CALIBRATION AND OPTIMIZATION OF COMPUTER MODELS WITH APPLICATIONS TO ACOUSTICS AND MATERIALS DISCOVERY

A Dissertation
Presented to
The Academic Faculty

By

Arvind Krishna

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial & Systems Engineering

Georgia Institute of Technology

August 2021

CALIBRATION AND OPTIMIZATION OF COMPUTER MODELS WITH APPLICATIONS TO ACOUSTICS AND MATERIALS DISCOVERY

Approved by:

Dr. Roshan V. Joseph, Advisor
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. C. F. Jeff Wu
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Xiaoming Huo
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. William Brenneman
Statistics & Data Science Department
Procter & Gamble

Dr. Rampi Ramprasad
School of Material Science & Engineering
Georgia Institute of Technology

Dr. Chengzhi Shi
School of Mechanical Engineering
Georgia Institute of Technology

Date approved: July 14, 2021

For my parents and elder brother

ACKNOWLEDGMENTS

Failures are very common and frequent in the journey of a PhD student. It would have been impossible for me to write my PhD dissertation without a solid support in various forms from several people. I am humbled to be among the few fortunate individuals who received this support. I would like to express my sincere regards and appreciation to all the people who motivated me to begin my PhD journey, as well as those who supported me during it.

First, I would like to express my deep gratitude towards my advisor, Prof. V. Roshan Joseph who guided me with utmost care and patience throughout the past five years. Under his guidance, I grew from a naive researcher to a mature one. His hands-on approach and attention to detail was key in helping me realize my shortcomings and making amends. Despite numerous setbacks, his belief in my ability helped me maintain enthusiasm towards my research throughout the five year period.

I would like to thank the committee members for their support at critical moments of my PhD. I am extremely grateful to Dr. Brenneman for providing me with constructive feedback and several insights not only on my research, but also on building an academic career. I deeply appreciate the opportunities provided to me by Prof. Rampi Ramprasad and Prof. Chengzhi Shi for interdisciplinary research. I am very thankful to Prof. Jeff Wu and Prof. Xiaoming Huo for their insightful comments on my dissertation proposal and advice regarding an academic career.

I would thankful to the U.S. National Science Foundation grants DMS-1712642 and CMMI-1921646 for supporting my research.

I am indebted to Simon Mak for patiently guiding me in my first research project, and providing several important writing tips. Discussions with Simon enhanced my critical thinking abilities. I would also like to extend my appreciation towards my interdisciplinary research collaborators, Huan Tran and Steven Craig, for helping me with the domain knowl-

edge necessary to implement my ideas. I am also extremely grateful to Prof. Ramesh Singh, from IIT Bombay, for providing me the undergraduate research experience, which gave me the confidence to pursue a doctorate degree.

I feel very fortunate to have an amazing group of friends who made my PhD an enjoyable experience, filled with laughter and fun. They have been the closest point of contact during the five year period, and I would not have made it without them. I treasure the incredible friendships with Prashant Kumar, Li-Hsiang Lin, Prasoon Suchandra, Sushil Varma, Namjoon Suh, Jaekang Kim, Hua Jiang, Wendy Kohn, Ana Maria, and Digvijay Boob.

Finally, I would like to express my deepest regards towards my parents and my brother who always motivated me to pursue education up to the highest level since I was in middle school. My mother, being a mathematics teacher, provided me the motivation to pursue a quantitative discipline. On the other hand, my elder brother provided me an exposure to the field of engineering. This dissertation is dedicated to them.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

In the past, physical systems/processes/phenomena were studied using expensive and time-consuming physical experiments. However, with the advancement of computational resources, computer models are now used extensively to minimize the need for such experimentation. A computer model provides a cheap alternative to explore the behavior of physical processes in desired scenarios and make inferences. This thesis begins with the topic of model calibration, a method to estimate unknown model parameters, and adjust the model to mimic reality. This is followed by a couple of applications of computer models in the field of material informatics and acoustic metasurface design. Novel machine learning algorithms are developed that leverage the computer models for efficient exploration of the physical processes, optimization of parameters of interest, and making inferences.

In Chapter 1, we propose a novel methodology to obtain a robust experimental design for model calibration. A computer model can be used for predicting an output only after specifying the values of some unknown physical constants known as calibration parameters. The unknown calibration parameters can be estimated from real data by conducting physical experiments. This chapter presents an approach to optimally design such a physical experiment. The problem of optimally designing a physical experiment, using a computer model, is similar to the problem of finding an optimal design for fitting nonlinear models. However, the problem is more challenging than the existing work on nonlinear optimal design because of the possibility of model discrepancy, that is, the computer model may not be an accurate representation of the true underlying model. Therefore, we propose an optimal design approach that is robust to potential model discrepancies. We show that our designs are better than the commonly used physical experimental designs that do not make use of the information contained in the computer model and other nonlinear optimal designs that ignore potential model discrepancies. We illustrate our approach using a toy example and a real example from the Procter & Gamble company.

In Chapter 2, we present a novel machine learning algorithm for discovering new materials crystal structure. A material is (thermodynamically) stable and exists naturally when its building blocks, i.e., the constituent atoms, are arranged so that the potential energy is (globally) minimized. We aim to find such minimum energy configurations, to discover a new crystal structure. We leverage density functional theory (DFT) to compute the potential energy for a given configuration of the atoms. The problem is challenging because there are infinitely large number of configurations, the DFT code for computing the energy is expensive, and the potential energy surface is highly non-linear and multi-modal. We propose a novel expansion-exploration-exploitation framework to find the global minimum. The space spanned by a few known crystal structure configurations is expanded to obtain a candidate set of configurations. A key feature of this step is that it tends to generate a space-filling design without the knowledge of the boundaries of the domain space. Once a candidate set of configurations is obtained, it is explored and exploited simultaneously, using Bayesian optimization, to find the global minimum of potential energy. Gaussian Process modeling along with the Expected Improvement algorithm is used to iteratively update the model and guide the search towards the global minimum. We show the effectiveness of our methodology on toy examples and a real problem of predicting the crystal structure configuration of Al_8.

In Chapter 3, we address the problem of designing acoustic metasurfaces for independent amplitude and phase control of acoustic waves. Acoustic metasurfaces are material structures of subwavelength thickness that are used for modulating propagating sound waves. Several applications of acoustic metasurfaces, such as non-invasive biomedical treatments, require independent phase and amplitude modulation of the reflected and transmitted waves. These reflection and transmission outputs (or acoustic outputs) are governed by the geometry of the acoustic metasurface. We model the geometry of the metasurface as a unit cell with mn equal-sized square shaped elements, or a grid-size of $m \times n$. Each element can either be empty or filled with solid material leading to a total of 2^{mn} unique

geometries! This makes it challenging to identify the relevant geometries for obtaining the desired range of acoustic outputs, which are simulated using the COMSOL Multiphysics software. We leverage the expansion algorithm developed in Chapter 2 to start with a few geometries and iteratively add geometries to the set such that they span the entire range of acoustic outputs using only a small fraction of the total number of possible geometries. The algorithm is modified to identify and eliminate redundancy in the chosen geometries due to various kinds of symmetry and other factors. With our modified expansion algorithm, we were able to identify the smallest grid-size necessary for spanning the entire space of the acoustic outputs using only around $24,000$ or $4.2 \times 10^{-9}\%$ of the total possible number of distinct simulations.

CHAPTER 1

ROBUST EXPERIMENTAL DESIGNS FOR MODEL CALIBRATION

1.1 Introduction

A physical system can be explored or optimized by conducting experiments, but they can be expensive and time consuming. The whole field of experimental design in statistics is focused on how to perform these experiments in an efficient way so that maximum information about the system can be obtained with minimum cost [1]. Another way to reduce the experimental cost is to develop mathematical models that can mimic the physical system and explore the system through simulations [2]. These mathematical models can be very complex, such as a system of partial differential equations, which needs to be solved numerically. Computer implementation of the mathematical model using a numerical solver is sometimes called a computer model. Exploration using computer models is useful and can provide conclusive results only if they are good in representing the physical system. However, the mathematical model and thus, the computer model is only an approximation to the complex phenomenon that we are trying to explore in the physical system. Therefore, the computer simulations should only be used to assist the physical experimentation and physical experiments should always be performed to validate the computer models. This article examines how to perform a physical experiment when a computer model is available to the investigator.

A computer model can contain unknown parameters. Choosing them based on the physical experimental data can make the computer models closer to reality. This approach is known as model calibration and the unknown parameters are often referred to as calibration parameters [3]. Therefore, one possible approach to physical experiments is to design them in such a way that the calibration parameters can be estimated efficiently from data.

Since the computer models are often nonlinear in the calibration parameters, one can use results from the nonlinear optimal design theory to design such experiments [4]. One of the major pitfalls of this approach is that the optimal design is overly dependent on the computer model and is not robust against possible model violations. However, detecting such possible violations is at the core of the model calibration problem and is our main aim. Thus, we need to design experiments that account for the computer model but at the same time are robust to the misspecifications of the model.

The importance of designing experiments robust to the model assumptions has long been recognized in the literature since [5]. A good account of the follow-up research on their seminal paper and other related developments can be found in the review by [6]. Unfortunately, most of these approaches rely on an alternative class of possible models, the specification of which is non-trivial. Therefore, the research in this area has not led to practically implementable solutions other than in very simple settings such as adding a center point when fitting a plane, etc. Moreover, the literature seems to be scarce on model-robust designs for nonlinear models. Nonlinear models add complexity to the problem because the optimal designs become functions of the unknown parameters. Robust designs can be developed by expressing the uncertainties in the unknown parameters through a prior distribution and using Bayesian optimal designs [7] or robust-Bayesian optimal designs [8]. Introducing model uncertainty into this framework makes the problem even harder to solve. Furthermore, most of the existing work focuses on simple nonlinear models such as a logistic regression model and not on the complex computer models that we are interested in. Computationally expensive computer models add another layer of complexity because they are known only in the places where the computer simulations have been performed. Therefore, we also need to entertain uncertainties arising due to incomplete knowledge about the true computer model output.

We are not the first to look into the problem of designing physical experiments when computer models are available. Based on the Bayesian model calibration framework of

[9], optimal designs for both computer and physical experiments using integrated mean squared prediction error have been proposed by [10]. However, it has been recognized that the Kennedy and O'Hagan model has severe identifiability issues [11, 12] and so, optimal designs based on their model can inherit similar problems. [13] have proposed designing physical experiments to mitigate the identifiability issues in the Kennedy and O'Hagan model, but their procedure is computationally intensive. In this article, we will propose a much simpler approach to deal with the identifiability issue. [14] and [15] have proposed follow-up experimental designs for model calibration, but not an initial design that we plan to develop here.

This article is organized as follows. In Section 1.2, we discuss the methodology of obtaining a robust experimental design for model calibration. In Section 1.3, we perform simulations on a toy example to illustrate the robustness of our proposed design to model discrepancy. In Section 1.4, we apply our design methodology to a problem relating to the diaper line from the Procter & Gamble (P&G) company. We conclude the chapter with some remarks in Section 1.5.

1.2 Robust Experimental Designs

We will first explain how to develop experimental designs that are robust to model-form uncertainties. Then we will explain how to make them robust to parameter uncertainties. Finally, we will explain how to incorporate computer model approximation uncertainties into this framework. These are now discussed in the following three subsections.

1.2.1 Model-form uncertainties

Let y be the output of the physical system and $\boldsymbol{x} = \{x_1, ..., x_p\}$ the set of inputs. Following [9], we model the output as

$$y = f(\boldsymbol{x}; \boldsymbol{\eta}) + \delta(\boldsymbol{x}) + \epsilon, \tag{1.1}$$

where $f(\cdot;\cdot)$ is the computer model, $\boldsymbol{\eta} = \{\eta_1, ..., \eta_q\}$ the set of unknown calibration parameters, $\delta(\boldsymbol{x})$ the discrepancy function, and $\epsilon \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$ the random error. For the moment, we will assume that the computer model is an easy-to-evaluate function or that the complex computer model has been replaced by an easy-to-evaluate surrogate model whose uncertainties can be ignored. Later we will see how such uncertainties can also be included in our approach.

As mentioned earlier, the model in Equation 1.1 has identifiability issues in the sense that for any value of $\boldsymbol{\eta}$ we can find a $\delta(\boldsymbol{x})$ to get the same prediction [11, 12] and thus, $\boldsymbol{\eta}$ and $\delta(\boldsymbol{x})$ cannot be estimated based on the data on y alone unless some additional assumptions are imposed. Therefore, we will not directly use this model for developing the robust designs. We will put some belief in the computer model and hope that, if properly calibrated, we will not need the discrepancy term. Thus, we will first find an optimal design for estimating $\boldsymbol{\eta}$ ignoring the model discrepancy term. We will then separately find an optimal design for estimating $\delta(\boldsymbol{x})$ and then integrate them together. This approach follows the sequential model building strategy proposed by [16], where $\boldsymbol{\eta}$ is estimated by first ignoring the discrepancy. The discrepancy term is added only if it is necessary and if added, it is estimated by fixing $\boldsymbol{\eta}$ at its initial estimate. This mitigates the identifiability issues that are present in a joint estimation procedure.

Thus, first consider the case with no discrepancy, that is, $\delta(\boldsymbol{x}) = 0$. Our aim is to choose a design to efficiently estimate $\boldsymbol{\eta}$ from the model

$$y = f(\boldsymbol{x}; \boldsymbol{\eta}) + \epsilon. \tag{1.2}$$

Suppose we have a total budget of N runs. We will use n runs to efficiently estimate $\boldsymbol{\eta}$ and the remaining $N - n$ runs to estimate the discrepancy function. Choice of n depends on how much confidence we have in the computer model form. If we are fully confident that there is no discrepancy, then $n = N$. On the other hand, if we have no confidence in

the computer model, then $n = 0$, that is, we will not use the computer model to design the physical experiment. Let $\gamma \in [0, 1]$ be a parameter that measures the experimenter's confidence in the computer model form. Then, $n = \gamma N$, the nearest integer to γN. In the absence of any knowledge about the confidence in the computer model, we recommend choosing $\gamma = 2q/N$. This recommendation is based on the fact that at least q runs are needed to estimate q parameters in the nonlinear model [e.g. 17]. Thus using twice the minimum number of runs, we are likely to have two replicates for each run, which can be used for estimating the unknown error variance, σ^2.

The *approximate* optimal design for estimating $\boldsymbol{\eta}$ can be viewed as a discrete probability distribution in the experimental region $\mathcal{X}$ at points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ with probabilities $\{w_1, \ldots, w_n\}$ [18]. Denote the approximate optimal design (or the distribution) by ξ, which contains the design points as well as their probabilities. The Fisher Information matrix of $\boldsymbol{\eta}$ is given by:

$$M(\xi; \boldsymbol{\eta}) = \sum_{i=1}^{n} w_i \nabla f(\boldsymbol{x}_i; \boldsymbol{\eta}) \nabla f(\boldsymbol{x}_i; \boldsymbol{\eta})^T, \tag{1.3}$$

where $\nabla f(\boldsymbol{x}_i; \boldsymbol{\eta})^T = \left(\frac{\partial f(\boldsymbol{x}_1; \boldsymbol{\eta})}{\partial \eta_1}, \ldots, \frac{\partial f(\boldsymbol{x}_1; \boldsymbol{\eta})}{\partial \eta_q} \right)$. Now an approximate locally D-optimal design can be obtained by maximizing the determinant of $M(\boldsymbol{\eta})$ for a given value of $\boldsymbol{\eta}$. Suppose $\boldsymbol{\eta}_0$ is a guess value of $\boldsymbol{\eta}$. Then, we can obtain the approximate design as

$$\xi^* = \arg\max_{\xi} |M(\xi; \boldsymbol{\eta}_0)|. \tag{1.4}$$

In general, this is a very difficult optimization problem. Here we will use the metaheuristic algorithm proposed in [17], which is implemented in the R package `ICAOD` [19].

The approximate design can be converted to an exact design by rounding $nw_1, \ldots, nw_n$ to the nearest integers. However, the total number of design points after rounding need not be equal to n unless special care is taken [20]. Instead of the rounding approach, here we propose a resampling-based approach, that is, sample with replacement from $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ with probabilities $\{w_1, \ldots, w_n\}$. However, the commonly used random resampling meth-

ods are not suitable for generating an optimal design because the results can vary due to the randomness involved in sampling. [21] recently proposed a deterministic resampling method known as Importance Support Points (ISP), which finds an optimal set of samples with equal weights to approximate a discrete probability distribution. The optimal samples are obtained as the solution to the following optimization problem:

$$\{\tilde{\boldsymbol{x}}_i\}_{i=1}^n \in \underset{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_n \in \{\boldsymbol{x}_j\}_{j=1}^n}{\arg\min} \frac{2}{n}\sum_{i=1}^n\sum_{j=1}^n w_j\|\boldsymbol{u}_i-\boldsymbol{x}_j\|_2 - \frac{1}{n^2}\sum_{i=1}^n\sum_{j=1}^n\|\boldsymbol{u}_i-\boldsymbol{u}_j\|_2 \ . \quad (1.5)$$

Thus, we obtain the final design $\mathcal{D}_\eta = \{\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n\}$ for efficiently estimating $\boldsymbol{\eta}$ by doing an ISP resample on ξ^* in which some of the points may be replicated.

Now that we have an optimal design to estimate the calibration parameters, we can focus our attention on the design to estimate the potential model discrepancy, $\delta(\boldsymbol{x})$. Let us denote the design by $\mathcal{D}_\delta$, which contains $N-n$ runs. So the final design will be $\mathcal{D} = \mathcal{D}_\eta \bigcup \mathcal{D}_\delta$.

[9] proposed to nonparametrically estimate $\delta(\boldsymbol{x})$ using a Gaussian process. However, a Gaussian process model contains a set of unknown correlation parameters. These correlation parameters are nonlinear, which brings up the same issue as the calibration parameters. As before, we can guess the values of the correlation parameters and try to develop locally optimal designs. However, there is something peculiar about these correlation parameters. The optimal design criteria such as the integrated mean squared error [10] are dominated by settings that produce low values of correlations [22]. Thus, we only need to focus on a setting of the correlation parameters that minimizes the correlation. [23] have shown that in such limiting cases the D-optimal designs will reduce to maximin designs and G-optimal designs to minimax designs. In fact, these space-filling designs can be independently motivated using geometric considerations and are known to be robust to modeling choices. Since our objective is to develop model-robust designs, they seem to be a perfect fit for our problem. Besides maximin and minimax, there are many choices for space-filling de-

signs [24]. In this article, we will illustrate the methodology using maximum projection (MaxPro) designs [25], but we will keep this choice flexible for the experimenter. It is important to note that these space-filling designs should be generated to optimally *augment* the existing points in $\mathcal{D}_\eta$ and should not be generated independent of $\mathcal{D}_\eta$.

The proposed method for generating the design is summarized in Algorithm 1.

Algorithm 1 : Generating the physical experimental design, accounting for model-form uncertainties.

1: Input $f(\boldsymbol{x}, \boldsymbol{\eta}), N, \gamma, \boldsymbol{\eta}_0$
2: $n \leftarrow \gamma N$
3: Find ξ^*, as in (Equation 1.4) {`R` package: `ICAOD` }
4: Find $\mathcal{D}_\eta$, as in (Equation 1.5), using ISP resampling on ξ^*
5: Find $\mathcal{D}_\delta$, a space-filling design with $N - n$ runs, to augment $\mathcal{D}_\eta$ {`R` package: `MaxPro` }
6: Output $\mathcal{D}_\eta \bigcup \mathcal{D}_\delta$

As a simple example, suppose the computer model is a linear model given by $f(\boldsymbol{x}; \boldsymbol{\eta}) = \eta_0 + \eta_1 x_1 + \eta_2 x_2 + \eta_3 x_1 x_2$. D-optimal design for estimating this model is a 2^2 full factorial design. Suppose we have a total budget for $N = 13$ runs, and there is no knowledge of the confidence in the computer model. If we choose $\gamma = 2q/N = 8/N$, then $n = \gamma N = 8$ and we will have two replicates for each of the four points in the 2^2 full factorial design. The remaining $N - n = 5$ runs can be chosen to augment the eight runs using a space-filling criterion. For example, if we use the MaxPro criterion, then the augmented design can be obtained sequentially by adding one point at a time using the `MaxProAugment` function in the R package `MaxPro` [26].

Figure 1.1 (right) shows the 13-run design obtained using the Maxpro design for augmentation along with maximin (left) and minimax (center) augmentation strategies. We can see that all these designs will provide protection against possible departures from the linear model assumption. At first glance, the maximin design seems most suitable for a physical experiment as it has fewer factor levels. However, in a high-dimensional space, this design may not give a good validation set as the points will occupy mostly the corners and boundaries of the hypercube. On the other hand, a MaxPro design simultaneously

ensures space-fillingness in the full dimensional space as well as in all lower dimensional subspaces. Moreover, it can easily be extended to incorporate qualitative factors [27]. However, a disadvantage of the MaxPro design is that it can produce too many levels for the factors, which can be inconvenient for a physical experiment. One quick remedy to this problem is to treat the factors as discrete-numeric with a specified number of levels in the `MaxProAugment` function [26].

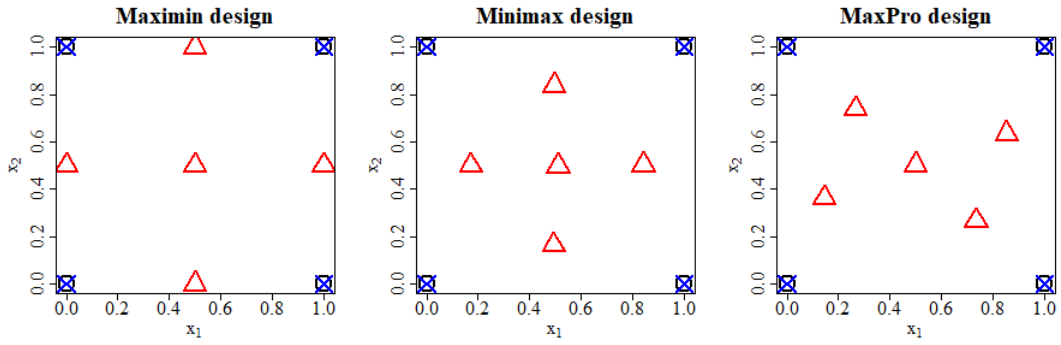

Figure 1.1: Three model-robust designs in 13 runs for estimating a linear model in two variables. The replicates of the optimal design points are shown as circles and crosses, and the space-filling points as triangles.

Now consider a nonlinear model:

$$f(\boldsymbol{x};\eta) = \exp\{-\eta(x_1 - 1.5x_2)^2\} + \exp\{-2\eta(x_1 + x_2 - 0.7)^2\}, \tag{1.6}$$

where $x_1, x_2 \in [0,1]^2$. Suppose that we have a budget to perform eight physical experiments. Assume $\eta_0 = 0.5$, and that there is no information about the confidence in the computer model. Then, we choose $\gamma = 2q/N = 2/N$. This leads to $n = \gamma N = 2$. Then, the exact locally D-optimal optimal design is a one-point design at $\{(0.50, 1.00)\}$ with two replicates. The remaining six runs are obtained by augmenting these two points with the MaxPro design. The final model-robust optimal design is shown in Figure 1.2 (left). For comparison, we have also shown a 2^2 design with two replicates in Figure 1.2 (right), which would be a reasonable choice to make when we don't have a computer model. Furthermore,

we also show the D-optimal design with eight replicates in Figure 1.2 (center).We call this design as the "pure computer model" design.

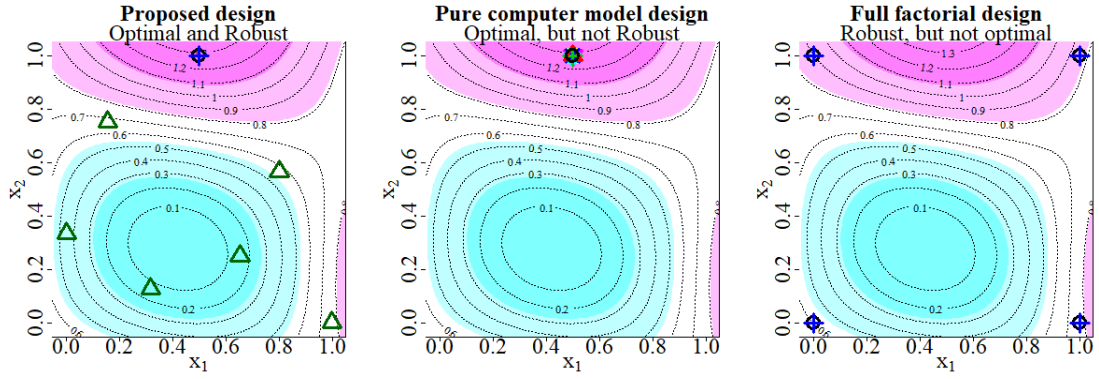

Figure 1.2: Comparison of our proposed design with the "pure computer model" design, and full factorial design, over the computer model gradient contour. The replicates of the D-optimal design points are shown as circles and crosses and the space-filling points as triangles.

If there is no model discrepancy, the "pure computer model" design is likely to provide the most accurate estimate of η. On the other hand, the full factorial design is likely to give the least accurate estimate of η. In the presence of model discrepancy, the "pure computer model" design is likely to perform poorly as it does not contain any space-filling points for estimating model discrepancy. However, in both cases - presence or absence of model discrepancy - our proposed design, though it may not be the best, is likely to provide a relatively accurate calibrated model. This is because it contains both - the D-optimal design points optimal for estimating η and the space-filling design points optimal for estimating the model discrepancy.

1.2.2 Parameter uncertainties

A weakness of the locally optimal designs is that the solution depends on the guessed value of $\boldsymbol{\eta}$. If the guessed value is not close to the (unknown) true value, the results may not be accurate especially when the model is highly nonlinear. As discussed in the introduction, a natural way to address these uncertainties is to use a Bayesian approach by placing a prior

on $\boldsymbol{\eta}$. So let $p(\boldsymbol{\eta})$ be the prior distribution. The $\boldsymbol{\eta}_0$ that we used in the previous section could be viewed as the mean or mode of this prior distribution.

