

Inverse Modeling through Machine Learning: Optimizing Microstructure Properties in Multi-Phase Ceramics

**Inverse Modellierung durch maschinelles Lernen: Optimierung
der Mikrostruktureigenschaften in mehrphasiger Keramik**

Master Thesis

at

Universität Bayreuth

Institut für Informatik

Lehrstuhl für Angewandte Informatik VIII

Rayan Hamid Mohiuddin

Matrikelnummer: 1762370

January 15, 2024

Erstprüfer

Dr. Simon Pirkelmann

Zweitprüfer

Prof. Dr. Jörg Müller

Dritter Prüfer

Dr. Arthur Fleig

Abstract

This thesis tackles inverse modeling for multi-phase ceramics, using Zirconia Toughened Alumina (ZTA) and Alumina Toughened Zirconia (ATZ) as examples, where the goal is to identify how the microstructures should be designed to achieve a desired material property. This is formulated as an optimization problem by minimizing the difference between the forward function and the desired property. FEM simulations are typically used as the forward function, which is usually time-consuming and computationally expensive. This work addresses this by replacing FEM simulations with Gaussian Process Regressors as a surrogate function in the optimization process.

The research in this work is focused on the investigation of the robustness, efficiency, and potential of Machine Learning (ML) models, in particular, Gaussian Process Regressors (GPR) as surrogate models in inverse modeling for ceramic materials. This can be divided into two primary research tasks. The initial segment is focused on effective modeling of the forward function using GPRs and finding the combination of kernels and hyperparameters that achieve the best results in terms of accuracy. The latter segment tackles the effective formulation of the inverse modeling methodology and performs a comparative study in the context of inverse modeling with GPRs to assess the performance with different kernels and optimization techniques. The findings from both these segments are combined to identify potential trade-offs in computational speed and overall accuracy and to identify the set of modeling parameters and methods that achieve the best performance for inverse design.

This thesis also investigates the influence of adaptive sampling with online Finite Element (FE) simulations in improving GPR training and reducing the size of the training data. Finally, a comparison of the optimization-based inverse modeling approach to a direct ML model of the inverse function is shown.

In the broader context, this thesis can be seen to contribute to ongoing re-

search by providing a complete analysis of the performance of GPRs in an inverse modeling context for multi-phase ceramics. The results showcase that GPRs with a *RBF + White* kernel not only serve as efficient surrogate models for most material properties but, when complemented with the L-BFGS-B gradient-based optimizer with analytically calculated gradients, they turn into powerful and highly accurate tools for inverse design in multi-phase ceramics. Additionally, adaptive sampling with GPRs is shown to be a powerful approach to improve the generalizability of GPRs which directly improves the quality and performance of the inverse design methodology.

Zusammenfassung

Diese Arbeit befasst sich mit der inversen Modellierung für mehrphasige Keramiken am Beispiel von Zirkonoxidverstärktes Aluminiumoxid (ZTA) und Aluminiumoxidverstärktes Zirkoniumoxid (ATZ), wobei das Ziel darin besteht, herauszufinden, wie die Mikrostrukturen gestaltet sein sollten, um eine gewünschte Materialeigenschaft zu erreichen. Dies wird als Optimierungsproblem formuliert, indem die Differenz zwischen der Vorwärtsfunktion und der gewünschten Eigenschaft minimiert wird. Als Vorwärtsfunktion werden meist FEM-Simulationen verwendet, die in der Regel zeit- und rechenaufwendig sind. In dieser Arbeit wird dieses Problem angegangen, indem FEM-Simulationen durch Gaußsche Prozessregressoren als Ersatzfunktion im Optimierungsprozess ersetzt werden.

Die Forschung in dieser Arbeit konzentriert sich auf die Untersuchung der Robustheit, der Effizienz und des Potenzials von ML-Modellen, insbesondere von Gaußschen Prozessregressoren (GPR) als Ersatzmodelle bei der inversen Modellierung für keramische Werkstoffe. Dies kann in zwei primäre Forschungsaufgaben unterteilt werden. Der erste Teil der Arbeit konzentriert sich auf die effektive Modellierung der Vorwärtsfunktion unter Verwendung von GPRs und die Suche nach der Kombination von Kerneln und Hyperparametern, die die besten Ergebnisse in Bezug auf die Genauigkeit erzielen. Der zweite Teil befasst sich mit der effektiven Formulierung der inversen Modellierungsmethodik und führt eine vergleichende Studie im Rahmen der inversen Modellierung mit GPRs durch, um die Leistung mit verschiedenen Kerneln und Optimierungstechniken zu bewerten. Die Ergebnisse dieser beiden Abschnitte werden kombiniert, um potenzielle Kompromisse bei der Rechengeschwindigkeit und der Gesamtgenauigkeit zu ermitteln und die Modellierungsparameter und -methoden zu bestimmen, mit denen der effiziente inverse Entwurf von mehrphasigen Keramiken ermöglicht wird.

In dieser Arbeit wird auch der Einfluss des adaptiven Samplings mit Online-Finite-Elemente-Simulationen auf die Verbesserung des GPR-Trainings und

die Verringerung des Umfangs der Trainingsdaten untersucht. Schließlich wird ein Vergleich des optimierungsisierten inversen Modellierungsansatzes mit einem direkten ML-Modell der inversen Funktion gezeigt.

Im weiteren Kontext kann diese Arbeit als Beitrag zur laufenden Forschung gesehen werden, indem sie eine vollständige Analyse der Leistung von GPRs in einem inversen Modellierungskontext für mehrphasige Keramiken liefert. Die Ergebnisse zeigen, dass GPRs mit einem *RBF + White* Kernel als effiziente Ersatzmodelle für die relevantesten keramischen Materialeigenschaften dienen. Daraüber hinaus können sie mit der gradientenbasierten Optimierungsroutine L-BFGS-B mit analytisch berechneten Gradienten ergänzt werden, wodurch sie zu leistungsfähigen und hochgenauen Werkzeugen für die inverse Konstruktion von Mehrphasenkeramiken werden. Darauber hinaus hat sich gezeigt, dass das adaptive Sampling mit GPRs ein leistungsfähiger Ansatz zur Verbesserung der Verallgemeinerbarkeit von GPRs ist, der die Qualität und Leistung der inversen Entwurfsmethodik direkt verbessert.

