c^m + \frac{\gamma}{\gamma - 2}(\lambda_s^l - \lambda_s^{l-1}) \tag{3.10}$$

   or:

$$|c_j(\lambda_s^l, \lambda_c^{m-1})| < \lambda_s^l + \lambda_c^m + \frac{\gamma}{\gamma - 2}(\lambda_c^m - \lambda_c^{m-1}). \tag{3.11}$$

2. If there are no active effects in the *sibling* group $\mathcal{S}$ for penalty setting $(\lambda_s^{l-1}, \lambda_c^m)$, then effect j is marked as *inactive* for penalty setting $(\lambda_s^l, \lambda_c^m)$ if:

$$|c_j(\lambda_s^{l-1}, \lambda_c^m)| < \lambda_s^l + \Delta'_{\mathscr{C}} + \frac{\gamma}{\gamma - (\Delta'_{\mathscr{C}}/\lambda_c^m + 1)}(\lambda_s^l - \lambda_s^{l-1}), \tag{3.12}$$

   where $\Delta'_{\mathscr{C}} = \lambda_c^m \exp\left\{-\tau \|\boldsymbol{\beta}_{\mathscr{C}}(\lambda_s^{l-1}, \lambda_c^m)\|_{\lambda_c^m, \gamma}/\lambda_c^m\right\}$.

3. If there are no active effects in the *cousin* group $\mathscr{C}$ for penalty setting $(\lambda_s^l, \lambda_c^{m-1})$, then effect j is marked as *inactive* for penalty setting $(\lambda_s^l, \lambda_c^m)$ if:

$$|c_j(\lambda_s^l, \lambda_c^{m-1})| < \Delta'_{\mathcal{S}} + \lambda_c^m + \frac{\gamma}{\gamma - (\Delta'_{\mathcal{S}}/\lambda_s^l + 1)}(\lambda_c^m - \lambda_c^{m-1}), \tag{3.13}$$

where $\Delta'_{\mathcal{S}} = \lambda_s^l \exp\left\{-\tau \|\boldsymbol{\beta}_{\mathcal{S}}(\lambda_s^l, \lambda_c^{m-1})\|_{\lambda_s^l,\gamma} / \lambda_s^l\right\}$.

A theoretical derivation of these rules is provided in Appendix C.7.

While these three rules may appear complicated and technical, they are in fact quite intuitive to understand. All three rules consider conditions under which it would be "safe" to screen out effect j from the optimization problem for the penalty setting $(\lambda_s^l, \lambda_c^m)$. The first rule applies when there are no active effects in $\mathcal{S}$ and $\mathcal{C}$ from previous penalty settings, and screens out effect j if the previous inner-products $c_j(\lambda_s^{l-1}, \lambda_c^m)$ or $c_j(\lambda_s^l, \lambda_c^{m-1})$ are within the upper bounds provided in (3.10) and (3.11). The intuition here is that if effect j is not correlated enough with the residual vectors at the previous penalty settings $(\lambda_s^{l-1}, \lambda_c^m)$ or $(\lambda_s^l, \lambda_c^{m-1})$, then it cannot "catch up" in time to be active for the current setting $(\lambda_s^l, \lambda_c^m)$ (see [111] for details). This first rule can be viewed as an extension of the MC+ strong rule in [112] to the current model. The second rule applies when there are no active effects in the sibling group $\mathcal{S}$ (but some in cousin group $\mathcal{C}$) for the previous setting $(\lambda_s^{l-1}, \lambda_c^m)$. In such a scenario, effect j is screened out if the previous inner-product $c_j(\lambda_s^{l-1}, \lambda_c^m)$ is within the upper bound in (3.12). The key difference between this and the first rule is that, as more effects are selected in the cousin group $\mathcal{C}$, the linearized slope $\Delta'_{\mathcal{C}}$ decays smaller than λ_c^m, which then decreases the screening bound in (3.12) compared to the original bounds in (3.10) and (3.11)[3]. In other words, the presence of coupled cousin effects from a previous setting can *decrease* the screening power of strong rules for the current setting. This is quite similar to the CME coupling phenomenon in Section 3.2, except instead of encouraging the *selection* of effect j, the coupled CMEs make it more diffcult to *screen out* effect j via strong rules. The third rule, which applies when there are no previously-active cousins in $\mathcal{C}$ (but some siblings in $\mathcal{S}$), enjoys a similar interpretation: as more siblings are coupled in from $\mathcal{S}$ at a previous setting, effect j becomes more difficult to screen out via strong rules.

Lastly, we note that while these three rules do screen out a large proportion of inert

[3]Here, we assume the last terms in (3.10) and (3.12) are nearly equal in this comparison; the discrepancy between (3.10) and (3.12) is dominated by the first two terms for most feasible parameter settings.

Table 3.2: Test settings for simulation study.

Simulation parameters	*Settings*
Sample size	$n = 50,\ 100$ or 150
# of main effects considered (total effects considered)	$p = 50,\ 100$ or 150 $\left(p' = p + 4\binom{p}{2} = 4,950,\ 19,900 \text{ or } 44,850\right)$
# of active groups	6 or 8
# of active effects within a group	2 or 3
Effect type	Siblings, cousins, main effects
Latent correlation	$\rho = 0$ or $1/\sqrt{2}$

CMEs, it is possible (but highly unlikely) that an active CME is erroneously screened out. This is illustrated numerically in the following section. To prevent any false-negative screenings, we recommend that the KKT conditions (see equation (C.3) in the Appendix) be checked as a final step for each optimization problem.

3.5 Simulations

We now explore the performance of the proposed method in several simulation studies. Table 3.2 summarizes the test settings for these simulations, with varying sample sizes n and main effects p, varying number of active groups x and active effects within a group y (denoted as GxAy), and whether the grouped effects are siblings or cousins (main effect models are considered here as well). Active effects are assigned a value of 1 in the coefficient vector $\boldsymbol{\beta}$, and non-active effects assigned a value of 0. Each simulation case is then replicated 100 times, with the model matrix $\mathbf{X}$ simulated from the equicorrelated latent model in Section 3.2.2 with $\rho = 0$ and $\rho = 1/\sqrt{2}$, and the response $\mathbf{y}$ simulated independently from $\mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \mathbf{I}_n)$. For brevity, we only report the results for $(n, p) = (50, 50), (100, 100)$ and $(150, 150)$ with G4A2 and G6A2, but similar conclusions hold for other settings.

Under such a set-up, our simulations aim to answer two questions: (a) Does the proposed method `cmenet` yield improved selection of CMEs compared to more generic selection methods? (b) For an active CME, say $J|K+$, is `cmenet` more effective at identifying

this conditional, non-additive relation between J and K, compared to the more traditional 2FI analysis? To answer the first question, we compare `cmenet` with two generic variable selection techniques from the literature: the LASSO [113] using the R package GLMNET [114], and `SparseNet` [107] using the R package SPARSENET [104]. All three methods perform selection on the *same* set of MEs and CMEs, with penalty parameters tuned using 10-fold CV. In this comparison, a better selection performance for `cmenet` shows that the proposed penalty $Q(\boldsymbol{\beta})$ is more appropriate for selecting CMEs compared to generic penalties. To answer the second question, we compare `cmenet` with a popular selection method called `hierNet` [115] for selecting 2FIs. A better selection performance for `cmenet` over `hierNet` thereby demonstrates the effectiveness of the proposed method in identifying the conditional, non-additive nature of CMEs.

We employ two criteria to conduct the above comparisons. The first criterion returns the number of misspecified variables: $\#\{\mathscr{A} \setminus \hat{\mathscr{A}}_n\} + \#\{\hat{\mathscr{A}}_n \setminus \mathscr{A}\}$, where $\mathscr{A}$ is the true active set of MEs and CMEs, and $\hat{\mathscr{A}}_n$ is the set of selected effects after n observations. Smaller values of this indicate better selection performance. Such a criterion is appropriate for `cmenet`, LASSO and `SparseNet`, which perform selection on the MEs and CMEs in $\mathscr{A}$, but a slight modification is needed for `hierNet`, which performs selection on the traditional 2FIs. To this end, let $\mathscr{A}^{(ME)}$ consist of the original MEs in active set $\mathscr{A}$ as well as the parent MEs of the CMEs in $\mathscr{A}$, and let $\mathscr{A}^{(2FI)}$ consist of the 2FIs corresponding to the CMEs in $\mathscr{A}$. The misspecification criterion for `hierNet` can then be written as: $\#\{\mathscr{A}^{(ME)} \setminus \hat{\mathscr{A}}_n^{(ME)}\} + \#\{\hat{\mathscr{A}}_n^{(ME)} \setminus \mathscr{A}^{(ME)}\} + \#\{\mathscr{A}^{(2FI)} \setminus \hat{\mathscr{A}}_n^{(2FI)}\} + \#\{\hat{\mathscr{A}}_n^{(2FI)} \setminus \mathscr{A}^{(2FI)}\}$, where $\hat{\mathscr{A}}_n^{(ME)}$ and $\hat{\mathscr{A}}_n^{(2FI)}$ are the selected MEs and 2FIs from `hierNet`. Put another way, this modified criterion first translates the true CME model into its component MEs and 2FIs (see the identities in (3.1)), then reports the number of misspecifications for the fitted `hierNet` model based on these component effects. The second criterion is the mean-squared prediction error (MSPE): $\mathbb{E}\|\mathbf{y}_{new} - \mathbf{X}_{new}\hat{\boldsymbol{\beta}}\|_2^2$, where $(\mathbf{X}_{new}, \mathbf{y}_{new})$ is an out-of-sample dataset with $n_{new} = 20$ observations simulated from the true model $\mathscr{A}$. Smaller

MSPE values suggest better predictive performance. Here, the focus is on a method which yields the best *selection* performance of CMEs (first criterion); however, such a method should have comparable *predictive* performance to other methods (second criterion).

Figures 3.3 show the number of misspecifications and MSPE for the four methods with $\rho = 0$ and $\rho = 1/\sqrt{2}$, under the simulation settings presented earlier. Consider first the sibling and cousin models in the $\rho = 0$ setting (left part of Figure 3.3), where the underlying MEs are uncorrelated. For these models, `cmenet` provides noticeably improved selection performance over LASSO and `SparseNet` for nearly all simulation settings. This shows that the penalization scheme in $Q(\boldsymbol{\beta})$ is indeed more effective than generic penalties for selecting active CMEs; by accounting for the implicit group structure of CMEs, the proposed method can better guide the variable selection procedure using the novel principle of CME coupling. `cmenet` also yields a sizable selection improvement over `hierNet` for sibling and cousin models, which shows that the proposed approach can better identify the conditional, non-linear nature of CMEs compared to traditional 2FI analysis. One likely explanation is that, because a CME can be decomposed into its component ME and 2FI effects (recall the identities in (3.1) and Rule 1 of [92]), the selection signal of an active CME is much stronger than the signals from its component ME and 2FI effects. `cmenet`, by performing selection directly on the CMEs with greater signal, can more easily identify the underlying active effects compared to `hierNet`, which performs selection on its component ME and 2FI effects with diluted signals. As for MSPE, `cmenet` enjoys comparable or improved performance to the other three methods, which is as desired.

Consider next the main effect models for $\rho = 0$ (left part of Figure 3.3). We see that `cmenet` enjoys superior selection performance to LASSO and `SparseNet`, which demonstrates the effectiveness of the CME reduction principle in reducing selected CMEs into its underlying parent ME. Compared to `hierNet`, `cmenet` provides comparable (but slightly worse) selection for these main effect models, an observation not too surprising given that the proposed method specifically tackles the problem of CME selection.

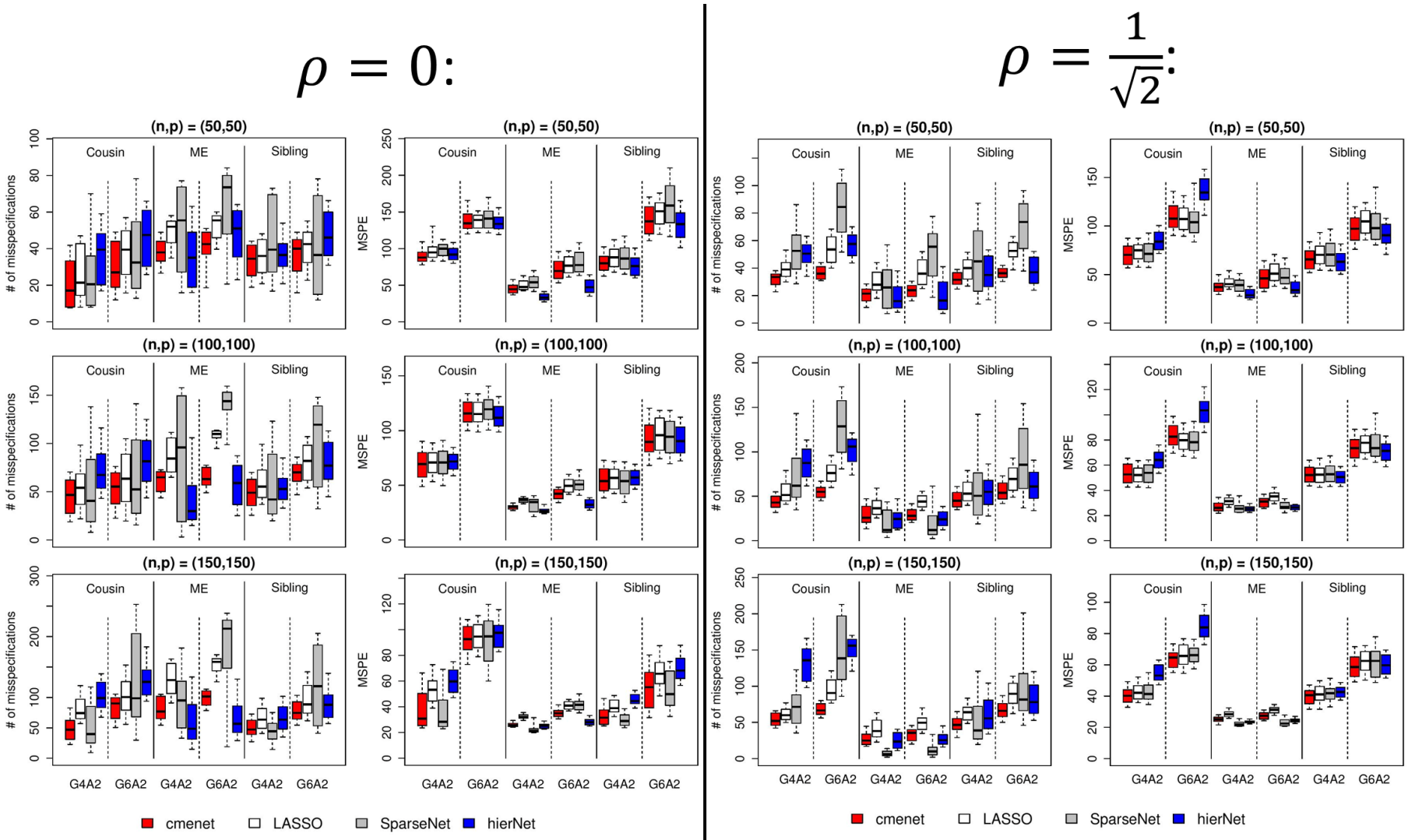

Figure 3.3: Boxplots of the 10%, 25%, 50%, 75% and 90% quantiles for the # of misspecifications and MSPE, with $(n, p) = (50, 50)$ (top), $(n, p) = (100, 100)$ (middle) and $(n, p) = (150, 150)$ (bottom), using a latent correlation of $\rho = 0$ (left) and $\rho = 1/\sqrt{2}$ (right).

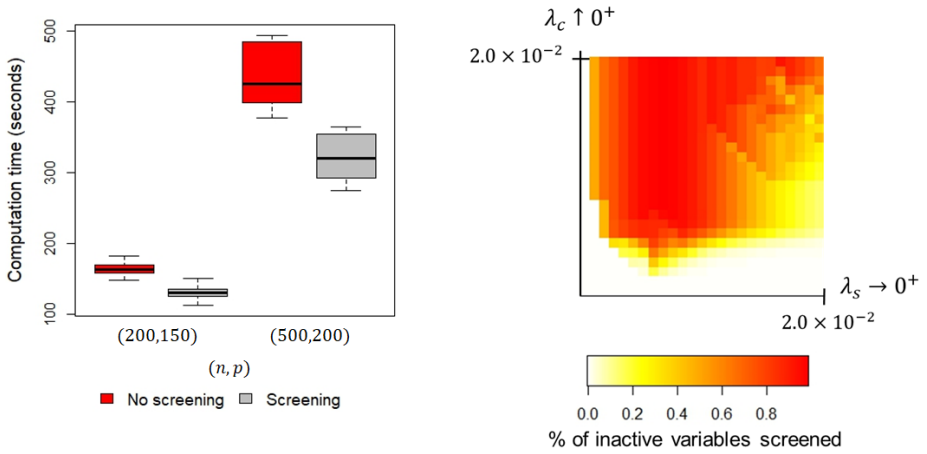

Figure 3.4: (Left) Boxplots of computation times for `cmenet` with $(n, p) = (200, 150)$ and $(500, 200)$; (Right) Proportion of inactive variables screened for $(n, p) = (200, 150)$.

`cmenet` is therefore most effective in applications where one expects some conditional effects to be active in the model; in other words, in applications where CMEs represent interpretable, domain-specific phenomena.

Finally, consider the results for $\rho = 1/\sqrt{2}$ (right part of Figure 3.3), where the underlying MEs are moderately correlated. For the sibling and cousin models, `cmenet` again provides an improvement in selection performance over the other three methods, with this improvement much greater than that for the uncorrelated setting $\rho = 0$. Such an observation is expected in light of Section 3.2.2, because the CME group structure is most prominent for moderate choices of ρ. For the main effect models, `cmenet` and `hierNet` again provide the best selection performance, with the relative performance of `cmenet` noticeably better than that for $\rho = 0$. This again can be explained by the more pronounced CME group structure for moderate ρ, which allows for more effective CME reduction. As before, the MSPE for `cmenet` is comparable to or better than the other three methods, which is as desired.

To numerically demonstrate the effectiveness of the CME screening rules in Section C.7, the left plot in Figure 3.4 shows the boxplots of the computation times for `cmenet` with $(n, p) = (200, 150)$ and $(500, 200)$, under a G2A6 sibling model with latent corre-

lation $\rho = 0$. We see that the proposed screening rules significantly reduce computation time, with over 20% reduction in median time for $(n, p) = (200, 150)$, and 30% reduction for $(n, p) = (500, 200)$. This effectiveness appears to grow for larger sample sizes n and greater number of main effects p, which is as desired. The right plot in Figure 3.4 shows the proportion of inactive variables removed by the screening procedure for $(n, p) = (200, 150)$ as a function of the sibling and cousin penalties λ_s and λ_c. We see that the proposed screening rules correctly remove a large proportion of inactive variables (over 80% for smaller λ_s and λ_c), which greatly speeds up the ensuing coordinate descent algorithm. In total, only 3 active variables were incorrectly screened over all values of (λ_s, λ_c) tested, and all such violations were corrected in post-convergence check of KKT conditions.

3.6 Polygenic association study on fly wing shape

In this section, we demonstrate the usefulness of `cmenet` for an important, real-world problem on polygenic association. Polygenes are a group of non-epistatic genes which serve as biological markers for many characteristics of interest called phenotypes (e.g., susceptibility to diabetes for youth [116] and major depressive disorders [117]), and the association of influential polygenes to particular phenotypes is an important area of research in the biomedical community. Here, we investigate the polygenic association for the wing shape of *Drosophila Melanogaster*, the common fruit fly.

The data employed here is collected from a study by [118], where the authors considered $p = 48$ homozygous (i.e., binary[4]) polygene markers on the second chromosome of *Drosophila Melanogaster* and its effect on fly wing shape, using $n = 701$ observations collected from recombinant isogenic lines. The response of interest is a continuous index

[4]For organisms with diploid cells (including *Drosophila Melanogaster*), chromosomes are found in pairs; these chromosome pairs can be further categorized as either *heterozygous* – meaning the pair contains different alleles for each gene, or *homozygous* – meaning the pair contains identical alleles for each gene. For alleles with levels + and –, heterozygous pairs allow for four allele combinations (+,+), (+,–), (–,+) and (–,–), while homozygous pairs allow for two combinations (+,+) and (–,–). For this fly wing study, [118] found very little heterozygous behavior on chromosome 2, and reported subsequent results using modified homozygous chromosomes, which are binary and fit within the framework of this chapter.

Table 3.3: Number of selected effects and some selected effects (p-values bracketed) from `cmenet` and `hierNet` in the gene association study of fly wing shape.

Method	*# of selected effects*	*Some selected effects (p-values)*	
`cmenet`	21	g14|g27- (6.1×10^{-4}), g14|g38+ (2.0×10^{-2}), g17|g14- (1.6×10^{-12}), g23|g14+ (2.5×10^{-30})	g45|g10+ (7.3×10^{-7})
`hierNet`	129	g14 (8.3×10^{-1})	g45 (1.5×10^{-1}), g45 $*$ g10 (8.1×10^{-1})

for wing shape, which incorporates both the width of the wing across the middle and the width across the base. As in simulation studies, our focus lies primarily on the *selection* of important CMEs, which here represents the effect of a gene conditional on another gene being active or absent. This is because the identification of these novel conditional effects yields valuable insight into the activation structure of gene-gene interactions, whereas the more traditional two-factor interaction analysis can be less interpretable in such a setting.

Here, we compare the analysis provided by `cmenet` with that from `hierNet`. As before, `cmenet` performs selection on MEs and CMEs ($p' = p + 4\binom{p}{2} = 4,560$ variables in total), while `hierNet` performs MEs and 2FIs ($p'' = p + \binom{p}{2} = 1,176$ variables in total). The purpose of such a comparison is to understand the practical advantages and disadvantages in employing the novel CMEs as basis functions, compared to the typical approach of using 2FIs for analyzing gene-gene interactions [119]. For brevity, we do not include either the LASSO or `SparseNet` selection of CMEs in this comparison, because it was already shown in Section 3.5 that `cmenet` enjoys better selection performance.

Consider first Table 3.3, which shows (a) the number of selected effects for `cmenet` and `hierNet`, and (b) some selected effects for each method, along with their corresponding p-values from a regular linear model fit. We see that the fitted model from `cmenet`, which has 21 selected effects, is much smaller than the model returned by `hierNet`, which has 129 selected effects. This model parsimony for `cmenet` suggests that there are indeed active CMEs for the problem at hand, i.e., there are certain polygenes which affect wing

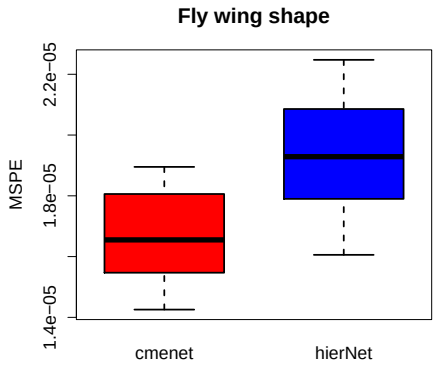

Figure 3.5: Boxplots of the 10%, 25%, 50%, 75% and 90% MSPE quantiles for `cmenet` and `hierNet` in the gene association study of fly wing shape.

shape *only* in the presence or absence of other polygenes. Taking a closer look at some of the selected effects for `cmenet` and `hierNet` from Table 3.3, two interesting insights can be observed on this conditional gene association structure. From the first column of selected effects, `hierNet` deemed the 14-th polygene g14 to be active, while `cmenet` instead selected the two sibling effects g14|g27- and g14|g38+, and the two cousin effects g17|g14- and g23|g14+. In other words, under traditional analysis, gene g14 is deemed influential in all situations, whereas the conclusion is more nuanced under the proposed CME analysis, with g14 influential (a) when gene g27 is absent or gene g38 is active, or (b) in inhibiting gene g17 or activating gene g23. The latter provides a more careful analysis of the signal from g14, and judging by the much smaller p-values for these conditional effects, also yields greater insight on the underlying gene activation structure. From the second column of selected effects, `hierNet` deemed both g45 and its interaction g45$*$g10 to be active, while `cmenet` selected only the CME g45|g10+. This nicely illustrates why `cmenet` provides parsimonious models: by selecting the CME g45|g10+ in place of its component ME g45 and 2FI g45$*$g10, we obtain a smaller model with considerably smaller p-values, which is as desired (this is akin to Rule 1 of [92] for selecting CMEs in designed experiments; see Section 3.2.1, especially equation (3.1)).

Consider next Figure 3.5, which shows the MSPE boxplots for `cmenet` and `hierNet` in predicting the continuous wing shape index. Here, MSPE is estimated by randomly

sampling 80% of the data for model training, then using the remaining 20% to test the trained model; this procedure is then repeated 200 times to provide error variability. We see that `cmenet` enjoys considerable improvements over `hierNet` in terms of MSPE, yielding at least a 12% reduction at all five error quantiles. This again reaffirms the likely conditional nature of the underlying polygenic association structure, with certain polygenes active only in the presence or absence of other polygenes.

To summarize, this gene association study highlights two important advantages of `cmenet`. First, in applications where CMEs are interpretable phenomena, the proposed selection method can provide much more parsimonious models compared to traditional analysis using two-way interactions, and can yield greater insight on the underlying problem of interest. This is particularly true in genetic applications, where selected CMEs can be used to further investigate *why* some genes are conditionally active, and *why* some play a more supportive role in *activating* or *inhibiting* other genes. Second, when CMEs have natural domain-specific interpretations, using such effects as basis functions can also improve the predictive performance of the fitted model as well.

3.7 Conclusion and future work

In this chapter, a new method is presented for selecting binary variables and a set of reparametrized variables called conditional main effects (CMEs) from observation data. While CMEs are intuitive basis functions with appealing interpretations in many applications, existing selection methods can perform poorly due to the inherent grouped structure of these effects. We proposed a novel selection method called `cmenet`, which accounts for this underlying grouped structure using two selection principles called CME coupling and reduction; the former allows CMEs to more easily enter the model given selected siblings or cousins, and the latter encourages the selection of the underlying ME given many selected siblings or cousins. A coordinate descent algorithm is then introduced for minimizing the selection criterion, and several computational tools are proposed for efficient optimization

and parameter tuning in high-dimensions. Simulation studies showed considerable improvements for `cmenet` over existing methods with respect to selection accuracy. Applied to a real-world gene association study on fly wing shape, the proposed method provides not only improved predictive performance over the standard two-way interaction analysis, but also a more parsimonious and interpretable model which reveals important insights on gene activation behavior.

Given the positive results here, there are many exciting avenues for future work. First, in the high-dimensional setting of $p \gg 1$, the tuning of the four selection parameters in $Q(\boldsymbol{\beta})$ can be computationally expensive due to the grid structure of feasible parameter combinations in `cv.cmenet`. With recent advances on the topic of optimal designs for convex spaces (e.g., [120, 61]), it may be interesting to see whether the use of such designs as candidate settings allows for more efficient parameter tuning. Second, we are working to broaden the proposed methodology to higher-order conditional effects, e.g., the effect of A conditional on both $B+$ and $C+$. The main challenge here is again computational efficiency, but such a direction would enable the investigation of, say, more complex activation phenomena in the earlier gene study. Lastly, we are interested in extending the current framework for selecting the continuous CMEs mentioned earlier in Section 3.4.1. This would allow the proposed methodology to be applicable for more general datasets, and we look forward to exploring this in future research.

CHAPTER 4
AN EFFICIENT SURROGATE MODEL FOR EMULATION AND PHYSICS EXTRACTION OF LARGE EDDY SIMULATIONS

4.1 Introduction

In the quest for designing advanced propulsion and power-generation systems, there is an increasing need for an effective methodology that combines engineering physics, computer simulations and statistical modeling. A key point of interest in this design process is the treatment of turbulence flows, a subject that has far-reaching scientific and technological importance [121]. Turbulence refers to the irregular and chaotic behavior resulting from motion of a fluid flow [122], and is characterized by the formation of eddies and vortices which transfer flow kinetic energy due to rotational dynamics. Such a phenomenon is an unavoidable aspect of everyday life, present in the earth's atmosphere and ocean waves, and also in chemically reacting flows in propulsion and power-generation devices. In this chapter, we develop a surrogate model, or emulator, for predicting turbulent flows in a swirl injector, a mechanical component with a wide variety of engineering applications.

There are two reasons why a statistical model is required for this important task. First, the time and resources required to develop an effective engineering device with desired functions may be formidable, even at a *single* design setting. Second, even with the availability of high-fidelity simulation tools, the computational resources needed can be quite costly, and only a handful of design settings can be treated in practical times. For example, the flow simulation of a single injector design takes over 6 days of computation time, parallelized using 200 CPU cores. For practical problems with large design ranges and/or many

The paper based on this chapter will appear in *Journal of the American Statistical Association*.

design inputs, the use of only high-fidelity simulations is insufficient for surveying the full design space. In this setting, emulation provides a powerful tool for efficiently predicting flows at any design geometry, using a small number of flow simulations as training data. A central theme of this chapter is that, by properly *eliciting* and *applying* physical properties of the fluid flow, simplifying assumptions can be made on the emulator which greatly reduce computation and improve prediction accuracy. In view of the massive simulation datasets, which can exceed many gigabytes or even terabytes in storage, such efficiency is paramount for the usefulness of emulation in practice.

The proposed emulator utilizes a popular technique called *kriging* [123], which employs a Gaussian Process (GP) for modeling computer simulation output over a desired input domain. The main appeal of kriging lies in the fact that both the emulation predictor and its associated uncertainty can be evaluated in closed-form. For our application, a kriging model is required which can predict flows at any injector geometry setting; we refer to this as *flow kriging* for the rest of the chapter. In recent years, there have been important developments in flow kriging, including the works of [124] and [125] on regular spatial grids (i.e., outputs are observed at the same spatial locations over all simulations), and [126] on irregular grids. Unfortunately, it is difficult to apply these models to the more general setting in which the *dimensions* of spatial grids vary greatly for different input variables. In the present work, for instance, the desired design range for injector length varies from 20 mm to 100 mm. Combined with the high spatial and temporal resolutions required in simulation, the resulting flow data is much too large to process using existing models, and data-reduction methods are needed.

There has been some work on using reduced-basis models to compact data for emulation, including the functional linear models by [127], wavelet models by [128] and principal component models by [129] and [130]. Here, we employ a generalization of the latter method called *proper orthogonal decomposition (POD)* [131], which is better known in statistical literature as the Karhunen-Loève decomposition [132, 133]. From a flow physics

perspective, POD separates a simulated flow into key instability structures, each with its corresponding spatial and dynamic features. Such a decomposition is, however, inappropriate for emulation, because there is no way to connect the extracted instabilities of one input setting to the instabilities of another setting. To this end, we propose a new method called the *common POD* (CPOD) to extract *common* instabilities over the design space. This technique exploits a simple and physically justifiable linearity assumption on the spatial distribution of instability structures.

In addition to efficient flow emulation, our model also provides two important features. First, the same domain-specific model simplifications (e.g., on the spatio-temporal correlation structure) which enable efficient prediction also allow for an efficient uncertainty quantification (UQ) for such a prediction. This UQ is highly valuable in practice, since the associated uncertainties for variable disturbance propagations can then be used for mitigating flow instabilities [134]. Second, by incorporating known properties of the fluid flow into the model, the proposed emulator can in turn provide valuable insights on the dominant physics present in the system, which can then be used to guide further scientific investigations. One key example of this is the learning of dominant flow coupling mechanisms using a large co-kriging model [135, 12] under sparsity constraints.

The chapter is structured as follows. Section 4.2 provides a brief overview of the physical model of concern, including injector design, governing equations and experimental design. Section 4.3 introduces the proposed emulator model, and proposes a parallelized algorithm for efficient parameter estimation. Section 4.4 presents the emulation prediction and UQ for a new injector geometry, and interprets important physical correlations extracted by the emulator. Section 5 concludes with directions for future work.

4.2 Injector schematic and large eddy simulations

We first describe the design schematic for the swirl injector of concern, then briefly outline the governing partial differential equations and simulation tools. A discussion on experi-

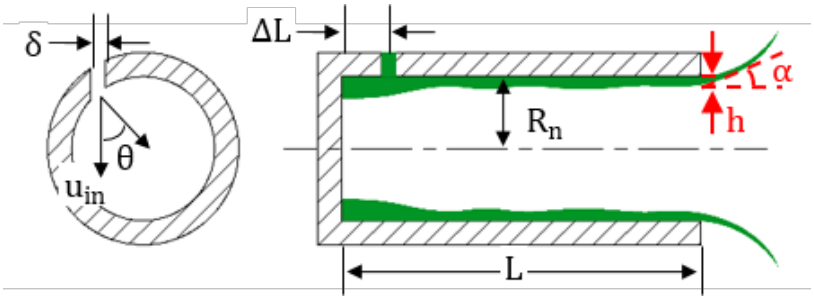

Figure 4.1: Schematic of injector configuration.

Table 4.1: Range of geometric parameters.

Parameter	**Range**
L	20 mm - 100 mm
R_n	2.0 mm - 5.0 mm
δ	0.5 mm - 2.0 mm
θ	$45° - 75°$
ΔL	1.0 mm - 4.0 mm

mental design is provided at the end of this section.

4.2.1 Injector design

Figure 4.1 shows a schematic of the swirl injector under consideration. It consists of an open-ended cylinder and a row of tangential entries for liquid fluid injection. The configuration is typical of many propulsion and power-generation applications [136, 137, 138]. Liquid propellant is tangentially introduced into the injector and forms a thin film attached to the wall due to the swirl-induced centrifugal force. A low-density gaseous core exists in the center region in accordance with conservation of mass and angular momentum. The liquid film exits the injector as a thin sheet and mixes with the ambient gas. The swirl injection and atomization process involves two primary mechanisms: disintegration of the liquid sheet as it swirls and stretches, and sheet breakup due to the interaction with the surroundings. The design of the injector significantly affects the atomization characteristics and stability behaviors.

Figure 4.1 shows the five design variables considered for injector geometry: the injector length L, the nozzle radius R_n, the inlet diameter δ, the injection angle θ, and the distance between inlet and head-end ΔL. From flow physics, these five variables are influential for liquid film thickness h and spreading angle α (see Figure 4.1), which are key measures of injector performance of a swirl injector. For example, a larger injection angle θ induces greater swirl momentum in the liquid oxygen flow, which in turn causes thinner film thickness and smaller spreading angle. Table 4.1 summarizes the design ranges for these five variables. To ensure the applicability of our work, broad geometric ranges are considered, covering design settings for several existing rocket injectors. Specifically, the range for injector length L covers the length of RD-0110 and RD-170 liquid-fuel rocket engines.

4.2.2 Flow simulation

The numerical simulations here are performed with a pressure of 100 atm, which is typical of contemporary liquid rocket engines with liquid oxygen (LOX) as the propellant. The physical processes modeled here are turbulent flows, in which various sizes of turbulent eddies are involved. A direct numerical simulation to resolve all eddy length-scales is computationally prohibitive. To this end, we employ the large eddy simulation (LES) technique, which directly simulates large turbulent eddies and employs a model-based approach for small eddies. To provide initial turbulence, broadband Gaussian noise is superimposed onto the inlet velocity components. Thermodynamic and transport properties are simulated using the techniques in [139] and [140]; the theoretical and numerical framework can be found in [141] and [142]. To optimize computational speed, a multi-block domain decomposition technique combined with the message-passing interface for parallel computing is applied. Each LES simulation takes 6 days of computation time, parallelized over 200 CPU cores, to obtain $T = 1,000$ snapshots with a time-step of 0.03 ms after the flow reaches statistically stationary state. From this, six flow variables of interest can be extracted: axial (u), radial (v), and circumferential (w) components of velocity, temperature (T), pressure

Table 4.2: Elicited flow physics and corresponding assumptions for the emulator model.

Flow physics	**Model assumption**
Coherent structures in turbulent flow [131]	POD-based kriging
Similar Reynolds numbers for cold-flows [145]	Linear-scaling modes in CPOD
Dense simulation time-steps	Time-independent emulator
Couplings between flow variables [122]	Co-kriging framework with covariance matrix $\mathbf{T}$
Few-but-significant couplings [122]	Sparsity on $\mathbf{T}^{-1}$

(P) and density (ρ).

Numerical simulations are conducted for $n = 30$ injector geometries in the timeframe set for this project. These simulation runs are allocated over the design space in Table 4.1 using the maximum projection (MaxPro) design proposed by [69]. Compared to Latin-hypercube-based designs (e.g., [143], [144]), MaxPro designs enjoy better space-filling properties in all possible projections of the design space, and also provide better predictions for GP modeling. While $n = 30$ simulation runs may appear to be too small of a dataset for training the proposed flow emulator, we show this sample size can provide accurate flow predictions for the application at hand, through an elicitation of flow physics and the incorporation of such physics into the model. For these 30 runs, one issue which arises is that the simulation data is massive, requiring nearly a hundred gigabytes in computer storage. For such large data, a blind application of existing flow kriging methods may require weeks for flow prediction, which entirely defeats the purpose of emulation, because simulated flows can generated in 6 days. Again, by properly eliciting and incorporating physics as simplifying assumptions for the emulator model, accurate flow predictions can be achieved in hours despite a limited run size. We elaborate on this elicitation procedure in the following section.

4.3 Emulator model

We first introduce the new idea of CPOD, then present the proposed emulator model and a parallelized algorithm for parameter estimation. A key theme in this section (and indeed, for this chapter) is the elicitation and incorporation of flow physics within the emulator model. This not only allows for efficient and accurate flow predictions through simplifying model assumptions, but also provides a data-driven method for extracting useful flow physics, which can then guide future experiments. As demonstrated in Section 4, both objectives can be achieved despite limited runs and complexities inherent in flow data. Table 4.2 summarizes the elicited flow physics and the corresponding emulator assumptions; we discuss each point in greater detail below.

4.3.1 Common POD

A brief overview of POD is first provided, following [131]. For a *fixed* injector geometry, let $Y(\mathbf{x}, t)$ denote a flow variable (e.g., pressure) at spatial coordinate $\mathbf{x} \in \mathbb{R}^2$ and flow time t. POD provides the following decomposition of $Y(\mathbf{x}, t)$ into separable spatial and temporal components:

$$Y(\mathbf{x}, t) = \sum_{k=1}^{\infty} \beta_k(t)\phi_k(\mathbf{x}), \tag{4.1}$$

with the spatial eigenfunctions $\{\phi_k(\mathbf{x})\}_{k=1}^{\infty}$ and temporal coefficients $\{\beta_k(t)\}_{k=1}^{\infty}$ given by:

$$\phi_k(\mathbf{x}) = \underset{\substack{\|\psi\|_2=1,\\ \langle\psi,\phi_l\rangle=0,\forall l<k}}{\operatorname{argmax}} \int \left\{ \int Y(\mathbf{x}, t)\psi(\mathbf{x})\, d\mathbf{x} \right\}^2 dt, \quad \beta_k(t) = \int Y(\mathbf{x}, t)\phi_k(\mathbf{x})\, d\mathbf{x}. \tag{4.2}$$

Following [146], we refer to $\{\phi_k(\mathbf{x})\}_{k=1}^{\infty}$ as the *spatial POD modes* for $Y(\mathbf{x}, t)$, and its corresponding coefficients $\{\beta_k(t)\}_{k=1}^{\infty}$ as *time-varying coefficients*.

There are two key reasons for choosing POD over other reduced-basis models. First, one can show [133] that any truncated representation in (4.1) gives the best flow reconstruc-

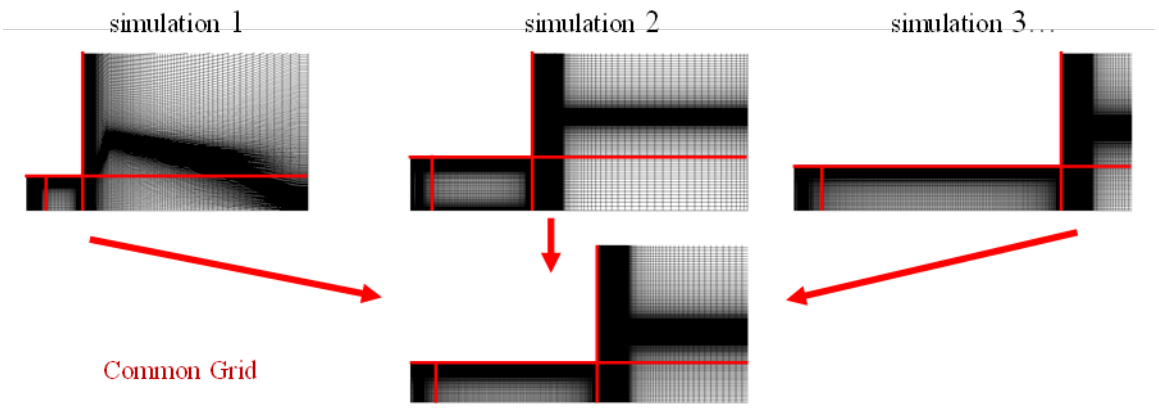

Figure 4.2: Common grid using linearity assumption for CPOD.

tion of $Y(\mathbf{x}, t)$ in L_2-norm, compared to any other linear expansion of space/time products with the same number of terms. This property is crucial for our application, since it allows the massive simulation data to be optimally reduced to a smaller training dataset for the proposed emulator. Second, the POD has a special interpretation in terms of turbulent flow. In the seminal paper by [131], it is shown that, under certain conditions, the expansion in (4.1) can extract *physically meaningful* coherent structures which govern turbulence instabilities. For this reason, physicists use POD as an experimental tool to pinpoint key flow instabilities, simply through an inspection of $\phi_k(\mathbf{x})$ and the dominant frequencies in $\beta_k(t)$. For example, using POD analysis, [136] showed that the two flow phenomena, hydrodynamic wave propagation on LOX film and vortex core excitation near the injector exit, are the key mechanisms driving flow instability. This is akin to the use of principal components in regression, which can yield meaningful results in applications where such components have innate interpretability.

Unfortunately, POD is only suitable for extracting instability structures at a *single* geometry, whereas for emulation, a method is needed that can extract common structures over *varying* geometries. With this in mind, we propose a new decomposition called common POD (CPOD). The key assumption of CPOD is that, under a *physics-guided partition* of the computational domain, the spatial distribution of coherent structures *scales linearly* over varying injector geometries. For cold flows, this can be justified by similar Reynolds

numbers (which characterize flow dynamics) over different geometries [145]. This is one instance of model simplification through elicitation, because such a property likely does not hold for general flows. This linearity assumption is highly valuable for computational efficiency, because flows from different geometries can then be rescaled onto a common spatial grid for instability extraction. Figure 4.2 visualizes this procedure. The grids for each simulation are first split into four parts: from injector head-end to the inlet, from the inlet to the nozzle exit, and the top and bottom portions of the downstream region. Each part is then proportionally rescaled to a common, reference grid according to changes in the geometric variables L, R_n and ΔL (see Figure 4.1). From a physics perspective, such a partition is necessary for the linearity assumption to hold.

Stating this mathematically, let $\mathbf{c}_1, \cdots, \mathbf{c}_n \in \mathbb{R}^p$ be the n simulated geometries, let $Y(\mathbf{x}, t; \mathbf{c}_i)$ be the simulated flow at setting $\mathbf{c}_i$, and fix some setting $\mathbf{c} \in \{\mathbf{c}_i\}_{i=1}^n$ as the geometry for the common grid. Next, define $\mathscr{M}_i : \mathbb{R}^2 \to \mathbb{R}^2$ as the linear map which rescales spatial modes on the common geometry $\mathbf{c}$ back to the i-th simulated geometry $\mathbf{c}_i$ according to geometric changes in L, R_n and ΔL. $\mathscr{M}_i$ can be viewed as the inverse map of the procedure described in the previous paragraph and visualized in Figure 4.2, which rescales modes from $\mathbf{c}_i$ to the common geometry $\mathbf{c}$ (see Appendix A.1 for details). CPOD provides the following decomposition of $Y(\mathbf{x}, t; \mathbf{c}_i)$:

$$Y(\mathbf{x}, t; \mathbf{c}_i) = \sum_{k=1}^{\infty} \beta_k(t; \mathbf{c}_i) \mathscr{M}_i\{\phi_k(\mathbf{x})\}, \tag{4.3}$$

with the spatial CPOD modes $\{\phi_k(\mathbf{x})\}$ and time-varying coefficients $\{\beta_k(t; \mathbf{c}_i)\}$ defined as:

$$\begin{aligned} \phi_k(\mathbf{x}) &= \underset{\substack{\|\psi\|_2=1, \\ \langle \psi, \phi_l \rangle = 0, \forall l < k}}{\operatorname{argmax}} \sum_{i=1}^{n} \int \left\{ \int Y(\mathbf{x}, t; \mathbf{c}_i) \mathscr{M}_i\{\psi(\mathbf{x})\} \, d\mathbf{x} \right\}^2 dt, \\ \beta_k(t; \mathbf{c}_i) &= \int Y(\mathbf{x}, t; \mathbf{c}_i) \mathscr{M}_i\{\phi_k(\mathbf{x})\} \, d\mathbf{x}. \end{aligned} \tag{4.4}$$

Here, $\phi_k(\mathbf{x})$ is the spatial distribution for the k-th common flow structure, with $\beta_k(t; \mathbf{c}_i)$ its

time-varying coefficient for geometry $\mathbf{c}_i$. As in POD, leading terms in CPOD can also be interpreted in terms of flow physics, a property we demonstrate later in Section 4.4. CPOD therefore not only provides optimal data-reduction for the simulation data, but also extracts physically meaningful structures which can then be incorporated for emulation.

Algorithmically, the CPOD expansion can be computed by rescaling and interpolating all flow simulations to the common grid, computing the POD expansion, and then rescaling the resulting modes back to their original grids. Interpolation is performed using the inverse distance weighting method in [147], and can be justified by dense spatial resolution of the data (with around 100,000 grid points for each simulation). Letting T be the total number of time-steps, a naive implementation of this decomposition requires $O(n^3T^3)$ work, due to a singular-value-decomposition (SVD) step. Such a decomposition therefore becomes computationally intractable when the number of runs grows large or when simulations have dense time-steps (as is the case here). To avoid this computational issue, we use an iterative technique from [148] called the implicitly restarted Arnoldi method, which approximates leading terms in (4.3) using periodically restarted Arnoldi decompositions. The full algorithm for CPOD is outlined in Appendix A.

4.3.2 Model specification

After the CPOD extraction, the extracted time-varying coefficients $\{\beta_k(t;\mathbf{c}_i)\}_{i,k}$ are then used as data for fitting the proposed emulator. There has been some existing work on dynamic emulator models, such as [149], [150] and [151], but the sheer number of simulation time-steps here can impose high computation times and numerical instabilities for these existing methods [126]. As mentioned previously, computational efficiency is paramount for our problem, since simulation runs can be performed within a week. Moreover, existing emulators cannot account for cross-correlations between different dynamic systems, while the flow physics represented by different CPOD modes are known to be highly coupled from governing equations. Here, we exploit the dense temporal resolution of the flow by

using a *time-independent (TI)* emulator that employs independent kriging models at each slice of time. The rationale is that, because time-scales are so fine, there is no practical need to estimate temporal correlations (even when they exist), since prediction is not required between time-steps. This time-independent simplification is key for emulator efficiency, since it allows us to fully exploit the power of parallel computing for model fitting and flow prediction.

The model is as follows. Suppose R flow variables are considered (with $R = 6$ in the present case), and the CPOD expansion in (4.3) is truncated at K_r terms for flow $r = 1, \cdots, R$. Let $\boldsymbol{\beta}^{(r)}(t;\mathbf{c}) = (\beta_1^{(r)}(t;\mathbf{c}), \cdots, \beta_{K_r}^{(r)}(t;\mathbf{c}))^T$ be the vector of K_r time-varying coefficients for flow variable r at design setting $\mathbf{c}$, with $\boldsymbol{\beta}(t;\mathbf{c}) = (\boldsymbol{\beta}^{(1)}(t;\mathbf{c})^T, \cdots, \boldsymbol{\beta}^{(R)}(t;\mathbf{c})^T)^T$ the coefficient vector for all flows at $\mathbf{c}$. We assume the following *time-independent GP model* on $\boldsymbol{\beta}(t;\mathbf{c})$:

$$\boldsymbol{\beta}(t;\mathbf{c}) \sim GP\{\boldsymbol{\mu}(t), \boldsymbol{\Sigma}(\cdot,\cdot;t)\}, \quad \boldsymbol{\beta}(t;\mathbf{c}) \perp \boldsymbol{\beta}(t';\mathbf{c}) \text{ for } t \neq t'. \tag{4.5}$$

Here, $K = \sum_{r=1}^{R} K_r$ is the number of extracted modes over all R flow variables, $\boldsymbol{\mu} \in \mathbb{R}^K$ is the process mean vector, and $\boldsymbol{\Sigma}(\cdot,\cdot) : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^{K\times K}$ its corresponding covariance matrix function defined below. Since the GPs are now time-independent, we present the specification for *fixed* time t, and refer to $\boldsymbol{\beta}(t;\mathbf{c})$, $\boldsymbol{\mu}(t)$ and $\boldsymbol{\Sigma}(\cdot,\cdot;t)$ as $\boldsymbol{\beta}(\mathbf{c})$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}(\cdot,\cdot)$ for brevity.

For computational efficiency, the following separable form is assumed for $\boldsymbol{\Sigma}(\cdot,\cdot)$:

$$\boldsymbol{\Sigma}(\mathbf{c}_1,\mathbf{c}_2) = r_\tau(\mathbf{c}_1,\mathbf{c}_2)\mathbf{T}, \quad r_\tau(\mathbf{c}_1,\mathbf{c}_2) = \prod_{j=1}^{p} \tau_j^{4(c_{1j}-c_{2j})^2}, \quad \mathbf{c}_1,\mathbf{c}_2 \in \mathbb{R}^p, \quad \tau_j \in (0,1), \tag{4.6}$$

where $\mathbf{T} \in \mathbb{R}^{K\times K}$ is a symmetric, positive definite matrix called the *CPOD covariance matrix*, and $r_\tau(\cdot,\cdot)$ is the correlation function over the design space, parameterized by $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_p)^T \in (0,1)^p$. This can be viewed as a large co-kriging model [135] over the design space, with the multivariate observations being the extracted CPOD coefficients

for all flow variables. Note that r_τ is a reparametrization of the squared-exponential (or Gaussian) correlation function $\exp\{-\sum_{j=1}^{p}\theta_j(c_{1j}-c_{2j})^2\}$, with $\theta_j = -4\log\tau_j$. In our experience, such a reparametrization allows for a more numerically stable optimization of MLEs, because the optimization domain $\tau_j \in (0,1)$ is now bounded. Our choice of the Gaussian correlation is also well-justified for the application at hand, since fully-developed turbulence dynamics are known to be relatively smooth.

Suppose simulations are run at settings $\mathbf{c}_1, \cdots, \mathbf{c}_n$, and assume for now that model parameters are known. Invoking the conditional distribution of the multivariate normal distribution, the time-varying coefficients at a new setting $\mathbf{c}_{new}$ follow the distribution:

$$\begin{aligned}\boldsymbol{\beta}(\mathbf{c}_{new})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n \sim \mathcal{N}\Bigg(&\boldsymbol{\mu} + (\mathbf{T}\otimes\mathbf{r}_{\tau,new})^T\left(\mathbf{T}^{-1}\otimes\mathbf{R}_\tau^{-1}\right)(\boldsymbol{\beta}-\mathbf{1}_n\otimes\boldsymbol{\mu}),\\ &\mathbf{T} - (\mathbf{T}\otimes\mathbf{r}_{\tau,new})^T\left(\mathbf{T}^{-1}\otimes\mathbf{R}_\tau^{-1}\right)(\mathbf{T}\otimes\mathbf{r}_{\tau,new})\Bigg),\end{aligned} \tag{4.7}$$

where $\mathbf{r}_{\tau,new} = (r_\tau(\mathbf{c}_{new},\mathbf{c}_1),\cdots,r_\tau(\mathbf{c}_{new},\mathbf{c}_n))^T$ and $\mathbf{R}_\tau = [r_\tau(\mathbf{c}_i,\mathbf{c}_j)]_{i=1}^n{}_{j=1}^n$. Using algebraic manipulations, the minimum-MSE (MMSE) predictor for $\boldsymbol{\beta}(\mathbf{c}_{new})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n$ and its corresponding variance is given by

$$\hat{\boldsymbol{\beta}}(\mathbf{c}_{new}) = \boldsymbol{\mu} + \left((\mathbf{r}_{\tau,new}^T\mathbf{R}_\tau^{-1})\otimes\mathbf{I}_K\right)(\boldsymbol{\beta}-\mathbf{1}_n\otimes\boldsymbol{\mu}), \mathbb{V}\{\boldsymbol{\beta}(\mathbf{c}_{new})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n\} = \left(1-\mathbf{r}_{\tau,new}^T\mathbf{R}_\tau^{-1}\mathbf{r}_{\tau,new}\right)\mathbf{T}, \tag{4.8}$$

where $\mathbf{I}_K$ and $\mathbf{1}_n$ denote a $K\times K$ identity matrix and a 1-vector of n elements, respectively. Substituting this into the CPOD expansion (4.3), the predicted r-th flow variable becomes:

$$\hat{Y}^{(r)}(\mathbf{x},t;\mathbf{c}_{new}) = \sum_{k=1}^{K_r}\hat{\beta}_k^{(r)}(\mathbf{c}_{new})\mathscr{M}_{new}\{\phi_k^{(r)}(\mathbf{x})\}, \tag{4.9}$$

with the associated spatio-temporal variance:

$$\mathbb{V}\{Y^{(r)}(\mathbf{x},t;\mathbf{c}_{new})|\{Y^{(r)}(\mathbf{x},t;\mathbf{c}_i)\}_{i=1}^n\} = \sum_{k=1}^{K_r}\mathbb{V}\{\beta_k^{(r)}(\mathbf{c}_{new})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n\}\left[\mathscr{M}_{new}\{\phi_k^{(r)}(\mathbf{x})\}\right]^2, \tag{4.10}$$

where $\phi_k^{(r)}(\mathbf{x})$ is the k-th CPOD mode for flow variable r. This holds because the CPOD

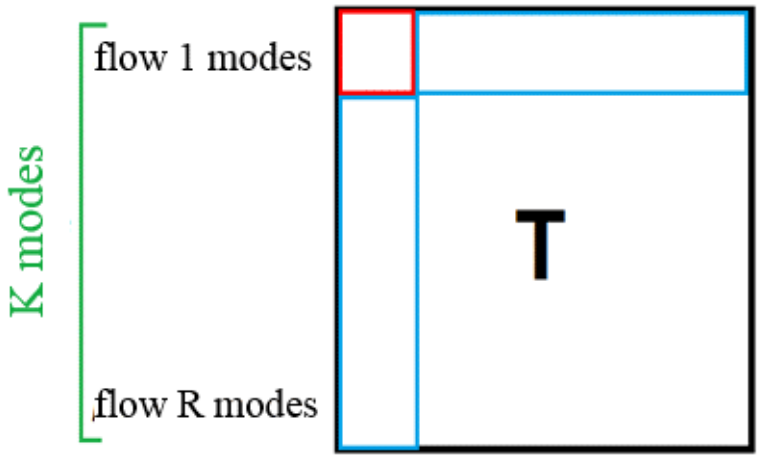

Figure 4.3: Illustration of the CPOD correlation matrix $\mathbf{T}$. Red indicates a diagonal matrix, while blue indicates non-diagonal entries.

modes for a fixed flow variable are orthogonal (see Section 3.1).

It is worth noting that, when model parameters are known, the MMSE predictor in (4.8) from the proposed co-kriging model (which we call M_A) is the same as the MMSE predictor from the simpler *independent* GP model with $\mathbf{T}$ diagonal (which we call M_0). One advantage of the co-kriging model M_A, however, is that it provides improved UQ compared to the independent model M_0, as we show below. Moreover, the MMSE predictor for a derived function g of the flow can be quite different between M_A and M_0. This is demonstrated in the study of turbulent kinetic energy in Section 4.3.

CPOD covariance matrix

We briefly describe why the CPOD covariance matrix $\mathbf{T}$ is appealing from both a physical and a statistical perspective. From the underlying governing equations, it is well known that certain dynamic behaviors are strongly *coupled* for different flow variables [122]. For example, pressure oscillation in the form of acoustic waves within an injector can induce velocity and density fluctuations. In this sense, $\mathbf{T}$ incorporates knowledge of these physical couplings within the emulator itself, with $\mathbf{T}_{ij} \gg 0$ indicating the presence of a significant

coupling between modes i and j, and vice versa. The covariance selection and estimation of $\mathbf{T}$ therefore provide a data-driven way to *extract* and *rank* significant flow couplings, which is of interest in itself and can be used to guide further experiments. Note that the block submatrices of $\mathbf{T}$ corresponding to the same flow variable (marked in red in Figure 4.3) should be diagonal, by the orthogonality of CPOD modes.

The CPOD covariance matrix $\mathbf{T}$ also plays an important statistical role in emulation. Specifically, when significant cross-correlations exist between modes (which we know to be true from the flow couplings imposed by governing equations), the incorporation of this correlation structure within our model ought to provide a more accurate quantification of uncertainty. This is indeed true, and is made precise by the following theorem.

Theorem 19. *Consider the two models* $M_0 : \boldsymbol{\beta}(\mathbf{c}) \in \mathbb{R}^K \sim GP\{\boldsymbol{\mu}, \boldsymbol{\Sigma}^{(0)}\}$ *and* $M_A : \boldsymbol{\beta}(\mathbf{c}) \sim GP\{\boldsymbol{\mu}, \boldsymbol{\Sigma}^{(A)}\}$*, where* $\boldsymbol{\Sigma}^{(0)}(\mathbf{c}_1, \mathbf{c}_2) = r_\tau(\mathbf{c}_1, \mathbf{c}_2)\mathbf{D}$ *and* $\boldsymbol{\Sigma}^{(A)}(\mathbf{c}_1, \mathbf{c}_2) = r_\tau(\mathbf{c}_1, \mathbf{c}_2)\mathbf{T}$ *with* $\mathbf{T} \succeq 0$ *and* $\mathbf{D} = \text{diag}\{\mathbf{T}\}$*. Let* C_0 *be the* $100(1-\alpha)\%$ *highest-density confidence region (HDCR, see [152]) of* $\boldsymbol{\beta}(\mathbf{c}_{new})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n$ *under* M_0*. Suppose* $\lambda_{min}(\mathbf{T}^{1/2}\mathbf{D}^{-1}\mathbf{T}^{1/2}) > 1$*. Then:*

$$\mathbb{P}\left\{\boldsymbol{\beta}(\mathbf{c}_{new}) \in C_0 | M_A, \{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n\right\} < 1 - \alpha.$$

Proof. For brevity, let $\boldsymbol{\beta} \equiv \boldsymbol{\beta}(\mathbf{c}_{new})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n$, and let $\hat{\boldsymbol{\beta}} \equiv \mathbb{E}[\boldsymbol{\beta}(\mathbf{c}_{new})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n]$. Letting $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$, it is easy to show that

$$\boldsymbol{\beta} - \hat{\boldsymbol{\beta}} | M_0 \sim \mathcal{N}\left\{\mathbf{0}, \left(1 - \mathbf{r}_{\tau,new}^T \mathbf{R}_\tau^{-1} \mathbf{r}_{\tau,new}\right)\mathbf{D}\right\} \overset{d}{=} \sqrt{1 - \mathbf{r}_{\tau,new}^T \mathbf{R}_\tau^{-1} \mathbf{r}_{\tau,new}}\,\mathbf{D}^{1/2}\mathbf{Z}, \quad \text{and}$$

$$\boldsymbol{\beta} - \hat{\boldsymbol{\beta}} | M_A \sim \mathcal{N}\left\{\mathbf{0}, \left(1 - \mathbf{r}_{\tau,new}^T \mathbf{R}_\tau^{-1} \mathbf{r}_{\tau,new}\right)\mathbf{T}\right\} \overset{d}{=} \sqrt{1 - \mathbf{r}_{\tau,new}^T \mathbf{R}_\tau^{-1} \mathbf{r}_{\tau,new}}\,\mathbf{T}^{1/2}\mathbf{Z}.$$

Under the independent model M_0, the $100(1-\alpha)\%$ HDCR becomes:

$$C_0 = \{\boldsymbol{\xi} \,:\, \left(1 - \mathbf{r}_{\tau,new}^T \mathbf{R}_\tau^{-1} \mathbf{r}_{\tau,new}\right)^{-1} (\boldsymbol{\xi} - \hat{\boldsymbol{\beta}})^T \mathbf{D}^{-1} (\boldsymbol{\xi} - \hat{\boldsymbol{\beta}}) \leq \chi_K^2(1-\alpha)\},$$

where $\chi^2_K(1-\alpha)$ be the $(1-\alpha)$-quantile of a χ^2-distribution with K degrees of freedom. Now, let λ_{min} denote the minimum eigenvalue of $\mathbf{T}^{1/2}\mathbf{D}^{-1}\mathbf{T}^{1/2}$. It follows that

$$\begin{aligned}
\mathbb{P}\left(\boldsymbol{\beta} \in C_0 | M_A\right) &= \mathbb{P}\left\{(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})^T\mathbf{D}^{-1}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}) \leq \left(1-\mathbf{r}^T_{\tau,new}\mathbf{R}^{-1}_{\tau}\mathbf{r}_{\tau,new}\right)\chi^2_K(1-\alpha)\middle| M_A\right\} \\
&= \mathbb{P}\left\{\mathbf{Z}^T(\mathbf{T}^{1/2}\mathbf{D}^{-1}\mathbf{T}^{1/2})\mathbf{Z} \leq \chi^2_K(1-\alpha)\right\} \\
&\leq \mathbb{P}\left\{\mathbf{Z}^T\mathbf{Z} \leq \lambda^{-1}_{min}\chi^2_K(1-\alpha)\right\},
\end{aligned}$$

since $\mathbf{Z}^T(\mathbf{T}^{1/2}\mathbf{D}^{-1}\mathbf{T}^{1/2})\mathbf{Z} \geq \lambda_{min}\mathbf{Z}^T\mathbf{Z}$ almost surely. The asserted result follows because $\mathbb{P}\left\{\mathbf{Z}^T\mathbf{Z} \leq \lambda^{-1}_{min}\chi^2_K(1-\alpha)\right\}$ is strictly less than $1-\alpha$ when $\lambda_{min} > 1$.