Following [8], now we will generate multiple values for $\boldsymbol{\eta}$:

$$\boldsymbol{\eta}_i \sim p(\boldsymbol{\eta}), \quad i = 1, \ldots, m.$$

We can find approximate locally D-optimal designs for each of these values as described in the previous section to obtain $\xi_1, \ldots, \xi_m$. Thus, we have a total of nm points with nm probabilities. After re-scaling all the weights to sum to one, we can again use the ISP resampling method to obtain the desired n-point optimal design $\mathcal{D}_\eta$.

As we need to find ξ^i for $i = 1, \ldots, m$, the procedure is much more computationally expensive than before. One idea to reduce the computational burden is to sample $\boldsymbol{\eta}_i$ using support points [28] instead of random values from $p(\boldsymbol{\eta})$, because support points will be able to represent the prior distribution with fewer points than a Monte Carlo sample. Thus we can use a much smaller m. The procedure is summarized in Algorithm 2.

Algorithm 2 : Generating the physical experimental design, accounting for model-form and parameter uncertainties.

Input $f(\boldsymbol{x}, \boldsymbol{\eta}), N, \gamma, p(\boldsymbol{\eta})$
$n \leftarrow \gamma N$
Find m realizations of $\boldsymbol{\eta}$ from $p(\boldsymbol{\eta})$ {`R` package: `support`}
for $i \in \{1, \cdots, m\}$ **do**
    $\boldsymbol{\eta}_0 \leftarrow \boldsymbol{\eta}_i$
    Find ξ_i^*, as in (Equation 1.4) {`R` package: `ICAOD` }
    $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i / m$
end for
Find $\mathcal{D}_\eta$, as in (Equation 1.5), using ISP resampling on ξ^*
Find $\mathcal{D}_\delta$, a space-filling design with $N - n$ runs, to augment $\mathcal{D}_\eta$ {`R` package: `MaxPro` }
Output $\mathcal{D}_\eta \bigcup \mathcal{D}_\delta$

Consider again the toy example with one calibration parameter used in the previous section. We assume that the calibration parameter has a prior distribution $\eta \sim \mathcal{N}(0.5,\, 0.2^2)$. We obtain $m = 20$ support points to represent this distribution using the R package

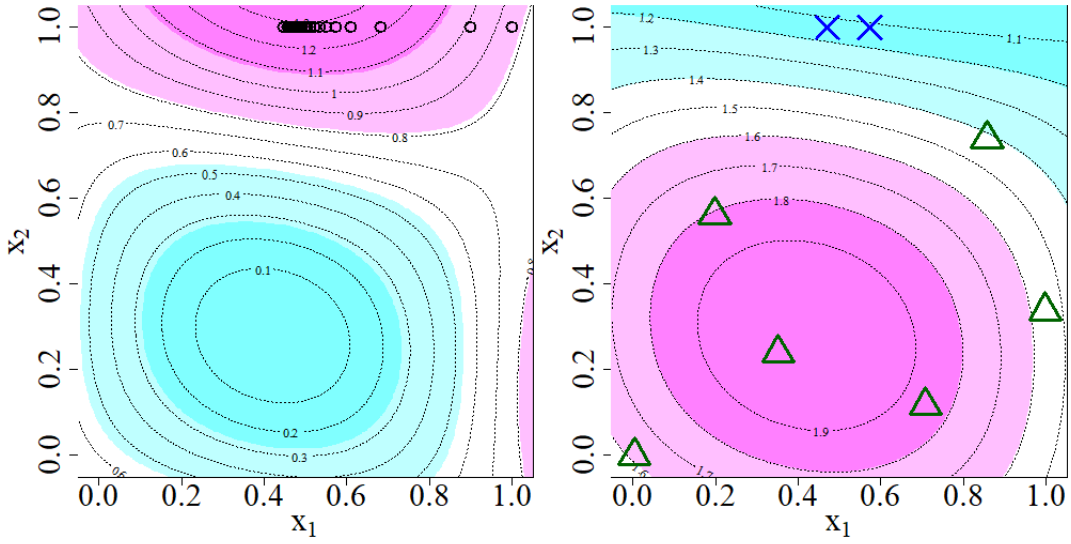

Figure 1.3: (left): Approximate locally D-optimal design points incorporating parameter uncertainties shown over the gradient contour of the computer model for $\eta = 0.5$; (right): The desired $N = 8$-run design over the computer model contour for $\eta = 0.5$, the crosses correspond to the exact D-optimal design for estimating $\boldsymbol{\eta}$, and the triangles correspond to the space-filling design.

`support` [29]. For each of the $m = 20$ realizations, we obtain two-point approximate locally D-optimal design $\xi_i^*, i \in \{1, \cdots, m\}$. The points corresponding to the $m = 20$ approximate optimal designs obtained are shown in Figure 1.3 (left).

Now we use ISP resampling to obtain $n = 2$ runs as shown by crosses in Figure 1.3 (right). This two-run design is augmented with the MaxPro design, shown by triangles in Figure 1.3 (right), to obtain the desired 8-run design. We observe that the design obtained is somewhat similar to the one obtained without incorporating parameter uncertainties in Figure 1.2 (left). However, unlike the design in Figure 1.2 (left), the two points of the D-optimal design for estimating η are a bit farther apart (instead of overlapping) to account for parameter uncertainty.

Now consider the case where the prior is misspecified. The left panel of Figure 1.4 shows the set of locally D-optimal designs (circles) and $\mathcal{D}_\eta$ selected using ISP resampling when the prior is $\mathcal{U}[0, 1]$. The results are quite similar to the design obtained earlier using the prior $\mathcal{N}(0.5,\ 0.2^2)$ except that the two selected design points are slightly more spaced-

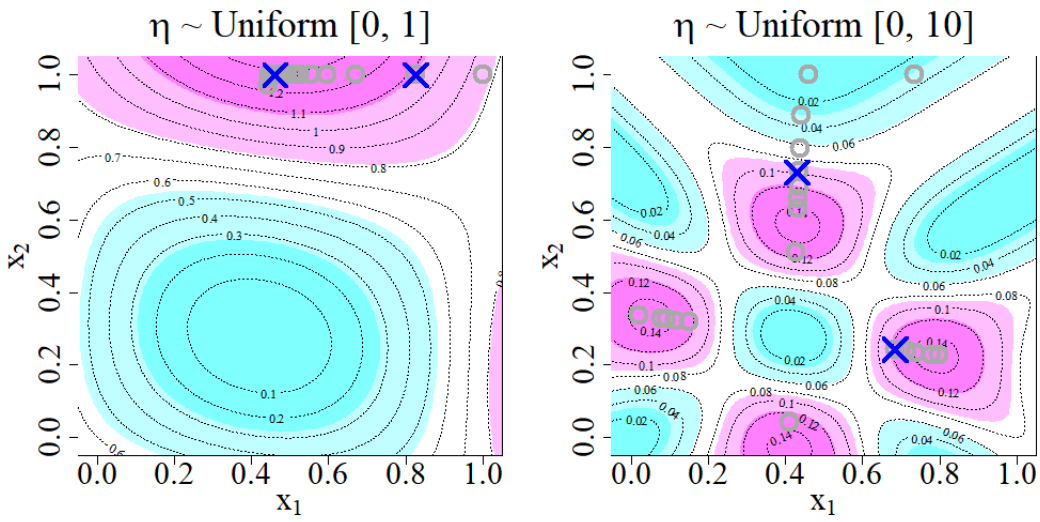

Figure 1.4: Approximate locally D-optimal design points (circles), and $\mathcal{D}_\eta$ (crosses) for (left): $\eta \sim \mathcal{U}[0, 1]$, over the computer model gradient contour for $\eta = 0.5$; (right): $\eta \sim \mathcal{U}[0, 10]$, over the computer model gradient contour for $\eta = 5$.

out. Now consider the prior $\eta \sim \mathcal{U}[0, 10]$, where the center of the distribution is far away from the previous center. The results are shown in the right panel of Figure 1.4. We can see that the $\mathcal{D}_\eta$ in Figure 1.3 is not useful anymore for efficiently estimating η. However, because of the space-filling design points, the overall design is still good and would perform much better than a pure computer model design.

1.2.3 Surrogate model uncertainties

So far we had assumed that $f(\boldsymbol{x}; \boldsymbol{\eta})$ is an easy-to-evaluate model. In reality the computer model can be very expensive to evaluate. In such cases, we will first perform a computer experiment to obtain an approximation of $f(\boldsymbol{x}; \boldsymbol{\eta})$. The approximate model is called a surrogate model or an emulator. Although there exist many different methods to obtain the surrogate model, Gaussian process modeling (or kriging) seems to be the most popular choice because of its ability to provide uncertainty estimates [30].

Let $\mathcal{S}$ denote the computer experimental design and $\boldsymbol{y}$ be the output values. Note that unlike in the physical experiment, the calibration parameters can be varied in the computer

experiment. Thus, $\mathcal{S}$ has $p+q$ columns. For simplifying the notations, let $\boldsymbol{u} = (\boldsymbol{x}, \boldsymbol{\eta})$ be the inputs in the computer experiment. Assume that $f(\cdot)$ is a realization of a Gaussian process:

$$f(\boldsymbol{u})|\boldsymbol{\eta} \sim GP(\mu, C(\boldsymbol{u}; \cdot)),$$

where μ is the mean and $C(\boldsymbol{u}; \boldsymbol{v}) = Cov\{f(\boldsymbol{u}), f(\boldsymbol{v})\}$ is the covariance function. See [2] for details on Gaussian process modeling. Given the data, the posterior distribution of $f(\boldsymbol{u})$ is also a Gaussian process given by

$$f(\boldsymbol{u})|\boldsymbol{\eta}, \boldsymbol{y} \sim GP(\hat{f}(\boldsymbol{u}), C(\boldsymbol{u}; \cdot) - C(\boldsymbol{u}; \boldsymbol{S})C^{-1}(\boldsymbol{S}; \boldsymbol{S})C(\boldsymbol{S}; \cdot)), \tag{1.7}$$

where $\hat{f}(\boldsymbol{u}) = \mu + C(\boldsymbol{u}; \boldsymbol{S})C^{-1}(\boldsymbol{S}; \boldsymbol{S})(\boldsymbol{y} - \mu\mathbf{1})$ is the surrogate model, $C(\boldsymbol{u}; \boldsymbol{S})$ is the covariance vector with ith element $C(\boldsymbol{u}; \boldsymbol{S}_i)$, $C(\boldsymbol{S}; \boldsymbol{S})$ is the covariance matrix, and $\mathbf{1}$ is a vector of 1's.

Incorporating the surrogate model uncertainties into our design construction is conceptually very simple. We generate m samples from $p(\boldsymbol{\eta})$ and for each sample, we generate a realization of the function from (Equation 2.8):

$$\begin{aligned} \boldsymbol{\eta}_i &\sim p(\boldsymbol{\eta}), \\ f_i(\boldsymbol{u})|\boldsymbol{\eta}_i, \boldsymbol{y} &\sim p(f(\boldsymbol{u})|\boldsymbol{\eta}_i, \boldsymbol{y}) \end{aligned}$$

for $i = 1, \ldots, m$. Now we can proceed to find the optimal design in the same way as in the previous section except for one difference. The gradients needed for the sensitivity matrix need to be calculated numerically for each new realization of $f(\cdot)$. This makes the procedure computationally very expensive. The procedure is summarized in Algorithm 3.

It is a good idea to make the physical experimental design a subset of the computer experiment design, that is, a nested design [31]. This avoids the confounding between the surrogate model approximation error and the discrepancy. To get a nested design, we have

Algorithm 3 : Generating the physical experimental design, accounting for model-form, parameter, and surrogate-model uncertainties.

Input $p(f(\boldsymbol{x}, \boldsymbol{\eta})|\boldsymbol{\eta}, \boldsymbol{y}), N, \gamma, p(\boldsymbol{\eta})$
$n \leftarrow \gamma N$
Find m realizations of $\boldsymbol{\eta}$ from $p(\boldsymbol{\eta})$ {R package: `support`}
for $i \in \{1, \cdots, m\}$ **do**
    $\boldsymbol{\eta}_0 \leftarrow \boldsymbol{\eta}_i$
    Find a realization of $f(\cdot)$ from $p(f(\boldsymbol{x}, \boldsymbol{\eta})|\boldsymbol{\eta} = \boldsymbol{\eta}_0, \boldsymbol{y})$
    Find ξ_i^*, as in (Equation 1.4) {R package: `ICAOD` }
    $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i/m$
end for
Find $\mathcal{D}_\eta$, as in (Equation 1.5), using ISP resampling on ξ^*
Find $\mathcal{D}_\delta$, a space-filling design with $N - n$ runs, to augment $\mathcal{D}_\eta$ {R package: `MaxPro` }
Output $\mathcal{D}_\eta \bigcup \mathcal{D}_\delta$

two options: (i) go back and run the computer simulations at the optimal physical experimental design points or (ii) choose the nearest points in $\mathcal{S}$ as the physical experimental design. If we use option (ii), there will be some loss of efficiency. Therefore, option (i) is preferred if the optimal design points are far away from $\mathcal{S}$ and it is feasible to run the computer simulation again. It is possible that the new simulations can change the surrogate model and hence the optimal design. So it seems some iterations will be needed to finalize the physical experimental design. However, we do not expect this to happen in most realistic cases unless there is too much uncertainty in the surrogate model approximation.

Consider again the toy example. For illustrative purposes, we assume that the computer model is too expensive to compute. We consider a 30-run MaxPro design in x_1, x_2, and η to develop a surrogate Gaussian process model. Instead of the computer model $f(\boldsymbol{x}; \boldsymbol{\eta})$, we use random realizations from the surrogate model $f_i(\boldsymbol{u})|\boldsymbol{\eta}_i, \boldsymbol{y}$, for $i = 1, \ldots, m$, to obtain the $m = 20$ approximate locally D-optimal designs (Figure 1.5 (left)). It can be seen that due to uncertainty in the surrogate model prediction, the design points are more dispersed than in the case of no surrogate model uncertainty (Figure 1.3 (left)).

As in the previous section, we use ISP resampling to obtain $n = 2$ runs as shown by crosses in Figure 1.5 (right). This two-run design is augmented with the Maxpro design,

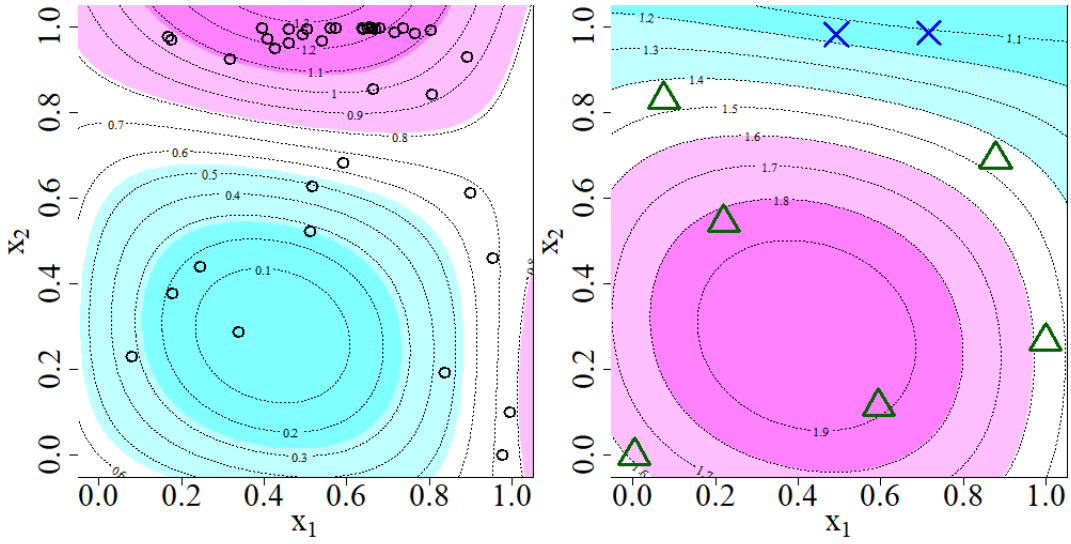

Figure 1.5: (left): Approximate locally D-optimal design points incorporating parameter and surrogate model uncertainties shown over the gradient contour of the computer model for $\eta = 0.5$; (right): The desired $N = 8$-run design over the computer model contour for $\eta = 0.5$, the crosses correspond to the D-optimal design for estimating $\boldsymbol{\eta}$, and the triangles correspond to the space-filling design.

shown by triangles in Figure 1.5 (right), to obtain the desired 8-run design. It can be seen that with the addition of surrogate model uncertainty, the two-points of the D-optimal design for estimating $\boldsymbol{\eta}$ are farther away as compared to the case in the previous section (Figure 1.3 (right)).

1.3 Simulations

In this section we will investigate the robustness of the proposed design to potential model discrepancies. Consider the toy example in (Equation 1.6) again. Let the output be

$$y = f(\boldsymbol{x}; \eta) + \delta(\boldsymbol{x}) + \epsilon, \tag{1.8}$$

where $\epsilon \overset{i.i.d}{\sim} \mathcal{N}(0, 0.05^2)$. Assume that $\delta(\boldsymbol{x}) \sim GP(0, \tau^2 R(\cdot))$, where $R(\cdot)$ is a stationary correlation function and τ^2 the variance. We will use the Gaussian correlation function given by $R(\boldsymbol{h}) = \exp(-\theta\|\boldsymbol{h}\|^2)$ with $\theta = 10$. Now we will generate the outputs by

increasing the magnitude of the model discrepancy (i.e., by increasing τ^2) and study the performance of the three designs shown in Figure Figure 1.2.

Given the design and the output, estimation of the model in Equation 1.8 is done in two steps. First, ignoring $\delta(\boldsymbol{x})$, the posterior distribution of η is found using Markov chain Monte Carlo (MCMC) simulations. Then, using the posterior samples η^i, $i = 1, \ldots, N$, the physical experiment output at a point $\boldsymbol{x}$ is estimated as:

$$\hat{f}(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{x}; \eta^i). \tag{1.9}$$

The discrepancy at each of the design points can be obtained as $\delta_i = y(\boldsymbol{x}_i) - \hat{f}(\boldsymbol{x}_i)$ for $i = 1, \ldots, n$. Let $\hat{\delta}(\boldsymbol{x})$ be the posterior mean of $\delta(\boldsymbol{x})$ given $\mathcal{D}$ and $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_n)'$. Then the bias-corrected calibrated model at a point $\boldsymbol{x}$ is given by:

$$\hat{y}(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}) + \hat{\delta}(\boldsymbol{x}). \tag{1.10}$$

The prediction accuracy of the model is evaluated on a 500-point Sobol test dataset generated using the R package `randtoolbox` [32].

Figure 1.6 plots the root mean squared prediction error (RMSPE) with $\tau^2 = 0.0, 0.01, \ldots, 0.5$. We observe that, when there is no discrepancy, the "pure computer model" design is the best, as expected, followed closely by our proposed design. However, as the model discrepancy increases, our proposed design performs better than the other two designs, and this performance gap increases with increasing discrepancy. This is because both the full factorial and the "pure computer model" designs lack the space-filling points that are critical in estimating the non linear discrepancy.

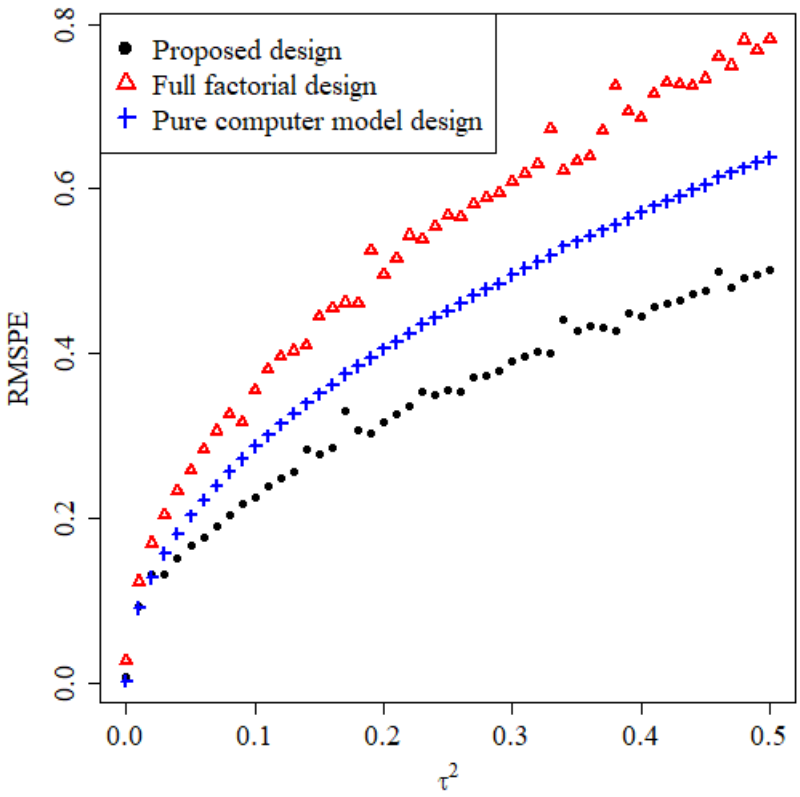

Figure 1.6: Comparing performance of our proposed design with increasing non-linear model discrepancy.

1.4 A Real Example

We apply our design strategy to a real example from P&G. The model is based on first principles, and involves a transformation of the P&G diaper line, where absorbent gelling material is being applied to a substrate. Figure 1.7 shows a schematic of the physical process. The example has been slightly modified for the benefit of simplicity and to prevent disclosure of any potential sensitive information. The anonymized physics-based model is:

$$f(\boldsymbol{x};\boldsymbol{\eta}) = AGM \times B \times 2000, \tag{1.11}$$

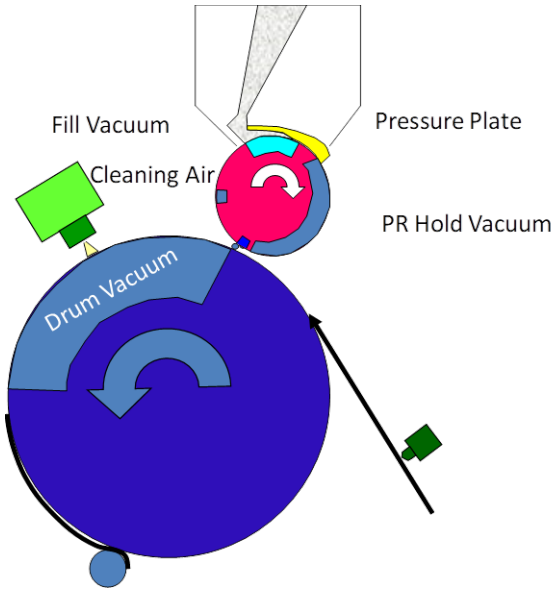

Figure 1.7: Schematic of the physical process of the real example from P&G.

where:

$$AGM = \left\{ \frac{(10^3 x_2)^{2\eta_3} {x_5}^{2\eta_3 - 1}}{\eta_1^2 (c_4)^{2\eta_3 - 1}} + 10^6 k_2 x_1 x_2 \right\} \left\{ (e^{x_4} - c_1)c_2 + c_3 \right\} x_5,$$

$$B = 1 - \left\{ k_1 - \left(\frac{10^{-3} A}{(e^{x_4} - c_1)c_2 + c_3} \right) \frac{1}{x_5} - x_3 \right\}^2 \frac{1}{\eta_2},$$

where the controllable process variables are $\boldsymbol{x} = \{x_1, ..., x_5\}$ and the unknown calibration parameters $\boldsymbol{\eta} = \{\eta_1, ..., \eta_3\}$. The details of the variables are omitted for confidentiality reasons. The budget to calibrate the physics-based model is assumed to be $N = 16$ points.

P&G has provided us with 646 physical experimental data points gathered for multiple purposes including calibration of the physics-based model. The unknown values of the three calibration parameters were estimated from this dataset as $\boldsymbol{\eta}_0 = (1.5, 200, 0.2)'$. Figure 1.8 (left) plots the physical experiment data against the predictions from the calibrated model, which shows a good agreement between the two. Therefore, we will take this calibrated model as the "true" model and use it to evaluate the proposed design with the existing designs.

Suppose we have a budget for $N = 16$ runs. No information about the confidence in the computer model is assumed. Therefore, we choose $\gamma = 2q/N = 6/N$. This leads to $n = \gamma N = 6$, which can be used for efficiently estimating the calibration parameters. The remaining 10 runs can be used for estimating the model discrepancy. Assume the following prior distributions for the calibration parameters: $\eta_1 \sim \mathcal{U}[0.1, 10], \eta_2 \sim \mathcal{U}[0.1, 1000]$, and $\eta_3 \sim \mathcal{U}[0, 0.4]$. Since the computer model is cheap-to-evaluate, there is no need to incorporate any surrogate model uncertainties. So, we use Algorithm 2 to generate the proposed design.

We will compare the proposed design with (a) "pure computer model" design and (b) Maximin augmented nested Latin hypercube design (MmANLHD) proposed by [10]. Since the computer model is cheap-to-evaluate, we can view the LHD used for approximating the computer model to have an infinite number of runs and therefore, the MmANLHD reduces to a maximin design. It is easy to see that the maximin design in 16 runs is a 2^{5-1} maximum resolution fractional factorial design. For generating the "pure computer model" design, we use Algorithm 2 with $\gamma = 1$.

For each of the experimental designs, the output is simulated as:

$$y_i = f(\boldsymbol{x}_i; \boldsymbol{\eta}_0) + \epsilon_i, \tag{1.12}$$

where $\epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 3$. We follow the same model fitting method described in Section 3 except that we do not include the model discrepancy.