Contents

1	Introduction	1
2	Background	4
2.1	Inverse Modeling	4
2.2	Gaussian Process Regressors	5
2.2.1	Mean and Variance Functions	6
2.2.2	Kernel Functions	8
3	Related Work	11
3.1	Inverse Design with Bayesian Optimization	11
3.2	Gaussian Models in Material and Electromagnetic Design	13
3.3	Other Machine Learning Approaches in Foward Modeling and Inverse Design of Microstructures	16
3.4	ML-based Surrogates for Inverse Modeling in other Domains .	18
4	Methodology	22
4.1	Data Preparation	23
4.2	Data Preprocessing	26
4.3	Data Exploration	28
4.4	Surrogate Modeling with GPR	34
4.5	Adaptive GPRs	36
4.6	Inverse Modeling	40
4.7	Direct Inverse Training	43
5	Results and Discussion	45
5.1	GPR Training	45
5.2	Inverse Modeling	53
5.2.1	Initial Results with Gradient-Based Optimizer: A case study	54
5.2.2	Gradient Based Optimizers vs. Gradient Free Optimizers	63

5.2.3	Comparison to Baseline GBR	64
5.2.4	Comparison of Kernels	66
5.3	Direct Inverse Training	70
5.4	Adaptive GPRs	75
6	Conclusions and Future Work	86
References		89

List of Figures

1.1	Framework for automated microstructure-property simulations [8].	2
2.1	Example of GPR modeling possibly infinite posterior functions that fit the observed data points and the mean function obtained from the probability distribution of these functions as the red line [10].	6
3.1	Comparison of optimization methods for solving the inverse problem in scatterometric data [12].	12
3.2	Comparison of DIM with GA optimization [14].	14
3.3	fGP predictions compared with CPFE simulated data for 3 RVEs not used in the training data [16].	15
3.4	ML-based preprocessing framework: A data preparation step (1) followed by a search path refinement (2) and a search region reduction (3) to concentrate the search force [18].	17
3.5	Compliance tensors of the optimal structures plotted against the desired values [19].	18
3.6	Cauchy stress with estimated parameters along fibre (a) and sheet (b) directions and residual Cauchy stress along fibre (c) and sheet (d) directions for different approaches [20].	19
3.7	Region of viable solutions predicted by the LSTM-based ML framework compared with the true response predicted by FE for part temperature histories in the inverse heat problem [21].	20
4.1	(a) RVE generated by GeoVal for a two-phase ceramic composite; (b) Tetrahedral mesh generated with VoxSM; (c) FE result example for thermal conductivity simulations (d) Homogenization technique using a larger cell composed of multiple small RVEs [8].	23

4.2	A scanning electron microscopy (SEM) image of a ZTA microstructure (dark gray = Alumina, light gray = Zirconia) [8].	25
4.3	Parallel plot of the Volume fractions and Material Properties.	31
4.4	Volume Fraction (Zirconia) vs Material Properties with the effect of Porosity using the full feature set.	32
4.5	Volume Fraction (Zirconia) vs Material Properties with the effect of particle size ratio using the 2-feature set.	33
4.6	Distribution of microstructures in the 2-feature design space. .	34
4.7	Local error estimate density for a parameter space during adaptive training [5].	39
5.1	R^2 scores for each combination of the grid-search.	48
5.2	Grid Search results for Young's Modulus without normalization. .	50
5.3	R^2 scores of the different feature sets.	51
5.4	Visualisation of optimized microstructures for CTE and Poisson Ratio.	56
5.5	Visualisation of optimized microstructures for CTC and Young's Modulus.	58
5.6	Objective function for CTC; red points indicate local minima found through multi-starts.	59
5.7	Visualisation of optimized microstructures for CTC and Young's Modulus using the modified objective function (Equation 4.12). .	61
5.8	Visualisation of optimized microstructures for CTE and Poisson Ratio using the modified objective function (Equation 4.12). .	62
5.9	Fit of the property values of the optimized structures and the initially desired structures with different surrogate models. .	65
5.10	Inverse design using GBR.	66
5.11	Visualisation of predicted microstructures for material properties with DIT using NNs, GPR and RF, shaded region shows the prediction plane of a forward GBR surrogate.	74
5.12	Evolution of the R^2 accuracy as training samples are added with adaptive GP for each material property.	77
5.13	Evolution of Volume Fraction (Zirconia) vs. Material Property for fixed particle size ratio = 1 with adaptive approach.	78
5.14	Design space at different iterations in the adaptive GPR training for CTC.	79
5.15	Global reconstruction error per iteration with adaptive GP. .	82
5.16	Visualisation of optimized microstructures for CTC with the GP trained with adaptive approach, shaded region shows the GP prediction plane \bar{f}_*	83

List of Tables

5.1	Comparison of GBR and GPR Performance Metrics for 8-feature set	47
5.2	Comparison of GBR and GPR Performance Metrics for 2-feature set	53
5.3	Inverse Design Performance Metrics with GPR surrogate model.	54
5.4	Performance Metrics of different optimization techniques. Metrics include RR^2 , RRMSE, and Computational Time in seconds (C.Time).	63
5.5	Inverse Design Performance Metrics with GBR surrogate model.	64
5.6	Performance metrics of different kernels in inverse design using analytical and approximated gradients.	67
5.7	Performance metrics of DIT for different properties with Neural Networks (NN), GP, and Random Forests (RF).	72
5.8	Inverse Design Performance Metrics with Adaptive GP.	85

Chapter 1

Introduction

Inverse modeling has historically leaned on computational simulations and heuristic methods. However, the recent upsurge in machine learning (ML) technologies offers a fresh perspective on this age-old challenge. ML techniques possess the ability to approximate complex functions and intricate data relationships and the resulting models output instantaneously in real-time. Leveraging this capability, there's a growing interest in exploring how we can extend inverse design in material science with well-known ML techniques from computer science.

There has been rapid growth and development of inverse design in recent years. This is true for plastics [1], and metals and alloys [2]. Neural networks have shown great promise for modeling the relationship between structural descriptors and material properties [3]. However, neural networks typically require a lot of training data to be effective. Availability of data is always a constraint when simulations are computationally intensive and time-consuming. Gaussian Process Regressors show great promise in this regard with being able to closely approximate forward models with a smaller amount of data [4]. Another added advantage of GPRs is that they make an adaptive sampling approach for the training data feasible, where the number and position of sample points is determined adaptively and evaluated with online simulations [5], which can reduce the amount of training data required and improve the generalizability of the model.

When it comes to the inverse design of multi-phase ceramics, the level and depth of research in this sub-field is comparatively limited. While there exist some studies into the forward design of ceramics with various simulation methodologies [6, 7], studies into a holistic approach for the inverse problem

are still conspicuously missing. This thesis focuses on the example of a two-phase Zirconia/Alumina composite. This is either Zirconia-Toughened Alumina (ZTA) or Alumina-Toughed Zirconia (ATZ) depending on which phase is the primary phase with the higher volume fraction. Before an inverse design can be optimized, a top-down design model is required as a surrogate function. Often, analytical models such as rule-of-mixtures or Ondracek's model are used in a multi-phase ceramic design problem. However, these are not suitable for practical applications where the porosity, distribution of the size of the individual grains, and the interfaces between the phases play a role in the material property.