□

In words, this theorem quantifies the effect on coverage probability when the true co-kriging model M_A, which accounts for cross-correlations between modes, is misspecified as M_0, the independent model ignoring such cross-correlations. Note that an increase in the number of significant non-zero cross-correlations in $\mathbf{T}$ causes $\mathbf{T}^{1/2}\mathbf{D}^{-1}\mathbf{T}^{1/2}$ to deviate further from unity, which in turn may increase λ_{min}. Given enough such correlations, Theorem 19 shows that the coverage probability from the misspecified model M_0 is less than the desired $100(1-\alpha)\%$ rate. In the present case, this suggests that when there are enough significant flow couplings, the co-kriging model M_A provides more accurate UQ for the *joint* prediction of flow variables when compared to the misspecified, independent model M_0. This improvement also holds for functions of flow variables (as we demonstrate later in Section 4.4), although a formal argument is not presented here.

It is important to mention here an important trade-off for co-kriging models in general, and why the proposed model is appropriate for the application at hand in view of such a trade-off. It is known from spatial statistics literature (see, e.g., [12, 153]) that when the matrix $\mathbf{T}$ exhibits strong correlations and can be estimated well, one enjoys improved predictive performance through a co-kriging model (this is formally shown for the current model in Theorem 19). However, when such correlations are absent or cannot be estimated

well, a co-kriging model can yield poorer performance to an independent model! We claim that the former is true for the current application at hand. First, the differential equations governing the simulation procedure explicitly impose strong dependencies between flow variables, so we know *a priori* the existence of strong correlations in $\mathbf{T}$. Second, we will show later in Section 4.4.4 that the dominant correlations selected in $\mathbf{T}$ are physically interpretable in terms of fluid mechanic principles and conservation laws, which provides strong evidence for the correct estimation of $\mathbf{T}$.

One issue with fitting M_A is that there are many more parameters to estimate. Specifically, since the CPOD covariance matrix $\mathbf{T}$ is $K \times K$ dimensional, there is insufficient data for estimating all entries in $\mathbf{T}$ using the extracted coefficients from the CPOD expansion. One solution is to impose the sparsity constraint $\|\mathbf{T}^{-1}\|_1 \leq \gamma$, where $\|\mathbf{A}\|_1 = \sum_{k=1}^{K} \sum_{l=1}^{K} |A_{kl}|$ is the element-wise L_1 norm. For a small choice of γ, this forces nearly all entries in $\mathbf{T}^{-1}$ to be zero, thus permitting consistent estimation of the few significant correlations. Sparsity can also be justified from an engineering perspective, because the number of significant couplings is known to be small from flow physics. γ can also be adjusted to extract a pre-specified number of flow couplings, which is appealing from an engineering point-of-view. The justification for sparsifying $\mathbf{T}^{-1}$ instead of $\mathbf{T}$ is largely computational, because, algorithmically, the former problem can be handled much more efficiently than the latter using the graphical LASSO ([81]; see also [154]). Such efficiency is crucial here, since GP parameters need to be jointly estimated as well.

Although the proposed model is similar to the one developed in [155] for emulating qualitative factors, there are two key distinctions. First, our model allows for different process variances for each coefficient, whereas their approach restricts all coefficients to have equal variances. Second, our model incorporates sparsity on the CPOD covariance matrix, an assumption necessary from a statistical point-of-view and appealing from a physics extraction perspective. Lastly, the algorithm proposed below can estimate $\mathbf{T}$ more efficiently than the semi-definite programming approach in [155].

4.3.3 Parameter estimation

To estimate the model parameters $\boldsymbol{\mu}$, $\mathbf{T}$ and $\boldsymbol{\tau}$, maximum-likelihood estimation (MLE) is used in favor of a Bayesian implementation. The primary reason for this choice is computational efficiency: for the proposed emulator to be used as a fast investigative tool for surveying the design space, it should generate flow predictions much quicker than a direct LES simuation, which requires several days of parallelized computation.

From (4.5) and (4.6), the maximum-likelihood formulation can be written as $\operatorname{argmin}_{\boldsymbol{\mu},\mathbf{T},\boldsymbol{\tau}}$ $l_\lambda(\boldsymbol{\mu}, \mathbf{T}, \boldsymbol{\tau})$, where $l_\lambda(\boldsymbol{\mu}, \mathbf{T}, \boldsymbol{\tau})$ is the *penalized* negative log-likelihood:

$$l_\lambda(\boldsymbol{\mu}, \mathbf{T}, \boldsymbol{\tau}) = n \log\det \mathbf{T} + K \log\det \mathbf{R}_\tau + (\mathbf{B} - \mathbf{1}_n \otimes \boldsymbol{\mu})^T [\mathbf{R}_\tau^{-1} \otimes \mathbf{T}^{-1}](\mathbf{B} - \mathbf{1}_n \otimes \boldsymbol{\mu}) + \lambda \|\mathbf{T}^{-1}\|_1. \tag{4.11}$$

Note that, because the formulation is convex in $\mathbf{T}^{-1}$, the sparsity constraint $\|\mathbf{T}^{-1}\|_1 \leq \gamma$ has been incorporated into the likelihood through the penalty $\lambda\|\mathbf{T}^{-1}\|_1$ using strong duality. Similar to γ, a larger λ results in a smaller number of selected correlations, and vice versa. The tuning method for λ should depend on the desired end-goal. For example, if predictive accuracy is the primary goal, then λ should be tuned using cross-validation techniques [86]. However, if correlation extraction is desired or prior information is available on flow couplings, then λ should be set so that a fixed (preset) number of correlations is extracted. We discuss this further in Section 4.4.

Assume for now a fixed penalty $\lambda > 0$. To compute the MLEs in (4.11), we propose the following *blockwise coordinate descent* (BCD) algorithm. First, assign initial values for $\boldsymbol{\mu}$, $\mathbf{T}$ and $\boldsymbol{\tau}$. Next, iterate the following two updates until parameters converge: (a) for fixed GP parameters $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$, optimize for $\mathbf{T}$ in (4.11); and (b) for fixed covariance matrix $\mathbf{T}$, optimize for $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$ in (4.11). With the use of the graphical LASSO algorithm from [81], the first update can be computed efficiently. The second update can be computed using non-linear optimization techniques on $\boldsymbol{\tau}$ by means of a closed-form expression for $\boldsymbol{\mu}$. In our implementation, this is performed using the L-BFGS algorithm [156], which offers

Algorithm 6 BCD algorithm for maximum likelihood estimation

1: **for** each time-step $t = 1, \cdots, T$ **do parallel**
- Set initial values $\boldsymbol{\mu} \leftarrow \mathbf{0}_K$, $\mathbf{T} \leftarrow \mathbf{I}_K$ and $\boldsymbol{\tau} \leftarrow \mathbf{1}_p$, and set $\mathbf{B} \leftarrow (\boldsymbol{\beta}(\mathbf{c}_1), \cdots, \boldsymbol{\beta}(\mathbf{c}_n))^T$

2: **repeat**

3: Optimizing $\mathbf{T}$:
- Set $\mathbf{W} \leftarrow \frac{1}{n}(\mathbf{B} - \mathbf{1}_n \otimes \boldsymbol{\mu}^T)^T \mathbf{R}_\tau^{-1}(\mathbf{B} - \mathbf{1}_n \otimes \boldsymbol{\mu}^T) + \lambda \cdot \mathbf{I}_K$

4: **repeat**

5: **for** $j = 1, \cdots, K$ **do**
- Solve $\tilde{\boldsymbol{\delta}} = \mathrm{argmin}_{\boldsymbol{\delta}} \left\{ \frac{1}{2} \|\mathbf{W}_{-j,-j}^{1/2} \boldsymbol{\delta}\|_2^2 + \lambda \|\boldsymbol{\delta}\|_1 \right\}$ using LASSO
- Update $\mathbf{W}_{-j,j} \leftarrow \mathbf{W}_{-j,-j} \tilde{\boldsymbol{\delta}}$ and $\mathbf{W}_{j,-j}^T \leftarrow \mathbf{W}_{-j,-j} \tilde{\boldsymbol{\delta}}$

6: **until** $\mathbf{W}$ converges
- Update $\mathbf{T} \leftarrow \mathbf{W}^{-1}$

7: Optimizing $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$:
- Update $\boldsymbol{\tau} \leftarrow \mathrm{argmin}_\tau \, l_\lambda(\boldsymbol{\mu}_{\boldsymbol{\tau}}, \mathbf{T}, \boldsymbol{\tau})$ with L-BFGS, with $\boldsymbol{\mu}_{\boldsymbol{\tau}} = (\mathbf{1}_n^T \mathbf{R}_\tau^{-1} \mathbf{1}_n)^{-1} (\mathbf{1}_n^T \mathbf{R}_\tau^{-1} \mathbf{B})$
- Update $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}_{\boldsymbol{\tau}}$

8: **until** $\boldsymbol{\mu}$, $\mathbf{T}$ and $\boldsymbol{\tau}$ converge

9: **end parallel for**
- **return** $\boldsymbol{\mu}(t)$, $\mathbf{T}(t)$ and $\boldsymbol{\tau}(t)$

a super-linear convergence rate without the cumbersome evaluation and manipulation of the Hessian matrix [79]. The following theorem guarantees that the proposed algorithm converges to a stationary point of (4.11) (see Appendix B for proof).

Theorem 20. *The BCD scheme in Algorithm 6 converges to some solution* $(\hat{\boldsymbol{\mu}}, \hat{\mathbf{T}}, \hat{\boldsymbol{\tau}})$ *which is stationary for the penalized log-likelihood* $l_\lambda(\boldsymbol{\mu}, \mathbf{T}, \boldsymbol{\tau})$.

It is worth noting that the proposed algorithm does not provide global optimization. This is not surprising, because the log-likehood l_λ is non-convex in $\boldsymbol{\tau}$. To this end, we run multiple threads of Algorithm 6 in parallel, each with a different initial point $\boldsymbol{\tau}_0$ from a large space-filling design on $[10^{-3}, 1 - 10^{-3}]^p$, then choose the converged parameter setting which yields the largest likelihood value from (4.11). In our experience, this heuristic performs quite well in practice.

4.4 Emulation results

In this section, we present in four parts the emulation performance of the proposed model, when trained using the database of $n = 30$ flow simulations described in Section 2. First,

we briefly introduce key flow characteristics for a swirl injector, and physically interpret the flow structures extracted from CPOD. Second, we compare the numerical accuracy of our flow prediction with a validation simulation at a new injector geometry. Third, we provide a spatio-temporal quantification of uncertainty for our prediction, and discuss its physical interpretability. Lastly, we summarize the extracted flow couplings from $\mathbf{T}$, and explain why these are both intuitive and intriguing from a flow physics perspective.

4.4.1 Visualization and CPOD modes

We employ three flow snapshots of circumferential velocity (shown in Figure 4.4) to introduce key flow characteristics for a swirl injector: the fluid transition region, spreading angle, surface wave propagation and center recirculation. These characteristics will be used for assessing emulator accuracy, UQ and extracted flow physics.

- *Fluid transition region:* The fluid transition region is the region which connects compressed-liquid near the wall (colored blue in Figure 4.4) to light-gas (colored red) near the centerline at supercritical pressure [137]. This region is crucial for analyzing injector flow characteristics, as it provides the instability propagation and feedback mechanisms between the injector inlet and exit. An important emulation goal is to accurately predict both the spatial location of this region and its dynamics, because such information can be used to assess feedback behavior at new geometries.

- *Spreading angle:* The spreading angle α (along with the LOX film thickness h) is an important physical metric for measuring the performance of a swirl injector. A larger α and smaller h indicate better performance of injector atomization and breakup processes. The spreading angle can be seen in Figure 4.4 from the blue LOX flow at injector exit (see Figure 4.1 for details).

- *Surface wave propagation:* Surface waves, which transfer energy through the fluid medium, manifest themselves as wavy structures in the flowfield. These waves allow for propaga-

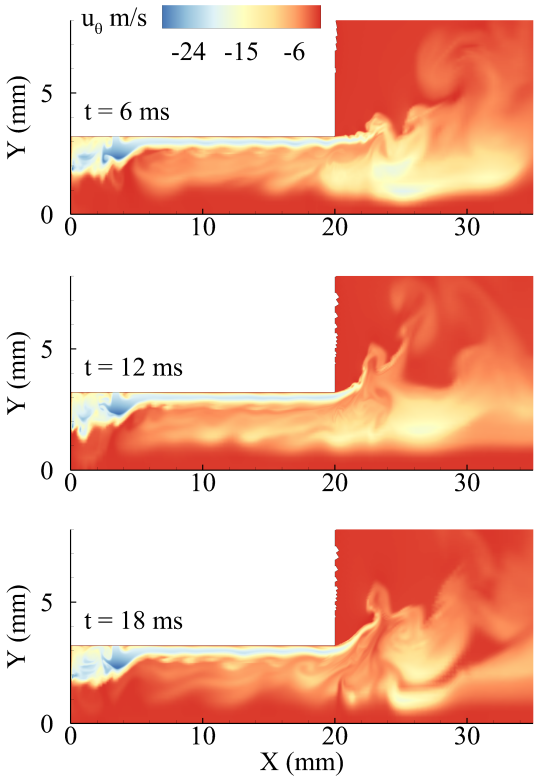

Figure 4.4: Flow snapshots of circumferential velocity at t = 6, 12 and 18 ms.

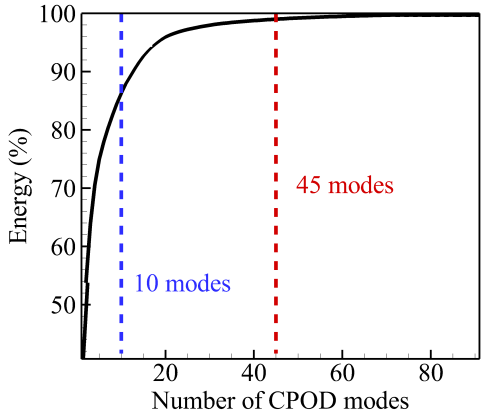

Figure 4.5: Energy distribution of CPOD modes for circumferential velocity flow.

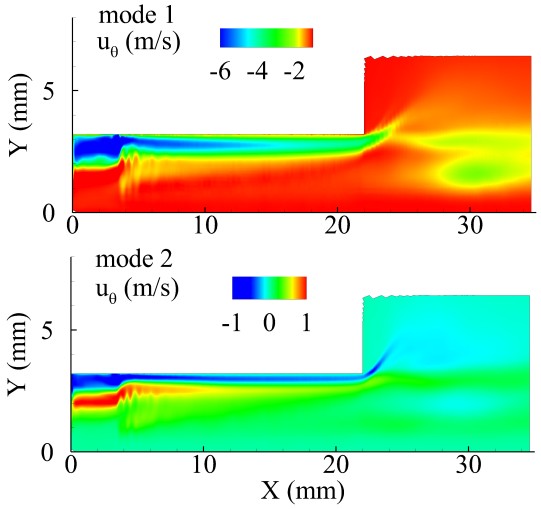

Figure 4.6: The leading two spatial CPOD modes for circumferential velocity flow.

tion of flow instabilities between upstream and downstream regions of the injector, and can be seen in the first snapshot of Figure 4.4 along the LOX film boundary.

- *Center recirculation:* Center recirculation, another key instability structure, is the circular flow of a fluid around a rotational axis (this circular region is known as the vortex core). From the third snapshot in Figure 4.4, a large vortex core (in white) can be seen at the injector exit, which is expected because of sudden expansion of the LOX stream and subsequent generation of adverse pressure gradient.

Regarding the CPOD expansion, Figure 4.5 shows the energy ratio captured using the leading M terms in (4.3) for circumferential velocity, with this ratio defined as:

$$\xi(M) = \frac{\sum_{k=1}^{M} \sum_{i=1}^{n} \int \left[\int \beta_k(t; \mathbf{c}_i) \mathscr{M}_i\{\phi_k(\mathbf{x})\} \, d\mathbf{x} \right]^2 \, dt}{\sum_{k=1}^{\infty} \sum_{i=1}^{n} \int \left[\int \beta_k(t; \mathbf{c}_i) \mathscr{M}_i\{\phi_k(\mathbf{x})\} \, d\mathbf{x} \right]^2 \, dt}.$$

Only $M = 10$ and $M = 45$ modes are needed to capture 90% and 99% of the total flow energy over *all* $n = 30$ simulation cases, respectively. Compared to a similar experiment in [136], which required around $M = 20$ modes to capture 99% flow energy for a *single* geometry, the current results are very promising, and show that the CPOD gives a reasonably compact representation. This also gives empirical evidence for the linearity assumption used for computation efficiency. Similar results also hold for other flow variables as well, and are not reported for brevity. Additionally, the empirical study in [136] showed that the POD modes capturing the top 95% energy have direct physical interpretability in terms of known flow instabilities. To account for these (and perhaps other) instability structures in the model, we set the truncation limit K_r as the smallest value of M satisfying $\xi(M) \geq 99\%$, which appears to provide a good balance between predictive accuracy and computational efficiency.

The extracted CPOD terms can also be interpreted in terms of flow physics. We illustrate this using the leading two CPOD terms for circumferential velocity, whose spatial distributions are shown in Figure 4.6. Upon an inspection of these spatial plots and their corresponding spectral frequencies, both modes can be identified as hydrodynamic instabilities in the form of longitudinal waves propagating along the LOX film boundary. Specifically, the first mode corresponds to the first harmonic mode for this wave, and the second mode represents the second harmonic and shows the existence of an antinode in wave propagation. As we show in Section 4.4, the interpretability of CPOD modes allows the proposed model to extract physically meaningful couplings for further analysis.

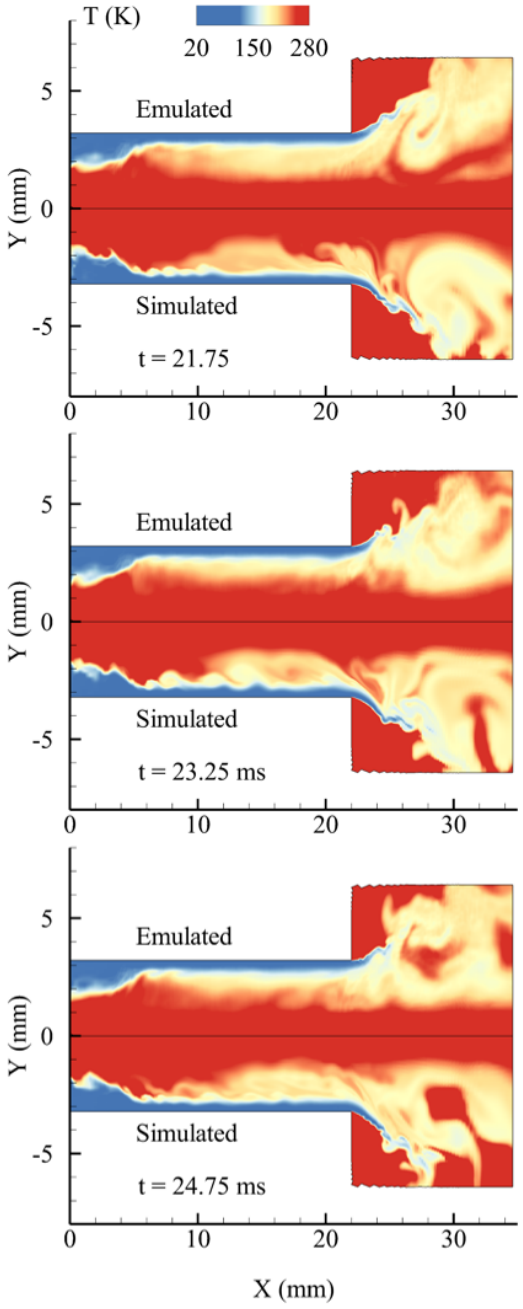

Figure 4.7: Simulated and emulated temperature flow at $t = 21.75$ ms, 23.25 ms and 24.75 ms.

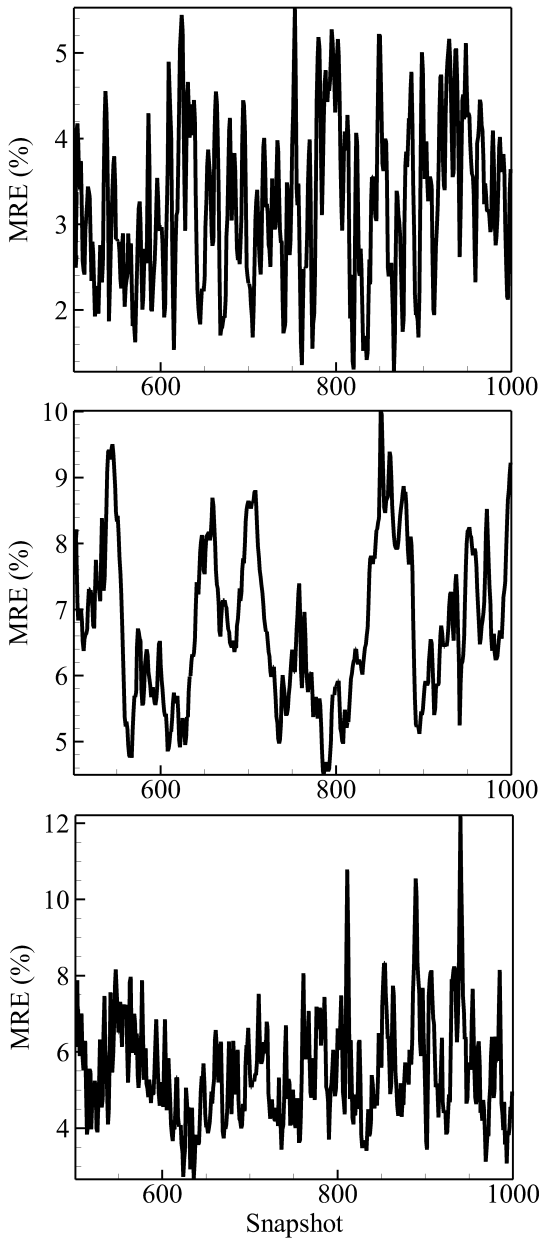

Figure 4.8: MRE at injector inlet (top), fluid transition region (middle) and injector exit (bottom).

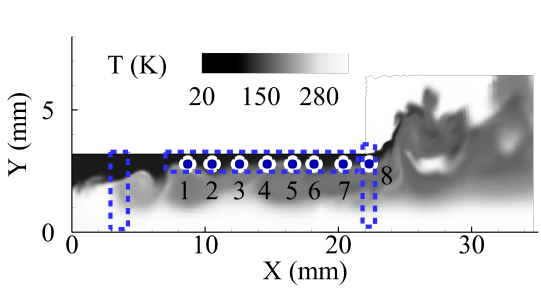

Figure 4.9: Injector subregions (dotted in blue) and probe locations (circled in white).

4.4.2 Emulation accuracy

To ensure that our emulator model provides accurate flow predictions, we perform a validation simulation at the new geometric setting: $L = 22$ mm, $R_n = 3.215$ mm, $\Delta L = 3.417$ mm, $\theta = 58.217°$ and $\delta = 0.576$ mm. This new geometry provides a 10% variation on an existing injector used in the RD-0110 liquid-fuel engine [157]. Since the goal is predictive accuracy, the sparsity penalty λ in (4.11) is tuned using 5-fold cross-validation [86]. We provide below a qualitative comparison of the predicted and simulated flows, and then discuss several metrics for quantifying emulation accuracy.

Figure 4.7 shows three snapshots of the simulated and predicted fully-developed flows for temperature, in intervals of 1.5 ms starting at 21.75 ms. From visual inspection, the predicted flow closely mimics the simulated flow on several performance metrics, including the fluid transition region, film thickness and spreading angle. The propagation of surface waves is also captured quite well within the injector, with key downstream recirculation zones correctly identified in the prediction as well. This comparison illustrates the effectiveness of the proposed emulator in capturing key flow physics, and demonstrates the importance of incorporating known flow properties of the fluid as assumptions in the statistical model.

Next, three metrics are used to quantify emulation accuracy. The first metric, which reports the mean relative error in important sub-regions of the injector, measures the *spatial* aspect of prediction accuracy. The second metric, which inspects spectral similarities

between the simulated and predicted flows, measures *temporal* accuracy. The last metric investigates how well the predicted flow captures the underlying flow physics of an injector.

For spatial accuracy, the following mean relative error (MRE) metric is used:

$$\text{MRE}(t;\mathcal{S}) = \frac{\int_{\mathcal{S}} |Y(\mathbf{x},t;\mathbf{c}_{new}) - \hat{Y}(\mathbf{x},t;\mathbf{c}_{new})|\ d\mathbf{x}}{\int_{\mathcal{S}} |Y(\mathbf{x},t;\mathbf{c}_{new})|\ d\mathbf{x}} \times 100\%,$$

where $Y(\mathbf{x},t;\mathbf{c}_{new})$ is the simulated flow at setting $\mathbf{c}_{new}$, and $\hat{Y}(\mathbf{x},t;\mathbf{c}_{new})$ is the flow predictor in (4.9) (for brevity, the superscript for flow variable r is omitted here). In words, MRE$(t;\mathcal{S})$ provides a measure of emulation accuracy within a desired sub-region $\mathcal{S}$ at time t, relative to the overall flow energy in $\mathcal{S}$. Since flow behaviors within the injector inlet, fluid transition region and injector exit (outlined in Figure 4.9) are crucial for characterizing injector instability, we investigate the MRE specifically for these three sub-regions. Figure 4.8 plots MRE$(t,\mathcal{S})$ for $t = 15 - 30$ ms, when the flow has fully developed. For all three sub-regions, the relative error is within a tolerance level of 10% for nearly all time-steps, which is very good from an engineering perspective.

To assess temporal accuracy, we conduct a power spectral density (PSD) analysis of predicted and simulated pressure flows at eight specific probes along the region of surface wave propagation (see Figure 4.9). This analysis is often performed as an empirical tool for assessing injector stability (see [136]), because surface waves allow for feedback loops between upstream and downstream oscillations [158]. Figure 4.10 shows the PSD spectra for the predicted and simulated flow at four of these probes. Visually, the spectra look very similar, both at low and high frequencies, with peaks nearly identical for the predicted and simulated flow. Such peaks are highly useful for analyzing flow physics, because they can be used to identify physical properties (e.g., hydrodynamic, acoustic, etc.) of dominant instability structures. In this sense, the proposed emulator does an excellent job in mimicking important physics of the simulated flow.

Finally, we investigate the film thickness h and spreading angle α, which are key per-

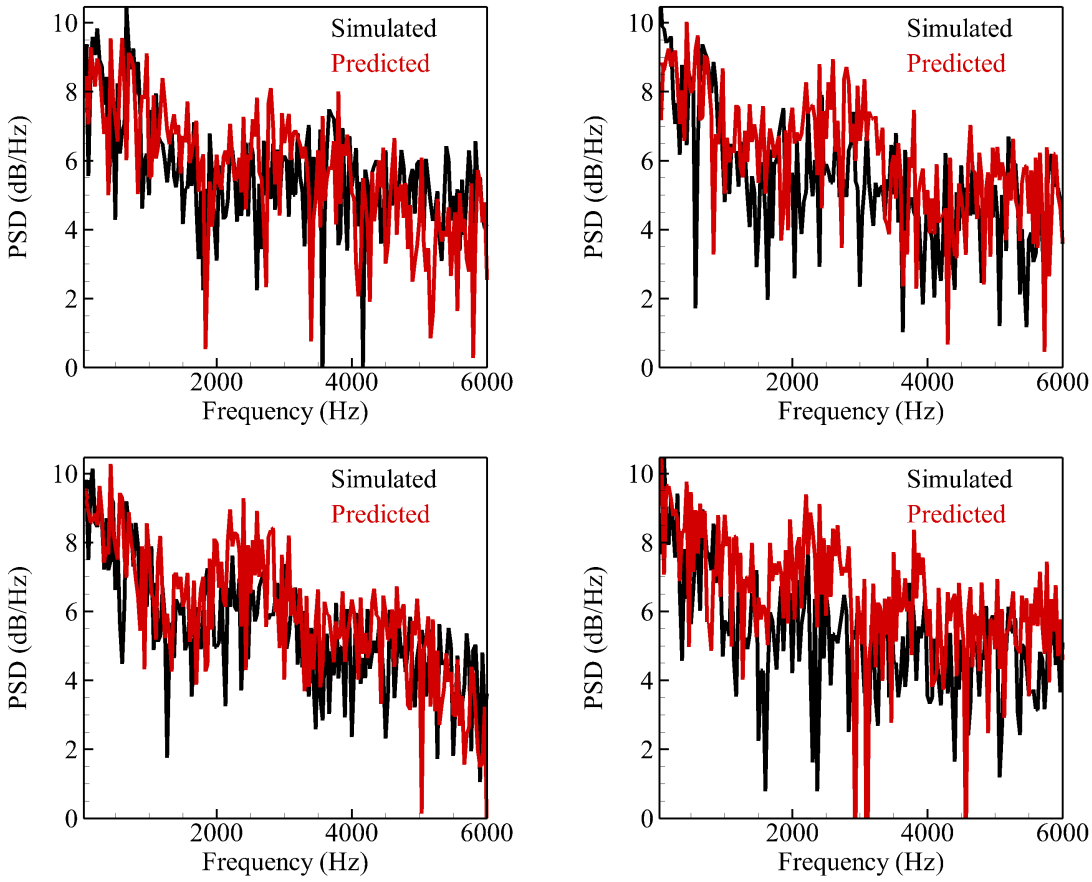

Figure 4.10: PSD spectra for pressure flow at probes 1, 3, 5 and 7 (see Figure 4.9).

formance metrics for injector performance. Since both of these metrics are computed using spatial gradients of flow variables, an accurate emulation of these measures suggests accurate flow emulation as well. For the validation setting, the simulated (predicted) flow has a film thickness of 0.47 mm (0.42 mm) and a spreading angle of 103.63° (107.36°), averaged over the fully-developed timeframe from $t = 15 - 30$ ms. This corresponds to relative errors of 10.6% and 3.60%, respectively, and is within the desired error tolerance from an engineering perspective.

4.4.3 Uncertainty quantification

For computer experiments, the quantification of predictive uncertainty can be as important as the prediction itself. To this end, we provide a spatio-temporal representation of this UQ,

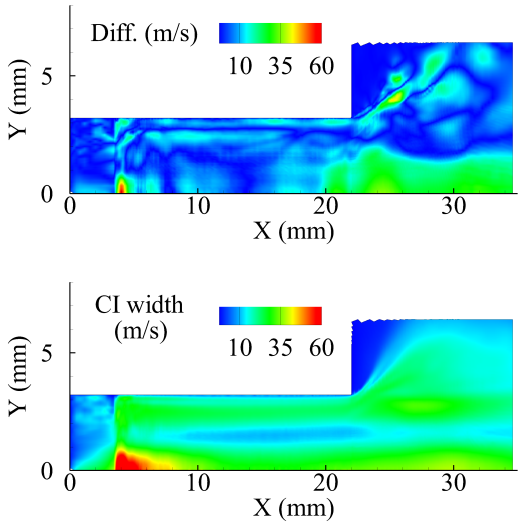

Figure 4.11: Absolute prediction error (top) and pointwise CI width (bottom) for x-velocity at $t = 15$ ms.

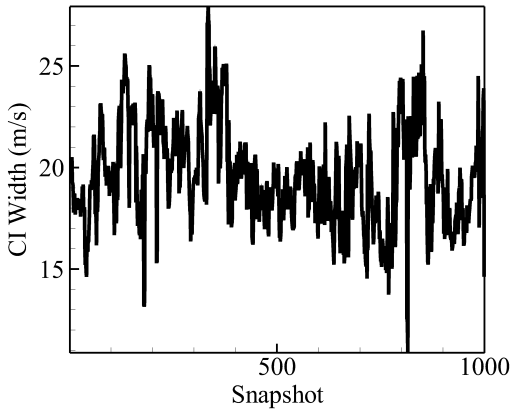

Figure 4.12: CI width of x-velocity at probe 1.

and show that it has a useful and appealing physical interpretation. For spatial UQ, the top plot of Figure 4.11 shows the one-sided width of the 99% pointwise confidence interval (CI) from (4.10) for x-velocity at $t = 15$ ms. It can be seen that the emulator is most certain in predicting near the inlet and centerline of the injector, but shows high predictive uncertainty at the three gaseous cores downstream (in green). This makes physical sense, because these cores correspond to flow recirculation vortices, and therefore exhibit highly unstable flow behavior. From the bottom plot of Figure 4.11, which shows the absolute emulation error of the same flow, the pointwise confidence band not only covers the realized prediction error, but roughly mimics its spatial distribution as well.

For temporal UQ, Figure 4.12 shows the same one-sided CI width at probe 1 (see Figure 4.9). We see that this temporal uncertainty is relatively steady over t, except for two abrupt spikes at time-steps around 300 and 800. These two spikes have an appealing physical interpretation: the first indicates a flow displayment effect of the central vortex core, whereas the second can be attributed to the boundary development of the same core. This again demonstrates the usefulness of UQ not only as a measure of predictive uncertainty, but also

as a means for extracting useful flow physics without the need for expensive simulations.

To illustrate the improved UQ of the proposed model (see Theorem 19), we use a derived quantity called turbulent kinetic energy (TKE). TKE is typically defined as:

$$\kappa(\mathbf{x},t)=\frac{1}{2}\sum_{r\in\{u,v,w\}}\left\{Y^{(r)}(\mathbf{x},t)-\bar{Y}^{(r)}(\mathbf{x})\right\}^2, \tag{4.12}$$

where $Y^{(u)}(\mathbf{x},t), Y^{(v)}(\mathbf{x},t)$ and $Y^{(w)}(\mathbf{x},t)$ are flows for x-, y- and circumferential velocities, respectively, with $\bar{Y}^{(u)}(\mathbf{x}), \bar{Y}^{(v)}(\mathbf{x})$ and $\bar{Y}^{(w)}(\mathbf{x})$ its corresponding time-averages. Such a quantity is particularly important for studying turbulent instabilities, because it measures fluid rotation energy within eddies and vortices.

For the sake of simplicity, assume that (a) the time-averages $\bar{Y}^{(u)}(\mathbf{x}), \bar{Y}^{(v)}(\mathbf{x})$ and $\bar{Y}^{(w)}(\mathbf{x})$ are fixed, and (b) the parameters $(\boldsymbol{\mu}, \mathbf{T}, \boldsymbol{\tau})$ are known. The following theorem provides the MMSE predictor and pointwise confidence interval for $\kappa(\mathbf{x},t)$ (proof in Appendix C).

Theorem 21. *For fixed* $\mathbf{x}$ *and* t*, the MMSE predictor of* $\kappa(\mathbf{x},t)$ *at a new setting* $\mathbf{c}_{new}$ *is*

$$\hat{\kappa}(\mathbf{x},t)=\frac{1}{2}\sum_{r\in\{u,v,w\}}\left\{\hat{Y}^{(r)}(\mathbf{x},t)-\bar{Y}^{(r)}(\mathbf{x})\right\}^2+tr\{\Phi(\mathbf{x},t)\}, \tag{4.13}$$

where $\hat{Y}^{(u)}(\mathbf{x},t), \hat{Y}^{(v)}(\mathbf{x},t)$ *and* $\hat{Y}^{(w)}(\mathbf{x},t)$ *are predicted flows for* x*-,* y*- and circumferential velocities from* (4.9)*, and* $\Phi(\mathbf{x},t)$ *is defined in (C.1) of Appendix C. Moreover,* $\hat{\kappa}(\mathbf{x},t)$ *is distributed as a weighted sum of non-central* χ^2 *random variables, with an explicit expression given in (C.3) of Appendix C.*

In practice, plug-in estimates are used for both time-averaged flows and model parameters.

With this in hand, we compare the prediction and UQ of TKE from the proposed model M_A and the independent model M_0 (see Theorem 19) with the simulated TKE at the validation setting. Figure 4.13 shows the predicted TKE $\hat{\kappa}(\mathbf{x},t)$ at probe 8 over the fully-developed time-frame of $t=15-30$ ms, along with the 90% lower pointwise confidence band constructed using Theorem 21. Visually, the proposed model M_A provides

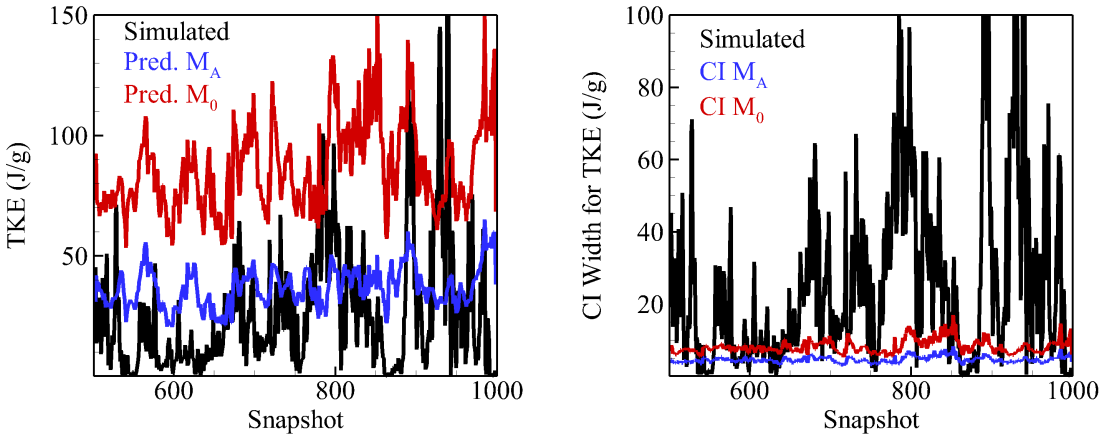

Figure 4.13: Predicted TKE and lower 90% confidence band for M_A and M_0 at probe 8.

Table 4.3: Computation time for each step of the proposed emulator, parallelized over 200 processing cores.

Step	**Comp. time (mins)**
CPOD extraction	33.91
Parameter estimation	11.31
Flow prediction	20.19
Total	65.41

an improved prediction of the simulated TKE than the independent model M_0. As for the confidence bands, the average coverage rate for M_A over the fully-developed time-frame (85.0%) is much closer to the desired nominal rate of 90% compared to that for M_0 (73.8%). The proposed model therefore provides a coverage rate closer to the desired nominal rate of 90%. The poor coverage rate for the independent model is shown in the right plot of Figure 4.13, where the simulated TKE often dips below the lower confidence band. By incorporating prior knowledge of flow couplings, the proposed model can provide improved predictive performance and uncertainty quantification.

4.4.4 Correlation extraction

Finally, we demonstrate the use of the proposed model as a tool for extracting common flow couplings on the design space. Setting the sparsity penalty λ so that only the top nine cor-

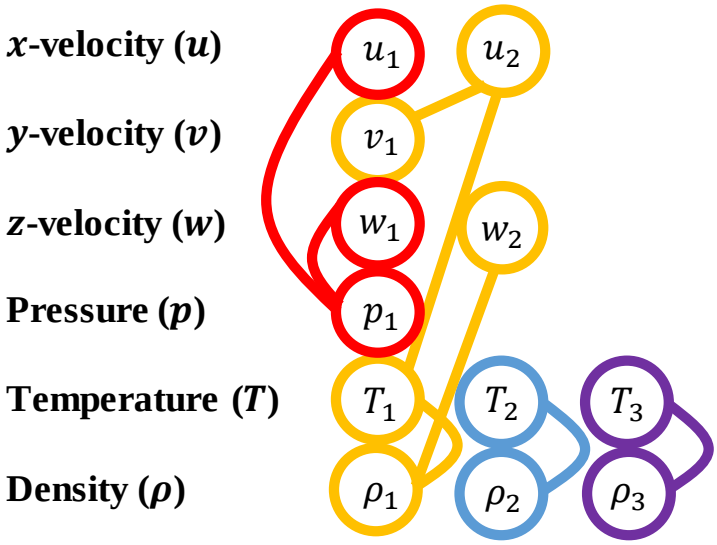

Figure 4.14: Graph of selected flow couplings from $\mathbf{T}$. Nodes represent CPOD modes, and edges represent non-zero correlations.

relations are chosen, Figure 4.14 shows the corresponding graph of the extracted couplings of CPOD modes. Nodes on this graph represent CPOD modes for each flow variable, with edges indicating the presence of a non-zero correlation between two modes. Each connected subgraph in Figure 4.14 is interpretable in terms of flow physics. For example, the subgraph connecting u_1, w_1 and P_1 (first modes for x-velocity, circumferential velocity and pressure) makes physical sense, because u_1 and w_1 are inherently coupled by Bernoulli's equation for fluid flow [159], while w_1 and P_1 are connected by the centrifugal acceleration induced by circular momentum of LOX flow. Likewise, the subgraph connecting T_1, ρ_1 and w_2 also provides physical insight: T_1 and ρ_1 are coupled by the equation of state and conservation of energy, while ρ_1 and w_2 are connected by conservation of momentum.

The interpretability of these extracted flow couplings in terms of fundamental conservation laws from fluid mechanics is not only appealing from a flow physics perspective, but also provides a reassuring check on the estimation of the co-kriging matrix $\mathbf{T}$. Recall from the discussion in Section 4.3.2 that an accurate estimate of $\mathbf{T}$ is needed for the improved predictive guarantees of Theorem 19 to hold. The consistency of the selected flow couplings (and the ranking of such couplings) with established physical principles provides

confidence that the proposed estimation algorithm indeed returns an accurate estimate of $\mathbf{T}$. These results nicely illustrate the dual purpose of the CPOD matrix $\mathbf{T}$ in our co-kriging model: not only does it allow for more accurate UQ, it also extracts interesting flow couplings which can guide further experiments.

4.4.5 Computation time

In addition to accurate flow emulation and physics extraction, the primary appeal of the proposed emulator is its efficiency. Table 4.3 summarizes the computation time required for each step of the emulation process, with timing performed on a parallelized system of 200 Intel Xeon E5-2603 1.80GHz processing cores. Despite the massive training dataset, which requires nearly 100GB of storage space, we see that the proposed model can provide accurate prediction, UQ and coupling extraction in slightly over an hour of computation time. Moreover, because both CPOD extraction and parameter estimation need to be performed only once, the surrogate model can generate flow predictions for hundreds of new settings within a day's time, thereby allowing for the exploration of the full design space in practical turn-around times. Through a careful elicitation and incorporation of flow physics into the surrogate model, we show that an efficient and accurate flow prediction is possible despite a limited number of simulation runs, with the trained model extracting valuable physical insights which can be used to guide further investigations.

4.5 Conclusions and future work

In this chapter, a new emulator model is proposed which efficiently predicts turbulent cold-flows for rocket injectors with varying geometries. An important innovation of our work lies in its *elicitation* and *incorporation* of flow properties as model assumptions. First, exploiting the deep connection between POD and turbulent flows [131], a novel CPOD decomposition is used for extracting common instabilities over the design space. Next, taking advantage of dense temporal resolutions, a time-independent emulator is proposed

that considers independent emulators at each simulation time-step. Lastly, a sparse covariance matrix $\mathbf{T}$ is employed within the emulator model to account for the few significant couplings among flow variables. Given the complexities inherent in spatio-temporal flows and the massive datasets at hand, such simplifications are paramount for accurate flow predictions in practical turn-around times. This highlights the need for careful elicitation in flow emulation, particularly for engineering applications where the time-consuming nature of simulations limits the number of available runs.

Applying the model to simulation data, the proposed emulator provides accurate flow predictions and captures several key metrics for injector performance. In addition, the proposed model offers two appealing features: (a) it provides a physically meaningful quantification of spatio-temporal uncertainty, and (b) it extracts significant couplings between flow instabilities. A key advantage of our emulator over existing flow kriging methods is that it provides accurate predictions using only a fraction of the time required by simulation. This efficiency is very appealing for engineers, because it allows them to fully explore the desired design space and make timely decisions.

Looking ahead, we are pursuing several directions for future research. First, while the CPOD expansion appears to work well for cold-flows, the justifying assumption of similar Reynolds numbers does not hold for more complicated (e.g., reacting) turbulent flows. To this end, we are working on ways to incorporate pattern recognition techniques [160] and machine learning methods [161] into the GP kriging framework to jointly (a) identify common instability structures that scale non-linearly over varying geometries, then (b) predict such structures at new geometric settings. The key hurdle is again computational efficiency, and the treed GP models in [111] or the local GP models in [162] and [163] appear to be attractive options. Some preliminary results on these extensions can be found in [164] and [165]. Next, a new design is proposed recently in [61] which combines the MaxPro methodology with minimax coverage, and it will be interesting to see whether such designs can provide improved performance. Lastly, to evaluate the stability of new injector

geometries, the UQ for the emulated flow needs to be fed forward through an acoustics solver. Since each evaluation of the solver can be time-intensive, this forward uncertainty propagation can be performed more quickly by reducing this UQ to a set of representative points, and the support points in [72] can prove to be useful for conducting such a task. The exploration of a physics-guided uncertainty quantification method is also of interest; preliminary results on this can be found in [166] and [167].

CHAPTER 5

MINIMAX AND MINIMAX PROJECTION DESIGNS USING CLUSTERING

5.1 Introduction

For a desired design space $\mathcal{X} \subseteq \mathbb{R}^p$, a *minimax distance design* (or simply *minimax design*) is the set of points which *minimizes* the *maximum* distance from any point in $\mathcal{X}$ to its nearest design point. In other words, minimax designs provide a uniform coverage of the design space $\mathcal{X}$ in worst-case scenarios, by ensuring every point in $\mathcal{X}$ is sufficiently well-covered by a design point. The emphasis on mitigating worst-case scenarios allows minimax designs to be applied in a wide range of settings. One such application is in the field of computer experiments, where the goal is to construct a computationally cheap emulator of an expensive simulator using a small number of simulation runs. By conducting these simulations at the points of a minimax design, it can be shown [168] that the resulting emulator minimizes worst-case prediction error. Minimax designs are also useful for sensor allocation. In particular, by placing sensors according to a minimax design, the minimum information sensed at any point can be maximized. This is particularly important in health and safety monitoring (see, e.g., [169]), where failure to detect faults in any part of $\mathcal{X}$ may result in catastrophic human or structural loss. Minimax designs are also useful for resource allocation problems for which an equitable distribution of limited resources is desired [170].

Despite its many uses, there has been little algorithmic developments for computing minimax designs [171]. A major reason for this is that, when $\mathcal{X}$ is a continuous space, the minimax objective (introduced later in Section 2) requires evaluating the supremum over

The paper based on this chapter will appear in *Journal of Computational and Graphical Statistics*.

an infinite set, which is costly to approximate. Some existing work include the seminal paper on minimax designs by [168] and the minimax Latin hypercube designs proposed by [172], but both papers only consider two-dimensional designs with restricted design sizes. This greatly limits the applicability of these methods in practice. There has also been some work on minimax designs when $\mathcal{X}$ is approximated by a finite set of points. For example, [173] studied these designs in the context of two-level factorial experiments, and [174] proposed a set-covering binary integer program (BIP) for computing minimax designs when points restricted to a finite candidate set of size $N < \infty$. As we show later, BIP can be very time-consuming and provides poor minimax designs for high-dimensional regions. In this chapter, we propose a hybrid clustering algorithm which can generate near-optimal minimax designs efficiently, both for large design sizes and in high-dimensions.

Although most clustering-based designs are not intended for minimax use, there are two reasons for discussing and comparing these designs in this chapter. First, an understanding of clustering-based designs allows us to better motivate the proposed minimax clustering algorithm. Second, since the proposed algorithm is similar to the popular Lloyd's algorithm [175, 8] used in k-means clustering, our simulation studies show that many clustering-based designs indeed possess good minimax properties, and it would be worthwhile to use these designs as a comparison benchmark. The use of clustering in experimental design dates back to [6] and [7], who proposed designs for optimal stratified sampling. K-means clustering using Lloyd's algorithm is also employed for generating a variety of designs, such as *principal points* [3], *minimum-MSE quantizers* [176] and *mse-rep-points* [2]. To foreshadow, we show later that minimax designs can be obtained using a modification of Lloyd's algorithm. More recent applications of clustering in design include the Fast Flexible space-Filling (FFF) designs proposed by [177], which make use of hierarchical clustering to generate space-filling designs for computer experiments. A more in-depth discussion of these designs is provided in Section 2.

The chapter is outlined as follows. To better motivate the need for minimax designs,

Section 2 begins with an overview of existing methods, then compares these methods with the proposed algorithm for a real-world example on air quality monitoring. Section 3 presents the new hybrid clustering algorithm for generating minimax designs, and provides some theoretical results on its correctedness and running time. Section 4 then outlines some numerical simulations comparing the proposed method with existing algorithms for a variety of design spaces. Section 5 introduces a new type of experimental design called *minimax projection designs*, which are obtained by performing a simple refinement step on a minimax design. Finally, Section 6 discusses some future research directions.

5.2 Background and motivation

We begin by formally defining a minimax design:

Definition 13. *[168] Let $\mathcal{X} \subseteq \mathbb{R}^p$ be a desired design space. An n-point minimax design on $\mathcal{X}$ is defined as the optimal solution of*

$$\underset{\mathcal{D}_n \in \mathbb{D}_n}{\operatorname{argmin}} \sup_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - Q(\mathbf{x}, \mathcal{D}_n)\|, \tag{5.1}$$

where $\mathbb{D}_n \equiv \{\{\mathbf{m}_i\}_{i=1}^n : \mathbf{m}_i \in \mathcal{X}\}$ is the set of all unordered n-tuples on $\mathcal{X}$, and $Q(\mathbf{x}, \mathcal{D}_n) \equiv \operatorname{argmin}_{\mathbf{z} \in \mathcal{D}_n} \|\mathbf{x} - \mathbf{z}\|$ returns the nearest design point to $\mathbf{x}$ under norm $\|\cdot\|$.

For the remainder of this chapter, $\|\cdot\|$ is taken to be the Euclidean norm $\|\cdot\|_2$, although the proposed algorithm can easily be generalized to other norms.

This section begins by detailing the existing methods for generating minimax designs mentioned in the Introduction. A real-world application on air monitoring is then presented to motivate the importance of minimax designs in practice.

5.2.1 Existing algorithms

We first introduce the BIP algorithm in [174], which generates minimax designs on the finite design space $\mathcal{X} = \{\mathbf{y}_i\}_{i=1}^N$. Let $I_1, \cdots, I_N$ be binary decision variables, with $I_j = 1$

indicating point j is included in the design and $I_j = 0$ otherwise. Also, let Ω_i denote the index set of points in $\mathcal{X}$ with (Euclidean) distance at most S. The BIP algorithm optimizes the following problem:

$$z(S) = \min_{I_1,\cdots,I_N} \sum_{j=1}^{N} I_j \quad \text{s.t.} \quad \sum_{j\in\Omega_i} I_j \geq 1, \quad i = 1,\cdots,N, \quad I_j \in \{0,1\}, \quad j = 1,\cdots,N,$$
$$d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|_2, \quad \Omega_i = \{j \ : \ d_{ij} \leq S, j = 1,\cdots,N\}. \tag{5.2}$$

In words, the optimization in (5.2) chooses the smallest number of design points from $\mathcal{X}$, denoted as $z(S)$, needed to ensure all points in $\mathcal{X}$ are at most a distance of S away from its nearest design point. The n-point minimax design can then be obtained by finding the smallest radius S for which the optimal design size $z(S)$ satisfies $z(S) = n$. When the candidate points $\{\mathbf{y}_j\}_{j=1}^N$ are, in some sense, representative of a continuous design space, the design generated by BIP can be used to approximate the minimax design in (5.1).

Unfortunately, BIP has a major caveat which greatly limits its applicability in practice: the optimization in (5.2) is computationally tractable only when the number of candidate points N is small. For example, due to memory and time constraints, N cannot exceed 1,000 for most desktop computers. In this sense, BIP is not only computationally demanding, but provides poor minimax designs when p is large, since 1,000 points are insufficient for representing a high-dimensional space. This is illustrated in the simulations in Section 4.

Next, we discuss two types of clustering-based designs: principal points [3] and FFF designs [177]. Assume the design space $\mathcal{X}$ is convex and bounded, and let $U(\mathbf{X})$ denote the uniform distribution on $\mathcal{X}$. Just as minimax designs are defined as a minimizer of the minimax objective in (5.1), the *principal points* of $U(\mathbf{X})$ are similarly defined as a minimizer of the integrated squared-error criterion:

$$\operatorname*{argmin}_{\mathcal{D}_n\in\mathbb{D}_n} \int_{\mathcal{X}} \|\mathbf{x} - Q(\mathbf{x},\mathcal{D}_n)\|_2^2 \, d\mathbf{x}, \tag{5.3}$$

where $\mathbb{D}_n$ and $Q(\mathbf{x}, \mathscr{D}_n)$ are defined as in (5.1). In words, principal points aim to provide a uniform coverage of $\mathbf{X}$ by ensuring that, for a point uniformly sampled on $\mathfrak{X}$, the expected squared-distance to its closest design point is minimized. Principal points are also known as *minimum-MSE quantizers* in signal processing literature [176], and *mse-rep-points* in quasi-Monte Carlo literature [2].

To compute principal points, [178] proposed the following two-step algorithm. First, generate a large random sample $\{\mathbf{y}_j\}_{j=1}^N \overset{i.i.d.}{\sim} U(\mathbf{X})$, along with an initial design $\{\mathbf{m}_i\}_{i=1}^n \overset{i.i.d.}{\sim} U(\mathbf{X})$. K-means clustering using Lloyd's algorithm [175, 8] is then performed with the large sample $\{\mathbf{y}_j\}_{j=1}^N$ as clustering data. In particular, Lloyd's algorithm iterates the following two updates until design points converge: (a) each sample point in $\{\mathbf{y}_j\}_{j=1}^N$ is first assigned to its closest design point; (b) each design point is then updated as the arithmetic mean of sample points assigned to it. The converged design is then taken as the principal points of $U(\mathfrak{X})$. A similar algorithm is used in the popular Linde-Buzo-Gray (LBG) algorithm [176] for generating minimum-MSE quantizers.

Justifying why such an algorithm provides locally optimal solutions of (5.3) requires two lines of reasoning. First, using the random sample $\{\mathbf{y}_j\}_{j=1}^N$, the Monte Carlo approximation of (5.3) becomes:

$$\min_{\boldsymbol{\gamma}, \mathbf{m}_1, \cdots, \mathbf{m}_n} \frac{1}{N} \sum_{i=1}^{n} \sum_{j=1}^{N} \gamma_{ij} \|\mathbf{y}_j - \mathbf{m}_i\|_2^2 \quad \text{s.t.} \quad \gamma_{ij} \in \{0,1\},\, i = 1, \cdots, n,\; j = 1, \cdots, N;$$
$$\mathbf{m}_i \in \mathbb{R}^p,\, i = 1, \cdots, n;\; \sum_{i=1}^{n} \gamma_{ij} = 1,\; j = 1, \cdots, N. \tag{5.4}$$

Here, $\boldsymbol{\gamma} = \{\gamma_{ij}\}$ is the set of binary decision variables, with $\gamma_{ij} = 1$ indicating the assignment of sample point $\mathbf{y}_j$ to design point $\mathbf{m}_i$. These binary variables serve the same role as $Q(\mathbf{x}, \mathscr{D}_n)$ in (5.3), namely, to assign each point in $\mathfrak{X}$ to its closest design point. Likewise, the decision variables $\{\mathbf{m}_i\}_{i=1}^n$ correspond to the design optimization of $\mathscr{D}_n \in \mathbb{D}_n$ in (5.3). Second, the two updates in Lloyd's algorithm iteratively optimize the assignment variables

$\{\gamma_{ij}\}$ and design points $\{\mathbf{m}_i\}$ in (5.4) respectively, while keeping other decision variables fixed. Specifically, by assigning each sample point $\mathbf{y}_j$ to its closest design point $\mathbf{m}_i$, the assignment variables $\{\gamma_{ij}\}$ in (5.4) are optimized for a fixed design $\{\mathbf{m}_i\}$. Similarly, by updating each design point $\mathbf{m}_i$ as the arithmetic mean of sample points assigned to it, the design $\{\mathbf{m}_i\}_{i=1}^n$ in (5.4) is optimized for fixed assignment variables. Iterating these updates until convergence therefore returns a locally optimal design for (5.3).

The FFF designs proposed by [177] are of a similar flavor to principal points. These designs are generated by first obtaining a large sample $\{\mathbf{y}_j\}_{j=1}^N \overset{i.i.d.}{\sim} U(\mathfrak{X})$, conducting hierarchical clustering with Ward's minimum-variance criterion [179] to form n clusters of $\{\mathbf{y}_j\}_{j=1}^N$, then using cluster centroids as design points. The computation time of FFF designs can be shown to be $O(pN^2 \log N)$ [180], which suggests that, although these designs can be generated efficiently in high-dimensions for a fixed sample size N, its computation may be prohibitive when N increases. To contrast, the proposed algorithm generates minimax designs efficiently both in high-dimensions and for large sample sizes.

In this chapter, we compare the minimax performance of BIP designs, principal points and FFF designs to the designs generated by the proposed method. To reiterate, while the latter two designs are not intended for minimax use, they are included to provide a benchmark for our algorithm, and to show that such designs indeed provide decent minimax performance.

5.2.2 Motivating example: Air quality monitoring

To motivate the use of minimax designs in real-world situations, consider the problem of air quality monitoring in the state of Georgia. With wildfire occurrences and air pollution levels on the rise in many parts of the United States [181], there is an increasing need for precise air quality monitoring, both for supporting warning systems and for guiding public health and policy decisions. To this end, many states have adopted the Ambient Monitoring Program (AMP), which requires hourly reporting of concentration levels for

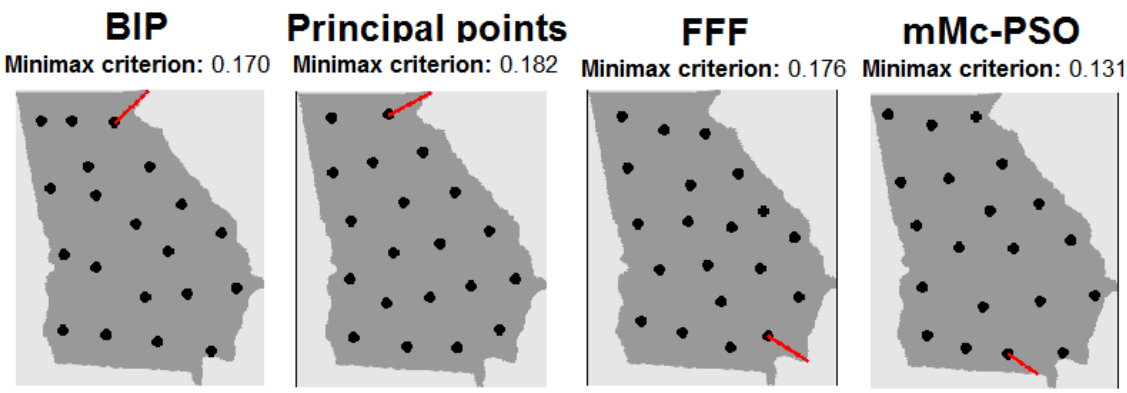

Figure 5.1: Four different 20-point designs for the state of Georgia. The red line on each plot connects the point in Georgia furthest from the design to its nearest design point, with its length equal to the minimax criterion of the design. Of these four designs, the new method MMC-PSO provides the best minimax design.

six key air pollutants. Unfortunately, only a small number of monitoring stations can be set-up for each state, since the building and maintenance of these stations can be very expensive. As a result, there are only 30 such stations situated in the state of Georgia [182]. A key problem then is to allocate these limited stations in such a way that each part of the state is covered sufficiently well by a station. The optimal allocation scheme, by definition, is that provided by a minimax design.

Figure 5.1 plots the 20-point designs generated by the three existing methods: BIP, principal points and FFF, along with the design generated by the proposed algorithm MMC-PSO. The red line on each plot connects the point in Georgia furthest from the design to its nearest design point. Note that the minimax criterion in (5.1) (reported at the top of each plot) corresponds to the length of this line. Two key observations can be made here. First, principal points and MMC-PSO appear to provide the best visual uniformity of the four methods, whereas the design generated by BIP appears to be visually non-uniform. Second, MMC-PSO provides the lowest minimax distance of the four methods, which illustrates the improvement that the proposed method offers over existing methods. We show that this improvement holds for a wide range of design regions in Section 4.

5.3 Methodology

In this section, we first present the *minimax clustering* algorithm as a generalization of Lloyd's algorithm, then establish theoretical results for the correctedness and running time for the proposed method. Finally, we introduce a global optimization modification for minimax clustering, which allows near-optimal minimax designs to be generated.

5.3.1 Minimax clustering

To begin, we introduce a new type of center for a finite set of points:

Definition 14. *For a finite set of m points $\mathfrak{X} = \{\mathbf{z}_i\}_{i=1}^m \subseteq \mathbb{R}^p$, its C_q-center is defined as:*

$$\operatorname*{argmin}_{\mathbf{z}\in\mathbb{R}^p} D_q(\mathbf{z};\mathfrak{X}) \equiv \operatorname*{argmin}_{\mathbf{z}\in\mathbb{R}^p} \frac{1}{mq}\sum_{i=1}^m \|\mathbf{z}-\mathbf{z}_i\|_2^q. \tag{5.5}$$

C_q-centers can be seen as Fréchet means [183], which are of the form:

$$\operatorname*{argmin}_{\mathbf{z}\in\mathbb{R}^p} \sum_{i=1}^m w_i d(\mathbf{z},\mathbf{z}_i),$$

with weights $w_i = 1/(mq)$ and distance function $d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_2^q$. With $q = 2$, the C_q-center becomes the arithmetic mean, used for updating cluster centers in Lloyd's algorithm. More importantly, as $q \to \infty$, the C_q-center returns the point which *minimizes* the *maximum* distance between it and a point in $\mathfrak{X}$. To foreshadow, C_∞-centers will be used in place of arithmetic means in the proposed clustering scheme.

The intuition for minimax clustering can then be presented by direct analogy to principal points. Consider the minimax objective in (5.1), and note that for sufficiently large choices of $q > 0$, this objective can be approximated as:

$$\operatorname*{argmin}_{\mathfrak{D}_n\in\mathbb{D}_n} \left(\int_{\mathfrak{X}} \|\mathbf{x}-Q(\mathbf{x},\mathfrak{D}_n)\|_2^q \, d\mathbf{x}\right)^{1/q}, \tag{5.6}$$

Algorithm 7 Minimax clustering

1: **function** MMC($\{\mathbf{m}_i\}_{i=1}^n, N, q, t_{mMc}, \epsilon_{in}$) ▷ $\{\mathbf{m}_i\}_{i=1}^n$ - initial design, t_{mMc} - max. iterations
- Initialize $\{\mathbf{y}_j\}_{j=1}^N$ using a Sobol' sequence

2: **repeat**
- For $j = 1, \cdots, N$, assign $\mathbf{y}_j$ to its closest design point in Euclidean norm.
- For $i = 1, \cdots, n$, update $\mathbf{m}_i \leftarrow C_q\text{-AGD}(\mathfrak{X}_i, q, \epsilon_{in})$, where $\mathfrak{X}_i$ is the set of points assigned to $\mathbf{m}_i$
- $t \leftarrow t + 1$.