The absolute prediction error is computed on each of the 646 points in the real dataset using the calibrated model for each of the designs. Figure 1.8 (right) compares the distribution of absolute prediction errors of the competing "pure computer model" design and maximin design, with our proposed design. We observe that the "pure computer model" design corresponds to the least prediction error. This is because there is no discrepancy in the computer model as seen in Figure 1.8 (left). Since all the 16 runs of the "pure computer

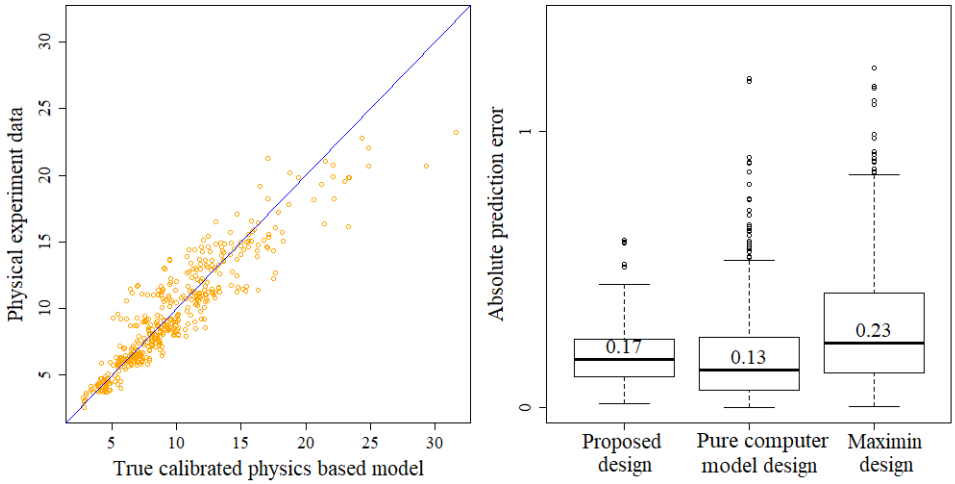

Figure 1.8: (left): Physical experiment data vs the true calibrated physics-based model from P&G; (right): Distribution of the absolute prediction error ratio in case of an unbiased physics-based model.

model" design are based on D-optimal design points, it provides the most accurate estimate of the calibration parameters, and thereby the most accurate calibration. As our proposed design has six runs of the D-optimal design points, it does better than the maximin design.

We are particularly interested to see the performance of our proposed design in the presence of model discrepancy. As the computer model $f(\boldsymbol{x};\boldsymbol{\eta})$ is unbiased, we will add a randomly generated realization of a Gaussian Process discrepancy to the true model to simulate the physical experiment output. Instead of simulating the physical experiment output using (Equation 1.12), it is simulated as:

$$y = f(\boldsymbol{x}_i;\boldsymbol{\eta}_0) + \delta(\boldsymbol{x}_i) + \epsilon_i, \tag{1.13}$$

where $\epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0,3)$, and $\delta(\boldsymbol{x})$ is a random realization from $GP(0,\tau^2 R(\cdot))$ with $\tau^2 = 6$, and $R(\boldsymbol{h}) = \exp(-\|\boldsymbol{h}\|^2)$. We fit the model in (Equation 1.1) on the simulated output to estimate $\boldsymbol{\eta}$ and the model discrepancy $\delta(\boldsymbol{x})$. The same method of estimation, as in Section 3, is used. The calibrated model is given by (Equation 1.10).

Figure 1.9 (right) plots the absolute prediction errors of the three designs. We observe

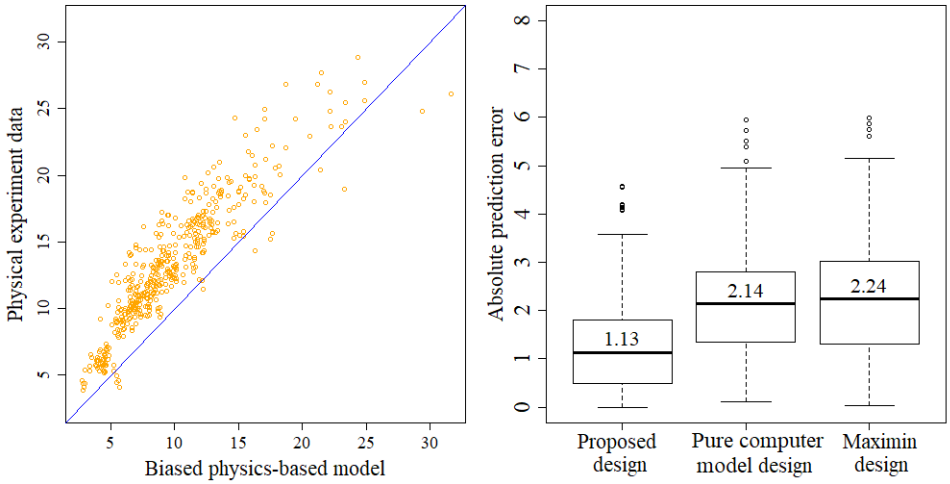

Figure 1.9: (left): Physical experiment output vs the biased physics-based model output; (right): Distribution of the absolute prediction error ratio in case of a biased physics-based model.

that our proposed design now corresponds to the least prediction error, which shows that it provides the most accurately calibrated model. A very drastic drop in the relative performance is observed in the case of the "pure computer model" design, as it does not have any points to estimate model discrepancy. On the other hand, the maximin design does not use any information from the computer model. So, it does not have optimal points to estimate the calibration parameters, which leads to a sub-optimal performance. In contrast, our proposed design uses the information from the computer model, while also accounting for the model discrepancy, which leads to a better overall performance than both the competing designs.

1.5 Conclusion

This article presented a strategy for designing physical experiments when a computer model is available to the experimenter. Optimal designs can be generated by directly using the computer model, but such designs are susceptible to possible model violations. Our design strategy augments the optimal design points with space-filling points. These space-filling points act as check points for the computer model and protect against possible model viola-

tions. The proposed designs can become inferior to the optimal designs when the computer model is perfect but will be superior when the computer model is imperfect. Since the optimal designs are a subset of our proposed design, the loss of efficiency when the computer model is perfect is minimal, and for that reason we claim that our designs are model-robust.

The proposed design strategy is simple and also flexible to be extended. Our strategy can be modified to augment points when some information of the model discrepancy is available. For example, adding points corresponding to the maximum and minimum of the computer model can be used to efficiently estimate a location-scale bias of the computer model. These can be viewed as "features" of the computer model that can act as useful model validation points. In fact, this also suggests that experimenters can extract other features from the computer model, such as all the local maxima and minima, and add them to the design, even when the connection to a discrepancy model is not evident. We hope to investigate this further in a future work.

CHAPTER 2

EXPANSION-EXPLORATION-EXPLOITATION FRAMEWORK FOR DISCOVERING NEW MATERIALS CRYSTAL STRUCTURE

2.1 Introduction

One of the most ambitious goals of material scientists is to discover and design new materials with desirable properties and applications [33, 34, 35, 36, 37, 38]. Until the present time, material discoveries are largely driven by expensive and time-consuming trial-and-error approaches, i.e., they must be physically synthesized and tested in a laboratory with limited guidance beyond empirical rules and experience. However, under some scenarios, some properties of a material can be determined without synthesizing it, if its atomic structure is known. Thus, determining the material atomic structure is a popular research problem in material science.

The specific class of materials we are concerning in this work is crystal. A crystal can be imagined as an infinitely repeated array (or lattice) of a unit cell along three Cartesian dimensions. Any arrangement of the atoms within this cell is replicated throughout the space. Our aim is to determine both the unit cell parameters and the position of the atoms within this unit cell. Crystal materials are dominant in material science because of two main reasons. First, a majority of materials are crystals and/or can be modeled very well by crystal models. Second, because of its periodicity, crystal models are small enough so that physics-based computational methods such as the Density Functional Theory (DFT) [39, 40], the most reliable (but expensive) parameter-free computational method, may be used. Thus, crystal structure prediction has been an important problem in material science since the 1950s (Desiraju 2002).

Researchers have been studying materials crystal structures since a long time. [41]

laid out a set of rules governing the structure of ionic crystals, based on crystal energy. Later, [42] introduced some geometric principles to assist in crystal structure prediction of organic materials. With the advancement of computational resources in the 21^{st} century, researchers started collecting and analyzing data, and found correlations between the structures of material and their corresponding chemistry. This led to topological approaches [43, 44, 45], and data mining approaches [46] for crystal structure prediction. However, as these approaches are based on existing data, they are prone to bias, and unlikely to lead to the discovery of novel or unexpected structures.

The least biased and non-empirical approaches to crystal structure prediction involve computational optimization [47]. These approaches involve explicit computation of the potential energy of the crystal structure, followed by solving an optimization problem to find the structure configuration corresponding to the least energy, or the most stable crystal structure. With the availability of high-speed computation, random-search based methods were developed to find the stable crystal structure configuration [48, 49]. The underlying idea in these methods is to use the DFT to compute the energies of a randomly generated set of potential crystal structure configurations. Thereafter, local optimization algorithms are used to optimize the structures to the nearest local minimum of the energy. With a large number of randomly selected samples, it is found that these approaches can successfully determine the most stable crystal structure configuration in some cases. However, these approaches have a couple of issues - inefficiency and in-applicability to large systems. Let us first consider the issue of inefficiency. The set of randomly generated configurations considered, also called the candidate set of configurations, may not be representative of all the potential configurations of the crystal structure. Thus, even with a large number of samples, we may miss to consider the most stable crystal structure configuration. The second issue is that the approach is impractical to use in case of a large number of atoms in a unit cell. The number of local minima of potential energy scale up exponentially with the number of atoms [50, 51]. This makes the random search-based methods impractical

to use for $N_A > 10$, where N_A is the number of atoms in a unit cell.

To address the second issue of the limitation of the random search-based methods to small systems ($N_A \leq 10$), material scientists focused on only low-energy regions of the potential energy surface. Methods such as simulated annealing [52, 53, 54], basin hopping [55], minima hopping [56], metadynamics [57], and evolutionary algorithms [58] were developed, which could correctly estimate the stable crystal structure configuration for some larger systems ($N_A \gtrsim 20 - 30$). However, this restricted the search for crystal structures in certain regions, which added bias to the results, and inhibited the discovery of unexpected stable configurations.

The most important requirement for a stable crystal structure configuration is that its potential energy, which depends on the position of its constituent atoms, should be minimized, ideally at the global minimum of the (multi-dimensional) potential energy surface (PES). However, external perturbations such as temperature, pressure, and other kinetic-related factors may bring a local minimum down to be the global minimum at a specific condition [59, 60, 61], or drive the atomic configuration to land at some nearby (accessible) local minima. Thus, certain local minima of the potential energy, which is a strong function of the atomic arrangement (or configuration), are also of interest in many scenarios. Therefore, another rule is that the configurations that are very far from (and/or very well-separated by a high potential energy barrier with) the global minimum are also reliable models. A classic example is diamond, which corresponds to a local minimum of elemental Carbon at ambient conditions but can be stabilized at high pressure and temperatures [62]. Because transforming it back to the global minimum (graphite) requires extremely high energy (as much as needed to completely destroy the diamond lattice into isolated atoms and then rebuild it), diamond is actually stable with infinite lifetime, and is called kinetically stable.

Predicting the stable atomic configurations of a given set of atoms can be mathematically formulated as an optimization problem, identifying the global minimum of a manifold

in a very high-dimensional space. This is a very active research area in the emerging era of materials discovery and design, when a large number of hypothetical materials should be examined by computational methods before some of them can be advanced to the synthesizing and testing steps. The two main objectives of materials structure prediction [63] are (1) searching for low-energy atomic configurations of a given set of atoms and (2) exploring new and possibly unusual domains of the PES where reliable local minima with novel properties may be found [64].

We have developed an expansion-exploration-exploitation framework that addresses the objectives of material structure prediction and avoids the issues with the current computational optimization-based state-of-the-art approaches, mentioned earlier. It is assumed that a few possible configurations of the crystal structure are known. We expand the space spanned by these configurations by perturbing them and generating more configurations in their neighborhood. The configurations are generated such that they continuously expand the spanned domain space of configurations, especially towards the low-energy regions of the domain space. Once a representative candidate set of configuration is obtained, a Bayesian optimization procedure [65] is used to explore the domain space regions with high uncertainty in the potential energy estimate while simultaneously exploiting the low-energy regions to find the global minimum and reliable local minima.

This article is organized as follows. In Section 2.2, we mention details about the crystal structure, its representation and energy computation using the DFT. In Section 2.3, we mention the constraints and challenges of the problem. In Section 2.4, we describe the developed methodology that addresses these constraints and challenges. In Section 2.5, we illustrate the effectiveness of our methodology on a real example. We conclude the chapter with some remarks in Section 2.6.

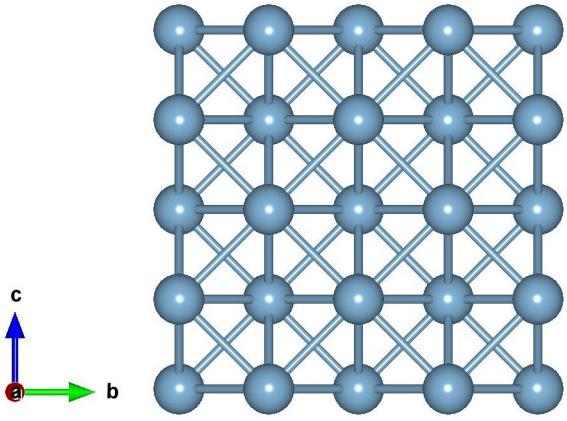

Figure 2.1: Material crystal structure of Al_8, obtained by infinitely repeating the unit cell.

2.2 The Crystal model

This section describes the crystal structure and the computation of its potential energy. In Section 2.2.1, the parameters of a crystal structure are presented. In Section 2.2.2, we explain the crystal structure representation for efficient potential energy computation using the DFT. In Section 2.2.3, we mention details regarding the DFT computations.

2.2.1 Parameters of a crystal structure

A crystal model of a material includes a parallelepiped unit cell defined by three basis vectors $\vec{a}$, $\vec{b}$, and $\vec{c}$, a given set of N_A atoms arranged in the unit cell, and an assumption that the unit cell is infinitely repeated along $\vec{a}$, $\vec{b}$, and $\vec{c}$ (see Figure 2.1 for an illustration). Because a material does not change under rigid translations and rotations, three vectors $\vec{a}$, $\vec{b}$, and $\vec{c}$ can be uniquely determined by six independent numbers. Therefore, the crystal structure prediction is mathematically equivalent to a global optimization problem on the PES defined in a $3N_A + 6$ dimensional space (N_A has no upper limit, and its typical values can be as high as 100).

2.2.2 Representing a crystal structure: AGNI fingerprint

As mentioned earlier, we use the DFT to compute the potential energy of a crystal structure. DFT computation requires a new and suitable representation of material structures, as representation in the Cartesian coordinate system leads to redundancy in energy computations. This is because the potential energy of a crystal structure depends only on the relative distance between its atoms, and not on the absolute positions of atoms. This implies that the energy is invariant to translational, rotational and permutational operations on the crystal structure configuration. Such transformations change the configuration in the Cartesian coordinate system, but do not change the material in any physical and chemical way. For this reason, we use the recently developed *AGNI* (Adaptive, Generalizable and Neighborhood Informed) fingerprint [66] which captures the atomic-level information of the structure pretty well while preserving the material presentation under such "identity" transformations in the materials space. In particular, the fingerprint used in this work is defined as $f \equiv (\{S_k; V_k\}_{k=1}^{n})$ where the scalar components S_k and the vectorial components V_k are given by

$$S_k = \sum_{i \neq j} \mathcal{G}(r_{ij}, \sigma_k) f_c(r_{ij}), \tag{2.1}$$

and

$$V_k = \sqrt{\sum_{\alpha=x,y,z} \left[\sum_{i \neq j} \frac{r_{ij}^{\alpha}}{r_{ij}} \mathcal{G}(r_{ij}, \sigma_k) f_c(r_{ij}) \right]^2}, \tag{2.2}$$

respectively. Here, r_{ij} is the distance between atoms i and j, r_{ij}^{α} is the projection of r_{ij} onto the Cartesian axis α, $\mathcal{G}(r, \sigma_k)$ is the Gaussian function centered at 0 with varying width σ_k:

$$\mathcal{G}(r, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(\frac{-r^2}{2\sigma_k^2}\right), \tag{2.3}$$

and $f_c(.)$ is a cutoff function that is used to disregard interaction among atoms that are further than a distance R_c from each other:

$$f_c(r) \equiv \frac{1}{2}\left[\cos(\pi r/R_c) + 1\right]. \tag{2.4}$$

While summarizing over the atoms i and j, the periodicity of the unit cell is considered, i.e., for an atom, its neighbors in all the repeated images of the unit cell are also taken into account. Thus, the cutoff function, $f_c(r)$, defined in (Equation 2.4) is used to restrict the neighborhood to a radius of R_c. We used $R_c = 8$ Å in this work, because the interaction between two atoms at this distance is negligible. From the mathematical point of view, *AGNI* fingerprint is a way of projecting the atomic positions onto a set of predefined basis functions. Here, we used $n = 16$ functions $\mathcal{G}(r, \sigma_k)$, thus our fingerprint f has $2n = 32$ components or dimensions. Note that the accuracy of the model using a fingerprint increases as the dimension increases and then saturates. Our tests indicate that after 32 dimensions, the increase in model accuracy with increasing dimensions is negligible. The *AGNI* fingerprint is one of the numerous material fingerprints [67, 68] developed during the last decade. In the new language, each atomic structure is now represented by a numerical of given dimensionality, and such vectors are then used for machine-learning algorithms.

2.2.3 Computing the potential energy of a crystal structure: DFT

The adaptive expansion step in our approach is guided by single-point DFT computations to determine the potential energy of the atomic configurations examined. Thus, the aim of our density functional theory (DFT) computations is to determine the potential energy of any atomic structure examined without any local optimizations. Such (single-point) calculations are performed using the ABINIT package [69], employing the Perdew-Burke-Ernzerhof functional [70] for the quantum mechanical exchange-correlation energies. The electron-nuclear interactions are computed with help from the norm-conserving

Hartwigsen-Goedecker-Hutter pseudopotentials [71]. For our calculations, the Brilouin zone is sampled by a dense Monkhorst-Pack k-point mesh [72], and a basis set of plane waves with kinetic energy up to 550 eV.

2.3 The Problem: Constraints and Challenges

Crystal structure prediction has several constraints that lead to the corresponding challenges. To be effective, the solution must address these challenges presented in the following subsections.

2.3.1 Crystal structure representation: One-sided mapping

As mentioned earlier, we use the $AGNI$ fingerprint to represent the crystal structure configuration. There is one-to-one mapping between an atomic configuration and its energy in the $AGNI$ system, which is essential to find a unique crystal structure configuration that corresponds to the minimum potential energy.

Although the $AGNI$ system eliminates redundancy in the Cartesian coordinate representation of the crystal structure configuration, and reduces the PES dimensionality from $3N_A+6$ to 32 (as described in Section 2.2), it introduces a constraint in solving the problem. It is not possible to obtain the Cartesian coordinate system configuration for an $AGNI$ fingerprint, as one fingerprint corresponds to several redundant Cartesian coordinate system configurations. We can only map a configuration in the Cartesian coordinate system to the $AGNI$ system, but not vice-versa.

The constraint mentioned above leads to a challenge in solving the optimization problem of finding the crystal structure configuration with the least potential energy. In the absence of this constraint, we may have used a continuous optimization procedure in the *AGNI* fingerprint space to find a solution and transform it to the physically interpretable Cartesian coordinate system. However, in the presence of this constraint, we can only use discrete optimization approaches to find the configuration with the least energy. Thus,

we will need to consider a finite set of configurations (also called a candidate set of configurations), find their corresponding fingerprints, and find the one with the least energy. The discrete optimization approach gives rise to a huge challenge that the candidate set of fingerprints must contain the solution(s) or fingerprints "close enough" to the solution(s).

2.3.2 Crystal structure representation: Unknown domain space

We are aware of the range of the coordinates of the crystal structure configuration in the Cartesian coordinate system. However, we do not know the range of the configuration coordinates of the corresponding $AGNI$ fingerprint. This implies that we do not know the domain space of the fingerprints. This gives rise to the challenge of obtaining a candidate set of fingerprints that are representative of all fingerprints, or a candidate set of crystal structure configurations that are representative of all possible configurations. In other words, it is challenging to know if a configuration, which is very different from the rest, is missing from our candidate set. If we knew the domain space, we could have used a space-filling design [24] to obtain a representative candidate set of fingerprints. However, the challenge is to find a representative set of fingerprints without the knowledge of the domain space of the fingerprints. In other words, we need to find a space-filling design without knowing the boundaries of the space to be filled!

2.3.3 Expensive energy computation: Density functional theory (DFT)

To find the most stable crystal structure configuration, we need to find the one with the least potential energy. As mentioned earlier, we use the quantum mechanical modelling method known as the density functional theory (DFT), to compute the potential energy of a given crystal structure configuration. However, each evaluation of the potential energy using DFT requires several hours or even days. This is why predicting a simple crystal structure by computations was regarded as "*one of the continuing scandals in the physical sciences*" in 1988 by a Nature's editor, Sir John Maddox [73]. Although structure prediction methods

have evolved dramatically since then and have led to numerous new materials predicted computationally and realized experimentally [63], this remains a major bottleneck of contemporary materials discoveries. The expensive DFT computations constrain us to evaluate the energy for only a few configurations, which gives rise to the challenge of optimizing a huge potential energy surface, while observing it at only a few points.

2.3.4 Multi-modal potential energy surface (PES)

The potential energy surface is highly nonlinear and multi-modal. Given that we can observe it at only a few points (see constraint 2.3.3), it becomes challenging to accurately model all the modalities. As the number of local minima scale up exponentially with the number of atoms in a unit cell, N_A, the challenge is even bigger for crystal structures with large N_A. However, modeling the multi-modalities is necessary to find the global minimum as well as other reliable local minima.

2.4 Methodology

We have developed an expansion-exploration-exploitation framework for crystal structure prediction that addresses all the challenges presented in Section 2.3. This framework is implemented in two steps. The first step is domain space expansion, where we expand the space spanned by a small set of known configurations by iteratively adding more configurations. This leads to a candidate set of configurations that will ideally either span the entire domain space of possible configurations or at least span the space of stable configurations. The expansion step consists of a sequence of two sub-steps: non-adaptive expansion and adaptive expansion. Non-adaptive expansion refers to adding configurations without considering their potential energy. This tends to include unexpected configurations in our candidate set. If needed, this step is followed by adaptive expansion, which tends to add configurations that further expand the low-energy regions of the domain space. The expansion step is followed by simultaneous exploration and exploitation of the domain space

spanned by the candidate set to find the configuration that corresponds to the minimum potential energy. We will explain these steps in the three sections below.

2.4.1 Non-adaptive domain space expansion

The purpose of this step is to obtain a candidate set of configurations that span as much domain space as possible. We start from a set of few known configurations, and iteratively add those configurations to the set that expand their spanned domain space. The potential energy of the configurations is ignored, while developing the candidate set, to serve two purposes. First, it may lead us to regions of the domain space where a low-energy configuration is unexpected. Second, it saves the computational resources for calculating the energy and helps us obtain a large candidate set within a given time period.

Let us explain the algorithm with a toy example. Although the $AGNI$ fingerprint space is 32-dimensional, we consider a two-dimensional toy example to visualize the algorithm. Let the space of all possible fingerprints, or the fingerprint domain space, be $[-3, 3] \times [-3, 3]$. However, in practice, we are not aware of the fingerprint domain space. So, we will not feed this space information to our algorithm. Nevertheless, the objective of our algorithm will be to find a candidate set of fingerprints that represent this domain space.

The algorithm begins by considering a set of known crystal structure configurations, and their fingerprints as the initial candidate set. Let the initial candidate set of atomic configurations be $\mathcal{C} = \{\boldsymbol{c}_1, \cdots, \boldsymbol{c}_I\}$, where $\boldsymbol{c}_1, \cdots, \boldsymbol{c}_I$ are the I initial atomic configurations. Let the fingerprints corresponding to these atomic configurations be $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_I\}$, where $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_I$ are the I initial fingerprints. We will iteratively add fingerprints to this candidate set that make it more representative of the domain space of fingerprints. Let us assume that there are a set of $I = 5$ known fingerprints for our toy example, as shown in Figure 2.2 (left), and we have a budget of expanding it to N fingerprints. To find a representative set of fingerprints, we must find fingerprints in the least represented regions of the domain space. To identify the least represented region, we will find the space spanned

by each fingerprint in the current candidate set.

Consider the two-dimensional fingerprints in Figure 2.2 (left). Intuitively, the space spanned by a fingerprint depends on its proximity to its neighboring fingerprints. If the neighbors on the left and right of the fingerprint, and below it are close (such as those for $\boldsymbol{x}_2$), then it spans a small space in those neighborhoods. Such neighborhoods are already well-represented in the candidate set, and we do not need to generate more fingerprints in them. However, suppose the neighbor above the fingerprint is far away. Then, the space spanned by the fingerprint in the neighborhood above it is large. This implies that there is unexplored domain space in the neighborhood above the fingerprint, and we should generate fingerprints in that space to increase its representation. Thus, we need to generate new fingerprints around the one that has the largest unexplored neighborhood, or the largest span in any of its neighborhoods. For a p-dimensional fingerprint, there are $2p$ neighborhoods - two on either side of it along each dimension. We intend to identify the fingerprint that has the largest span in any of its $2p$ neighborhoods.

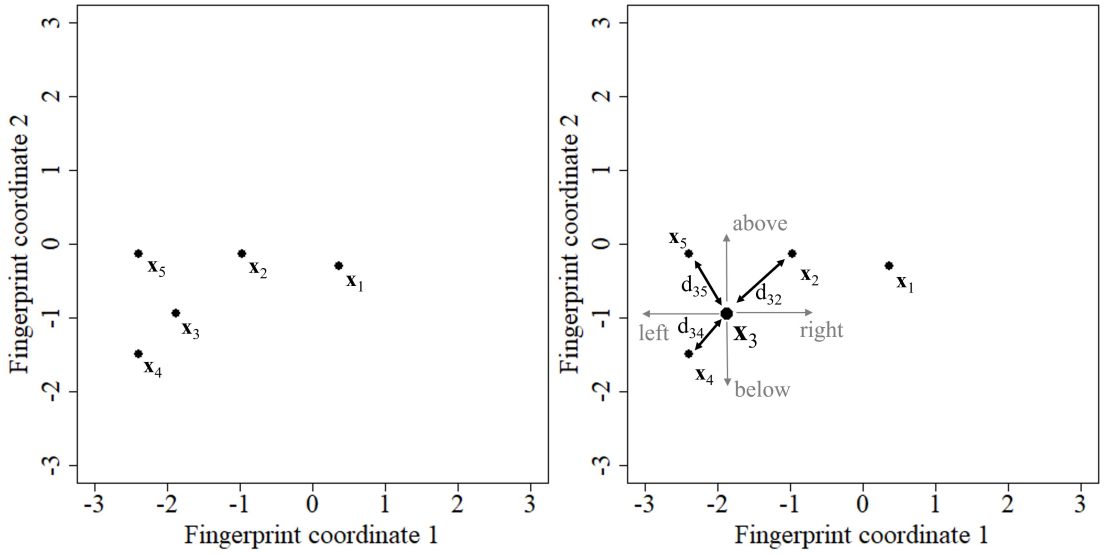

Figure 2.2: (left): Initial $I = 5$ fingerprints; (right): Distances to the closest neighbors of fingerprint $\boldsymbol{x}_3$ in all of its $2p = 4$ neighborhoods.