This thesis builds upon the foundation laid down by Pirkelmann et al. [8], where a methodology for the forward design of multi-phase ceramics using Finite element (FE) simulations was developed. They developed an automated simulation toolchain to understand the relationships between the microstructure of the ceramics and the occurring material properties. This toolchain is then able to generate a database of microstructures and their associated material and thermal properties. A forward model using machine learning methods can then be trained effectively to skip the expensive simulation steps as demonstrated in Figure 1.1.

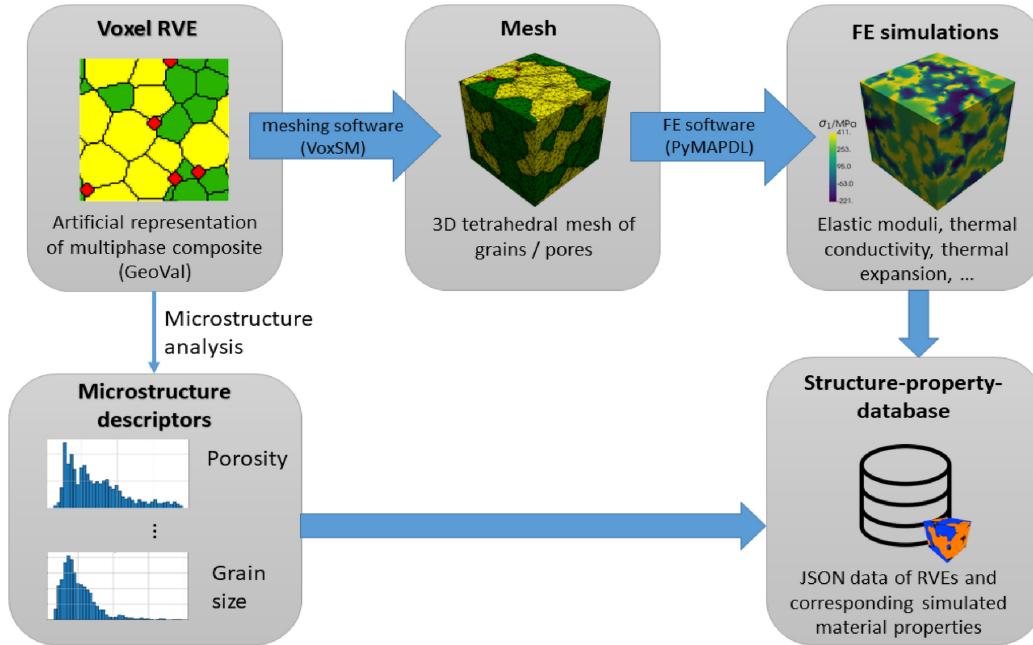


Figure 1.1: Framework for automated microstructure-property simulations [8].

Building on this foundational work, this research innovatively extends the framework to address the challenges of top-down design in ceramics. This is typically achieved by solving a minimization problem for an objective function based on the difference between the target property and the forward surrogate function (predicted property). In the case of gradient-based optimizers, this also requires that the surrogate function be differentiable at all points. It is shown that certain ML methods such as Gradient Boosting Regressors (originally used in the work by Pirkelmann et al. [8]) model the function in a staggered manner, leading to gradients of zero at various points. This, of course, poses challenges with gradient-based optimizers. To address this, Gaussian Process Regressors are employed, which can model the relationships with a smooth function. The added advantage of using such a model is that the gradients can be mathematically derived and calculated analytically, significantly speeding up the optimization routine.

Given the promising potential of GPRs in the inverse design of ceramics, this thesis seeks to address the following research questions:

- What are the performance characteristics and limitations of standard Gaussian Process Regression techniques in the context of inverse design of ceramics? How do different kernel functions and hyperparameter settings impact the accuracy and robustness of the forward models?
- What are the benefits of using adaptive sampling strategies in improving GPR training for ceramic design?
- How does the choice of kernels and gradient evaluations affect the performance of the inverse design process?
- How does this inverse design methodology compare to training ML models directly in the reverse direction?

Chapter 2

Background

2.1 Inverse Modeling

"Inverse Modeling", also known as Inverse Design or Parameter Reconstruction, refers to the process of finding optimal parameters that would result in a desired outcome or a material with a desired property. Instead of the traditional forward modeling that asks "Given X, what is Y?", inverse design asks, "I want Y to be a specific value, so what should X be?" Typically, this is done by solving an optimization problem as,

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} \quad J(\mathbf{x}) \\ & \text{where} \quad J(\mathbf{x}) = \|f(\mathbf{x}) - y^*\| \end{aligned} \tag{2.1}$$

where $J(\mathbf{x})$ is the objective function, between $f(\mathbf{x})$ is the forward function $f : X \rightarrow Y$, where X represents the input design and Y represents the output property, and y^* is the desired output property. Therefore the objective function represents the discrepancy between the surrogate function that predicts outcomes based on input x and the desired outcome. In the context of multi-phase ceramics, the input designs (X) correspond to the microstructure descriptors, such as volume fractions, particle sizes, etc., of different phases, and the output properties (Y) correspond to a material property, such as Thermal Expansion, Thermal Conductivity, Young's Modulus and Poisson Ratio.

Traditionally, $f(\mathbf{x})$ is evaluated using computationally demanding simulations such as the Finite Element Method (FEM) which are usually complex mathematical procedures. For inverse design, this requires evaluating the

function at several points in the input space, making the entire process a significantly expensive task. By replacing the computationally expensive simulations with a surrogate function based on an ML model such as a GPR, $f(\mathbf{x})$ can be closely approximated and evaluated quickly in real-time, and the entire task can significantly speed up.

The minimization problem can be solved using various standardized optimization techniques, which can be gradient-free or gradient-based routines. Gradient-based optimizers require the gradient (or the first derivative) of the objective function $J(\mathbf{x})$ to guide the search for the minima. They are typically faster and more efficient when the gradient information is available and the problem is smooth and differentiable when compared to gradient-free techniques. For the objective function $J(\mathbf{x})$ to be differentiable, the surrogate function $f(\mathbf{x})$, or the GPR prediction function in the case of this thesis, needs to be differentiable. This is explored in Section 2.2.

2.2 Gaussian Process Regressors

Gaussian Process Regression (GPR), often simply called Gaussian Processes (GP), is a powerful non-parametric method used for regression, classification, and other tasks [9]. GPR is based on the Bayesian inference principle and provides not just the prediction function (as in standard regression methods) but also a measure of uncertainty, which is helpful with inverse design if a choice has to be made between multiple solutions [10]. Most importantly, for inverse design, the prediction function should be differentiable and easy to compute to utilize gradient-based optimization techniques efficiently.