3: **until** design points converge **OR** $t \geq t_{mMc}$.
- **return** converged design $\{\mathbf{m}_i\}_{i=1}^n$.

In practice, q should be large enough to provide a good approximation of (5.1), yet small enough to avoid numerical instability. The choice of q is discussed further in Section 3.2.1.

The similarities between the approximation (5.6) and the integrated squared-error (5.3) allows for a modification of Lloyd's algorithm to generate minimax designs. First, generate a large sample $\{\mathbf{y}_j\}_{j=1}^N \overset{i.i.d.}{\sim} U(\mathfrak{X})$, along with initial cluster centers $\{\mathbf{m}_i\}_{i=1}^n \overset{i.i.d.}{\sim} U(\mathfrak{X})$. The Monte Carlo approximation of (5.6) becomes:

$$\min_{\boldsymbol{\gamma}, \mathbf{m}_1, \cdots, \mathbf{m}_n} \frac{1}{N} \sum_{i=1}^{n} \sum_{j=1}^{N} \gamma_{ij} \|\mathbf{y}_j - \mathbf{m}_i\|_q^2 \quad \text{s.t.} \quad \gamma_{ij} \in \{0, 1\},\ i = 1, \cdots, n,\ j = 1, \cdots, N;$$
$$\mathbf{m}_i \in \mathbb{R}^p,\ i = 1, \cdots, n;\ \sum_{i=1}^{n} \gamma_{ij} = 1,\ j = 1, \cdots, N. \tag{5.7}$$

where $\boldsymbol{\gamma} = \{\gamma_{ij}\}$ is again the set of binary assignment variables, and $\{\mathbf{m}_i\}_{i=1}^n$ the set of design points. Minimax clustering then iteratively applies the following two updates until design points converge: (a) each sample point in $\{\mathbf{y}_j\}_{j=1}^N$ is first assigned to its closest design point, which optimizes the assignment variables $\{\gamma_{ij}\}$ in (5.7) for a fixed design $\{\mathbf{m}_i\}$; (b) each design point is then updated as the $C^{(q)}$-center of points assigned to it, which optimizes the design $\{\mathbf{m}_i\}_{i=1}^n$ in (5.7) for fixed assignments. By iterating these two updates until convergence, one should obtain a locally-optimal minimax design. The above procedure, which we call *minimax clustering* (or MMC for short), is summarized in Algorithm 7.

In our implementation, deterministic low-discrepancy sequences [17] are used in place

of random samples for $\{\mathbf{y}_j\}_{j=1}^N$, since such sequences provide a better approximation of integrals compared to Monte Carlo methods. Assume for now that the design space $\mathfrak{X}$ is $[0,1]^p$, the unit hypercube in $\mathbb{R}^p$. We employ a specific type of low-discrepancy sequence in Algorithm 7 called a *Sobol' sequence* [24], which can be generated efficiently using the function `sobol` in the R package `randtoolbox` [52]. Section 4.2 provides a brief discussion on low-discrepancy sequences for general design spaces.

5.3.2 Convergence results

The above discussion still leaves two questions unanswered. First, how can C_q-centers computed efficiently? Second, does minimax clustering indeed converge in finite iterations to a local optimum, and if so, at what rate? These concerns are addressed in this subsection.

Since the discussion below is quite technical, readers interested in the hybridization of MMC with particle swarm should skip to Section 5.3.3. Some background readings on convex programming (e.g., [184] and [185]) may also be useful for understanding the developments in this subsection. For brevity, proofs are deferred to the Appendix.

Computing C_q-centers

We first present an algorithm for computing C_q-centers, and prove that this algorithm converges quickly even when the number of points m or dimension p become large. The following theorem shows that the objective $D_q(\mathbf{z}; \mathfrak{X})$ in (5.5) is strictly convex, and that the C_q-center of $\mathfrak{X}$ is unique and contained in the convex hull of $\mathfrak{X}$, defined as $\text{conv}(\mathfrak{X}) = \{\mathbf{z} = \sum_{i=1}^m \alpha_i \mathbf{z}_i : \alpha_i \geq 0, \sum_{i=1}^m \alpha_i = 1\}$.

Theorem 22. *Let $\mathfrak{X} = \{\mathbf{z}_i\}_{i=1}^m$ and let $q \geq 2$. Then $D_q(\mathbf{z}; \mathfrak{X})$ is strictly convex in* $\mathbf{z}$*. Moreover, the C_q-center $C_q(\mathfrak{X})$ in* (5.5) *is unique, and contained in* $\text{conv}(\mathfrak{X})$.

Next, recall that a function $h : \mathbb{R}^p \to \mathbb{R}$ is *β-Lipschitz smooth* (or simply *β-smooth*) if:

$$\|\nabla h(\mathbf{z}) - \nabla h(\mathbf{z}')\|_2 \leq \beta \|\mathbf{z} - \mathbf{z}'\|_2,$$

Algorithm 8 Computing C_q-centers

1: **function** C_q-AGD($\{\mathbf{z}_i\}_{i=1}^m$, q, ϵ_{in}) ▷ ϵ_{in} - desired tolerance
- Set $t = 1$ and initialize starting points $\mathbf{z}^{[1]} \leftarrow \frac{1}{m}\sum_{i=1}^m \mathbf{z}_i$, $\mathbf{u}^{[1]} \leftarrow \frac{1}{m}\sum_{i=1}^m \mathbf{z}_i$.
- Initialize the sequences $\{\lambda_t\}_{t=0}^\infty$ and $\{\gamma_t\}_{t=1}^\infty$ from (5.9).
- Compute the Lipschitz constant $\bar{\beta}$ in (5.8).

2: **while** $\|\mathbf{z}^{[t]} - \mathbf{z}^{[t-1]}\|_2 < \epsilon_{in}$ **do**
- Update $\mathbf{u}^{[t+1]} \leftarrow \mathbf{z}^{[t]} - \frac{1}{\bar{\beta}}\left(\frac{1}{m}\sum_{i=1}^m \|\mathbf{z}^{[t]} - \mathbf{z}_i\|_2^{q-2}(\mathbf{z}^{[t]} - \mathbf{z}_i)\right)$.
- Update $\mathbf{z}^{[t+1]} \leftarrow (1-\gamma_t)\mathbf{u}^{[t+1]} + \gamma_t \mathbf{u}^{[t]}$.
- $t \leftarrow t + 1$.
- **return** $\mathbf{z}^{[t]}$.

where ∇h is the gradient of h. Likewise, h is μ-*strongly convex* if:

$$(\nabla h(\mathbf{z}) - \nabla h(\mathbf{z}'))^T(\mathbf{z} - \mathbf{z}') \geq \mu\|\mathbf{z} - \mathbf{z}'\|_2.$$

We show next that, for some specified $\bar{\beta} > 0$ and $\bar{\mu} > 0$, the objective function $D_q(\mathbf{z}; \mathfrak{Z})$ is $\bar{\beta}$-smooth and $\bar{\mu}$-strongly convex.

Theorem 23. *For $q \geq 4$, $D_q(\mathbf{z}; \mathfrak{Z})$ is $\bar{\beta}$-smooth and $\bar{\mu}$-strongly convex for $\mathbf{z} \in$ conv$(\mathfrak{Z})$, where:*

$$\bar{\beta} = (q-1)(q-2)\max_{j=1,\cdots,m} D_{q-2}(\mathbf{z}_j; \mathfrak{Z}) > 0 \quad \textit{and} \quad \bar{\mu} = (q-2)D_{q-2}(C_{q-2}(\mathfrak{Z}); \mathfrak{Z}) > 0. \tag{5.8}$$

The $\bar{\beta}$-smoothness and $\bar{\mu}$-strong convexity in Theorem 23 allow us to employ a quick convex optimization technique called *accelerated gradient descent* [186], or AGD, to compute C_q-centers. The implementation of AGD is straightforward. Suppose $h : \mathbb{R}^p \to \mathbb{R}$, the desired objective to minimize, is twice-differentiable, convex and β-smooth. Let $\mathbf{u}^{[t]} \in \mathbb{R}^p$ be the t-th solution iterate, and let $\mathbf{z}^{[t]} \in \mathbb{R}^p$ be an intermediate vector. Also, define the sequences $\{\lambda_t\}_{t=0}^\infty$ and $\{\gamma_t\}_{t=1}^\infty$ by the recursion equations:

$$\lambda_0 = 0, \quad \lambda_t = \frac{1 + \sqrt{1 + 4\lambda_{t-1}^2}}{2}, \quad \gamma_t = \frac{1 - \lambda_t}{\lambda_{t+1}} \quad \text{for } t = 1, 2, 3, \cdots. \tag{5.9}$$

AGD then iterates the following two updates until the solution sequence $\{\mathbf{u}^{[t]}\}_{t=1}^{\infty}$ converges:

$$\mathbf{u}^{[t+1]} \leftarrow \mathbf{z}^{[t]} - \frac{1}{\beta}\nabla h(\mathbf{z}^{[t]}), \quad \mathbf{z}^{[t+1]} \leftarrow (1-\gamma_t)\mathbf{u}^{[t+1]} + \gamma_t \mathbf{u}^{[t]}. \tag{5.10}$$

A direct application of AGD for the optimization in (5.5) is provided in Algorithm 8.

One may perhaps ask why this accelerated scheme is preferred over traditional line-search methods (see, e.g., [79]), in which the solution sequence $\{\mathbf{u}^{[t]}\}_{t=1}^{\infty}$ is updated by the line-search optimization:

$$\mathbf{u}^{[t+1]} = \mathbf{u}^{[t]} - \eta_t \nabla h(\mathbf{u}^{[t]}), \quad \eta_t = \operatorname*{argmin}_{\eta>0} h(\mathbf{u}^{[t]} - \eta \nabla h(\mathbf{u}^{[t]})). \tag{5.11}$$

In other words, for a given iterate $\mathbf{u}^{[t]}$, the next iterate $\mathbf{u}^{[t+1]}$ in line-search methods is obtained by searching for the optimal step-size η_t to move along the direction of its negative gradient $-\nabla h(\mathbf{u}^{[t]})$. The advantages of AGD are two-fold. First, AGD exploits the β-smoothness and μ-convexity of (5.5) to achieve an optimal rate of convergence among gradient-based optimization methods [187]. Second, the step-size optimization in (5.11) requires multiple evaluations of the objective h and its gradient ∇h. Since the evaluation of both $D_q(\mathbf{z}; \mathfrak{X})$ and $\nabla D_q(\mathbf{z}; \mathfrak{X})$ require $O(mp)$ work, such evaluations become prohibitively expensive to compute when either the number of points m or dimension p are large. AGD avoids this problem by replacing the optimized step-size η_t with a fixed stepsize $1/\bar{\beta}$.

Using Theorem 23, the correctedness and running time of Algorithm 8 can be established.

Corollary 3. *For $\mathfrak{X} = \{\mathbf{z}_i\}_{i=1}^m$ and $q \geq 4$, consider the sequence of solutions $\{\mathbf{z}^{[t]}\}_{t=1}^{\infty}$ from Algorithm 8. To guarantee an ϵ_{in}-accuracy for the objective in* (5.5)*, i.e., $|D_q(\mathbf{z}^{[t]}; \mathfrak{X}) - D_q(C_q(\mathfrak{X}); \mathfrak{X})| < \epsilon_{in}$, the computation work required is:*

$$O\left(mp\sqrt{(q-1)\kappa_{q-2}(\mathfrak{X})}\log\frac{1}{\epsilon_{in}}\right), \text{ where } \kappa_q(\mathfrak{X}) = \frac{\max_{j=1,\cdots,m} D_q(\mathbf{z}_j; \mathfrak{X})}{D_q(C_q(\mathfrak{X}); \mathfrak{X})} \tag{5.12}$$

is the ratio of maximum and minimum values of $D_q(\mathbf{z}; \mathfrak{X})$ *for* $\mathbf{z} \in$ *conv*$(\mathfrak{X})$.

Several illuminating observations can be made from this corollary. First, considering only the error tolerance ϵ_{in}, the computational work required for AGD to achieve ϵ_{in}-accuracy is $O(\log(1/\epsilon_{in}))$, which is sizably smaller than the $O(1/\epsilon_{in})$ work needed for standard line-search methods [79]. Hence, Algorithm 8 not only avoids multiple evaluations of the objective and gradient, but also converges with fewer iterations compared to line-search methods. Second, the bound in (5.12) grows on the order of $\sqrt{q}$, meaning Algorithm 8 takes longer to terminate as q grows larger. This illustrates the trade-off between performance and accuracy: a larger value of q ensures a better approximation of the minimax criterion (5.6), but requires longer time to compute. In our simulations, $q = 10$ appears to provide a good compromise in this trade-off. Lastly, the bound in (5.12) grows as $\kappa_{q-2}(\mathfrak{X})$ increases, meaning C_q-centers may take longer to compute when points in $\mathfrak{X}$ are more scattered.

Correctedness and running time of minimax clustering

The correctedness and running time of minimax clustering can then be established by direct analogy to that for Lloyd's algorithm. This is formally demonstrated below.

Theorem 24. *Algorithm 7 terminates after at most* N^n *iterations. Moreover, assuming* $n \leq N^{1/2}$*, each iteration of the loop in Algorithm 7 requires* $O\left(N^{3/2} p \sqrt{q-1} \log \frac{1}{\epsilon_{in}}\right)$ *work, where* ϵ_{in} *is the inner tolerance in Corollary 3. Lastly, when* C_q*-center updates in* (5.5) *are exact, Algorithm 7 also returns a locally optimal design for* (5.7)*.*

Unfortunately, it is difficult to establish a bound on the number of iterations required for termination of Algorithm 7, since there is still a gap between theory and practice for the same problem in Lloyd's algorithm. Theoretical work ([188, 189]) suggests that in the worst-case, the number of iterations can grow rapidly in the number of clustering points N. However, in practice, Lloyd's algorithm nearly always terminates after several iterations,

leading many practitioners (see, e.g., [190]) to evaluate total running time by the running time of one iteration. From our simulations, Algorithm 7 also converges after a small number of iterations, so we similarly use the single-iteration time in Theorem 24 to measure for total running time of minimax clustering.

In this light, the running time of Theorem 24 illustrates two computational advantages of minimax clustering. First, since this time is linear in p, minimax clustering can be performed efficiently in high-dimensions, which is similar to what is observed for FFF designs in Section 2.1. Furthermore, the running time of minimax clustering grows at a rate of $N^{3/2}$, which is much faster than the $O(N^2 \log N)$ work for FFF designs. Hence, a larger number of approximating points N can be used in minimax clustering, suggesting that the proposed method provides higher quality minimax designs when $\mathcal{X}$ is high-dimensional. As we see later in Section 4, this is indeed the case.

5.3.3 Minimax clustering with particle swarm optimization

Due to its greedy nature, Lloyd's algorithm has two drawbacks: it is sensitive to choices of initial cluster centers, and may return a locally optimal design which is far from the global design [191]. Since minimax clustering employs the same greedy steps, it suffers from the same downfalls. A simple but computationally expensive remedy is to perform Lloyd's algorithm multiple times with different initial centers, then pick the solution with the smallest criterion in (5.4). More elaborate methods requiring less computation include kernel k-means [192], sequential k-means [193], and combining k-means with particle swarm optimization [194]. To retain the iterative nature of Algorithm 7, we adopt the latter hybrid approach for global optimization of minimax clustering.

Particle swarm optimization [195], or PSO for short, is a stochastic, derivative-free algorithm for global minimization of a general function h. This algorithm can be described as follows. First, a representative set of s feasible solutions, or a *swarm* of *particles*, is chosen. Each particle is then guided towards the solution with lowest objective encoun-

Algorithm 9 Minimax clustering with PSO

function MMC-PSO($n, N, q, s, t_{mMc}, t_{pp}, \epsilon_{in}$)

- Generate $\{\mathbf{y}_j\}_{j=1}^N$ using a Sobol' sequence and initial design particles $\mathscr{D}_k = \{\mathbf{m}_i^k\}_{i=1}^n, k = 1, \cdots, s$ using scrambled Sobol' sequences.
- Define h_q as the objective in (5.7), and h as the minimax criterion in (5.1) with $\mathfrak{X} = \{\mathbf{y}_j\}_{j=1}^N$.
- *Minimax clustering PSO:* Initialize local-best designs $\mathscr{L}_k \leftarrow \mathscr{D}_k, k = 1, \cdots, s$, and global-best design $\mathscr{G} \leftarrow \operatorname{argmin}_{\mathscr{D}_k} h_q(\mathscr{D}_k)$. Set initial velocities $\mathbf{v}_k \leftarrow \mathbf{0}, k = 1, \cdots, s$.

for $t = 1, \cdots, t_{mMc}$ **do** ▷ t_{mMc} - max. PSO iterations

  for $k = 1, \cdots, s$ **do** ▷ For each design particle...

- $\mathscr{D}_k \leftarrow$ MMC($\mathscr{D}_k, N, q, 1, \epsilon_{in}$) ▷ One step of minimax clustering
- $\mathbf{v}_k \leftarrow w\mathbf{v}_k + c_1\mathbf{r}_1(\mathscr{L}_k - \mathscr{D}_k) + c_2\mathbf{r}_2(\mathscr{G} - \mathscr{D}_k),\ \mathbf{r}_1, \mathbf{r}_2 \overset{i.i.d.}{\sim} U[0,1]^{np}$ ▷ Update vel.
- $\mathscr{D}_k \leftarrow \mathscr{D}_k + \mathbf{v}_k$ ▷ Move particle towards best positions

    if $h_q(\mathscr{D}_k) < h_q(\mathscr{L}_k)$ **then** $\mathscr{L}_k \leftarrow \mathscr{D}_k$ ▷ Update local-best designs

    if $h_q(\mathscr{D}_k) < h_q(\mathscr{G})$ **then** $\mathscr{G} \leftarrow \mathscr{D}_k$ ▷ Update global-best design

- *Post-processing:* Reset global-best design $\mathscr{G} \leftarrow \operatorname{argmin}_{\mathscr{D}_k} h(\mathscr{D}_k)$ and velocities $\mathbf{v}_k \leftarrow \mathbf{0}$.

for $t = 1, \cdots, t_{pp}$ **do** ▷ t_{pp} - max. post-proc. iterations

  for $k = 1, \cdots, s$ **do** ▷ For each design particle...

- $\mathbf{v}_k \leftarrow w\mathbf{v}_k + c_1\mathbf{r}_1(\mathscr{L}_k - \mathscr{D}_k) + c_2\mathbf{r}_2(\mathscr{G} - \mathscr{D}_k),\ \mathbf{r}_1, \mathbf{r}_2 \overset{i.i.d.}{\sim} U[0,1]^{np}$ ▷ Update vel.
- $\mathscr{D}_k \leftarrow \mathscr{D}_k + \mathbf{v}_k$ ▷ Move particle towards best positions

    if $h(\mathscr{D}_k) < h(\mathscr{L}_k)$ **then** $\mathscr{L}_k \leftarrow \mathscr{D}_k$ ▷ Update local-best designs

    if $h(\mathscr{D}_k) < h(\mathscr{G})$ **then** $\mathscr{G} \leftarrow \mathscr{D}_k$ ▷ Update global-best design

- **return** global-best design $\mathscr{G}$.

tered along its own path (called the *local-best solution*), as well as the solution with lowest objective over the entire swarm (called the *global-best solution*). In this sense, PSO mimics the behavior of a bird flock searching for food: each bird naturally flies towards the closest position to a food source explored by the flock, but is also guided by the closest position explored along its own flight. When the optimization problem at hand has some desirable structure, PSO can be combined (or *hybridized*) with other algorithms to provide quicker convergence. We therefore propose a hybridization scheme below which combines PSO with the minimax clustering algorithm MMC.

The details are as follows. First, generate the set of approximating points $\{\mathbf{y}_j\}_{j=1}^N$ using a Sobol' sequence, and generate the s initial designs (forming the *particle swarm*) using scrambled Sobol' sequences [196]. In non-technical terms, these scrambled sequences provide different initial designs in the swarm, with each retaining its low-discrepancy property. Next, repeat the following steps:

- For each design particle, do one iteration of minimax clustering.
- Move each design particle towards to its local-best and global-best designs.
- Update the local-best and global-best designs for the desired objective in (5.7).

Finally, as a post-processing step, the general version of PSO described previously is applied to the minimax objective (5.1), with $\mathfrak{X}$ approximated by $\{\mathbf{y}_j\}_{j=1}^N$. The above procedure, which we call MMC-PSO, is detailed in Algorithm 9. MMC-PSO will be used to generate the minimax designs in our simulations later.

Three parameters are used to control the PSO behavior of MMC-PSO: c_1 and c_2, which account for the velocities at which each particle drifts towards its local-best and global-best solutions respectively, and w, which controls each particle's momentum from one iteration to the next. For the PSO of Lloyd's algorithm proposed by [194], the authors recommend the setting of $w = 0.72$ and $c_1 = c_2 = 1.49$, which can be shown to provide quick empirical convergence. Since this variant is similar to MMC-PSO, we adopt the same choices here. Other settings have also been tested, but we found this setting to provide the best minimax performance.

To illustrate the ability for MMC-PSO to generate near-global minimax designs, we compare the 7-point design for $p = 2$ from MMC-PSO with the global minimax design in [168]. Here, $N = 10^5$ approximating points are used, along with $s = 10$ PSO particles. The maximum iteration counts are set at $t_{mMc} = 300$ and $t_{pp} = 300$. The left plot in Figure 5.2 compares the design generated by MMC-PSO with the global minimax design. Visually, these two designs are nearly identical. Objective-wise, the minimax distance (5.1) for MMC-PSO is within 0.001 of the global minimum, suggesting that the proposed algorithm indeed provides near-global optimization of (5.1). Similar results also hold for the remaining designs in [168], but these are not reported for brevity.

The right plot in Figure 5.2, which outlines the 7-point design from Algorithm 7 (minimax clustering *without* PSO) and the global-best design $\mathcal{G}$ in MMC-PSO *before* post-processing, highlights the effectiveness of both PSO and post-processing. From this figure,

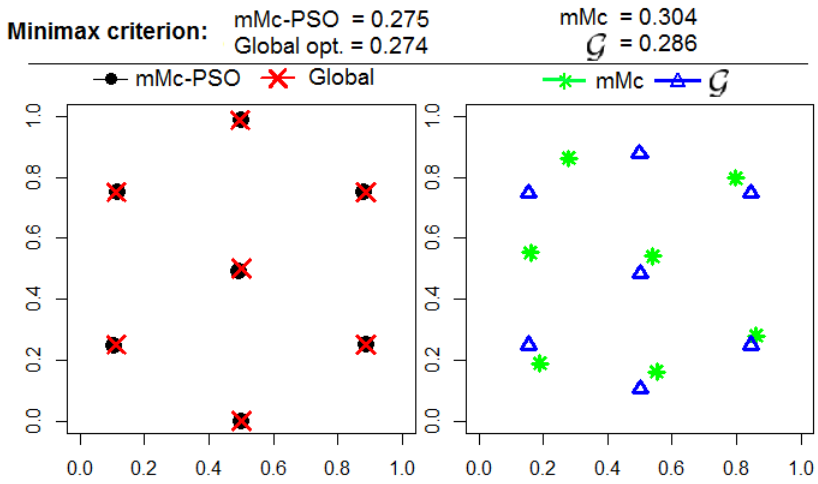

Figure 5.2: (Left) The 7-point design using MMC-PSO and the global minimax design in [168]. Since these designs are nearly identical, this demonstrates the near-global minimax performance of MMC-PSO. (Right) The 7-point design using MMC and the global-best design $\mathcal{G}$ in MMC-PSO before post-processing. The reduction in minimax distance for the latter design highlights the need for PSO.

$\mathcal{G}$ clearly gives a better approximation of the global design than mMc, both visually and criterion-wise, which suggests that the proposed PSO for minimax clustering is indeed effective. However, there is one glaring problem with $\mathcal{G}$: design points are pushed away from the boundaries of $[0,1]^2$, whereas two design points can be found on the top and bottom boundaries for the global minimax design. The post-processing step on $\mathcal{G}$, which performs PSO directly on the minimax criterion (5.1), allows design points to move towards their globally optimal positions on design boundaries.

5.4 Numerical simulations

In this section, we compare the minimax performance of designs using MMC-PSO with the existing methods in Section 2.1. The comparison is first made on the unit hypercube $[0,1]^p$, then on the unit simplex and ball. This section concludes by returning to the original motivating example on air quality monitoring.

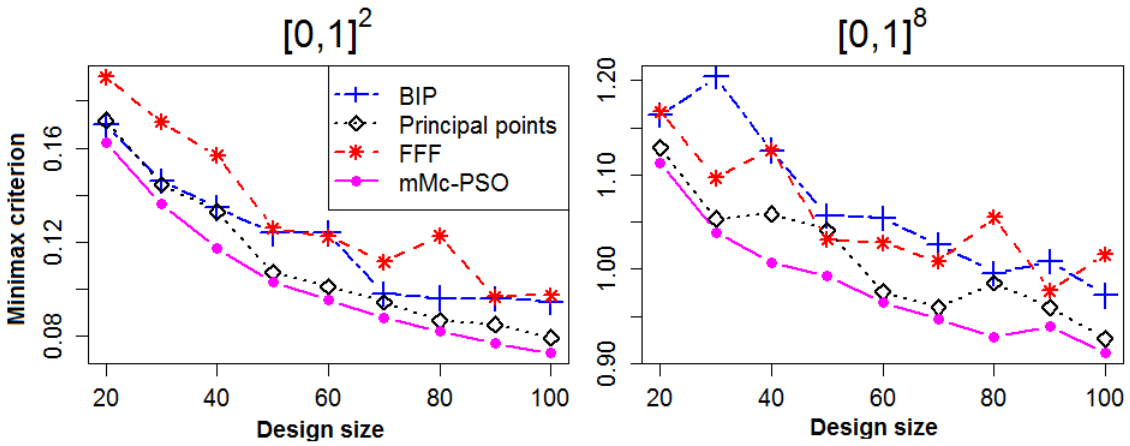

Figure 5.3: Minimax criterion for various design sizes on $[0,1]^2$ and $[0,1]^8$. Designs generated by MMC-PSO consistently give the lowest minimax distance for all design sizes.

5.4.1 Minimax designs on $[0,1]^p$

We first illustrate the minimax performance and computation time of MMC-PSO on the unit hypercube $[0,1]^p$ in $p = 2, 4, 6$ and 8 dimensions. For brevity, only results for $p = 2$ and $p = 8$ are reported here, with additional results deferred to the Appendix. The simulation settings are as follows. For MMC-PSO, we generate $n = 20, 30, \cdots, 100$-point designs using $s = 10$ PSO particles with $N = 10^5$ approximating points. The maximum iterations in Algorithm 9 are set at $t_{mMc} = 500$ and $t_{pp} = 250$. Our implementation of MMC-PSO is written in C++, and is available in the R package `minimaxdesign` [197] in CRAN. For principal points, $N = 10^5$ approximating points are also used to provide a fair comparison with MMC-PSO. Lastly, for BIP, designs of the same sizes are generated with the candidate set taken from the first 1,000 points of the Sobol' sequence. FFF designs are also generated from JMP 12 using the cluster centers option.

For each design, Figure 5.3 plots the minimax criterion (5.1) with $\mathcal{X} = [0,1]^p$ approximated by the first 10^7 points from the Sobol' sequence. For $p = 2$, designs generated using MMC-PSO have the lowest minimax distance of the four methods for all design sizes n, which shows the proposed method indeed provides better minimax designs compared to existing methods. FFF designs, on the other hand, have the largest minimax distance for

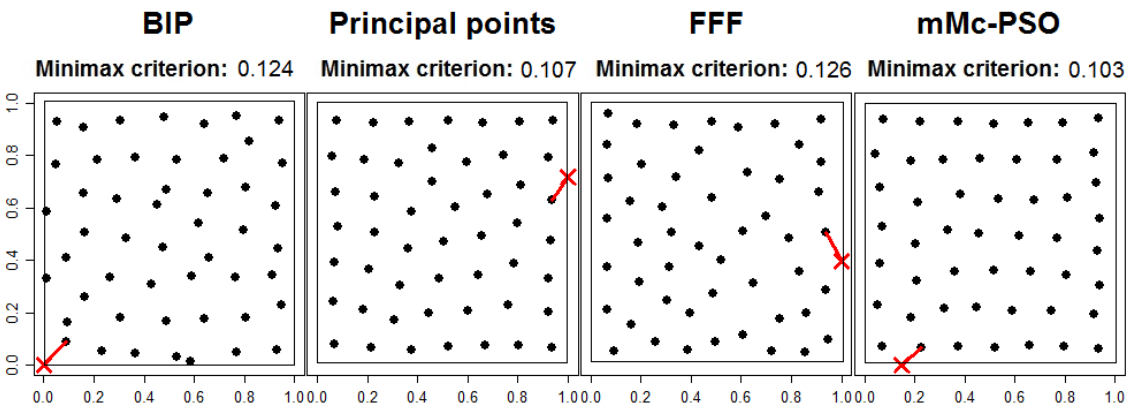

Figure 5.4: Four different 50-point designs for $[0, 1]^2$. The red line on each plot connects the point in $[0, 1]^2$ furthest from the design (marked by 'x') to its nearest design point, with its length equal to the minimax criterion. The proposed method MMC-PSO again provides the best minimax design.

nearly all design sizes. Surprisingly, designs generated using BIP also have large minimax distances, suggesting that a candidate set of 1,000 design points is insufficient for representing the unit hypercube even in 2 dimensions. On the other hand, even though principal points provide relatively higher minimax distance compared to MMC-PSO, it is consistently better than BIP or FFF. Hence, although principal points are not intended for minimax use, the minimax performance of these designs can be quite good. From Figure 5.4, which plots the 50-point designs for the four methods, principal points and MMC-PSO also enjoy a more visually uniform coverage of $[0, 1]^2$ compared to FFF and BIP.

From the right plot of Figure 5.3, similar results hold for $p = 8$ as well. MMC-PSO again provides the best minimax designs, with the improvement gap in minimax distance greater than that for $p = 2$. This suggests that MMC-PSO provides an increasing improvement over existing methods as dimension p increases. A contributing factor is the ability for MMC-PSO to manipulate a larger number of approximating points N compared to FFF or BIP, an observation which was made in Section 3.2.2. This then allows the proposed algorithm to provide better minimax designs in high-dimensions.

For computation time, Figure 5.5 plots the time (in log-seconds) required for each of the four methods, with computation performed on a 6-core 3.2 Ghz desktop computer.

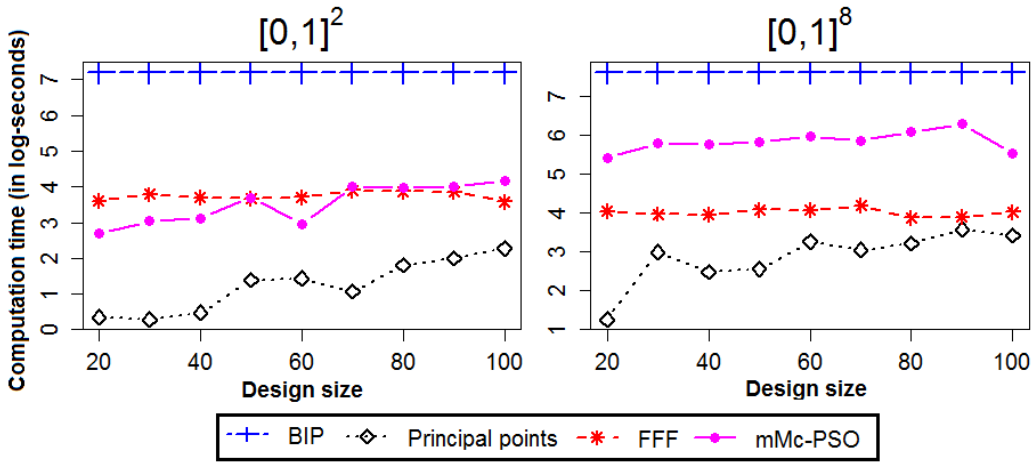

Figure 5.5: Time (in log-seconds) required for generating designs on $[0,1]^p$. The computation times for MMC-PSO are slightly higher than principal points and FFF, but lower than BIP.

Since the BIP optimization in (5.2) searches for the *smallest design* for a fixed minimax criterion, instead of the *smallest criterion* for a fixed design size, the timing for each BIP design is instead reported as the average time needed to generate all $n = 20, 30, \cdots, 100$-point designs. From Figure 5.5, the computation time for MMC-PSO appears to be quite reasonable. For $p = 2$, this time ranges from 15 to 90 seconds, whereas for $p = 8$, this time ranges from 4 to 8 minutes. Not surprisingly, BIP takes the longest computation time, requiring nearly 30 minutes for each design. FFF designs can be computed faster than MMC-PSO, but provide inferior minimax performance since fewer approximating points can be used. Lastly, although principal points provide higher minimax distances than MMC-PSO, they can be computed the quickest of the four methods. These points can therefore be used as crude minimax designs when computation time is limited.

5.4.2 Minimax designs on convex and bounded sets

Next, we investigate the minimax performance of MMC-PSO for other convex and bounded design regions. Although much of existing literature considers designs on $[0,1]^p$, designs on other design regions are also of practical importance. For example, in studying the

effects of temperature and pressure on injection molding, a hypercube design may be inappropriate since, from an engineering perspective, regions with high temperature and pressure may cause combustion of molding material, and experimental runs allocated in these regions therefore become wasted. MMC-PSO can be easily modified to generate minimax designs on design regions $\mathcal{X}$ which are convex and bounded. Convexity of $\mathcal{X}$ is necessary, since it ensures the C_q-centers updates in MMC-PSO remain in $\mathcal{X}$.

As mentioned previously, the key reason for using low-discrepancy sequences as the representative sample $\{\mathbf{y}_j\}_{j=1}^N$ is because such sequences provide a better approximation of the integral in (5.6). The question is how to generate these sequences for non-hypercube design regions, and to this end, this section is divided into two parts. First, when the Rosenblatt inverse transform for $U(\mathcal{X})$ (defined later) is easy to compute, there is an easy way to generate such sequences on $\mathcal{X}$. We illustrate this by computing minimax designs on the unit simplex and ball. When this transform is difficult to compute, uniform random sampling can be used as a last resort. This latter scenario is demonstrated using the motivating air quality example in Section 2.2.

Minimax clustering using the Rosenblatt transform

We begin by first defining the Rosenblatt transform $t_{\mathcal{X}}$:

Definition 15. *Let $\mathcal{X} \subseteq \mathbb{R}^p$, and define the random vector $\mathbf{X} = (X_1, \cdots, X_p) \sim U(\mathcal{X})$. The Rosenblatt transform is defined as the transform $t_{\mathcal{X}} : \mathbb{R}^p \to \mathbb{R}^p$ satisfying:*

$$(x_1, \cdots, x_p) \mapsto (y_1, \cdots, y_p), \text{ where } y_1 = F_1(x_1),\ y_i = F_i(x_i|x_1, \cdots, x_{i-1}), \quad i = 2, \cdots, p, \tag{5.13}$$

where $F_1(\cdot)$ is the distribution function (d.f.) of X_1, and $F_i(\cdot|x_1, \cdots, x_{i-1})$ is the conditional d.f. of X_i given $X_1, \cdots, X_{i-1}$.

It can be shown [2] that the inverse Rosenblatt transform of a low-discrepancy sequence on $[0, 1]^p$ also has low-discrepancy on $\mathcal{X}$. Hence, when $t_{\mathcal{X}}^{-1}$ can be easily computed, mini-

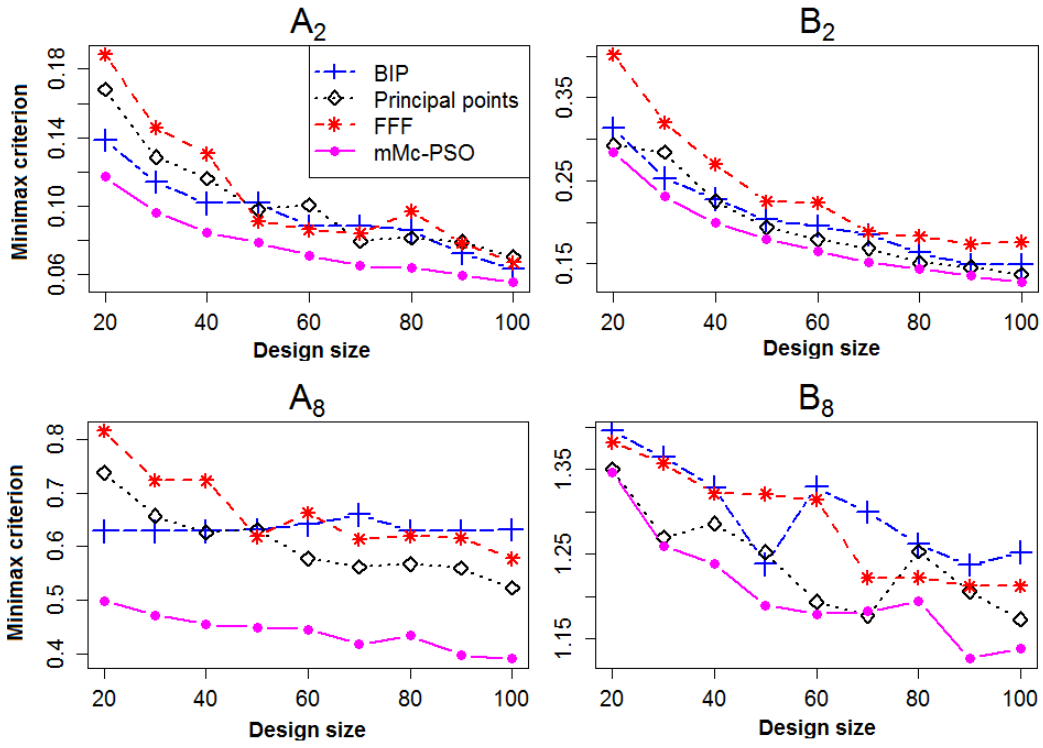

Figure 5.6: Minimax criterion for various design sizes on A_2, B_2, A_8 and B_8. Designs from MMC-PSO consistently give the lowest minimax distance for nearly all design sizes.

max designs can be generated with Algorithm 9 by simply taking the representative points $\{\mathbf{y}_j\}_{j=1}^N$ as the inverse transform of a Sobol' sequence.

Fortunately, when $\mathcal{X}$ is regularly-shaped, closed-form equations exist for the inverse Rosenblatt transform $t_{\mathcal{X}}^{-1}$. Transforms for common geometric shapes can be found in [2]. Using these equations, we generate minimax designs for the two regions:

1. The *unit simplex* in $\mathbb{R}^p$: $A_p \equiv \{(x_1, \cdots, x_p) \in \mathbb{R}^p \ : \ 0 \leq x_1 \leq \cdots \leq x_p \leq 1\}$,
2. The *unit ball* in $\mathbb{R}^p$: $B_p \equiv \{(x_1, \cdots, x_p) \in \mathbb{R}^p \ : \ x_1^2 + \cdots + x_p^2 \leq 1\}$.

The simulation settings are the same as before, with the exception that the candidate set for BIP is taken as the inverse transform of the first 1,000 points of a Sobol' sequence. Figure 5.6 plots the minimax criterion of designs for $p = 2$ and $p = 8$, and Figure 5.7 plots the corresponding 80-point designs. Two interesting observations can be made. First, for both $p = 2$ and $p = 8$, MMC-PSO provides the best minimax designs for every design

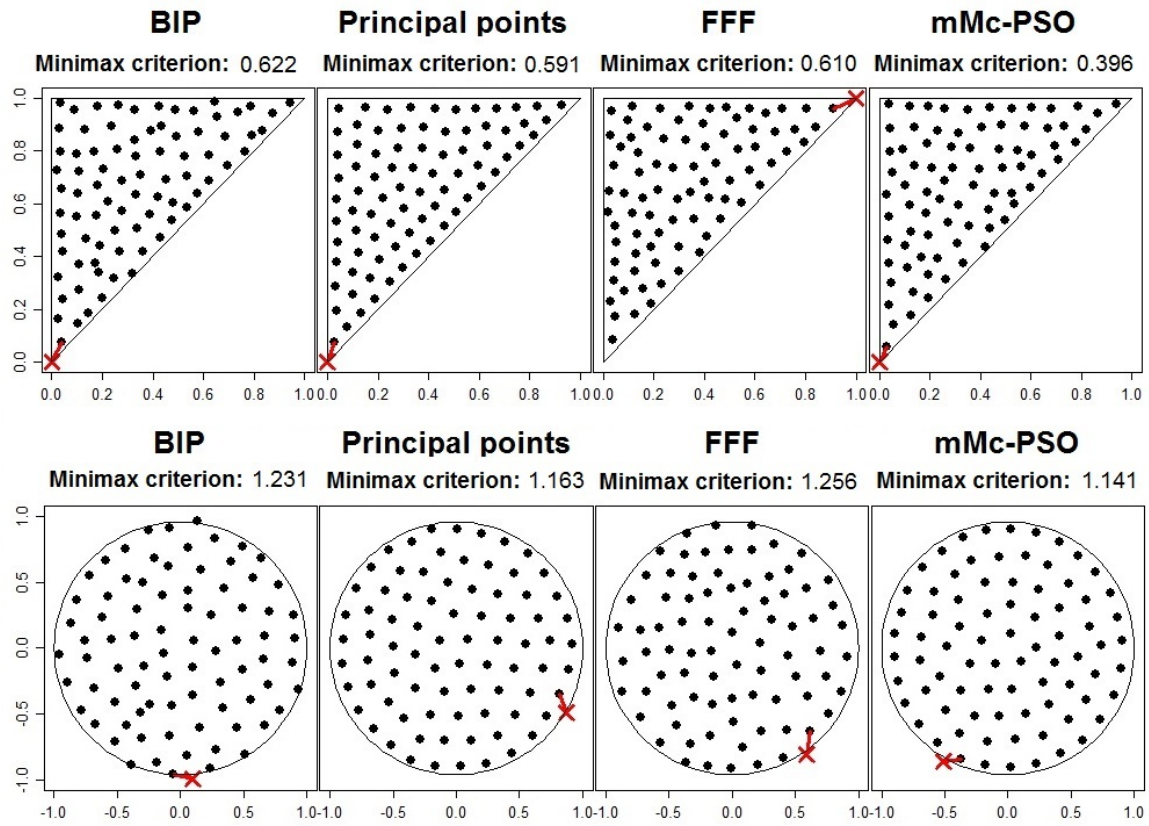

Figure 5.7: Four different 80-point designs for A_2 and B_2. The red line connects the point in $\mathcal{X}$ furthest from the design (marked by 'x') to its nearest design point, with its length equal to the minimax criterion. The proposed method MMC-PSO again provides the best minimax designs.

size n, which confirms the superiority of the proposed method in both low and high dimensions. Second, compared to principal points, MMC-PSO performs much better for the unit simplex A_p compared to the unit ball B_p. This can be intuitively justified by the fact that both the arithmetic mean and C_∞-center of a unit ball correspond to the same point, the center of the ball. However, when the design region is highly asymmetric, these two centers can indeed be quite different, which explains the sizable improvement of MMC-PSO over principal points for the unit simplex A_p.

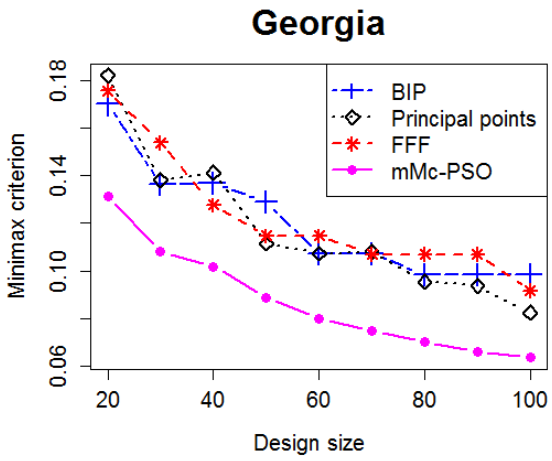

Figure 5.8: Minimax criterion for various design sizes on Georgia. Designs from MMC-PSO give the best minimax designs for all design sizes.

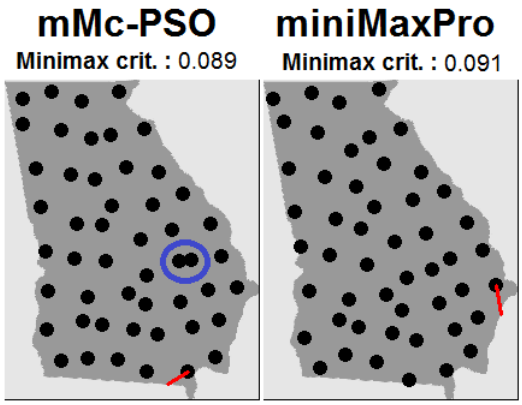

Figure 5.9: 50-point designs on Georgia using MMC-PSO and MINIMAXPRO. The refinement step in the latter corrects some visual non-uniformities in the former design (circled in blue).

Back to the motivating example

When $\mathcal{X}$ is irregularly-shaped, the inverse transform $t_{\mathcal{X}}^{-1}$ can be difficult to compute. In this case, the approximating points $\{\mathbf{y}_j\}_{j=1}^N$ can be generated using uniform random sampling on $\mathcal{X}$. We illustrate this using the earlier example of air quality monitoring in the state of Georgia. Note that, while the state of Georgia is not convex, it is "convex enough" to ensure C_q-centers remain in $\mathcal{X}$, so the proposed method can still be applied.

Figure 5.8 compares the minimax performance of $n = 20, 30, \cdots, 100$-point designs generated on Georgia, with the 20-point designs plotted in Figure 5.9. The simulation settings used here are the same as before. From the first figure, the minimax performance of MMC-PSO is sizably lower than existing methods for all design sizes, which illustrates the effectiveness of the proposed algorithm. One caveat of MMC-PSO, however, is that the generated designs appear visually non-uniform. For example, the 50-point design from MMC-PSO in the left plot of Figure 5.9 shows several design points huddled closely together (such as the pair of points circled in blue), despite the design having a low minimax distance. One way to improve visual uniformity is to improve the uniformity of the design when projected onto the horizontal or vertical axis. This can be accomplished by perform-

ing the refinement step introduced in the following section. The right design in Figure 5.9, obtained by applying this refinement to the left design, is more visually uniform compared to the original design, despite having a slightly larger minimax distance. Users should therefore apply this refinement depending on whether visual uniformity or minimaxity is desired.

5.5 Minimax projection designs

As mentioned previously, minimax designs minimize the worst-case prediction error in computer experiment emulation [168]. However, when a computer experiment has a large number of input variables, minimax designs as defined in (5.1) may not be appropriate. This is because, by the *effect sparsity* principle [68], only a few of these inputs are expected to be active. Emulator designs in high dimensions should therefore provide not only good minimax performance on the full space $\mathcal{X}$, but also for *projected subspaces* of $\mathcal{X}$. Recent developments in this vein include the MaxPro designs proposed by [69], which minimize the criterion:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{d_{prod}(\mathbf{m}_i, \mathbf{m}_j)}, \quad d_{prod}(\mathbf{m}_i, \mathbf{m}_j) = \prod_{k=1}^{p} (m_{ik} - m_{jk})^2, \tag{5.14}$$

where $\mathbf{m}_i = (m_{i1}, \cdots, m_{ip})$ denotes the i-th design point. Extending this idea, we present below a new type of design called minimax projection designs, which are obtained by refining the minimax design from MMC-PSO using the MaxPro criterion in (5.14).

In words, this refinement step *improves* projected minimaxity while *maintaining* the low minimax distance of the original MMC-PSO design. The details are as follows. Let $\mathcal{D} = \{\mathbf{m}_i\}_{i=1}^n$ be the design generated by MMC-PSO. Define the minimax distance of *each*

Algorithm 10 Minimax projection designs

1: **function** MINIMAXPRO($\cdots$) ▷ $\cdots$ - MMC-PSO params.
- Generate an n-point minimax design $\mathcal{D} = \{\mathbf{m}_i\}_{i=1}^n \leftarrow$ MMC-PSO($\cdots$).

2: **repeat**

3: **for** $i = 1, \cdots, n$ **do**
- Update $\{d_i\}_{i=1}^n$ in (5.15).
- Update $d^* = \max_i d_i$.
- Update $\mathbf{m}_i$ by (5.16).

4: **until** design points converge.
- **return** miniMaxPro design $\{\mathbf{m}_i\}_{i=1}^n$.

design point $\mathbf{m}_i$ as:

$$d_i = \sup_{\mathbf{x} \in \mathcal{X}_i} \|\mathbf{x} - \mathbf{m}_i\|, \quad \text{where } \mathcal{X}_i = \{\mathbf{x} \in \mathcal{X} : \|\mathbf{x} - \mathbf{m}_i\| \leq \|\mathbf{x} - \mathbf{m}_j\|,\ \forall j = 1, \cdots, n\} \tag{5.15}$$

is the collection of points in $\mathcal{X}$ closest in distance in $\mathbf{m}_i$. Note that the overall minimax distance in (5.1) is simply the maximum of these distances, $d^* = \max_{i=1,\cdots,n} d_i$. For each point $\mathbf{m}_i$, the refinement step consist of two parts. First, compute the minimax distances $\{d_i\}_{i=1}^n$ and d^*. Next, update $\mathbf{m}_i$ by the optimization:

$$\mathbf{m}_i \leftarrow \operatorname*{argmin}_{\mathbf{m} \in \mathbb{R}^p} \sum_{j=1, j \neq i}^{n} \frac{1}{d_{prod}(\mathbf{m}, \mathbf{m}_i)} \quad \text{s.t.} \quad \|\mathbf{m} - \mathbf{m}_i\| \leq d^* - d_i,\ \mathbf{m}_i \in \mathcal{X}. \tag{5.16}$$

This update can be viewed as the block-wise minimization of the MaxPro criterion (5.14) for the i-th design point $\mathbf{m}_i$, with the constraint $\|\mathbf{m} - \mathbf{m}_i\| \leq d^* - d_i$ ensuring the updated point is sufficiently close to the previous point. In our implementation, (5.16) is computed using the R package `nloptr` [198]. Repeating this two-stage refinement for each design point until convergence gives a point set which enjoys good space-filling properties after projections. Algorithm 10 summarizes the detailed steps for generating this so-called minimax projection (miniMaxPro) design.

An appealing feature of miniMaxPro designs is that its projective space-fillingness does not come at a cost of increased minimax distance! That is, the minimax distance of the converged miniMaxPro design has the *same* minimax distance on $\mathcal{X}$ as the original design

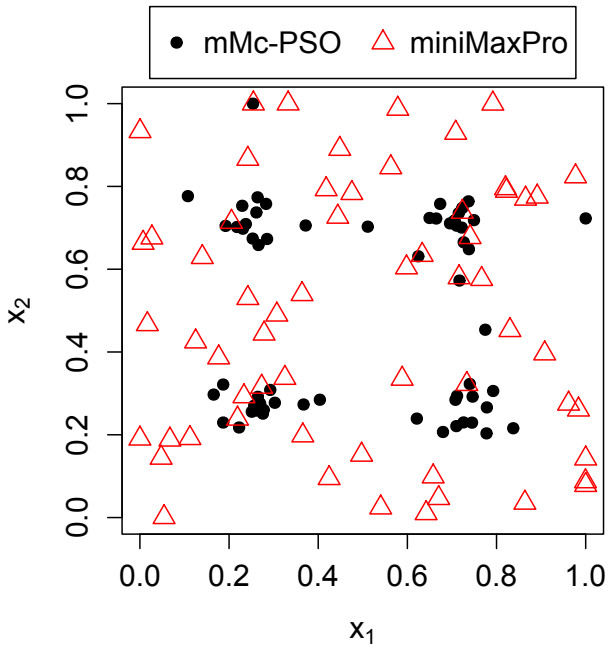

Figure 5.10: A 2-d projection of 60-point MMC-PSO and miniMaxPro designs. The refinement step in MINIMAXPRO improves projected minimaxity.

from MMC-PSO. This is stated formally in the following proposition:

Proposition 5. *When* $\{d_i\}_{i=1}^n$ *and* d^* *are computed exactly, the two-stage refinement in lines 6 - 8 of Algorithm 10 does not increase the minimax distance of* $\mathcal{D}$ *in line 2.*

The proof of this proposition relies on the constraint $\|\mathbf{m} - \mathbf{m}_i\| \leq d^* - d_i$ in (5.16); see Appendix for details. In practice, $\{d_i\}_{i=1}^n$ and d^* are *estimated* by appsroximating $\mathcal{X}$ using a finite representative set $\{\mathbf{y}_m\}_{m=1}^N$ (a Sobol' sequence is used in our implementation), so the overall minimax distance may increase after refinement. However, this increase is quite small when the number of approximating points N is large (i.e., $N = 10^5$), as shown in the simulations below.

To illustrate the effectiveness of this refinement, Figure 5.10 plots a two-dimensional projection of the 60-point design from MMC-PSO on $[0, 1]^8$ and its corresponding miniMaxPro design. The MMC-PSO design clearly has poor minimax coverage after projection onto this 2-d subspace, with points closely focused around the four points $(0.5 \pm 0.25, 0.5 \pm$

0.25). The miniMaxPro design, on the other hand, exhibits much better minimax performance after projection, which shows the refinement performs as intended.

Since one use of miniMaxPro designs is for computer experiment emulation, we compare its performance with two existing computer experiment designs: the MaxPro design [69] and the FFF design [177]. Three metrics are used to evaluate projective space-fillingness: mM_k, avg_k and Mm_k, which are defined as:

$$\begin{aligned}
\text{mM}_k &= \max_{r=1,\cdots,\binom{p}{k}} \sup_{\mathbf{x}\in\mathscr{P}_r(\mathfrak{X})} \left\{ \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\|\mathbf{x}-\mathscr{P}_r\mathbf{m}_i\|^{2k}} \right\}^{-1/(2k)}, \\
\text{avg}_k &= \max_{r=1,\cdots,\binom{p}{k}} \int_{\mathscr{P}_r(\mathfrak{X})} \|\mathbf{x} - Q(\mathbf{x}, \{\mathscr{P}_r\mathbf{m}_i\}_{i=1}^n)\|\, d\mathbf{x} \text{ and} \\
\text{Mm}_k &= \min_{r=1,\cdots,\binom{p}{k}} \frac{1}{\binom{n}{2}} \left\{ \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{1}{\|\mathscr{P}_r\mathbf{m}_i - \mathscr{P}_r\mathbf{m}_j\|^{2k}} \right\}^{-1/(2k)}.
\end{aligned}$$

Here, $r = 1, \cdots, \binom{p}{k}$ enumerates all projections of $\mathfrak{X} \subseteq \mathbb{R}^p$ onto a subspace of dimension k, with $\mathscr{P}_r$ its corresponding projection operator. The metrics mM_k and Mm_k were proposed in [69] to incorporate the minimax and maximin index of the design when projected into k dimensions. The last metric avg_k measures the average distance to a design point when projected into k dimensions. Larger values of Mm_k suggest better space-fillingness in terms of maximin, whereas smaller values of mM_k and avg_k indicate better space-fillingness in terms of minimax and average distance, respectively.

Figure 5.11 plots mM_k, avg_k and Mm_k for the 60-point MaxPro, FFF, miniMaxPro and the design from MMC-PSO (we refer to the latter as simply "minimax design" below). Similar results hold for other design sizes, and are not reported for brevity. For the minimax metric mM_k, both the miniMaxPro and minimax designs enjoy sizably improved performance in moderate dimensions ($4 \leq k \leq 8$). In lower dimensions ($1 \leq k \leq 3$), the refinement step for the miniMaxPro design allows it to be comparable with MaxPro. For the average distance metric avg_k, the miniMaxPro design appears to be the best choice over all projection dimensions. For the maximin metric Mm_k, the minimax and miniMaxPro

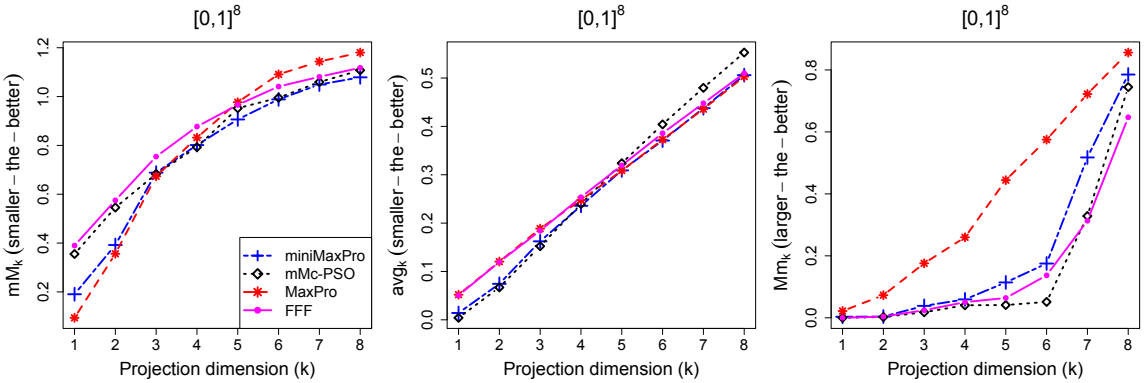

Figure 5.11: mM_k, avg_k and Mm_k for four different 60-point designs on $[0,1]^8$. The proposed miniMaxPro design provides the best performance for mM_k and avg_k, but performs worse for Mm_k.

designs give poorer performance to MaxPro. The refinement step for the latter, however, allows for sizable improvements with respect to maximin. To summarize, miniMaxPro designs appear to enjoy an improvement over existing designs in terms of projected minimax and average distance, but this comes at a cost of poorer performance for the projected maximin criterion.

5.6 Discussion

Minimax designs, by *minimizing* the *maximum* distance from any point in the design space $\mathcal{X} \subseteq \mathbb{R}^p$ to its closest design point, provide uniform coverage of $\mathcal{X}$ in the worst-case. Despite its many uses in computer experiments, optimal sensor placement and resource allocation problems, there have been little work on generating these designs efficiently. In this chapter, we propose a new algorithm called MMC-PSO for computing minimax designs on convex and bounded design spaces, and demonstrate the efficiency of this method in low and highdimensions. Simulations on the unit hypercube, the unit simplex and ball, and the state of Georgia show that MMC-PSO provides better minimax designs compared to existing methods in literature. A new experimental design, called miniMaxPro designs, can then be constructed by refining the minimax design from MMC-PSO to ensure good projective space-fillingness.

Despite the developments in this chapter, there are still many avenues for further work. One of these is exploring the properties of minimax designs when the Euclidean norm is replaced by another norm for $\|\cdot\|$ in (5.1). Pursuing this may reveal better ways for generating designs in high-dimensions with good projective space-filling properties. Another direction is to explore more sophisticated hybridization schemes (e.g., [199, 200]) for incorporating PSO within clustering algorithms. This allows better minimax designs to be generated using less computational resources.

CHAPTER 6

ACTIVE MATRIX COMPLETION WITH UNCERTAINTY QUANTIFICATION

6.1 Introduction

Low-rank matrices play an important role in a variety of applications in statistics, machine learning and engineering. For many such applications, however, only a small portion of matrix entries can be observed as data. The reasons for this are two-fold: the underlying matrix $\mathbf{X} \in \mathbb{R}^{m_1 \times m_2}$ can be high-dimensional, or the cost of observing each entry can be expensive. For example, in genetic studies, the expression levels of various genes across different diseases can be viewed as a low-rank matrix [201]. Here, not only is such a matrix high-dimensional (spanning millions of genes and thousands of diseases), but measuring the expression level at each gene-disease pair also requires expensive experiments. The problem of recovering the low-rank matrix $\mathbf{X}$ from noisy, incomplete observations is known as *noisy matrix completion* [202]. In this chapter, we propose a novel, information-theoretic approach for active sampling (or designing) of matrix entries in $\mathbf{X}$ via uncertainty quantification (UQ), and demonstrate its effectiveness over random sampling for noisy matrix completion.

In recent years, there has been significant progress on the topic of matrix completion, particularly on theoretical properties of such a completion via convex optimization. This includes the pioneering work of [203], [204] and [205], who established bounds on error convergence under uniform random sampling and nuclear-norm minimization. The noisy matrix completion problem – where matrix entries are observed with noise – has also received considerable attention, with important theoretical results in [202, 206, 207], among

The paper based on this chapter has been submitted to *IEEE Transactions on Signal Processing*.

others. We consider the latter noisy setting in our work.

This chapter presents a novel approach for *designing* the entries to observe in $\mathbf{X}$ for matrix completion, with the goal of maximizing information on $\mathbf{X}$ via such samples. While most of the matrix completion literature assumes that entries are sampled uniformly-at-random, there have been some recent work on adaptive sampling schemes. [49] employed several intuitive metrics for guiding sequential sampling. [208] used graph regularization methods with a query-by-committee framework for sequential sampling. [209] investigated the problem of active sequential sampling for completing positive semi-definite matrices. [210] proposed a method for querying entries by evaluating the instability of an underlying system of linear equations. Our approach differs from these works in several ways. First, we offer an *integrated* approach to sampling and UQ, in that the uncertainties for unobserved entries are employed within an integrated framework to guide active sampling. Second, this framework yields new insights on the link between information-theoretic sampling, compressive sensing, and statistical experimental design. Using such insights and the so-called *maximum entropy* principle [211], we derive an efficient algorithm for active sampling on $\mathbf{X}$.

To learn this adaptive sampling scheme, the proposed method also makes use of a new *uncertainty quantification* approach for noisy matrix completion. Here, UQ measures how *uncertain* the completed matrix entries are from their true values, given a partial observation of $\mathbf{X}$. UQ plays a central role in many areas in engineering and applied math [212], and for the matrix completion problem, this UQ can be nearly as valuable as the completed matrix itself. In the earlier gene study example, the UQ of gene expression levels at unobserved gene-disease pairs allows a biologist to test which genes are most influential for a particular disease. One way to perform UQ is via a stochastic model on $\mathbf{X}$; in this sense, Bayesian matrix completion methods [213, 214, 34] can be used to quantify uncertainty (even though this may not be their primary focus). Our UQ approach is novel in the following ways. First, using a new Bayesian modeling framework on $\mathbf{X}$, our method allows for effective

learning and UQ of the *subspaces* of $\mathbf{X}$ via an efficient Gibbs sampling algorithm. Second, our integrated framework incorporates this learned subspace information to guide active *sampling* on $\mathbf{X}$.