Let us find the space spanned by the fingerprint $\boldsymbol{x}_3$ in Figure 2.2 (right). For that we will

find the space spanned in $2p = 2 \times 2 = 4$ neighborhoods - above and below $\boldsymbol{x}_3$, and right and left of $\boldsymbol{x}_3$. The nearest neighbors above and below are $\boldsymbol{x}_3$ are at a distance of $d_{35} = 0.96$ and $d_{34} = 0.75$ respectively, while those on the left and right are at distance of $d_{34} = 0.75$ and $d_{32} = 1.22$ respectively. Since we are interested in the fingerprint with the largest unexplored neighborhood, we compute the distance to the farthest neighbor among all the four neighborhoods. The farthest neighbor of $\boldsymbol{x}_3$ is at a distance of $\max(d_{32}, d_{34}, d_{35}) = 1.22$. We define the space spanned by a fingerprint as a circle with radius equal to half of its distance to the farthest neighbor. Let $\boldsymbol{R} = \{r_1, \cdots, r_I\}$ be the vector of radii of the circle corresponding to the space spanned by each fingerprint. Thus, the space spanned by $\boldsymbol{x}_3$ is a circle of radius $r_3 = 0.5 \times 1.22 = 0.61$.

Figure 2.3 (left) shows the space spanned by each fingerprint. Clearly, the fingerprint $\boldsymbol{x}_1$ is spanning the largest domain space. In other words, the domain space around $\boldsymbol{x}_1$ consists of the most unexplored or sparse neighborhood. Let us label the fingerprint $\boldsymbol{x}_1$ as $\boldsymbol{x}_{sparse}$. So, we will find a fingerprint in the domain space around $\boldsymbol{x}_{sparse}$, and add it to the candidate set to increase its representation. Note that an arbitrary shape might have been more accurate to define the space spanned by each fingerprint, instead of a circle. This is because the space spanned by a fingerprint differs by neighborhood. However, we have considered a circle since it is defined by a single parameter, which makes the computation much cheaper. Furthermore, we only need to identify the fingerprint with the most unexplored neighborhood, and not the exact area/volume of that neighborhood.

Once the fingerprint with the most unexplored neighborhood is identified, we intend to find a new fingerprint around it, and add it to the candidate set. To do so, we randomly perturb the atomic configuration corresponding to the identified fingerprint to generate another configuration. Then, we fingerprint this randomly generated atomic configuration to obtain the new fingerprint. This new fingerprint is added to the candidate set if it is "far enough" from its nearest neighboring fingerprint. We use a threshold distance t, which is updated in each iteration, and will be defined later, to check if the randomly generated fingerprint is

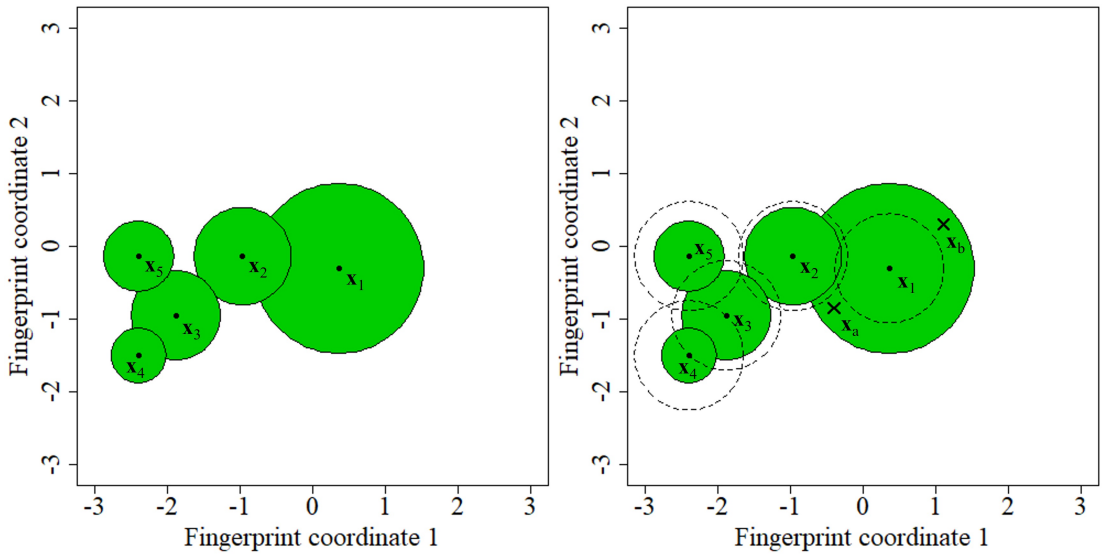

Figure 2.3: (left): Space spanned by the initial $I = 5$ fingerprints; (right): Dotted circle around each fingerprint with radius t, showing the minimum distance necessary between them and an acceptable newly generated fingerprint; Two examples of acceptable new fingerprints for inclusion in the candidate set - $\boldsymbol{x}_a, \boldsymbol{x}_b$.

"far enough". If it is not "far enough", then we discard it, update the threshold distance t, and again perturb the same atomic configuration.

The purpose of the threshold distance t is twofold. First, it is used to avoid redundancy of fingerprints in the candidate set. Second, it ensures that the fingerprints are evenly spaced-out in their domain space. The threshold distance is updated in each iteration, irrespective of acceptance or rejection of the newly generated fingerprint. If t_i is the threshold distance in the ith iteration, and $d_{min,i}$ is the distance of the new fingerprint to its nearest neighbor in this iteration, then the threshold distance for the next iteration is given by:

$$t_{i+1} = 0.5(t_i + d_{min,i}) \; \forall i > 1. \tag{2.5}$$

The term $d_{min,i}$ ensures that the threshold distance is large when large parts of the domain space are unexplored, and small if the domain space is already well-explored. This makes the fingerprints spread farther apart until the entire domain space has been explored. Once the domain space has been explored, the threshold distance decreases so that new finger-

prints may be added to the candidate set, until the budget of N fingerprints is exhausted. The term t_i ensures that the threshold distance does not change abruptly for an abrupt change in $d_{min,i}$. For the first iteration, $i = 1$, t_1 is taken as the mean of the distances to the nearest neighboring fingerprint for each fingerprint.

In our toy example, $\boldsymbol{x}_1$ is the fingerprint identified with the most unexplored neighborhood. So, we will perturb the atomic configuration corresponding to it to generate another one, and fingerprint it. Figure 2.3 (right) shows the five fingerprints in the candidate set with a dotted circle around them whose radius is equal to the threshold distance $t_1 = 0.75$. If the new fingerprint falls inside any of the dotted circles, then it will be rejected on account of being redundant with the fingerprints in the candidate set. Figure 2.3 (right) shows two examples of an acceptable new fingerprint - $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$. The two distinct examples throw light on two different aspects of our adaptive space exploration algorithm. If $\boldsymbol{x}_a$ is the new fingerprint, then it contributes in filling the domain space between $\boldsymbol{x}_1$ and the rest of the fingerprints in the candidate set. In other words, it fills gaps in the domain space of the candidate set. If $\boldsymbol{x}_b$ is the new fingerprint, then it contributes to expanding the convex hull of the domain space. In other words, it expands the boundaries of the domain space. Thus, the algorithm populates relatively sparse regions of the domain space, while also expanding its boundaries. This dual behavior is key in obtaining a set of atomic configurations that are representative of all possible atomic configurations of the crystal structure.

We repeat the exercise of identifying and adding a fingerprint around the most unexplored neighborhoods, until we have added the desired number of fingerprints in the candidate set. Figure 2.4 shows the results obtained when we have a candidate set of $N = 200$ fingerprints, and $N = 400$ fingerprints. Our algorithm performs well in (a) providing a candidate set of fingerprints that are representative of the domain space, (b) spacing-out fingerprints such that they evenly span the domain space within a given budget of N fingerprints. Thus, our algorithm addresses the two challenges - (a) challenge 3.1 of exploring all regions of the domain space, so that the region corresponding to the stable fingerprint(s)

will be represented in the candidate set, (b) challenge 3.2 of obtaining a candidate set that fills the domain space without the knowledge of the boundaries of the domain space. Once the algorithm hits a boundary of the domain space, it moves to unexplored regions away from the boundary. In essence, our algorithm "learns" the boundaries of the domain space, and leverages the threshold distance t to provide the desired space-filling design. The non-adaptive domain space expansion algorithm is summarized as algorithm 4.

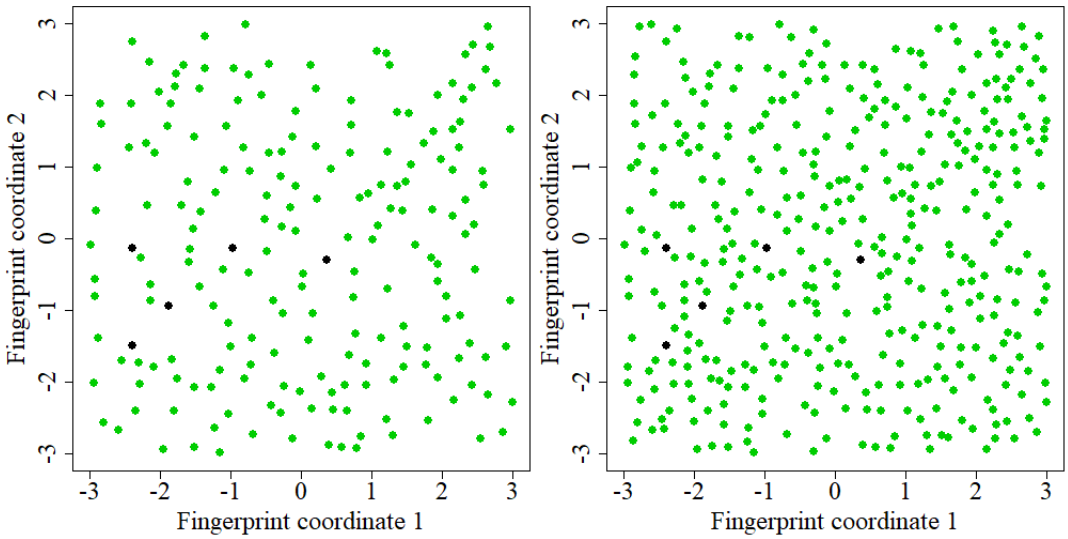

Figure 2.4: Candidate set of (left): $N = 200$ fingerprints; (right): $N = 400$ fingerprints, where the unknown fingerprint domain space is assumed to be $[-3, 3] \times [-3, 3]$.

Let us discuss the choice of N. Ideally, the minimum value of N should be the number of fingerprints needed to span over the domain space of all fingerprints. This means that we must keep on adding fingerprints in the candidate set at least until no more domain space remains to be explored. Simultaneous occurring of two events will indicate the exploration of the entire domain space. First, the convex hull of the candidate set of fingerprints will stop expanding, and become constant. Second, the threshold distance t will start decreasing. When these two events occur, it means that the algorithm is further exploring the already explored regions of the domain space. It is reasonable to stop adding fingerprints to the candidate set at this moment. However, in practice, the domain space of fingerprints is too large, and the convex hull is likely to continue expanding. In such cases,

Algorithm 4 : Non-adaptive domain space expansion

Input $N_{max}, \mathcal{C}, \mathcal{X}$
$i \leftarrow I$
Compute $\boldsymbol{R} = \{r_1, \cdots, r_I\}$ {Space spanned by each fingerprint}
Compute $\boldsymbol{D} = \{d_1, \cdots, d_I\}$ {Distance to the nearest neighbor of each fingerprint}
$t \leftarrow$ mean($\boldsymbol{D}$)
while $i \leq N_{max}$ **do**
    per $\leftarrow \arg\max_i r_i$
    Randomly perturb $\boldsymbol{c}_{per}$ to generate $\boldsymbol{c}_{new}$
    $\boldsymbol{x}_{new} \leftarrow$ fingerprint($\boldsymbol{c}_{new}$)
    Compute d_{min} {Distance to the nearest neighbor of $\boldsymbol{x}_{new}$}
    if $d_{min} > t$ **then**
        $\mathcal{C} \leftarrow$ append($\mathcal{C}, \boldsymbol{c}_{new}$)
        $\mathcal{X} \leftarrow$ append($\mathcal{X}, \boldsymbol{x}_{new}$)
        $i \leftarrow i + 1$
        Update $\boldsymbol{R}, \boldsymbol{D}$
    end if
    $t \leftarrow 0.5(t + d_{min})$
end while
Output $\mathcal{C}, \mathcal{X}$

the number of fingerprints will need to be constrained by the maximum permissible budget $N = N_{max}$. Suppose the unknown domain space of fingerprints is $[-20, 20] \times [-20, 20]$. Then, the candidate set of fingerprints obtained for $N_{max} = 200$ and $N_{max} = 400$ are as shown in Figure 2.5. In these cases, the convex hull of the candidate set continues to expand and so we stop after the budget of N_{max} fingerprints is exhausted. As a thumb-rule, for a p-dimensional fingerprint, we suggest a candidate set of size $N_{max} = 100p$ in the non-adaptive expansion procedure.

As the non-adaptive expansion algorithm does not consider the potential energy of the configurations, it addresses the second objective of crystal structure prediction mentioned earlier, i.e., exploring new and possibly unusual domains of the PES.

2.4.2 Adaptive domain space expansion

Adaptive domain space expansion is an optional step and may not be needed if we obtain a "good enough" candidate set with the non-adaptive expansion algorithm. A candidate set

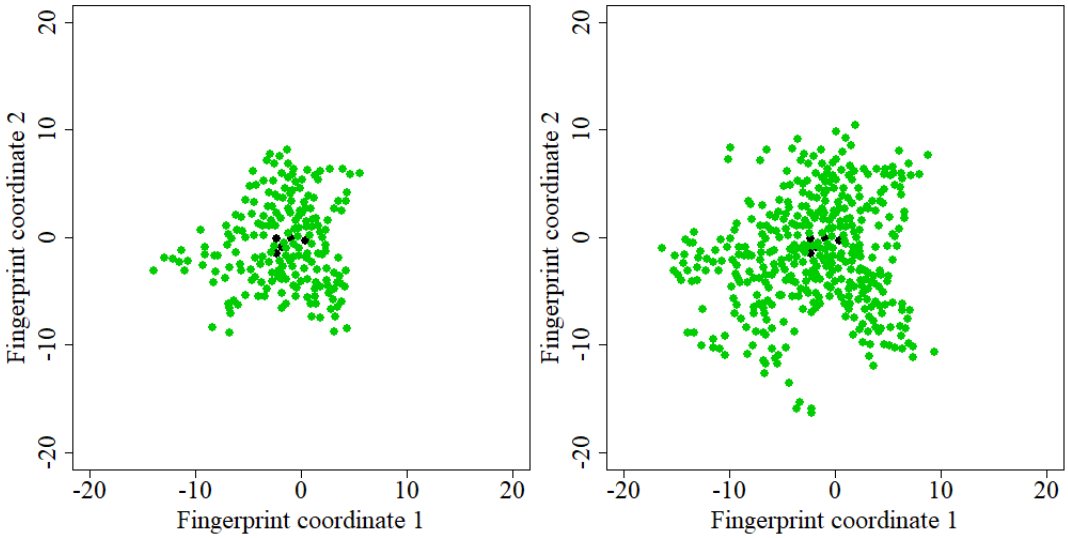

Figure 2.5: Candidate set of (left): $N_{max} = 200$ fingerprints; (right): $N_{max} = 400$ fingerprints, where the unknown fingerprint domain space is assumed to be $[-20, 20] \times [-20, 20]$.

is "good enough" if it spans over the entire domain space of crystal structure configurations. Spanning the entire domain space is likely if the convex hull of the candidate set of fingerprints stops expanding during the non-adaptive expansion procedure. Let us consider the two dimensional toy example of Section 4.1. In Figure 2.4 the convex hull of $N = 200$ fingerprints is approximately $[-3, 3] \times [-3, 3]$, which is the same as the convex hull of $N = 400$ fingerprints. Thus, the convex hull does not expand after we have $N = 200$ fingerprints in the candidate set. This leads to the conclusion that $N \geq 200$ fingerprints represent the entire domain space of crystal structure configurations and further expansion is not needed.

In practical scenario, we do not expect the candidate set to span the entire domain space of configurations due to the typically large size of the PES. In Figure 2.5 we observe that the convex hull of the candidate set of fingerprints continues to expand throughout the non-adaptive expansion procedure. Thus, we conclude that even $N = 400$ fingerprints do not span the entire domain space, or there is unexplored domain space that may contain the global minimum. We need to identify, and if necessary, then further expand the low-energy regions of the domain space. To illustrate the need for adaptive expansion, we will consider

this variation of the example in the rest of this section, i.e., $N_{max} = 400$ and the unknown fingerprint domain space assumed to be $[-20, 20] \times [-20, 20]$.

As the budget of non-adaptive expansion ($N_{max} = 400$) is exhausted, we will focus on the "promising regions" or the low-energy regions of the domain space, instead of the sparsely populated regions. For identifying low-energy regions, we use DFT to compute the energy of, say, $10p = 20$ fingerprints, from the candidate set of $N = 400$ fingerprints. Then, a Gaussian process model based on this known-data is used to estimate the energy of all the configurations, and identify the low-energy "promising regions".

To develop the best possible model, we need to carefully choose the appropriate 20 configurations for which to compute the potential energy. First, all the five fingerprints corresponding to the initially known configurations are chosen, as those are likely to be stable. The remaining 15 fingerprints are chosen by augmenting these five fingerprints with the maximum projection (MaxPro) design [25]. The MaxPro design is a space-filling design that simultaneously ensures space-fillingness in the full dimensional space as well as in all the lower dimensional subspaces. The augmented design can be obtained sequentially by adding one fingerprint at a time using the `MaxProAugment` function in the R package `MaxPro` [26]. Note that the function selects 15 space-filling fingerprints from the candidate set of $N_{max} = 400$ fingerprints. Figure 2.6 shows the MaxPro design obtained for our toy example. The five crosses in the center are the initial fingerprints that are augmented sequentially by 15 space-filling fingerprints from the candidate set of $N_{max} = 400$ fingerprints.

Let us assume that the potential energy obtained using DFT in our toy example is given by:

$$e(\boldsymbol{x}) = \sqrt{\left(\frac{x_1}{15}\right)^2 + \left(\frac{x_2}{15}\right)^2} \tag{2.6}$$

We obtain 20 fingerprints, $\boldsymbol{x} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{20}\}$ using the MaxPro design, and their corresponding potential energy $\boldsymbol{e} = \{e_1, \cdots, e_{20}\}$ using (Equation 2.6).

A Gaussian Process model is developed on the known-data, and is used as a cheap

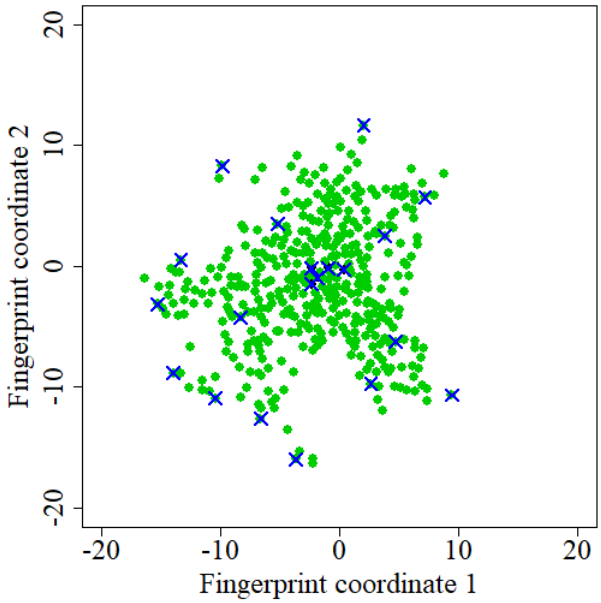

Figure 2.6: The space-filling MaxPro design, shown as crosses, for the toy example with $N_{max} = 400$ and the unknown fingerprint domain space as $[-20, 20] \times [-20, 20]$.

surrogate to estimate the potential energy of all the configurations in the candidate set. The flexible and smooth Gaussian process model is a good fit for the highly multi-modal and non-linear potential energy surface. We assume that $e(\cdot)$ is a realization of a Gaussian process:

$$e(\boldsymbol{x}) \sim GP(\mu, C(\boldsymbol{x}; \cdot)), \tag{2.7}$$

where μ is the mean and $C(\boldsymbol{x}_u; \boldsymbol{x}_v) = Cov\{f(\boldsymbol{x}_u), f(\boldsymbol{x}_v)\}$ is the covariance function. See [2] for details on Gaussian process modeling. Given the known-data, the posterior distribution of $e(\boldsymbol{x})$ is also a Gaussian process given by

$$e(\boldsymbol{x})|\boldsymbol{e} \sim \mathcal{N}(\hat{e}(\boldsymbol{x}), s(\boldsymbol{x})), \tag{2.8}$$

where $\hat{e}(\boldsymbol{x}) = \mu + C(\boldsymbol{x}; \boldsymbol{S})C^{-1}(\boldsymbol{S}; \boldsymbol{S})(\boldsymbol{e} - \mu\mathbf{1})$ is the surrogate model, $s(\boldsymbol{x}) = C(\boldsymbol{x}; \cdot) - C(\boldsymbol{x}; \boldsymbol{S})C^{-1}(\boldsymbol{S}; \boldsymbol{S})C(\boldsymbol{S}; \cdot)$ is the standard error, $C(\boldsymbol{x}; \boldsymbol{S})$ is the covariance vector with ith element $C(\boldsymbol{x}; \boldsymbol{S}_i)$, $C(\boldsymbol{S}; \boldsymbol{S})$ is the covariance matrix, and $\mathbf{1}$ is a vector of 1's. We use the R package `DiceKriging` [74] to develop the Gaussian process model. Figure 2.7 (left)

shows the potential energy contour based on the developed Gaussian process model. We see that the estimated global minimum seems to lie in the "interior" of the candidate set of configurations. In this case, there is no need to further expand the "low-energy" region, as it is already surrounded by the candidate set of configurations.

Now, let us consider another scenario, where the potential energy obtained using DFT is given by:

$$e(\boldsymbol{x}) = \sqrt{\left(\frac{x_1 - 10}{15}\right)^2 + \left(\frac{x_2 - 10}{15}\right)^2} \tag{2.9}$$

Figure 2.7 (right) shows the potential energy contour based on the developed Gaussian process model. The estimated global minimum seems to lie at the "boundary" of the candidate set of configurations. In this case the "low-energy" region is not well explored on all sides. Thus, in this case, we need to further expand the "low-energy" region to ensure that the minimum of the "low-energy" region, which may potentially be the global minimum or a reliable local minimum, is included in the candidate set of configurations.

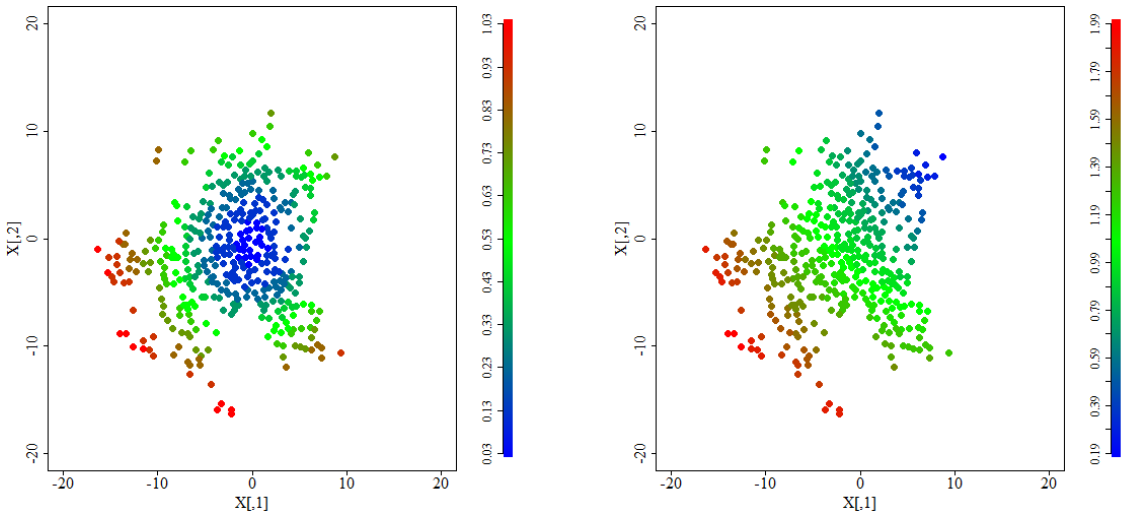

Figure 2.7: Estimated potential energy contour for the toy example with $N_{max} = 400$, the unknown fingerprint domain space as $[-20, 20] \times [-20, 20]$, and the DFT function as given in (left): equation (Equation 2.6); (right): equation (Equation 2.9).

The definition of the "boundary" and the "interior" of the candidate set of configurations is based on the dimension p of the fingerprint. A two-dimensional fingerprint, assumed to

be at the origin, lies in the interior of the domain space if it has neighboring fingerprints in each of the four quadrants, within a distance r around it. Otherwise, it lies on the boundary of the domain space spanned by the candidate set of fingerprints. Here r is taken to be the maximum distance to the nearest neighbor for the candidate set of N fingerprints obtained with the non-adaptive expansion algorithm. Similarly, a three-dimensional fingerprint should have neighboring fingerprints in each of the eight octants (instead of quadrants for the two-dimensional case) within a radius r around it, to lie in interior of the spanned domain space, and so on. Figure 2.8 shows examples of two-dimensional fingerprints that lie on the boundary or in the interior of the domain space spanned by the candidate set.