The goal of regression models typically is to find a function that fits the given data. However, in general, it is possible to find multiple functions that all fit the same data. A GPR, based on this idea, is defined as a probability distribution over all such functions, any finite samples of which have a joint Gaussian distribution [11]. This is illustrated in Figure 2.1.

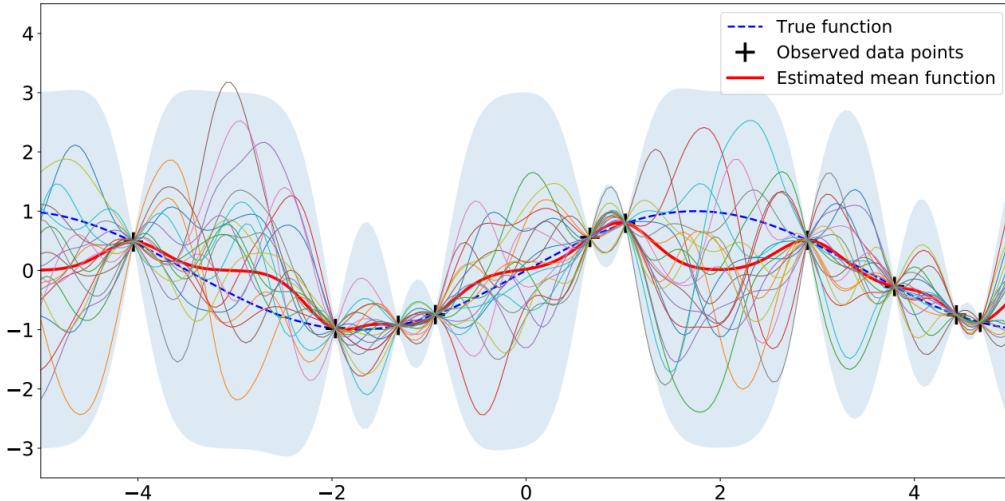


Figure 2.1: Example of GPR modeling possibly infinite posterior functions that fit the observed data points and the mean function obtained from the probability distribution of these functions as the red line [10].

2.2.1 Mean and Variance Functions

In Bayesian methods, a prior is the initial belief about the function to be estimated before observing any data. This prior is updated when data is observed to get a posterior. With GPRs, the prior is defined over functions. Specifically, the prior consists of:

A mean function $m(x)$

The mean function is typically set to zero, but in principle, it can be set based on prior knowledge about the data.

A covariance function $K(x, x')$

The covariance function (or kernel function) represents the belief about the similarity or relationship between data points in the input space. This will be discussed further in Subsection 2.2.2.

Once data is observed, the GPR only keeps functions that fit the data points. This is essentially the posterior, which is the prior updated with the observed data. For regression tasks, the mean function calculated by the posterior distribution of possible functions is the function used for regression predictions. The variance function calculated by the posterior distribution of possible functions quantifies the uncertainty of the predictions made by the GPR.

Thus, for an inverse design application, it is beneficial if both the mean function and the variance function are differentiable at a new observation.

Given a set of training data \mathbf{X} representing the input and \mathbf{y} representing the observed outputs, GPRs aim to predict the function values f_* at some new input points \mathbf{X}_* .

The joint distribution of the observed outputs \mathbf{y} and the function values f_* at the new input points, given the training inputs and new inputs, under the Gaussian Process prior is estimated as [9]:

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

Where $K(\mathbf{X}, \mathbf{X}')$ is the covariance matrix computed using the chosen kernel function for every pair of points in \mathbf{X} and \mathbf{X}' , \sim refers to approximate equality, \mathcal{N} represents the multivariate normal distribution, and σ_n^2 is the noise variance (assuming independent and identically distributed (i.i.d.) Gaussian noise).

Using properties of multivariate Gaussian distributions, the predictive distribution f_* is expressed as:

$$f_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*))$$

Where:

$$\bar{\mathbf{f}}_* \triangleq \mathbb{E}[\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_*] = \mathbf{m}(\mathbf{X}_*) + K(\mathbf{X}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1}(\mathbf{y} - \mathbf{m}(\mathbf{X})) \quad (2.2)$$

is the predictive mean, and:

$$\text{cov}(\mathbf{f}_*) = K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1}K(\mathbf{X}, \mathbf{X}_*) \quad (2.3)$$

is the predictive variance.

It can be observed that both the equations are differentiable if the kernel functions are differentiable. These kernel functions are discussed in the next section.

2.2.2 Kernel Functions

Kernel functions (denoted as $K(x, x')$), also known as covariance functions, are central to GPRs [9]. They define the covariance or similarity between points in the input space, governing how function values at different points relate to one another. The choice of kernel affects the ability of the GP to effectively capture functions.

Squared Exponential (SE) or Radial Basis Function (RBF) Kernel

$$k_{SE}(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right) \quad (2.4)$$

Where σ^2 is the variance, and l is the length-scale parameter, and the notation $\|x - x'\|$ represents the euclidean distance.

This kernel is one of the most widely used in GPRs. The *RBF* kernel assumes that points closer in input space have a higher covariance than points farther apart. The hyperparameter l , known as the length-scale, determines the "wavelength" or scale of the function's fluctuations. A smaller length-scale results in more rapid oscillations, while a larger value produces smoother functions.

RBF kernel is ideal when the underlying function is believed to be a smooth and continuous function. It is also infinitely differentiable, making it an ideal choice for inverse design.

Matern Kernel

$$k_{Matern}(x, x') = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\sqrt{2\nu} \frac{\|x - x'\|}{l} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{\|x - x'\|}{l} \right) \quad (2.5)$$

Where ν controls the smoothness of the function, $K_\nu(.)$ is a modified Bessel function, $\Gamma(.)$ is the gamma function, and l is the length-scale.

The *Matern* kernel provides a generalization of the *RBF* kernel, introducing an extra parameter, ν , that controls the function's smoothness. The *RBF* kernel corresponds to the *Matern* kernel with $\nu \rightarrow \infty$. With increasing values of ν , the functions become smoother, and with decreasing values of ν , the function becomes less smooth. A common choice is $\nu = \frac{3}{2}$ or $\nu = \frac{5}{2}$, as these values offer a balance between computational effort and flexibility in modeling different degrees of smoothness.

Matern kernel is ideal for modeling physical processes that are not as smooth as those modeled by the *RBF* kernel, making it more realistic for certain data than the *RBF*. It should be noted that a GP with *Matern* kernel is only $\lceil \nu \rceil - 1$ times differentiable.