Our work also makes novel contributions to the topic of information-theoretic design for matrix completion. In recent years, there has been a large body of literature on information-theoretic design (e.g., for compressive sensing), including the seminal paper [215] on the connection between mutual information and parameter estimation for linear vector Gaussian channels, and its important developments [70, 216, 217] for compressive sensing and phase retrieval. Our approach differs from these works in that, instead of maximizing the mutual information between signal (i.e., $\mathbf{X}$) and observed entries (denoted as $\mathbf{Y}_\Omega$), we study a dual but equivalent problem of maximizing the *entropy* of observations $\mathbf{Y}_\Omega$. Using the maximum entropy principle, this dual view yields new insights on the link between matrix completion sampling and code design, and provides a simple, closed-form criterion for sequential sampling.

This integrated sampling approach also has interesting connections to the idea of *hyperparameter tuning* in machine learning [218]. There, hyperparameters refer to parameters which control certain properties of a learning algorithm [219]. The tuning of hyperparameters from data plays an important role in ensuring the effectiveness of state-of-the-art machine learning algorithms (e.g., Google's Cloud Machine Learning system [220]). In our framework, hyperparameters encode important subspace properties for the matrix $\mathbf{X}$. Given such hyperparameters, the proposed model yields a closed-form scheme for sequential sampling; however, these parameters need to be adaptively learned via the UQ method. Our integrated sampling strategy can be viewed as a *learning active learning* approach [221] for noisy matrix completion, in that it adaptively *learns* key subspace hyperparameters on $\mathbf{X}$, before using such parameters for *active learning*.

Contribution. We summarize three important contributions of our work. First, we present a novel *integrated* framework which tackles sampling and UQ for noisy matrix

completion, via a new Bayesian model for $\mathbf{X}$. Second, we reveal several insights on the role of compressive sensing (e.g., coherence) and coding design (e.g., Latin squares) on the sampling performance and UQ for noisy matrix completion, which then yields new results on error monotonicity and decay. Lastly, using such insights along with information-theoretic design principles, an efficient sampling scheme is developed, which can yield improved matrix completion performance over random sampling.

The chapter is organized as follows. Section 2 introduces a new Bayesian model framework for matrix completion. Section 3 reveals some useful insights on the role of coherence on UQ and error convergence. Section 4 outlines the maximum entropy design principle, then derives several novel sampling properties for initial and sequential learning on $\mathbf{X}$. Section 5 incorporates these properties into a practical sampling and UQ algorithm. Sections 6 and 7 demonstrate the effectiveness of the proposed methodology in simulation studies and in two real-world collaborative filtering datasets. Finally, Section 8 concludes with directions for future work.

6.2 A Bayesian model for matrix completion

We begin with a brief problem set-up, then introduce the singular matrix-variate Gaussian model for $\mathbf{X}$. This serves as a versatile probabilistic model for the low-rank matrices of interest. We then show how a Bayesian implementation of this model plays an important role in sampling and UQ.

6.2.1 Problem set-up

Let $\mathbf{X} = (X_{i,j}) \in \mathbb{R}^{m_1 \times m_2}$ be the low-rank matrix of interest. Suppose $\mathbf{X}$ is observed with noise at N indices $\Omega_{1:N} = \{(i_n, j_n)\}_{n=1}^{N} \subseteq [m_1] \times [m_2]$[1] (this is sometimes denoted as Ω for brevity). Let $Y_{i,j}$ be the observation at index $(i, j) \in \Omega$, and assume $Y_{i,j}$ follows the

[1] $[m] := \{1, \cdots, m\}$.

Gaussian noise model:

$$Y_{i,j} = X_{i,j} + \epsilon_{i,j}, \quad \epsilon_{i,j} \overset{i.i.d.}{\sim} \mathcal{N}(0, \eta^2). \tag{6.1}$$

Further let $\mathbf{X}_\Omega \in \mathbb{R}^N$ and $\mathbf{Y}_\Omega \in \mathbb{R}^N$ denote the vectorized entries of $\mathbf{X}$ and $\mathbf{Y}$ at observed indices Ω, and let $\mathbf{X}_{\Omega^c} \in \mathbb{R}^{m_1 m_2 - N}$ and $\mathbf{Y}_{\Omega^c} \in \mathbb{R}^{m_1 m_2 - N}$ denote the vectorized entries of $\mathbf{X}$ and $\mathbf{Y}$ at unobserved indices $\Omega^c = ([m_1] \times [m_2]) \setminus \Omega$. The noisy matrix completion problem aims to recover the full matrix $\mathbf{X}$ from the noisy and partial observations $\mathbf{Y}_\Omega$.

6.2.2 Model specification

The singular matrix-variate Gaussian distribution

The motivation for our model comes from the popular use of Gaussian processes for functional approximation [222]. There, the goal is to recover an unknown function $f : \mathbb{R}^p \to \mathbb{R}$ by observing it at several sampled points $\mathbf{f} = [f(\mathbf{x}_1), \cdots, f(\mathbf{x}_N)]^T$. Assuming f follows a Gaussian process parametrized by some correlation function, the vector $\mathbf{f}$ then follows a multivariate Gaussian distribution. This can then be used to derive closed-form expressions for (a) predicting the function f at unobserved points, and (b) quantifying the uncertainty of such predictions. The ability to quantify uncertainty in closed-form is an important advantage of Gaussian process learning over other learning methods. With this in mind, our strategy is to employ the so-called singular matrix-variate Gaussian model (introduced below) – an extension of Gaussian process modeling for low-rank matrices – to derive similar closed-form expressions for noisy matrix completion. Such expressions will then play a central role for UQ and active matrix sampling.

Consider now the following model for the low-rank matrix $\mathbf{X}$ (assumed to be normalized with zero mean):

Definition 16 (Singular matrix-variate Gaussian (SMG); Definition 2.4.1, [223])**.** *Let* $\mathbf{Z} \in \mathbb{R}^{m_1 \times m_2}$ *be a random matrix with entries* $Z_{i,j} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$ *for* $(i, j) \in [m_1] \times [m_2]$*. The ran-*

dom matrix $\mathbf{X}$ *has a* singular matrix-variate Gaussian (SMG) *distribution if* $\mathbf{X} \overset{d}{=} \mathscr{P}_{\mathscr{U}}\mathbf{Z}\mathscr{P}_{\mathscr{V}}$ *for some choice of projection matrices* $\mathscr{P}_{\mathscr{U}} = \mathbf{U}\mathbf{U}^T$ *and* $\mathscr{P}_{\mathscr{V}} = \mathbf{V}\mathbf{V}^T$*, where* $\mathbf{U} \in \mathbb{R}^{m_1 \times R}$*,* $\mathbf{U}^T\mathbf{U} = \mathbf{I}$*,* $\mathbf{V} \in \mathbb{R}^{m_2 \times R}$*,* $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ *and* $R < m_1 \wedge m_2$.[2] *We will denote this as* $\mathbf{X} \sim \mathcal{SMG}(\mathscr{P}_{\mathscr{U}}, \mathscr{P}_{\mathscr{V}}, \sigma^2, R)$.

In other words, a realization from the SMG distribution can be obtained by first (a) simulating a matrix $\mathbf{Z}$ from a Gaussian ensemble with variance σ^2 (i.e., a matrix with i.i.d. $\mathcal{N}(0, \sigma^2)$ entries), then (b) performing a left and right projection of $\mathbf{Z}$ using the projection matrices $\mathscr{P}_{\mathscr{U}}$ and $\mathscr{P}_{\mathscr{V}}$. Recall that the projection operator $\mathscr{P}_{\mathscr{U}} = \mathbf{U}\mathbf{U}^T \in \mathbb{R}^{m_1 \times m_1}$ maps a vector in $\mathbb{R}^{m_1}$ to its orthogonal projection on the R-dimensional subspace $\mathscr{U}$ spanned by the columns of $\mathbf{U}$. By performing this left-right projection, the resulting matrix $\mathbf{X} = \mathscr{P}_{\mathscr{U}}\mathbf{Z}\mathscr{P}_{\mathscr{V}}$ can be shown to be of rank $R < m_1 \wedge m_2$, with its row and column spaces $\mathscr{U}$ and $\mathscr{V}$ corresponding to the subspaces for $\mathscr{P}_{\mathscr{U}}$ and $\mathscr{P}_{\mathscr{V}}$. With a small choice of R, this distribution provides a flexible model for the low-rank structure of $\mathbf{X}$.

We will illustrate throughout this chapter why projection matrices provide a useful parametrization for both sampling and UQ. The reasons are two-fold. First, it is known [224] that for each projection operator $\mathscr{P} \in \mathbb{R}^{m \times m}$ of rank R, there exists a unique R-dim. hyperplane (or an R-plane) in $\mathbb{R}^m$ containing the origin which corresponds to the image of such a projection. This connects the space of rank R projection matrices and the *Grassmann manifold* $\mathcal{G}_{R,m-R}$, the space of R-planes in $\mathbb{R}^m$. Viewed this way, the projection matrices parametrizing $\mathbf{X} \sim \mathcal{SMG}(\mathscr{P}_{\mathscr{U}}, \mathscr{P}_{\mathscr{V}}, \sigma^2, R)$ encode valuable information on the row and column spaces of $\mathbf{X}$. Second, since the projection of a Gaussian random vector is still Gaussian, the left-right projection of the Gaussian ensemble $\mathbf{Z}$ results in each entry of $\mathbf{X}$ being Gaussian-distributed as well. This is crucial for deriving closed-form expressions for sampling and UQ below.

The following lemma provides several important properties of this model for matrix completion:

[2] $m_1 \wedge m_2 := \min(m_1, m_2), m_1 \vee m_2 := \max(m_1, m_2)$.

Lemma 6 (Distributional properties). *Let* $\mathbf{X} \sim \mathcal{SMG}(\mathcal{P}_{\mathcal{U}}, \mathcal{P}_{\mathcal{V}}, \sigma^2, R)$*, with* $\mathcal{P}_{\mathcal{U}} \in \mathbb{R}^{m_1 \times m_1}$*,* $\mathcal{P}_{\mathcal{V}} \in \mathbb{R}^{m_2 \times m_2}$*,* $\sigma^2 > 0$ *and* $R < m_1 \wedge m_2$ *known. Define the linear space*

$$\mathcal{T} := \bigcup_{u_k \in \mathcal{U}, v_k \in \mathcal{V}} \text{span}(\{\mathbf{u}_k \mathbf{v}_k^T\}_{k=1}^R), \tag{6.2}$$

where $\mathcal{U} \in \mathcal{G}_{R,m_1-R}$ *and* $\mathcal{V} \in \mathcal{G}_{R,m_2-R}$ *are the* R*-planes for* $\mathcal{P}_{\mathcal{U}}$ *and* $\mathcal{P}_{\mathcal{V}}$*. Then:*

(a) It follows that $\mathbf{X} \in \mathcal{T}$*, with the density of* $\mathbf{X}$ *given by*

$$f(\mathbf{X}) = (2\pi\sigma^2)^{-R^2/2} \text{ etr}\left\{-\frac{1}{2\sigma^2}\left[(\mathbf{X}\mathcal{P}_{\mathcal{V}})^T(\mathcal{P}_{\mathcal{U}}\mathbf{X})\right]\right\}, \tag{6.3}$$

where $\text{etr}(\cdot) := \exp\{\text{tr}(\cdot)\}$*. Equivalently,* $\text{vec}(\mathbf{X}) \in \mathbb{R}^{m_1 m_2}$ *follows the degenerate Gaussian distribution* $\mathcal{N}\{\mathbf{0}, \sigma^2(\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})\}$ *when restricted to* $\text{vec}(\mathcal{T})$*.*

(b) Consider the block decomposition of $\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}}$*:*

$$\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}} = \begin{pmatrix} (\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})_{\Omega} & (\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})_{\Omega,\Omega^c} \\ (\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})_{\Omega,\Omega^c}^T & (\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})_{\Omega^c} \end{pmatrix}. \tag{6.4}$$

Conditional on the observed noisy entries $\mathbf{Y}_{\Omega}$*, the unobserved entries* $\mathbf{X}_{\Omega^c}$ *follow the distribution*[3]

$$[\mathbf{X}_{\Omega^c}|\mathbf{Y}_{\Omega}] \sim \mathcal{N}(\mathbf{X}_{\Omega^c}^P . \mathbf{\Sigma}_{\Omega^c}^P). \tag{6.5}$$

Here, $\gamma^2 := \eta^2/\sigma^2$*, and*

$$\mathbf{R}_N(\Omega) := (\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})_{\Omega} \in \mathbb{R}^{N \times N}, \tag{6.6}$$

$$\mathbf{X}_{\Omega^c}^P := (\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})_{\Omega,\Omega^c}^T \left[\mathbf{R}_N(\Omega) + \gamma^2\mathbf{I}\right]^{-1} \mathbf{Y}_{\Omega}, \tag{6.7}$$

$$\mathbf{\Sigma}_{\Omega^c}^P := \sigma^2\Big\{(\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})_{\Omega^c} - (\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})_{\Omega,\Omega^c}^T \left[\mathbf{R}_N(\Omega) + \gamma^2\mathbf{I}\right]^{-1} (\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})_{\Omega,\Omega^c}\Big\}.$$

[3]Here, $[X]$ denotes the distribution of a random variable (r.v.) X, and $[X|Y]$ denotes the distribution of a r.v. X given r.v. Y.

Remark: Lemma 6 reveals two key properties of the SMG model. First, *prior* to observing data, part (a) shows that the low-rank matrix $\mathbf{X}$ lies on the linear space $\mathcal{T}$, and follows a degenerate multivariate Gaussian distribution with mean zero and covariance matrix $\sigma^2(\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})$ (the Kronecker product of projection matrices for $\mathbf{X}$). Second, *after* observing the noisy entries $\mathbf{Y}_\Omega$, part (b) shows that the conditional distribution of $\mathbf{X}_{\Omega^c}$ (the unobserved entries in $\mathbf{X}$) given $\mathbf{Y}_\Omega$ is still multivariate Gaussian, with closed-form expressions for its mean vector $\mathbf{X}^P_{\Omega^c}$ and covariance matrix $\mathbf{\Sigma}^P_{\Omega^c}$ in (6.7).

Prior specification

In most practical settings, there is little-to-no prior knowledge on either the rank of $\mathbf{X}$ or its subspaces. In such cases, a Bayesian approach [225] assigns non-informative prior distributions to model parameters, which here are the projection matrices $\mathcal{P}_{\mathcal{U}}$, $\mathcal{P}_{\mathcal{V}}$, the variance parameters η^2, σ^2 and the matrix rank R. To this end, we assume that $\mathcal{P}_{\mathcal{U}}$ and $\mathcal{P}_{\mathcal{V}}$ are uniformly and independently distributed over their corresponding Grassmann manifolds, i.e.:

$$[\mathcal{P}_{\mathcal{U}}] \sim U(\mathcal{G}_{R,m_1-R}), \quad [\mathcal{P}_{\mathcal{V}}] \sim U(\mathcal{G}_{R,m_2-R}). \tag{6.8}$$

For the remaining model parameters, we assign the non-informative priors:

$$[\eta^2] \sim IG(\alpha_{\eta^2}, \beta_{\eta^2}),\ [\sigma^2] \sim IG(\alpha_{\sigma^2}, \beta_{\sigma^2}),\ \mathbb{P}(R = r) = \pi_r, \tag{6.9}$$

where $\sum_{r=1}^{m_1 \wedge m_2} \pi_r = 1$, and $IG(\alpha, \beta)$ is the Inverse-Gamma distribution with shape and rate parameters α and β. These Inverse-Gamma priors provide so-called conjugate priors [225] for the proposed model, which allow for an efficient, closed-form sampling scheme for UQ (see Section 6.5.1). The full model is summarized in Table 6.1 and visualized in Figure 6.1.

Table 6.1: Model specification for noisy matrix completion.

Model	*Distribution*
Observations	$[\mathbf{Y}_\Omega \vert \mathbf{X}, \eta^2]$: $Y_{i,j} \overset{i.i.d.}{\sim} \mathcal{N}(X_{i,j}, \eta^2)$
Low-rank matrix	$[\mathbf{X} \vert \mathscr{P}_\mathscr{U}, \mathscr{P}_\mathscr{V}, \sigma^2, R]$: $\mathbf{X} \sim \mathcal{SMG}(\mathscr{P}_\mathscr{U}, \mathscr{P}_\mathscr{V}, \sigma^2, R)$
Priors	$[\mathscr{P}_\mathscr{U}, \mathscr{P}_\mathscr{V}, \sigma^2, \eta^2, R]$ $= [\mathscr{P}_\mathscr{U} \vert R]\ [\mathscr{P}_\mathscr{V} \vert R]\ [\eta^2][\sigma^2][R]$
Mtx. subspaces	$[\mathscr{P}_\mathscr{U} \vert R] \sim U(\mathcal{G}_{R,m_1-R})$
	$[\mathscr{P}_\mathscr{V} \vert R] \sim U(\mathcal{G}_{R,m_2-R})$
Meas. noise	$[\eta^2] \sim IG(\alpha_{\eta^2}, \beta_{\eta^2})$
Mtx. variance	$[\sigma^2] \sim IG(\alpha_{\sigma^2}, \beta_{\sigma^2})$
Rank	$[R] \sim$ Discrete$(\{\pi_r\}_{r=1}^{m_1 \wedge m_2})$

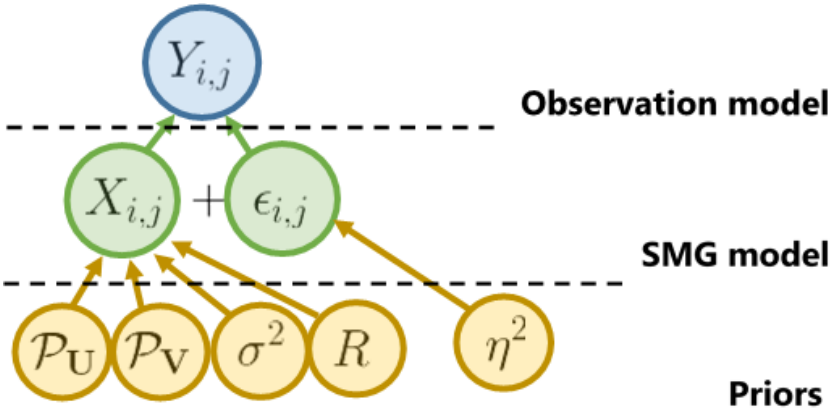

Figure 6.1: Visualization of model specification.

6.2.3 Connection to existing estimators

The following lemma reveals an inherent connection between the SMG model and existing completion methods:

Lemma 7 (MAP estimator)**.** *Assume the model in Table 6.1, with* $\pi_r \propto 1$*, and* η^2 *and* σ^2 *fixed. Conditional on* $\mathbf{Y}_\Omega$*, the maximum-a-posteriori (MAP) estimator* $\tilde{\mathbf{X}}$ *for* $\mathbf{X}$ *becomes*

$$\underset{\mathbf{X} \in \mathbb{R}^{m_1 \times m_2}}{\text{Argmin}} \left[\frac{\|\mathbf{Y}_\Omega - \mathbf{X}_\Omega\|_2^2}{\eta^2} + \log(2\pi\sigma^2)\text{rank}^2(\mathbf{X}) + \frac{\|\mathbf{X}\|_F^2}{\sigma^2} \right], \tag{6.10}$$

where $\|\mathbf{X}\|_F = \sqrt{\sum_{i,j} X_{i,j}^2}$ *is the Frobenius norm of* $\mathbf{X}$*.*

The MAP estimator $\tilde{\mathbf{X}}$ in (6.10) reveals an illuminating connection between our model and existing (deterministic) matrix completion methods (see [226] and references therein).

Consider the following approximation to the MAP formulation (6.10). Treating $\log(2\pi\sigma^2)\text{rank}^2(\mathbf{X})$ as a Lagrange multiplier, we can replace this by the constraint $\text{rank}(\mathbf{X}) \leq \sqrt{\xi}$. Changing this constraint back to its Lagrangian form, and replacing the rank function $\text{rank}(\mathbf{X})$ by its nuclear norm $\|\mathbf{X}\|_*$ (its tightest convex relaxation [206]), the optimization in (6.10) becomes:

$$\underset{\mathbf{X}\in\mathbb{R}^{m_1\times m_2}}{\text{Argmin}} \left[\|\mathbf{Y}_\Omega - \mathbf{X}_\Omega\|_2^2 + \lambda\left\{\alpha\|\mathbf{X}\|_* + (1-\alpha)\|\mathbf{X}\|_F^2\right\}\right], \tag{6.11}$$

for some choice of $\lambda > 0$ and $\alpha \in (0,1)$. Using (6.11) to approximate (6.10), the MAP estimator can then be viewed as an analogue of the *elastic net* estimator [103] from linear regression for noisy matrix completion.

To see the connection between the MAP estimator $\tilde{\mathbf{X}}$ and existing matrix completion methods, set $\alpha = 1$ in (6.11). The problem then reduces to:

$$\hat{\mathbf{X}} = \underset{\mathbf{X}\in\mathbb{R}^{m_1\times m_2}}{\text{Argmin}} \left[\sum_{(i,j)\in\Omega}(Y_{i,j} - X_{i,j})^2 + \lambda\|\mathbf{X}\|_*\right], \tag{6.12}$$

which is precisely the nuclear-norm formulation widely used for matrix completion [203, 204, 205]. This link will be used later to develop an efficient subspace learning algorithm for active matrix sampling.

6.3 Coherence and uncertainty quantification

Next, we review the notion of (subspace) *coherence*, then discuss its connection to UQ and error convergence.

6.3.1 The role of coherence in matrix completion

Consider the following definition of subspace coherence from [203] (ignoring scaling factors):

Definition 17 (Coherence; Definition 1.2, [203])**.** *Let $\mathcal{U} \in \mathcal{G}_{R,m-R}$ be an R-plane in $\mathbb{R}^m$, and let $\mathcal{P}_\mathcal{U}$ be the orthogonal projection onto $\mathcal{U}$. The* coherence *of subspace $\mathcal{U}$ with respect*

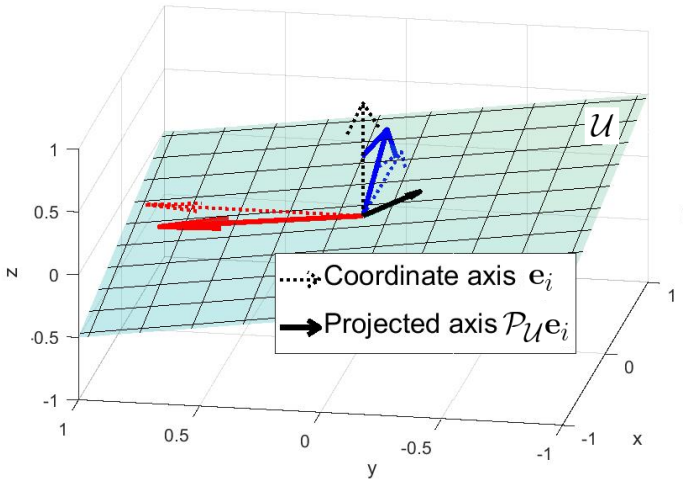

Figure 6.2: A visualization of near-maximal coherence (red basis vector) and minimal coherence (black basis vector) for subspace $\mathcal{U}$.

to the i-th basis vector, $\mathbf{e}_i$, is defined as

$$\mu_i(\mathcal{U}) := \|\mathcal{P}_{\mathcal{U}}\mathbf{e}_i\|_2^2, \tag{6.13}$$

and the coherence *of $\mathcal{U}$ is defined as* $\mu(\mathcal{U}) = \max_{i=1,\cdots,m} \mu_i(\mathcal{U})$.

In words, coherence measures how *correlated* a subspace $\mathcal{U}$ is with the basis vectors $\{\mathbf{e}_i\}_{i=1}^m$. A large $\mu_i(\mathcal{U})$ suggests that $\mathcal{U}$ is highly correlated with the i-th basis vector $\mathbf{e}_i$, in that the projection of $\mathbf{e}_i$ onto $\mathcal{U}$ preserves much of its original length; a small value of $\mu_i(\mathcal{U})$ suggests that $\mathcal{U}$ is nearly orthogonal with $\mathbf{e}_i$, so a projection of $\mathbf{e}_i$ onto $\mathcal{U}$ loses most of its length. Figure 6.2 visualizes these two cases using the projection of three basis vectors on a two-dim. subspace $\mathcal{U}$. Note that the projection of the red vector onto $\mathcal{U}$ retains nearly unit length, so $\mathcal{U}$ has near-maximal coherence for this basis. On the other hand, the projection of the black vector onto $\mathcal{U}$ results in a sizable length reduction, so $\mathcal{U}$ has near-minimal coherence for this basis. Here, the overall coherence of $\mathcal{U}$, $\mu(\mathcal{U})$, is large due to the high coherence of the red basis vector.

In matrix completion literature, coherence is widely used to quantify the *recoverability* of a low-rank matrix $\mathbf{X}$. To see why, let $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ be the singular value decomposition (SVD) of $\mathbf{X}$. Consider two simple examples for $\mathbf{X}$. For the first example, set $\mathbf{U} = \mathbf{V} = \mathbf{e}_1$

and $\mathbf{D} = 1$, which results in maximal coherences for both the row and column spaces $\mathcal{U}$ and $\mathcal{V}$. The matrix $\mathbf{X}$ then consists of all zeroes, except for an entry of 1 in the first row and column. Clearly, there is no hope of recovering $\mathbf{X}$ from incomplete entries here, because one would need to observe nearly all entries to detect the lone non-zero entry. This shows that higher coherence for $\mathcal{U}$ or $\mathcal{V}$ leads to greater matrix "spikiness", so $\mathbf{X}$ *is more difficult to complete when its row or column space has high coherence*. For the second example, set $\mathbf{U} = (1/\sqrt{m_1})\mathbf{1}$ and $\mathbf{V} = (1/\sqrt{m_2})\mathbf{1}$, which results in minimal coherences for $\mathcal{U}$ and $\mathcal{V}$. $\mathbf{X}$ then becomes a constant matrix with entries $1/\sqrt{m_1 m_2}$, which can be completed from observing a single entry. In other words, $\mathbf{X}$ *is easier to complete when its row and column spaces have low coherence*. A more rigorous argument of this is found in [204, 202, 203], where it is shown that the matrix completion error bound via nuclear-norm minimization depends explicitly on the coherence term $\max\{\mu(\mathcal{U}), \mu(\mathcal{V})\}$.

6.3.2 The role of coherence in uncertainty quantification (UQ)

Here, the same notion of coherence arises in a different context – within the uncertainty quantification for the proposed model. We show this first for the unconditional model uncertainty (i.e., prior to observing any entries), then for the conditional uncertainty after observing noisy entries $\mathbf{Y}_\Omega$.

Consider first the case where no matrix entries have been observed. From Lemma 6 (a), $\text{vec}(\mathbf{X})$ follows the degenerate Gaussian distribution $\mathcal{N}\{\mathbf{0}, \sigma^2(\mathcal{P}_\mathcal{V} \otimes \mathcal{P}_\mathcal{U})\}$. The variance of the (i, j)-th entry in $\mathbf{X}$ can then be shown to be:

$$\text{Var}(X_{i,j}) = \sigma^2(\mathbf{e}_i \mathcal{P}_\mathcal{U} \mathbf{e}_i)(\mathbf{e}_j \mathcal{P}_\mathcal{V} \mathbf{e}_j) = \sigma^2 \mu_i(\mathcal{U})\mu_j(\mathcal{V}). \tag{6.14}$$

Hence, prior to observing data, the model uncertainty for entry $X_{i,j}$ is proportional to the product of coherences for the row and column spaces $\mathcal{U}$ and $\mathcal{V}$, with respect to the i-th and j-th basis vectors. Put another way, *the proposed model assigns greater variation to*

matrix entries with high subspace coherence in either its row or column index. This is quite appealing in view of the original role of coherence in matrix completion, where larger row (or column) coherences imply greater *"spikiness"* for entries; our framework accounts for this by assigning greater *model uncertainty* to such entries.

Consider next the case where noisy entries $\mathbf{Y}_\Omega$ have been observed. A more general notion of coherence is then required:

Definition 18 (Cross-coherence)**.** *Adopt the notation in Definition 17. The* cross-coherence *of subspace $\mathcal{U}$ with respect to the basis vectors $\mathbf{e}_i$ and $\mathbf{e}_{i'}$ is defined as $\nu_{i,i'}(\mathcal{U}) = \mathbf{e}_{i'}^T \mathcal{P}_\mathcal{U} \mathbf{e}_i$.*

In words, the cross-coherence $\nu_{i,i'}(\mathcal{U})$ quantifies how correlated the basis vectors $\mathbf{e}_i$ and $\mathbf{e}_{i'}$ are, *after* a projection onto $\mathcal{U}$. For example, in Figure 6.2, the pair of red / blue projected basis vectors have negative cross-coherence for $\mathcal{U}$, whereas the pair of blue / black projected vectors have positive cross-coherence. When $i = i'$, this cross-coherence reduces to the original coherence in Definition 17.

Define now the cross-coherence vector $\boldsymbol{\nu}_i(\mathcal{U}) = [\nu_{i,i_n}(\mathcal{U})]_{n=1}^N \in \mathbb{R}^N$, where again $\Omega = \{(i_n, j_n)\}_{n=1}^N$. From equation (6.7) in Lemma 6, the conditional variance of entry $X_{i,j}$ for an unobserved index $(i,j) \in \Omega^c$ becomes:

$$\text{Var}(X_{i,j}|\mathbf{Y}_\Omega) = \sigma^2 \mu_i(\mathcal{U})\mu_j(\mathcal{V}) - \sigma^2 \boldsymbol{\nu}_{i,j}^T \left[\mathbf{R}_N(\Omega) + \gamma^2 \mathbf{I}\right]^{-1} \boldsymbol{\nu}_{i,j}, \tag{6.15}$$

where $\boldsymbol{\nu}_{i,j} := \boldsymbol{\nu}_i(\mathcal{U}) \circ \boldsymbol{\nu}_j(\mathcal{V})$, and $\circ$ denotes the entry-wise (Hadamard) product. The expression in (6.15) also enjoys a nice interpretation. From a UQ perspective, the first term in (6.15), $\mu_i(\mathcal{U})\mu_j(\mathcal{V})$, is simply the unconditional uncertainty for entry $X_{i,j}$, *prior* to observing data. The second term, $\boldsymbol{\nu}_{i,j}^T[\mathbf{R}_N(\Omega) + \gamma^2\mathbf{I}]^{-1}\boldsymbol{\nu}_{i,j}$, can be viewed as the *reduction* in uncertainty, *after* observing the noisy entries $\mathbf{Y}_\Omega$. This uncertainty reduction is made possible by the correlation structure imposed on $\mathbf{X}$, via the SMG model. (6.15) also yields valuable insight in terms of subspace correlation. The first term $\mu_i(\mathcal{U})\mu_j(\mathcal{V})$ can be seen as the joint correlation between (a) row space $\mathcal{U}$ to row index i, and (b) column space $\mathcal{V}$ to

column index j, *prior* to any observations. The second term can be viewed as the portion of this correlation *explained* by observed indices Ω.

6.3.3 UQ, error monotonicity and error convergence

Using this link between coherence and uncertainty, we present two novel insights on expected error decay. The following theorem forms the basis for these insights:

Theorem 25 (Variance reduction)**.** *Suppose $\mathcal{U}$ and $\mathcal{V}$ are fixed. Let $\mathbf{Y}_\Omega$ contain the entries at $\Omega \subseteq [m_1] \times [m_2]$, and let $\mathbf{Y}_{\Omega \cup (i,j)}$ contain an additional observation at $(i,j) \in \Omega^c$. For any index $(k,l) \in [m_1] \times [m_2]$, the conditional variance of $X_{k,l}$ can be decomposed as*

$$\text{Var}(X_{k,l}|\mathbf{Y}_{\Omega\cup(i,j)}) = \text{Var}(X_{k,l}|\mathbf{Y}_\Omega) - \frac{\text{Cov}^2(X_{k,l}, X_{i,j}|\mathbf{Y}_\Omega)}{\text{Var}(X_{i,j}|\mathbf{Y}_\Omega) + \eta^2}, \tag{6.16}$$

where

$$\begin{aligned} &\text{Cov}(X_{i,j}, X_{k,l}|\mathbf{Y}_\Omega) = \\ &\qquad \sigma^2\{\nu_{i,k}(\mathcal{U})\nu_{j,l}(\mathcal{V}) - \boldsymbol{\nu}_{i,j}^T \left[\mathbf{R}_N(\Omega) + \gamma^2\mathbf{I}\right]^{-1} \boldsymbol{\nu}_{k,l}\}. \end{aligned} \tag{6.17}$$

Remark: This theorem shows, given observed indices Ω, the reduction in uncertainty (as measured by variance) for an unobserved entry $X_{k,l}$, after observing an additional entry at index (i,j). The last term in (6.16) quantifies this reduction, and can be interpreted as follows. For an unobserved index $(k,l) \notin \Omega \cup (i,j)$, this uncertainty reduction can be seen as a *signal-to-noise* ratio, the signal being the conditional squared-covariance between the "unobserved" entry $X_{k,l}$ and the "to-be-observed" entry $X_{i,j}$, and the noise being the conditional variance of the "to-be-observed" entry.

The first insight of *error monotonicity* follows immediately:

Corollary 4 (Error monotonicity; arbitrary sequential sampling)**.** *Suppose $\mathcal{U}$ and $\mathcal{V}$ are fixed. Let $[(i_n, j_n)]_{n=1}^{m_1 m_2} \subseteq [m_1] \times [m_2]$ be an arbitrary sampling scheme, where $(i_n, j_n) \neq$*

$(i_{n'}, j_{n'})$ *for* $n \neq n'$. *Let* $X^P_{k,l}$ *be the* (k,l)*-th entry of the conditional mean in* (6.7)*. Define*

$$\epsilon^2_N(k,l) := \mathbb{E}\left\{ \left(X_{k,l} - X^P_{k,l}\right)^2 \middle| \mathbf{Y}_{\Omega_{1:N}} \right\},\ (k,l) \in [m_1] \times [m_2]$$

as the expected squared-error for $X_{k,l}$ *after observing* $\mathbf{Y}_{\Omega_{1:N}}$. *Then* $\epsilon^2_{N+1}(k,l) \leq \epsilon^2_N(k,l)$ *for any* $N = 1, 2, \cdots$.

Remark: This corollary shows that, for any sequential sampling scheme and any index (k,l), the expected squared-error in estimating $X_{k,l}$ with the conditional mean $X^P_{k,l}$ is always monotonically decreasing as more samples are collected. This is intuitive, since one expects to gain more information on the unknown matrix $\mathbf{X}$ as more entries are observed. The fact that the proposed model quantifies this monotonicity property provides a reassuring check on our UQ approach.

The second insight connects expected error decay with the entry-wise correlations from the model:

Corollary 5 (Lower bound for error decay; arbitrary sequential sampling)**.** *Adopt the same notation in Corollary 4. For any* $N \geq 1$ *and* $(k,l) \notin \Omega_{1:N}$,

$$\epsilon^2_N(k,l) \geq \sigma^2 \mu_k(\mathcal{U}) \mu_l(\mathcal{V}) \cdot \left[\prod_{n=1}^{N} \left(1 - \frac{\text{Corr}^2(X_{i_n,j_n}, X_{k,l} | \mathbf{Y}_{\Omega_{1:(n-1)}})}{1+\gamma^2} \right) \right]. \tag{6.18}$$

where $\text{Corr}(X_{i,j}, X_{k,l}|\mathbf{Y}_\Omega)$ *is the correlation between entries* $X_{i,j}$ *and* $X_{k,l}$ *given observations* $\mathbf{Y}_\Omega$.

Remark: Corollary 5 shows the expected squared-error $\epsilon^2_N(k,l)$ is lower bounded by the coherence term $\sigma^2 \mu_k(\mathcal{U}) \mu_l(\mathcal{V})$, times a product of terms quantifying the correlation between the unobserved entry $X_{k,l}$ and the observed entries $\{X_{i_n,j_n}\}_{n=1}^N$. Note that a larger conditional correlation for $\text{Corr}^2(X_{i_n,j_n}, X_{k,l}|\mathbf{Y}_{\Omega_{n-1}})$ results in a smaller value for $1 - \text{Corr}^2(X_{i_n,j_n}, X_{k,l}|\mathbf{Y}_{\Omega_{n-1}})/(1+\gamma^2)$, which in turn yields a quicker error decay from (6.18).

This makes sense intuitively, because one expects an improved recovery of the unobserved entry $X_{k,l}$ when previously observed samples $\{X_{i_n,j_n}\}_{n=1}^N$ are highly correlated with $X_{k,l}$.

While such insights are valuable, it is difficult to use (6.16) or (6.18) as a optimization criterion for sampling. This is because, for *each* potential index (i,j) to sample, one would need to evaluate the error reduction term in (6.16) over *all* unobserved entries (k,l), which quickly becomes computationally infeasible. We introduce next an efficient information-theoretic sampling scheme which, using the so-called maximum entropy principle, achieves the desired properties from Corollary 5.

6.4 Maximum entropy sampling for matrix completion

With this model in hand, we now present a information-theoretic approach based on *entropy* for sampling (or designing) matrix entries for matrix completion. This sampling method consists of two stages: (a) an *initial design* strategy for preliminary learning on $\mathbf{X}$, and (b) a *sequential design* strategy to greedily maximize information gain. We first review the maximum entropy principle for noisy matrix completion, then present several novel insights on information-theoretic design for both initial and sequential sampling.

6.4.1 The maximum entropy sampling principle

The principle of maximum entropy sampling was first introduced in [211] and further developed in [227] for (statistical) experimental design of spatio-temporal models. In words, this principle states that, under regularity assumptions on an observation model with unknown parameters, *a sampling scheme which maximizes the entropy of observations also maximizes information gain on model parameters*. Here, this means the sampling scheme which maximizes information on the unknown matrix $\mathbf{X}$ is the same sampling scheme which maximizes the entropy of the observed entries $\mathbf{Y}_\Omega$. As we show below, the maximum entropy principle yields two advantages: (a) it reveals several novel insights on information-theoretic design for matrix completion, and (b) it allows for an efficient sam-

pling algorithm.

To present this formally, we first define some notation. Let (X, Y) be a pair of r.v.s with marginal densities $(f_X(x), f_Y(y))$ and joint density $f_{X,Y}(x, y)$. The *entropy* of X [228] is defined as $\mathrm{H}(X) = \mathbb{E}[-\log f_X(X)]$, with larger values indicating greater uncertainty for r.v. X. Similarly, the *joint entropy* of (X, Y) is defined as $\mathrm{H}(X, Y) = \mathbb{E}[-\log f_{X,Y}(X, Y)]$, and the *conditional entropy* of Y given X is defined as $\mathrm{H}(Y|X)$, the entropy of the conditional r.v. $Y|X$. The well-known chain rule (Theorem 2.2.1 in [228]) connects the joint entropy $\mathrm{H}(X, Y)$ with the conditional entropy $\mathrm{H}(Y|X)$:

$$\mathrm{H}(X, Y) = \mathrm{H}(X) + \mathrm{H}(Y|X). \tag{6.19}$$

We will use this identity below to derive the maximum entropy principle for matrix completion.

Consider now the noisy matrix completion problem. Here, the parameter-of-interest is the unknown low-rank matrix $\mathbf{X}$, the design scheme is the choice of sampled indices Ω, and the collected data are the observed entries $\mathbf{Y}_\Omega$. Applying the chain rule (6.19), we get the following decomposition:

$$\mathrm{H}(\mathbf{Y}_\Omega, \mathbf{X}) = \mathrm{H}(\mathbf{Y}_\Omega) + \mathrm{H}(\mathbf{X}|\mathbf{Y}_\Omega). \tag{6.20}$$

The first term $\mathrm{H}(\mathbf{Y}_\Omega, \mathbf{X})$ is the joint entropy of observations $\mathbf{Y}_\Omega$ and matrix $\mathbf{X}$, the middle term $\mathrm{H}(\mathbf{Y}_\Omega)$ is the entropy of observations $\mathbf{Y}_\Omega$ at entries Ω, and the last term $\mathrm{H}(\mathbf{X}|\mathbf{Y}_\Omega)$ is the conditional entropy of matrix $\mathbf{X}$ after observing $\mathbf{Y}_\Omega$. To *maximize* the information gained on the unknown matrix $\mathbf{X}$ from observing $\mathbf{Y}_\Omega$, we want to sample indices Ω which *minimize* the conditional entropy $\mathrm{H}(\mathbf{X}|\mathbf{Y}_\Omega)$.

We can now derive the maximum entropy principle for matrix completion. Let $\boldsymbol{\epsilon}_\Omega := (\epsilon_{i,j})_{(i,j)\in\Omega}$ be the vector of measurement errors. Applying the chain rule to the joint entropy

$\mathrm{H}(\mathbf{Y}_\Omega, \mathbf{X})$ in (6.20), we get:

$$\begin{aligned}
\mathrm{H}(\mathbf{Y}_\Omega, \mathbf{X}) &= \mathrm{H}(\mathbf{X}) + \mathrm{H}(\mathbf{Y}_\Omega|\mathbf{X}) && \text{(by (6.19))} \\
&= \mathrm{H}(\mathbf{X}) + \mathrm{H}(\mathbf{X}_\Omega + \boldsymbol{\epsilon}_\Omega|\mathbf{X}) \\
&= \mathrm{H}(\mathbf{X}) + \mathrm{H}(\boldsymbol{\epsilon}_\Omega|\mathbf{X}) && (\mathbf{X}_\Omega \text{ is fixed given } \mathbf{X}) \\
&= \mathrm{H}(\mathbf{X}) + \mathrm{H}(\boldsymbol{\epsilon}_\Omega). && (\boldsymbol{\epsilon}_\Omega \text{ indep. of } \mathbf{X})
\end{aligned}$$

Since the measurement noise in $\boldsymbol{\epsilon}_\Omega$ are i.i.d. Gaussian, its entropy $\mathrm{H}(\boldsymbol{\epsilon}_\Omega)$ does not depend on the choice of sampled indices Ω. Hence, the final quantity $\mathrm{H}(\mathbf{X}) + \mathrm{H}(\boldsymbol{\epsilon}_\Omega)$ above *does not depend on* Ω. It follows that the joint entropy $\mathrm{H}(\mathbf{Y}_\Omega, \mathbf{X})$ also does not depend on Ω, and by (6.20), the indices Ω which *minimize* $\mathrm{H}(\mathbf{X}|\mathbf{Y}_\Omega)$ also *maximize* $\mathrm{H}(\mathbf{Y}_\Omega)$. This yields the maximum entropy sampling principle for matrix completion – *a sampling scheme which maximizes the entropy of observations* $\mathbf{Y}_\Omega$ *also yields maximum information gain on* $\mathbf{X}$. This principle allows us to manipulate the simpler entropy term $\mathrm{H}(\mathbf{Y}_\Omega)$ as an efficient proxy for the desired entropy term $\mathrm{H}(\mathbf{X}|\mathbf{Y}_\Omega)$, the latter being more complicated and difficult to optimize in high-dimensions.

Consider now the observational entropy $\mathrm{H}(\mathbf{Y}_\Omega)$, which we abbreviate as $\mathrm{H}(\Omega_{1:N})$. For the proposed model on $\mathbf{X}$, the following lemma gives a closed-form expression for $\mathrm{H}(\mathbf{Y}_\Omega)$:

Lemma 8 (Observational entropy)**.** *For fixed* $\mathscr{P}_{\mathscr{U}}$ *and* $\mathscr{P}_{\mathscr{V}}$*,*

$$\mathrm{H}(\Omega_{1:N}) := \mathrm{H}(\mathbf{Y}_\Omega) = \det\{\sigma^2 \mathbf{R}_N(\Omega_{1:N}) + \eta^2 \mathbf{I}\}, \tag{6.21}$$

where $\mathbf{R}_N(\Omega)$ *is the covariance matrix defined in* (6.6)*.*

The index set maximizing this entropy is then defined as:

Definition 19 (Maximum entropy index set)**.** *For fixed* $\mathscr{P}_{\mathscr{U}}$ *and* $\mathscr{P}_{\mathscr{V}}$*, the* maximum entropy

index set $\Omega^*_{1:N}$ *is defined as*

$$\Omega^*_{1:N} := \underset{\Omega_{1:N}\in([m_1]\times[m_2])^N}{\text{Argmax}} \mathrm{H}(\Omega_{1:N}). \tag{6.22}$$

Remark: By maximizing $\mathrm{H}(\Omega_{1:N})$, the maximum entropy index set minimizes the conditional entropy term $\mathrm{H}(\mathbf{X}|\mathbf{Y}_{\Omega_{1:N}})$ via the maximum entropy principle. Sampling at these indices should then maximize information on $\mathbf{X}$, and yield improved completion performance to uniform sampling. One way to quantify the connection between $\mathrm{H}(\mathbf{X}|\mathbf{Y}_{\Omega_{1:N}})$ and completion error is via the lower bound (Eq. 27 in [229]):

$$\mathbb{E}[\|\mathbf{X}-\tilde{\mathbf{X}}\|_F^2|\mathbf{Y}_{\Omega_{1:N}}] \geq \frac{1}{2\pi e}\exp\left\{2\mathrm{H}(\mathbf{X}|\mathbf{Y}_{\Omega_{1:N}})\right\}. \tag{6.23}$$

This bound shows that by maximizing information gain on $\mathbf{X}$ (i.e., minimizing $\mathrm{H}(\mathbf{X}|\mathbf{Y}_{\Omega_{1:N}})$), one can minimize the expected completion error $\mathbb{E}[\|\mathbf{X}-\tilde{\mathbf{X}}\|_F^2|\mathbf{Y}_{\Omega_{1:N}}]$ under the proposed model on $\mathbf{X}$. The advantage in using an entropy-based sampling criterion is that it allows us to work with the simpler observation entropy $\mathrm{H}(\Omega_{1:N})$, whereas minimizing the error term $\mathbb{E}[\|\mathbf{X}-\tilde{\mathbf{X}}\|_F^2|\mathbf{Y}_{\Omega_{1:N}}]$ directly is more cumbersome. We show below several novel properties of maximum entropy sampling for initial and sequential learning on $\mathbf{X}$.

6.4.2 Initial sampling: Latin square design

Consider first the initial sampling problem. For simplicity, assume $m_1 = m_2 = m$ (this will be generalized later), with total initial samples $N = m$. The following lemma shows that a certain balance property is desirable for initial sampling:

Proposition 6 (Lower bound on observation entropy)**.** *For fixed $\mathscr{P}_{\mathscr{U}}$ and $\mathscr{P}_{\mathscr{V}}$, we have*

$$\begin{aligned}\mathrm{H}^{1/N}(\Omega_{1:N}) \geq \min_{n=1,\cdots,N}\Bigg[&\sigma^2\mu_{i_n}(\mathscr{U})\mu_{j_n}(\mathscr{V})+\eta^2- \\ &\frac{\sigma^2(N-1)}{2}\left\{\max_{n':n'\neq n}\nu^2_{i_n,i_{n'}}(\mathscr{U})+\max_{n':n'\neq n}\nu^2_{j_n,j_{n'}}(\mathscr{V})\right\}\Bigg].\end{aligned} \tag{6.24}$$

Remark: Proposition 6 can be interpreted as follows. Take first the right-hand side of (6.24), which provides a lower bound for the entropy term $\mathrm{H}^{1/N}(\Omega_{1:N})$ for fixed $\mathscr{P}_{\mathscr{U}}$ and $\mathscr{P}_{\mathscr{V}}$. Given no prior knowledge on subspaces $\mathscr{U}$ and $\mathscr{V}$, it makes sense to assume $\mathscr{P}_{\mathscr{U}}$ and $\mathscr{P}_{\mathscr{V}}$ are uniformly distributed on the Grassmann manifolds $\mathscr{G}_{R,m_1-R}$ and $\mathscr{G}_{R,m_2-R}$, i.e.:

$$[\mathscr{P}_{\mathscr{U}}] \propto 1, \quad [\mathscr{P}_{\mathscr{V}}] \propto 1. \tag{6.25}$$

Under (6.25), the expected left-hand term in (6.24), $\mathbb{E}_{\mathscr{P}_{\mathscr{U}},\mathscr{P}_{\mathscr{V}}}\{\sigma^2\mu_{i_n}(\mathscr{U})\mu_{j_n}(\mathscr{V})\}$, is constant for any index (i_n, j_n), since the uniform distributions on $\mathscr{G}_{R,m_1-R}$ and $\mathscr{G}_{R,m_2-R}$ are rotation invariant. Moreover, under (6.25), the right-hand term in (6.24) becomes:

$$\frac{\sigma^2(N-1)}{2}\left\{\max_{n\neq n'}(\mathbf{e}_{i_n}^T\mathbf{e}_{i_{n'}})^2 + \max_{n\neq n'}(\mathbf{e}_{j_n}^T\mathbf{e}_{j_{n'}})^2\right\}. \tag{6.26}$$

Next, consider the minimization of (6.26) over all possible index sets $\Omega_{1:N} = \{(i_n, j_n)\}_{n=1}^N$, which serves as a proxy for the maximization of $\mathrm{H}(\Omega_{1:N})$ via the lower bound in (6.24). This amounts to jointly minimizing the two terms in (6.26), i.e.:

$$\min_{\{i_n\}_{n=1}^N\in[m]^N}\max_{n\neq n'}(\mathbf{e}_{i_n}^T\mathbf{e}_{i_{n'}})^2 \text{ and } \min_{\{j_n\}_{n=1}^N\in[m]^N}\max_{n\neq n'}(\mathbf{e}_{j_n}^T\mathbf{e}_{j_{n'}})^2. \tag{6.27}$$

Clearly, if $i_n = i_{n'}$ for some $n \neq n'$ (i.e., the same row is sampled twice), then the first term in (6.27) attains the maximum possible value of 1. Likewise, if $j_n = j_{n'}$ for some $n \neq n'$ (i.e., the same column is sampled twice), then the second term in (6.27) attains the maximum possible value of 1 as well. Both scenarios are undesirable, because the goal is to jointly *minimize* the two objectives in (6.27). Hence, with no prior knowledge on the subspaces of $\mathbf{X}$, an initial sampling scheme satisfying maximum entropy should be *balanced*, in that *no row or column is sampled more than once in* $\mathbf{X}$.

This desired balance of $\Omega_{1:N}^*$ has an illuminating connection to existing work in matrix

$$\begin{pmatrix} \textcircled{1} & 3 & 2 \\ 3 & 2 & \textcircled{1} \\ 2 & \textcircled{1} & 3 \end{pmatrix} \qquad \begin{pmatrix} \textcircled{1} & 2 & 3 & 4 \\ 3 & 4 & \textcircled{1} & 2 \\ 4 & 3 & 2 & \textcircled{1} \\ 2 & \textcircled{1} & 4 & 3 \end{pmatrix}$$

Figure 6.3: A 3×3 and a 4×4 Latin square. A balanced sampling scheme is obtained by sampling the entries with 1 (circled).

completion, specifically the *injectivity property* introduced in [203]. This property arises when the sampling operator $\mathscr{R}_\Omega$ (which maps $\mathbf{X}$ to $\mathbf{X}_\Omega$) is injective over a large class of low-rank matrices. In [203], the authors showed that this property is necessary to ensure a unique solution for the nuclear-norm formulation in (6.12). One consequence of this injectivity property is that the sampling operator must observe (at least) one entry from every row and column, which is precisely the balance property of $\Omega^*_{1:N}$ derived earlier. In this sense, sampling an entry in every row and column not only improves theoretical guarantees for completion, but also yields greater information gain on $\mathbf{X}$. More importantly, instead of achieving such a property via uniform random sampling (which is the typical approach in the literature, and requires $N = \mathcal{O}(m \log m)$ samples), *we instead impose this balance directly within the initial sampling scheme* (reducing the required samples to $N = \mathcal{O}(m)$).

This balance property of $\Omega^*_{1:N}$ can be nicely represented as a *Latin square*, which has been used extensively for designing error-correcting codes [230, 231] and in experimental design [232]. An $m \times m$ Latin square is an arrangement of the elements $[m] = \{1, \cdots, m\}$ in an $m \times m$ square, so that each row and column contains every entry of $[m]$ exactly once. Figure 6.3 shows an example of a 3×3 and a 4×4 Latin square. Consider now an initial sampling scheme obtained by sampling the entries of a Latin square at a given value (say, '1'). From Figure 6.3, the resulting design has exactly one sample in every row and column, which is as desired. This can easily extended for generating initial designs for non-square $\mathbf{X}$ (see Section 6.5.2).

Of course, there are multiple ways to select a balanced initial sampling scheme. For example, one can sample the entries labeled '2' in the Latin squares in Figure 6.3, and

end up with a different balanced design. A natural question to ask is whether all balanced designs yield the same performance on average. From an information-theoretic perspective, the following theorem answers this in the affirmative:

Proposition 7 (Equivalence of balanced designs)**.** *Suppose* $\mathscr{P}_{\mathscr{U}}, \mathscr{P}_{\mathscr{V}} \overset{i.i.d.}{\sim} U(\mathscr{G}_{R,m-R})$*. For any two balanced designs* Ω_1 *and* Ω_2*, with* $|\Omega_1| = |\Omega_2| = m$*, we have* $\mathbb{E}_{\mathscr{P}_{\mathscr{U}},\mathscr{P}_{\mathscr{V}}}\{\mathrm{H}(\Omega_1)\} = \mathbb{E}_{\mathscr{P}_{\mathscr{U}},\mathscr{P}_{\mathscr{V}}}\{\mathrm{H}(\Omega_2)\}$.

In other words, under the belief that all row and column spaces are equally likely, all balanced sampling schemes yield the same expected information gain on $\mathbf{X}$. To take advantage of this, we will employ an initial sampling algorithm using random Latin squares; more on this in Section 6.5.2.

6.4.3 Sequential design: Insights from coherence

Consider now the setting where the noisy entries $\mathbf{Y}_\Omega$ have been observed at indices $\Omega_{1:N}$, and suppose informed estimates can be obtained on the subspaces $\mathscr{U}$ and $\mathscr{V}$ from such observations (more on this in Section 6.5.2). Fixing the observed indices $\Omega_{1:N}$, the sequential problem of sampling the next index $(i,j) \notin \Omega_{1:N}$ maximizing observational entropy $\mathrm{H}(\Omega_{1:N} \cup (i,j))$ can be formulated as follows:

Lemma 9. *For fixed* $\mathscr{P}_{\mathscr{U}}$*,* $\mathscr{P}_{\mathscr{V}}$ *and observed indices* $\Omega_{1:N}$*,*

$$\begin{aligned}
&\underset{(i,j)\in\Omega_{1:N}^c}{\text{Argmax}}\ \mathrm{H}(\Omega_{1:N} \cup (i,j)) \\
&\quad = \underset{(i,j)\in\Omega_{1:N}^c}{\text{Argmax}}\ \left\{\mu_i(\mathscr{U})\mu_j(\mathscr{V}) - \boldsymbol{\nu}_{i,j}^T[\mathbf{R}_N(\Omega_{1:N}) + \gamma^2\mathbf{I}]^{-1}\boldsymbol{\nu}_{i,j}\right\} \\
&\quad =: \underset{(i,j)\in\Omega_{1:N}^c}{\text{Argmax}}\ \mathrm{H}((i,j)|\Omega_{1:N}),
\end{aligned} \tag{6.28}$$

where $\boldsymbol{\nu}_{i,j} = \boldsymbol{\nu}_i(\mathscr{U}) \circ \boldsymbol{\nu}_j(\mathscr{V})$.

In other words, given observations at $\Omega_{1:N}$, the next index $(i,j) \in \Omega_{1:N}^c$ maximizing information gain on $\mathbf{X}$ can be obtained via the maximization problem on the right side of (6.28).

This information-greedy sampling approach has been employed in a variety of fields, e.g., compressive sensing [84].

Lemma 9 is appealing from a computational perspective, because it provides an easy-to-evaluate criterion for greedily maximizing information gain on $\mathbf{X}$. Note that, for each unobserved index $(i, j) \in \Omega^c$, the left-hand criterion $\mathrm{H}(\Omega_{1:N} \cup (i, j))$ requires $\mathcal{O}(N^3)$ work to evaluate, so a total work of $\mathcal{O}(|\Omega^c|N^3)$ is needed for optimizing this criterion. On the other hand, the right-hand criterion $\mathrm{H}((i, j)|\Omega_{1:N})$ can be evaluated in $\mathcal{O}(N^2)$ work (assuming $[\mathbf{R}_N(\Omega_{1:N}) + \gamma^2\mathbf{I}]^{-1}$ is computed beforehand with $\mathcal{O}(N^3)$ work), which reduces total optimization work to $\mathcal{O}(N^3 + |\Omega^c|N^2)$. This computation reduction becomes valuable when m_1 and m_2 grow large (i.e., in high-dimensions). We will provide an efficient implementation of this sequential optimization in Section 6.5.2.

Lemma 9 also reveals a curious link between this information-greedy sequential sampling and the earlier discussion on UQ, coherence, and error convergence in Section 6.3. The clue lies in the reformulated right-hand criterion in (6.28) and the conditional variance in (6.15), which are identical up to constants. This reveals three insights. First, the sequential criterion in (6.28) can be seen as the information gained from entry $X_{i,j}$ *prior* to any observations (first term), minus the information gained on $X_{i,j}$ *after* observing the indices in Ω (second term). The optimization in (6.28) then samples the entry with the largest *residual* information unexplained by Ω. Second, sampling the entry with *maximum information gain* is equivalent to sampling the entry with *maximum uncertainty* (conditional on observations in Ω), or sampling the entry with the *greatest unexplained "spikiness"* (as measured by coherence). Third, by sampling the row and column with greatest unexplained coherence, we jointly maximize the signal-to-noise ratios in (6.16) for unobserved entries with large variances, which then improves error convergence by Corollary 5.

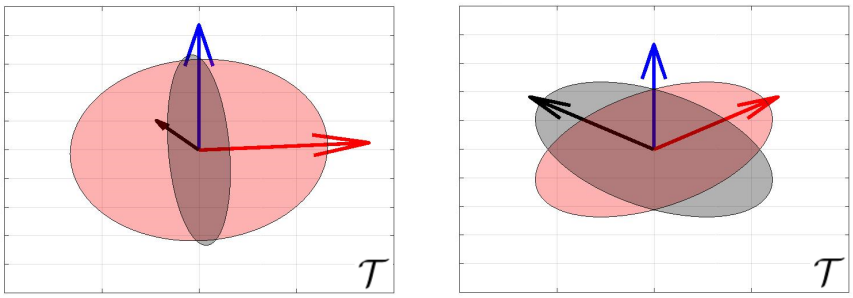

Figure 6.4: Two visualizations of $\mathrm{H}(\Omega_{1:N})$. The red ellipse is the covariance matrix for the red and blue entries (projected onto $\mathcal{T}$); the black ellipse for the black and blue entries.

6.4.4 Coherence and sampling: A geometric view

This maximum entropy sampling approach also yields a nice geometric interpretation. To see this, recall the form of the observational entropy $\mathrm{H}(\Omega_{1:N})$:

$$\mathrm{H}(\Omega_{1:N}) = \det\{\sigma^2 \mathbf{R}_N(\Omega_{1:N}) + \eta^2 \mathbf{I}\}, \tag{6.29}$$

which we wish to maximize. Rewrite $\mathbf{R}_N(\Omega_{1:N})$ as:

$$\mathbf{R}_N(\Omega_{1:N}) = [\langle \mathscr{P}_{\mathscr{U}} \mathbf{M}_n \mathscr{P}_{\mathscr{V}}, \mathscr{P}_{\mathscr{U}} \mathbf{M}_{n'} \mathscr{P}_{\mathscr{V}} \rangle_F]_{n,n'=1}^N, \tag{6.30}$$

where $\mathbf{M}_n := \mathbf{e}_{i_n} \mathbf{e}_{j_n}^T$ and $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product. Here, $\mathbf{M}_n$ can be seen as a rank-1 binary measurement mask [233] which returns the entrywise measurement $X_{i_n,j_n} = \langle \mathbf{M}_n, \mathbf{X} \rangle_F$. From (6.30), the (n, n')-th entry in $\mathbf{R}_N(\Omega_{1:N})$ can be viewed as the inner product between the binary masks $\mathbf{M}_n$ and $\mathbf{M}_{n'}$, after projection onto the subspaces of $\mathbf{X}$. Finally, ignoring the noise term $\eta^2 \mathbf{I}$ in (6.29), the entropy $\mathrm{H}(\Omega_{1:N})$ can then be interpreted as the *ellipsoid volume* of the covariance matrix for the N masks (for observed entries), after a projection onto the subspaces of $\mathbf{X}$.

Figure 6.4 visualizes two examples of $\mathrm{H}(\Omega_{1:N})$ for three entries to sample in $\mathbf{X}$. Here, the solid vectors (black, blue and red) represent the binary masks $\mathbf{M}_n$ for these sampled entries, projected onto $\mathcal{T}$ (see (6.2)). The red ellipse is the covariance matrix for the red

and blue sampled entries, and the black ellipse the covariance matrix for the black and blue sampled entries. Consider first the right-hand plot. Here, the red and black ellipses have the same volume, which suggests that (a) sampling the red and blue entries, and (b) sampling the black and blue entries yield the same information gain on $\mathbf{X}$. Consider next the left-hand plot. Here, the red ellipse has much larger volume than the black ellipse, which suggests that sampling scheme (a) yields greater information gain on $\mathbf{X}$.

This interpretation nicely visualizes two desired sampling properties derived earlier. First, *rows and columns with high coherences should be prioritized in sampling.* In Figure 6.4, this means choosing vectors with the greatest lengths after projection onto $\mathcal{T}$, which increases ellipsoid volume and thereby information gain on $\mathbf{X}$. Second, *a new sample should maximize the information left unexplained by observed entries in* Ω. This is akin to choosing vectors as orthogonal as possible in Figure 6.4, which again increases ellipsoid volume and maximizes information gain.

6.5 UQ and sampling algorithms for matrix completion

We now combine the insights from previous sections into a practical, information-theoretic matrix sampling algorithm using UQ. We first outline a posterior sampling algorithm, `gibbs.mc`, which makes use of manifold sampling methods to quantify uncertainty on $\mathbf{X}$ via its subspaces, then present an information-theoretic design scheme, `MaxEnt`, which employs this UQ to guide the active sampling algorithm.

6.5.1 `gibbs.mc`: A posterior sampling algorithm for UQ

We first present a posterior sampling algorithm for quantifying uncertainty on $\mathbf{X}$. For noisy matrix completion, posterior sampling refers to sampling from the so-called posterior distribution $[\mathbf{X}|\mathbf{Y}_\Omega]$, which encodes information learned on the unknown matrix $\mathbf{X}$ given observed noisy entries $\mathbf{Y}_\Omega$. Sampling from this distribution provides insight on not only likely values for unobserved entries, but a measure of uncertainty (UQ) for such entries

as well. Note that this posterior sampling algorithm is different from the matrix sampling algorithm introduced later: the former provides uncertainty on $\mathbf{X}$ given observed entries $\mathbf{Y}_\Omega$, while the latter is used to guide the data collection procedure at unobserved entries.