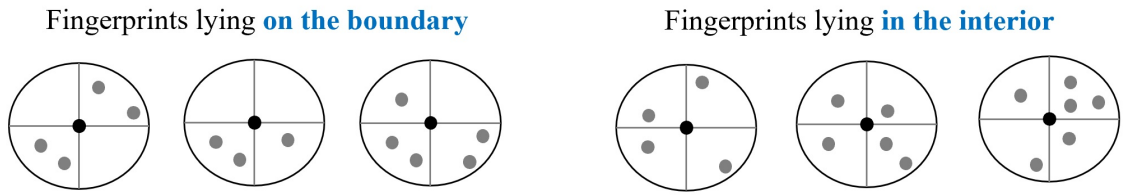

Figure 2.8: Examples of two-dimensional fingerprints that lie (left): on the boundary of the domain space spanned by the candidate set; (right): in the interior of the domain space spanned by the candidate set. Note that the radius of the circle is r.

Based on the different variations of the toy example presented in Figure 2.4 and Figure 2.7, we conclude that adaptive expansion is needed when two conditions are satisfied. First, the convex hull of the candidate set of fingerprints continues to expand in the non-adaptive expansion procedure. In other words, the domain space of all possible configurations has not not been fully explored. Second, the estimated minimum of the candidate set lies at the boundary of the spanned domain space. In other words, the low-energy region(s) of the domain space have not been fully explored.

For the case shown in Figure 2.7 (right), there is a need to further expand the low-energy region, as the minimum lies at the boundary of the spanned domain space. New fingerprints are generated as in the non-adaptive expansion algorithm, i.e., by perturbing a fingerprint already in the candidate set. However, the choice of the fingerprint to perturb

is focused on expanding the low-energy region. As we need to push the boundary of the low-energy region, we perturb the lowest energy fingerprint that lies on the boundary of the spanned domain space. As we continue to expand the spanned domain space, we also need to add data points to the known-data to update our model. With the addition of every 10 fingerprints to the candidate set, DFT is used to compute the potential energy of the fingerprint with the least energy estimate. The Gaussian process model is then updated to better estimate the energy in the newly explored lower-energy domain space. A periodic model-update helps navigate the expansion of the lower-energy region.

A criterion is required to stop the adaptive expansion once the low-energy regions have been well explored. If the fingerprint having the minimum estimated potential energy does not change within 10 successive DFT computations, it means the low-energy minimum is surrounded by the candidate set of fingerprints, and we stop the algorithm. On the other hand, if the fingerprint with the minimum estimated potential energy continues to change, it means the algorithm is expanding the spanned domain space to lower-energy regions, and we must continue to generate new fingerprints adaptively.

The adaptive domain space exploration algorithm is summarized as algorithm 5.

Figure 2.9 shows the result of applying the adaptive expansion algorithm to the scenario presented in Figure 2.7 (right). The algorithm adaptively adds 110 fingerprints to the set of 400 fingerprints obtained through non-adaptive expansion. Note that the algorithm continues to expand until the low-energy region is fully explored, and the minimum is well surrounded by the candidate set of fingerprints.

As the adaptive expansion algorithm is focused on further expanding the low-energy regions of the domain space, it addresses the first objective of crystal structure prediction mentioned earlier, i.e., searching for low-energy atomic configurations of a given set of atoms.

Algorithm 5 : Adaptive domain space expansion

Import $\mathcal{C}, \mathcal{X}, \boldsymbol{R}, \boldsymbol{D}, t$ {Obtained at the end of the non-adaptive expansion procedure}
$\mathcal{X}_I \leftarrow \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_I\}$
Augment $\mathcal{X}_I$ by $10p - I$ space-filling fingerprints to obtain $\mathcal{X}_{DFT}$ {R package:`MaxPro`}
$\boldsymbol{e} \leftarrow DFT(\mathcal{X}_{DFT})$
model $\leftarrow GP(\mathcal{X}_{DFT}, \boldsymbol{e})$
$\hat{\boldsymbol{e}} \leftarrow GP.predict(\mathcal{X})$
flag $\leftarrow 1$; DFT_period $\leftarrow 0$; iter $\leftarrow 0$;
while flag $= 1$ **do**
  Find $\boldsymbol{c}_{per}$, the configuration with minimum estimated energy lying on the boundary
  Lines $8 - 10$ from the non-adaptive domain space expansion algorithm
  if $d_{min} > t$ **then**
    Lines $12 - 15$ from the non-adaptive domain space expansion algorithm
    DFT_period $\leftarrow$ DFT_period+1
    $e_{new} \leftarrow GP.predict(\mathcal{X}_{new})$
    $\hat{\boldsymbol{e}} \leftarrow$ append($\hat{\boldsymbol{e}}, e_{new}$)
    if DFT_period $= 10$ **then**
      iter $\leftarrow$ iter+1
      if iter $= 10$ **then**
        Find $\boldsymbol{c}_{min}$, the configuration with the minimum estimated potential energy
        $e_{min} \leftarrow DFT(\boldsymbol{c}_{min})$
        $\boldsymbol{e} \leftarrow$ append($\boldsymbol{e}, e_{min}$)
        $\mathcal{X}_{DFT} \leftarrow$ append($\mathcal{X}_{DFT}, \boldsymbol{c}_{min}$)
        model $\leftarrow GP(\mathcal{X}_{DFT}, \boldsymbol{e})$
        $\hat{\boldsymbol{e}} \leftarrow GP.predict(\mathcal{X})$
        iter $= 0$
        if $\boldsymbol{c}_{min}$ has not changed since the last 10 DFT computations **then**
          flag $= 0$
        end if
        $e_{min} \leftarrow DFT(\boldsymbol{c}_{min})$
        $\boldsymbol{e} \leftarrow$ append($\boldsymbol{e}, e_{min}$)
        $\mathcal{X}_{DFT} \leftarrow$ append($\mathcal{X}_{DFT}, \boldsymbol{c}_{min}$)
        model $\leftarrow GP(\mathcal{X}_{DFT}, \boldsymbol{e})$
        $\hat{\boldsymbol{e}} \leftarrow GP.predict(\mathcal{X})$
        iter $= 0$
      end if
    end if
  end if
  $t \leftarrow 0.5(t + d_{min})$
end while
Output $\mathcal{C}, \mathcal{X}$

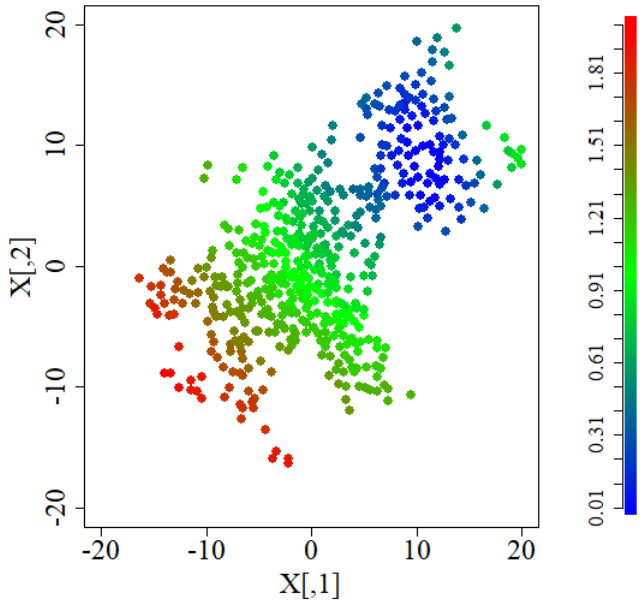

Figure 2.9: Adaptive expansion further expands the low-energy region. This is for the variation of the toy example with $N_{max} = 400$, the unknown fingerprint domain space as $[-20, 20] \times [-20, 20]$, and the DFT function assumed to be as in (Equation 2.9).

2.4.3 Exploration and exploitation of the domain space: Bayesian optimization

The purpose of this step is to identify the crystal structure configuration with the least potential energy in this candidate set of size N obtained from the expansion steps. The naive method is to compute the energy of each of the N fingerprints in the candidate set, and find the one with the minimum energy. However, as mentioned in Section 3.3, energy computation using the density functional theory (DFT) is too expensive, and so such a method will be impractical to use for a high value of N. So, we will use the Gaussian process model developed during the adaptive expansion procedure to estimate the energy of all the N fingerprints. Bayesian optimization [65] will then be used to iteratively optimize and update the model. The method will let us identify the global minimum with energy computations over only a small fraction of N fingerprints in the candidate set.

Bayesian optimization involves iterative improvement of the global minimum estimate based on the surrogate model. Let $\boldsymbol{e}$ be the vector of potential energy, computed using DFT, for n fingerprints, and $s(.)$ be the standard error of their energy estimate. Then, the expected

improvement of the global minimum estimate at $\boldsymbol{x}$ can be expressed as the following closed form jones1998efficient:

$$EI(\boldsymbol{x}) = [\min(\boldsymbol{e}) - \hat{e}(\boldsymbol{x})]\Phi\left(\frac{\min(\boldsymbol{e}) - \hat{e}(\boldsymbol{x})}{s(\boldsymbol{x})}\right) + s(\boldsymbol{x})\phi\left(\frac{\min(\boldsymbol{e}) - \hat{e}(\boldsymbol{x})}{s(\boldsymbol{x})}\right), \quad (2.10)$$

where Φ and ϕ are respectively the cumulative distribution function and the probability density function of the standard normal distribution. As the surrogate model is cheap, we evaluate (Equation 2.10) on all the N fingerprints in the candidate set, and find the one that maximizes it:

$$\boldsymbol{x}_{new} = \arg\max_{\boldsymbol{x}} EI(\boldsymbol{x}). \quad (2.11)$$

DFT is used to compute the potential energy at $\boldsymbol{x}_{new}$, and the known-data set is updated to include $[\boldsymbol{x}_{new}, e(\boldsymbol{x}_{new})]$. Then, we use (Equation 2.8) to update our surrogate model based on the updated known-data.

We continue repeating the exercise of finding a new fingerprint using (Equation 2.11), computing the potential energy for this fingerprint, and updating the surrogate model using (Equation 2.8). Ideally, the algorithm should stop when it spots the global minimum. However, in practice, we will not know when we have found the global minimum. One option is to perform as many iterations as per the maximum available computation budget. However, [75] argue that such an approach may lead to wastage of resources, and it is reasonable to stop iterations if the expected improvement is less than a user-specified small threshold constant t_{EI}. The choice of t_{EI} depends on the problem. For example, for our real example (details deferred to Section 2.5) we chose to stop iterations when the expected improvement became less than 0.001% of the current minimum potential energy estimate. Once the algorithm stops, the fingerprint with the least potential energy in the known-data is considered to be the most stable among the candidate set of N fingerprints.

The Bayesian optimization algorithm is summarized as algorithm 6.

The Bayesian optimization algorithm addresses the remaining two challenges in Section

Algorithm 6 : Bayesian optimization

1: Import $\mathcal{C}, \mathcal{X}$ {Obtained from the expansion algorithms}
2: Input t_{EI}
3: Lines $2-5$ from the adaptive domain space expansion algorithm
4: $e_{min} \leftarrow \min(\boldsymbol{e})$
5: $EI_i \leftarrow EI(\boldsymbol{x}_i); i \in \{1, \cdots, N\}$ {Use (Equation 2.10)}
6: percent_improve $\leftarrow \max(\boldsymbol{EI})/e_{min}$
7: **while** percent_improve $\leq t_{EI}$ **do**
8: $\quad i\_maxEI \leftarrow \arg\max_i(\boldsymbol{EI}); i \in \{1, \cdots, N\}$
9: $\quad e_{new} \leftarrow DFT(\boldsymbol{x}_{i\_maxEI})$
10: $\quad \mathcal{X}_{DFT} \leftarrow$ append$(\mathcal{X}_{DFT}, \boldsymbol{x}_{i_maxEI})$
11: $\quad \boldsymbol{e} \leftarrow$ append$(\boldsymbol{e}, e_{new})$
12: $\quad model \leftarrow GP(\mathcal{X}_{DFT}, \boldsymbol{e})$
13: $\quad EI_i \leftarrow EI(\boldsymbol{x}_i); i \in \{1, \cdots, N\}$ {Use (Equation 2.10)}
14: $\quad$ percent_improve $\leftarrow \max(\boldsymbol{EI})/e_{min}$
15: **end while**
16: $i\_stable \leftarrow \arg\min_i \boldsymbol{e}; i \in \{1, \cdots, nrows(\mathcal{X}_{DFT})\}$
17: Output $\boldsymbol{x}_{i\_stable}, \boldsymbol{c}_{i\_stable}$

$3-3.3$ and 3.4. The algorithm computes the potential energy for only a small fraction of fingerprints in the candidate set. This minimizes the expensive DFT computations, thereby addressing challenge 3.3. As the Gaussian process model is very good in modeling highly non-linear and smooth surfaces, it models the highly multimodel potential energy surface well, which helps us spot the global minimum quickly. Thus, the flexibility and smoothness of our surrogate model addresses challenge 3.4. The expected improvement (EI) criterion balances exploration of the PES with exploitation, thereby simultaneously addressing both the objectives of crystal structure prediction - exploiting low-energy regions to search for the minimum and exploring new and possibly unusual domains of the PES.

2.5 Real example

We will demonstrate our developed methodology on a real example. The objective is to find the most stable crystal structure configuration of Al_8. In other words, we need to search for the global minimum of the potential energy surface that corresponds to the space of all the atomic structures, each of which contains eight Aluminum (Al) atoms arranged in

a parallelepiped unit cell. The extreme point of this PES corresponds to the face-centered cubic (fcc) structure of Aluminum, which is already known (Figure 2.1). The eight-atoms structures were created by (1) randomly choosing a specific value of volume v falling within $\pm 5\%$ of the known specific volume of the Al fcc structure, (2) randomly selecting three vectors $\vec{a}$, $\vec{b}$, and $\vec{c}$ of the unit cell so that its volume, given by $V \equiv \vec{a} \cdot (\vec{b} \times \vec{c}) = 8v$, and (3) randomly arranging the eight Al atoms in the cell so that the distance between any pairs is larger than 2.0 Å. Two constrains (1) and (3) of this procedure, which were formulated from the known facts of the fcc Al structure, clearly limit the examined configuration space but the search domain remains staggering and certainly contains the global minimum.

Within our expansion-exploration-exploitation framework, we start from a finite set of structure configurations (a few hundreds) for which the energy is computed at the DFT level. Then, we aim at (1) adding more structures into the initial set such that the configuration domain space is expanded and filled efficiently and (2) searching for the global minimum of the expanded dataset. Note that the DFT calculations performed herein involve *only single-point energy calculations* but not any local optimizations, which may be $10^3 - 10^4$ times more expensive. Therefore, in general, none of the examined structures is a local minimum of the PES. However, the main objectives of this work, i.e., *(diversely) filling the configuration space and searching for the global minimum of a big and diverse structure dataset*, can be demonstrated and is very useful for material structure prediction.

To obtain a candidate set of fingerprints, we start with the non-adaptive domain space expansion algorithm (Algorithm 1). The algorithm requires three inputs - an initial set of atomic configurations $\mathcal{C}$, their corresponding fingerprints $\mathcal{X}$, and the size of the candidate set of fingerprints N_{max}. We have a set of $I = 270$ known atomic configurations of Al_8. These configurations become the input $\mathcal{C}$, and their corresponding fingerprints become the input $\mathcal{X}$. The fingerprints have a dimension of $p = 32$. As per the thumb-rule mentioned earlier, we take $N_{max} = 100p = 3,200$. Here, we have assumed that the size of the convex hull of the candidate set of fingerprints will keep increasing, or the algorithm will keep

exploring new domain space, as we continue to add fingerprints in the candidate set. This assumption is later found to be true. As we do not expect the candidate set to fully span the huge domain space of configurations, we do not consider it necessary to monitor the convex hull of the candidate set during the non-adaptive expansion algorithm.

As the $AGNI$ fingerprint is 32-dimensional, it cannot be visualized directly. We use Principal Component Analysis [76], or PCA to reduce its dimensionality. We found that the first three PCs captured 97% of the variance of the candidate set of $AGNI$ fingerprints. Thus, the first three PCs are sufficient to visualize the $AGNI$ fingerprints obtained by our non-adaptive domain space expansion algorithm. We will compare them with the PCs corresponding to the fingerprints obtained from the state-of-the-art approach.

Figure 2.10 (left) shows the initial set of fingerprints (grey circles) that are input to the non-adaptive space expansion algorithm (Algorithm 1). The solid black circle (in this and all the subsequent figures) is the most stable fingerprint, or the fingerprint that has the minimum potential energy. This is not a part of the initial candidate set. However, this is the solution that we hope to achieve. Note that there is a relatively large gap between it and the initial set of fingerprints. Ideally, our expansion algorithms will expand the initial candidate set to include the most stable fingerprint, and then the Bayesian optimization algorithm should identify this fingerprint as the global minimum.

Figure 2.10 (center) shows the candidate set of $N_{max} = 3,200$ fingerprints obtained using our non-adaptive space expansion algorithm. There are a couple of points to note regarding the algorithm. First, the algorithm fills the gap between different clusters of fingerprints in the initial candidate set. Second, the algorithm expands the volume of the domain space spanned by the initial candidate set of fingerprints. Thus, the algorithm found fingerprints in unexplored regions of the domain space, and added them to the candidate set, as was desired.

Figure 2.10 (right) shows the candidate set of $N_{max} = 3,200$ fingerprints obtained using the random search-based state-of-the-art approach. In this approach, the fingerprints

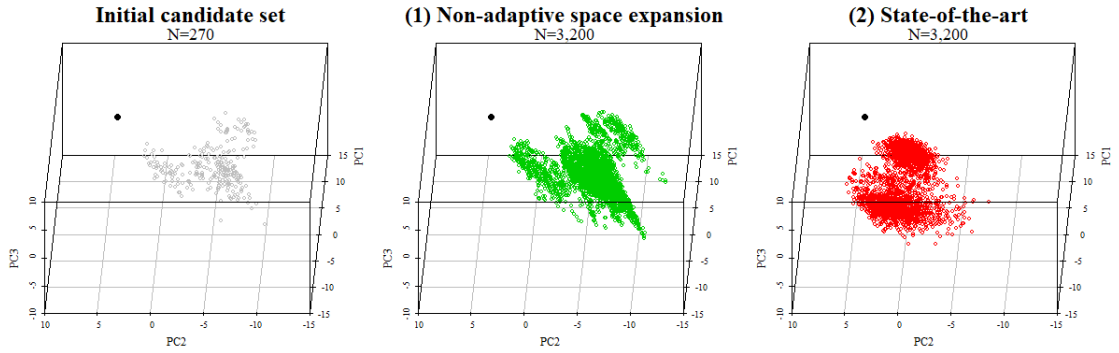

Figure 2.10: (left): Initial set of fingerprints; (center): candidate set of $N_{max} = 3,200$ fingerprints obtained using our non-adaptive space expansion algorithm (Algorithm 1); (right): candidate set of $N_{max} = 3,200$ fingerprints obtained using the random search-based state-of-the-art approach. The solid black circle is the true global optimum, not included in the candidate set.

are randomly perturbed to generate new ones, and added to the candidate set. These fingerprints seem to span a similar volume of the domain space as spanned by our proposed candidate set.

Although for $N_{max} = 3,200$ fingerprints, we did not see a significant difference in the space spanned by the two candidate set of fingerprints (Figure 2.10), this difference becomes larger as the candidate set size increases. Figure 2.11 shows the median distance to the nearest neighbor of a fingerprint as the candidate set size increases. Note that our proposed candidate set keeps expanding to new regions of the domain space with the addition of fingerprints, thereby leading to a minimal change in its proximity to the nearest neighbor. On the other hand, the state-of-the-art approach fails to expand the spanned volume of the domain space, resulting in drastic decrease in the median distance to the nearest neighbor as the candidate set size increases. With the state-of-the-art approach, generating more fingerprints may lead to wastage of resources after a certain point of time as the algorithm restricts itself to the already explored domain space. On the other hand, our non-adaptive expansion algorithm will continue to expand the explored domain space, thereby always providing more information with each newly generated fingerprint.

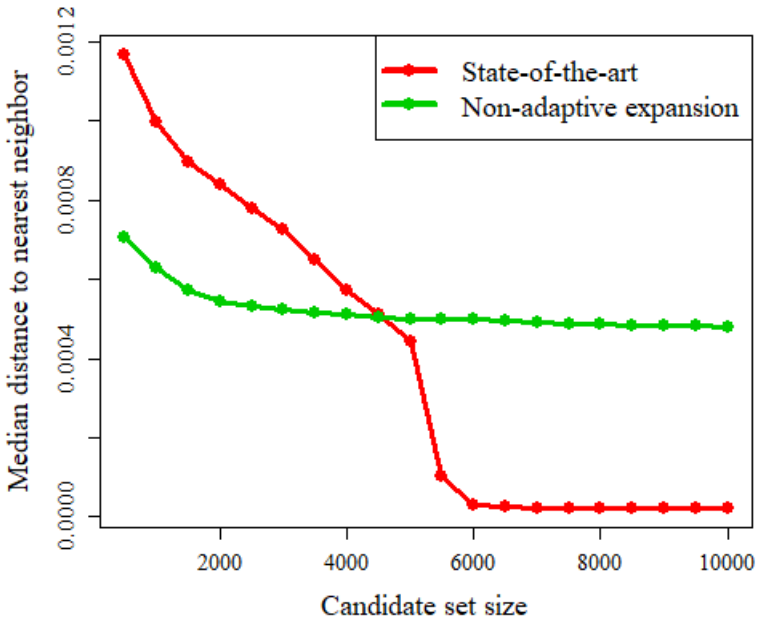

Figure 2.11: Distance to the nearest neighbor vs candidate set size for fingerprints in our candidate set obtained using the non-adaptive space expansion algorithm, and the candidate set obtained using the state-of-the-art approach.

Figure 2.12 shows examples of three configurations corresponding to the fingerprints obtained from the non-adaptive domain space expansion algorithm. The stark differences in these configurations shows that the algorithm spans through quite distinct regions of the domain space.

After obtaining a candidate set of $3,200$ fingerprints from the non-adaptive expansion algorithm, we need to determine if adaptive expansion of the candidate set is needed. We find that the convex hull of the fingerprints has continued to expand throughout the non-adaptive expansion procedure. So, it is likely that there exists unexplored domain space. As we have already exhausted the budget of non-adaptive (or free) expansion, we may need to identify and further expand the low-energy regions of the domain space.

For estimating the low energy regions, a Gaussian process model is developed. To develop the model, DFT is used to compute the potential energy for $10p = 320$ fingerprints. To select these 320 appropriate fingerprints for DFT computations, first we include

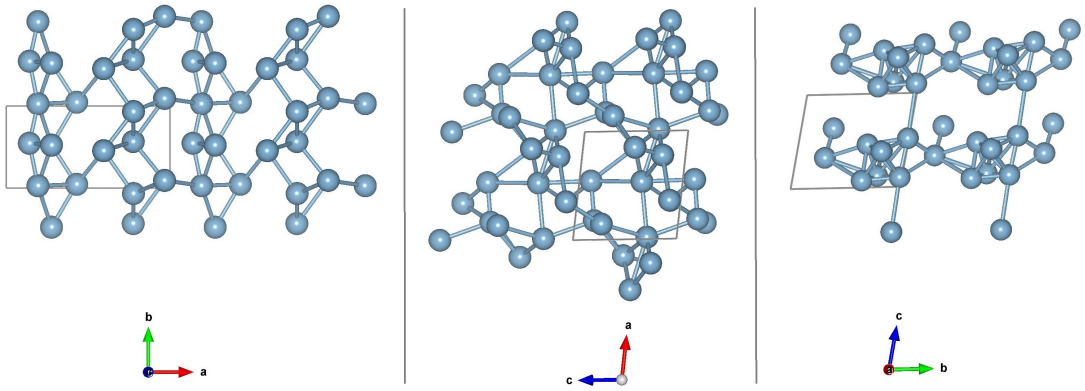

Figure 2.12: Sample configurations from the set of $3,200$ configurations obtained with the non-adaptive domain space expansion algorithm.

the initial candidate set of 270 fingerprints corresponding to known configurations. The remaining 50 fingerprints are chosen from the candidate set by augmenting the set of 270 initial fingerprints with the space-filling MaxPro design. Figure 2.13 shows the initial fingerprints and the ones augmented using the MaxPro design. The MaxPro design ensures that fingerprints for DFT computations are well spread throughout the candidate set.

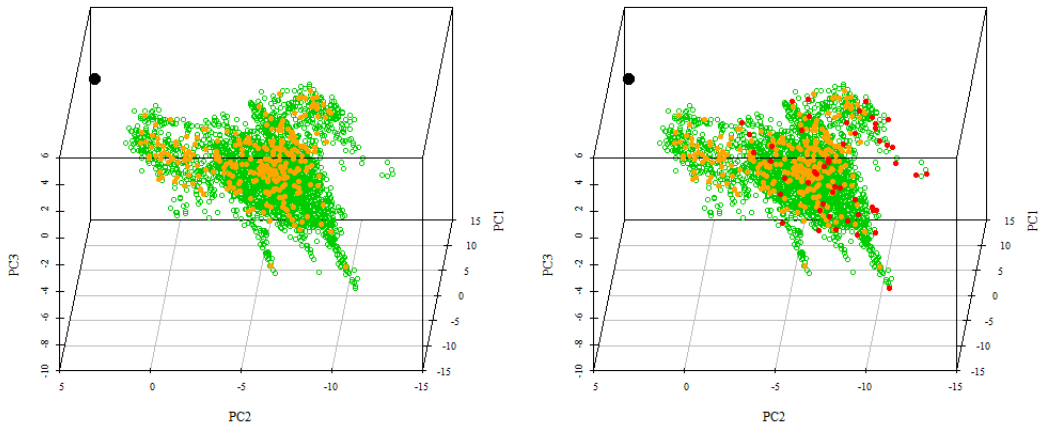

Figure 2.13: (left): Initial set of fingerprints (orange) over the candidate set of fingerprints (green); (right): Initial set of fingerprints augmented by the space-filling MaxPro design (red).

We develop a Gaussian process model based on the data collected by DFT computa-

tions. The model is used to estimate the potential energy of the entire candidate set of fingerprints. Figure 2.14 (left) shows the estimated potential energy contour. A low-energy regions seems to be around the boundary of the spanned domain space. Using the boundary definition mentioned earlier and illustrated in Figure 2.8, we find that the estimated minimum of the candidate set is actually at the boundary of the spanned domain space. So, it is necessary to further expand this low energy region as the it may lead to further minimization of the current estimate of the energy-minimum. Note that we work in the space of the first three principal components for determining the boundary, as it is excessively expensive to work in the 32-dimensional fingerprint space.