White Kernel

$$k_{white}(x, x') = \sigma^2 \text{ if } x == x' \text{ else } 0 \quad (2.6)$$

where σ^2 is the noise variance.

The *White* kernel is essentially noise. It adds a component of i.i.d. Gaussian noise to the model. This kernel contributes only to the diagonal of the covariance matrix, capturing the observational noise in the data. It is typically combined with other kernels as part of a sum-kernel (for example, an *RBF + White* sum-kernel) to account for the inherent noise present in real-world data, ensuring the GPR doesn't overfit to noisy observations.

Dot Product Kernel

$$k_{dot}(x, x') = \mathbf{x} \cdot \mathbf{x}' + \sigma_0^2 \quad (2.7)$$

Where $\sigma_0^2 \geq 0$ is a constant. This kernel computes the dot product between input vectors.

The *DotProduct* kernel is ideal when only a linear relationship is expected to be modeled by the function.

Rational Quadratic Kernel

$$k_{RQ}(x, x') = \left(1 + \frac{\|x - x'\|^2}{2\alpha l^2} \right)^{-\alpha} \quad (2.8)$$

where α is the scale-mixture parameter and l is the length-scale.

RatioanalQuadratic is seen as a scale mixture (infinite sum) of *RBF* kernels with different length-scales. It is ideal for modeling functions with varying degrees of smoothness and exhibits behavior at multiple length scales. This can happen when some features vary over shorter distances and some vary over longer distances.

For an effective GPR model, especially in complex datasets, it's often beneficial to combine multiple kernels. This approach captures various patterns

in the data, from long-range trends to short-range nuances and noise. Kernels can be combined using addition or multiplication, providing a rich set of functional forms the GP can approximate.

The kernels described above are all differentiable and their gradients will be stated in Section 4.6. The effectiveness of these kernels in obtaining an accurate forward model and a comparison of the performance of the kernel functions and their gradient evaluations during the optimization routines in the context of multi-phase ceramics will be explored in this work.

Chapter 3

Related Work

In this chapter, an overview on previous works on inverse modeling using machine learning models is presented. This should shed some light on the current progress in ongoing research, and how this thesis contributes to the overlapping fields of material science and computer science. The work in inverse modeling for multi-phase ceramics is non-existent, and there are no directly related works aside from the work by Pirkelmann et al. [8] described in Chapter 1 which this thesis builds upon. This, however, highlights the novelty of this study. Instead, research works that investigate inverse modeling in different domains, along with works that explore effective forward modeling with microstructures of different types of materials, are explored.

3.1 Inverse Design with Bayesian Optimization

Traditionally, various works have used Bayesian Optimisation for applications in inverse design. A study by Schneider et al. [12] compared various optimization techniques in optical scatterometry. Their forward problem was to evaluate the scattering behavior of a sample at a point in the parameter space using the geometrical parameters of the microstructures, which was done using a FEM implementation. To solve the inverse problem of reconstructing the geometric parameters for the scatterometric data, the authors specifically discussed gradient-based local optimization methods such as BFGS, L-BFGS-B, and the truncated Newton method along with the primary focus on Bayesian optimization with a GPR as its stochastic model for the objective function.

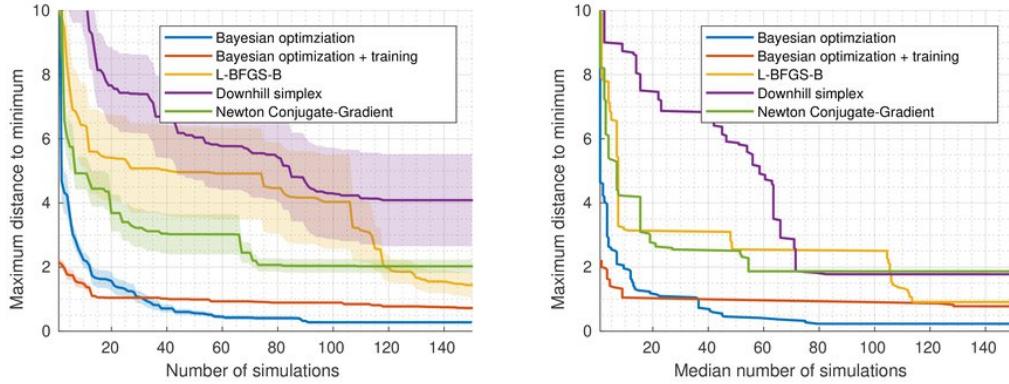


Figure 3.1: Comparison of optimization methods for solving the inverse problem in scatterometric data [12]. Left: Average objective function value as a function of the number of simulations for six independent simulation runs; Right: Median number of simulations required to achieve an objective function value $\lg[F(x)]$.

The results in Figure 3.1 from the study highlighted the efficiency of Bayesian optimization over local optimization methods. When equipped with training, Bayesian optimization was observed to converge after around nine iterations. Without training, the same convergence required around 25 iterations. Among the local minimization techniques compared, only L-BFGS-B converged within 150 simulations to the desired measurement uncertainty. This work did not use a surrogate function to model the forward function for fast evaluations but rather trained a GP to model the objective function instead as the stochastic model for the Bayesian Optimisation approach. All the compared optimization routines in this work directly evaluated the forward function with FEM simulations, which adds significant overhead in real-time applications. While work here shows that the number of iterations was lower for Bayesian Optimization than other techniques, it disregards the training time of the stochastic model in the overall computation time.

In another study, Iyer et al. [13] presented a data-centric mixed-variable Bayesian optimization (BO) framework for designing polymer nanocomposites that account for both categorical and numerical variables. The framework aimed at improving multiple design parameters, such as the dielectric breakdown strength, the loss tangent, and the permittivity. The FEM simulations were modeled by a Random Forest (RF) model for fast evaluations of the forward model. For inverse design, the study employed Bayesian optimization that used a Latent variable Gaussian process (LVGP) model as its stochastic model to navigate the design space. The optimization was carried

out in two forms: single criterion BO, which focuses on optimizing a single objective, and multicriteria Bayesian optimization (MBO), which sought to balance multiple objectives simultaneously. The results demonstrated the efficacy of the BO framework in identifying optimal designs for nanocomposites with significant improvements over the baseline properties of pure polymers. For instance, in single criterion BO, an optimal design with a 75.9% improvement over pure polystyrene (PS) was identified. The MBO, which ran for 70 iterations starting with 30 random samples, successfully located improved designs that outperformed the random initial samples and provided a set of optimal solutions representing trade-offs among different properties. In slight contrast to the work by Schneider et al. [12], this work uses an RF surrogate for modeling the forward function, and the Bayesian optimization employed uses an additional LVGP model as the stochastic model of the objective function.