For efficient posterior sampling, we require a slight parametrization of $\mathbf{X}$ via its SVD $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$. Define first the *Stiefel manifold* $\mathcal{V}_{R,m}$, the space of $m \times R$ matrices with orthonormal columns (an R-*frame* in $\mathbb{R}^m$). By the SVD, the matrix of left and right singular vectors, $\mathbf{U}$ and $\mathbf{V}$, must lie on the Stiefel manifolds $\mathcal{V}_{R,m_1}$ and $\mathcal{V}_{R,m_2}$, respectively. Note that the span of an R-frame from the Stiefel manifold $\mathcal{V}_{R,m}$ corresponds to a unique R-plane from the Grassmann manifold $\mathcal{G}_{R,m-R}$, but an R-plane from $\mathcal{G}_{R,m-R}$ corresponds to infinitely many R-frames from $\mathcal{V}_{R,m}$.

For the proposed model in Table 6.1, we can then apply random matrix theory [234] to show that: (a) $\mathbf{U}$ and $\mathbf{V}$ are independently and uniformly distributed on the Stiefel manifolds $\mathcal{V}_{R,m_1}$ and $\mathcal{V}_{R,m_2}$, and (b) $\mathbf{D} = \text{diag}(\{d_k\}_{k=1}^R)$ follows the so-called *Quadrant Law* (QL; [234]), which has connections to the limiting spectral distribution of random matrices [235]. The uniform distributions on $\mathcal{V}_{R,m_1}$ and $\mathcal{V}_{R,m_2}$ can be seen more generally as the *von Mises-Fisher* (MF) distributions [224] $MF(m_1, R, \mathbf{0})$ and $MF(m_2, R, \mathbf{0})$, where a random matrix $\mathbf{W} \sim MF(m, R, \mathbf{F})$ has density [236]:

$$[\mathbf{W}|R, \mathbf{F}] = \left[{}_0F_1\left(;\frac{m}{2};\frac{\mathbf{F}^T\mathbf{F}}{4}\right)\right]^{-1} \text{etr}(\mathbf{F}^T\mathbf{W}),\ \mathbf{W} \in \mathcal{V}_{R,m}, \tag{6.31}$$

and ${}_0F_1(;\cdot;\cdot)$ is the hypergeometric function. The singular values $\mathbf{D}$ follow $QL(\mathbf{0}, \sigma^2)$, where $QL(\boldsymbol{\mu}, \delta^2)$ is the quadrant law with density:

$$[\mathbf{D}|\boldsymbol{\mu}, \delta^2] = \frac{\exp\left\{-\frac{1}{2\delta^2}\sum_{k=1}^R (d_k - \mu_k)^2\right\}}{Z_R(2\pi\delta^2)^{R/2}} \prod_{k,l=1;k<l}^R |d_k^2 - d_l^2|, \tag{6.32}$$

and Z_R is a normalization constant depending on R. Both QL and MF can be efficiently sampled via the Metropolis-Hastings algorithm [237, 238] and state-of-the-art manifold

Algorithm 11 `gibbs.mc`: Gibbs sampler for fixed rank R

Require: Observations $\mathbf{Y}_\Omega$, rank R, prior parameters $\alpha_{\eta^2}, \beta_{\eta^2}, \alpha_{\sigma^2}, \beta_{\sigma^2}$

- *Initialization*: Complete $\mathbf{X}_0$ from $\mathbf{Y}_\Omega$ via nuclear-norm minim. [202]. Initialize $[\mathbf{U}_0, \mathbf{D}_0, \mathbf{V}_0] \leftarrow \text{svd}(\mathbf{X}_0)$, η_0^2 and σ_0^2.
- *Gibbs sampler*: **For** $t = 1, \ldots, T$: ▷ T - total samples
  - $\mathbf{X}_t \leftarrow \mathbf{U}_{t-1}\mathbf{D}_{t-1}\mathbf{V}_{t-1}^T$.
  - Impute missing entries $\mathbf{Y}_{\Omega^c}$ by sampling from $[\mathbf{Y}_{\Omega^c}|\mathbf{Y}_\Omega] \sim \mathcal{N}(\mathbf{X}_{\Omega^c}^P, \mathbf{\Sigma}_{\Omega^c}^P + \eta^2\mathbf{I})$.
  - Sample $\mathbf{U}_t \sim MF(m_1, R, \mathbf{Y}\mathbf{V}_{t-1}\mathbf{D}_{t-1}/\eta_{t-1}^2)$.
  - Sample $\mathbf{V}_t \sim MF(m_2, R, \mathbf{Y}^T\mathbf{U}_t\mathbf{D}_{t-1}/\eta_{t-1}^2)$.
  - Sample $\mathbf{D}_t \sim QL(\boldsymbol{\mu}, \delta^2)$ using Metropolis-Hastings, where $\boldsymbol{\mu} = [\sigma_{t-1}^2\mathbf{u}_{k,t}^T\mathbf{Y}\mathbf{v}_{k,t}/(\eta_{t-1}^2 + \sigma_{t-1}^2)]_{k=1}^R$ and $\delta^2 = \eta_{t-1}^2\sigma_{t-1}^2/(\eta_{t-1}^2 + \sigma_{t-1}^2)$.
  - Sample $\sigma_t^2 \sim IG(\alpha_{\sigma^2} + R/2, \beta_{\sigma^2} + \text{tr}(\mathbf{D}_t^2)/2)$ and $\eta_t^2 \sim IG(\alpha_{\eta^2} + m_1m_2/2, \beta_{\eta^2} + \|\mathbf{Y} - \mathbf{X}_t\|_F^2/2)$.
- Return posterior samples $\Theta^{(R)} = \{(\mathbf{X}_t, \mathbf{U}_t, \mathbf{V}_t)\}_{t=1}^T$.

sampling methods [236], respectively.

With this in hand, we present an efficient UQ algorithm `gibbs.mc` for sampling the posterior distribution $[\mathbf{X}|\mathbf{Y}_\Omega]$ for *fixed* rank R, which makes use of an iterative, closed-form sampling method called *Gibbs sampling* [239]. We describe this sampler in several steps. First, conditional on $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{D}$, one can view the unobserved entries $\mathbf{Y}_{\Omega^c}$ as a missing data problem [240], and impute these missing entries using the distribution in Lemma 6 (b). Next, conditional on the imputed matrix $\mathbf{Y}$, the full conditional distributions for singular vectors, $[\mathbf{U}|\mathbf{V}, \mathbf{D}, \mathbf{Y}]$ and $[\mathbf{V}|\mathbf{U}, \mathbf{D}, \mathbf{Y}]$, can be then be sampled from the MF distributions. Lastly, the full conditional distribution for singular values, $[\mathbf{D}|\mathbf{U}, \mathbf{V}, \mathbf{Y}]$, can be sampled from the quadrant law. By iteratively sampling (a) the conditional uncertainty in unobserved entries, (b) the row and column spaces of $\mathbf{X}$, and (c) its singular values, `gibbs.mc` can quantify the full uncertainty in $\mathbf{X}$ given observations $\mathbf{Y}_\Omega$ and rank R. Algorithm 11 provides the detailed steps for `gibbs.mc`, which has a running time of $\mathcal{O}\{(m_1 \vee m_2)R^3 + N^3\}$ for each iteration. Technical derivations of this sampler and its running time are provided in Appendix F.10.

This framework can be extended to quantify the uncertainty of matrix rank R, which

is typically unknown in practice. Let $\Theta^{(r)}$ denote the model parameters for fixed rank r, and suppose the posterior samples $\{\Theta_t^{(r)}\}_{t=1}^T$ have been generated from `gibbs.mc` for $r = 1, \cdots, m_1 \wedge m_2$. The posterior distribution of R given $\mathbf{Y}_\Omega$ can be written as:

$$\begin{aligned}[R|\mathbf{Y}_\Omega] &= \int [R|\Theta^{(r)}, \mathbf{Y}_\Omega]\ d[\Theta^{(r)}|\mathbf{Y}_\Omega] \\ &\propto \int [\mathbf{Y}_\Omega|\Theta^{(r)}, R][\Theta^{(r)}|R][R]\ d[\Theta^{(r)}|\mathbf{Y}_\Omega].\end{aligned} \tag{6.33}$$

The posterior probabilities on R can be approximated via:

$$\pi_r^P := \mathbb{P}(R = r|\mathbf{Y}_\Omega) \approx \frac{\sum_{t=1}^T f(\mathbf{Y}_\Omega|\Theta_t^{(r)})p(\Theta_t^{(r)})\pi_r}{\sum_{r=1}^{m_1\wedge m_2}\sum_{t=1}^T f(\mathbf{Y}_\Omega|\Theta_t^{(r)})p(\Theta_t^{(r)})\pi_r}, \tag{6.34}$$

where $f(\mathbf{Y}_\Omega|\Theta_t^{(r)})$ is the Gaussian density for $\mathbf{Y}_\Omega$ given rank r and posterior sample $\Theta_t^{(r)}$ (see (6.3)), and $p(\Theta_t^{(r)})$ is the prior density of $\Theta_t^{(r)}$ given rank r (see (6.9), (6.31) and (6.32)).

These probabilities can then be used to provide inference and UQ on $\mathbf{X}$ with unknown rank. Using the posterior samples for each rank r, the posterior mean of $\mathbf{X}$ can be estimated by:

$$\mathbb{E}(\mathbf{X}|\mathbf{Y}_\Omega) = \mathbb{E}[\mathbb{E}(\mathbf{X}|R, \mathbf{Y}_\Omega)|\mathbf{Y}_\Omega] \approx \sum_{r=1}^{m_1\wedge m_2} \frac{\pi_r^P}{T}\sum_{t=1}^T \mathbf{X}_t^{(r)}. \tag{6.35}$$

Similarly, with unknown rank, one can perform UQ for an unobserved entry $X_{i,j}$, $(i,j) \in \Omega^c$ by iterating the two steps: (a) sample a potential rank R' from the posterior probabilities $\{\pi_r^P\}_{r=1}^{m_1\wedge m_2}$, then (b) select the (i,j)-th entry for a random matrix from the posterior samples $\{\mathbf{X}_t^{(R')}\}_{t=1}^T$. This yields a sample chain for the posterior distribution $[X_{i,j}|\mathbf{Y}_\Omega]$, from which one can then compute point estimates and confidence intervals quantifying the uncertainty of $X_{i,j}$.

6.5.2 `MaxEnt`: A maximum entropy active sampling algorithm

Next, we summarize the insights from Section 6.4 into an information-theoretic sampling algorithm called `MaxEnt` (see Algorithm 12). For initial sampling, recall that a balanced

Algorithm 12 `MaxEnt`: Maximum entropy matrix sampling

Require: Total samples $N_{\max} \geq m_1 \vee m_2$, $m_1 \geq m_2$

- *Initial* ($N_{ini} = m_1 \vee m_2$ samples):
  - Stack $\lfloor m_1/m_2 \rfloor$ random $m_2 \times m_2$ Latin squares to form an $(m_2 \lfloor m_1/m_2 \rfloor) \times m_2$ rectangle.
  - Set Ω as the entries labeled '1' from this rectangle. If $m_2 \nmid m_1$, add a random sample in each of the remaining $m_1 - \lfloor m_1/m_2 \rfloor m_2$ rows.
- *Sequential*: **For** $n = N_{ini} + 1, \ldots, N_{max} = N_{ini} + N_{seq}$:
  - Run `gibbs.mc` for $r = 1, \cdots, m_1 \wedge m_2$.

    [Obtain $\hat{\mathbf{X}}$ from $\mathbf{Y}_\Omega$ via nuclear-norm minimization. Estimate subspaces $(\hat{\mathcal{U}}, \hat{\mathcal{V}})$ from svd($\hat{\mathbf{X}}$).]
  - Compute the next index (i_n, j_n) from (6.36).

    [Compute the next index (i_n, j_n) from (6.28), with subspaces $(\mathcal{U}, \mathcal{V})$ estimated by $(\widehat{\mathcal{U}}, \widehat{\mathcal{V}})$.]
  - Update $\Omega \leftarrow \Omega \cup (i_n, j_n)$.
- Complete $\hat{\mathbf{X}}$ from $\mathbf{Y}_\Omega$ via nuclear-norm minimization.
- Return $\hat{\mathbf{X}}$.

design on $\mathbf{X}$ – one entry from each row and column – is desired. Assuming $m_1 \geq m_2$, we guarantee this balance property in `MaxEnt` by (a) generating $\lfloor m_1/m_2 \rfloor$ random Latin squares of size $m_2 \times m_2$ (see [241]), (b) vertically stacking these squares to form an $(m_2 \lfloor m_1/m_2 \rfloor) \times m_2$ rectangle, and (c) sampling the entries labeled '1' from this rectangle. By randomly allocating one sample in the remaining $m_1 - \lfloor m_1/m_2 \rfloor m_2$ rows of $\mathbf{X}$, this ensures at least one observation in each row and column for the initial N_{ini} samples.

Having observed the initial sample $\mathbf{Y}_\Omega$, the row and column spaces $\mathcal{U}$ and $\mathcal{V}$ can then be learned via the posterior subspace samples $\{\mathbf{U}_t^{(r)}, \mathbf{V}_t^{(r)}\}_{t=1}^T$ from `gibbs.mc`. Using this information, we then sample the unobserved matrix entry yielding the greatest expected posterior information gain on $\mathbf{X}$:

$$\text{Argmax}_{(i,j) \in \Omega^c} \left\{ \sum_{r=1}^{m_1 \wedge m_2} \frac{\pi_r^P}{T} \sum_{t=1}^{T} \mathrm{H}_t^{(r)}((i,j) | \Omega_{1:N}) \right\}, \tag{6.36}$$

where $\mathrm{H}_t^{(r)}((i,j)|\Omega_{1:N})$ is the sequential entropy criterion in (6.28) with fixed rank r and

subspace sample $(\mathbf{U}_t^{(r)}, \mathbf{V}_t^{(r)})$. These two steps are then repeated until a desired error is achieved on $\mathbf{X}$. From a machine learning perspective, this procedure can be viewed as an *learning active learning* method for matrix completion – we first learn key hyperparameters on the subspaces of $\mathbf{X}$ via the UQ algorithm `gibbs.mc`, then employ this learning to guide active learning on $\mathbf{X}$.

While the above approach offers closed-form updates for both UQ and sampling, it can be computationally intensive when the dimensions of $\mathbf{X}$ grow large. To this end, we found several computational speed-ups to be effective in high-dimensions. First, given the inherent connection between the MAP estimator $\tilde{\mathbf{X}}$ and the nuclear-norm estimator $\hat{\mathbf{X}}$ (Lemma 7), state-of-the-art algorithms for the latter (e.g., [203, 204]) can be used to efficiently obtain a point estimate of $\mathbf{X}$ for our model. An SVD of this point estimate yields estimates for subspaces $\mathcal{U}$ and $\mathcal{V}$, which can then be incorporated for sequential sampling. From a Bayesian perspective, one can view this as an *empirical Bayes* approach [242] for learning the active sampling procedure. This shortcut is bracketed in Algorithm 12. Second, for m_1 and m_2 large, the exhaustive search for the next index (either (6.28) or (6.36)) can be time-consuming. One way to reduce computation is to screen out indices which are likely poor entries to sample, then perform the search over a much smaller index set. In our implementation, we screened out unobserved indices (i, j) from rows and columns with small coherences $\mu_i(\mathcal{U})$ and $\mu_j(\mathcal{V})$, which ensures indices with small values of $\mathrm{H}((i, j)|\Omega)$ in (6.28) are screened out from optimization. Lastly, performing this sequential sampling point-by-point may also be computationally expensive in high-dimensions. In this case, one can simply extend the sequential optimization in (6.36) to select a *batch* of indices with greatest information gain (rather than just one index). Combined together, these speed-ups allow for an efficient and effective information-greedy sampling scheme which improves upon random sampling.

6.6 Numerical examples

6.6.1 Simulations

We now investigate the numerical performance of this integrated UQ and sampling method. For illustration, consider first a small 7×7 example, with $\mathbf{X} \in \mathbb{R}^{7\times 7}$ simulated from the model in Table 6.1. Here, the true matrix rank is $R = 2$, the variance parameters set at $\sigma^2 = 1$ and $\eta^2 = 10^{-4}$, with prior parameters $\alpha_{\eta^2} = \alpha_{\sigma^2} = 9$, $\beta_{\eta^2} = 10^{-3}$, $\beta_{\sigma^2} = 10$, and $\pi_r = 1/5,\ r = 1, \cdots, 5$. Posterior sampling is performed using `gibbs.mc`, with $T = 10,000$ posterior samples for each rank choice. Figure 6.5 shows the resulting posterior mean of $\mathbf{X}$ (see (6.35)), and the nuclear-norm estimator (6.12) optimized via the CVX solver [243]. Both methods employ the same $N_{ini} = 25$ observations (marked with 'x'), which are uniformly sampled. Visually, both estimates provide a close approximation of the true matrix $\mathbf{X}$, with our posterior mean estimate yielding slightly lower error. This shows the proposed model offers comparable completion performance to existing methods, and supports the connection in Lemma 7.

Using the same toy example, we show how the proposed UQ method `gibbs.mc` provides uncertainty for (a) unobserved entries in $\mathbf{X}$, (b) matrix rank, and (c) subspace properties. This is visualized in Figure 6.6. The left plot shows, for each unobserved matrix entry (not marked 'x'), the widths for the mean-symmetric entrywise 95% confidence intervals from posterior samples. Larger widths indicate greater uncertainty for an unobserved entry, and vice versa. We see that entries with greater uncertainty from our method (Figure 6.6, left) tend to have higher incurred errors as well (Figure 6.5, right), with the entrywise 95% posterior intervals covering the actual incurred errors for all unobserved entries. This shows our method not only identifies which entries are most uncertain in the completed matrix, but also yields reliable error bounds for such entries. The middle plot in Figure 6.6 shows the prior and posterior rank probabilities π_r and π_r^P; the former reflects prior belief on matrix rank, and the latter is the resulting rank uncertainty from our method after observing

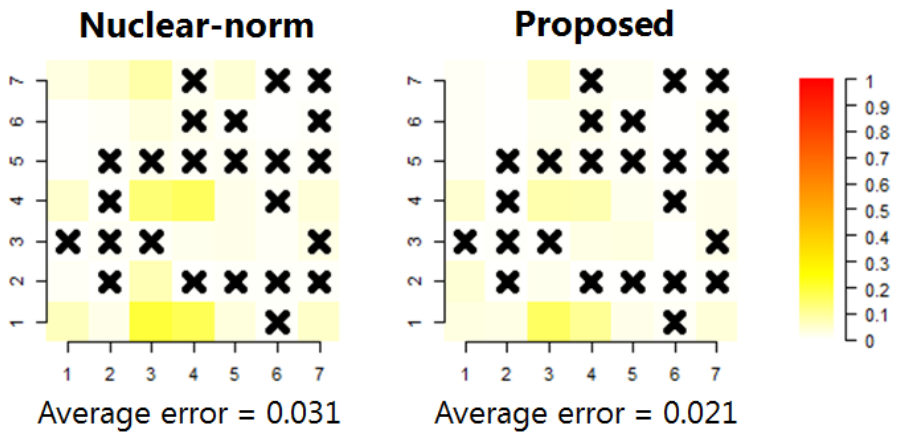

Figure 6.5: Absolute errors (in Frob. norm) for the nuclear-norm estimation of $\mathbf{X}$ (left) and the posterior mean for the proposed method (right). 'x' marks the observed noisy entries.

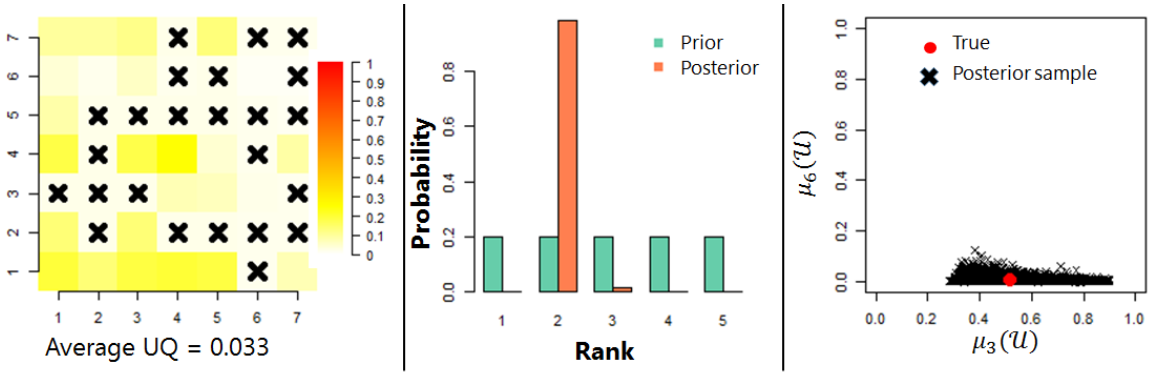

Figure 6.6: (Left) Confidence interval widths for unobserved entries in $\mathbf{X}$. (Middle) Prior and posterior probabilities for matrix rank. (Right) Posterior samples for row coherences $\mu_3(\mathcal{U})$ and $\mu_6(\mathcal{U})$. True coherences in red.

data. After observing $N_{ini} = 25$ entries, our UQ approach identifies with near certainty the true rank of $R = 2$, which is as desired. The right plot shows the posterior samples for two row coherences $\mu_3(\mathcal{U})$ and $\mu_6(\mathcal{U})$, with true coherence values marked in red. This posterior sample can be seen to be highly concentrated around the true coherence values, which shows our method provides effective subspace learning from partial observations.

Next, we compare the initial completion performance of a *balanced* sampling scheme compared to *uniformly sampled* entries. The left and middle plots in Figure 6.7 show, for two realizations of these sampling schemes with $N_{ini} = 7$, the absolute errors between $\mathbf{X}$ and its posterior mean estimate (6.35). We see that the balanced design, by ensuring at

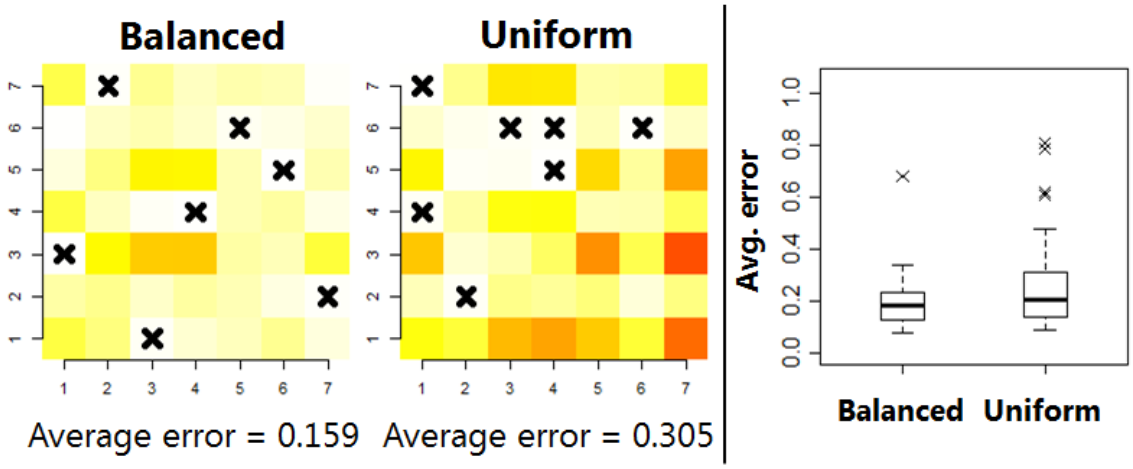

Figure 6.7: (Left and middle) Absolute errors (in Frob. norm) for balanced sampling and uniform sampling. (Right) Error boxplots for 25 randomized balanced and uniform samples.

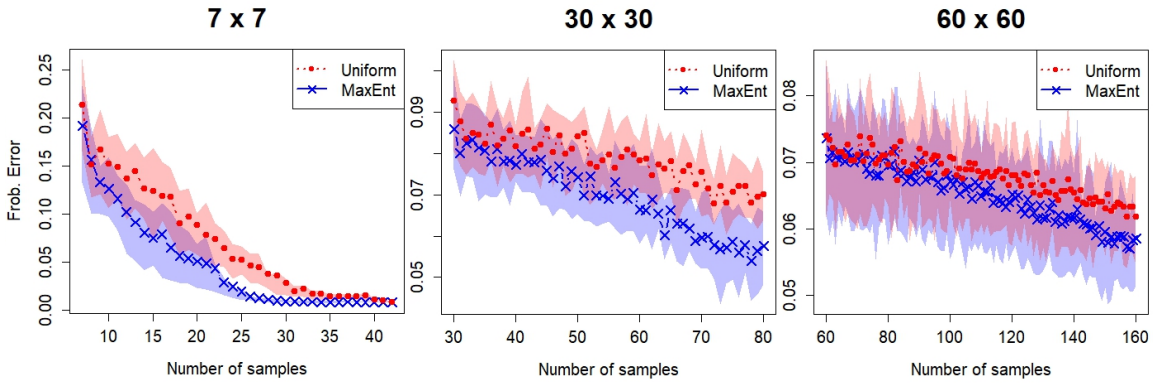

Figure 6.8: Avg. Frob. errors (line) and 25-th/75-th error quantiles (shaded) for the 7×7, 30×30 and 60×60 matrices, using `MaxEnt` and uniform sampling.

least one sample from every row and column, indeed provides lower errors than uniform sampling; the latter also yields much higher errors in *unsampled* rows or columns. The right plot in Figure 6.7 shows the error boxplots for 25 random designs with $N_{ini} = 7$. Again, balanced sampling yields lower errors to uniform sampling at all quantiles, which supports the insight from Section 6.4.2 on initial design.

Finally, we explore the sequential sampling performance of `MaxEnt` for this small 7×7 case, as well as for two larger matrices of sizes 30×30 and 60×60. Simulation settings are the same as before, except with the true rank set as $R = 3$ and $R = 4$ for the two larger matrices, respectively. In all three cases, we begin with an initial sample of $N_{ini} = m_1 = m_2$

entries. For the 7×7 case, $N_{seq} = 28$ entries are then observed sequentially; for the 30×30 case, $N_{seq} = 50$ entries; for the 60×60 case, $N_{seq} = 100$ entries. This procedure is then replicated 10 times to measure error variability. Figure 6.8 shows the averaged errors and the 25-th/75-th error quantiles for `MaxEnt` and uniform sampling. Again, the initial sampling for `MaxEnt` yields noticeably reduced errors to uniform sampling. Moreover, this improvement gap appears to grow larger as more sequential entries are observed; near the end of the sampling procedure, the averaged errors from uniform sampling are noticeably higher than the 75% error quantiles from `MaxEnt`. This shows the effectiveness of our integrated UQ / sampling framework in first (a) learning the underlying subspaces via the UQ model, then (b) incorporating this subspace learning to guide the active learning procedure.

The error decay in Figure 6.8 also reveals two insights. First, despite not knowing the subspaces $\mathcal{U}$ and $\mathcal{V}$ beforehand, the error decays for both sampling schemes are relatively monotone, which supports the error monotonicity result in Corollary 4. Second, the error decay for `MaxEnt` is considerably quicker than that for uniform sampling. When Ω is uniformly sampled, it is known [202] that the completion error $\|\mathbf{X} - \tilde{\mathbf{X}}\|_F$ is upper bounded by $\mathcal{O}\{\sqrt{(m_1 \wedge m_2)(2+p)/p}\}$, where $p = |\Omega|/(m_1 m_2)$ is the fraction of observed entries. Our numerical results suggest that `MaxEnt` may enjoy an improved theoretical error rate to uniform sampling; we look to establish this rate (perhaps via Corollary 5) in a future work.

6.6.2 Collaborative filtering

Finally, we investigate the performance of `MaxEnt` on two collaborative filtering datasets. The first, 'Jester', is collected from the Jester Online Joke Recommender System [244]. Jester contains anonymous user ratings (from -10 to +10) on a test bank of 100 jokes; Figure 6.9 shows some of the arguably better jokes in this test bank. Here, the goal of completing $\mathbf{X}$ from incomplete observations $\mathbf{Y}_\Omega$ can be viewed as *deducing the joke preferences of each person* from a partial survey of their ratings. The proposed sampling scheme

(1) A little girl asked her father, "Daddy? Do all fairy tales begin with 'Once Upon a Time'?" He replied, "No, there is a whole series of fairy tales that begin with 'If elected I promise'.

(2) I'm reading a great book about antigravity—I just can't put it down.

(3) Q: Why did the chicken cross the Mobius Strip? A: To get to the same side.

Figure 6.9: Sample jokes from the Jester dataset.

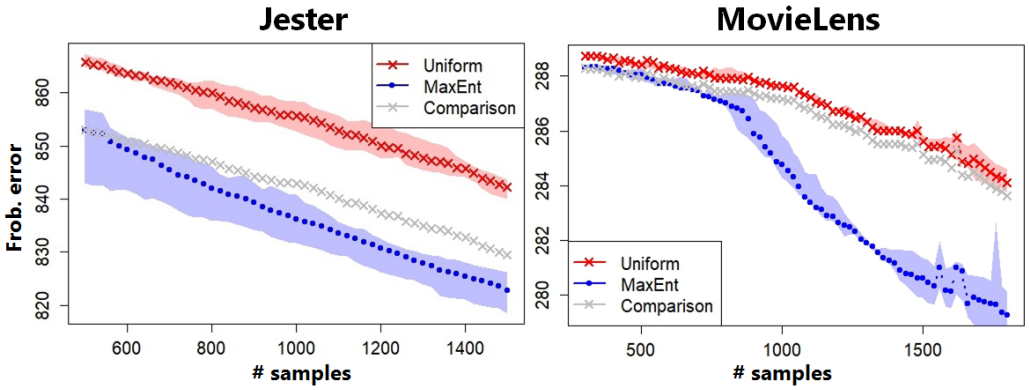

Figure 6.10: Avg. Frob. errors (line) and 25-th/75-th error quantiles (shaded) for Jester (left) and MovieLens (right), using `MaxEnt` and uniform sampling. The grey line marked 'Comparison' compares the error decays for the two methods, by tracing error decay for uniform sampling starting at the initial error for `MaxEnt`.

`MaxEnt` then provides guidance on which user and joke to *query* next, so that maximum information is gained on the joke preferences of the entire community. The second dataset, 'MovieLens', contains anonymous ratings (from 1 to 5) for 1,000 users on 1,700 movies. For this dataset, `MaxEnt` sheds light on which user and movie to *query* next, so that maximum information is gained on the movie preferences of the full userbase.

The simulation settings are as follows. For Jester, we randomly select $m_1 = 500$ users with completed ratings for all $m_2 = 100$ jokes, and take the resulting ratings matrix as $\mathbf{X}$. `MaxEnt` is then compared with uniform sampling, with an initial design of $N_{ini} = 500$ observations and an additional $N_{seq} = 1,000$ observations taken sequentially. For MovieLens, we first pick the $m_1 = 300$ users and $m_2 = 300$ movies with most ratings, and obtain $\mathbf{X}$ by completing the incomplete ratings from these users and movies. `MaxEnt` is then compared with uniform sampling, with $N_{ini} = 300$ and $N_{seq} = 1,500$. This procedure is replicated 10 times to provide a measure of error variability. Since these matrices are

quite large, the fully-Bayesian implementation of `gibbs.mc` can be time intensive, so we employ the computational speed-ups detailed in Section 6.5.2 for efficient active sampling.

Figure 6.10 shows the averaged errors and the 25-th/75-th error quantiles using `MaxEnt` and uniform sampling, for the Jester and MovieLens datasets. Two observations are of interest. First, `MaxEnt` yields noticeably lower *initial* errors to uniform sampling at all error quantiles, which again demonstrates the importance of a balanced initial sample. Second, the improvement gap between `MaxEnt` and uniform sampling grows larger as entries are observed sequentially, more so than from simulations. One reason for this is that high row and column coherences are present in both datasets – there may be users who are overly critical in their ratings, or jokes or movies which are particularly good or bad. By first (a) identifying these preference structures via subspace learning from the UQ model, then (b) incorporating this into an active learning procedure which maximizes information on $\mathbf{X}$, the proposed method offers an effective way of learning the underlying ratings matrix from partial observations.

6.7 Conclusion

In this chapter, we introduce a novel methodology for tackling the joint problems of uncertainty quantification (UQ) and sampling for noisy matrix completion. The proposed method has useful applications in many low-rank modeling problems in statistics, machine learning, and engineering, particularly when the cost of observing each matrix entry is expensive. The centerpiece of this method is a new Bayesian modeling framework, which parametrizes key subspace properties of the desired low-rank matrix $\mathbf{X}$. Using this model, we reveal several new insights on the connection between the problem of UQ and sampling for matrix completion, and well-known concepts from compressive sensing (e.g., coherence) and coding design (e.g., Latin squares). We then present (a) an efficient posterior sampling called `gibbs.mc`, which uses closed-form Gibbs sampling to provide uncertainty on both $\mathbf{X}$ and its subspaces, and (b) a novel information-theoretic active matrix

sampling algorithm called `MaxEnt`, which makes use of this learned subspace information to guide the matrix sampling procedure. Simulations and two real-world applications demonstrate the effectiveness of `MaxEnt` over uniform sampling, and confirm the insights developed in this chapter.

Looking forward, there are several intriguing directions for future work. First, it would be interesting to explore other flavors of design in the experimental design literature, e.g., integrated mean-squared error designs [245] or distance-based designs [72, 61, 246]. Second, it may be worth exploring the theoretical error rate of `MaxEnt` (perhaps via Corollary 5), and how such a rate compares to uniform sampling. Lastly, we are interested in applying `MaxEnt` to design experiments in real-world engineering problems, such as in gene expression studies [201, 247] and quantum state tomography [248].

Appendices

APPENDIX A

APPENDIX FOR CHAPTER 2

A.1 Proof of Proposition 1

It can be shown [249] that $E(F, F_n) = 2D_2^2(F, F_n)$, where F_n is the e.d.f. of $\{x_i\}_{i=1}^n \subseteq \mathfrak{X} \subseteq \mathbb{R}$ and $D_2(F, F_n)$ is the one-dimensional L_2-discrepancy in (1.2). This proves the assertion.

A.2 Proof of Theorem 2

The proof of this theorem relies on the following lemma, which slightly extends the Lévy continuity theorem to the almost-everywhere (a.e.) pointwise convergence setting.

Lemma 10. *Let $(F_n)_{n=1}^\infty$ be a sequence of d.f.s with characteristic functions (c.f.s) $(\phi_n(\mathbf{t}))_{n=1}^\infty$, and let F be a d.f. with c.f. $\phi(\mathbf{t})$. If $\mathbf{X}_n \sim F_n$ and $\mathbf{X} \sim F$, with $\lim_{n\to\infty} \phi_n(\mathbf{t}) = \phi(\mathbf{t})$ a.e. (in the Lebesgue sense), then $\mathbf{X}_n \xrightarrow{d} \mathbf{X}$.*

Proof. (Lemma 10) This proof is a straight-forward extension of Theorem 9.5.2 in [250], but we provide the full argument for clarity. Assume for brevity the univariate setting of $p = 1$, since the proof extends analogously for $p > 1$. Let $\Omega \subseteq \mathbb{C}$ be the set on which $\lim_{n\to\infty} \phi_n(t) = \phi(t)$. By the a.e. assumption, it follows that $\mu\{\Omega^c\} = 0$, where μ is the Lebesgue measure. We will first show that $\{F_n\}_{n=1}^\infty$ is *tight*, i.e., for all $\epsilon > 0$, $\exists$ a finite interval $I \subset \mathbb{R}$ satisfying:

$$G(I^c) \leq \epsilon, \quad \forall F \in \{F_n\}. \tag{A.1}$$

To prove (A.1), fix $M > 0$, $\epsilon > 0$, and let $I = [-M, M]$. By Lemma 9.6.3 in [250], $\exists \alpha \in (0, \infty)$ satisfying:

$$\limsup_{n\to\infty} F_n(I^c) \leq \limsup_{n\to\infty} \alpha M \int_{[0,M^{-1}]} \{1 - \text{Re } \phi_n(t)\} \, dt$$

$$
\begin{aligned}
&= \limsup_{n\to\infty} \alpha M \left[\int_{[0,M^{-1}]\cap\Omega} \{1 - \text{Re } \phi_n(t)\} \ dt + \int_{[0,M^{-1}]\cap\Omega^c} \{1 - \text{Re } \phi_n(t)\} \ dt \right] \\
&= \limsup_{n\to\infty} \alpha M \int_{[0,M^{-1}]\cap\Omega} \{1 - \text{Re } \phi_n(t)\} \ dt \\
&\quad \text{since } \mu\{[0, M^{-1}] \cap \Omega^C\} = 0, \\
&= \alpha M \int_{[0,M^{-1}]\cap\Omega} \limsup_{n\to\infty} \{1 - \text{Re } \phi_n(t)\} \ dt \\
&\quad \text{by dominated convergence, since } 1 - \phi_n(t) \text{ is bounded,} \\
&= \alpha M \int_{[0,M^{-1}]\cap\Omega} \{1 - \text{Re } \phi(t)\} \ dt.
\end{aligned}
$$

Since ϕ is a characteristic function, it follows that $\lim_{t\to 0} \phi(t) = 1$, so $\lim_{t\to 0} \{1 - \text{Re } \phi(t)\} = 0$. Hence, for M sufficiently large, the above becomes:

$$
\alpha M \int_{[0,M^{-1}]\cap\Omega} \{1 - \text{Re } \phi(t)\} \ dt \leq \alpha M \int_{[0,M^{-1}]\cap\Omega} \epsilon \ dt = \alpha\epsilon,
$$

which proves the tightness of $\{F_n\}$.

The remainder of the proof follows exactly as in Theorem 9.5.2 of [250]: one can show that any two convergent subsequences of $\{F_n\}$ must converge to the same limit, thereby proving the convergence of F_n to F. Readers can consult the aforementioned reference for details. □

Proof. (Theorem 2) Define the sequence of random variables $(\mathbf{Y}_i)_{i=1}^{\infty} \overset{i.i.d.}{\sim} F$, and let $\tilde{F}_n$ denote the e.d.f. of $\{\mathbf{Y}_i\}_{i=1}^n$. By the Glivenko-Cantelli lemma, $\lim_{n\to\infty} \sup_{\mathbf{x}\in\mathbb{R}^p} |\tilde{F}_n(\mathbf{x}) - F(\mathbf{x})| = 0$ a.s., so $\tilde{F}_n(\mathbf{x}) \to F(\mathbf{x})$ a.s. for all $\mathbf{x}$. Let $\phi(\mathbf{t})$ and $\tilde{\phi}_n(\mathbf{t})$ denote the c.f.s of F and $\tilde{F}_n$, respectively. Since $|\exp(\mathrm{i}\langle \mathbf{t}, \mathbf{x}\rangle)| \leq 1$, applying the Portmanteau theorem (Theorem 8.4.1 in [250]) and the dominated convergence theorem gives:

$$
\lim_{n\to\infty} \mathbb{E}[|\phi(\mathbf{t}) - \tilde{\phi}_n(\mathbf{t})|^2] = 0. \tag{A.2}
$$

Using Prop. 1 of [30] (this is a *duality* result connecting the energy distance with c.f.s), the expected energy between $\tilde{F}_n$ and F becomes:

$$\mathbb{E}[E(F,\tilde{F}_n)] = \frac{1}{a_p}\mathbb{E}\left[\int \frac{|\phi(\mathbf{t})-\tilde{\phi}_n(\mathbf{t})|^2}{\|\mathbf{t}\|_2^{p+1}}\,d\mathbf{t}\right] = \frac{1}{a_p}\int \frac{\mathbb{E}\left[|\phi(\mathbf{t})-\tilde{\phi}_n(\mathbf{t})|^2\right]}{\|\mathbf{t}\|_2^{p+1}}\,d\mathbf{t}, \tag{A.3}$$

where a_p is some constant depending on p, with the last step following from Fubini's theorem. Note that $\mathbb{E}\left[|\phi(\mathbf{t})-\tilde{\phi}_n(\mathbf{t})|^2\right] = \frac{1}{n}\text{Var}\left[\exp(\mathrm{i}\langle \mathbf{t}, \mathbf{Y}_1\rangle)\right]$, so $\mathbb{E}\left[|\phi(\mathbf{t})-\tilde{\phi}_n(\mathbf{t})|^2\right]$ is monotonically decreasing in n. By the monotone convergence theorem and (A.2), we have:

$$\lim_{n\to\infty}\mathbb{E}[E(F,\tilde{F}_n)] = \frac{1}{a_p}\int \lim_{n\to\infty}\frac{\mathbb{E}[|\phi(\mathbf{t})-\tilde{\phi}_n(\mathbf{t})|^2]}{\|\mathbf{t}\|_2^{p+1}}\,d\mathbf{t} = 0. \tag{A.4}$$

Consider now the e.d.f.s $(F_n)_{n=1}^{\infty}$ and c.f.s $(\phi_n)_{n=1}^{\infty}$ for support points. By Definition 2, $E(F,F_n) \leq \mathbb{E}[E(F,\tilde{F}_n)]$, so $\lim_{n\to\infty} E(F,F_n) = 0$ by (A.4) and the squeeze theorem. Take any subsequence $(n_k)_{k=1}^{\infty} \subseteq \mathbb{N}_+$, and note that:

$$\lim_{k\to\infty} E(F,F_{n_k}) = \lim_{k\to\infty}\int \frac{|\phi(\mathbf{t})-\phi_{n_k}(\mathbf{t})|^2}{\|\mathbf{t}\|_2^{p+1}}\,d\mathbf{t} = 0.$$

We know by the Riesz-Fischer Theorem (pg. 148 in [251]) that a sequence of functions (f_n) which converge to f in L_2 has a subsequence which converges pointwise a.e. to f. Applied here, this suggests the existence of a further subsequence $(n'_k)_{k=1}^{\infty} \subseteq (n_k)_{k=1}^{\infty}$ satisfying $\phi_{n'_k}(\mathbf{t}) \overset{k\to\infty}{\to} \phi(\mathbf{t})$ a.e., so by Lemma 10, $\mathbf{X}_{n'_k} \xrightarrow{d} \mathbf{X}$. Since $(n_k)_{k=1}^{\infty}$ was arbitrarily chosen, it follows by the proof of Corollary 1 in Chapter 9 of [252] that $\mathbf{X}_n \xrightarrow{d} \mathbf{X}$, which is as desired. □

A.3 Proof of Corollary 1

Proof. Part (a) follows from the continuous mapping theorem and Theorem 2. Part (b) follows by the Portmanteau theorem. □

A.4 Proof of Theorem 4

Proof. (Theorem 4) Consider first part (a). Let $\Phi(\cdot) = -\|\cdot\|_2$, and let $\hat{\Phi}$ be its GFT of order 1. From Theorem 8.16 of [32], we have the following *duality* representation:

$$\hat{\Phi}(\omega) = \frac{2^{p/2}\Gamma((p+1)/2)}{\sqrt{\pi}}\|\omega\|_2^{-p-1}, \quad \omega \in \mathbb{R}^p \setminus \{0\}.$$

By Corollary 8.18 of [32], $\Phi(\cdot)$ is also c.p.d. of order 1. Using the fact that $\Phi(\cdot)$ is even along with the continuity of $\hat{\Phi}(\omega)$ on $\mathbb{R}^p \setminus \{0\}$, an application of Theorem 10.21 in [32] completes the proof for part (a).

Consider now part (b). By Prop. 3 of [30], the kernel $\Phi(\cdot)$ is c.p.d. with respect to the space of constant functions $\mathcal{P} = \{f(\mathbf{x}) \equiv C \text{ for some } C \in \mathbb{R}\}$, with $\dim \mathcal{P} = 1$. Note that any choice of $\boldsymbol{\psi} \in \mathcal{X}$ provides a $\mathcal{P}$-unisolvent subset, with the Lagrange basis for the single point $\boldsymbol{\psi}$ being the unit function $p(\cdot) \equiv 1$. Hence, by Theorem 11, the native space $\mathcal{N}_\Phi(\mathbb{R}^p)$ can be transformed into a RKHS $\mathcal{G}_p$ by equipping it with a new inner product $\langle f, g\rangle_{\mathcal{G}_p} = \langle f, g\rangle_{\mathcal{N}_\Phi(\mathbb{R}^p)} + f(\boldsymbol{\psi})g(\boldsymbol{\psi})$. From the same theorem, the corresponding reproducing kernel for the RKHS $(\mathcal{G}_p, \langle\cdot,\cdot\rangle_{\mathcal{G}_p})$ becomes $\tilde{k}(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x} - \mathbf{y}) - \Phi(\boldsymbol{\psi} - \mathbf{y}) - \Phi(\boldsymbol{\psi} - \mathbf{x}) + 1$.

Next, let $\tilde{k}_{\mathbf{x}}(\mathbf{z}) = \tilde{k}(\mathbf{x}, \mathbf{z})$. We claim the function $\int \tilde{k}_{\mathbf{x}}(\cdot)\ d[F - F_n](\mathbf{x})$ belongs in $\mathcal{G}_p$. To see this, define the linear operator $\mathscr{L} : \mathcal{G}_p \to \mathbb{R}$ as $\mathscr{L}f = \int f(\mathbf{x})\ dF(\mathbf{x})$. Note that $\mathscr{L}$ is a bounded operator, because for all $f \in \mathcal{G}_p$:

$$\begin{aligned}
|\mathscr{L}f| = \Big|\int f(\mathbf{x})\ dF(\mathbf{x})\Big| &\leq \int |f(\mathbf{x})|\ dF(\mathbf{x}) \\
&= \int |\langle f(\cdot), \tilde{k}_{\mathbf{x}}(\cdot)\rangle_{\mathcal{G}_p}|\ dF(\mathbf{x}) \quad \text{(RKHS reproducing property)} \\
&\leq \int \|f\|_{\mathcal{G}_p}\|\tilde{k}_{\mathbf{x}}(\cdot)\|_{\mathcal{G}_p}\ dF(\mathbf{x}) \quad \text{(Cauchy-Schwarz)} \\
&= \|f\|_{\mathcal{G}_p}\int \tilde{k}^{1/2}(\mathbf{x}, \mathbf{x})\ dF(\mathbf{x}), \quad \text{(RKHS kernel trick)}
\end{aligned}$$

and the last expression must be bounded because $\int \tilde{k}^{1/2}(\mathbf{x},\mathbf{x})\ dF(\mathbf{x}) \leq [\int \tilde{k}(\mathbf{x},\mathbf{x})\ dF(\mathbf{x})]^{1/2}$, the latter of which is finite due to the earlier finite mean assumption on F. By the Riesz Representation Theorem (Theorem 8.12, [253]), there exists a unique $\tilde{f} \in \mathscr{G}_p$ satisfying $\mathscr{L}f = \int f(\mathbf{x})\ dF(\mathbf{x}) = \langle f, \tilde{f}\rangle_{\mathscr{G}_p}$ for all $f \in \mathscr{G}_p$. Setting $f(\mathbf{x}) = \tilde{k}_{\mathbf{z}}(\mathbf{x})$ in this expression, we get $\int \tilde{k}_{\mathbf{z}}(\mathbf{x})\ dF(\mathbf{x}) = \langle \tilde{k}_{\mathbf{z}}(\cdot), \tilde{f}\rangle_{\mathscr{G}_p} = \tilde{f}(\mathbf{z})$ by the RKHS reproducing property, so $\tilde{f} = \int \tilde{k}_{\mathbf{x}}(\cdot)\ dF(\mathbf{x}) \in \mathscr{G}_p$. Finally, note that $\int \tilde{k}_{\mathbf{x}}(\cdot)\ dF_n(\mathbf{x}) \in \mathscr{G}_p$ because a RKHS is closed under addition, so $\int \tilde{k}_{\mathbf{x}}(\cdot)\ d[F - F_n](\mathbf{x}) \in \mathscr{G}_p$, as desired.

With this in hand, the integration error can be bounded as follows:

$$\begin{aligned}
I(g; F, F_n) &= \left| \int g(\mathbf{x})\ d[F - F_n](\mathbf{x}) \right| \\
&= \left| \int \left\langle g(\cdot), \tilde{k}_{\mathbf{x}}(\cdot) \right\rangle_{\mathscr{G}_p} d[F - F_n](\mathbf{x}) \right| && \text{(Reproducing property)} \\
&= \left| \left\langle g(\cdot), \int \tilde{k}_{\mathbf{x}}(\cdot)\ d[F - F_n](\mathbf{x}) \right\rangle_{\mathscr{G}_p} \right| \\
&\leq \|g\|_{\mathscr{G}_p} \left\| \int \tilde{k}_{\mathbf{x}}(\cdot)\ d[F - F_n](\mathbf{x}) \right\|_{\mathscr{G}_p}. && \text{(Cauchy-Schwarz)}
\end{aligned}$$

The last term can be rewritten as:

$$\begin{aligned}
\sqrt{\left\| \int \tilde{k}_{\mathbf{x}}(\cdot)\ d[F - F_n](\mathbf{x}) \right\|_{\mathscr{G}_p}^2} &= \sqrt{\left\langle \int \tilde{k}_{\mathbf{x}}(\cdot)\ d[F - F_n](\mathbf{x}), \int \tilde{k}_{\mathbf{y}}(\cdot)\ d[F - F_n](\mathbf{y}) \right\rangle_{\mathscr{G}_p}} \\
&= \sqrt{\int \int \langle \tilde{k}_{\mathbf{x}}(\cdot), \tilde{k}_{\mathbf{y}}(\cdot)\rangle_{\mathscr{G}_p}\ d[F - F_n](\mathbf{x})\ d[F - F_n](\mathbf{y})} \\
&= \sqrt{\int \int \tilde{k}(\mathbf{x},\mathbf{y})\ d[F - F_n](\mathbf{x})\ d[F - F_n](\mathbf{y})} \\
& && \text{(Kernel trick)} \\
&= \sqrt{\int \int \Phi(\mathbf{x} - \mathbf{y})\ d[F - F_n](\mathbf{x})\ d[F - F_n](\mathbf{y})} \\
&= \sqrt{E(F, F_n)}, && \text{(Equation (1.4))}
\end{aligned}$$

where the second-last step follows because $\int \Phi(\boldsymbol{\psi} - \mathbf{y})\ d[F - F_n](\mathbf{x}) = \int \Phi(\boldsymbol{\psi} - \mathbf{x})\ d[F -$

$F_n](\mathbf{y}) = \int d[F - F_n](\mathbf{x}) = 0$. This completes the proof. □

A.5 Proof of Theorem 5

The proof of this theorem exploits the fact that $E(F, F_n)$ is a goodness-of-fit statistic. Specifically, writing $E(F, F_n)$ as a degenerate V-statistic V_n, we appeal to its limiting distribution and a uniform Barry-Esseen-like rate to derive an upper bound for the minimum of V_n. The full proof is outlined below, and relies on the following lemmas.

Lemma 11. ([254]) *Let* $(\mathbf{Y}_i)_{i=1}^{\infty} \overset{i.i.d.}{\sim} F$*, and let* k *be a symmetric, positive definite (p.d.) kernel with* $\mathbb{E}[k(\mathbf{x}, \mathbf{Y}_1)] = 0$*,* $\mathbb{E}[k^2(\mathbf{Y}_1, \mathbf{Y}_2)] < \infty$ *and* $\mathbb{E}|k(\mathbf{Y}_1, \mathbf{Y}_1)| < \infty$*. Define the V-statistic* $V_n \equiv n^{-2} \sum_{i=1}^{n} \sum_{j=1}^{n} k(\mathbf{Y}_i, \mathbf{Y}_j)$*. Then* $W_n \equiv nV_n \xrightarrow{d} \sum_{k=1}^{\infty} \lambda_k \chi_k^2 \equiv W_\infty$*, where* $(\chi_k^2)_{k=1}^{\infty} \overset{i.i.d.}{\sim} \chi^2(1)$*, and* $(\lambda_k)_{k=1}^{\infty}$ *are the weighted eigenvalues of* k *under* F*.*

Lemma 12. ([255]) *Adopt the same notation as in Lemma 11, and let* F_{W_n} *and* F_{W_∞} *denote the d.f.s for* W_n *and* W_∞*. If* $\mathbb{E}[k(\mathbf{x}, \mathbf{Y}_1)] = 0$*,* $\mathbb{E}|k(\mathbf{Y}_1, \mathbf{Y}_2)|^3 < \infty$ *and* $\mathbb{E}|k(\mathbf{Y}_1, \mathbf{Y}_1)|^{3/2} < \infty$*, then:*

$$\sup_x |F_{W_n}(x) - F_{W_\infty}(x)| = \mathcal{O}(n^{-1/2}), \tag{A.5}$$

with constants depending on dimension p*.*

Lemma 13 (Paley-Zygmund inequality; [256])**.** *Let* $X \geq 0$*, with constants* $a_1 > 1$ *and* $a_2 > 0$ *satisfying* $\mathbb{E}(X^2) \leq a_1 \mathbb{E}^2(X)$ *and* $\mathbb{E}(X) \geq a_2$*. Then, for any* $\theta \in (0, 1)$*,* $\mathbb{P}(X \geq a_2\theta) \geq (1 - \theta)^2 / a_1$*.*

The proof of Theorem 5 then follows:

Proof. (Theorem 5) Following Section 7.4 of [30], the energy distance $E(F, F_n)$ can be written as the order-2 V-statistic:

$$E(F, F_n) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) \leq \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k(\mathbf{Y}_i, \mathbf{Y}_j) \equiv V_n, \tag{A.6}$$

where $k(\mathbf{x}, \mathbf{y})$ is defined in Theorem 5 and $(\mathbf{Y}_i)_{i=1}^n \overset{i.i.d.}{\sim} F$. The last inequality follows by the definition of support points.

By [257], the kernel k is symmetric and p.d., and the conditions for Lemma 11 can easily be shown to be satisfied. Invoking this lemma, we have:

$$\inf\{x : F_{W_n}(x) > 0\} = nE(F, F_n), \tag{A.7}$$

The strategy is to lower bound the left-tail probability of W_∞, then use this to derive an upper bound for $\inf\{x : F_{W_n}(x) > 0\}$ using Lemma 12.

We first investigate the left-tail behavior of W_∞. Define $Z_t = \exp\{-tW_\infty\}$ for some $t > 0$ to be determined later. Since Z_t is bounded a.s., $\mathbb{E}(Z_t) = \prod_{k=1}^\infty (1 + 2\lambda_k t)^{-1/2}$ and $\mathbb{E}(Z_t^2) = \prod_{k=1}^\infty (1 + 4\lambda_k t)^{-1/2}$. From Lemma 13, it follows that, for fixed $x > 0$, if our choice of t satisfies:

$$\textbf{[A1]}: \ \mathbb{E}(Z_t) \geq 2\exp\{-tx\} > \exp\{-tx\}, \ \textbf{[A2]}: \ \mathbb{E}(Z_t^2) \leq a_1 \mathbb{E}^2(Z_t), \tag{A.8}$$

then, setting $\theta = 1/2$ and $a_2 = 2\exp\{-tx\}$, we have:

$$F_{W_\infty}(x) = \mathbb{P}(Z_t \geq \exp\{-tx\}) \geq \mathbb{P}(Z_t \geq \mathbb{E}(Z_t)/2) \geq (4a_1)^{-1}. \tag{A.9}$$

Consider **[A1]**, or equivalently: $tx \geq \log 2 + (1/2)\sum_{k=1}^\infty \log(1 + 2\lambda_k t)$. Since $\log(1 + x) \leq x^q/q$ for $x > 0$ and $0 < q < 1$, and $\sum_{k=1}^\infty \lambda_k^{1/\alpha} < \infty$ by assumption, a sufficient condition for **[A1]** is:

$$tx \geq \log 2 + (\alpha/2)\sum_{k=1}^\infty (2\lambda_k t)^{1/\alpha} \Leftrightarrow \ P_\alpha(s) \equiv s^\alpha - b_p s x^{-1} - (\log 2)x^{-1} \geq 0,$$

where $s = t^{1/\alpha}$ and $b_p = \alpha 2^{1/\alpha - 1}\sum_{k=1}^\infty \lambda_k^{1/\alpha} > 0$.

Since $\log 2 > 0$ and $b_p s x^{-1} > 0$, there exists exactly one (real) positive root for $P_\alpha(s)$.

Call this root r, so the above inequality is satisfied for $s > r$. Define $\bar{P}_\alpha(s)$ as the linearization of $P_\alpha(s)$ for $s > \bar{s} = (b_p x^{-1})^{1/(\alpha-1)}$, i.e.:

$$\bar{P}_\alpha(s) = \begin{cases} P_\alpha(s), & 0 \le s \le \bar{s} \\ -x^{-1}\log 2 + P'_\alpha(\bar{s}) \cdot (s - \bar{s}), & s > \bar{s}. \end{cases}$$

From this, the unique root of $\bar{P}_\alpha(s)$ can be shown to be $\bar{r} = \bar{s} + x^{-1}(\log 2)[P'_\alpha(\bar{s})]^{-1}$. Since $P_\alpha(s) \ge \bar{P}_\alpha(s)$ for all $s \ge 0$, $\bar{r} \ge r$, the following upper bound for $\bar{r}$ can be obtained for sufficiently small x:

$$\bar{r} = (b_p x^{-1})^{1/(\alpha-1)} + (\log 2)(\alpha - 1)^{-1} b_p^{-1} \le 2(b_p x^{-1})^{1/(\alpha-1)}.$$

Hence:

$$\begin{aligned} t = s^\alpha \ge 2^\alpha (b_p x^{-1})^{\alpha/(\alpha-1)} \Leftrightarrow s \ge 2(b_p x^{-1})^{1/(\alpha-1)} \ge \bar{r} \ge r \\ \Rightarrow \ s^\alpha - b_p x^{-1} s - (\log 2) x^{-1} \ge 0, \end{aligned} \tag{A.10}$$

so setting $t = 2^\alpha (b_p x^{-1})^{\alpha/(\alpha-1)} \equiv c_p x^{-\alpha/(\alpha-1)}$ satisfies **[A1]** in (A.8).

The next step is to determine the smallest a_1 satisfying **[A2]** in (A.8), or equivalently, $\frac{1}{2}\sum_{k=1}^{\infty} \log(1 + 4\lambda_k t) \ge \sum_{k=1}^{\infty} \log(1 + 2\lambda_k t) - \log a_1$. Again, since $\log(1+x) \le x^q/q$ for $x > 0$ and $0 < q < 1$, a sufficient condition for **[A2]** is:

$$\log a_1 \ge \sum_{k=1}^{\infty} \log(1 + 2\lambda_k t) \Leftarrow \log a_1 \ge \alpha \sum_{k=1}^{\infty} (2\lambda_k t)^{1/\alpha}$$

Plugging in $t = c_p x^{-\alpha/(\alpha-1)}$ from (A.10) and letting $d_p \equiv \alpha(2c_p)^{1/\alpha}\left(\sum_{k=1}^{\infty} \lambda_k^{1/\alpha}\right)$, we get $\log a_1 \ge d_p x^{-1/(\alpha-1)} \Leftrightarrow a_1 \ge \exp\left\{d_p x^{-1/(\alpha-1)}\right\}$.

The choice of $t = c_p x^{-\alpha/(\alpha-1)}$ and $a_1 = \exp\left\{d_p x^{-1/(\alpha-1)}\right\}$ therefore **[A1]** and **[A2]** in

(A.9). It follows from (A.9) that:

$$F_{W_\infty}(x) \geq (4a_1)^{-1} = \exp\{-d_p x^{-1/(\alpha-1)}\}/4, \tag{A.11}$$

so $F_{W_\infty}(x)$ converges to 0 at a rate of $\mathcal{O}(\exp\left\{-d_p x^{-1/(\alpha-1)}\right\})$ as $x \to 0^+$.

Consider now the behavior of $\inf\{x : F_{W_n}(x) > 0\}$ as $n \to \infty$. From the uniform bound in Lemma 12, there exists a sequence $(c_{n,p})_{n=1}^{\infty}, \lim_{n\to\infty} c_{n,p} = 0$ such that $|F_{W_n}(x) - F_{W_\infty}(x)| \leq c_{n,p} n^{-1/2}$ for all $x \geq 0$. Setting the right side of (A.11) equal to $2c_{n,p}n^{-1/2}$ and solving for x, we get:

$$x^* = \frac{d_p^{\alpha-1}}{[\frac{1}{2}\log n - \log(8c_{n,p})]^{\alpha-1}} \Rightarrow F_{W_\infty}(x^*) \geq \exp\left\{-d_p (x^*)^{-1/(\alpha-1)}\right\} = 2c_{n,p}n^{-1/2}. \tag{A.12}$$

so Lemma 12 ensures the above choice of x^* satisfies $F_{W_n}(x^*) \geq c_{n,p}n^{-1/2} > 0$.

Using this with (A.7), it follows that:

$$E(F, F_n) = \mathcal{O}\left\{n^{-1}(\log n)^{-(\alpha-1)}\right\},$$

with constants depending on p. Finally, by Theorem 4, we have:

$$I(g; F, F_n) = \mathcal{O}\{\|g\|_{\mathcal{G}_p} n^{-1/2}(\log n)^{-(\alpha-1)/2}\}$$

which is as desired. □

A.6 Proof of Theorem 6

Proof. We first show that $k(\mathbf{x}, \mathbf{y})$ is Lipschitz, i.e., $\exists L < \infty$ such that $\sup_{\mathbf{z}\in\mathfrak{X}} |k(\mathbf{x}, \mathbf{z}) - k(\mathbf{y}, \mathbf{z})| \leq L\|\mathbf{x} - \mathbf{y}\|_2$. Note that:

$$|k(\mathbf{x}, \mathbf{z}) - k(\mathbf{y}, \mathbf{z})| = |\mathbb{E}\|\mathbf{x} - \mathbf{Y}\|_2 - \|\mathbf{x} - \mathbf{z}\|_2 - \mathbb{E}\|\mathbf{y} - \mathbf{Y}\|_2 + \|\mathbf{y} - \mathbf{z}\|_2|$$

$$\leq \int \left| \|\mathbf{x}-\mathbf{z}\|_2 - \|\mathbf{y}-\mathbf{z}\|_2 \right| dF(\mathbf{z}) + \left| \|\mathbf{x}-\mathbf{z}\|_2 - \|\mathbf{y}-\mathbf{z}\| \right|$$
$$\leq \int \|\mathbf{x}-\mathbf{y}\|_2 \, dF(\mathbf{z}) + \|\mathbf{x}-\mathbf{y}\|_2 = 2\|\mathbf{x}-\mathbf{y}\|_2,$$

where the last step holds by the triangle inequality, because $\|\mathbf{x}-\mathbf{y}\|_2 \geq |\|\mathbf{x}-\mathbf{z}\|_2 - \|\mathbf{y}-\mathbf{z}\|_2|$. Hence, k is Lipschitz with $L = 2$.