The adaptive expansion algorithm (Algorithm 2) is used to further expand the identified low-energy region of the spanned domain space. Figure 2.14 (right) shows the added fingerprints and the updated potential energy contour after the adaptive expansion procedure. The algorithm adds 530 fingerprints to the candidate set, and DFT computations are done for every $10th$ fingerprint added to the set, i.e., for a total of 53 fingerprints. The algorithm stops when the estimated minimum does not change with 10 successive DFT computations. We observe that the algorithm succeeds in expanding the candidate set towards the unknown true global minimum, which is what we desired!

Once a candidate set of fingerprints is obtained by the expansion algorithms, we explore and exploit it for the global minimum of potential energy, using the Bayesian optimization algorithm (Algorithm 3). The algorithm requires two inputs - the candidate set of configurations $\mathcal{C}$ and their corresponding fingerprints $\mathcal{X}$. The input $\mathcal{X}$ in this example is the candidate set of $3,200 + 530 = 3,730$ fingerprints that we obtained using the expansion algorithms.

We will again use principal component analysis (PCA) to visualize the fingerprints obtained in the Bayesian optimization procedure. The circles in Figure 2.15 are the candidate set of $N = 3,200$ fingerprints. The potential energy was computed for a set of $320 + 53 = 373$ fingerprints (in the adaptive domain space expansion algorithm), shown as

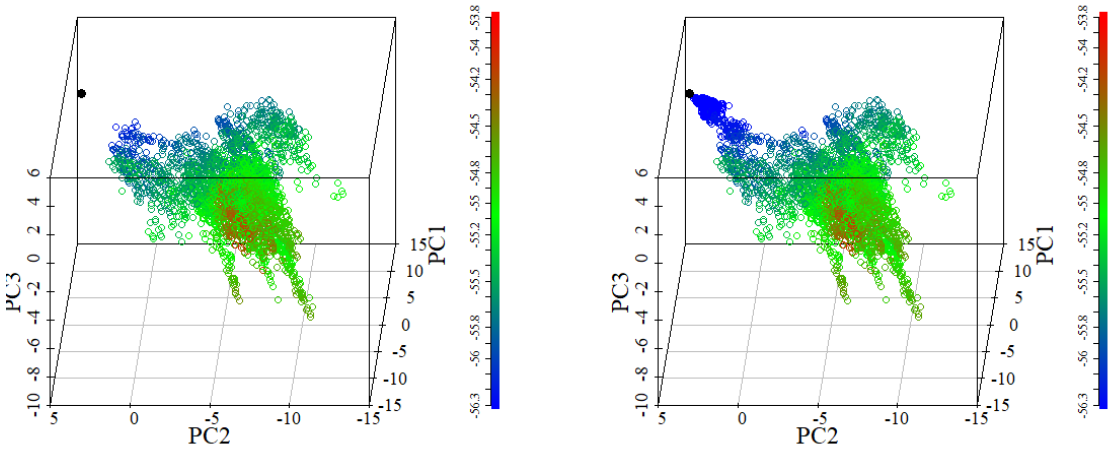

Figure 2.14: (left): Potential energy contour of the candidate set of fingerprints obtained after non-adaptive expansion; (right): Potential energy contour of the candidate set of fingerprints obtained after adaptive expansion.

squares in Figure 2.15 (left). A Gaussian process model was developed using the potential energy data of the 373 fingerprints. The model provided an estimate of the potential energy surface along with uncertainty in the estimate. Then, the Expected Improvement (EI) criterion was used to simultaneously exploit and explore the potential energy surface, to guide the search towards the fingerprint having the minimum potential energy. The iterative procedure of updating the Gaussian process model, and adding a point in the known-data based on the EI criterion is continued until the expected improvement becomes lesser than a threshold value t_{EI}. We chose t_{EI} to be 0.001% of the current minimum value of potential energy in the known-data. Thus, for the ith iteration, the threshold value is:

$$t_{EI} = 10^{-5} \times \min(e_1, \cdots, e_{n+i}). \tag{2.12}$$

This is a reasonably low value as the potential energy in the known-data varies by 5% around its mean. With the above threshold, the algorithm stopped after the $95th$ iteration.

The fingerprints iteratively added in the known-data are shown as triangles in Figure 2.15 (right). There are three points to note about the iteratively added fingerprints.

First, 93 of the 95 fingerprints are selected in the region around the global minimum, which shows that the algorithm does well in exploiting the "promising" region of the domain space. Second, two fingerprints are selected in regions far away from the global minimum. These are regions at the boundary of the domain space, where probably there is high uncertainty in the potential energy estimate of the Gaussian process surrogate model. This shows the exploratory nature of the algorithm, where it tries to find the global minimum in regions other than the "promising region". This exploratory feature of the algorithm makes it better than the state-of-the-art approaches such as basin hopping and minima hopping, which focus only on exploiting the "promising region" of the domain space for the global minimum. Third, though the true global minimum was not a part of the candidate set, we identified the fingerprint closest to it as the global minimum! Thus, the algorithm provided a solution that is potentially very similar to the true global minimum.

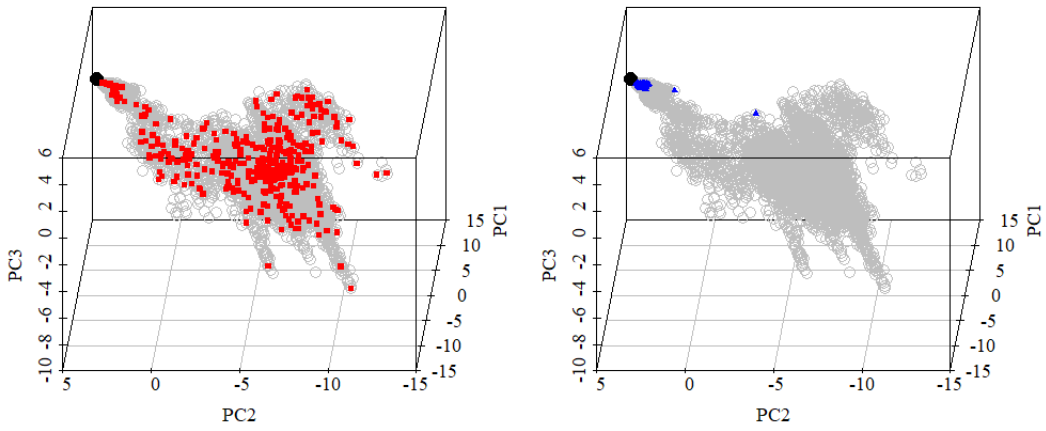

Figure 2.15: Candidate set of $3,200$ fingerprints (circles), (left): initial set of $320 + 53 = 373$ fingerprints for which the potential energy is computed using DFT (squares); (right) iteratively selected 95 fingerprints during the Bayesian optimization procedure (triangles).

Figure 2.16 (left) shows the expected improvement as a percentage of the current minimum estimate of potential energy from the known-data. The expected improvement has a decreasing trend with the number of iterations. As the algorithm learns the potential energy

surface and exploits promising locations for global minimum, a lesser improvement in the current global minimum estimate is expected in further iterations. Figure 2.16 (right) shows the potential energy of the first $n = 320$ observations of the known-data (black circles), followed by that of the 53 fingerprints (red circles) iteratively added to the known-data during the adaptive expansion procedure, which are in-turn followed by the 95 fingerprints (blue) added to the known data during the Bayesian optimization procedure. This figure makes it clear that the non-adaptive expansion algorithm expands the domain space without considering the potential energy, the adaptive expansion algorithm drives the expansion towards lower-energy regions, and the Bayesian optimization procedure explores the candidate set and exploits the low-energy regions for the global minimum of potential energy.

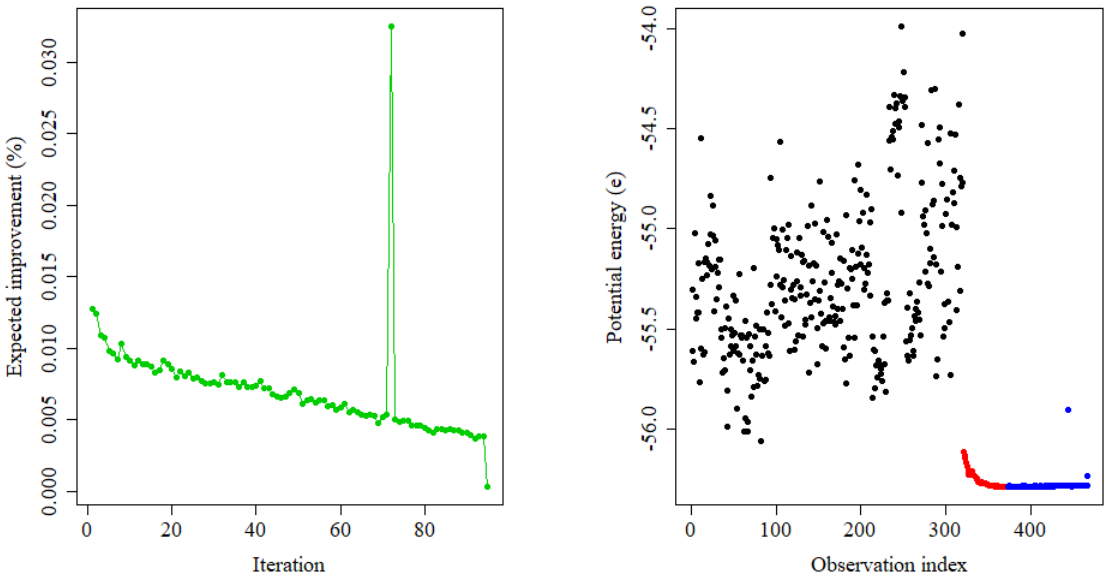

Figure 2.16: (left): Percentage of expected improvement with respect to the current energy minimum estimate; (right): Potential energy of all the fingerprints in the known-data - for the initial known fingerprints (in black), for the fingerprints added by non-adaptive expansion (in red), for the fingerprints added during the Bayesian optimization procedure (in blue).

The algorithm's output and our solution is the configuration corresponding to the fingerprint selected in the $368th$ DFT computation, as it has the minimum potential energy in the known-data. Note that this configuration is the most stable only in our candidate set

of configurations, and is not necessarily the most stable configuration. Material scientists need to perform the DFT-based local optimization at this configuration to find the closest local minimum. Figure 2.17 shows the estimated stable configuration of Al_8, as obtained by our expansion-exploration-exploitation framework. Our solution appears to be very similar to true stable configuration of Al_8 (Figure 2.1). It seems that with the DFT-based local optimization, our estimated configuration will converge to the true configuration!

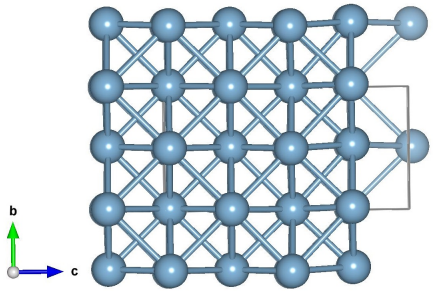

Figure 2.17: (left): Configuration with the least potential energy estimated by our expansion-exploration-exploitation framework.

2.6 Conclusion

We have developed an active learning method to obtain the most stable crystal structure configuration of a crystalline material given its chemical formula and the underlying thermodynamic conditions - temperature and pressure. The novelty of our approach is the expansion-exploration-exploitation framework that extends the traditionally used exploration-exploitation Bayesian optimization framework to better achieve the objectives of crystal structure prediction. The expansion algorithms ensure that the candidate set continues to expand with the addition of each fingerprint - first in arbitrary directions (with respect to potential energy) to explore new and possible unusual domains of the PES, and then towards lower-energy atomic configurations. The Bayesian optimization procedure brings a balance to the contrasting components of exploration and exploitation. Methods that tend to consider all possible atomic configurations, such as random-search [49], fail

in case of larger systems. The focus of these methods is skewed towards exploration. On the other hand, methods that consider only "promising regions" such as minima-hopping [56] fail in case of unexpected stable crystal structure configurations. The focus of these methods is skewed toward exploitation.

The first part of our method is the non-adaptive domain space expansion algorithm, which provides a candidate set of atomic configurations. There are two novel features of this algorithm. First, the expansion is focused towards the under-represented regions of the domain space. This makes the candidate set generation more efficient as compared to approaches that are guided by randomness [48]. Second, our algorithm provides a space-filling design without the knowledge of the boundaries of the design space. This is a novel contribution to the field of experimental design, where most of the work on space-filling design [24] is focused on cases of known design space.

The second part of our method is the adaptive domain space expansion. This step ensures that the local minima of all the low-energy regions are included in the candidate set. Even if the non-adaptive expansion algorithm "touches" a low-energy region, the corresponding local minimum will be included in the candidate set.

The third part of our method is the Bayesian optimization algorithm. For Al_8, the algorithm identified the most stable crystal structure configuration with additional DFT computations for only 95 or 3% of the candidate set configurations. This shows that once a representative candidate set of fingerprints is identified, the optimization is very efficient. This also shows that the Gaussian process model is appropriate for modeling the highly non-linear and multi-model potential energy surface.

Although our method worked well for Al_8, there are a couple of challenges that still need to be addressed. First, the adaptive space exploration algorithm involves a fixed perturbation of the selected atomic configuration. This means that even if a large part of the domain space is unexplored, the algorithm will take relatively small steps to fill the space. The amount of perturbation should be adaptive, instead of fixed. The perturbation should

initially be higher to rapidly span through the domain space. Once the convex hull of the candidate set of fingerprints tends to stop expanding, then the perturbation amount must be reduced. This will ensure that the entire domain space is explored even with a low budget of the candidate set size N. Second, the Expected Improvement criterion is known to be too greedy [77] in terms of favoring exploitation a lot more than exploration. The generalized Expected Improvement criterion [78] involves an additional parameter that determines the balance between exploration and exploitation. That parameter needs to be tuned to a value that works best for crystal structure prediction.

Although we demonstrated our approach on a simple problem, the new concepts are powerful and can easily be generalized to more realistic problems. However, heavy calculations, e.g., DFT-based local optimizations, which may contain $10^2 - 10^3$ single-point DFT calculation, were avoided in this work. Therefore, further critical developments are still needed for our approach to be used as a new structure prediction method.

We will address these challenges in a future work.

CHAPTER 3
DESIGN OF ACOUSTIC METASURFACES WITH INDEPENDENT AMPLITUDE AND PHASE CONTROL

3.1 Introduction

Acoustic metasurfaces are structures that modulate the acoustic properties, namely the phase and amplitude of sound waves. When a sound wave strikes an acoustic metasurface, a part of it may be reflected from it, while the remaining part may be transmitted through it. Thus, for a given acoustic metasurface (or the input), there are four acoustic outputs - (1) Reflection amplitude, (2) Reflection phase, (3) Transmission amplitude, and (4) Transmission phase. We will mention these as acoustic outputs from hereon, for brevity. These outputs rely on the design of the geometry of the acoustic metasurface. In this work, we have developed a method that provides the appropriate acoustic metasurface design to achieve the desired acoustic outputs.

Physically, acoustic metasurfaces are an artificial sheet of material with subwavelength thickness formed by different unit cells to modulate propagating sound waves [79]. The ultrathin feature of the metasurfaces allows wave-matter interactions and wave manipulations within a deep subwavelength space. This deep subwavelength wave modulation functionality has enabled many applications of metasurfaces, including beam steering and forming [80, 81], perfect sound absorbers [82, 83], high-speed communications and ground cloaking [84, 85], all of which are critical for the development of underwater exploration, non-invasive biomedical treatments, architectural designs, and noise control. In most of these applications, independent control of the acoustic amplitude and phase profile is crucial.

Despite the importance of metasurfaces, most existing designs lack the capability to

control the acoustic amplitude and phase independently. This is critical for most practical applications where the generation of an arbitrary wave pattern is significant for applications such as, acoustic illusions [85], holography [86], signal multiplexing for high-speed communications [87, 88], particle levitation and motion control [89]. A recent experimental study has suggested the introduction of controlled energy loss in the design of metasurfaces to realize a decoupled modulation of sound amplitude and phase [90]. However, the intrinsic loss resulted from this design method limits the efficiency of acoustic wave modulation. Yet, the additional loss parameter increases the complexity of the problem when compared to previous geometry-based metasurface designs that prevent the development of a systematic analytical design methodology for arbitrary acoustic metasurfaces with independent amplitude and phase controllability for different applications. Moreover, the loss design demonstrated can only be used for the modulation of reflected acoustic waves [90].

Let us consider the challenge in designing an acoustic metasurface for independent phase and amplitude modulation. We know that change in the geometry of the metasurface usually leads to the variation in acoustic outputs. A systematic discretization method is typically used to characterize the geometry of the acoustic metasurface and study its effect on the amplitude and phase of the transmitted and reflected acoustic waves. Because of the large number of possible geometries, performing simulations on all of them is practically infeasible. For example, a 5-by-5 meshed unit cell can produce more than 33 million possible geometries. This makes it challenging to understand the dependence of the acoustic outputs on geometry, and use it to identify the geometry required for the desired acoustic outputs. The state-of-the-art machine learning models are all data-driven models and do not obey physical principles. Therefore, even if we develop a model on a sample of simulated geometries, it will not perform well when extrapolation in the model space is needed.

In this work, we have proposed an approach to address the above challenges, and modulate the acoustic outputs independently. All possible geometries are characterized using a spatial discretization method. We propose a novel machine learning algorithm based on

the non-adaptive expansion algorithm described in Chapter 2. The algorithm starts from a set of few initial geometries, and then uses them to select geometries that efficiently fill the space of the desired acoustic outputs. The key feature of the algorithm is that it iterates through only a small fraction of the possible number of geometries to find the set of geometries that span the entire domain space of the desired acoustic outputs. Then, given the desired output, we can pick the geometry that corresponds to the output. Note that the domain space of transmitted and reflected amplitudes is $[0, 1]$, while the respective phases are in $[-180°, 180°]$.

This chapter is organized as follows. In Section 3.2, we describe the assumptions behind the acoustic metasurface modeling, characterization of the acoustic metasurface geometry, and the full wave simulation setup. In Section 3.3, we mention the objectives and preliminary simulation results. In Section 3.4, we describe the redundancies in the acoustic metasurface geometry with regard to the acoustic outputs. In Section 3.5, we describe our proposed methodology to achieve the objectives. In Section 3.6, we present the results obtained. We conclude the chapter with some remarks in Section 3.7.

3.2 Modeling the acoustic metasurface and simulation setup

3.2.1 Assumptions

Comparing the impedance of the solid materials used in the fabrication of metasurfaces, the acoustic impedance of air is significantly smaller. This strong acoustic impedance mismatch allows us to assume the structures formed by solid materials are hard wall acoustic boundaries. Another assumption in the design of the acoustic metasurface unit cells is that the couplings between the unit cells are assumed to be zero, where the acoustic metasurface is an array of repeated unit cells. This is a common assumption in metasurface designs. The coupling between the unit cells has been shown to be insignificant in many previous experiments with pure phase modulating metasurfaces [80, 81, 91, 92, 93, 94, 95, 96, 97]. Meanwhile, the air is assumed to be non-slippery at the interface between the solid and

air, and the classical fluid boundary layer assumption is used in the modeling to calculate the thermoviscous loss when small air channels exist in the design. Based on these two assumptions, we will perform the full wave simulations in COMSOL multiphysics.

3.2.2 Characterizing the geometry of an acoustic metasurface

To characterize all possible geometries for the design of an acoustic metasurface, we will discretize the two-dimensional continuous space into a finite number of elements. Note that this design method can be generalized for the development of three-dimensional structures as well. The size of the unit cell will first be specified in a rectangular region whose sides are functions of the wavelength of the target acoustic waves, as shown in Figure 3.1. The height of the unit cell is chosen to be 1/3 of wavelength $(\lambda/3)$ to cut off higher order acoustic waveguide modes. Because the goal of the development of metasurfaces is to modulate acoustic waves with the smallest possible space, the width of the specified rectangular region is specified to be $\lambda/3$.

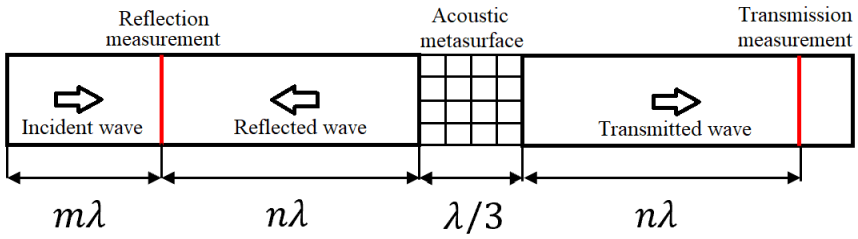

Figure 3.1: Single unit cell simulation in a one dimensional waveguide.

We discretize the unit cell into a grid of mn equal-sized square-shaped elements, where m is the number of elements perpendicular to the waveguide, and n is the number of elements along the length of the waveguide. Figure 3.1 shows a unit cell with $m = n = 4$, resulting in 16 elements. Each element can either be filled with a solid material or be empty. The solid material is assumed to act as a hard wall acoustic boundary, while an empty element is filled with air, where the acoustic waves can propagate.

We encode an element filled with a solid material as 1, while an empty element as 0.

Thus, the geometry of the unit-cell, which serves as the input, is an $m \times n$ grid of 0s and 1s. We need to choose an optimal grid size, i.e. optimal values for m and n, to perform the simulations. If the grid size is too small, there may not be enough number of distinct geometrical structures that can span the desired range of the amplitude and phase outputs. We wish to modulate amplitude to any value in $[0, 1]$ normalized by the incident wave, and the phase to any value in [-180º, 180º]. On the other hand, if the grid size is too large, it may lead to unnecessarily large number of geometries, leading to redundant outputs. For a grid size of $m \times n$, the number of possible geometries of the unit cell, where each element can be either 0 or 1, is 2^{mn}. For example, for a grid size of 4×4, the number of possible geometries is 216 = 65,536. Performing simulations for all these geometries can be very time consuming. Thus there are two important questions to answer: (i) what is the minimum grid size (m and n) that can generate the phase and amplitude for both reflection and transmission in the desired range with reasonable precision? and (ii) Given a grid size, which geometries (out of 2^{mn}) should we consider to achieve the desired span of acoustic outputs.

3.2.3 Full wave simulations

The full wave simulations of acoustic wave modulation by a single metasurface unit cell are performed in the Pressure Acoustic Module of COMSOL Multiphysics. Because we assume that the coupling between the unit cells is zero, each individual unit cell can be simulated separately. In order to calculate the amplitudes and phases of propagating acoustic waves transmitted and reflected by each unit cell, we will apply a one-dimensional acoustic waveguide as shown in Figure 3.1. The top and bottom boundaries will be set to be sound hard boundaries, which confines the acoustic energy inside the one-dimensional waveguide. An acoustic plane wave will be incident from the left end of the waveguide, whose phase will be set such that the incident acoustic phase is zero at the left boundary of the metasurface unit cell. The transmission and reflection will be calculated by the pres-

sure fields along two vertical lines with a distance to be an integer multiple of the targeted acoustic wavelength $(n\lambda)$ from the right and left boundaries of the metasurface unit cell, respectively, to avoid near-field effects. Note that the incident acoustic wave is known and the reflected acoustic wave equals the difference of the calculated pressure field and the incident acoustic wave at the same line. Because the distance between the lines used for the calculation of transmission and reflection and the boundaries of the unit cell is an integer multiple of wavelength, the acoustic phases retrieved are exactly the same as the phases at the boundaries of the unit cell, and therefore, identical to the transmitted and reflected phases. Figure 3.2 visualizes a COMSOL simulation for a 4×4 geometry. We can see the near-field effects at the boundaries of the metasurface. However, these effects disapper as we move farther away from the metasurface, and the amplitude and phase stablize across the cross-section of the wave. This modeling method is a classical method for the analysis of metasurface performance used in most previous studies. The amplitudes and phases of acoustic waves transmitted and reflected from all the geometries in the unit cell are the acoustic outputs that we intend to control.

3.3 Objectives and preliminary simulation results

Now that we have described the modeling framework, we will mention the precise objectives of this work. As described in the previous section, the larger the grid size, the higher will be the number of possible geometries. Higher number of geometries will imply more distinct paths for the acoustic wave, which in turn will lead to a broader range of acoustic outputs. However, as the grid size increases, it becomes more expensive to physically construct the acoustic metasurface. It also increases the redundancy in the acoustic outputs (described later) obtained from distinct geometries. Thus, our first objective is to find the smallest grid size (or the optimal grid size), $(m \times n)_{optimal}$, for which the observed acoustic outputs span over the entire domain space of the desired outputs. Once the optimal grid size is obtained, our second objective is to develop an algorithm to identify the geome-

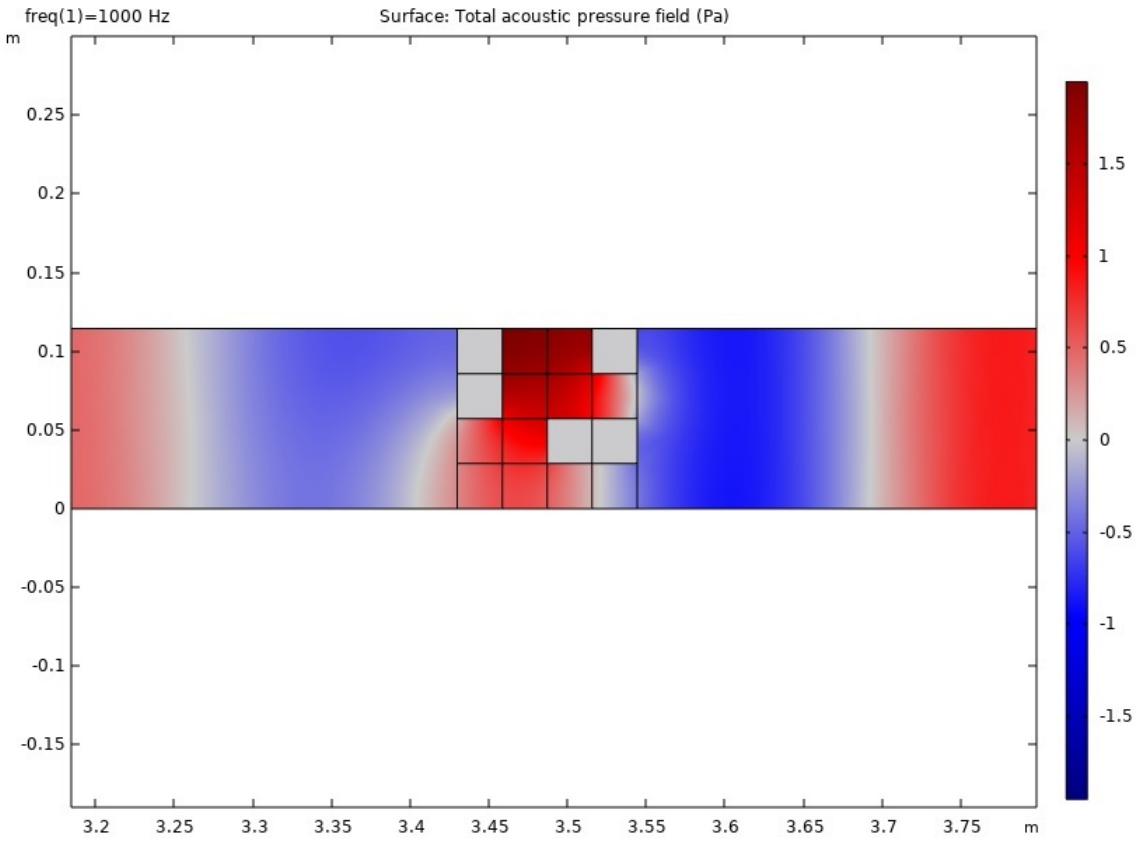

Figure 3.2: Example visualizing a COMSOL simulation for a 4×4 geometry.

try, based on the optimal grid size, that can produce the desired acoustic outputs, or the desired combination of reflection amplitude, reflection phase, transmission amplitude and transmission phase. Note that all the acoustic outputs except the reflection and transmission amplitudes are independent of each other. The reflection and transmission amplitudes are related by the principle of conservation of energy (explained later). Thus, our second objective is to provide geometries that can independently control three acoustic outputs - transmission phase, reflection phase, along with either the transmission amplitude or the reflection amplitude.