While Bayesian optimization was the main approach for inverse design in the two works [12, 13], it is potentially expensive to use in an inverse design methodology for real-time applications in multi-phase ceramics since it requires training a secondary GP to model the objective function as a stochastic model, which is different from modeling the forward function (for example, $f : X \rightarrow Y$, microstructure (X) to material property (Y)) as a surrogate model. The objective function changes every time a new material property is desired, and retraining a GP for every design would make the optimizations potentially time-consuming. Instead, replacing the FEM simulations with a cheaper to evaluate ML surrogate model, and skipping the training steps for the stochastic GP by using gradient-based optimizers which generally do not train an internal stochastic model (such as L-BFGS-B which was also shown to be effective by Schneider et al. [12]), can potentially be much faster and viable for real-time applications.

3.2 Gaussian Models in Material and Electromagnetic Design

Various works have applied Gaussian-based ML models for inverse design in different domains. A study by Sato et al. [14] explored the design of electromagnetic devices with a Direct Inverse Modeling (DIM) methodology. Their aim was to reduce the computing cost of the iterative forward field analysis with a methodology capable of finding multiple solutions for inverse design. Their forward analysis was performed using a FEM implementation, which would simulate the electrical properties for different sample design parame-

ters. The proposed DIM uses the Gaussian kernel regression as a surrogate function to replicate the FEM analysis. A residual function representing the difference between the required electrical properties and those predicted by the surrogate function is then minimized using Newton's method. Their proposed method outperformed traditional optimization techniques, such as genetic algorithms directly evaluated with FEM, in terms of computational cost and speed as seen in Figure 3.2 by a significant margin. Unlike the GA optimization, the proposed DIM once trained can perform inverse designs in seconds for any number of samples.

COMPARISON OF ELAPSED TIME

	GA with FEM	Direct inverse method
Training data set	N/A	11 hours (for 640 train sets)
GA Optimization process with FEM	70 hours	N/A.
Newton method	N/A	15 sec.
Total computational time	70 hours	11 hours + 15 sec.

Figure 3.2: Comparison of DIM with GA optimization [14].

In another study, Daud et al. [15] implemented Gaussian Process-Based Inverse Modeling for a Seabed Logging Application. The work uses a Gaussian Process (GP) for forward modeling in the Seabed Logging (SBL) application, where the goal is to predict the depth of a hydrocarbon reservoir for an observed electromagnetic (EM) profile efficiently. The authors used a FEM implementation with Computer Simulation Technology (CST) to generate prior electromagnetic (EM) responses at unknown depths, which are then used by the GP for training as a surrogate model. A gradient descent procedure is then used to minimize the squared error between the observed EM profile and the EM profile at the candidate depth (evaluated by the GP). Their results showed that their forward model with the GPR provided EM profiles at untried depths with good accuracy: an average RMSE of 3.49×10^{-9} , and with low computational time: less than 2 min for the GPR compared to 45 min for the CST data acquisition. The percentage error between estimates of depth by the GP-based inverse design methodology and the true depth was 0.0260% on average. The study showcased the potential of the GP-based methodology in effectively and efficiently modeling hydrocarbon depths in the SBL application.

Another work by Saunders et al. [16] used functional Gaussian process (fGP) to model mechanical behavior predictions of additively manufactured (AM) microstructures. fGPs are an extension of standard GPs, able to model functional data. The goal of their research was to create a method that is not computationally expensive and approximates the stress-strain behavior of AM microstructures.

The data was obtained from 50 generated RVEs and Crystal Plasticity Finite Element (CPFE) simulations were used for evaluating the mechanical response of polycrystalline materials represented by the RVEs. An fGP surrogate model was proposed to model the data from these simulations with functional input features: the uniform kinematic displacement boundary conditions (u) and a loading parameter (λ) such as amplitude over time, and non-functional input features: the constitutive model parameters (θ) and microstructural features.

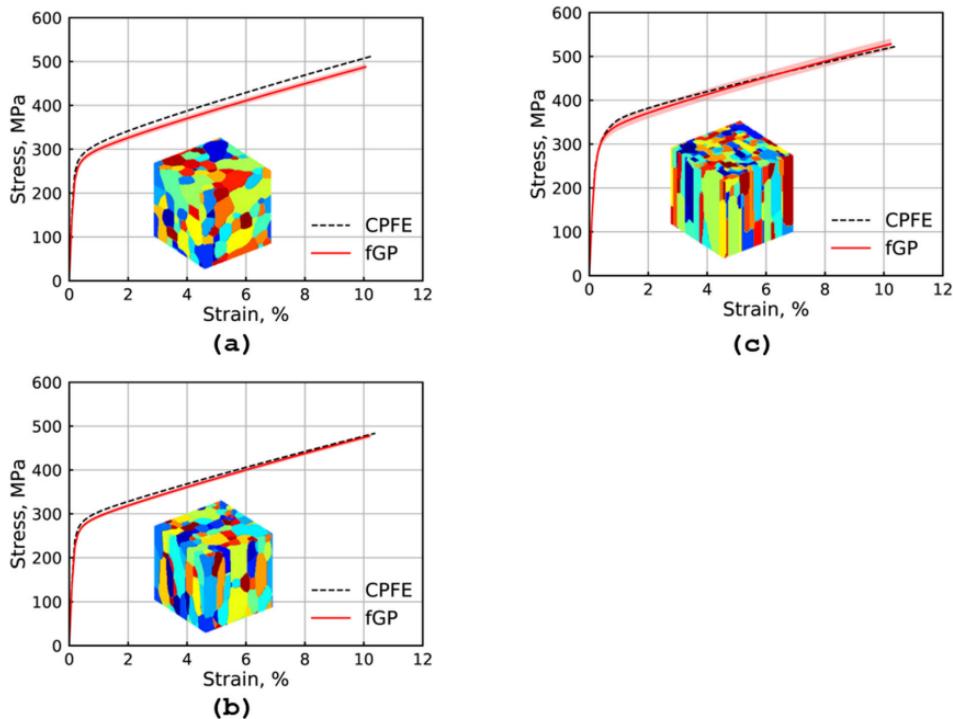


Figure 3.3: fGP predictions compared with CPFE simulated data for 3 RVEs not used in the training data [16].

The fGP was able to emulate the results of the CPFE simulations, making it applicable to various crystal plasticity data sets. They showed that the

fGP network could evaluate data three orders of magnitude faster than the corresponding CPFE model. The fGP predictions for 3 RVEs not used in the training data can be seen in Figure 3.3. Where the CPFE results took around 3 days of computation time, the fGP with 95% prediction intervals only took seconds. While this work did not use the fGPs in an inverse modeling application, the effectiveness of a functional form of GPs in forward modeling of microstructure design (similar to the work in this thesis) was shown, which is the first step towards inverse design.