Consider first part (a) of the theorem. Having satisfied this Lipschitz condition, it follows from Theorem 4 of [258] that $\lambda_k = \mathcal{O}\{k^{-(1+1/p)}\}$, so $\sum_{k=1}^{\infty} \lambda_k^{1/\alpha} < \infty$ for $\alpha \in (0, 1+1/p)$. Applying Theorem 5 proves part (a).

Consider next part (b). From the first example in Section 5 of [259], $\mathfrak{X}$ is $(p,1)$-compact. Moreover, because $F(\cdot)$ is bounded and k is Lipschitz, k must be in $Lip^{1,0}(\mathfrak{X}, F)$ (see [259] for specific definitions). Applying Theorem 5.4 of [259] and noting that $k(\mathbf{x},\mathbf{x}) = 2\mathbb{E}\|\mathbf{x}-\mathbf{Y}\|_2 - \mathbb{E}\|\mathbf{Y}-\mathbf{Y}'\|_2$, it follows that $\lambda_k = \mathcal{O}\{k^{-(1+\gamma/p)}\}$, where $\gamma = \beta/(\beta+1)$. Hence, $\sum_{k=1}^{\infty} \lambda_k^{1/\alpha} < \infty$ for $\alpha \in (0, 1+\gamma/p)$, and part (b) is proven using Theorem 5. □

A.7 Proof of Lemma 1

Proof. Clearly, $Q(\mathbf{x}'|\mathbf{x}') = \|\mathbf{x}'\|_2$. When $\mathbf{x} \neq \mathbf{x}'$, note that:

$$(\|\mathbf{x}\|_2 - \|\mathbf{x}'\|_2)^2 = \|\mathbf{x}\|_2^2 + \|\mathbf{x}'\|_2^2 - 2\|\mathbf{x}\|_2\|\mathbf{x}'\|_2 \geq 0 \Rightarrow \|\mathbf{x}\|_2^2 + \|\mathbf{x}'\|_2^2 \geq 2\|\mathbf{x}\|_2\|\mathbf{x}'\|_2,$$

so $Q(\mathbf{x}|\mathbf{x}') = \frac{\|\mathbf{x}\|_2^2}{2\|\mathbf{x}'\|_2} + \frac{\|\mathbf{x}'\|_2}{2} = \frac{\|\mathbf{x}\|_2^2 + \|\mathbf{x}'\|_2^2}{2\|\mathbf{x}'\|_2} \geq \frac{2\|\mathbf{x}'\|_2\|\mathbf{x}\|_2}{2\|\mathbf{x}'\|_2} = \|\mathbf{x}\|_2$. □

A.8 Proof of Lemma 2

Proof. The majorization claim follows directly from Lemma , and the minimizer in 1.11 can be obtained by first setting ∇h^Q to zero and solving for $\mathbf{x}_i$, $i = 1, \cdots, n$. □

A.9 Proof of Theorem 7

Proof. This follows by Theorem 1 of [260] under certain regularity conditions, which we verify below. Under the earlier assumption of pairwise distinctness for $\{\mathbf{x}'_j\}_{j=1}^n$, $h^Q(\cdot|\{\mathbf{x}'_j\}_{j=1}^n)$ majorizes $\hat{E}(\cdot)$ at $\{\mathbf{x}'_j\}_{j=1}^n$ by Lemma 6, which satisfies assumptions (A1) and (A2) in [260]. Moreover, h^Q is continuous, with its directional derivative $h^{Q\prime}(\{\mathbf{x}'_j\}_{j=1}^n, \mathbf{d}; \{\mathbf{x}'_j\}_{j=1}^n)$ equal to the directional derivative $\hat{E}'(\{\mathbf{x}'_j\}_{j=1}^n, \mathbf{d})$ for all feasible directions $\mathbf{d} \in \mathbb{R}^{np}$, which satisfies assumptions (A3) and (A4) in [260]. This proves the stationary convergence of `sp.ccp`. □

A.10 Proof of Theorem 8

Proof. Under certain regularity conditions, Prop. 3.4 of [45] shows that a stationary solution can be obtained for E by repeatedly applying MM iterations on the Monte Carlo approximation of E (namely, $\hat{E}$), with each iteration employing a new batch sample $\{\mathbf{y}_m^{[l]}\}$ independently generated from F. Such regularity conditions are satisfied by the compactness of $\mathfrak{X}$ and the existence of directional derivatives for $\hat{E}$ and E, so the claim holds. □

APPENDIX B

APPENDIX FOR CHAPTER 3

B.1 Proof of Theorem 9

Since $\gamma_{\boldsymbol{\theta}}$ is a strictly p.d. kernel, $-\gamma_{\boldsymbol{\theta}}$ must be strictly negative-definite (n.d.). By Prop. 3 in [30], the metric property holds for $E_{\boldsymbol{\theta}}(F, F_n)$.

To prove this for the π-expected kernel $-\mathbb{E}_{\boldsymbol{\theta}\sim\pi}[\gamma_{\boldsymbol{\theta}}]$, we need to show that this expected kernel is strictly n.d. Since $-\gamma_{\boldsymbol{\theta}}$ is strictly n.d., we know that for any function $c : \mathcal{X} \to \mathbb{R}$, $c \in L_2(\mathcal{X})$, we have $-\int_{\mathcal{X}}\int_{\mathcal{X}} c(\mathbf{x})c(\mathbf{y})\gamma_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})\ d\mathbf{x}\ d\mathbf{y} \geq 0$, with equality holding if and only if $c(\mathbf{x}) = 0$. Letting π be a proper prior for $\boldsymbol{\theta}$, note that:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\theta}}\left[\int_{\mathcal{X}}\int_{\mathcal{X}} |c(\mathbf{x})||c(\mathbf{y})|\gamma_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})\ d\mathbf{x}\ d\mathbf{y}\right] &\leq \int_{\Theta}\int_{\mathcal{X}}\int_{\mathcal{X}} |c(\mathbf{x})||c(\mathbf{y})|\pi(\boldsymbol{\theta})\ d\mathbf{x}\ d\mathbf{y}\ d\boldsymbol{\theta} \\ &\propto \int_{\mathcal{X}} c^2(\mathbf{x})\ d\mathbf{x} < \infty.\end{aligned}$$

Hence, by Fubini's theorem:

$$0 \leq -\mathbb{E}_{\boldsymbol{\theta}}\left[\int_{\mathcal{X}}\int_{\mathcal{X}} c(\mathbf{x})c(\mathbf{y})\gamma_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})\ d\mathbf{x}\ d\mathbf{y}\right] = -\int_{\mathcal{X}}\int_{\mathcal{X}} c(\mathbf{x})c(\mathbf{y})\mathbb{E}_{\boldsymbol{\theta}}\left[\gamma_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})\right]\ d\mathbf{x}\ d\mathbf{y},$$

with equality holding if and only if $c(\mathbf{x}) = 0$. Hence, $-\mathbb{E}_{\boldsymbol{\theta}\sim\pi}\gamma_{\boldsymbol{\theta}}$ must also be strictly n.d., and so the metric property holds for $E_{\boldsymbol{\theta}\sim\pi}(F, F_n)$.

B.2 Proof of Theorem 10

Let $\mathbf{Y}, \mathbf{Y}' \overset{i.i.d.}{\sim} F$. For any $i = 1, \cdots, n$ and $l = 1, \cdots, p$, note that:

$$\left|\frac{\partial}{\partial x_{il}} E_{\boldsymbol{\theta}}(F, F_n)\right| = \left|\frac{4\theta_l}{n}\mathbb{E}\left\{(x_{il} - Y_l)\gamma_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{Y})\right\} - \frac{4\theta_l}{n^2}\sum_{\substack{j=1 \\ j\neq i}}^{n}(x_{il} - x_{jl})\gamma_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j)\right|$$

(by dominated convergence)

$$\begin{aligned}
&\approx \left|\frac{4\theta_l\bar{\gamma}}{n}\int -(x_{il} - z)\; d[F_{l,n} - F_l](z)\right| && \text{(by Assumption 1)} \\
&\leq \left|\frac{4\theta_l\bar{\gamma}}{n}\int -|x_{il} - z|\; d[F_{l,n} - F_l](z)\right| \\
&= \left|\frac{4\theta_l\bar{\gamma}}{n}\left(\mathbb{E}|x_{il} - Y_l| - \frac{1}{n}\sum_{j=1}^{n}|x_{il} - x_{jl}|\right)\right|,
\end{aligned}$$

where the second-last line follows because $-|x-z|+(x-z) = -2(z-x)_+$ is conditionally p.d. (see [30]). It follows that:

$$\begin{aligned}
\|\nabla_{\mathbf{x}_{(l)}} E_\theta(F, F_n)\|_1 &= \sum_{i=1}^{n}\left|\frac{\partial}{\partial x_{il}} E_{\boldsymbol{\theta}}(F, F_n)\right| \\
&\leq 4\theta_l\bar{\gamma}\sum_{i=1}^{n}\left|\frac{1}{n}\mathbb{E}|x_{il} - Y_l| - \frac{1}{n^2}\sum_{j=1}^{n}|x_{il} - x_{jl}|\right| \\
&\approx 4\theta_l\bar{\gamma}\sum_{i=1}^{n}\left[\frac{2}{n}\mathbb{E}|x_{il} - Y_l| - \frac{1}{n^2}\sum_{j=1}^{n}|x_{il} - x_{jl}| - \mathbb{E}|Y_l - Y_l'|\right] \\
&\qquad \text{(by Assumption 2)} \\
&= 4\theta_l\bar{\gamma}E(F_l, F_{l,n}).
\end{aligned}$$

B.3 Proof of Theorem 11

We require an important lemma to prove this theorem:

Lemma 14. *[261] Suppose H is a separable Hilbert space of functions on $\mathfrak{X}$ with orthonormal basis $\{\phi_k(\mathbf{x})\}_{k=0}^{\infty}$. Then H is a RKHS if and only if $\sum_{k=0}^{\infty}|\phi_k(\mathbf{x})|^2 < \infty$ for any $\mathbf{x} \in \mathfrak{X}$, with unique kernel given by $k(\mathbf{x}, \mathbf{y}) = \sum_{k=0}^{\infty}\phi_k(\mathbf{x})\phi_k(\mathbf{y})$.*

We adopt a similar approach as [262] to derive the RKHS for $\gamma_{\boldsymbol{\theta}}$. Note that:

$$\begin{aligned}\gamma_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y}) &= \exp(-\|\mathbf{x}-\mathbf{y}\|_{\boldsymbol{\theta}}^2) \\ &= \exp(-\|\mathbf{x}\|_{\boldsymbol{\theta}}^2)\exp(-\|\mathbf{y}\|_{\boldsymbol{\theta}}^2)\exp(2\langle \mathbf{x},\mathbf{y}\rangle_{\boldsymbol{\theta}}) \\ &= \exp(-\|\mathbf{x}\|_{\boldsymbol{\theta}}^2)\exp(-\|\mathbf{y}\|_{\boldsymbol{\theta}}^2)\sum_{k=0}^{\infty}\frac{2^k}{k!}\sum_{|\boldsymbol{\alpha}|=k} C_{\boldsymbol{\alpha}}^k \mathbf{x}^{\boldsymbol{\alpha}}\mathbf{y}^{\boldsymbol{\alpha}}\boldsymbol{\theta}^{\boldsymbol{\alpha}},\end{aligned} \tag{B.1}$$

where the last step follows by the series expansion:

$$\exp(2\langle \mathbf{x},\mathbf{y}\rangle_{\boldsymbol{\theta}}) = \sum_{k=0}^{\infty}\frac{2^k\langle \mathbf{x},\mathbf{y}\rangle_{\boldsymbol{\theta}}^k}{k!} = \sum_{k=0}^{\infty}\frac{2^k}{k!}\sum_{|\boldsymbol{\alpha}|=k} C_{\boldsymbol{\alpha}}^k \mathbf{x}^{\boldsymbol{\alpha}}\mathbf{y}^{\boldsymbol{\alpha}}\boldsymbol{\theta}^{\boldsymbol{\alpha}}.$$

Now, assume $H_{\gamma,\boldsymbol{\theta}}$ is the space in (2.11) with inner product (2.12). The completeness of $H_{\gamma,\boldsymbol{\theta}}$ can be shown using a similar argument in [262], so $(H_{\gamma,\boldsymbol{\theta}}, \langle\cdot,\cdot\rangle_{\gamma,\boldsymbol{\theta}})$ is a valid Hilbert space. Define the basis $\phi_{\boldsymbol{\alpha}}(\mathbf{x}) = \sqrt{2^k C_{\boldsymbol{\alpha}}^k \boldsymbol{\theta}^{\boldsymbol{\alpha}}/|\boldsymbol{\alpha}|!}\exp(-\|\mathbf{x}\|_{\boldsymbol{\theta}}^2)\mathbf{x}^{\boldsymbol{\alpha}}$, $|\boldsymbol{\alpha}| \in \mathbb{N}_0$, and note that (a) this basis is orthonormal under the inner product in (2.12), and (b) $\text{span}\{\phi_{\boldsymbol{\alpha}}(\mathbf{x})\} = H_{\gamma,\boldsymbol{\theta}}$, which shows $H_{\gamma,\boldsymbol{\theta}}$ is separable. Moreover, because:

$$\sum_{k=0}^{\infty}\sum_{|\boldsymbol{\alpha}|=k}\phi_{\boldsymbol{\alpha}}^2(\mathbf{x}) < \infty$$

and:

$$\sum_{k=0}^{\infty}\sum_{|\boldsymbol{\alpha}|=k}\phi_{\boldsymbol{\alpha}}(\mathbf{x})\phi_{\boldsymbol{\alpha}}(\mathbf{y}) = \sum_{k=0}^{\infty}\sum_{|\boldsymbol{\alpha}|=k}\frac{2^k C_{\boldsymbol{\alpha}}^k \boldsymbol{\theta}^{\boldsymbol{\alpha}}}{k!}\exp(-\|\mathbf{x}\|_{\boldsymbol{\theta}}^2)\exp(-\|\mathbf{y}\|_{\boldsymbol{\theta}}^2)\mathbf{x}^{\boldsymbol{\alpha}}\mathbf{y}^{\boldsymbol{\alpha}} = \gamma_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y}),$$

it follows by Lemma 14 that $(H_{\gamma,\boldsymbol{\theta}}, \langle\cdot,\cdot\rangle_{\gamma,\boldsymbol{\theta}})$ is the RKHS corresponding to kernel $\gamma_{\boldsymbol{\theta}}$.

B.4 Proof of Lemma 3

By the reproducing property of $(H_{\gamma,\boldsymbol{\theta}}, \langle\cdot,\cdot\rangle_{\gamma,\boldsymbol{\theta}})$, it follows that $g(\mathbf{x}) = \langle g(\cdot), \gamma_{\boldsymbol{\theta}}(\mathbf{x},\cdot)\rangle_{\gamma,\boldsymbol{\theta}}$. Hence:

$$\begin{aligned}
I(g;F,F_n) \equiv \left|\int_{\mathbb{X}} g(\mathbf{x})\; d[F-F_n](\mathbf{x})\right| &= \left|\int_{\mathbb{X}} \langle g(\cdot), \gamma_{\boldsymbol{\theta}}(\mathbf{x},\cdot)\rangle_{\gamma,\boldsymbol{\theta}}\; d[F-F_n](\mathbf{x})\right| \\
&= \left|\left\langle g(\cdot), \int_{\mathbb{X}} \gamma_{\boldsymbol{\theta}}(\mathbf{x},\cdot)\; d[F-F_n](\mathbf{x})\right\rangle_{\gamma,\boldsymbol{\theta}}\right| \\
&\leq \|g\|_{\gamma,\boldsymbol{\theta}} \sqrt{\left\|\int_{\mathbb{X}} \gamma_{\boldsymbol{\theta}}(\mathbf{x},\cdot)\; d[F-F_n](\mathbf{x})\right\|^2_{\gamma,\boldsymbol{\theta}}} \\
&\qquad\qquad \text{(Cauchy-Schwarz)} \\
&= \|g\|_{\gamma,\boldsymbol{\theta}} \sqrt{\int_{\mathbb{X}}\int_{\mathbb{X}} \gamma_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})\; d[F-F_n](\mathbf{x})\; d[F-F_n](\mathbf{y})} \\
&\qquad\qquad \text{(using the kernel trick on } (H_{\gamma,\boldsymbol{\theta}}, \langle\cdot,\cdot\rangle_{\gamma,\boldsymbol{\theta}})) \\
&= \|g\|_{\gamma,\boldsymbol{\theta}} \sqrt{E_{\boldsymbol{\theta}}(F,F_n)},
\end{aligned}$$

where the inequality follows from a simple application of Cauchy-Schwarz.

B.5 Proof of Theorem 12

To prove this theorem, we require a lemma:

Lemma 15. *For fixed p and $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_p)$, $\alpha_l \in \mathbb{Z}_+$, $\lim_{k\to\infty} \sum_{|\boldsymbol{\alpha}|=k} 1/C^k_{\boldsymbol{\alpha}} = p$.*

Proof. Fix $p \in \mathbb{Z}_+$, and consider the following decomposition for sufficiently large $k \in \mathbb{Z}_+$:

$$\sum_{|\boldsymbol{\alpha}|=k} \frac{1}{C^k_{\boldsymbol{\alpha}}} = \sum_{|\boldsymbol{\alpha}|=k, \exists \alpha_l = k} \frac{1}{C^k_{\boldsymbol{\alpha}}} + \sum_{|\boldsymbol{\alpha}|=k, \exists \alpha_l = k-1} \frac{1}{C^k_{\boldsymbol{\alpha}}} + \cdots + \sum_{|\boldsymbol{\alpha}|=k, \exists \alpha_l = k-p+1} \frac{1}{C^k_{\boldsymbol{\alpha}}} + \sum_{|\boldsymbol{\alpha}|=k, \alpha_l \leq k-p} \frac{1}{C^k_{\boldsymbol{\alpha}}}.$$

For the first sum, it is easy to see that $\sum_{|\boldsymbol{\alpha}|=k, \exists \alpha_l = k} 1/C^k_{\boldsymbol{\alpha}} = p$, because there are p terms in this sum, with each term equal to 1. For the second sum, one can similarly show that

$\sum_{|\boldsymbol{\alpha}|=k,\exists\alpha_l=k-1} 1/C_{\boldsymbol{\alpha}}^k = \mathcal{O}(p^2/k)$, because there are $\mathcal{O}(p^2)$ terms in this sum, with each term bounded above by $1/k$. Extending the same argument for remaining terms, the above decomposition can be rewritten as:

$$\sum_{|\boldsymbol{\alpha}|=k} \frac{1}{C_{\boldsymbol{\alpha}}^k} = p + \mathcal{O}\left(\frac{p^2}{k}\right) + \cdots + \mathcal{O}\left(\frac{p^{p+1}}{k(k-1)\cdots(k-p+1)}\right) + \sum_{|\boldsymbol{\alpha}|=k,\alpha_l\leq k-p} \frac{1}{C_{\boldsymbol{\alpha}}^k}.$$

Consider now the last sum $\sum_{|\boldsymbol{\alpha}|=k,\alpha_l\leq k-p} 1/C_{\boldsymbol{\alpha}}^k$. Note that $|\{\boldsymbol{\alpha} : \sum_l \alpha_l = k\}| = \binom{k-1}{p-1}$ (this is the number of ways to put k balls in p containers), so there are at most $\binom{k-1}{p-1}$ terms in this term. Moreover, $1/C_{\boldsymbol{\alpha}}^k \leq p!/(k(k-1)\cdots(k-p+1))$ whenever $|\boldsymbol{\alpha}| = k$, $\alpha_l \leq k-p$. Combining these two facts, we get $\sum_{|\boldsymbol{\alpha}|=k,\alpha_l<k-p} 1/C_{\boldsymbol{\alpha}}^k \leq p/k$. Hence:

$$\lim_{k\to\infty} \sum_{|\boldsymbol{\alpha}|=k} \frac{1}{C_{\boldsymbol{\alpha}}^k} = \lim_{k\to\infty} \left\{ p + \mathcal{O}\left(\frac{p^2}{k}\right) + \cdots + \mathcal{O}\left(\frac{p^{p+1}}{k(k-1)\cdots(k-p+1)}\right) + \frac{p}{k} \right\} = p.$$

□

Consider the two terms in the bound of Lemma 3: $\|g\|_{\gamma,\boldsymbol{\theta}}$ and $\sqrt{E_{\boldsymbol{\theta}}(F, F_n)}$. Letting $\tilde{F}_n$ be the e.d.f. of $\{\mathbf{x}_i\}_{i=1}^n$, $(\mathbf{x}_i)_{i=1}^\infty \overset{i.i.d.}{\sim} F$, the expected discrepancy for this random point set becomes:

$$\begin{aligned}
\mathbb{E}[E_{\boldsymbol{\theta}}(F, \tilde{F}_n)] &= \mathbb{E}_{\{\mathbf{x}_i\}} \left[\mathbb{E}\{\gamma_{\boldsymbol{\theta}}(\mathbf{Y}, \mathbf{Y}')\} - \frac{2}{n}\sum_{i=1}^n \mathbb{E}\{\gamma_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{Y})\} + \frac{1}{n^2}\sum_{i=1}^n\sum_{j=1}^n \gamma_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j) \right] \\
&= \mathbb{E}\left[\gamma_{\boldsymbol{\theta}}(\mathbf{Y}, \mathbf{Y}') - \frac{2}{n}\sum_{i=1}^n \mathbb{E}_{\{\mathbf{x}_i\}}\{\gamma_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{Y})\} + \frac{1}{n^2}\sum_{i=1}^n\sum_{j=1}^n \mathbb{E}_{\{\mathbf{x}_i\}}\{\gamma_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j)\} \right] \\
&= \mathbb{E}\{\gamma_{\boldsymbol{\theta}}(\mathbf{Y}, \mathbf{Y}')\} - 2\mathbb{E}\{\gamma_{\boldsymbol{\theta}}(\mathbf{Y}, \mathbf{Y}')\} \\
&\quad + \frac{1}{n^2}\left[n(n-1)\mathbb{E}\{\gamma_{\boldsymbol{\theta}}(\mathbf{Y}, \mathbf{Y}')\} + n\mathbb{E}\{\gamma_{\boldsymbol{\theta}}(\mathbf{Y}, \mathbf{Y})\} \right] \\
&= \frac{1}{n}\left[1 - \mathbb{E}\{\gamma_{\boldsymbol{\theta}}(\mathbf{Y}, \mathbf{Y}')\} \right] \leq \frac{1}{n}.
\end{aligned}$$

Because PSPs are defined as the *minimizer* of $E_{\boldsymbol{\theta}}(F, F_n)$, it follows by the above averaging argument that $\sqrt{E_{\boldsymbol{\theta}}(F, F_n)} \leq 1/\sqrt{n}$.

Next, consider the second term $\|g\|_{\gamma,\boldsymbol{\theta}}$. By Theorem 11, we have:

$$\begin{aligned}
\|g\|_{\gamma,\boldsymbol{\theta}}^2 = \sum_{k=0}^{\infty} \frac{k!}{2^k} \sum_{|\boldsymbol{\alpha}|=k} \frac{w_{\boldsymbol{\alpha}}^2}{C_{\boldsymbol{\alpha}}^k \boldsymbol{\theta}^{\boldsymbol{\alpha}}} &= \sum_{k=0}^{\infty} \frac{k!}{2^k} \sum_{|\boldsymbol{\alpha}|=k} \left(\frac{1}{(C_{\boldsymbol{\alpha}}^k)^{3/2}} \right) \left(\frac{\sqrt{C_{\boldsymbol{\alpha}}^k} w_{\boldsymbol{\alpha}}^2}{\boldsymbol{\theta}^{\boldsymbol{\alpha}}} \right) \\
&\leq \sum_{k=0}^{\infty} \frac{k!}{2^k} \sqrt{\sum_{|\boldsymbol{\alpha}|=k} \frac{1}{(C_{\boldsymbol{\alpha}}^k)^3}} \sqrt{\sum_{|\boldsymbol{\alpha}|=k} \frac{C_{\boldsymbol{\alpha}}^k w_{\boldsymbol{\alpha}}^4}{\boldsymbol{\theta}^{2\boldsymbol{\alpha}}}} \quad \text{(Cauchy-Schwarz)} \\
&= \sum_{k=0}^{\infty} \frac{C}{\sqrt{p} 2^k} \sqrt{\sum_{|\boldsymbol{\alpha}|=k} \frac{1}{(C_{\boldsymbol{\alpha}}^k)^3}} \sqrt{\sum_{|\boldsymbol{\alpha}|=k} C_{\boldsymbol{\alpha}}^k \prod_{l=1}^{p} \left(\frac{w_l^4}{\theta_l^2} \right)^{\alpha_l}} \\
&\qquad \text{(POD form of } w_{\boldsymbol{\alpha}} \text{ and } \Gamma_{|\boldsymbol{\alpha}|} \leq C/\{p^{1/4}(|\boldsymbol{\alpha}|!)^{1/2}\}) \\
&\leq \sum_{k=0}^{\infty} \frac{C}{\sqrt{p} 2^k} \sqrt{\sum_{|\boldsymbol{\alpha}|=k} \frac{1}{(C_{\boldsymbol{\alpha}}^k)}} \sqrt{\sum_{|\boldsymbol{\alpha}|=k} C_{\boldsymbol{\alpha}}^k \prod_{l=1}^{p} \left(\frac{w_l^4}{\theta_l^2} \right)^{\alpha_l}} \\
&\leq \sum_{k=0}^{\infty} \frac{C}{\sqrt{p} 2^k} \sqrt{\sum_{|\boldsymbol{\alpha}|=k} \frac{1}{(C_{\boldsymbol{\alpha}}^k)}} \left(\sqrt{\sum_{l=1}^{p} \frac{w_l^4}{\theta_l^2}} \right)^k . \\
&\qquad \text{(Binomial theorem)}
\end{aligned}$$

Taking the limit as $k \to \infty$, Lemma 15 gives $\sqrt{\sum_{|\boldsymbol{\alpha}|=k} 1/(C_{\boldsymbol{\alpha}}^k)} \to \sqrt{p}$. Finally, if $\sum_{l=1}^{\infty} w_l^4/\theta_l^2 < 4$, the above series converges to a constant independent of p, as desired. Combining this with Lemma 3, the proof is complete.

B.6 Proof of Theorem 13

This follows by a direct extension of Theorems 4 and 5 in [72].

B.7 Proof of Proposition 2

Rewrite the $\tilde{\pi}$-expected discrepancy as $E_{\boldsymbol{\theta}\sim\tilde{\pi}}(F, F_n) = \int_{\mathcal{X}}\int_{\mathcal{X}} \mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}}\{\gamma_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})\}\, d[F-F_n](\mathbf{x})\, d[F-F_n](\mathbf{y})$. The integrand $\mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}}\{\gamma_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})\}$ can be simplified as:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}}\{\gamma_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})\} &= \mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}}\left[\exp\left\{-\sum_{l=1}^{p}\theta_l(x_l-y_l)^2\right\}\right] \\
&= \prod_{l=1}^{p}\left[\int_0^\infty \exp\{-\theta_l(x_l-y_l)^2\}\cdot\left\{\frac{\lambda^\nu}{\Gamma(\nu)}\theta_l^{\nu-1}\exp(-\lambda\theta_l)\right\}\, d\theta_l\right] \\
&= \prod_{l=1}^{p}\left\{\frac{\lambda}{(x_l-y_l)^2+\lambda}\right\}^{\nu},
\end{aligned}$$

which completes the proof.

B.8 Proof of Lemma 4

First consider the majorizing paraboloid $\bar{Q}$ in (2.22). It is easy to show that:

$$\nabla_{\mathbf{z}}\gamma_{\boldsymbol{\theta}}(\mathbf{z}) = -2\gamma_{\boldsymbol{\theta}}(\mathbf{z})\Omega_{\boldsymbol{\theta}}\mathbf{z} \quad \text{and} \quad \nabla^2_{\mathbf{z}}\gamma_{\boldsymbol{\theta}}(\mathbf{z}) = 2\gamma_{\boldsymbol{\theta}}(\mathbf{z})\left[2\Omega_{\boldsymbol{\theta}}\mathbf{z}(\Omega_{\boldsymbol{\theta}}\mathbf{z})^T - \Omega_{\boldsymbol{\theta}}\right].$$

Note that, for any $\mathbf{z}\in\mathbb{R}^p$:

$$\begin{aligned}
\nabla^2\gamma_{\boldsymbol{\theta}}(\mathbf{z}) \preceq 4\gamma_{\boldsymbol{\theta}}(\mathbf{z})(\Omega_{\boldsymbol{\theta}}\mathbf{z})(\Omega_{\boldsymbol{\theta}}\mathbf{z})^T &\preceq 4\gamma_{\boldsymbol{\theta}}(\mathbf{z})\|\Omega_{\boldsymbol{\theta}}\mathbf{z}\|_2^2\mathbf{I}_p \\
&\preceq 4\exp\left\{-\sum_{l=1}^{p}\Omega_{\boldsymbol{\theta},ll}\|\mathbf{z}_l\|_2^2\right\}\left(\sum_{l=1}^{p}\Omega_{\boldsymbol{\theta},ll}\|\mathbf{z}_l\|_2^2\right)\left(\max_l \Omega_{\boldsymbol{\theta},ll}\right)\mathbf{I}_p \\
&\preceq \frac{4}{e}\left(\max_l \Omega_{\boldsymbol{\theta},ll}\right)\mathbf{I}_p = 4\Delta_{\boldsymbol{\theta}}. \qquad (\min_z \exp\{-z^2\}z^2 = \frac{1}{e})
\end{aligned}$$

Using a second-order Taylor expansion of $\gamma_{\boldsymbol{\theta}}(\mathbf{z})$ at $\mathbf{z}=\mathbf{z}'$, the following must hold for some $\boldsymbol{\xi} = (1-t)\mathbf{z}+t\mathbf{z}'$ with $t\in[0,1]$:

$$\gamma_{\boldsymbol{\theta}}(\mathbf{z}) = \gamma_{\boldsymbol{\theta}}(\mathbf{z}') - 2[\gamma_{\boldsymbol{\theta}}(\mathbf{z}')\Omega_{\boldsymbol{\theta}}\mathbf{z}']^T(\mathbf{z}-\mathbf{z}') + \frac{1}{2}(\mathbf{z}-\mathbf{z}')^T[\nabla^2\gamma_{\boldsymbol{\theta}}(\boldsymbol{\xi})](\mathbf{z}-\mathbf{z}') \le \bar{Q}(\mathbf{z}|\mathbf{z}').$$

By definition, $\bar{Q}(\mathbf{z}|\mathbf{z}')$ majorizes $\gamma_{\boldsymbol{\theta}}(\mathbf{z})$ at $\mathbf{z} = \mathbf{z}'$.

Next, consider the minorizing paraboloid $\underline{Q}$ in (2.23). Note that $\exp(t) \geq (1-t')\exp(t')+ t\exp(t')$ by convexity. Hence:

$$\begin{aligned}\gamma_{\boldsymbol{\theta}}(\mathbf{z}) &\geq \gamma_{\boldsymbol{\theta}}(\mathbf{z}')\left[1 + \sum_{\varnothing \neq \mathbf{u} \subseteq [p]} \theta_{\mathbf{u}} \|\mathbf{z}'_{\mathbf{u}}\|_2^2\right] - \gamma_{\boldsymbol{\theta}}(\mathbf{z}') \sum_{\varnothing \neq \mathbf{u} \subseteq [p]} \theta_{\mathbf{u}} \|\mathbf{z}_{\mathbf{u}}\|_2^2 \\ &= \gamma_{\boldsymbol{\theta}}(\mathbf{z}')\left[1 + \mathbf{z}'\Omega_{\boldsymbol{\theta}}\mathbf{z}'\right] - \gamma_{\boldsymbol{\theta}}(\mathbf{z}')\mathbf{z}^T\Omega_{\boldsymbol{\theta}}\mathbf{z},\end{aligned}$$

which completes the proof.

B.9 Proof of Lemma 5

The majorization claim follows directly from Lemma 4, and the closed-form minimizer can be obtained by setting the gradient of h_i to zero and solving for $\mathbf{x}$.

B.10 Proof of Theorem 14

Under certain regularity conditions, parts (a) and (b) follow from Theorem 1 of [260] and Prop. 3.4 of [45], respectively. These conditions are satisfied by the closedness / compactness of $\mathfrak{X}$ and Θ, and the differentiability of $\gamma_{\boldsymbol{\theta}}(\cdot)$.

B.11 Proof of Theorem 15

Starting from the i-th entry of the diagonal of $\Omega_{\boldsymbol{\theta}}$, $i = 1, \cdots, p$, we get:

$$\begin{aligned}\Omega_{\boldsymbol{\theta},ii} = \sum_{i \in \mathbf{u} \subseteq [p]} \Gamma_{|\mathbf{u}|} \prod_{l \in \mathbf{u}} \theta_l &= \sum_{k=1}^{p} \sum_{i \in \mathbf{u} \subseteq [p], |\mathbf{u}|=k} \Gamma_{|\mathbf{u}|} \prod_{l \in \mathbf{u}} \theta_l \\ &= \theta_i \sum_{k=1}^{p} \Gamma_k \sum_{\mathbf{u} \subseteq [p] \setminus \{i\}, |\mathbf{u}|=k-1} \prod_{l \in \mathbf{u}} \theta_l = \theta_i \sum_{k=1}^{p} \Gamma_k r_{p,k-1}^{(-i)},\end{aligned}$$

where $r_{s,k}^{(-i)} = \sum_{\mathbf{u}\subseteq[s]\setminus\{i\},|\mathbf{u}|=k} \prod_{l\in\mathbf{u}} \theta_l$ for $s = 0, \cdots, p$. For $s > 0$, $s \neq i$, note that:

$$r_{s,k}^{(-i)} = \sum_{s\in\mathbf{u}\subseteq[s]\setminus\{i\},|\mathbf{u}|=k} \prod_{l\in\mathbf{u}} \theta_l + \sum_{s\notin\mathbf{u}\subseteq[s]\setminus\{i\},|\mathbf{u}|=k} \prod_{l\in\mathbf{u}} \theta_l = \theta_s r_{s-1,k-1}^{(-i)} + r_{s-1,k}^{(-i)},$$

with initial values $r_{s,0}^{(-i)} = 1$ and $r_{s,k}^{(-i)} = 0$ for $k > s$. This proves the correctedness of the recursive procedure.

APPENDIX C

APPENDIX FOR CHAPTER 4

C.1 Proof of Theorem 16

The proof of this requires a simple lemma on normal orthant probabilities:

Lemma 16. *[263] Let $(X_1, \cdots, X_p)$ follow the equicorrelated normal distribution, with $\mathbb{E}(X_j) = 0$, $\mathbb{E}(X_j^2) = 1$ and $\mathbb{E}(X_j X_k) = \rho$ for all $j \neq k$, and let $p_m = \mathbb{P}(X_1 > 0, \cdots, X_m > 0)$. Then:*

$$p_2 = \frac{\sin^{-1}\rho}{2\pi} + \frac{1}{4} \quad \textit{and} \quad p_3 = \frac{3\sin^{-1}\rho}{4\pi} + \frac{1}{8}.$$

For the main proof, note that each row of the latent matrix $\mathbf{Z}$ is i.i.d., so it suffices to fix $n = 1$ and explore the correlation amongst the scalar ME quantities $\tilde{x}_{1,A}$ and CME quantities $\tilde{x}_{1,A|B+}$. We denote these as $\tilde{x}_A$ and $\tilde{x}_{A|B+}$ for brevity. Under the latent equicorrelated distribution $\mathcal{N}\{\mathbf{0}, \rho\mathbf{J} + (1-\rho)\mathbf{I}\}$, it is easy to show that $\mathbb{E}[\tilde{x}_A] = 0$ and $\text{Var}[\tilde{x}_A] = 1$. Moreover, the CME $\tilde{x}_{A|B+}$ can be conditionally decomposed as $\tilde{x}_{A|B+} \overset{d}{=} R[2p_2]$ if $\tilde{x}_B = +1$, and 0 if $\tilde{x}_B = -1$, where $R[q]$ is the Rademacher random variable taking on +1 w.p. $q \in [0, 1]$ and -1 otherwise. From this, we get:

$$\mu_c \equiv \mathbb{E}[\tilde{x}_{A|B+}] = \mathbb{E}[\mathbb{E}[\tilde{x}_{A|B+}|\tilde{x}_B]] = \frac{1}{2}(4p_2 - 1),$$
$$\sigma_c^2 \equiv \text{Var}[\tilde{x}_{A|B+}] = \text{Var}[\mathbb{E}[\tilde{x}_{A|B+}|\tilde{x}_B]] + \mathbb{E}[\text{Var}[\tilde{x}_{A|B+}|\tilde{x}_B]] = \frac{1}{2} - \left(\frac{\sin^{-1}\rho}{\pi}\right)^2.$$

Consider the correlation between the MEs $\tilde{x}_A$ and $\tilde{x}_B$. Note that $\tilde{x}_A\tilde{x}_B$ equals +1 when $\tilde{x}_A$ and $\tilde{x}_B$ have the same sign, and equals -1 otherwise. Letting $\mathbb{P}(++)$ be the probability of $(\tilde{x}_A, \tilde{x}_B) = (+1, +1)$ (with similar notation for $+-$, $-+$ and $--$), Lemma 16 then

gives:

$$\text{Corr}(\tilde{x}_A, \tilde{x}_B) = [\mathbb{P}(++) + \mathbb{P}(++)] - [\mathbb{P}(+-) + \mathbb{P}(-+)] = 2p_2 - 2[1/2 - p_2] = \frac{2\sin^{-1}\rho}{\pi}.$$

Next, consider the two sibling CMEs $\tilde{x}_{A|B+}$ and $\tilde{x}_{A|C+}$. Note that $\tilde{x}_{A|B+}\tilde{x}_{A|C+}$ equals +1 when both $\tilde{x}_B = +1$ and $\tilde{x}_C = +1$, and equals 0 otherwise. It follows that:

$$\text{Corr}(\tilde{x}_{A|B+}, \tilde{x}_{A|C+}) = \frac{1}{\sigma_c^2}[\mathbb{P}(++) - \mu_c^2] = \frac{1}{\sigma_c^2}[p_2 - \mu_c^2] = \frac{1}{\sigma_c^2}\left\{-\left(\frac{\sin^{-1}\rho}{\pi}\right)^2 + \frac{\sin^{-1}\rho}{2\pi} + \frac{1}{4}\right\}.$$

The correlation for parent-child pairs can be proved in an analogous way.

Consider now the two cousin CMEs $\tilde{x}_{B|A+}$ and $\tilde{x}_{C|A+}$. Note that $\tilde{x}_{B|A+}\tilde{x}_{C|A+}$ equals +1 when $\tilde{x}_A = +1$ and $\tilde{x}_B = \tilde{x}_C$, $\tilde{x}_{B|A+}\tilde{x}_{C|A+}$ equals -1 when $\tilde{x}_A = +1$ and $\tilde{x}_B \neq \tilde{x}_C$, and equals 0 otherwise. We then have:

$$\begin{aligned}
\text{Corr}(\tilde{x}_{B|A+}, \tilde{x}_{C|A+}) &= \frac{1}{\sigma_c^2}\left[\{\mathbb{P}(+++) + \mathbb{P}(+--)\} - \{\mathbb{P}(++-) + \mathbb{P}(++-)\} - \mu_c^2\right] \\
&= \frac{1}{\sigma_c^2}\left[\{\mathbb{P}(+++) + (\mathbb{P}(--) - \mathbb{P}(---))\}\right. \\
&\quad \left. -2\{\mathbb{P}(++) - \mathbb{P}(+++)\} - \mu_c^2\right] \\
&= \frac{1}{\sigma_c^2}[2p_3 - p_2 - \mu_c^2] = \frac{1}{\sigma_c^2}\left\{-\left(\frac{\sin^{-1}\rho}{\pi}\right)^2 + \frac{\sin^{-1}\rho}{\pi}\right\}.
\end{aligned}$$

C.2 Proof of Theorem 17

Let $\mathbf{X} \in \mathbb{R}^{n \times p'}$ be the normalized model matrix consisting of all main effects and CMEs, where $p' = p + 4\binom{p}{2}$. By the strong law of large numbers, the sample covariance matrix $\mathbf{C}_n = \mathbf{X}^T\mathbf{X}/n$ converges elementwise to some matrix $\mathbf{C} \in \mathbb{R}^{p' \times p'}$ with unit diagonal entries and off-diagonal entries given in Theorem 16. Consider the following block partition of $\mathbf{C} = \begin{pmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{pmatrix}$, where $\mathbf{C}_{11}$ is the block for the active set $\mathcal{A}$, and $\mathbf{C}_{22}$ the block for the remaining variables. [102] proved that the LASSO is sign-selection consistent only when

the (weak) *irrepresentability condition* holds: $\forall \boldsymbol{\zeta} \in \{-1,+1\}^{p'},\ |\mathbf{C}_{21}\mathbf{C}_{11}^{-1}\boldsymbol{\zeta}| < \mathbf{1}$ (this is a slight simplification of the original condition under the current i.i.d. setting). Hence, sign-selection inconsistency can be proven if $\exists \boldsymbol{\zeta} \in \{-1,+1\}^{p'}$ and an inactive effect j satisfying:

$$|\mathbf{C}_{21,j}\mathbf{C}_{11}^{-1}\boldsymbol{\zeta}| \geq 1, \quad \text{where} \quad \mathbf{C}_{21,j} \text{ is the row corresponding to effect } j. \tag{C.1}$$

Consider first a model with only $q \geq 3$ active siblings of the form $A|B+$, $A|C-$, ..., $A|R-$. Using the same principles as in Theorem 16, $\mathbf{C}_{11}$ can be shown to be a $q \times q$ matrix with unit diagonal, $[(1/2-p_2)-\mu_c^2]/\sigma_c^2$ for off-diagonal entries in the first row and column, and $\psi_{sib}(\rho)$ for all other off-diagonal entries [1]. Letting A be the inactive effect, we have $\mathbf{C}_{21,A} = \psi_{pc}(\rho)\mathbf{1}_q^T$, and letting $\boldsymbol{\zeta} = \mathbf{1}_q$, it follows that $|\mathbf{C}_{21,A}\mathbf{C}_{11}^{-1}\boldsymbol{\zeta}| \geq 1$ for $\rho \geq 0$. By (C.1), part (a) is proven.

Next, consider a model with only $q = 2$ active main effects, say, A and $-B$. From Theorem 16, $\mathbf{C}_{11}$ is a $q \times q$ matrix with unit diagonal and $-\psi_{me}(\rho)$ on the off-diagonals. Let $A|B-$ be the inactive effect, so $\mathbf{C}_{21,A|B-} = (\psi_{pc}(\rho), \tilde{\psi}(\rho))$. Taking $\boldsymbol{\zeta} = (1,1)^T$, $|\mathbf{C}_{21,A|B-}\mathbf{C}_{11}^{-1}\boldsymbol{\zeta}| \geq 1$ for $\rho \geq 0.27$, thereby proving selection inconsistency.

Lastly, consider a model with only $q \geq 6$ active cousins of the form $B|A+$, $C|A-$, ..., $R|A-$. Using the same principles as in Theorem 16, $\mathbf{C}_{11}$ is a $q \times q$ matrix with unit diagonal, $-\mu_c^2/\sigma_c^2$ for the off-diagonal entries in the first row and column, and $\psi_{cou}(\rho)$ for all other off-diagonal entries. Let B be the inactive effect with $\mathbf{C}_{21,B} = (\psi_{sib}(\rho), \tilde{\psi}(\rho)\mathbf{1}_{q-1})$. Taking $\boldsymbol{\zeta} = \mathbf{1}_q$, $|\mathbf{C}_{21,B}\mathbf{C}_{11}^{-1}\boldsymbol{\zeta}| \geq 1$ for $\rho \geq 0.29$, which proves inconsistency.

[1] $\psi_{me}(\rho)$, $\psi_{sib}(\rho)$, $\psi_{pc}(\rho)$ and $\psi_{cou}(\rho)$ are the pairwise correlations in Theorem 16 for main effects, siblings, parent-child pairs and cousins, respectively. $\tilde{\psi}(\rho) = \sin^{-1}(\rho)/(\pi\sigma_c)$ is the pairwise correlation between a CME and its conditioned effect.

C.3 Proof of Proposition 3

As a note, since the objective $Q(\boldsymbol{\beta})$ is non-differentiable at $\boldsymbol{\beta} = \mathbf{0}$, what we mean by strict convexity here is that $\nabla^2_{\mathbf{u}} Q(\boldsymbol{\beta})$, the directional Hessian of $Q(\boldsymbol{\beta})$ in direction $\mathbf{u}$, is positive-definite for all $\boldsymbol{\beta}$ and all $\|\mathbf{u}\| = 1$. We follow a similar approach as Proposition 1 of [101]. Note that $\nabla^2 \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = 2\mathbf{X}^T\mathbf{X}$. Moreover, with $\eta'_{\lambda,\tau}(\theta) = \lambda \exp(-\theta\tau/\lambda)$ and $\eta''_{\lambda,\tau}(\theta) = -\tau \exp(-\theta\tau/\lambda)$, one can show that $\nabla^2_{\mathbf{u}} P_s(\boldsymbol{\beta}) \geq -\tau(1) + \lambda(-1/(\lambda\gamma)) = -\tau - 1/\gamma$ and similarly $\nabla^2_{\mathbf{u}} P_c(\boldsymbol{\beta}) \geq -\tau - 1/\gamma$, for all $\mathbf{u}$ and $\boldsymbol{\beta}$. Hence, for all $\mathbf{u}$ and $\boldsymbol{\beta}$:

$$\nabla^2_{\mathbf{u}} Q(\boldsymbol{\beta}) = \nabla^2_{\mathbf{u}} \left\{ \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + P_s(\boldsymbol{\beta}) + P_c(\boldsymbol{\beta}) \right\} \geq \frac{\lambda_{min}(\mathbf{X}^T\mathbf{X})}{n} - 2\left(\tau + \frac{1}{\gamma}\right),$$

which is strictly positive when $\tau + 1/\gamma < \lambda_{min}(\mathbf{X}^T\mathbf{X})/(2n)$. The second part of the claim follows by replacing $\mathbf{X}$ with $\mathbf{x}_j$ in the argument above, and using the fact that $\|\mathbf{x}_j\|_2^2 = n$.

C.4 Proof of Theorem 18 and Corollary 2

The majorization claim $a)$ follows from a first-order Taylor expansion of the outer penalty: $\eta_{\lambda,\tau}(\|\boldsymbol{\beta}_g\|_{\lambda,\gamma}) \geq \eta_{\lambda,\tau}(\|\tilde{\boldsymbol{\beta}}_g\|_{\lambda,\gamma}) + \tilde{\Delta}_g \left\{ \|\boldsymbol{\beta}_g\|_{\lambda,\gamma} - \|\tilde{\boldsymbol{\beta}}_g\|_{\lambda,\gamma} \right\}$, where the inequality holds due to the concavity of η. See Lemma 1 in [101] for details.

To derive the threshold function in $b)$, take the following optimization problem:

$$\hat{\beta}_j = \underset{\beta_j}{\operatorname{argmin}} \left\{ \frac{1}{2n} \|\mathbf{r} - \mathbf{x}_j\beta_j\|_2^2 + \Delta_1 g_{\lambda_1,\gamma}(\beta_j) + \Delta_2 g_{\lambda_2,\gamma}(\beta_j) \right\}. \tag{C.2}$$

The KKT condition for (C.2) is:

$$0 \in -\frac{1}{n}\mathbf{x}_j^T\mathbf{r} + \hat{\beta}_j + \Delta_1 \partial_{\lambda_1,\gamma}\hat{\beta}_j + \Delta_2 \partial_{\lambda_2,\gamma}\hat{\beta}_j, \quad \partial_{\lambda,\gamma}\beta_j = \begin{cases} \operatorname{sgn}(\beta_j)\left(1 - \dfrac{|\beta_j|}{\lambda\gamma}\right)_+ & \text{if } |\beta_j| > 0, \\ [-1, 1] & \text{if } \beta_j = 0. \end{cases} \tag{C.3}$$

Without loss of generality, assume $z \equiv \mathbf{x}_j^T \mathbf{r}/n > 0$. Consider the same four cases for z as presented in (3.9):

1. $z \geq \lambda_{(1)}\gamma$: Suppose $\hat{\beta}_j = z$. Then the KKT condition (C.3) becomes $0 \in -z + \hat{\beta}_j$, which is satisfied. Since (C.2) is strictly convex, $\hat{\beta}_j = z$ must be its unique solution.

2. $c_2 \leq z < \lambda_{(1)}\gamma$ (see (3.9) for c_2): Suppose $\hat{\beta}_j = (z - \Delta_{(1)})/\left(1 - \frac{\Delta_{(1)}}{\lambda_{(1)}\gamma}\right)$. Since $\lambda_{(2)}\gamma \leq \hat{\beta}_j < \lambda_{(1)}\gamma$, the KKT condition (C.3) becomes $0 \in -z + \hat{\beta}_j + \Delta_{(1)}\left(1 - \frac{\hat{\beta}_j}{\lambda_{(1)}\gamma}\right)$, which is satisfied. Hence, $\hat{\beta}_j$ is the unique solution to (C.2).

3. $\Delta_{(1)} + \Delta_{(2)} \leq z < c_2$ (see (3.9) for c_3): Suppose $\hat{\beta}_j = (z - \Delta_{(1)} - \Delta_{(2)})/\left(1 - \frac{\Delta_{(1)}}{\lambda_{(1)}\gamma} - \frac{\Delta_{(2)}}{\lambda_{(2)}\gamma}\right)$. Since $0 < \hat{\beta}_j < \lambda_{(2)}\gamma$, the KKT condition (C.3) becomes $0 \in -z + \hat{\beta}_j + \Delta_{(1)}\left(1 - \frac{\hat{\beta}_j}{\lambda_{(1)}\gamma}\right) + \Delta_{(2)}\left(1 - \frac{\hat{\beta}_j}{\lambda_{(2)}\gamma}\right)$, which is satisfied. Hence, $\hat{\beta}_j$ is the unique solution to (C.2).

4. $0 \leq z < \Delta_{(1)} + \Delta_{(2)}$: Suppose $\hat{\beta}_j = 0$. The KKT condition then becomes $0 \in -z + (\Delta_{(1)} + \Delta_{(2)})[-1, 1]$, which is satisfied, so $\hat{\beta}_j$ is the unique solution to (C.2).

From this, Corollary 2 can be proved in a similar way as Proposition 3 of [101].

C.5 Proof of Proposition 4

Since $Q(\boldsymbol{\beta})$ is strictly convex, it must have at most one minimizer $\boldsymbol{\beta}$. By definition, $\boldsymbol{\beta}$ must satisfy the KKT condition:

$$0 \in -\frac{1}{n}\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \Delta_{\mathcal{S}}(\boldsymbol{\beta})\partial_{\lambda_s,\gamma}\beta_j + \Delta_{\mathscr{C}}(\boldsymbol{\beta})\partial_{\lambda_c,\gamma}\beta_j, \quad j = 1, \cdots, p', \tag{C.4}$$

where $\partial_{\lambda,\gamma}\beta_j$ is the subgradient defined in (C.3), and $\Delta_{\mathcal{S}}(\boldsymbol{\beta})$ and $\Delta_{\mathscr{C}}(\boldsymbol{\beta})$ are the linearized slopes in (3.5) for the sibling and cousin groups of effect j. Setting $\beta = \mathbf{0}$, the right side of (C.4) becomes:

$$-\frac{1}{n}\mathbf{x}_j^T\mathbf{y} + \lambda_s[-1, 1] + \lambda_c[-1, 1] = -\frac{1}{n}\mathbf{x}_j^T\mathbf{y} + [-\lambda_s - \lambda_c, \lambda_s + \lambda_c],$$

which contains 0 when $\lambda_s + \lambda_c \geq |\mathbf{x}_j^T \mathbf{y}|/n$. Hence, when $\lambda_s + \lambda_c \geq \max_{j=1,\cdots,p'} |\mathbf{x}_j^T \mathbf{y}|/n$, one can invoke the strict convexity of $Q(\boldsymbol{\beta})$ to show that the trivial solution $\boldsymbol{\beta} = \mathbf{0}$ is indeed the unique minimizer.

C.6 Algorithm statement for `cv.cmenet`

Algorithm 13 `cv.cmenet`: A cross-validation algorithm for tuning `cmenet`

function CV.CMENET($\mathbf{X}, \mathbf{y}, K$)

- Initialize grid of potential parameters $\max_{j=1,\cdots,p'} |\mathbf{x}_j^T \mathbf{y}|/n > \lambda_s^1 > \cdots > \lambda_s^L > 0$, $\max_{j=1,\cdots,p'} |\mathbf{x}_j^T \mathbf{y}|/n > \lambda_c^1 > \cdots > \lambda_c^M > 0$, $\gamma^1 < \cdots < \gamma^G$ and $\tau^1 < \cdots < \tau^T$ (satisfying $\tau + 1/\gamma < 1/2$).
- Obtain the tuned MC+ parameters (λ^*, γ^*) using `cv.sparsenet` in the R package SPARSENET, and set $\lambda_s^*, \lambda_c^* \leftarrow \lambda^*/2$ as an initial estimate.
- Randomly partition the data $\mathcal{D} = (\mathbf{X}, y)$ into K equal pieces $\{\mathcal{D}_1, \cdots, \mathcal{D}_K\}$.

for $k = 1, \cdots, K$ **do** ▷ K-fold CV for tuning γ and τ

  for $\gamma \in \{\gamma_1, \cdots, \gamma_G\}$ **do** ▷ For each γ...

- $\boldsymbol{\beta}_{prev} \leftarrow \mathbf{0}_{p'}$ ▷ Reset warm start solution

    for $\tau \in \{\tau_1, \cdots, \tau_T\}$ **do** ▷ For each τ...

- $\boldsymbol{\beta}_{\lambda_s^*, \lambda_c^*}(\gamma, \tau; k) \leftarrow \texttt{cmenet}(\mathbf{X}_{-k}, \mathbf{y}_{-k}, \lambda_s^*, \lambda_c^*, \gamma, \tau, \boldsymbol{\beta}_{prev})$ ▷ Train w/o part k
- $\boldsymbol{\beta}_{prev} \leftarrow \boldsymbol{\beta}_{\lambda_s^*, \lambda_c^*}(\gamma, \tau; k)$ ▷ Update warm start solution
- $(\gamma^*, \tau^*) \leftarrow \underset{\gamma,\tau}{\operatorname{argmin}} \sum_{k=1}^{K} \|\mathbf{y}_k - \mathbf{X}_k \boldsymbol{\beta}_{\lambda_s^*, \lambda_c^*}(\gamma, \tau; k)\|_2^2$ ▷ Estimate optimal γ and τ

for $k = 1, \cdots, K$ **do** ▷ K-fold CV for tuning λ_s and λ_c

  for $\lambda_c \in \{\lambda_c^1, \cdots, \lambda_c^M\}$ **do** ▷ For each λ_c...

- $\boldsymbol{\beta}_{prev} \leftarrow \mathbf{0}_{p'}$

    for $\lambda_s \in \{\lambda_s^1, \cdots, \lambda_s^L\}$ **do** ▷ For each λ_s...

      if $\lambda_c + \lambda_s < \max_{j=1,\cdots,p'} |\mathbf{x}_j^T \mathbf{y}|/n$ **then**

- Screen using the three strong rules in Section C.7.
- $\boldsymbol{\beta}_{\lambda_s, \lambda_c}(\gamma^*, \tau^*; k) \leftarrow \texttt{cmenet}(\mathbf{X}_{-k}, \mathbf{y}_{-k}, \lambda_s, \lambda_c, \gamma^*, \tau^*, \boldsymbol{\beta}_{prev})$, using only screened effects.
- Check KKT conditions on converged solution $\boldsymbol{\beta}_{\lambda_s, \lambda_c}(\gamma^*, \tau^*; k)$.
- $\boldsymbol{\beta}_{prev} \leftarrow \boldsymbol{\beta}_{\lambda_s, \lambda_c}(\gamma^*, \tau^*; k)$
- $(\lambda_s^*, \lambda_c^*) \leftarrow \underset{\lambda_s,\lambda_c}{\operatorname{argmin}} \sum_{k=1}^{K} \|\mathbf{y}_k - \mathbf{X}_k \boldsymbol{\beta}_{\lambda_s, \lambda_c}(\gamma^*, \tau^*; k)\|_2^2$ ▷ Estimate optimal λ_s and λ_c
- $\hat{\boldsymbol{\beta}} \leftarrow \texttt{cmenet}(\mathbf{X}, \mathbf{y}, \lambda_s^*, \lambda_c^*, \gamma^*, \tau^*, \mathbf{0}_{p'})$ ▷ Refit using optimal parameters

return optimal coefficients $\hat{\boldsymbol{\beta}}$.

Some comments on the implementation of active set optimization within `cmenet`:

- The active set of variables is initialized by performing the full coordinate descent cycle

for 25 iterations, then choosing the variables whose coefficients are non-zero.

- Repeat coordinate descent iterations over the active set until convergence.
- Perform a full coordinate descent cycle over all p' variables. If this cycle does not change the active set, `cmenet` is terminated; otherwise, the active set is updated, and the above steps repeated.

C.7 Theoretical derivation of CME screening rules

Fix γ and τ, and suppose $\hat{\beta}_j(\lambda_s, \lambda_c) \in (0, \min\{\Delta_{(1)} + \Delta_{(2)}, \lambda_{(2)}\gamma\})$. For brevity, we denote $\hat{\beta}_j(\lambda_s, \lambda_c)$ as $\hat{\beta}_j$ from here on. Using equation (3.9), we know that $\hat{\beta}_j$ takes the form:

$$\begin{aligned}\hat{\beta}_j &= \text{sgn}(z_j)\left(|z_j| - \Delta_{(1)} - \Delta_{(2)}\right)_+ \Big/ \left(1 - \frac{\Delta_{(1)}}{\lambda_{(1)}\gamma} - \frac{\Delta_{(2)}}{\lambda_{(2)}\gamma}\right) \\ &= \text{sgn}(z_j)\left(|z_j| - \Delta_S - \Delta_C\right)_+ \Big/ \left(1 - \frac{\Delta_S}{\lambda_S\gamma} - \frac{\Delta_C}{\lambda_C\gamma}\right),\end{aligned} \tag{C.5}$$

where $z_j = \mathbf{x}_j^T \mathbf{r}_{-j}/n$ (see Theorem 18), and $\Delta_{\mathcal{S}}$ and $\Delta_{\mathscr{C}}$ are the linearized slopes for the current penalty setting (λ_s, λ_c). Plugging this expression into (C.4), the KKT condition for $\hat{\beta}_j$ can be simplified to:

$$\begin{aligned}0 &= -c_j(\lambda_s, \lambda_c) + \text{sgn}(\hat{\beta}_j)\Delta_{\mathcal{S}}\left\{1 - \frac{(|z_j| - \Delta_{\mathcal{S}} - \Delta_{\mathscr{C}})_+}{\lambda_s\left(\gamma - \frac{\Delta_{\mathcal{S}}}{\lambda_s} - \frac{\Delta_{\mathscr{C}}}{\lambda_c}\right)}\right\} + \text{sgn}(\hat{\beta}_j)\Delta_{\mathscr{C}}\left\{1 - \frac{(|z_j| - \Delta_{\mathcal{S}} - \Delta_{\mathscr{C}})_+}{\lambda_c\left(\gamma - \frac{\Delta_{\mathcal{S}}}{\lambda_s} - \frac{\Delta_{\mathscr{C}}}{\lambda_c}\right)}\right\} \\ \Leftrightarrow\ c_j(\lambda_s, \lambda_c) &= \text{sgn}(\hat{\beta}_j)\Delta_{\mathcal{S}}\left\{1 - \frac{(|z_j| - \Delta_{\mathcal{S}} - \Delta_{\mathscr{C}})_+}{\lambda_s\left(\gamma - \frac{\Delta_{\mathcal{S}}}{\lambda_s} - \frac{\Delta_{\mathscr{C}}}{\lambda_c}\right)}\right\} + \text{sgn}(\hat{\beta}_j)\Delta_{\mathscr{C}}\left\{1 - \frac{(|z_j| - \Delta_{\mathcal{S}} - \Delta_{\mathscr{C}})_+}{\lambda_c\left(\gamma - \frac{\Delta_{\mathcal{S}}}{\lambda_s} - \frac{\Delta_{\mathscr{C}}}{\lambda_c}\right)}\right\}.\end{aligned} \tag{C.6}$$

Suppose no effects are active in either the sibling group $\mathcal{S}$ or the cousin group $\mathscr{C}$, in which case $\Delta_{\mathcal{S}} = \lambda_s$ and $\Delta_{\mathscr{C}} = \lambda_c$. The KKT condition in (C.6) can then be rewritten as:

$$c_j(\lambda_s, \lambda_c) = \text{sgn}(\hat{\beta}_j)\left\{\lambda_s - \frac{(|z_j| - \lambda_s - \lambda_c)_+}{\gamma - 2}\right\} + \text{sgn}(\hat{\beta}_j)\left\{\lambda_c - \frac{(|z_j| - \lambda_s - \lambda_c)_+}{\gamma - 2}\right\}. \tag{C.7}$$

Taking the derivative with respect to λ_s (and assuming z_j is approximately constant in λ_s, following [112]), we get:

$$\Big|\frac{\partial}{\partial\lambda_s}c_j(\lambda_s,\lambda_c)\Big| \lesssim 1+\frac{1}{\gamma-2}+\frac{1}{\gamma-2}=\frac{\gamma}{\gamma-2}. \tag{C.8}$$

A similar argument shows that this approximate upper bound also holds for $|(\partial/\partial\lambda_c)\ c_j(\lambda_s,\lambda_c)|$.