For finding the optimal grid size that can cover the entire range of the desired acoustic outputs, we will first consider small grid sizes for which it is practically feasible to simulate all possible geometries on the COMSOL Multiphysics software. We perform a

set of preliminary simulations for which we consider grid sizes containing a maximum of 16 elements, which corresponds to a maximum of $2^{16} = 65,536$ distict geometries. This included grid sizes such as $4 \times 2, 8 \times 2, 2 \times 8$ and 4×4.

Figure 3.3 and Figure 3.4 visualize the simulation results for all the grid-sizes under consideration. These results help us make a couple of key inferences. First, it is clear that even the largest grid sizes with 16 elements are not sufficient to span the entire domain space of the reflection and transmission outputs. Thus, we infer that the optimal grid size will have more than 16 elements. Second, the spanned space of acoustic outputs increases with increase in the grid size. However, for grid-sizes with the same numbers elements, the increase in spanned space for both the transmission and reflection outputs seems to be the highest when $m = n$. Thus, we infer that the optimal grid size will have $m = n$. In other words, our first objective is to find the optimal grid-size having the form $(n \times n)_{Optimal}$.

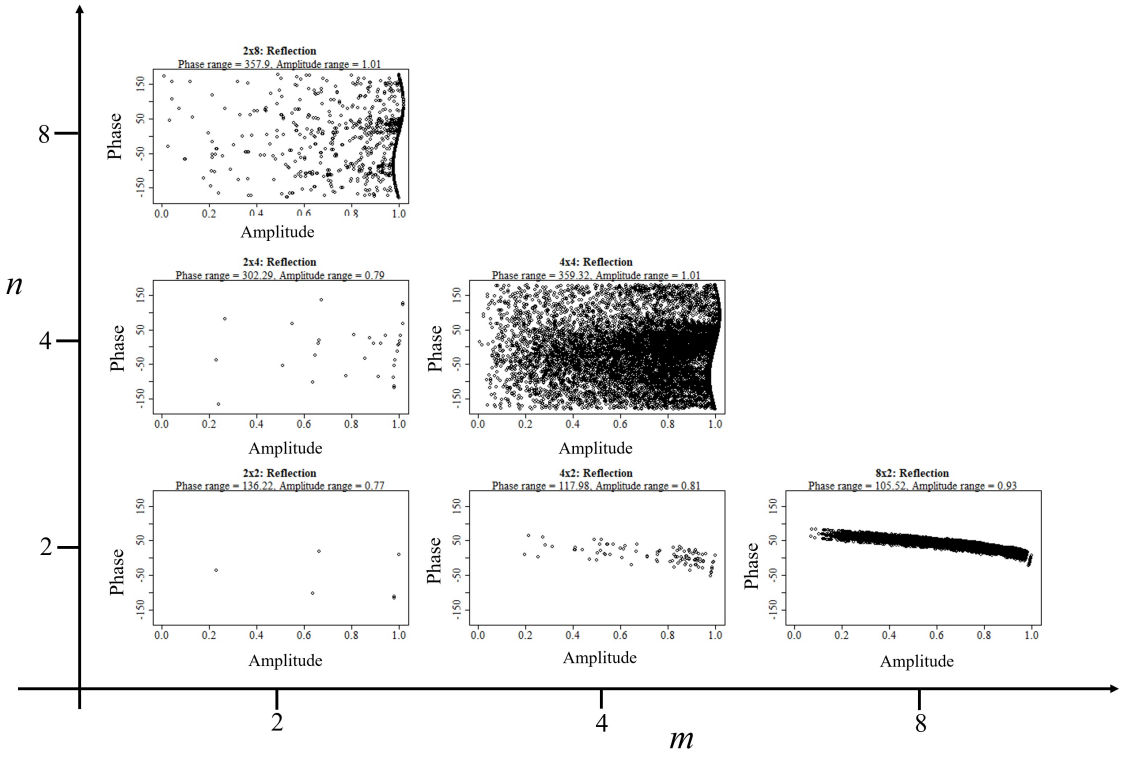

Figure 3.3: Visualizing reflection outputs for different grid sizes of the unit cell.

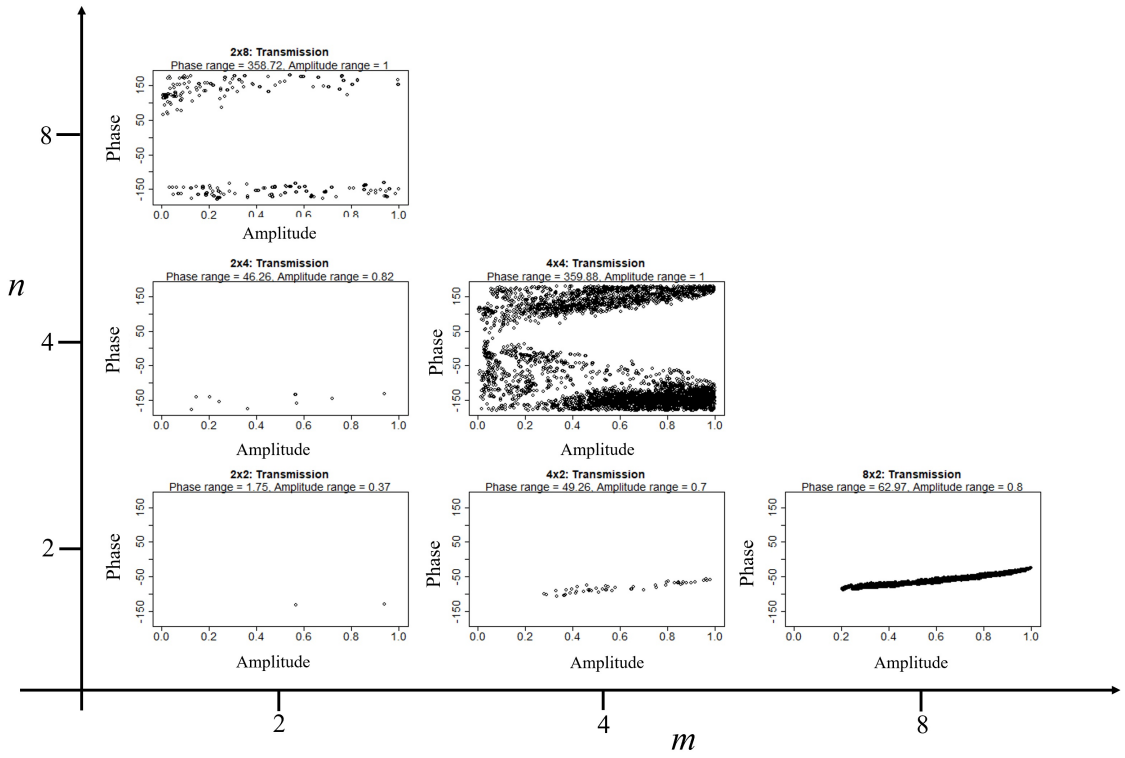

Figure 3.4: Visualizing transmission outputs for different grid sizes of the unit cell.

Figure 3.5 shows the output from the simulation of a 4×4 grid as a scatterplot matrix. We can see an almost perfect relationship between the amplitudes of reflected and transmitted waves. This is expected because in our simulations we have used lossless acoustic elements. According to the conservation of energy in a lossless acoustic system, the two amplitudes satisfy the relationship:

$$T^2 + R^2 = I^2, \tag{3.1}$$

where $T, R,$ and I are the amplitudes of transmission, reflection, and incidence, respectively.

Analyzing these results, we observed that some distinct geometries provide the same acoustic outputs. It appeared that due to certain geometrical symmetries, there was redundancy in geometries. In the next section, we will show that discarding such redundant

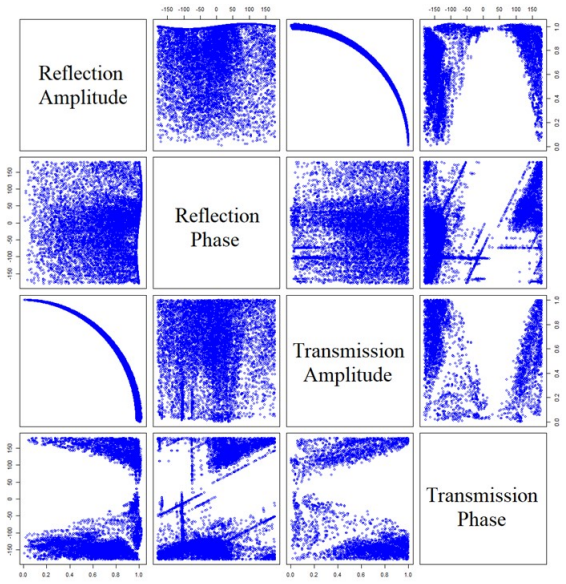

Figure 3.5: Scatter plot matrix of output data from the simulation (with lossless acoustic elements) of a 4×4 grid.

geometries from consideration greatly reduces the total number of possible geometries. Such a reduction is critical for analyzing geometries with higher grid-sizes.

3.4 Redundancies in acoustic metasurface geometry

A big proportion of the possible 2^{mn} geometries are redundant, producing the same output, and thus can be ignored. Redundancy is present because of the following reasons, which we explain using the 4×4 grid as an example.

a. Vertical flip: If the geometry is flipped vertically, then it produces the same output. For example, geometry pairs such as those shown in Figure 3.6 are redundant, and one of each pair can be discarded. We remove the redundant geometries, which results in data reduction by around 50%. For the 4×4 example, the number of geometries reduce from $65,536$ to $32,896$.

b. Redundant solid material: In some cases, the solid material in an element of the grid may be redundant. This may happen when the acoustic wave does not strike the element, when other elements "shield" the element under consideration. Figure 3.7 shows examples

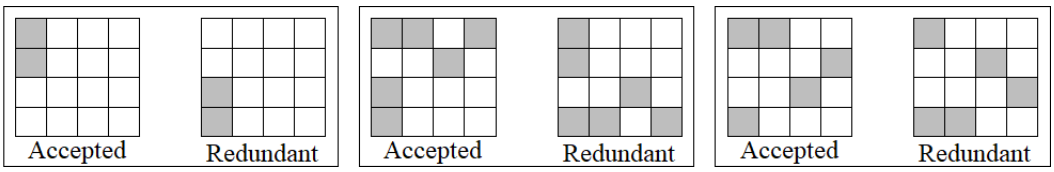

Figure 3.6: Geometries flipped along the horizontal axis produce the same output.

of redundant geometries. For the 4×4 example, the number of geometries further reduce from $32,896$ to $10,146$.

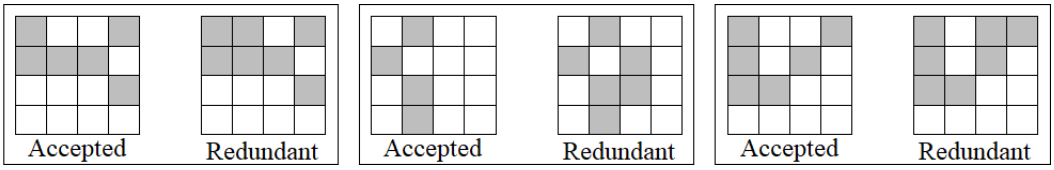

Figure 3.7: Geometries with redundant solid materials produce the same output.

c. Translation of sub-geometry: If the entire set of solid cells shifts along the direction of the incident wave within the unit cell, then all the acoustic outputs are same, except the reflection phase. The reflection phase changes by a constant per unit translation of the solid unit cells. Figure 3.8 shows an example with this type of redundancy. For the 4×4 example, the number of geometries further reduce from $10,146$ to $8,912$.

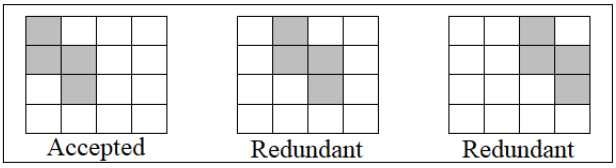

Figure 3.8: Translation of the set of solid unit cells along the direction of the acoustic waves produces the same or perfectly correlated output.

d. Vertical flip along multiple horizontal axis: This redundancy is found in geometries that are symmetric along the central horizontal axis, or produce the same geometry with a vertical flip. If the upper half and lower half of these geometries are flipped along their respective horizontal axis, then the same acoustic outputs are obtained. Figure 3.9 shows

examples of such redundant geometries. For the 4×4 example, the number of geometries further reduce from $8,9126$ to $8,838$ leading to a total reduction of 87% in the data.

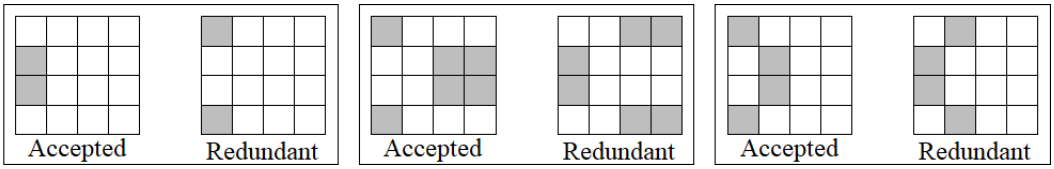

Figure 3.9: Geometries that are symmetric along the central horizontal axis lead to the same output when their upper and lower halves are flipped vertically along their respective horizontal axis.

e. Horizontal flip: The transmission amplitude and phase do not change on flipping the geometry horizontally along its central vertical axis, as the acoustic wave still passes through the same path, albeit in the opposite direction. Figure 3.10 shows examples of such redundant geometries. For the 4×4 example, the number of geometries with regard to transmission further reduce from $8,838$ to $4,184$ leading to a total reduction of 94% in the transmission data.

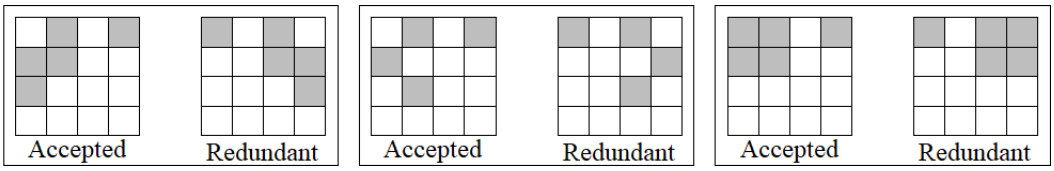

Figure 3.10: Geometries flipped along the vertical axis produce the same transmission output.

The algorithms to eliminate each of the above redundancies are given in Appendix A.

3.5 Methodology: Acoustic domain space expansion algorithm

Although identification of redundancies eliminates a large proportion of the possible geometries, the reduction is still insufficient to consider higher grid-sizes. For example, let us consider the grid-size of 5×5. The total number of geometries before reduction is $2^{25} = 33$ million. Even with a 99% reduction in the total number of geometries under con-

sideration, it will be infeasible to consider the remaining 1%, or 330 thousand geometries. Thus, elimination of redundancy may assist, but will not suffice to achieve our objectives.

As we cannot simulate even all possible non-redundant geometries, we should identify and simulate only those that will be sufficient to fill the domain space of the acoustic outputs. This problem is similar to the materials crystal structure prediction problem of Chapter 2. For crystal structure prediction, we wanted to have a set of fingerprints that spanned the entire domain space of fingerprints. To obtain such a set, we used the non-adaptive expansion algorithm (Algorithm 4). The algorithm began with a few known configurations, and perturbed the configuration corresponding to the most sparsely located fingerprint. This generated a fingerprint in the lesser represented regions of the domain space, and either expanded or filled a gap in the domain space spanned by the set of fingerprints. Similarly, in this problem, we wish to obtain a set of acoustic outputs that span their entire domain space. We are performing COMSOL simulation to find the acoustic outputs, instead of fingerprinting a crystal structure configuration to find the fingerprint. We will start with simulating the outputs of a few geometries. Let $\mathcal{G} = \{\boldsymbol{g}_1, \cdots, \boldsymbol{g}_I\}$ denote the set of I initial geometries, and $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_I\}$ denote the set of their corresponding acoustic outputs. Then, the geometry corresponding to the most sparsely located acoustic output will be "perturbed" (perturbation of a geometry will be explained later). This will generate another geometry whose output is likely to be in the sparsely populated region of the output domain space. This in turn will lead to filling the "gaps" in the domain space of the desired acoustic outputs.

Te purpose of perturbation is to produce a geometry that produces an output that is similar to the output of the geometry that is being perturbed. Intuitively, a similar geometry is likely to produce a similar output. So, we perturb the chosen geometry in one of the following ways to produce a similar geometry: (1) Change one of the unit-cell elements from solid to empty, (2) Change one of the unit-cell elements from empty to solid, or (3) Interchange the position of a solid element with a neighboring empty element. The type of

perturbation is randomly chosen from the above three types. Figure 3.11 shows an example of the three types of perturbation.

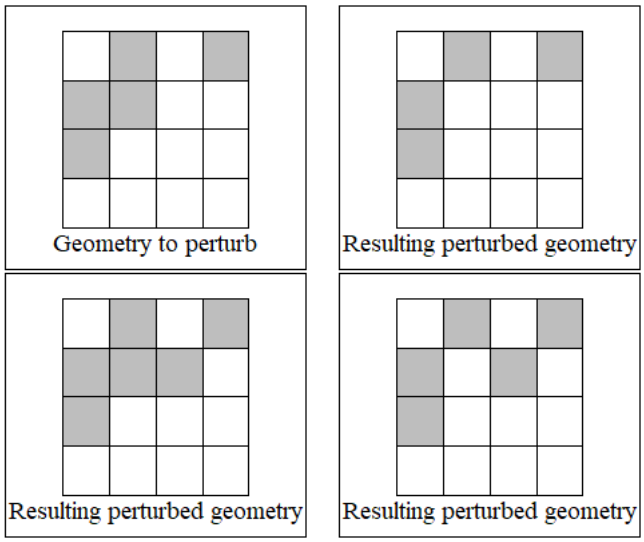

Figure 3.11: Examples of geometries obtained by perturbation of a geometry.

Even though the non-adaptive expansion algorithm (algorithm 4) can be directly used for this problem, there are three unique characteristics of this problem which help us in making the algorithm much faster. First, we know the domain space of the desired outputs. So, the algorithm does not need to "learn" the boundaries of the domain space, which saves several iterations. Second, after obtaining the perturbed geometry, we perform a redundancy check, and discard the geometry if it is redundant (as described in Section 3.4) with any of the geometries already existing in the set of geometries for which acoustic outputs have been obtained using COMSOL simulations. Thus, each simulated geometry produces a unique acoustic output, which leads to a faster filling of the domain space. Third, since the domain space of the desired acoustic outputs is known, we know when the outputs have spanned the entire domain space. This gives us a precise stopping criterion. We stop the algorithm when the largest gap in the domain space is lesser than the tolerance, say tol, of the desired acoustic outputs. Although there are four desired acoustic outputs, reflection amplitude and transmission amplitude are related as shown in (Equation 3.1).

Thus, with this algorithm we aim to fill the three-dimensional space of reflection amplitude, reflection phase and transmission phase.

The algorithm used in this work is based on the non-adaptive expansion algorithm (Algorithm 4), and is called the acoustic domain space expansion algorithm. It is summarized as Algorithm 7.

Algorithm 7 : Acoustic domain space expansion

Input $\mathcal{G}, \mathcal{X}, tol$
$i \leftarrow I$
Compute $\boldsymbol{R} = \{r_1, \cdots, r_I\}$ {Space spanned by each acoustic output}
Compute $\boldsymbol{D} = \{d_1, \cdots, d_I\}$ {Distance to the nearest neighbor of each acoustic output}

$t \leftarrow$ mean($\boldsymbol{D}$)
$Gap_{max} \leftarrow \max(\boldsymbol{D})$
while $Gap_{max} > tol$ **do**
    per $\leftarrow \arg\max_i r_i$
    Perturb $\boldsymbol{g}_{per}$ to generate $\boldsymbol{g}_{new}$
    if $\boldsymbol{g}_{new}$ is redundant, goto (9)
    $\boldsymbol{x}_{new} \leftarrow$ COMSOL_simulation($\boldsymbol{g}_{new}$)
    Compute d_{min} {Distance to the nearest neighbor of $\boldsymbol{x}_{new}$}
    if $d_{min} > t$ **then**
        $\mathcal{G} \leftarrow$ append($\mathcal{G}, \boldsymbol{g}_{new}$)
        $\mathcal{X} \leftarrow$ append($\mathcal{X}, \boldsymbol{x}_{new}$)
        $i \leftarrow i + 1$
        Update $\boldsymbol{R}, \boldsymbol{D}$
        $Gap_{max} \leftarrow \max(\boldsymbol{D})$
    end if
    $t \leftarrow 0.5(t + d_{min})$
end while
Output $\mathcal{G}, \mathcal{X}$

3.6 Results

We used the acoustic domain space expansion algorithm to simulate geometries for higher grid-sizes such as 5×5, 6×6, and 7×7. Figure 3.12 shows the scatter plot of the acoustic outputs for the 7×7 grid size. With this grid size, the acoustics outputs obtained span the entire desired domain space of the acoustic outputs, with a tolerance of 5.4%. Thus, the

7×7 grid-size is the optimal grid size, which addresses our first objective. As reflection amplitude is related to the transmission amplitude by the conservation of energy principle, it is impossible to control it independently with transmission amplitude, and thus we have ignored it in our method.

Note that our algorithm simulated only $24,000$ geometries, which is only $4.2 \times 10^{-9}\%$ of the 563 trillion possible geometries for the 7×7 grid-size. Thus, our algorithm addressed the problem of infeasibilty of simulating geometries of higher grid-sizes.

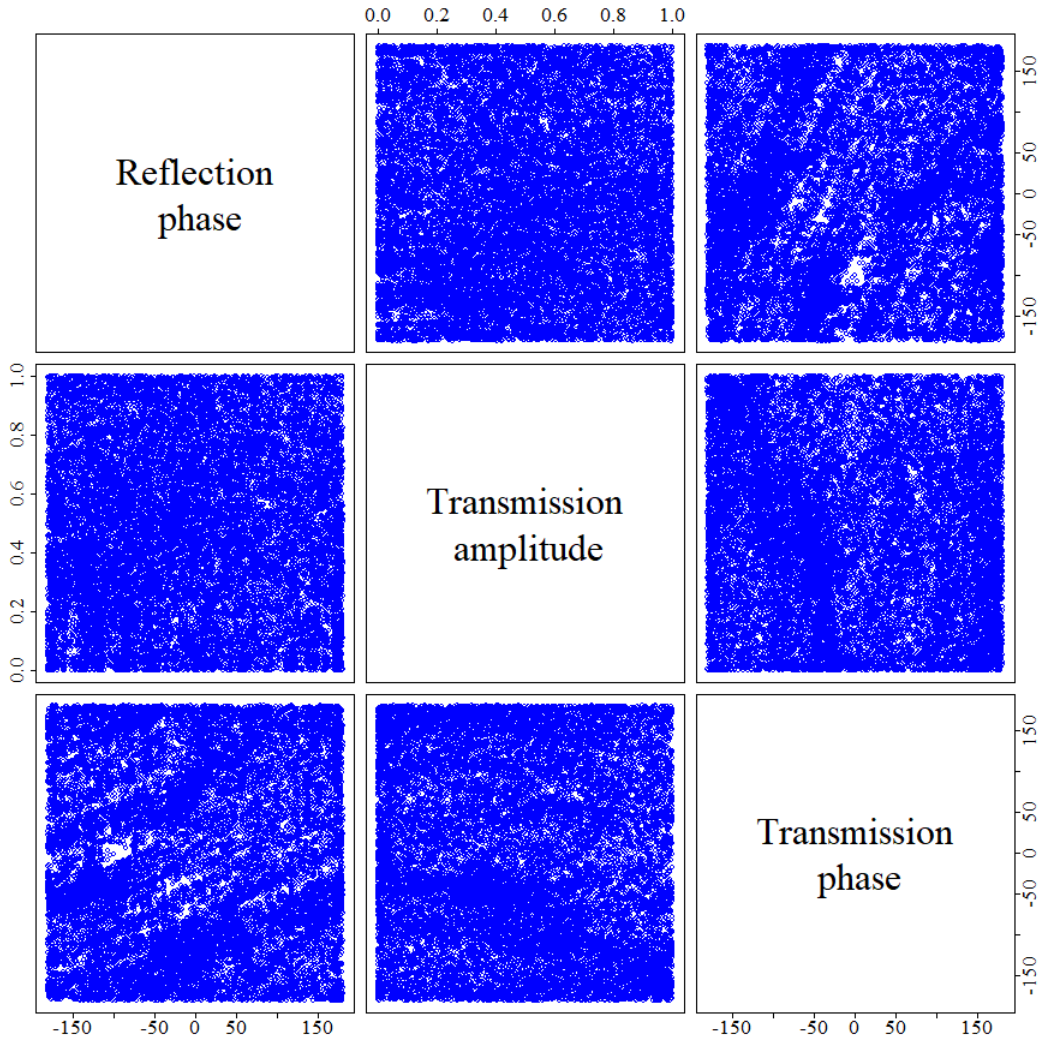

Figure 3.12: Scatter plot matrix of output data from the simulation (with lossless acoustic elements) of a 7×7 grid.