Another study by Govinath et al. [17] applied GPRs and Minimax Probability Machine Regression (MPMR) to model the Hardness and Fracture Toughness of Liquid Phase Sintered (LPS) Alumina ceramics using microstructural descriptors such as grain size and porosity. Their results showed that the GPR is much better than the MPMR with validation using both the training set and testing sets. The R^2 score for the GPR was found to be very close to unit value ($= 1$) for both hardness and Fracture Toughness, suggesting its applicability in forward modeling in other ceramic parameters. LPS Alumina is a class of ceramics as well, and the microstructural descriptors in this work are very similar to the microstructural data dealt with in this thesis for multi-phase ceramics. This is potentially promising for the application of GPRs for forward modeling in this work.

These works [14, 15, 16, 17] show the successful application of ML techniques such as GPRs as surrogates in material design and suggest their applicability for inverse design, which further motivates the work in this thesis.

3.3 Other Machine Learning Approaches in Foward Modeling and Inverse Design of Microstructures

Many advanced ML approaches have been developed for tackling inverse problems in material design. R. Liu et al. [18] performed a study into a predictive machine learning approach for microstructure optimization in polycrystalline alloys with the example of the magnetoelastic Fe-Ga alloy. They aimed to understand how its microstructure could be optimized for achieving the desired Young modulus (E), yield strength (Y), and magnetostrictive strain (ms). They proposed a machine learning (ML) methodology as an alternative to traditional optimization techniques. Their approach consisted of an ML-based preprocessing to locate the most significant regions in the search space so that the search force could be guided to a smaller space.

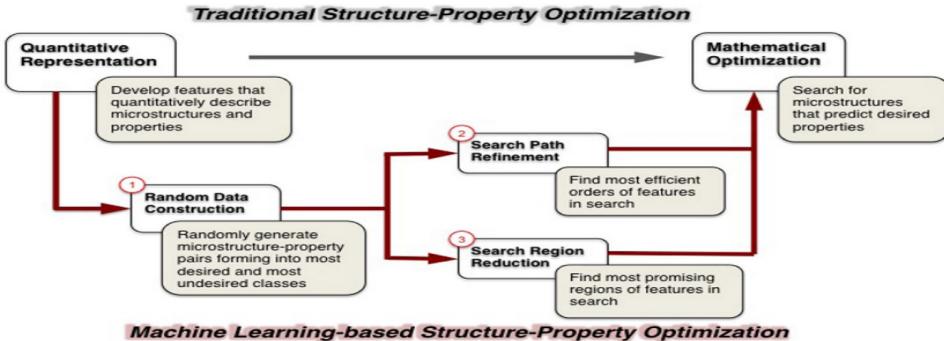


Figure 3.4: ML-based preprocessing framework: A data preparation step (1) followed by a search path refinement (2) and a search region reduction (3) to concentrate the search force [18].

The ML framework can be seen in Figure 3.4. The ML-based preprocessing steps consisted of a search path refinement and a search region reduction before the actual optimization search for the microstructures. The results showcased a reduction in average running time by up to 81.62% when compared to exhaustive search and guided search, while also achieving solution optimality that traditional methods such as linear programming and genetic algorithms failed to reach. Moreover, for problems with multiple optimal design solutions, the ML method proved superior in identifying a more complete set of solutions. This work gave insights into how one can approach inverse problems by addressing high-dimensional design spaces and the non-uniqueness of solutions, by dimensionality reduction and identifying a complete set of multiple solutions.

Tan et al. [19] proposed a convolutional neural network (CNN) based methodology for the inverse design in multi-phase microstructures directly using microstructural images. A deep convolutional generative adversarial network (DCGAN) is trained to generate images of microstructures that satisfy certain geometric constraints based on the initial data, and this along with the initial data is used as training data. The target outputs are the associated compliance tensors, a stress-based material property of the microstructures based on a FEM analysis. A CNN is then trained to map the microstructural images to their corresponding compliance tensors, providing a fast and accurate forward model for predicting mechanical properties from image data. The combined DCGAN-CNN model is then utilized in an inverse design framework, where given desired compliance tensor values, the loss function based on the MSE between the desired output and predicted output is mini-

mized using L-BFGS-B. The designs were evaluated by plotting the predicted compliance tensor of the final optimized structures against the desired compliance tensors as seen in Figure 3.5. The plots suggested that the accuracies of the inverse designs were within 10% of desired values, showcasing that the combined DCGAN-CNN framework with an L-BFGS-B optimizer was effective at inverse modeling in multi-phase microstructures. This work employs a CNN to directly model a forward function with microstructural image data which is impressive but unnecessary for the work in this thesis as the microstructural descriptors for multi-phase ceramics can be naturally quantified as described in Section 4.1. This also means that with numerically quantified features such as the microstructural descriptors, it is viable to restrict the design space to contain only physically meaningful designs. The same cannot be said for the optimized microstructures generated by GAN, where it is possible that physically infeasible microstructures can be generated. Regardless, this work is another example of the effectiveness of the L-BFGS-B optimizer for inverse design.

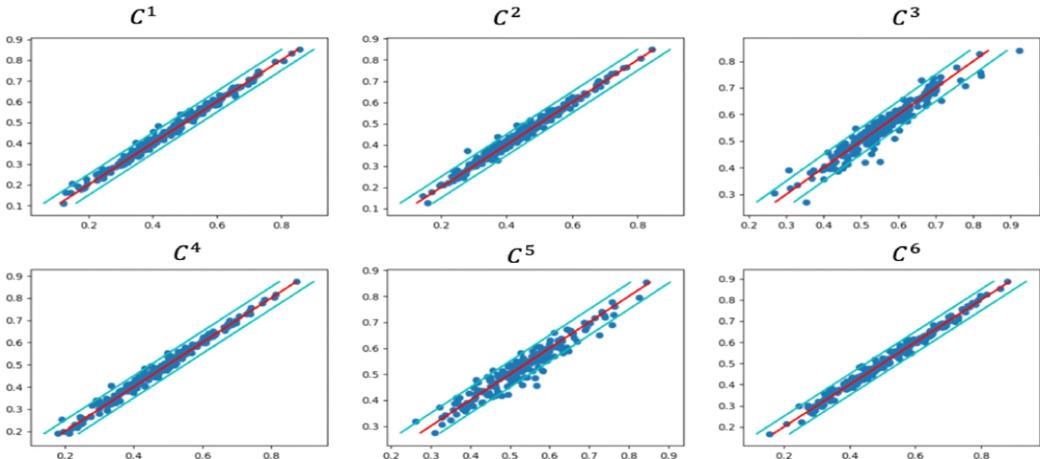


Figure 3.5: Compliance tensors of the optimal structures plotted against the desired values [19].