Now, suppose no effects are active in the sibling group $\mathcal{S}$ (but some in the cousin group $\mathcal{C}$), in which case $\Delta_{\mathcal{S}}=\lambda_s$. The KKT condition in (C.6) can then be rewritten as:

$$c_j(\lambda_s,\lambda_c)=\text{sgn}(\hat{\beta}_j)\left\{\lambda_s-\frac{(|z_j|-\lambda_s-\Delta_{\mathcal{C}})_+}{\gamma-1-\frac{\Delta_{\mathcal{C}}}{\lambda_c}}\right\}+\text{sgn}(\hat{\beta}_j)\Delta_{\mathcal{C}}\left\{1-\frac{(|z_j|-\lambda_s-\Delta_{\mathcal{C}})_+}{\lambda_c\left(\gamma-1-\frac{\Delta_{\mathcal{C}}}{\lambda_c}\right)}\right\}. \tag{C.9}$$

Taking the derivative on λ_s (and assuming z_j is approximately constant in λ_s), we get:

$$\Big|\frac{\partial}{\partial\lambda_s}c_j(\lambda_s,\lambda_c)\Big| \lesssim 1+\frac{1}{\gamma-1-\frac{\Delta_{\mathcal{C}}}{\lambda_c}}+\frac{\frac{\Delta_{\mathcal{C}}}{\lambda_c}}{\gamma-1-\frac{\Delta_{\mathcal{C}}}{\lambda_c}}=\frac{\gamma}{\gamma-1-\frac{\Delta_{\mathcal{C}}}{\lambda_c}}. \tag{C.10}$$

Finally, suppose there are no active effects in the cousin group $\mathcal{C}$ (but some in sibling group $\mathcal{S}$). One can do a similar approximation and show that:

$$\Big|\frac{\partial}{\partial\lambda_c}c_j(\lambda_s,\lambda_c)\Big| \lesssim 1+\frac{1}{\gamma-\frac{\Delta_{\mathcal{S}}}{\lambda_s}-1}+\frac{\frac{\Delta_{\mathcal{S}}}{\lambda_s}}{\gamma-\frac{\Delta_{\mathcal{S}}}{\lambda_s}-1}=\frac{\gamma}{\gamma-\frac{\Delta_{\mathcal{S}}}{\lambda_s}-1}. \tag{C.11}$$

These upper bounds on the absolute derivatives of $c_j(\lambda_s,\lambda_c)$, along with the proposed strong rules in Section C.7, can then be used to demonstrate the inactivity of effect j at penalty setting $(\lambda_s^l,\lambda_c^m)$:

1. Consider the first part of the first strong rule, which applies when no active effects are in $\mathcal{S}$ and $\mathcal{C}$ for setting $(\lambda_s^{l-1},\lambda_c^m)$. This rule discards effect j at setting $(\lambda_s^l,\lambda_c^m)$ if:

$$|c_j(\lambda_s^{l-1},\lambda_c^m)|<\lambda_s^l+\lambda_c^m+\frac{\gamma}{\gamma-2}(\lambda_s^l-\lambda_s^{l-1}).$$

   This can be justified as follows. Using the approximate upper bound in (C.8), the

inner-product of effect j at setting $(\lambda_s^l, \lambda_c^m)$ can be approximately upper bounded as:

$$\begin{aligned}
|c_j(\lambda_s^l, \lambda_c^m)| &\leq |c_j(\lambda_s^l, \lambda_c^m) - c_j(\lambda_s^{l-1}, \lambda_c^m)| + |c_j(\lambda_s^{l-1}, \lambda_c^m)| \\
&\approx \Big|\frac{\partial}{\partial \lambda_s} c_j(\lambda_s^{l-1}, \lambda_c^m)\Big|(\lambda_s^{l-1} - \lambda_s^l) + |c_j(\lambda_s^{l-1}, \lambda_c^m)| \\
&< \frac{\gamma}{\gamma - 2}(\lambda_s^{l-1} - \lambda_s^l) + \left[\lambda_s^l + \lambda_c^m + \frac{\gamma}{\gamma - 2}(\lambda_s^l - \lambda_s^{l-1})\right] \\
&= \lambda_s^l + \lambda_c^m.
\end{aligned}$$

Assuming effect j is the first variable to potentially be selected in $\mathcal{S}$ or $\mathcal{C}$ at current setting $(\lambda_s^l, \lambda_c^m)$, the KKT conditions in (C.4) suggest that effect j is inactive, which justifies the screening rule. A similar argument can be used to derive the second part of this rule.

2. Consider next the second strong rule, which applies when no active effects are in $\mathcal{S}$ for setting $(\lambda_s^{l-1}, \lambda_c^m)$. This rule discards effect j at setting $(\lambda_s^l, \lambda_c^m)$ if:

$$|c_j(\lambda_s^{l-1}, \lambda_c^m)| < \lambda_s^l + \Delta'_{\mathcal{C}} + \frac{\gamma}{\gamma - (\Delta'_{\mathcal{C}}/\lambda_c^m + 1)}(\lambda_s^l - \lambda_s^{l-1}).$$

This can be justified as follows. Using the approximate upper bound in (C.10), the inner-product of effect j at setting $(\lambda_s^l, \lambda_c^m)$ can be approximately upper bounded as:

$$\begin{aligned}
|c_j(\lambda_s^l, \lambda_c^m)| &\leq |c_j(\lambda_s^l, \lambda_c^m) - c_j(\lambda_s^{l-1}, \lambda_c^m)| + |c_j(\lambda_s^{l-1}, \lambda_c^m)| \\
&\approx \Big|\frac{\partial}{\partial \lambda_s} c_j(\lambda_s^{l-1}, \lambda_c^m)\Big|(\lambda_s^{l-1} - \lambda_s^l) + |c_j(\lambda_s^{l-1}, \lambda_c^m)| \\
&< \frac{\gamma}{\gamma - (\Delta'_{\mathcal{C}}/\lambda_c^m + 1)}(\lambda_s^{l-1} - \lambda_s^l) \\
&\quad + \left[\lambda_s^l + \Delta'_{\mathcal{C}} + \frac{\gamma}{\gamma - (\Delta'_{\mathcal{C}}/\lambda_c^m + 1)}(\lambda_s^l - \lambda_s^{l-1})\right] \\
&= \lambda_s^l + \Delta'_{\mathcal{C}}.
\end{aligned}$$

Assuming:

- Effect j is the first variable to potentially be selected in $\mathcal{S}$ at current setting $(\lambda_s^l, \lambda_c^m)$,
- The linearized slope $\Delta'_{\mathscr{C}}$ at previous setting $(\lambda_s^{l-1}, \lambda_c^m)$ is approximately the linearized slope $\Delta_{\mathscr{C}}$ at current setting $(\lambda_s^l, \lambda_c^m)$,

the KKT conditions in (C.4) suggest that effect j is inactive, which justifies the screening rule.

3. The third strong rule can be justified in a similar manner to the above two rules.

APPENDIX D

APPENDIX FOR CHAPTER 5

D.1 Computing the CPOD expansion

The driving idea behind CPOD is that a common spatial domain is needed to extract common instabilities over multiple injector geometries, since each simulation run has different geometries and varying grid points. We first describe a physically justifiable method for obtaining such a common domain, and then use this to compute the CPOD expansion.

D.1.1 Common grid

1. Identify the densest grid (i.e., with the most grid points) among the n simulation runs, and set this as the common reference grid.

2. For each simulation, partition the grid into the following four parts: (a) from injector head-end to the inlet, (b) from the inlet to the nozzle exit, (c) the top portion of the downstream region and (d) the bottom portion of the downstream region (see Figure D.1 for an illustration). This splits the flow in such a way that the linearity assumption can be physically justified.

3. Linearly rescale each part of the partition to the common grid by the corresponding geometry parameters L, R_n and ΔL (see Figure D.1).

4. For each simulation, interpolate the original flow data onto the spatial grid of the common geometry. This step ensures the flow is realized over a common set of grid points for all n simulations. In our implementation, the *inverse distance weighting* interpolation method [147] is used with 10 nearest neighbours.

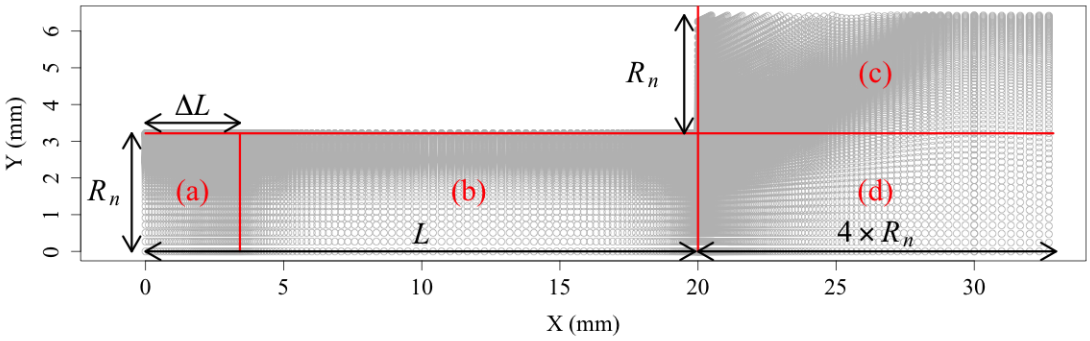

Figure D.1: Partition of the spatial grid for the first simulation case.

D.1.2 POD expansion

After flows from each simulation have been rescaled onto the common grid, the original POD expansion can be used to extract common flow instabilities. Let $\{\mathbf{x}_j\}_{j=1}^{J}$ and $\{t_m\}_{m=1}^{T}$ denote the set of common grid points and simulated time-steps, respectively, and let $\tilde{Y}(\mathbf{x}, t; \mathbf{c}_i)$ be an interpolated flow variable for geometric setting $\mathbf{c}_i$, $i = 1, \cdots, n$ (for brevity, assume a single flow variable, e.g., x-velocity, for the exposition below). The CPOD expansion can be computed using the following three steps.

1. For notational convenience, we combine all combinations of geometries and time-steps into a single index. Set $N = nT$ and let $l = 1, \cdots, N$ index all combinations of n design settings and T time-steps, and let $\tilde{Y}_l(\mathbf{x}) \equiv \tilde{Y}(\mathbf{x}, (t, \mathbf{c})_l)$. Define $\mathbf{Q} \in \mathbb{R}^{N \times N}$ as the following inner-product matrix:

$$\mathbf{Q}_{l,m} = \sum_{j=1}^{J} \tilde{Y}_l(\mathbf{x}_j)\tilde{Y}_m(\mathbf{x}_j).$$

   Such an inner-product is possible because all n simulated flows are observed on a set of *common* gridpoints set.

   First, compute the eigenvectors $\mathbf{a}_k \in \mathbb{R}^N$ satisfying:

$$\mathbf{Q}\mathbf{a}_k = \lambda_k \mathbf{a}_k,$$

where λ_k is the k-th largest eigenvalue of $\mathbf{Q}$. Since a full eigendecomposition requires $O(N^3)$ work, this step may be intractible to perform when the temporal resolution is dense. To this end, we employed a variant of the implicitly restarted Arnoldi method [148], which can efficiently approximate leading eigenvalues and eigenvectors.

2. Compute the k-th mode $\phi_k(\mathbf{x})$ as:

$$\begin{bmatrix} \phi_k(\mathbf{x}_1) \\ \phi_k(\mathbf{x}_2) \\ \vdots \\ \phi_k(\mathbf{x}_J) \end{bmatrix} = \begin{pmatrix} \tilde{Y}_1(\mathbf{x}_1) & \cdots & \tilde{Y}_N(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \tilde{Y}_1(\mathbf{x}_J) & \cdots & \tilde{Y}_N(\mathbf{x}_J) \end{pmatrix} \mathbf{a}_k.$$

To ensure orthonormality, apply the following normalization:

$$\phi_k(\mathbf{x}_j) := \frac{\phi_k(\mathbf{x}_j)}{\|\phi_k(\mathbf{x})\|}, \quad \|\phi_k(\mathbf{x})\| = \sqrt{\sum_{j=1}^{J} \phi_k(\mathbf{x}_j)^2}$$

3. Lastly, derive the CPOD coefficients $(\beta_{l,1}, \cdots, \beta_{l,N})^T$ for the snapshot at index l (i.e., with design setting and time-step $(\mathbf{c}, t)_l$) as:

$$\begin{bmatrix} \beta_{l,1} \\ \beta_{l,2} \\ \vdots \\ \beta_{l,N} \end{bmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \cdots & \phi_1(\mathbf{x}_J) \\ \vdots & \ddots & \vdots \\ \phi_N(\mathbf{x}_1) & \cdots & \phi_N(\mathbf{x}_J) \end{pmatrix} \begin{bmatrix} \tilde{Y}_l(\mathbf{x}_1) \\ \tilde{Y}_l(\mathbf{x}_2) \\ \vdots \\ \tilde{Y}_l(\mathbf{x}_J) \end{bmatrix}.$$

Using these coefficients and a truncation at $K_r < N$ modes, it is easy to show the following decomposition of the flow at the design setting $\mathbf{c}_i$ and time-step t_m indexed by l:

$$Y(\mathbf{x}_j, t_m; \mathbf{c}_i) \approx \sum_{k=1}^{K_r} \beta_{l,k} \mathscr{M}_i\{\phi_k(\mathbf{x}_j)\}, \quad j = 1, \cdots, J,$$

as asserted in (3).

D.2 Proof of Theorem 2

Define the map $A : \mathbb{R}^K \times \mathbb{R}^{K\times K} \times \mathbb{R}^p \to \mathbb{R}^K \times \mathbb{R}^{K\times K} \times \mathbb{R}^p$ as a single-loop of the graphical LASSO operator for optimizing $\mathbf{T}$ with $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$ fixed, and define $B : \mathbb{R}^K \times \mathbb{R}^{K\times K} \times \mathbb{R}^p \to \mathbb{R}^K \times \mathbb{R}^{K\times K} \times \mathbb{R}^p$ as the L-BFGS map for a single line-search when optimizing $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$ with $\mathbf{T}$ fixed. Each BCD cycle in Algorithm 1 then follows the map composition $S = A^M \circ B^N$, where $M < \infty$ and $N < \infty$ are the iteration count for the graphical LASSO operator and number of line-searches, respectively. The parameter estimates at iteration m of the BCD cycle can then be given by:

$$\Theta_{m+1} = S(\Theta_m), \quad \text{where } \Theta_m = (\boldsymbol{\mu}_m, \mathbf{T}_m, \boldsymbol{\tau}_m).$$

Define the set of stationary solutions as $\Gamma = \{\Theta \ : \ \nabla l_\lambda(\Theta) = \mathbf{0}\}$, where ∇l_λ is the gradient of the negative log-likelihood l_λ. Using the Global Convergence Theorem (see Section 7.7 of [264]), we can prove stationary convergence:

$$\lim_{m\to\infty} \Theta_m = \Theta^* \in \Gamma,$$

if the following three conditions hold:

(i) $\{\Theta_m\}_{m=1}^{\infty}$ is contained within a compact subset of $\mathbb{R}^K \times \mathbb{R}^{K\times K} \times \mathbb{R}^p$,

(ii) l_λ is a continuous descent function on Γ under map S,

(iii) S is closed for points outside of Γ.

We will verify these conditions below.

(i) This is easily verified by the fact that $|\boldsymbol{\mu}_m| \leq \left(\max_{i,r,k} |\beta_k^{(r)}(\mathbf{c}_i)|\right) \mathbf{1}_K$, $\mathbf{0} \preceq \mathbf{T}_m \preceq$

$\left(\max_{k,r} s^2\{\beta_k^{(r)}(\mathbf{c}_i)\}_{i=1}^n\right) \mathbf{I}_K$ and $\boldsymbol{\tau}_m \in [0,1]^p$, where $s^2\{\cdot\}$ returns the sample standard deviation for a set of scalars.

(ii) To prove that S is a descent function, we need to show that if $\Theta \in \Gamma$, then $l_\lambda\{S(\Theta)\} = l_\lambda\{\Theta\}$, and if $\Theta \notin \Gamma$, then $l_\lambda\{S(\Theta)\} < l_\lambda\{\Theta\}$. The first condition is trivial, since $M = 0$ and $N = 0$ when Θ is stationary. The second condition follows from the fact that the maps A and B incur a strict decrease in l_λ whenever $\mathbf{T}$ and $(\boldsymbol{\mu}, \boldsymbol{\tau})$ are non-stationary, respectively.

(iii) Note that A^M is a continuous map (since the graphical LASSO map is a continuous operator) and the line-search map B^N is also continuous. Since $S = A^M \circ B^N$, it must be continuous as well, from which the closedness of S follows.

D.3 Proof of Theorem 3

Fix some spatial coordinate $\mathbf{x}$ and time-step t, and let:

$$\mathbf{y} = (Y^{(u)}(\mathbf{x}, t; \mathbf{c}_{new}), Y^{(v)}(\mathbf{x}, t; \mathbf{c}_{new}), Y^{(w)}(\mathbf{x}, t; \mathbf{c}_{new}))^T$$

be the true simulated flows for x-, y- and circumferential velocities at the new setting $\mathbf{c}_{new}$,

$$\hat{\mathbf{y}} = (\hat{Y}^{(u)}(\mathbf{x}, t; \mathbf{c}_{new}), \hat{Y}^{(v)}(\mathbf{x}, t; \mathbf{c}_{new}), \hat{Y}^{(w)}(\mathbf{x}, t; \mathbf{c}_{new}))^T$$

be its corresponding prediction from (9), and

$$\bar{\mathbf{y}} = (\bar{Y}^{(u)}(\mathbf{x}; \mathbf{c}_{new}), \bar{Y}^{(v)}(\mathbf{x}; \mathbf{c}_{new}), \bar{Y}^{(w)}(\mathbf{x}; \mathbf{c}_{new}))^T$$

be its time-averaged flow. It is easy to verify that, given the simulation data $\mathscr{D} = \{Y^{(r)}(\mathbf{x}, t; \mathbf{c}_i)\}$, the conditional distribution of $\mathbf{y}|\mathscr{D}$ is $\mathcal{N}(\hat{\mathbf{y}}, \Phi(\mathbf{x}, t))$, where:

$$\Phi(\mathbf{x}, t) \equiv \begin{bmatrix} \mathbf{m}^{(u)} & 0 & 0 \\ 0 & \mathbf{m}^{(v)} & 0 \\ 0 & 0 & \mathbf{m}^{(w)} \end{bmatrix} [\mathbb{V}\{\boldsymbol{\beta}(t; \mathbf{c}_{new})|\{\boldsymbol{\beta}(t; \mathbf{c}_i)\}_{i=1}^n\}]_{uvw} \begin{bmatrix} \mathbf{m}^{(u)} & 0 & 0 \\ 0 & \mathbf{m}^{(v)} & 0 \\ 0 & 0 & \mathbf{m}^{(w)} \end{bmatrix}^T, \tag{D.1}$$

with:

$$\mathbf{m}^{(r)} = \begin{bmatrix} \mathscr{M}_{new}\{\phi_1^{(r)}(\mathbf{x})\}, & \mathscr{M}_{new}\{\phi_2^{(r)}(\mathbf{x})\}, & \cdots & \mathscr{M}_{new}\{\phi_{K_r}^{(r)}(\mathbf{x})\} \end{bmatrix}, \quad r = u, v, w.$$

Letting $\Phi(t) = \mathbf{U}\Lambda\mathbf{U}^T$ be the eigendecomposition of $\Phi(t)$, with $\Lambda = \text{diag}\{\lambda_j\}$, it follows that $\Lambda^{-1/2}\mathbf{U}^T(\mathbf{y} - \bar{\mathbf{y}})|\mathscr{D} \overset{d}{=} \mathcal{N}(\boldsymbol{\mu}, \mathbf{I}_K)$, where $\boldsymbol{\mu} = \Lambda^{-1/2}\mathbf{U}^T(\hat{\mathbf{y}} - \bar{\mathbf{y}})$ and $K = K_u + K_v + K_w$. Denoting $\mathbf{a} = \Lambda^{-1/2}\mathbf{U}^T(\mathbf{y} - \bar{\mathbf{y}})$, the TKE expression in (13) can be rewritten as:

$$\begin{aligned} \kappa(\mathbf{x}, t) &= \frac{1}{2}(\mathbf{y} - \bar{\mathbf{y}})^T(\mathbf{y} - \bar{\mathbf{y}}) = \frac{1}{2}(\mathbf{U}\Lambda^{1/2}\mathbf{a})^T(\mathbf{U}\Lambda^{1/2}\mathbf{a}) \\ &= \frac{1}{2}(\mathbf{a}^T\Lambda^{1/2}\mathbf{U}^T\mathbf{U}\Lambda^{1/2}\mathbf{a}) \\ &= \frac{1}{2}\mathbf{a}^T\Lambda\mathbf{a} = \frac{1}{2}\sum_{j=1}^K \lambda_j a_j^2. \end{aligned} \tag{D.2}$$

Since $\mathbf{a} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I}_K)$, a_j^2 has a non-central chi-square distribution with one degree-of-freedom and non-centrality parameter μ_j^2 (we denote this as $\chi_1^2(\mu_j^2)$). $\kappa(\mathbf{x}, t)$ then becomes:

$$\sum_{j=1}^K \frac{\lambda_j}{2}\chi_1^2(\mu_j^2), \tag{D.3}$$

which is a sum of weighted non-central chi-squared distributions. The computation of the distribution function for such a random variable has been studied extensively, see, e.g., [265], [266, 267], [268], and [269], and we appeal to these methods for computing the

pointwise confidence interval of $\kappa(\mathbf{x}, t)$ in Section 4. Specifically, we employ the method of [269] through the R [270] package `CompQuadForm` [271].

APPENDIX E

APPENDIX FOR CHAPTER 6

E.1 Proof of Theorem 22

Lemma 17. *Let $h : \mathbb{R}^p \to \mathbb{R}_+$ be a strictly convex function, and let $g : \mathbb{R}_+ \to \mathbb{R}_+$ be a convex and strictly increasing function. Then the composition $g \circ h : \mathbb{R}^p \to \mathbb{R}_+$ is strictly convex.*

Proof. (Lemma 17) This is easy to show using first principles. Let $\alpha \in (0, 1)$ and let $\mathbf{z} \neq \mathbf{z}'$ be two points in $\mathbb{R}^p$. By strict convexity, we have:

$$h(\alpha\mathbf{z} + (1-\alpha)\mathbf{z}') < \alpha h(\mathbf{z}) + (1-\alpha)h(\mathbf{z}').$$

Moreover, since g is strictly increasing and convex, it follows that:

$$(g \circ h)(\alpha\mathbf{z} + (1-\alpha)\mathbf{z}') < g(\alpha h(\mathbf{z}) + (1-\alpha)h(\mathbf{z}')) \leq \alpha(g \circ h)(\mathbf{z}) + (1-\alpha)(g \circ h)(\mathbf{z}'),$$

which proves the strict convexity of $g \circ h$. □

Proof. (Theorem 22) Let $g(x) = x^{q/2}$ and $h(\mathbf{z}) = \|\mathbf{z} - \mathbf{z}_i\|_2^2$. It is easy to verify that h is strictly convex, and g is convex and strictly increasing on $\mathbb{R}_+$. By Lemma 1, it follows that $(g \circ f)(\mathbf{x}) = \|\mathbf{z} - \mathbf{z}_i\|_2^q$ is strictly convex. Hence, for any $\alpha \in (0, 1)$ and $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^p, \mathbf{z} \neq \mathbf{z}'$, we have:

$$\begin{aligned} D_q(\alpha\mathbf{z} + (1-\alpha)\mathbf{z}'; \mathfrak{X}) &= \frac{1}{mq} \sum_{i=1}^{n} \| \{(\alpha\mathbf{z} + (1-\alpha)\mathbf{z}') - \mathbf{z}_i\} \|_2^q \\ &< \frac{1}{mq} \sum_{i=1}^{n} \{\alpha\|\mathbf{z} - \mathbf{z}_i\|_2^q + (1-\alpha)\|\mathbf{z}' - \mathbf{z}_i\|_2^q\} \\ &= \alpha D_q(\mathbf{z}; \mathfrak{X}) + (1-\alpha) D_q(\mathbf{z}'; \mathfrak{X}), \end{aligned}$$

so the objective $D_q(\mathbf{z}; \mathfrak{X})$ is strictly convex in $\mathbf{z}$.

Using this fact, we show that (5.5) has a unique minimizer. Note that the objective $D_q(\mathbf{z}; \mathfrak{X})$ is continuous and coercive on the closed set $\mathbb{R}^p$, where the latter term implies that for all sequences $\{\mathbf{z}_k\}_{k=1}^{\infty}$ satisfying $\|\mathbf{z}_k\|_2 \to \infty$, $\lim_{k\to\infty} D_q(\mathbf{z}_k; \mathfrak{X}) = \infty$. It follows from Proposition A.8 in [272] and the strict convexity of $D_q(\mathbf{z}; \mathfrak{X})$ that there exists exactly one one global minimum of (5.5), so $C_q(\mathfrak{X})$ is uniquely defined.

To prove that the unique minimizer $C_q(\mathfrak{X})$ is contained in conv$(\mathfrak{X})$, note that by first-order optimality conditions, $C_q(\mathfrak{X})$ must satisfy:

$$\begin{aligned}\nabla D_q(C_q(\mathfrak{X}); \mathfrak{X}) = \frac{1}{n}\sum_{i=1}^{m} \left\{ \|C_q(\mathfrak{X}) - \mathbf{z}_i\|_2^{q-2}(C_q(\mathfrak{X}) - \mathbf{z}_i) \right\} = \mathbf{0} \\ \Leftrightarrow C_q(\mathfrak{X}) = \sum_{i=1}^{m} \left\{ \frac{\|C_q(\mathfrak{X}) - \mathbf{z}_i\|_2^{q-2}}{\sum_{j=1}^{n} \|C_q(\mathfrak{X}) - \mathbf{z}_j\|_2^{q-2}}\, \mathbf{z}_i \right\} \equiv \sum_{i=1}^{m} \alpha_i \mathbf{z}_i.\end{aligned}$$

Since the weights $\{\alpha_i\}_{i=1}^{m}$ satisfy $\alpha_i \geq 0$ and $\sum_{i=1}^{m} \alpha_i = 1$, it follows by definition that $C_q(\mathfrak{X}) \in$ conv$(\mathfrak{X})$, which is as desired.

□

E.2 Proof of Theorem 23

Lemma 18. *Let $\mathfrak{X} = \{\mathbf{z}_i\}_{i=1}^{m}$ be a set of points in $\mathbb{R}^p$. Then there exists some point $\mathbf{z}_j \in \mathfrak{X}$ such that $D_q(\mathbf{z}_j; \mathfrak{X}) \geq D_q(\mathbf{z}; \mathfrak{X})$ for all $\mathbf{z} \in$ conv$(\mathfrak{X})$.*

Proof. (Lemma 18) Since conv$(\mathfrak{X})$ is a compact set, the set of maximizers in:

$$\mathscr{M} = \text{argmax}_{\mathbf{z}\in\text{conv}(\mathfrak{X})} D_q(\mathbf{z}; \mathfrak{X})$$

is non-empty, so an equivalent claim is that $\mathbf{z}_j \in \mathscr{M}$ for some $j = 1, \cdots, m$. Suppose, for contradiction, that $\mathbf{z}_j \notin \mathscr{M}$ for all $j = 1, \cdots, m$, and let $\mathbf{z}' = \sum_{i=1}^{m} \alpha_j \mathbf{z}_j \notin \mathfrak{X}$ be a

maximizer in $\mathscr{M}$, with $\alpha_j \geq 0$ and $\sum_{j=1}^m \alpha_j = 1$. Then, by convexity, we have:

$$D_q(\mathbf{z}';\mathfrak{Z}) = \frac{1}{mq}\sum_{i=1}^m \left\| \sum_{j=1}^m \alpha_j(\mathbf{z}_j - \mathbf{z}_i) \right\|_2^q \leq \frac{1}{mq}\sum_{i=1}^m\sum_{j=1}^m \alpha_j \|\mathbf{z}_j - \mathbf{z}_i\|_2^q = \frac{1}{mq}\sum_{j=1}^m \alpha_j \left(\sum_{i=1}^m \|\mathbf{z}_j - \mathbf{z}_i\|_2^q \right)$$
$$= \sum_{j=1}^m \alpha_j D_q(\mathbf{z}_j;\mathfrak{Z}),$$

which implies that $D_q(\mathbf{z}';\mathfrak{Z}) \leq D_q(\mathbf{z}_j;\mathfrak{Z})$ for at least one $j = 1, \cdots, m$. Since $\mathbf{z}' \in \mathscr{M}$, this implies that $\mathbf{z}_j \in \mathscr{M}$, which is a contradiction. The lemma therefore holds. □

Proof. (Theorem 23) Since $D_q(\mathbf{z};\mathfrak{Z})$ is twice-differentiable, it is β-smooth on conv($\mathfrak{Z}$) if and only if:

$$\nabla^2 D_q(\mathbf{z};\mathfrak{Z}) \preceq \beta \mathbf{I} \quad \text{for all } \mathbf{z} \in \text{conv}(\mathfrak{Z}). \tag{E.1}$$

Letting $\lambda_{max}\{\mathbf{A}\}$ denote the largest eigenvalue of $\mathbf{A}$, it follows that:

$$\begin{aligned}
&\lambda_{max}\{\nabla^2 D_q(\mathbf{z};\mathfrak{Z})\} \\
&= \lambda_{max}\left\{ \frac{q-2}{m}\sum_{i=1}^m \left\{ \|\mathbf{z}-\mathbf{z}_i\|_2^{q-4}(\mathbf{z}-\mathbf{z}_i)(\mathbf{z}-\mathbf{z}_i)^T \right\} + \frac{1}{m}\sum_{i=1}^m \|\mathbf{z}-\mathbf{z}_i\|_2^{q-2}\mathbf{I} \right\} \\
&\leq \frac{q-2}{m}\sum_{i=1}^m \|\mathbf{z}-\mathbf{z}_i\|_2^{q-4}\lambda_{max}\left\{ (\mathbf{z}-\mathbf{z}_i)(\mathbf{z}-\mathbf{z}_i)^T \right\} + \frac{1}{m}\sum_{i=1}^m \|\mathbf{z}-\mathbf{z}_i\|_2^{q-2}\lambda_{max}\{\mathbf{I}\} \\
&= \frac{q-2}{m}\sum_{i=1}^m \|\mathbf{z}-\mathbf{z}_i\|_2^{q-4}\cdot\|\mathbf{z}-\mathbf{z}_i\|_2^2 + \frac{1}{m}\sum_{i=1}^m \|\mathbf{z}-\mathbf{z}_i\|_2^{q-2} \\
&= \frac{q-1}{m}\sum_{i=1}^m \|\mathbf{z}-\mathbf{z}_i\|_2^{q-2} \leq \frac{q-1}{m}\max_{j=1,\cdots,m}\sum_{i=1}^m \|\mathbf{z}_j-\mathbf{z}_i\|_2^{q-2} = \bar{\beta},
\end{aligned}$$

where the last inequality holds by Lemma 18. Hence, $\nabla^2 D_q(\mathbf{z};\mathfrak{Z}) \preceq \bar{\beta}\mathbf{I}$ for all $\mathbf{z} \in$ conv($\mathfrak{Z}$), so $D_q(\mathbf{z};\mathfrak{Z})$ is $\bar{\beta}$-smooth on conv($\mathfrak{Z}$) by (E.1).

Likewise, since $D_q(\mathbf{z};\mathfrak{Z})$ is twice-differentiable, it is μ-strongly convex on conv($\mathfrak{Z}$) if and only if:

$$\mu\mathbf{I} \preceq \nabla^2 D_q(\mathbf{z};\mathfrak{Z}) \quad \text{for all } \mathbf{z} \in \text{conv}(\mathfrak{Z}). \tag{E.2}$$

Letting $\lambda_{min}\{\mathbf{A}\}$ denote the smallest eigenvalue of $\mathbf{A}$, we have:

$$\begin{aligned}
\lambda_{min}\{\nabla^2 D_q(\mathbf{z};\mathfrak{Z})\} &= \lambda_{min}\left\{\frac{q-2}{m}\sum_{i=1}^{m}\left\{\|\mathbf{z}-\mathbf{z}_i\|_2^{q-4}(\mathbf{z}-\mathbf{z}_i)(\mathbf{z}-\mathbf{z}_i)^T\right\} + \frac{1}{m}\sum_{i=1}^{m}\|\mathbf{z}-\mathbf{z}_i\|_2^{q-2}\mathbf{I}\right\} \\
&\geq \frac{q-2}{m}\sum_{i=1}^{m}\|\mathbf{z}-\mathbf{z}_i\|_2^{q-4}\lambda_{min}\left\{(\mathbf{z}-\mathbf{z}_i)(\mathbf{z}-\mathbf{z}_i)^T\right\} + \frac{1}{m}\sum_{i=1}^{m}\|\mathbf{z}-\mathbf{z}_i\|_2^{q-2}\lambda_{min}\{\mathbf{I}\} \\
&\geq \frac{q-2}{m}\sum_{i=1}^{m}\|\mathbf{z}-\mathbf{z}_i\|_2^{q-4}\cdot 0 + \frac{1}{m}\sum_{i=1}^{m}\|\mathbf{z}-\mathbf{z}_i\|_2^{q-2} \\
&\geq \frac{1}{m}\sum_{i=1}^{m}\|C_{q-2}(\mathfrak{Z})-\mathbf{z}_i\|_2^{q-2} \\
&= \bar{\mu},
\end{aligned}$$

where the last inequality holds by definition of $C_{q-2}(\mathfrak{Z})$. Hence by (E.2), $D_q(\mathbf{z};\mathfrak{Z})$ is $\bar{\mu}$-strongly convex. □

E.3 Proof of Corollary 3

Consider a β-smooth and μ-strongly convex function h with unique minimizer $\mathbf{u}^*$. It can be shown [273] that an iteration upper bound of $t = O\left(\sqrt{\frac{\beta}{\mu}}\log\frac{1}{\epsilon_{in}}\right)$ guarantees an ϵ_{in}-accuracy in objective, i.e. $|h(\mathbf{u}^{[t]}) - h(\mathbf{u}^*)| < \epsilon_{in}$. Combining this iteration bound with the result in Theorem 23, and using the fact that each update requires $O(mp)$ work, we get the desired result.

E.4 Proof of Theorem 24

The three parts of this theorem are individually easy to verify. For finite termination, we showed in Section 3.1 that the objective in (5.7) strictly decreases after each loop iteration of Algorithm 7. Moreover, there are exactly N^n possible assignments of the sample $\{\mathbf{y}_j\}_{j=1}^N$ to the design points $\{\mathbf{m}_i\}_{i=1}^n$. Suppose, for contradiction, that Algorithm 7 does not terminate after N^n iterations. Then there exists at least two iterations which begin with the same assignment of $\{\mathbf{y}_j\}_{j=1}^N$. This, in turn, generates the same design $\{\mathbf{m}_i\}_{i=1}^n$ at the

end of both iterations, which presents a contradiction to the strictly decreasing objective values induced by each loop iteration of Algorithm 7. The first claim therefore holds.

Next, regarding running time, consider the two updates in a single loop iteration of Algorithm 7. The first update assigns each sample point in $\{\mathbf{y}_j\}$ to its closest design point, which requires $O(Nnp)$ work. The second update computes, for each design point, the C_q-center of samples assigned to it. Let $\mathfrak{X} = \{\mathbf{z}_j\}_{j=1}^{m_i}$ be the m_i points assigned to the i-th design point. From Corollary 3, the computation of its C_q-center requires $O(m_i p\sqrt{(q-1)\kappa_{q-2}(\mathfrak{X})\log(1/\epsilon_{in})})$ work. Letting $\tilde{\mathbf{z}} = \text{argmax}_{j=1,\cdots,m_i} D_q(\mathbf{z}_j;\mathfrak{X})$, it follows that for any $q \geq 2$:

$$\begin{aligned}\kappa_q(\mathfrak{X}) = \frac{D_q(\tilde{\mathbf{z}};\mathfrak{X})}{D_q(C_q(\mathfrak{X});\mathfrak{X})} &\leq \frac{\sum_{i=1}^{m_i}\|\mathbf{z}_i - C_q(\mathfrak{X})\|_2^q + m_i\|\tilde{\mathbf{z}} - C_q(\mathfrak{X})\|_2^q}{\sum_{i=1}^{m_i}\|\mathbf{z}_i - C_q(\mathfrak{X})\|_2^q} \\ &\leq 1 + \frac{m_i\|\tilde{\mathbf{z}} - C_q(\mathfrak{X})\|_2^q}{\sum_{i=1}^{m_i}\|\mathbf{z}_i - C_q(\mathfrak{X})\|_2^q} \leq m_i + 1.\end{aligned}$$

Hence, updating C_q-centers for all n design points require a total work of:

$$\begin{aligned}\sum_{i=1}^{n} O(m_i p\sqrt{(q-1)\kappa_{q-2}(\mathfrak{X})\log(1/\epsilon_{in})}) &\leq O\left(\left\{\sum_{i=1}^{n} m_i^{3/2}\right\} p\sqrt{q-1}\log\frac{1}{\epsilon_{in}}\right) \\ &\leq O\left(\left\{\sum_{i=1}^{n} m_i\right\}^{3/2} p\sqrt{q-1}\log\frac{1}{\epsilon_{in}}\right) \\ &= O\left(N^{3/2} p\sqrt{q-1}\log\frac{1}{\epsilon_{in}}\right).\end{aligned}$$

Finally, since $n \leq N^{1/2}$, the running time of the second step dominates the first, which completes the argument.

Finally, assume that the C_q-center updates in (5.5) are exact. By the termination conditions of Algorithm 7, the converged design is optimal given fixed assignments, and the converged assignment variables are optimal given a fixed design. Hence, the converged design (as well as its corresponding assignment) are locally optimal for (5.7).

E.5 Proof of Proposition 5

This can be shown by a simple application of the triangle inequality. Let $\mathcal{D} = \{\mathbf{m}_i\}_{i=1}^n$ be the design at the current iteration, and without loss of generality, suppose the first design point $\mathbf{m}_1$ is to be updated. Also, let $\{d_i\}_{i=1}^n$ be the minimax distances for each design point (defined in (5.15)), with $d^* = \max_i d_i$ being the overall minimax distance of $\mathcal{D}$.

Let $\tilde{\mathbf{m}}_1$ be the optimal design point in (5.16), and note that, by optimization constraints, $\|\tilde{\mathbf{m}}_1 - \mathbf{m}_1\| \leq d^* - d_1$. Denoting $\tilde{d}^*$ as the overall minimax distance of the new design $\tilde{\mathcal{D}} = \{\tilde{\mathbf{m}}_1, \mathbf{m}_2, \cdots, \mathbf{m}_n\}$, the claim is that $\tilde{d}^* \leq d^*$. To prove this, let $\mathbf{x}$ be the point in $\mathcal{X}$ achieving the minimax distance $\tilde{d}^*$, and consider the following three cases:

- If $Q(\mathbf{x}, \tilde{\mathcal{D}})$, the closest design point to $\mathbf{x}$ in $\tilde{\mathcal{D}}$, equals $\tilde{\mathbf{m}}_1$, then:

$$\tilde{d}^* = \|\mathbf{x} - \tilde{\mathbf{m}}_1\| \leq \|\mathbf{x} - \mathbf{m}_1\| + \|\mathbf{m}_1 - \tilde{\mathbf{m}}_1\| \leq d_1 + (d^* - d_1) = d^*.$$

- If $Q(\mathbf{x}, \tilde{\mathcal{D}}) = \mathbf{m}_i$ for some $i = 2, \cdots, n$, and $Q(\mathbf{x}, \mathcal{D}) = \mathbf{m}_1$, then:

$$\tilde{d}^* = \|\mathbf{x} - \mathbf{m}_i\| \leq \|\mathbf{x} - \tilde{\mathbf{m}}_1\| \leq \|\mathbf{x} - \mathbf{m}_1\| + \|\mathbf{m}_1 - \tilde{\mathbf{m}}_1\| \leq d_1 + (d^* - d_1) = d^*.$$

- If $Q(\mathbf{x}, \tilde{\mathcal{D}}) = \mathbf{m}_i$ for some $i = 2, \cdots, n$, and $Q(\mathbf{x}, \mathcal{D}) = \mathbf{m}_j$ for some $j = 1, \cdots, n$, then it must be the case that $i = j$, since the only change from $\mathcal{D}$ to $\tilde{\mathcal{D}}$ is the first design point. Hence:

$$\tilde{d}^* = \|\mathbf{x} - \mathbf{m}_i\| \leq d_i \leq d^*.$$

This proves the proposition.

E.6 Additional minimax designs on $[0,1]^p$

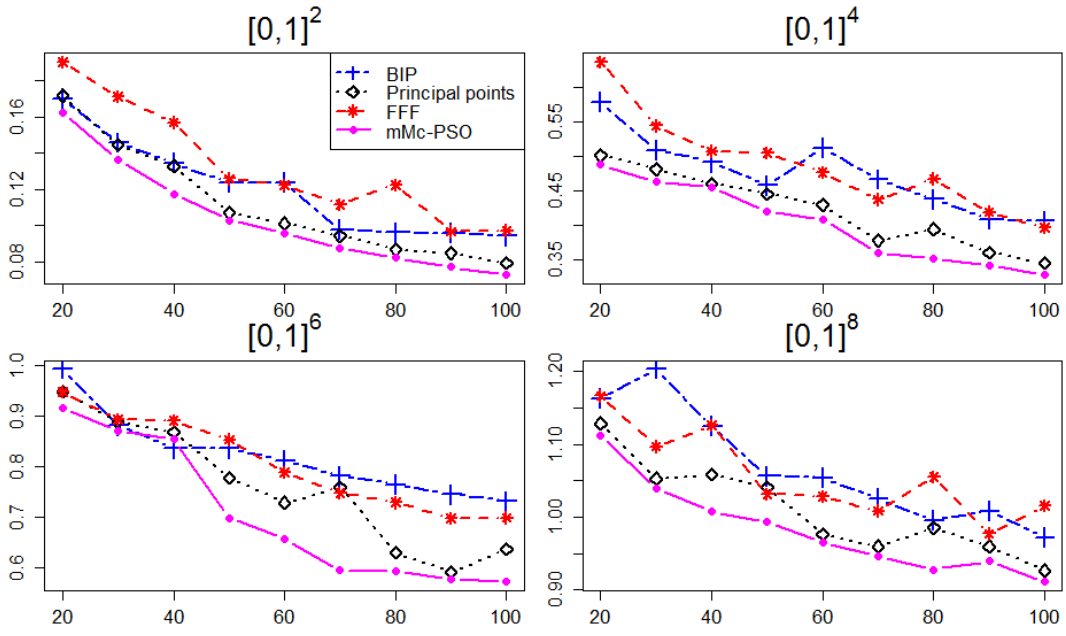

Figure E.1: Minimax criterion on $[0,1]^p$ for $p = 2, 4, 6$ and 8.

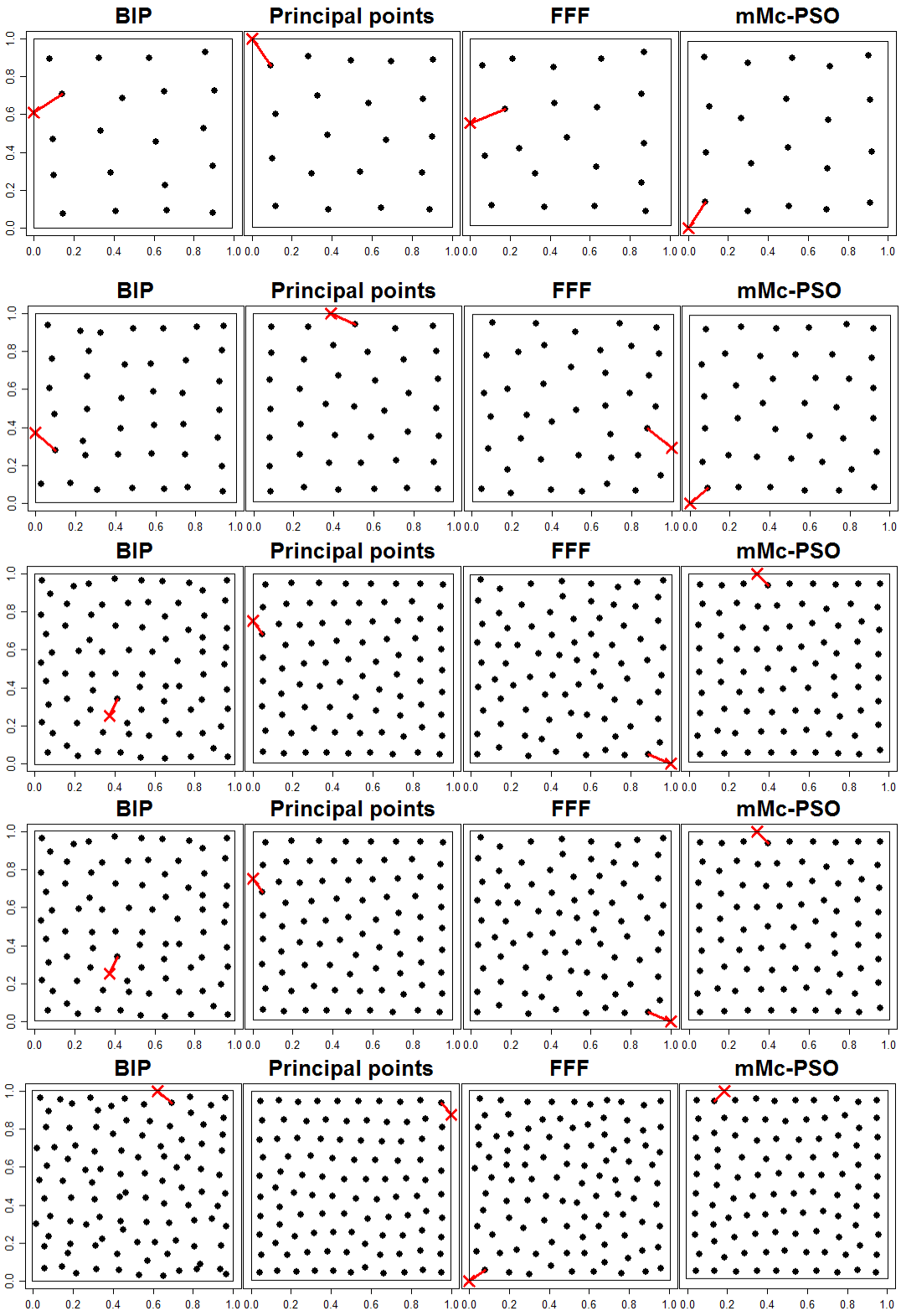

Figure E.2: 20-, 40-, 60-, 80- and 100-point designs on the unit hypercube $[0, 1]^2$.

E.7 Additional minimax designs on A_p and B_p

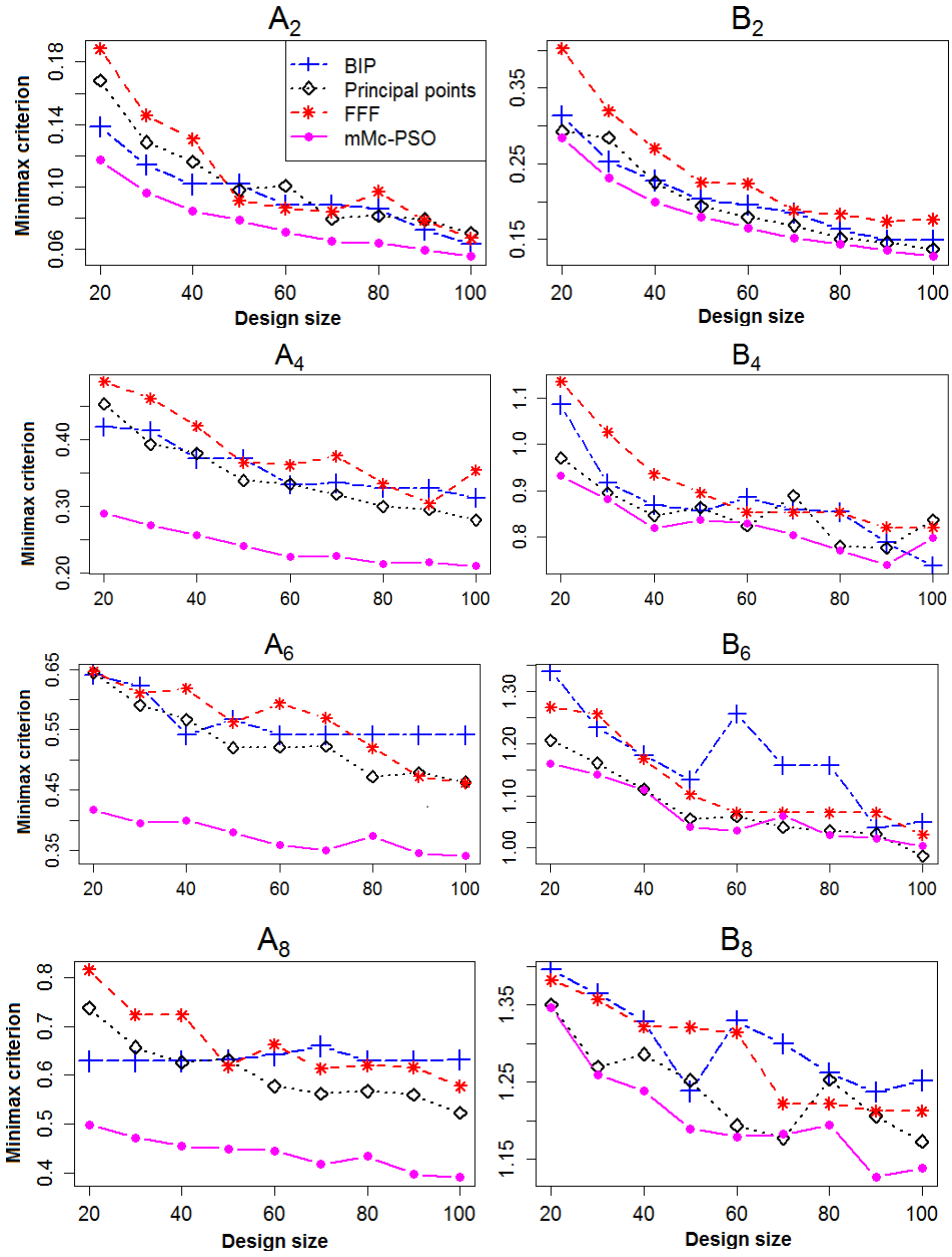

Figure E.3: Minimax criterion on A_p and B_p for $p = 2, 4, 6$ and 8.

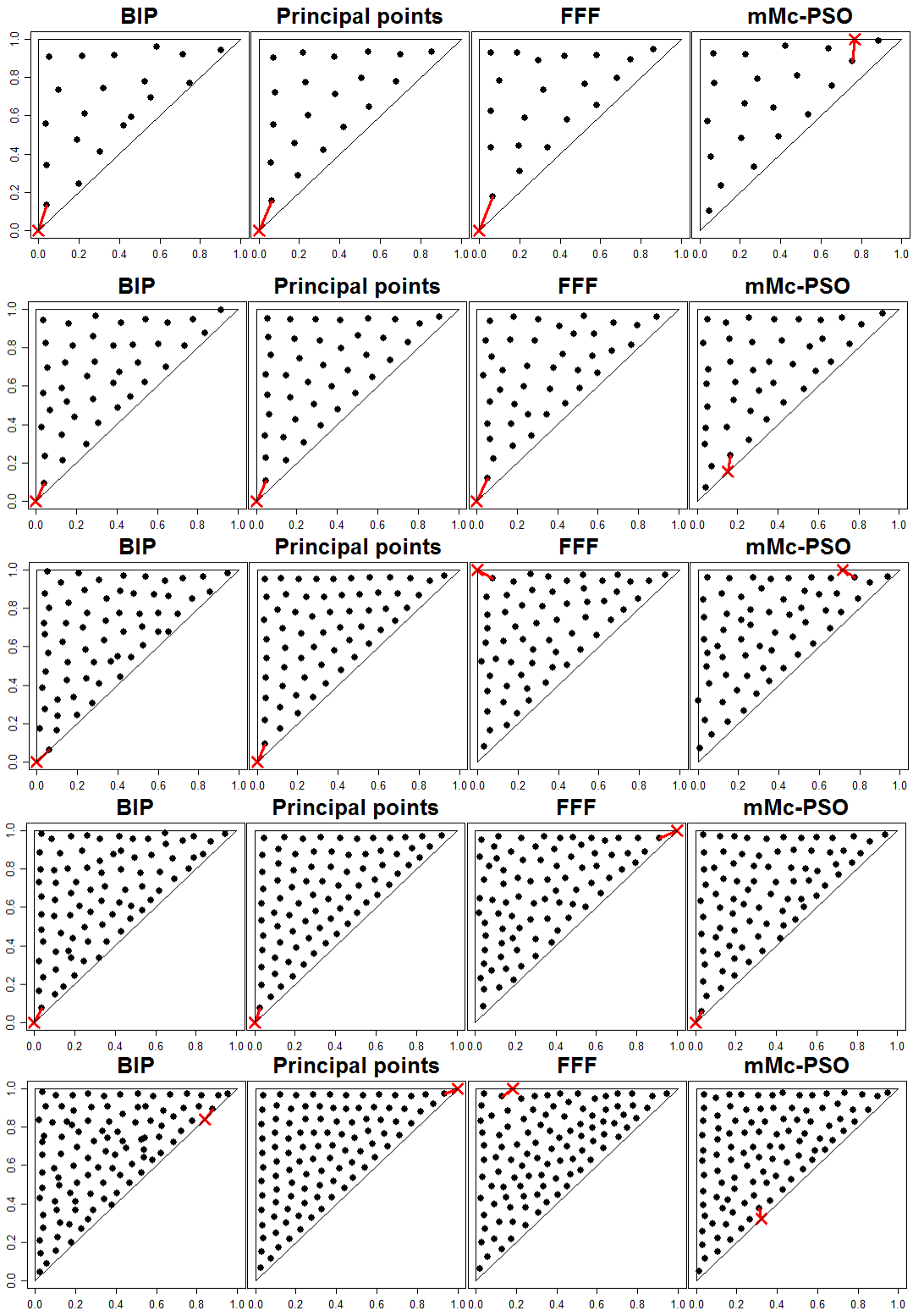

Figure E.4: 20-, 40-, 60-, 80- and 100-point designs on the unit simplex A_2.

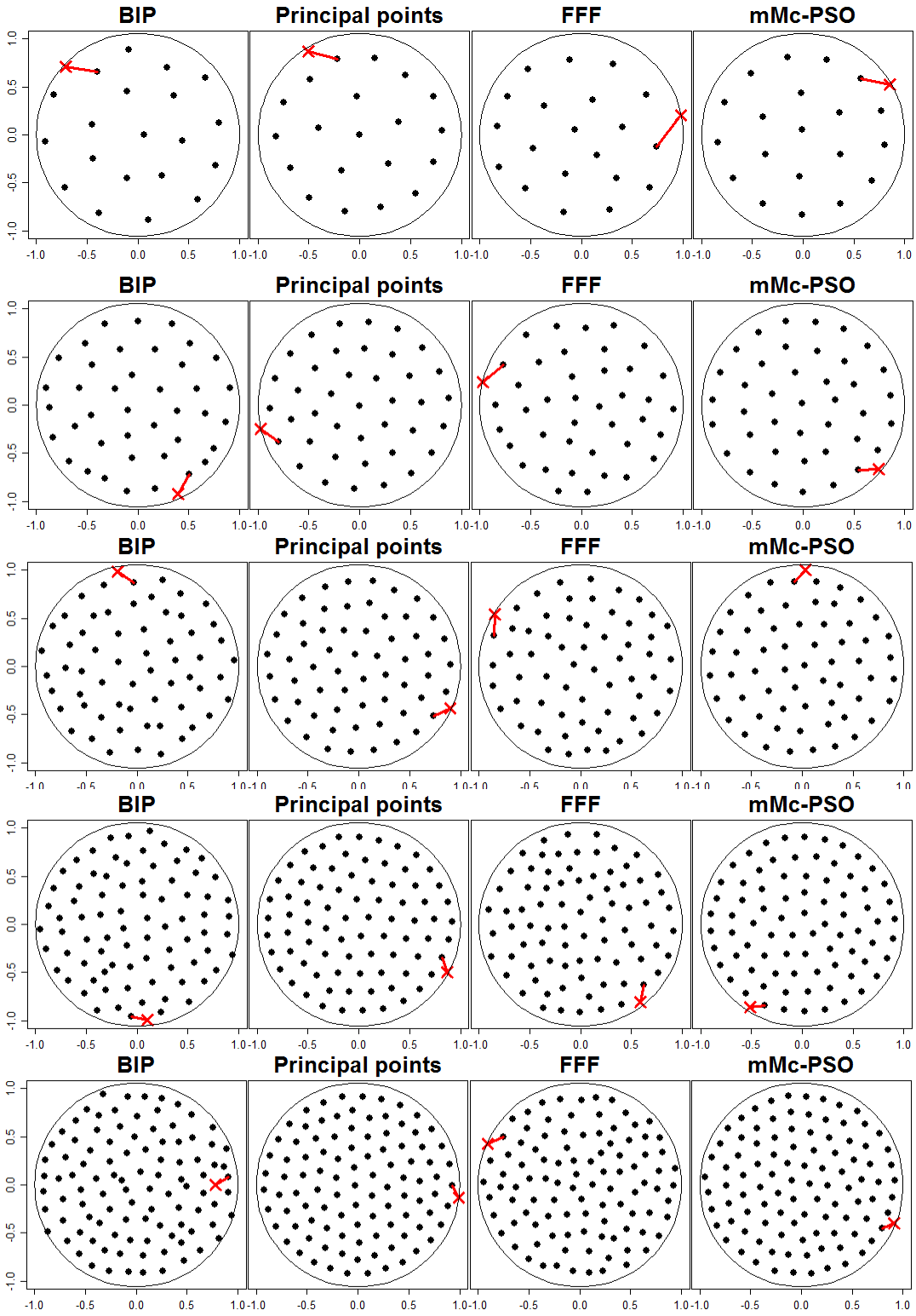

Figure E.5: 20-, 40-, 60-, 80- and 100-point designs on the unit ball B_2.

E.8 Additional minimax designs on Georgia

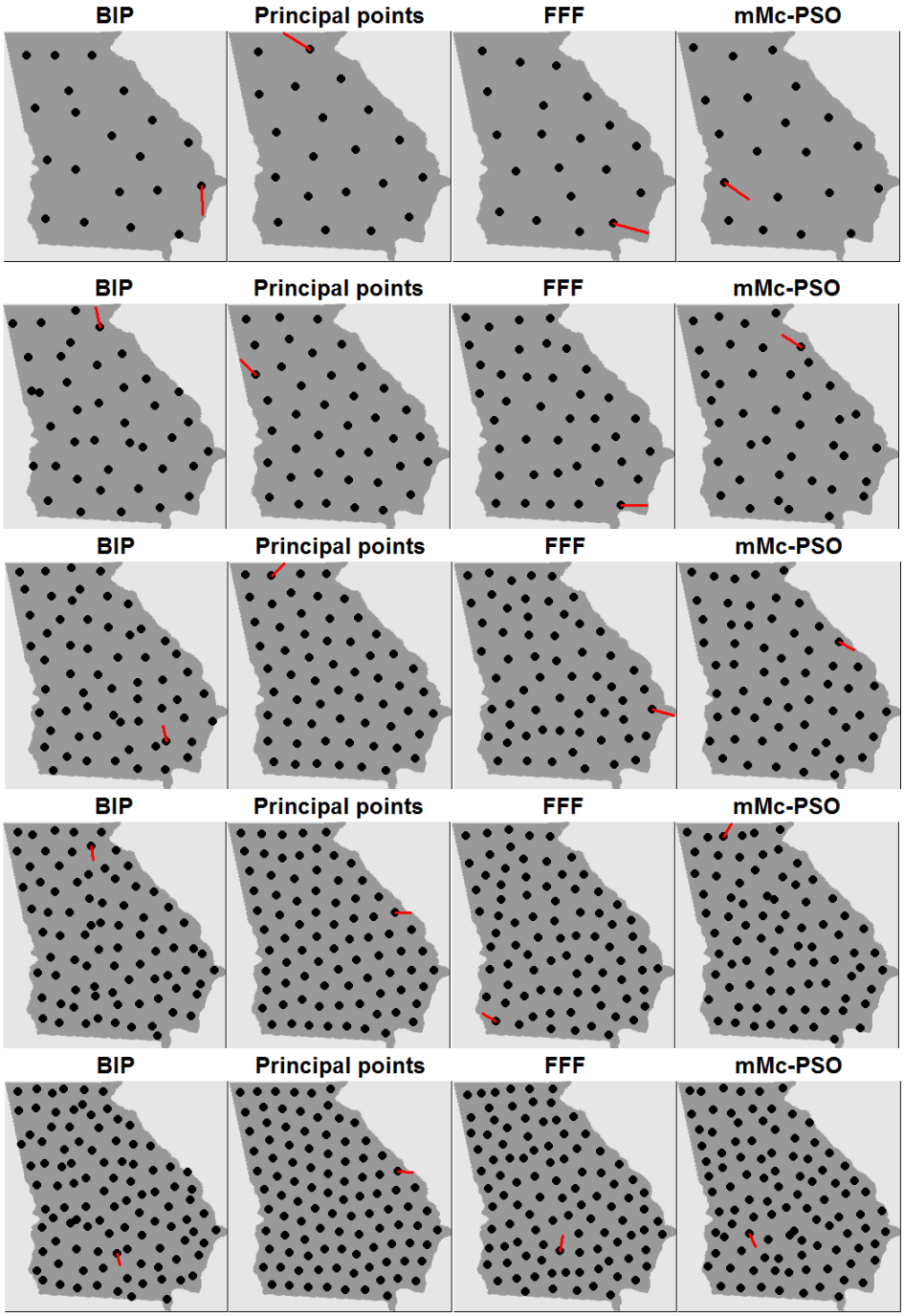

Figure E.6: 20-, 40-, 60-, 80- and 100-point designs on Georgia.