With our algorithm, we have collected the simulation data for the 7×7 grid-size, i.e., the set of geometries and the corresponding acoustic outputs. Given a desired output, we will pick out a geometry that provides the output within 5.4% error. Note that 5.4% will be the maximum error among the errors for each of the individual four acoustic outputs. Thus, using the geometries selected by our algorithm, we will be able to independently modulate the phase and amplitude of the reflected and transmitted acoustic waves. This addresses our second objective.

3.7 Conclusion

In this work, we have developed a novel acoustic expansion algorithm to find the acoustic metasurface geometries that efficiently fill the domain space of the set of desired acoustic outputs - namely the reflection amplitude, reflection phase, transmission amplitude and transmission phase. The key feature of our algorithm is that it simulates only a small fraction of the total number of possible geometries to obtain the set of relevant ones. Thus, it addresses the infeasibility issue of simulating all possible geometries, and provides a small candidate set of geometries to independently modulate the phase and amplitudes of the reflected and transmitted acosutic waves.

Our approach has several advantages over the traditional machine learning algorithm. First, the traditional algorithms will work well only within the domain space of the data used to train them. They often fail to predict the geometry that will provide them the acoustic outputs not observed in the data. Second, the traditional algorithms will depend on the grid-size. A set of predictors significant for a given grid-size may not be significant for another grid-size. This will lead to a different model for each grid-size, and add to complexities in identifying the optimal grid-size. Third, our approach can be easily extended to modulate additional acoustic outputs, such as the backward reflection phase. It will only add a dimension to the domain space to be filled. However, the traditional machine learning algorithm may require separate training on the data corresponding to the new acoustic

output.

In future work, we will extend our acoustic domain space expansion algorithm to modulate additional acoustic outputs independently. Even though the phases for forward and backward transmissions are identical (as seen in Section 3.4), the reflection phases for forward and backward propagations can be different . Our next goal will be to control the phases of the two reflections independently. This is typically important for metasurface designs because independent control of the two reflection phases can lead to larger steering angle of the acoustic beamforming and asymmetric noise cancellation.

Appendices

APPENDIX A

DESIGN OF ACOUSTIC METASURFACES WITH INDEPENDENT AMPLITUDE AND PHASE CONTROLS

Algorithm 8 can be used to obtain the vertically flipped geometry.

Algorithm 8 : Vertical flip

Input $\boldsymbol{g}, m, n$ {n: number of columns; m: number of rows}
for `i in 1:n` **do**
  for `j in 1:m` **do**
    $\boldsymbol{g}_{vert}[j][i] = \boldsymbol{g}[m+1-j][i]$
  end for
end for
Output $\boldsymbol{g}_{vert}$

Algorithm 9 can be used to identify if the geometry is redundant due to the presence of redundant solid material.

Algorithm 9 : Redundant solid material

Input $\boldsymbol{g}, m, n$ {n: number of columns; m: number of rows}
Find `empty_elements_indices` using $\boldsymbol{g}, m, n$ {indices of empty elements of the grid}
Find `wave_path` {indices of empty elements in the first column of the grid}
`current_wave_path` ← `first_col_empty`
while `current_wave_path` changes after each iteration **do**
  for `e in wave_path` **do**
    Find a_e {a_e: empty indices adjacent to `e`}
    `new_path` ← intersect(a_e,`empty_element_indices`)
    `wave_path` ← append(`wave_path`,`new_path`)
  end for
  `current_wave_path` ← `wave_path`
end while
if `wave_path` already exists in the candidate set of geometries **then**
  `redundant` ← 1
else
  `redundant` ← 0
end if
Output `redundant`

Algorithm 10 can be used to identify if the geometry is redundant due to translation of a sub-geometry.

Algorithm 10 : Translation of sub-geometry

1: Input $\boldsymbol{g}, m, n$ {n: number of columns; m: number of rows}
2: Find `empty_elements_indices` using $\boldsymbol{g}, m, n$ {indices of empty elements of the grid}
3: Find `solid_elements_indices` using `empty_elements_indices` {indices of solid elements of the grid}
4: **if** max(`solid_elements_indices`) $< mn - n$ or min(`solid_elements_indices`)$> n$ **then**
5: `Alternate_solid_elements_indices` $\leftarrow$ `solid_elements_indices` + $i \times h$, where $i \in \mathbb{Z}$ such that `Alternate_solid_elements_indices` $\in [1, mn]$
6: **end if**
7: **for** `a in alternate_solid_elements_indices` **do**
8: **if** `a` already exists in the candidate set of geometries **then**
9: `translated_subgeometry` $\leftarrow 1$
10: **else**
11: `translated_subgeometry` $\leftarrow 0$
12: **end if**
13: **end for**
14: Output `translated_subgeometry`

Algorithm 11 can be used to obtain the vertically flipped geometry along multiple horizontal axis, and Algorithm 12 can be used to obtain the horizontally flipped geometry.

Algorithm 11 : Vertical flip along multiple horizontal axis

Input $\boldsymbol{g}, m, n$ {n: number of columns; m: number of rows}
Find $\boldsymbol{g}_{vert}$ using Algorithm 8.
if $\boldsymbol{g}_{vert} = \boldsymbol{g}$ **then**
  $mc \leftarrow \texttt{ceiling}(m/2)$
  $mf \leftarrow \texttt{floor}(m/2)$
  for `i in 1:n` **do**
    for `j in 1:mf` **do**
      $\boldsymbol{g}_{vert2}[j][i] = \boldsymbol{g}[mf + 1 - j][i]$
    end for
    for `j in (mc+1):m` **do**
      $\boldsymbol{g}_{vert2}[j][i] = \boldsymbol{g}[2mc + 1 - (j - mf)][i]$
    end for
  end for
end if
Output $\boldsymbol{g}_{vert2}$

Algorithm 12 : Horizontal flip

Input $\boldsymbol{g}, m, n$ {n: number of columns; m: number of rows}
for `i in 1:m` **do**
  for `j in 1:n` **do**
    $\boldsymbol{g}_{hor}[i][j] = \boldsymbol{g}[i][m + 1 - j]$
  end for
end for
Output $\boldsymbol{g}_{hor}$

REFERENCES

[1] C. F. J. Wu and M. S. Hamada, *Experiments: Planning, Analysis, and Optimization*. John Wiley & Sons, 2011, vol. 552.

[2] T. J. Santner, B. J. Williams, and W. I. Notz, *The design and analysis of computer experiments*. Springer, 2018.

[3] G. E. P. Box and W. G. Hunter, "A useful method for model building," *Technometrics*, vol. 4, pp. 301–318, 1962.

[4] S. Silvey, *Optimal design: an introduction to the theory for parameter estimation*. Chapman & Hall, 1980, vol. 1.

[5] G. E. P. Box and N. R. Draper, "A basis for the selection of a response surface design," *Journal of the American Statistical Association*, vol. 54, no. 287, pp. 622–654, 1959.

[6] Y. J. Chang and W. I. Notz, "Model robust designs," *Handbook of statistics*, vol. 13, pp. 1055–1098, 1996.

[7] K. Chaloner and I. Verdinelli, "Bayesian experimental design: A review," *Statistical Science*, pp. 273–304, 1995.

[8] H. A. Dror and D. M. Steinberg, "Robust experimental design for multivariate generalized linear models," *Technometrics*, vol. 48, no. 4, pp. 520–529, 2006.

[9] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 3, pp. 425–464, 2001.

[10] E. R. Leatherman, A. M. Dean, and T. J. Santner, "Designing combined physical and computer experiments to maximize prediction accuracy," *Computational Statistics & Data Analysis*, vol. 113, pp. 346–362, 2017.

[11] R. Tuo and C. F. J. Wu, "Efficient calibration for imperfect computer models," *The Annals of Statistics*, vol. 43, no. 6, pp. 2331–2352, 2015.

[12] M. Plumlee, "Bayesian calibration of inexact computer models," *Journal of the American Statistical Association*, vol. 112, no. 519, pp. 1274–1285, 2017.

[13] P. D. Arendt, D. W. Apley, and W. Chen, "A preposterior analysis to predict identifiability in the experimental calibration of computer models," *IIE Transactions*, vol. 48, no. 1, pp. 75–88, 2016.

[14] P. Ranjan, W. Lu, D. Bingham, S. Reese, B. J. Williams, C.-C. Chou, F. Doss, M. Grosskopf, and J. P. Holloway, "Follow-up experimental designs for computer models and physical processes," *Journal of Statistical Theory and Practice*, vol. 5, no. 1, pp. 119–136, 2011.

[15] B. J. Williams, J. L. Loeppky, L. M. Moore, and M. S. Macklem, "Batch sequential design to achieve predictive maturity with calibrated computer models," *Reliability Engineering & System Safety*, vol. 96, no. 9, pp. 1208–1219, 2011.

[16] V. R. Joseph and S. N. Melkote, "Statistical adjustments to engineering models," *Journal of Quality Technology*, vol. 41, no. 4, pp. 362–375, 2009.

[17] E. Masoudi, H. Holling, B. P. Duarte, and W. K. Wong, "A metaheuristic adaptive cubature based algorithm to find bayesian optimal designs for nonlinear models," *Journal of Computational and Graphical Statistics*, vol. 28, no. 4, pp. 861–876, 2019.

[18] J. Kiefer, *Collected Papers III: Design of Experiments*. Springer, 1985.

[19] E. Masoudi, H. Holling, W. K. Wong, and S. Kim, "Icaod: An r package for finding optimal designs for nonlinear models using imperialist competitive algorithm. R package version 1.0.1," *URL: https://cran.r-project.org/web/packages/ICAOD*, 2020.

[20] F. Pukelsheim and S. Rieder, "Efficient rounding of approximate designs," *Biometrika*, vol. 79, no. 4, pp. 763–770, 1992.

[21] C. Huang, V. R. Joseph, and S. Mak, "Population quasi-monte carlo," *arXiv preprint arXiv:2012.13769*, 2020.

[22] V. R. Joseph, L. Gu, S. Ba, and W. R. Myers, "Space-filling designs for robustness experiments," *Technometrics*, vol. 61, pp. 24–37, 2019.

[23] 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.

[24] V. R. Joseph, "Space-filling designs for computer experiments: A review," *Quality Engineering*, vol. 28, no. 1, pp. 28–35, 2016.

[25] V. R. Joseph, E. Gul, and S. Ba, "Maximum projection designs for computer experiments," *Biometrika*, vol. 102, no. 2, pp. 371–380, 2015.

[26] S. Ba and V. R. Joseph, "Maxpro: Maximum projection designs. R package version 4.1-2," *URL: https://cran.r-project.org/web/packages/MaxPro*, 2018.

[27] V. R. Joseph, E. Gul, and S. Ba, "Designing computer experiments with multiple types of factors: The maxpro approach," *Journal of Quality Technology*, pp. 1–12, 2019.

[28] S. Mak and V. R. Joseph, "Support points," *The Annals of Statistics*, vol. 46, no. 6A, pp. 2562–2592, 2018.

[29] S. Mak, "Support points. R package version 0.1.4," *URL: https://cran.r-project.org/src/contrib/Arch* 2019.

[30] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Statistical science*, pp. 409–423, 1989.

[31] P. Z. Qian, "Nested latin hypercube designs," *Biometrika*, vol. 96, no. 4, pp. 957–970, 2009.

[32] D. Christophe and S. Petr, "Randtoolbox: Generating and testing random numbers. R package version 1.30.0," *URL: https://cran.r-project.org/web/packages/randtoolbox*, 2019.

[33] A. Franceschetti and A. Zunger, "The inverse band-structure problem of finding an atomic configuration with given electronic properties," *Nature*, vol. 402, no. 6757, pp. 60–63, 1999.

[34] T. Weymuth and M. Reiher, "Inverse quantum chemistry: Concepts and strategies for rational compound design," *International Journal of Quantum Chemistry*, vol. 114, no. 13, pp. 823–837, 2014.

[35] M. d'Avezac, J.-W. Luo, T. Chanier, and A. Zunger, "Genetic-algorithm discovery of a direct-gap and optically allowed superstructure from indirect-gap si and ge semiconductors," *Phys. Rev. Lett.*, vol. 108, no. 2, p. 027 401, 2012.

[36] H. Xiang, B. Huang, E. Kan, S.-H. Wei, and X. Gong, "Towards direct-gap silicon phases by the inverse band structure design approach," *Phys. Rev. Lett.*, vol. 110, no. 11, p. 118 702, 2013.

[37] T. D. Huan, A. Mannodi-Kanakkithodi, and R. Ramprasad, "Accelerated materials property predictions and design using motif-based fingerprints," *Phys. Rev. B*, vol. 92, no. 1, p. 014 106, 2015.

[38] A. Mannodi-Kanakkithodi, G. Pilania, T. D. Huan, T. Lookman, and R. Ramprasad, "Machine learning strategy for the accelerated design of polymer dielectrics," *Sci. Rep.*, vol. 6, p. 20 952, 2016.

[39] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," *Phys. Rev.*, vol. 136, B864–B871, 1964.

[40] W. Kohn and L. Sham, "Self-consistent equations including exchange and correlation effects," *Phys. Rev.*, vol. 140, A1133–A1138, 1965.

[41] L. Pauling, "The principles determining the structure of complex ionic crystals," *Journal of the american chemical society*, vol. 51, no. 4, pp. 1010–1026, 1929.

[42] A. Kitaigorodskii, "Organic crystal chemistry [in russian], izd," *An SSSR, Moscow*, p. 15, 1955.

[43] V. Blatov, L. Carlucci, G. Ciani, and D. Proserpio, "Interpenetrating metal–organic and inorganic 3d networks: A computer-aided systematic investigation. part i. analysis of the cambridge structural database," *CrystEngComm*, vol. 6, no. 65, pp. 377–395, 2004.

[44] I. A. Baburin and V. A. Blatov, "Three-dimensional hydrogen-bonded frameworks in organic crystals: A topological study," *Acta Crystallographica Section B: Structural Science*, vol. 63, no. 5, pp. 791–802, 2007.

[45] V. A. Blatov and D. M. Proserpio, "Topological relations between three-periodic nets. ii. binodal nets," *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 65, no. 3, pp. 202–212, 2009.

[46] S. Curtarolo, D. Morgan, K. Persson, J. Rodgers, and G. Ceder, "Predicting crystal structures with data mining of quantum calculations," *Phys. Rev. Lett.*, vol. 91, no. 13, p. 135 503, 2003.

[47] A. R. Oganov, *Modern methods of crystal structure prediction*. John Wiley & Sons, 2011.

[48] A. R. Oganov and C. W. Glass, "Crystal structure prediction using ab initio evolutionary techniques: Principles and applications," *The Journal of chemical physics*, vol. 124, no. 24, p. 244 704, 2006.

[49] W. W. Tipton and R. G. Hennig, "Random search methods," *Modern Methods of Crystal Structure Prediction*, pp. 55–66, 2010.

[50] R. S. Berry, "Potential surfaces and dynamics: What clusters tell us," *Chemical reviews*, vol. 93, no. 7, pp. 2379–2394, 1993.

[51] F. H. Stillinger, "Exponential multiplicity of inherent structures," *Physical Review E*, vol. 59, no. 1, p. 48, 1999.

[52] J. Pannetier, J. Bassas-Alsina, J. Rodriguez-Carvajal, and V. Caignaert, "Prediction of crystal structures from crystal chemistry rules by simulated annealing," *Nature*, vol. 346, no. 6282, pp. 343–345, 1990.

[53] J. C. Schön and M. Jansen, "First step towards planning of syntheses in solid-state chemistry: Determination of promising structure candidates by global optimization," *Angewandte Chemie International Edition in English*, vol. 35, no. 12, pp. 1286–1304, 1996.

[54] A. Tekin, R. Caputo, and A. Züttel, "First-principles determination of the ground-state structure of libh 4," *Phys. Rev. Lett.*, vol. 104, no. 21, p. 215 501, 2010.

[55] D. J. Wales and J. P. Doye, "Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms," *The Journal of Physical Chemistry A*, vol. 101, no. 28, pp. 5111–5116, 1997.

[56] S. Goedecker, "Minima hopping: An efficient search method for the global minimum of the potential energy surface of complex molecular systems," *J. Chem. Phys.*, vol. 120, no. 21, pp. 9911–9917, 2004.

[57] R. Martoňák, D. Donadio, A. R. Oganov, and M. Parrinello, "Crystal structure transformations in sio 2 from classical and ab initio metadynamics," *Nature materials*, vol. 5, no. 8, pp. 623–626, 2006.

[58] G. Trimarchi, A. J. Freeman, and A. Zunger, "Predicting stable stoichiometries of compounds via evolutionary global space-group optimization," *Phys. Rev. B*, vol. 80, no. 9, p. 092 101, 2009.

[59] T. D. Huan, "Pressure-stabilized binary compounds of magnesium and silicon," *Phys. Rev. Materials*, vol. 2, no. 2, p. 023 803, 2018.

[60] Y. Kobayashi, M. Naito, K. Sudoh, A. Gentils, C. Bachelet, and J. Bourçois, "Formation of crystallographically oriented metastable $Mg1.8Si$ in Mg ion-implanted Si," *Crystal Growth & Design*, vol. 19, no. 12, pp. 7138–7142, 2019.

[61] N. A. Gaida, K. Niwa, T. Sasaki, and M. Hasegawa, "Phase relations and thermoelasticity of magnesium silicide at high pressure and temperature," *J. Chem. Phys.*, vol. 154, no. 14, p. 144 701, 2021.

[62] F. P. Bundy, "The p, t phase and reaction diagram for elemental carbon, 1979," *J. Geophys. Res. Solid Earth*, vol. 85, no. B12, pp. 6930–6936, 1980.

[63] A. R. Oganov, C. J. Pickard, Q. Zhu, and R. J. Needs, "Structure prediction drives materials discovery," *Nat. Rev. Mater.*, vol. 4, no. 5, pp. 331–348, 2019.

[64] T. N. Vu, S. K. Nayak, N. T. T. Nguyen, S. P. Alpay, and H. Tran, "Atomic configurations for materials research: A case study of some simple binary compounds," *AIP Adv.*, vol. 11, no. 4, p. 045 120, 2021.

[65] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[66] R. Batra, H. D. Tran, C. Kim, J. Chapman, L. Chen, A. Chandrasekaran, and R. Ramprasad, "General atomic neighborhood fingerprint for machine learning-based methods," *J. Phys. Chem. C*, vol. 123, no. 25, pp. 15 859–15 866, 2019.

[67] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," *Phys. Rev. Lett.*, vol. 98, no. 14, p. 146 401, 2007.

[68] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," *Phys. Rev. Lett.*, vol. 104, no. 13, p. 136 403, 2010.

[69] X. Gonze, F. Jollet, F. A. Araujo, D. Adams, B. Amadon, T. Applencourt, C. Audouze, J.-M. Beuken, J. Bieder, A. Bokhanchuk, E. Bousquet, F. Bruneval, D. Caliste, M. Côté, F. Dahm, F. D. Pieve, M. Delaveau, M. D. Gennaro, B. Dorado, C. Espejo, G. Geneste, L. Genovese, A. Gerossier, M. Giantomassi, Y. Gillet, D. Hamann, L. He, G. Jomard, J. L. Janssen, S. L. Roux, A. Levitt, A. Lherbier, F. Liu, I. Lukačević, A. Martin, C. Martins, M. Oliveira, S. Poncé, Y. Pouillon, T. Rangel, G.-M. Rignanese, A. Romero, B. Rousseau, O. Rubel, A. Shukri, M. Stankovski, M. Torrent, M. V. Setten, B. V. Troeye, M. Verstraete, D. Waroquiers, J. Wiktor, B. Xu, A. Zhou, and J. Zwanziger, "Recent developments in the abinit software package," *Comput. Phys. Commun.*, vol. 205, pp. 106–131, 2016.

[70] J. P. Perdew, K. Burke, and M. Ernzerhof, "Generalized gradient approximation made simple," *Phys. Rev. Lett.*, vol. 77, pp. 3865–3868, 1996.

[71] C. Hartwigsen, S. Goedecker, and J. Hutter, "Relativistic separable dual-space gaussian pseudopotentials from H to Rn," *Phys. Rev. B*, vol. 58, p. 3641, 1998.

[72] H. J. Monkhorst and J. D. Pack, "Special points for brillouin-zone integrations," *Phys. Rev. B*, vol. 13, p. 5188, 1976.

[73] J. Maddox, "Crystals from first principles," *Nature*, vol. 335, no. 6187, pp. 201–201, 1988.

[74] O. Roustant, D. Ginsbourger, and Y. Deville, "Dicekriging, diceoptim: Two r packages for the analysis of computer experiments by kriging-based metamodeling and optimization," 2012.

[75] V. Nguyen, S. Gupta, S. Rana, C. Li, and S. Venkatesh, "Regret for expected improvement over the best-observed value and stopping condition," in *Asian Conference on Machine Learning*, PMLR, 2017, pp. 279–294.

[76] J. E. Jackson, *A user's guide to principal components*. John Wiley & Sons, 2005, vol. 587.

[77] C. Qin, D. Klabjan, and D. Russo, "Improving the expected improvement algorithm," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc., 2017.

[78] M. Schonlau, W. J. Welch, and D. R. Jones, "Global versus local search in constrained optimization of computer models," *Lecture Notes-Monograph Series*, pp. 11–25, 1998.

[79] S. A. Cummer, J. Christensen, and A. Alù, "Controlling sound with acoustic metamaterials," *Nature Reviews Materials*, vol. 1, no. 3, pp. 1–13, 2016.

[80] Y. Xie, W. Wang, H. Chen, A. Konneker, B.-I. Popa, and S. A. Cummer, "Wavefront modulation and subwavelength diffractive acoustics with an acoustic metasurface," *Nature communications*, vol. 5, no. 1, pp. 1–5, 2014.

[81] Y. Li, X. Jiang, R.-q. Li, B. Liang, X.-y. Zou, L.-l. Yin, and J.-c. Cheng, "Experimental realization of full control of reflected waves with subwavelength acoustic metasurfaces," *Physical Review Applied*, vol. 2, no. 6, p. 064 002, 2014.

[82] G. Ma, M. Yang, S. Xiao, Z. Yang, and P. Sheng, "Acoustic metasurface with hybrid resonances," *Nature materials*, vol. 13, no. 9, pp. 873–878, 2014.

[83] Y. Li and B. M. Assouar, "Acoustic metasurface-based perfect absorber with deep subwavelength thickness," *Applied Physics Letters*, vol. 108, no. 6, p. 063 502, 2016.

[84] C. Faure, O. Richoux, S. Félix, and V. Pagneux, "Experiments on metasurface carpet cloaking for audible acoustics," *Applied Physics Letters*, vol. 108, no. 6, p. 064 103, 2016.

[85] M. Dubois, C. Shi, Y. Wang, and X. Zhang, "A thin and conformal metasurface for illusion acoustics of rapidly changing profiles," *Applied Physics Letters*, vol. 110, no. 15, p. 151 902, 2017.

[86] K. Melde, A. G. Mark, T. Qiu, and P. Fischer, "Holograms for acoustics," *Nature*, vol. 537, no. 7621, pp. 518–522, 2016.

[87] C. Shi, M. Dubois, Y. Wang, and X. Zhang, "High-speed acoustic communication by multiplexing orbital angular momentum," *Proceedings of the National Academy of Sciences*, vol. 114, no. 28, pp. 7250–7253, 2017.

[88] X. Jiang, B. Liang, J.-C. Cheng, and C.-W. Qiu, "Twisted acoustics: Metasurface-enabled multiplexing and demultiplexing," *Advanced Materials*, vol. 30, no. 18, p. 1 800 257, 2018.

[89] A. Marzo, S. A. Seah, B. W. Drinkwater, D. R. Sahoo, B. Long, and S. Subramanian, "Holographic acoustic elements for manipulation of levitated objects," *Nature communications*, vol. 6, no. 1, pp. 1–7, 2015.

[90] Y. Zhu, J. Hu, X. Fan, J. Yang, B. Liang, X. Zhu, and J. Cheng, "Fine manipulation of sound via lossy metamaterials with independent and arbitrary reflection amplitude and phase," *Nature communications*, vol. 9, no. 1, pp. 1–9, 2018.

[91] Y. Li, B. Liang, Z.-m. Gu, X.-y. Zou, and J.-c. Cheng, "Reflected wavefront manipulation based on ultrathin planar acoustic metasurfaces," *Scientific reports*, vol. 3, no. 1, pp. 1–6, 2013.

[92] J. Zhao, B. Li, Z. Chen, and C.-W. Qiu, "Manipulating acoustic wavefront by inhomogeneous impedance and steerable extraordinary reflection," *Scientific reports*, vol. 3, no. 1, pp. 1–6, 2013.

[93] K. Tang, C. Qiu, M. Ke, J. Lu, Y. Ye, and Z. Liu, "Anomalous refraction of airborne sound through ultrathin metasurfaces," *Scientific reports*, vol. 4, no. 1, pp. 1–7, 2014.

[94] J. Mei and Y. Wu, "Controllable transmission and total reflection through an impedance-matched acoustic metasurface," *New Journal of Physics*, vol. 16, no. 12, p. 123 007, 2014.

[95] Y. Li, X. Jiang, B. Liang, J.-c. Cheng, and L. Zhang, "Metascreen-based acoustic passive phased array," *Physical Review Applied*, vol. 4, no. 2, p. 024 003, 2015.

[96] X. Zhu, K. Li, P. Zhang, J. Zhu, J. Zhang, C. Tian, and S. Liu, "Implementation of dispersion-free slow acoustic wave propagation and phase engineering with helical-structured metamaterials," *Nature Communications*, vol. 7, no. 1, pp. 1–7, 2016.

[97] X. Jiang, Y. Li, B. Liang, J.-c. Cheng, and L. Zhang, "Convert acoustic resonances to orbital angular momentum," *Physical review letters*, vol. 117, no. 3, p. 034 301, 2016.