APPENDIX F

APPENDIX FOR CHAPTER 7

F.1 Proof of Lemma 6

Proof. We first prove part (a) of the lemma. To show that $\mathbf{X} \in \mathcal{T}$ almost surely, let $\mathbf{Z}$ be an arbitrary matrix in $\mathbb{R}^{m_1 \times m_2}$, with SVD $\mathbf{Z} = \widetilde{\mathbf{U}}\mathbf{D}\widetilde{\mathbf{V}}^T$, $\mathbf{D} = \text{diag}(\{d_k\}_{k=1}^R)$. Letting $\mathbf{u}_k = \mathcal{P}_{\mathcal{U}}\widetilde{\mathbf{u}}_k$ and $\mathbf{v}_k = \mathcal{P}_{\mathcal{V}}\widetilde{\mathbf{v}}_k$, where $\widetilde{\mathbf{u}}_k$ and $\widetilde{\mathbf{v}}_k$ are column vectors for $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{V}}$ respectively, we have $\mathbf{u}_k \in \mathcal{U}$ and $\mathbf{v}_k \in \mathcal{V}$ for $k = 1, \cdots, R$. From Definition 16, $\mathbf{X}$ can then be written as $\mathbf{X} = \mathcal{P}_{\mathcal{U}}\mathbf{Z}\mathcal{P}_{\mathcal{V}} = (\mathcal{P}_{\mathcal{U}}\widetilde{\mathbf{U}})\mathbf{D}(\mathcal{P}_{\mathcal{V}}\widetilde{\mathbf{V}})^T = \sum_{k=1}^R d_k \mathbf{u}_k \mathbf{v}_k^T$, as desired. Next, note that the pseudo-inverse of $\mathcal{P}_{\mathbf{u}}$, $(\mathcal{P}_{\mathbf{u}})^+$, is simply $\mathcal{P}_{\mathbf{u}}$, since $\mathcal{P}_{\mathbf{u}}(\mathcal{P}_{\mathbf{u}})^+\mathcal{P}_{\mathbf{u}} = (\mathcal{P}_{\mathbf{u}})^+\mathcal{P}_{\mathbf{u}}(\mathcal{P}_{\mathbf{u}})^+ = \mathcal{P}_{\mathbf{u}}$ by the idempotency of $\mathcal{P}_{\mathbf{u}}$, and $\mathcal{P}_{\mathbf{u}}(\mathcal{P}_{\mathbf{u}})^+ = (\mathcal{P}_{\mathbf{u}})^+\mathcal{P}_{\mathbf{u}}$ are both symmetric. Moreover, letting $\det^*$ be the pseudo-determinant operator, we have $\det^*(\mathcal{P}_{\mathcal{U}}) = \det^*(\mathbf{U}\mathbf{U}^T) = \det(\mathbf{U}^T\mathbf{U}) = 1$, and $\det^*(\mathcal{P}_{\mathcal{V}}) = 1$ by the same argument. Using this along with Theorem 2.2.1 in [223], the density function $f(\mathbf{X})$ and the distribution of $\text{vec}(\mathbf{X})$ follow immediately.

We now prove part (b) of the lemma. From part (a), we have $\text{vec}(\mathbf{X}) \sim \mathcal{N}\{\mathbf{0}, \sigma^2(\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})\}$, so:

$$[\mathbf{Y}_\Omega, \mathbf{X}_{\Omega^c}] \sim \mathcal{N}\left\{\mathbf{0}, \begin{bmatrix} \sigma^2\mathbf{R}_N(\Omega) + \eta^2\mathbf{I} & \sigma^2(\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})_{\Omega,\Omega^c} \\ \sigma^2(\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})_{\Omega,\Omega^c}^T & \sigma^2(\mathcal{P}_{\mathcal{V}} \otimes \mathcal{P}_{\mathcal{U}})_{\Omega^c} \end{bmatrix}\right\}.$$

The expressions for $\mathbf{X}_{\Omega^c}^P$ and $\mathbf{\Sigma}_{\Omega^c}^P$ in (6.7) then follow from the conditional density of the multivariate Gaussian distribution. □

F.2 Proof of Lemma 7

Proof. Since $U(\mathcal{G}_{R,m-R})$ is a special case of the matrix Langevin distribution (Section 2.3.2 in [224]), it follows from (2.3.22) of [224] that $[\mathcal{P}_{\mathcal{U}}|R] \propto 1$ and $[\mathcal{P}_{\mathcal{V}}|R] \propto 1$. For

fixed η^2 and σ^2, the MAP estimator for $\mathbf{X}$ then becomes:

$$
\begin{aligned}
\tilde{\mathbf{X}} \in & \underset{\mathbf{X}\in\mathbb{R}^{m_1\times m_2}}{\text{Argmax}} \; [\mathbf{Y}_\Omega|\mathbf{X},\eta^2]\;[\mathbf{X}|\mathscr{P}_{\mathscr{U}},\mathscr{P}_{\mathscr{V}},\sigma^2,R]\cdot \\
& \qquad [\mathscr{P}_{\mathscr{U}}|R]\;[\mathscr{P}_{\mathscr{V}}|R]\;[R] \\
& \quad \text{s.t. } \mathscr{P}_{\mathscr{U}}\in\mathscr{G}_{R,m_1-R}, \mathscr{P}_{\mathscr{V}}\in\mathscr{G}_{R,m_2-R}, R\le m_1\wedge m_2 \\
\in & \underset{\mathbf{X}\in\mathbb{R}^{m_1\times m_2}}{\text{Argmax}} \; \exp\left\{-\frac{1}{2\eta^2}\|\mathbf{Y}_\Omega-\mathbf{X}_\Omega\|_2^2\right\}\cdot \\
& \qquad \left[\frac{1}{(2\pi\sigma^2)^{R^2/2}}\exp\left\{-\frac{1}{2\sigma^2}\text{tr}\left[(\mathbf{X}\mathscr{P}_{\mathscr{V}})^T(\mathscr{P}_{\mathscr{U}}\mathbf{X})\right]\right\}\right]\cdot \\
& \quad 1\cdot 1\cdot 1 \\
& \quad \text{s.t. } \mathscr{P}_{\mathscr{U}}\in\mathscr{G}_{R,m_1-R}, \mathscr{P}_{\mathscr{V}}\in\mathscr{G}_{R,m_2-R}, R\le m_1\wedge m_2 \\
\in & \underset{\mathbf{X}\in\mathbb{R}^{m_1\times m_2}}{\text{Argmin}} \; \left[\frac{1}{\eta^2}\|\mathbf{Y}_\Omega-\mathbf{X}_\Omega\|_2^2+\log(2\pi\sigma^2)R^2+\right. \\
& \qquad \left.\frac{1}{\sigma^2}\text{tr}\left[(\mathbf{X}\mathscr{P}_{\mathscr{V}})^T(\mathscr{P}_{\mathscr{U}}\mathbf{X})\right]\right] \\
& \quad \text{s.t. } \mathscr{P}_{\mathscr{U}}\in\mathscr{G}_{R,m_1-R}, \mathscr{P}_{\mathscr{V}}\in\mathscr{G}_{R,m_2-R}, R\le m_1\wedge m_2.
\end{aligned}
$$

Since $\mathbf{X}=\mathscr{P}_{\mathscr{U}}\mathbf{Z}\mathscr{P}_{\mathscr{V}}$, we have $\mathbf{X}=\mathbf{U}\mathbf{D}\mathbf{V}^T$ for some $\mathbf{D}=\text{diag}(\{d_k\}_{k=1}^R)$, $\mathbf{U}\in\mathbb{R}^{m_1\times R}$ and $\mathbf{V}\in\mathbb{R}^{m_2\times R}$, where $\mathbf{U}$ and $\mathbf{V}$ are R-frames satisfying $\mathscr{P}_{\mathscr{U}}=\mathbf{U}\mathbf{U}^T$ and $\mathscr{P}_{\mathscr{V}}=\mathbf{V}\mathbf{V}^T$. Hence:

$$
\begin{aligned}
&\text{tr}\left[(\mathbf{X}\mathscr{P}_{\mathbf{V}})^T(\mathscr{P}_{\mathbf{U}}\mathbf{X})\right] \\
&=\text{tr}\left[(\mathbf{V}\mathbf{V}^T)(\mathbf{V}\mathbf{D}\mathbf{U}^T)(\mathbf{U}\mathbf{U}^T)(\mathbf{U}\mathbf{D}\mathbf{V}^T)\right] \\
&=\text{tr}\left[(\mathbf{V}^T\mathbf{V})^2\mathbf{D}(\mathbf{U}^T\mathbf{U})^2\mathbf{D}\right] && \text{(cyclic invariance of trace)} \\
&=\text{tr}\left[\mathbf{D}^2\right] && (\mathbf{V}^T\mathbf{V}=\mathbf{I} \text{ and } \mathbf{U}^T\mathbf{U}=\mathbf{I}) \\
&=\|\mathbf{X}\|_F^2, && \text{(Frob. norm is equal to Schatten 2-norm)}
\end{aligned}
$$

which proves the expression in (6.10).

□

F.3 Proof of Theorem 25

Proof. Consider the following block decomposition:

$$\mathbf{R}_{N+1}(\Omega \cup (i,j)) + \gamma^2 \mathbf{I} = \begin{pmatrix} \mathbf{R}_N(\Omega) + \gamma^2 \mathbf{I} & \boldsymbol{\nu}_i(\mathcal{U}) \circ \boldsymbol{\nu}_j(\mathcal{V}) \\ [\boldsymbol{\nu}_i(\mathcal{U}) \circ \boldsymbol{\nu}_j(\mathcal{V})]^T & \mu_i(\mathcal{U})\mu_j(\mathcal{V}) + \gamma^2 \end{pmatrix}.$$

Using the Schur complement identity for matrix inverses [274], we have:

$$\left[\mathbf{R}_{N+1}(\Omega \cup (i,j)) + \gamma^2 \mathbf{I}\right]^{-1} = \begin{pmatrix} \boldsymbol{\Gamma} + \tau^{-1}\boldsymbol{\Gamma}\boldsymbol{\xi}\boldsymbol{\xi}^T\boldsymbol{\Gamma} & -\tau^{-1}\boldsymbol{\xi}^T\boldsymbol{\Gamma} \\ -\tau^{-1}\boldsymbol{\Gamma}\boldsymbol{\xi} & \tau^{-1} \end{pmatrix}, \tag{F.1}$$

where $\boldsymbol{\xi} = \boldsymbol{\nu}_i(\mathcal{U}) \circ \boldsymbol{\nu}_j(\mathcal{V})$, $\boldsymbol{\Gamma} = [\mathbf{R}_N(\Omega) + \gamma^2 \mathbf{I}]^{-1}$ and $\tau = \mu_i(\mathcal{U})\mu_j(\mathcal{V}) - \boldsymbol{\xi}^T \Gamma \boldsymbol{\xi} + \gamma^2$. Using the conditional variance expression in (6.15), $\tau = \text{Var}(X_{i,j}|\mathbf{Y}_\Omega)/\sigma^2 + \gamma^2$. Letting $\widetilde{\boldsymbol{\xi}} = \boldsymbol{\nu}_k(\mathcal{U}) \circ \boldsymbol{\nu}_l(\mathcal{V})$ and applying (6.15) again, it follows that:

$$\begin{aligned}
&\text{Var}(X_{k,l}|\mathbf{Y}_{\Omega \cup (i,j)}) \\
&= \sigma^2 \left\{ \mu_k(\mathcal{U})\mu_l(\mathcal{V}) - \widetilde{\boldsymbol{\xi}}^T \Gamma \widetilde{\boldsymbol{\xi}} \right\} \\
&\quad - \tau^{-1}\sigma^2 \left\{ \boldsymbol{\nu}_{i,j}^T \left[\mathbf{R}_N(\Omega) + \gamma^2 \mathbf{I}\right]^{-1} \boldsymbol{\nu}_{k,l} - \nu_{i,k}(\mathcal{U})\nu_{j,l}(\mathcal{V}) \right\}^2 \\
&\qquad\qquad\qquad\qquad \text{(using (F.1) and algebraic manipulations)} \\
&= \text{Var}(X_{k,l}|\mathbf{Y}_\Omega) - \frac{\text{Cov}^2(X_{i,j}, X_{k,l}|\mathbf{Y}_\Omega)}{\text{Var}(X_{i,j}|\mathbf{Y}_\Omega) + \eta^2}, \qquad \text{(from (6.7))}
\end{aligned}$$

which proves the theorem. □

F.4 Proof of Corollary 4

Proof. This follows directly from Theorem 25 and the fact that:

$$\text{Cov}^2(X_{i,j}, X_{k,l}|\mathbf{Y}_{\Omega_{1:N}})/\{\text{Var}(X_{i,j}|\mathbf{Y}_{\Omega_{1:N}}) + \eta^2\} \geq 0.$$

□

F.5 Proof of Corollary 5

Proof. Note that $\epsilon_N^2(k,l) = \text{Var}(X_{k,l}|\mathbf{Y}_{\Omega_{1:N}})$. From Theorem 25, it follows that:

$$\begin{aligned}
&\epsilon_{N+1}^2(k,l) \\
&= \epsilon_N^2(k,l) - \frac{\text{Corr}^2(X_{i_{N+1},j_{N+1}}, X_{k,l}|\mathbf{Y}_{\Omega_{1:N}})\text{Var}(X_{k,l}|\mathbf{Y}_{\Omega_{1:N}})}{1+\eta^2/\text{Var}(X_{i_{N+1},j_{N+1}}|\mathbf{Y}_{\Omega_{1:N}})} \\
&\geq \epsilon_N^2(k,l)\left(1 - \frac{\text{Corr}^2(X_{i_{N+1},j_{N+1}}, X_{k,l}|\mathbf{Y}_{\Omega_{1:N}})}{1+\gamma^2}\right).
\end{aligned}$$

where the last step follows because:

$$\begin{aligned}
\text{Var}(X_{i_{N+1},j_{N+1}}|\mathbf{Y}_{\Omega_{1:N}}) &= \epsilon_N^2(i_{N+1}, j_{N+1}) \\
&\leq \epsilon_{N-1}^2(i_{N+1}, j_{N+1}) \leq \cdots \\
&\leq \epsilon_0^2(i_{N+1}, j_{N+1}) \leq \sigma^2,
\end{aligned}$$

by the error monotonicity in Corollary 4, where $\epsilon_0^2(k,l) := \sigma^2\mu_k(\mathcal{U})\mu_l(\mathcal{V})$ from (6.14). Telescoping the first inequality, we get:

$$\epsilon_{N+1}^2(k,l) \geq \epsilon_0^2(k,l)\left[\prod_{n=1}^{N+1}\left(1 - \frac{\text{Corr}^2(X_{i_n,j_n}, X_{k,l}|\mathbf{Y}_{\Omega_{1:(n-1)}})}{1+\gamma^2}\right)\right].$$

This completes the proof. □

F.6 Proof of Lemma 8

Proof. A straight-forward extension of Lemma 6 (a) shows that, for fixed $\mathcal{P}_\mathcal{U}$, $\mathcal{P}_\mathcal{V}$, σ^2 and η^2, the noisy entries $\mathbf{Y}_\Omega$ follow the multivariate Gaussian distribution:

$$[\mathbf{Y}_\Omega|\mathcal{P}_\mathcal{U}, \mathcal{P}_\mathcal{V}, \sigma^2, \eta^2] \sim \mathcal{N}\{\mathbf{0}, \sigma^2\mathbf{R}_N(\Omega) + \eta^2\mathbf{I}\}.$$

The entropy expression for $\mathbf{Y}_\Omega$ then follows immediately. $\square$

F.7 Proof of Proposition 6

Proof. Note that the (i_n, j_n)-th entry of $\mathbf{X}$ can be written as $X_{i_n,j_n} = \langle \mathbf{M}_n, \mathbf{X} \rangle_F$, where $\mathbf{M}_n := \mathbf{e}_{i_n}\mathbf{e}_{j_n}^T$ is a rank-1 measurement mask on $\mathbf{X}$. This proposition then follows by applying Lemmas 4 and 5 from [63]. $\square$

F.8 Proof of Proposition 7

Proof. Assume the uniform priors $\mathscr{P}_\mathscr{U}, \mathscr{P}_\mathscr{V} \overset{i.i.d.}{\sim} U(\mathscr{G}_{R,m-R})$, and let Ω_1 and Ω_2 be two arbitrarily chosen balanced sampling schemes (i.e., with one observation in each row and column). By Section 1.4.2 in [224], the uniform measure $\mathscr{P} \sim U(\mathscr{G}_{R,m-R})$ is invariant under the transformation $\mathscr{P} \to H\mathscr{P}H^T$ for any $H \in O(m)$, where $O(m)$ is the orthogonal group of $m \times m$ orthonormal matrices. Equivalently, this means the uniform measure on the Grassmann manifold $\mathscr{G}_{R,m-R}$ is invariant under rotations around the origin). Since $\mathbf{R}_N(\Omega) = [\mathbf{e}_i\mathscr{P}_\mathscr{U}\mathbf{e}_{i'}\mathbf{e}_j\mathscr{P}\mathbf{e}_{j'}]_{(i,j)\in\Omega,(i',j')\in\Omega}$, it follows from (a) the above rotation invariance of $U(\mathscr{G}_{R,m-R})$, and (b) the balance of Ω_1 and Ω_2 that $\mathbf{R}_N(\Omega_1) \overset{d}{=} \mathbf{R}_N(\Omega_2)$. The claim then follows. $\square$

F.9 Proof of Lemma 9

Proof. This can be shown by a direct application of the determinant identity for Schur complements [274], which states that if $\mathbf{M}$ is in the block form:

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$$

with $\mathbf{D}$ invertible, then $\det(\mathbf{M}) = \det(\mathbf{D})\det(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})$. Using this along with the following block representation:

$$\mathbf{R}_{N+1}\{\Omega \cup (i,j)\} + \gamma^2\mathbf{I} = \begin{bmatrix} \mu_i(\mathcal{U})\mu_j(\mathcal{V}) + \gamma^2 & (\boldsymbol{\nu}_i(\mathcal{U}) \circ \boldsymbol{\nu}_j(\mathcal{V}))^T \\ \boldsymbol{\nu}_i(\mathcal{U}) \circ \boldsymbol{\nu}_j(\mathcal{V}) & \mathbf{R}_N(\Omega) + \gamma^2\mathbf{I} \end{bmatrix},$$

the expression for $\mathrm{H}\{(i,j)|\Omega_{1:N}\}$ then follows. □

F.10 Derivation of Gibbs sampler

Suppose, for the sake of derivation, that the full matrix $\mathbf{X}$ has been observed with noise (call this noisy matrix $\mathbf{Y}$); the imputation of missing entries in $\mathbf{Y}$ is discussed in a later step. For fixed rank R, the full posterior distribution of parameters $\mathbf{U}$, $\mathbf{D}$, $\mathbf{V}$, σ^2 and η^2 can be written as:

$$\begin{aligned}
&[\mathbf{U}, \mathbf{D}, \mathbf{V}, \sigma^2, \eta^2 | \mathbf{Y}] \\
&\quad \propto [\mathbf{Y}|\mathbf{U}, \mathbf{D}, \mathbf{V}, \eta^2, \sigma^2, R] \cdot [\mathbf{U}|R] \cdot [\mathbf{V}|R] \cdot [\mathbf{D}|\sigma^2] \cdot [\sigma^2] \cdot [\eta^2] \\
&\quad \propto \frac{1}{(\eta^2)^{(m_1 m_2)/2}} \exp\left\{ -\frac{1}{2\eta^2} \|\mathbf{Y} - \mathbf{U}\mathbf{D}\mathbf{V}^T\|_F^2 \right\} \\
&\qquad \cdot 1 \cdot 1 \cdot \frac{1}{(\sigma^2)^{R/2}} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{k=1}^{R} d_k^2 \right\} \prod_{\substack{k,l=1 \\ k<l}}^{R} |d_k^2 - d_l^2| \\
&\qquad \cdot \frac{1}{(\sigma^2)^{\alpha_{\sigma^2}+1}} \exp\left\{ -\frac{\beta_{\sigma^2}}{\sigma^2} \right\} \cdot \frac{1}{(\eta^2)^{\alpha_{\eta^2}+1}} \exp\left\{ -\frac{\beta_{\eta^2}}{\eta^2} \right\}.
\end{aligned}$$

From this, the full conditional Gibbs updates can be derived as follows (algebraic details omitted for brevity):

$$[\mathbf{Y}_{\Omega^c}|\mathbf{Y}_\Omega, \mathbf{U}, \mathbf{D}, \mathbf{V}, \sigma^2, \eta^2] \sim \mathcal{N}(\mathbf{X}^P_{\Omega^c}, \mathbf{\Sigma}^P_{\Omega^c} + \eta^2\mathbf{I}),$$

(Missing data imputation; see (6.5) and (6.7))

$$[\mathbf{U}|\mathbf{Y}, \mathbf{D}, \mathbf{V}, \sigma^2, \eta^2] \propto \mathrm{etr}\{(\mathbf{Y}\mathbf{V}\mathbf{D})^T\mathbf{U}/\eta^2\}$$

$$
\begin{aligned}
&\sim MF(m_1, R, \mathbf{YVD}/\eta^2), \\
[\mathbf{V}|\mathbf{Y}, \mathbf{U}, \mathbf{D}, \sigma^2, \eta^2] &\propto \text{etr}\{(\mathbf{Y}^T\mathbf{UD})^T\mathbf{V}/\eta^2\} \\
&\sim MF(m_2, R, \mathbf{Y}^T\mathbf{UD}/\eta^2), \\
[\mathbf{D}|\mathbf{Y}, \mathbf{U}, \mathbf{V}, \sigma^2, \eta^2] &\sim QL(\boldsymbol{\mu}, \delta^2) \\
&(\boldsymbol{\mu} = [\sigma^2\mathbf{u}_k^T\mathbf{Y}\mathbf{v}_k/(\eta^2+\sigma^2)]_{k=1}^R,\ \delta^2 = \eta^2\sigma^2/(\eta^2+\sigma^2)) \\
[\sigma^2|\mathbf{Y}, \mathbf{U}, \mathbf{D}, \mathbf{V}, \eta^2] &\sim IG(\alpha_{\sigma^2} + R/2, \beta_{\sigma^2} + \text{tr}(\mathbf{D}^2)/2) \\
[\eta^2|\mathbf{Y}, \mathbf{U}, \mathbf{D}, \mathbf{V}, \sigma^2] &\sim IG(\alpha_{\eta^2} + m_1m_2/2, \\
&\qquad \beta_{\eta^2} + \|\mathbf{Y} - \mathbf{UDV}^T\|_F^2/2).
\end{aligned}
$$

Regarding computation time, it can be shown [236] that the posterior sampling of $\mathbf{U}_t$ and $\mathbf{V}_t$ requires $\mathcal{O}(m_1R^3)$ and $\mathcal{O}(m_2R^3)$ work, and it is also easy to see that the imputation of $\mathbf{Y}_{\Omega^c}$ requires $\mathcal{O}(N^3)$ work. Each iteration of `gibbs.mc` therefore requires $\mathcal{O}\{(m_1 \vee m_2)R^3 + N^3\}$ work (remaining steps have negligible running time in the sense of big-$\mathcal{O}$).

REFERENCES

[1] C. J. Geyer, "Practical Markov chain Monte Carlo", *Statistical Science*, vol. 7, no. 4, pp. 473–483, 1992.

[2] K.-T. Fang and Y. Wang, *Number-Theoretic Methods in Statistics*. CRC Press, 1994, vol. 51.

[3] B. A. Flury, "Principal points", *Biometrika*, vol. 77, no. 1, pp. 33–41, 1990.

[4] S. Graf and H. Luschgy, *Foundations of Quantization for Probability Distributions*. Springer-Verlag Berlin Heidelberg, 2000.

[5] G. Pagès, H. Pham, and J. Printems, "Optimal quantization methods and applications to numerical problems in finance", in *S. T. Rachev (ed.), Handbook of Computational and Numerical Methods in Finance*, Birkhäuser, Boston, 2004, pp. 253–297.

[6] T. Dalenius, "The problem of optimum stratification", *Scandinavian Actuarial Journal*, vol. 1950, no. 3-4, pp. 203–213, 1950.

[7] D. R. Cox, "Note on grouping", *Journal of the American Statistical Association*, vol. 52, no. 280, pp. 543–547, 1957.

[8] S. Lloyd, "Least squares quantization in PCM", *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[9] P. Zador, "Asymptotic quantization error of continuous signals and the quantization dimension", *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 139–149, 1982.

[10] Y. Su, "Asymptotically optimal representative points of bivariate random vectors", *Statistica Sinica*, vol. 10, no. 2, pp. 559–576, 2000.

[11] V. R. Joseph, T. Dasgupta, R. Tuo, and C. F. J. Wu, "Sequential exploration of complex surfaces using minimum energy designs", *Technometrics*, vol. 57, no. 1, pp. 64–74, 2015.

[12] S. V. Borodachov, D. P. Hardin, and E. B. Saff, "Low complexity methods for discretizing manifolds via Riesz energy minimization", *Foundations of Computational Mathematics*, vol. 14, no. 6, pp. 1173–1208, 2014.

[13] G. J. Székely and M. L. Rizzo, "Testing for equal distributions in high dimension", *InterStat*, vol. 5, pp. 1–6, 2004.

[14] A. Kolmogorov, "Sulla determinazione empirica delle leggi di probabilita", *Giorn. Ist. Ital. Attuari*, vol. 4, pp. 1–11, 1933.

[15] J. Dick and F. Pillichshammer, *Digital Nets and Sequences: Discrepancy Theory and Quasi–Monte Carlo Integration*. Cambridge University Press, 2010.

[16] J. Dick, F. Y. Kuo, and I. H. Sloan, "High-dimensional integration: The quasi-Monte Carlo way", *Acta Numerica*, vol. 22, pp. 133–288, 2013.

[17] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, 1992.

[18] F. J. Hickernell, "Goodness-of-fit statistics, discrepancies and robust designs", *Statistics & Probability Letters*, vol. 44, no. 1, pp. 73–78, 1999.

[19] K.-T. Fang, "The uniform design: Application of number-theoretic methods in experimental design", *Acta Math. Appl. Sinica*, vol. 3, no. 4, pp. 363–372, 1980.

[20] K.-T. Fang, X. Lu, Y. Tang, and J. Yin, "Constructions of uniform designs by using resolvable packings and coverings", *Discrete Mathematics*, vol. 274, no. 1, pp. 25–40, 2004.

[21] I. H. Sloan, F. Y. Kuo, and S. Joe, "Constructing randomly shifted lattice rules in weighted Sobolev spaces", *SIAM Journal on Numerical Analysis*, vol. 40, no. 5, pp. 1650–1665, 2002.

[22] D. Nuyens and R. Cools, "Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces", *Mathematics of Computation*, vol. 75, no. 254, pp. 903–920, 2006.

[23] J. A. Nichols and F. Y. Kuo, "Fast CBC construction of randomly shifted lattice rules achieving $o(n^{-1+\delta})$ convergence for unbounded integrands over $\setminus^s$ in weighted spaces with POD weights", *Journal of Complexity*, vol. 30, no. 4, pp. 444–468, 2014.

[24] I. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals", *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 7, no. 4, pp. 784–802, 1967.

[25] A. B. Owen, "Scrambling Sobol' and Niederreiter–Xing points", *Journal of Complexity*, vol. 14, no. 4, pp. 466–489, 1998.

[26] M. Rosenblatt, "Remarks on a multivariate transformation", *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 470–472, 1952.

[27] A. B. Owen and S. D. Tribble, "A quasi-Monte Carlo Metropolis algorithm", *Proceedings of the National Academy of Sciences*, vol. 102, no. 25, pp. 8844–8849, 2005.

[28] M. Girolami and B. Calderhead, "Riemann manifold Langevin and Hamiltonian Monte Carlo methods", *Journal of the Royal Statistical Society: Series B*, vol. 73, no. 2, pp. 123–214, 2011.

[29] F. Hickernell, "A generalized discrepancy and quadrature error bound", *Mathematics of Computation*, vol. 67, no. 221, pp. 299–322, 1998.

[30] G. J. Székely and M. L. Rizzo, "Energy statistics: A class of statistics based on distances", *Journal of Statistical Planning and Inference*, vol. 143, no. 8, pp. 1249–1272, 2013.

[31] I. Gelfand and G. Shilov, *Generalized Functions, Vol. I: Properties and Operations*. Academic Press, New York, 1964.

[32] H. Wendland, *Scattered Data Approximation*. Cambridge University Press, 2005.

[33] E. Di Nezza, G. Palatucci, and E. Valdinoci, "Hitchhiker's guide to the fractional Sobolev spaces", *Bulletin des Sciences Mathématiques*, vol. 136, no. 5, pp. 521–573, 2012.

[34] H. Bahouri, J.-Y. Chemin, and R. Danchin, *Fourier Analysis and Nonlinear Partial Differential Equations*. Springer Science & Business Media, 2011, vol. 343.

[35] S. Mak and V. R. Joseph, "Supplement to "Support points"", 2017.

[36] J. Kiefer, "On large deviations of the empiric df of vector chance variables and a law of the iterated logarithm", *Pacific Journal of Mathematics*, vol. 11, no. 2, pp. 649–660, 1961.

[37] F. Y. Kuo and I. H. Sloan, "Lifting the curse of dimensionality", *Notices of the AMS*, vol. 52, no. 11, pp. 1320–1328, 2005.

[38] H. Tuy, "DC optimization: Theory, methods and algorithms", in *Handbook of Global Optimization*, Springer, 1995, pp. 149–216.

[39] P. D. Tao and L. T. H. An, "Convex analysis approach to DC programming: Theory, algorithms and applications", *Acta Mathematica Vietnamica*, vol. 22, no. 1, pp. 289–355, 1997.

[40] H. Tuy, "A general deterministic approach to global optimization via dc programming", in *J. B. Hiriart-Urruty (ed.), Fermat Days 1985: Mathematics for Optimization*, North-Holland, Amsterdam, 1986, pp. 137–162.

[41] T. Lipp and S. Boyd, "Variations and extension of the convex–concave procedure", *Optimization and Engineering*, vol. 17, no. 2, pp. 1–25, 2016.

[42] A. L. Yuille and A. Rangarajan, "The concave-convex procedure", *Neural Computation*, vol. 15, no. 4, pp. 915–936, 2003.

[43] K. Lange, *MM Optimization Algorithms*. SIAM, 2016.

[44] U. M. Ascher and C. Greif, *A First Course on Numerical Methods*. SIAM, 2011.

[45] J. Mairal, "Stochastic majorization-minimization algorithms for large-scale optimization", in *Advances in Neural Information Processing Systems*, 2013, pp. 2283–2291.

[46] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. SIAM, 2000.

[47] O. Bousquet and L. Bottou, "The tradeoffs of large scale learning", in *Advances in Neural Information Processing Systems*, 2008, pp. 161–168.

[48] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming", *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.

[49] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*. Springer Science & Business Media, 2013.

[50] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator", *ACM Transactions on Modeling and Computer Simulation*, vol. 8, no. 1, pp. 3–30, 1998.

[51] R Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2017. [Online]. Available: `https://www.R-project.org`.

[52] C Dutang and P Savicky, "`randtoolbox`: Generating and testing random numbers", *R package*, 2013.

[53] S. Joe and F. Y. Kuo, "Remark on algorithm 659: Implementing Sobol's quasirandom sequence generator", *ACM Transactions on Mathematical Software*, vol. 29, no. 1, pp. 49–57, 2003.

[54] A. Genz, "Testing multidimensional integration routines", in *Proc. of International Conference on Tools, Methods and Languages for Scientific and Engineering Computation*, Elsevier North-Holland, Inc., 1984, pp. 81–94.

[55] B. A. Worley, "Deterministic uncertainty analysis", Oak Ridge National Laboratories, Tech. Rep. ORNL-6428, 1987.

[56] W. A. Link and M. J. Eaton, "On thinning of chains in MCMC", *Methods in Ecology and Evolution*, vol. 3, no. 1, pp. 112–115, 2012.

[57] N. R. Draper and H. Smith, *Applied Regression Analysis*. John Wiley & Sons, 1981.

[58] B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabilistic programming language", *Journal of Statistical Software*, vol. 76, no. 1, pp. 1–32, 2017.

[59] S. Mak, support*: Support points*, R package version 0.1.0, 2017. [Online]. Available: `https://CRAN.R-project.org/package=support`.

[60] V. R. Joseph, E. Gul, and S. Ba, "Maximum projection designs for computer experiments", *Biometrika*, vol. 102, no. 2, pp. 371–380, 2015.

[61] S. Mak and V. R. Joseph, "Minimax and minimax projection designs using clustering", *Journal of Computational and Graphical Statistics*, 2017, In press.

[62] S. Mak, C.-L. Sung, X. Wang, S.-T. Yeh, Y.-H. Chang, V. R. Joseph, V. Yang, and C. F. J. Wu, "An efficient surrogate model for emulation and physics extraction of

large eddy simulations", *Journal of the American Statistical Association*, 2017, To appear.

[63] S. Mak and Y. Xie, "Maximum entropy low-rank matrix recovery", *arXiv preprint arXiv:1712.03310*, 2017.

[64] R. E. Caflisch, W. J. Morokoff, and A. B. Owen, *Valuation of Mortgage Backed Securities using Brownian Bridges to reduce Effective Dimension*. Department of Mathematics, University of California, Los Angeles, 1997.

[65] I. H. Sloan and H. Woźniakowski, "When are Quasi-Monte Carlo algorithms efficient for high dimensional integrals?", *Journal of Complexity*, vol. 14, no. 1, pp. 1–33, 1998.

[66] G. E. Box and J. S. Hunter, "The 2^{k-p} fractional factorial designs", *Technometrics*, vol. 3, no. 3, pp. 311–351, 1961.

[67] M. Hamada and C. F. J. Wu, "Analysis of designed experiments with complex aliasing", *Journal of Quality Technology*, vol. 24, no. 3, pp. 130–137, 1992.

[68] C. F. J. Wu and M. S. Hamada, *Experiments: Planning, Analysis, and Optimization*. John Wiley & Sons, 2009, vol. 552.

[69] V. R. Joseph, E. Gul, and S. Ba, "Maximum projection designs for computer experiments", *Biometrika*, vol. 102, no. 2, pp. 371–380, 2015.

[70] Y. Chen, M. Welling, and A. Smola, "Super-samples from kernel herding", *arXiv preprint arXiv:1203.3472*, 2012.

[71] F. Bach, S. Lacoste-Julien, and G. Obozinski, "On the equivalence between herding and conditional gradient algorithms", *arXiv preprint arXiv:1203.4523*, 2012.

[72] S. Mak and V. R. Joseph, "Support points", *Annals of Statistics*, 2017, To appear.

[73] G. J. Székely and M. L. Rizzo, "Testing for equal distributions in high dimension", *InterStat*, vol. 5, pp. 1–6, 2004.

[74] Y. Chen, L. Bornn, N. De Freitas, M. Eskelin, J. Fang, and M. Welling, "Herded Gibbs sampling", *Journal of Machine Learning Research*, vol. 17, no. 10, pp. 1–29, 2016.

[75] F. Y. Kuo, C. Schwab, and I. H. Sloan, "Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients", *SIAM Journal on Numerical Analysis*, vol. 50, no. 6, pp. 3351–3374, 2012.

[76] V. R. Joseph, "A Bayesian approach to the design and analysis of fractionated experiments", *Technometrics*, vol. 48, no. 2, pp. 219–229, 2006.

[77] C. A. Micchelli, "Interpolation of scattered data: Distance matrices and conditionally positive definite functions", in *Approximation Theory and Spline Functions*, Springer, 1984, pp. 143–145.

[78] G. Mercier and M. Lennon, "Support vector machines for hyperspectral image classification with spectral-based kernels", in *Geoscience and Remote Sensing Symposium*, IEEE, vol. 1, 2003, pp. 288–290.

[79] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Science & Business Media, 2006.

[80] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization", *Journal of Optimization Theory and Applications*, vol. 109, no. 3, pp. 475–494, 2001.

[81] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso", *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[82] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent", *Journal of Statistical Software*, vol. 33, no. 1, p. 1, 2010.

[83] A. O'Hagan, "Bayes–Hermite quadrature", *Journal of Statistical Planning and Inference*, vol. 29, no. 3, pp. 245–260, 1991.

[84] F.-X. Briol, C. Oates, M. Girolami, and M. A. Osborne, "Frank-Wolfe Bayesian quadrature: Probabilistic integration with theoretical guarantees", in *Advances in Neural Information Processing Systems*, 2015, pp. 1162–1170.

[85] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2013.

[86] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics, 2001, vol. 1.

[87] J. Dick and F. Pillichshammer, "Discrepancy theory and quasi-monte carlo integration", in *A Panorama of Discrepancy Theory*, Springer, 2014, pp. 539–619.

[88] S. Chari and I. Dworkin, "The conditional nature of genetic interactions: The consequences of wild-type backgrounds on mutational interactions in a genome-wide modifier screen", *PLoS Genetics*, vol. 9, no. 8, e1003661, 2013.

[89] D. C. Montgomery, *Design and Analysis of Experiments*. John Wiley & Sons, 2008.

[90] C. F. J. Wu, "Post-Fisherian experimentation: From physical to virtual", *Journal of the American Statistical Association*, vol. 110, no. 510, pp. 612–620, 2015.

[91] D. Finney, "The fractional replication of factorial arrangements", *Annals of Eugenics*, vol. 12, pp. 291–303, 1945.

[92] H. Su and C. F. J. Wu, "CME analysis: A new method for unraveling aliased effects in two-level fractional factorial experiments", *Journal of Quality Technology*, vol. 49, no. 1, pp. 1–10, 2017.

[93] R. Tibshirani, "The lasso method for variable selection in the Cox model", *Statistics in Medicine*, vol. 16, no. 4, pp. 385–395, 1997.

[94] I. E. Frank and J. H. Friedman, "A statistical view of some chemometrics regression tools", *Technometrics*, vol. 35, no. 2, pp. 109–135, 1993.

[95] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties", *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[96] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty", *The Annals of Statistics*, vol. 38, no. 2, pp. 894–942, 2010.

[97] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[98] L. Jacob, G. Obozinski, and J.-P. Vert, "Group lasso with overlap and graph lasso", in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 433–440.

[99] T. T. Wu and K. Lange, "Coordinate descent algorithms for lasso penalized regression", *The Annals of Applied Statistics*, vol. 2, no. 1, pp. 224–244, 2008.

[100] P. Breheny and J. Huang, "Penalized methods for bi-level variable selection", *Statistics and Its Interface*, vol. 2, no. 3, p. 369, 2009.

[101] P. Breheny, "The group exponential lasso for bi-level variable selection", *Biometrics*, vol. 71, no. 3, pp. 731–740, 2015.

[102] P. Zhao and B. Yu, "On model selection consistency of lasso", *The Journal of Machine Learning Research*, vol. 7, pp. 2541–2563, 2006.

[103] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net", *Journal of the Royal Statistical Society: Series B*, vol. 67, no. 2, pp. 301–320, 2005.

[104] R. Mazumder, J. H. Friedman, and T. Hastie, "Sparsenet: Coordinate descent with nonconvex penalties", *Journal of the American Statistical Association*, 2012.

[105] W. J. Fu, "Penalized regressions: The bridge versus the lasso", *Journal of Computational and Graphical Statistics*, vol. 7, no. 3, pp. 397–416, 1998.

[106] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization", *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[107] R. Mazumder, J. H. Friedman, and T. Hastie, "SparseNet: Coordinate descent with nonconvex penalties", *Journal of the American Statistical Association*, vol. 106, no. 495, pp. 1125–1138, 2011.

[108] D. L. Donoho, "De-noising by soft-thresholding", *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[109] K. Lange, *Numerical analysis for statisticians*. Springer Science & Business Media, 2010.

[110] L. Meier, S. Van De Geer, and P. Bühlmann, "The group lasso for logistic regression", *Journal of the Royal Statistical Society: Series B*, vol. 70, no. 1, pp. 53–71, 2008.

[111] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani, "Strong rules for discarding predictors in lasso-type problems", *Journal of the Royal Statistical Society: Series B*, vol. 74, no. 2, pp. 245–266, 2012.

[112] S. Lee and P. Breheny, "Strong rules for nonconvex penalties and their implications for efficient algorithms in high-dimensional regression", *Journal of Computational and Graphical Statistics*, vol. 24, no. 4, pp. 1074–1091, 2015.

[113] R. Tibshirani, "Regression shrinkage and selection via the lasso", *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[114] J. Friedman, T. Hastie, and R. Tibshirani, "GLMNET: Lasso and elastic-net regularized generalized linear models", *R package version 1*, 2009.

[115] J. Bien, J. Taylor, and R. Tibshirani, "A lasso for hierarchical interactions", *The Annals of Statistics*, vol. 41, no. 3, pp. 1111–1141, 2013.

[116] A. L. Rosenbloom, J. R. Joe, R. S. Young, and W. E. Winter, "Emerging epidemic of type 2 diabetes in youth.", *Diabetes Care*, vol. 22, no. 2, pp. 345–354, 1999.

[117] M. H. De Moor, S. M. Van Den Berg, K. J. Verweij, R. F. Krueger, M. Luciano, A. A. Vasquez, L. K. Matteson, J. Derringer, T. Esko, and N. Amin, "Meta-analysis of genome-wide association studies for neuroticism, and the polygenic association with major depressive disorder", *JAMA Psychiatry*, vol. 72, no. 7, pp. 642–650, 2015.

[118] K. Weber, R. Eisman, S. Higgins, L. Morey, A. Patty, M. Tausek, and Z.-B. Zeng, "An analysis of polygenes affecting wing shape on chromosome 2 in Drosophila Melanogaster", *Genetics*, vol. 159, no. 3, pp. 1045–1057, 2001.

[119] H. J. Cordell, "Detecting gene–gene interactions that underlie human diseases", *Nature Reviews Genetics*, vol. 10, no. 6, pp. 392–404, 2009.

[120] R. Lekivetz and B. Jones, "Fast flexible space-filling designs for nonrectangular regions", *Quality and Reliability Engineering International*, vol. 31, no. 5, pp. 829–837, 2015.

[121] W. D. McComb, *The Physics of Fluid Turbulence*. Clarendon Press, 1990.

[122] S. B. Pope, *Turbulent Flows*. Cambridge University Press, Cambridge, 2001.

[123] G. Mathéron, "Principles of geostatistics", *Economic Geology*, vol. 58, no. 8, pp. 1246–1266, 1963.

[124] B. Williams, D. Higdon, J. Gattiker, L. Moore, M. McKay, and S. Keller-McNulty, "Combining experimental data and computer simulations, with an application to flyer plate experiments", *Bayesian Analysis*, vol. 1, no. 4, pp. 765–792, 2006.

[125] J. Rougier, "Efficient emulators for multivariate deterministic functions", *Journal of Computational and Graphical Statistics*, vol. 17, no. 4, pp. 827–843, 2008.

[126] Y. Hung, V. R. Joseph, and S. N. Melkote, "Analysis of computer experiments with functional response", *Technometrics*, vol. 57, no. 1, pp. 35–44, 2015.

[127] K.-T. Fang, R. Li, and A. Sudjianto, *Design and Modeling for Computer Experiments*. CRC Press, 2006.

[128] M. Bayarri, J. Berger, J. Cafeo, G Garcia-Donato, F Liu, J Palomo, R. Parthasarathy, R Paulo, J. Sacks, and D Walsh, "Computer model validation with functional output", *The Annals of Statistics*, vol. 35, no. 5, pp. 1874–1906, 2007.

[129] J. O. Ramsay and B. W. Silverman, *Applied Functional Data Analysis: Methods and Case Studies*. Springer-Verlag, New York, 2002.

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

[131] J. L. Lumley, "The structure of inhomogeneous turbulent flows", *Atmospheric Turbulence and Radio Wave Propagation*, 1967, 166–178.

[132] K. Karhunen, *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. Universitat Helsinki, 1947, vol. 37.

[133] M. Loève, *Probability Theory; Foundations, Random Sequences*. New York: D. Van Nostrand Company, 1955.

[134] D. You, D. D. Ku, and V. Yang, "Acoustic waves in baffled combustion chamber with radial and circumferential blades", *Journal of Propulsion and Power*, vol. 29, no. 6, pp. 1453–1467, 2013.

[135] A Stein and L. Corsten, "Universal kriging and cokriging as a regression procedure", *Biometrics*, vol. 47, no. 2, pp. 575–587, 1991.

[136] N. Zong and V. Yang, "Cryogenic fluid dynamics of pressure swirl injectors at supercritical conditions", *Physics of Fluids*, vol. 20, no. 5, p. 056 103, 2008.

[137] X. Wang, H. Huo, Y. Wang, and V. Yang, "Comprehensive study of cryogenic fluid dynamics of swirl injectors at supercritical conditions", *AIAA Journal*, 2017, In press.

[138] X. Wang, Y. Wang, and V. Yang, "Geometric effects on liquid oxygen/kerosene bi-swirl injector flow dynamics at supercritical conditions", *AIAA Journal*, 2017, In press.

[139] Y.-M. Hu, M. Hendry, and I. S. Heng, "Efficient exploration of multi-modal posterior distributions", *arXiv preprint arXiv:1408.3969*, 2014.

[140] X. Wang, H. Huo, and V. Yang, "Counterflow diffusion flames of oxygen and n-alkane hydrocarbons ($CH_4 - C_{16}H_{34}$) at subcritical and supercritical conditions", *Combustion Science and Technology*, vol. 187, no. 1-2, pp. 60–82, 2015.

[141] J. C. Oefelein and V. Yang, "Modeling high-pressure mixing and combustion processes in liquid rocket engines", *Journal of Propulsion and Power*, vol. 14, no. 5, pp. 843–857, 1998.

[142] N. Zong, H. Meng, S.-Y. Hsieh, and V. Yang, "A numerical study of cryogenic fluid injection and mixing under supercritical conditions", *Physics of Fluids*, vol. 16, no. 12, pp. 4248–4261, 2004.

[143] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code", *Technometrics*, vol. 42, no. 1, pp. 55–61, 1979.

[144] M. D. Morris and T. J. Mitchell, "Exploratory designs for computational experiments", *Journal of Statistical Planning and Inference*, vol. 43, no. 3, pp. 381–402, 1995.

[145] G. G. Stokes, *On the Effect of the Internal Friction of Fluids on the Motion of Pendulums*. Pitt Press, 1851, vol. 9.

[146] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows", *Annual Review of Fluid Mechanics*, vol. 25, no. 1, pp. 539–575, 1993.

[147] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data", *Proceedings of the 23rd ACM National Conference*, 1968, 517–524.

[148] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM, Philadelphia, 1998, vol. 6.

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

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

[151] F. Liu and M. West, "A dynamic modelling strategy for Bayesian computer model emulation", *Bayesian Analysis*, vol. 4, no. 2, pp. 393–411, 2009.

[152] R. J. Hyndman, "Computing and graphing highest density regions", *The American Statistician*, vol. 50, no. 2, pp. 120–126, 1996.

[153] S. Mak, D. Bingham, and Y. Lu, "A regional compound Poisson process for hurricane and tropical storm damage", *Journal of the Royal Statistical Society: Series C*, vol. 65, no. 5, pp. 677–703, 2016.

[154] J. Bien and R. J. Tibshirani, "Sparse estimation of a covariance matrix", *Biometrika*, vol. 98, no. 4, pp. 807–820, 2011.

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

[156] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization", *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[157] V. Yang and W. Anderson, *Liquid Rocket Engine Combustion Instability*. AIAA, 1995, vol. 169.

[158] V. G. Bazarov and V. Yang, "Liquid-propellant rocket engine injector dynamics", *Journal of Propulsion and Power*, vol. 14, no. 5, pp. 797–806, 1998.

[159] I. H. Shames and I. H. Shames, *Mechanics of Fluids*. McGraw-Hill, New York, 1982.

[160] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Academic Press, 2013.

[161] S. Mak and Y. Xie, "Active matrix completion with uncertainty quantification", *arXiv preprint arXiv:1706.08037*, 2017.

[162] R. B. Gramacy and D. W. Apley, "Local Gaussian process approximation for large computer experiments", *Journal of Computational and Graphical Statistics*, vol. 24, no. 2, pp. 561–578, 2015.

[163] C.-L. Sung, R. B. Gramacy, and B. Haaland, "Potentially predictive variance reducing subsample locations in local Gaussian process regression", *Statistica Sinica*, vol. to appear, 2017, arXiv preprint arXiv:1604.04980.

[164] S.-T. Yeh, X. Wang, C.-L. Sung, S. Mak, Y.-H. Chang, L. Zhang, C. Wu, and V. Yang, "Data-driven analysis and common proper orthogonal decomposition (CPOD)-based spatio-temporal emulator for design exploration", *arXiv preprint arXiv:1709.07841*, 2017.

[165] Y.-H. Chang, L. Zhang, X. Wang, S.-T. Yeh, S. Mak, C.-L. Sung, C. Wu, and V. Yang, "Kernel-smoothed proper orthogonal decomposition (KSPOD)-based emulation for prediction of spatiotemporally evolving flow dynamics", *arXiv preprint arXiv:1802.08812*, 2018.

[166] Y. Li, X. Wang, S. Mak, S.-T. Yeh, L.-H. Lin, C.-F. J. Wu, and V. Yang, "A two-stage transfer function identification methodology and its applications to bi-swirl injectors", in *53rd AIAA/SAE/ASEE Joint Propulsion Conference*, 2017, p. 4933.

[167] Y. Li, X. Wang, S. Mak, C.-L. Sung, J. Wu, and V. Yang, "Uncertainty quantification of flame transfer function under a Bayesian framework", in *2018 AIAA Aerospace Sciences Meeting*, 2018, p. 1187.

[168] M. E. Johnson, L. M. Moore, and D. Ylvisaker, "Minimax and maximin distance designs", *Journal of statistical planning and inference*, vol. 26, no. 2, pp. 131–148, 1990.

[169] O. A. Vanli, C. Zhang, A. Nguyen, and B. Wang, "A minimax sensor placement approach for damage detection in composite structures", *Journal of Intelligent Material Systems and Structures*, vol. 23, no. 8, pp. 919–932, 2012.

[170] H. Luss, "On equitable resource allocation problems: A lexicographic minimax approach", *Operations Research*, vol. 47, no. 3, pp. 361–378, 1999.

[171] M. Patan, *Optimal Sensor Networks Scheduling in Identification of Distributed Parameter Systems*. Springer Science & Business Media, 2012, vol. 425.

[172] E. R. van Dam, "Two-dimensional minimax Latin hypercube designs", *Discrete Applied Mathematics*, vol. 156, no. 18, pp. 3483–3493, 2008.

[173] P. John, M. Johnson, L. Moore, and D Ylvisaker, "Minimax distance designs in two-level factorial experiments", *Journal of Statistical Planning and Inference*, vol. 44, no. 2, pp. 249–263, 1995.

[174] M. H. Tan, "Minimax designs for finite design regions", *Technometrics*, vol. 55, no. 3, pp. 346–358, 2013.

[175] S. Lloyd, "Binary block coding", *Bell System Technical Journal*, vol. 36, no. 2, pp. 517–535, 1957.

[176] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design", *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.

[177] G. P. Leung, M. J. Aristizabal, N. J. Krogan, and M. S. Kobor, "Conditional genetic interactions of RTT107, SLX4, and HRQ1 reveal dynamic networks upon dna damage in S. cerevisiae", *G3: Genes—Genomes—Genetics*, vol. 4, no. 6, pp. 1059–1069, 2014.

[178] B. D. Flury, "Estimation of principal points", *Journal of the Royal Statistical Society, Series C*, vol. 42, no. 1, pp. 139–151, 1993.

[179] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function", *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[180] D. Eppstein, "Fast hierarchical clustering and other applications of dynamic closest pairs", *Journal of Experimental Algorithmics*, vol. 5, pp. 1–23, 2000.

[181] S. Banerjee, A. E. Gelfand, A. O. Finley, and H. Sang, "Gaussian predictive process models for large spatial data sets", *Journal of the Royal Statistical Society: Series B*, vol. 70, no. 4, pp. 825–848, 2008.

[182] D. Oser, *Georgia air monitoring*, `http://amp.georgiaair.org/`, Accessed: 2016-10-15, 2016.

[183] F. Nielsen and R. Bhatia, *Matrix Information Geometry*. Springer, 2013.

[184] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM, 2001, vol. 2.

[185] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[186] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $o(1/k^2)$", *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376, 1983.

[187] ——, *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science & Business Media, 2013, vol. 87.

[188] D. Arthur, B. Manthey, and H Roglin, "K-means has polynomial smoothed complexity", in *IEEE Annual Symposium on Foundations of Computer Science*, 2009, pp. 405–414.

[189] A. Bhowmick, "A theoretical analysis of lloyd's algorithm for k-means clustering", PhD thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur, 2009.

[190] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding", in *ACM-SIAM Symposium on Discrete Algorithms*, 2007, pp. 1027–1035.

[191] A. K. Jain, "Data clustering: 50 years beyond k-means", *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[192] G. F. Tzortzis and C. Likas, "The global kernel-means algorithm for clustering in feature space", *IEEE Transactions on Neural Networks*, vol. 20, no. 7, pp. 1181–1194, 2009.

[193] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm", *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[194] D. Van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization", *Evolutionary Computation*, vol. 1, pp. 215–220, 2003.

[195] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.

[196] A. B. Owen, "Randomly permuted (t, m, s)-nets and (t, s)-sequences", in *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, Springer, 1995, pp. 299–317.

[197] S. Mak, minimaxdesign*: Minimax and minimax projection designs*, R package version 0.1.0, 2016. [Online]. Available: `https://CRAN.R-project.org/package=minimaxdesign`.

[198] J Ypma, nloptr*: R interface to NLopt*, R package version 1.0.4, 2014.

[199] T. Krink and M. Løvbjerg, "The lifecycle model: Combining particle swarm optimisation, genetic algorithms and hillclimbers", in *International Conference on Parallel Problem Solving from Nature*, Springer, 2002, pp. 621–630.

[200] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 39, no. 6, pp. 1362–1381, 2009.

[201] N. Natarajan and I. S. Dhillon, "Inductive matrix completion for predicting gene–disease associations", *Bioinformatics*, vol. 30, no. 12, pp. 60–68, 2014.

[202] E. J. Candès and Y. Plan, "Matrix completion with noise", *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[203] E. Candès and B. Recht, "Exact matrix completion via convex optimization", *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[204] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion", *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.

[205] B. Recht, "A simpler approach to matrix completion", *Journal of Machine Learning Research*, vol. 12, pp. 3413–3430, 2011.

[206] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries", *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2980–2998, 2010.

[207] S. Negahban and M. J. Wainwright, "Restricted strong convexity and weighted matrix completion: Optimal bounds with noise", *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 1665–1697, 2012.

[208] C. Croux, P. Filzmoser, and H. Fritz, "Robust sparse principal component analysis", *Technometrics*, vol. 55, no. 2, pp. 202–214, 2013.

[209] A. Bhargava, R. Ganti, and R. Nowak, "Active positive semidefinite matrix completion: Algorithms, theory and applications", in *Artificial Intelligence and Statistics*, 2017, pp. 1349–1357.

[210] N. Ruchansky, M. Crovella, and E. Terzi, "Matrix completion with queries", in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2015, pp. 1025–1034.

[211] M. C. Shewry and H. P. Wynn, "Maximum entropy sampling", *Journal of Applied Statistics*, vol. 14, no. 2, pp. 165–170, 1987.

[212] R. C. Smith, *Uncertainty Quantification: Theory, Implementation, and Applications*. SIAM, 2013.

[213] Y. J. Lim and Y. W. Teh, "Variational Bayesian approach to movie rating prediction", in *Proceedings of Knowledge Discovery and Data Mining (KDD)*, vol. 7, 2007, pp. 15–21.

[214] N. D. Lawrence and R. Urtasun, "Non-linear matrix factorization with Gaussian processes", in *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009, pp. 601–608.

[215] D. P. Palomar and S. Verdú, "Gradient of mutual information in linear vector Gaussian channels", *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 141–154, 2006.

[216] L. Wang, A. Razi, M. Rodrigues, R. Calderbank, and L. Carin, "Nonlinear information-theoretic compressive measurement design", in *International Conference on Machine Learning*, 2014, pp. 1161–1169.

[217] N. Shlezinger, R. Dabora, and Y. C. Eldar, "Measurement matrix design for phase retrieval based on mutual information", *arXiv preprint arXiv:1704.08021*, 2017.

[218] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization", in *Advances in Neural Information Processing Systems*, 2011, pp. 2546–2554.

[219] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms", in *Advances in Neural Information Processing Systems*, 2012, pp. 2951–2959.

[220] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro, and D Sculley, "Google Vizier: A service for black-box optimization", in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2017, pp. 1487–1495.

[221] K. Konyushkova, R. Sznitman, and P. Fua, "Learning active learning from data", in *Advances in Neural Information Processing Systems*, 2017, pp. 4228–4238.

[222] C. E. Rasmussen and C. K. Williams, "Gaussian processes for machine learning", 2006.

[223] A. K. Gupta and D. K. Nagar, *Matrix Variate Distributions*. CRC Press, 1999.

[224] Y. Chikuse, *Statistics on Special Manifolds*. Springer Science & Business Media, 2012.

[225] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis*. Taylor & Francis, 2014, vol. 2.

[226] M. A. Davenport and J. Romberg, "An overview of low-rank matrix recovery from incomplete observations", *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 608–622, 2016.

[227] P. Sebastiani and H. P. Wynn, "Maximum entropy sampling and optimal Bayesian experimental design", *Journal of the Royal Statistical Society, Series B*, vol. 62, no. 1, pp. 145–157, 2000.

[228] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 2012.

[229] S. Prasad, "Certain relations between mutual information and fidelity of statistical estimation", *arXiv preprint arXiv:1010.1508*, 2010.

[230] C. J. Colbourn, T. Klove, and A. C. Ling, "Permutation arrays for powerline communication and mutually orthogonal Latin squares", *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1289–1291, 2004.

[231] S. Huczynska, "Powerline communication and the 36 officers problem", *Philosophical Transactions of the Royal Society of London A*, vol. 364, no. 1849, pp. 3199–3214, 2006.

[232] R. A. Fisher, *The Design of Experiments*. Oliver and Boyd, London, 1937.

[233] E. J. Candes and Y. Plan, "Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements", *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2342–2359, 2011.

[234] J. Shen, "On the singular values of Gaussian random matrices", *Linear Algebra and its Applications*, vol. 326, no. 1-3, pp. 1–14, 2001.

[235] E. P. Wigner, "Characteristic vectors of bordered matrices with infinite dimensions", *Annals of Mathematics*, vol. 62, pp. 548–564, 1955.

[236] P. D. Hoff, "Simulation of the matrix Bingham–von Mises–Fisher distribution, with applications to multivariate and relational data", *Journal of Computational and Graphical Statistics*, vol. 18, no. 2, pp. 438–456, 2009.

[237] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines", *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[238] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications", *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[239] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 721–741, 1984.

[240] R. J. Little and D. B. Rubin, *Statistical Analysis with Missing Data*. John Wiley & Sons, 2014.

[241] M. T. Jacobson and P. Matthews, "Generating uniformly distributed random Latin squares", *Journal of Combinatorial Designs*, vol. 4, no. 6, pp. 405–437, 1996.

[242] B. P. Carlin and T. A. Louis, *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall, 2000, vol. 17.

[243] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming", 2008.

[244] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm", *Information Retrieval*, vol. 4, no. 2, pp. 133–151, 2001.

[245] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments", *Statistical Science*, pp. 409–423, 1989.

[246] S. Mak and V. R. Joseph, "Projected support points, with application to optimal MCMC reduction", *arXiv preprint arXiv:1708.06897*, 2017.

[247] S. Mak and C. Wu, "Cmenet: A new method for bi-level variable selection of conditional main effects", *Journal of the American Statistical Association*, 2018, To appear.

[248] L. E. Ghaoui, V. Viallon, and T. Rabbani, "Safe feature elimination for the lasso and sparse supervised learning problems", *arXiv preprint arXiv:1009.4219*, 2010.

[249] G. J. Székely, "E-statistics: The energy of statistical samples", Bowling Green State University, Department of Mathematics and Statistics, Tech. Rep. 03-05, 2003.

[250] S. I. Resnick, *A Probability Path*. Springer Science & Business Media, 1999.

[251] H. L. Royden and P. Fitzpatrick, *Real Analysis*. Macmillan New York, 2010.

[252] G. R. Shorack, *Probability for Statisticians*. Springer Science & Business Media, 2000.

[253] J. K. Hunter and B. Nachtergaele, *Applied Analysis*. World Scientific Publishing, 2001.

[254] R. J. Serfling, *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons, 2009.

[255] V. Korolyuk and Y. V. Borovskikh, "Convergence rate for degenerate von Mises functionals", *Theory of Probability & Its Applications*, vol. 33, no. 1, pp. 125–135, 1989.

[256] R. Paley and A Zygmund, "On some series of functions", in *Mathematical Proceedings of the Cambridge Philosophical Society*, Cambridge Univ Press, vol. 26, 1930, pp. 337–357.

[257] G. Yang, "The energy goodness-of-fit test for univariate stable distributions", PhD thesis, Bowling Green State University, 2012.

[258] F. Cobos and T. Kühn, "Eigenvalues of integral operators with positive definite kernels satisfying integrated Hölder conditions over metric compacta", *Journal of Approximation Theory*, vol. 63, no. 1, pp. 39–55, 1990.

[259] J. Ferreira and V. Menegatto, "Eigenvalues of integral operators defined by smooth positive definite kernels", *Integral Equations and Operator Theory*, vol. 64, no. 1, pp. 61–81, 2009.

[260] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization", *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[261] N. Aronszajn, "Theory of reproducing kernels", *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.

[262] H. Q. Minh, "Some properties of Gaussian reproducing kernel Hilbert spaces and their implications for function approximation and learning theory", *Constructive Approximation*, vol. 32, no. 2, pp. 307–338, 2010.

[263] A. Stuart and J. Ord, *Kendall's Advanced Theory of Statistics, Volume 1: Distribution Theory*. Arnold London, 1994.

[264] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*. Springer Science & Business Media, 2008, vol. 116.

[265] J. P. Imhof, "Computing the distribution of quadratic forms in normal variables", *Biometrika*, vol. 48, no. 3/4, pp. 419–426, 1961.

[266] R. B. Davies, "Numerical inversion of a characteristic function", *Biometrika*, vol. 60, no. 2, pp. 415–417, 1973.

[267] ——, "Algorithm AS 155: The distribution of a linear combination of χ^2 random variables", *Journal of the Royal Statistical Society. Series C*, vol. 29, no. 3, pp. 323–333, 1980.

[268] A. Castaño-Martínez and F. López-Blázquez, "Distribution of a sum of weighted noncentral chi-square variables", *TEST*, vol. 14, no. 2, pp. 397–415, 2005.

[269] H. Liu, Y. Tang, and H. H. Zhang, "A new chi-square approximation to the distribution of non-negative definite quadratic forms in non-central normal variables", *Computational Statistics and Data Analysis*, vol. 53, no. 4, pp. 853–856, 2009.

[270] R Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2015. [Online]. Available: `https://www.R-project.org/`.

[271] P. Duchesne and P. L. de Micheaux, "Computing the distribution of quadratic forms: Further comparisons between the Liu-Tang-Zhang approximation and exact methods", *Computational Statistics and Data Analysis*, vol. 54, no. 4, pp. 858–862, 2010.

[272] D. P. Bertsekas, *Nonlinear Programming*. Athena scientific, 1999.

[273] Y. Nesterov, "Gradient methods for minimizing composite objective function", UCL, Tech. Rep., 2007.

[274] K. Hoffman and R. Kunze, *Linear Algebra*. Englewood Cliffs, New Jersey, 1971.