3.4 ML-based Surrogates for Inverse Modeling in other Domains

L. Cai et al. [20] compared the performance of the K-nearest neighbor (KNN), XGBoost, and multi-layer perceptron (MLP) for inverse modeling of the left ventricular (LV) myocardium. The study employed a forward FEM simulator of LV diastolic filling to generate training data and ML models

are trained to model the relationships between pressure–volume and pressure–strain, respectively. The inverse parameter estimation problem was then formulated using the surrogate ML models by minimizing a squared error-based objective function with Trust-region-reflective, a gradient-based optimizer. In Figure 3.6, it can be seen that in the curves obtained with inverse design using the surrogate models, XGBoost performs the best in accordance with their simulations. The study indicated that the XGBoost model was the most effective in learning the relationships of pressure-volume and pressure-strain in the LV myocardium, with significantly lower uncertainties in the inversely solved parameter estimates compared to the KNN and MLP models. The study also briefly explored a GPR using the RBF kernel, which is the primary model used in this thesis, and the resulting performance was indicated to be only slightly poorer than the XGBoost for inverse modeling, further showcasing its potential in inverse problems.

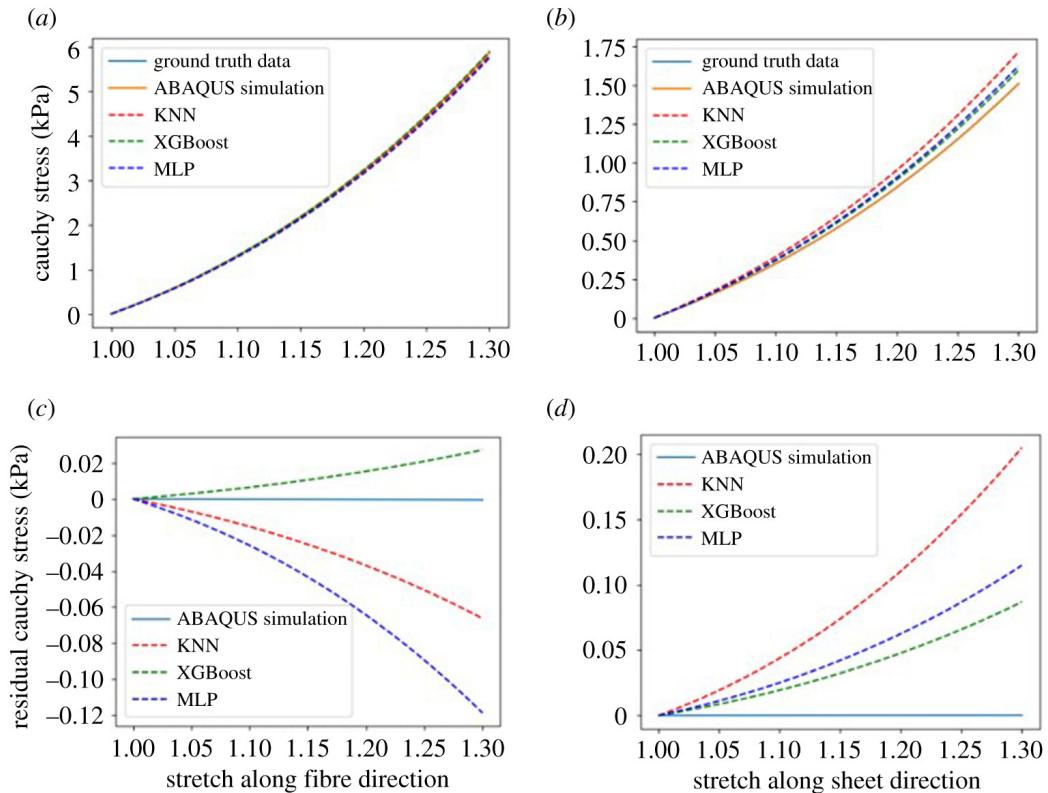


Figure 3.6: Cauchy stress with estimated parameters along fibre (a) and sheet (b) directions and residual Cauchy stress along fibre (c) and sheet (d) directions for different approaches [20].

Humfield et al. [21] proposed using Long Short-Term Memory (LSTM) neural networks for solving an inverse heat transfer problem in curing composite parts. Curing is a heat-induced process where composite materials are solidified to achieve their final strength. A finite element model was used to simulate the curing of composite parts, which involves using thermal stacks and air temperature profiles, collectively, the boundary conditions (BC), to estimate the temperature histories of the composite and temperature histories of the tool. Two LSTM neural networks were employed to model the forward simulations. One LSTM model predicts the tool temperature and the other predicts the part temperature. A third NN was designed with a simple feedforward architecture for multi-objective optimization of the temperature cycle. In this third NN, thermal stacks and air temperature profiles are provided as inputs and conformity to specs (pass or fail) are predicted as outputs. The LSTM prediction errors compared to FE outputs were less than 1°C .

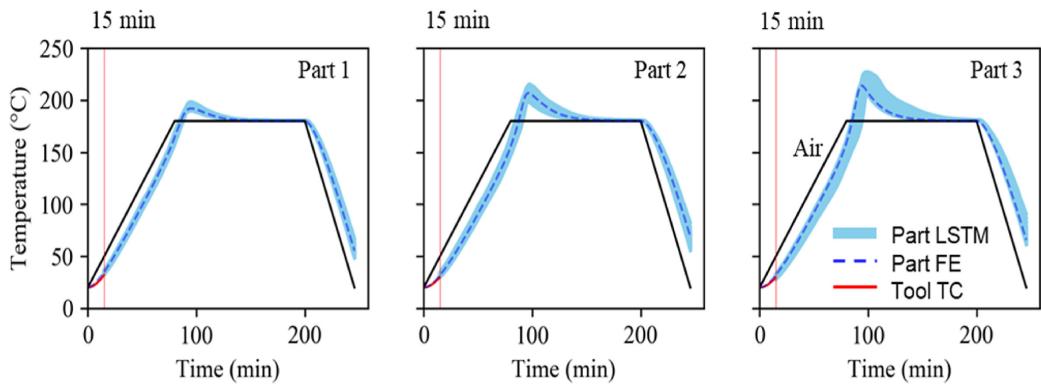


Figure 3.7: Region of viable solutions predicted by the LSTM-based ML framework compared with the true response predicted by FE for part temperature histories in the inverse heat problem [21].

The study used an example of curing of three HEXCEL AS4/8552 composite parts with different thicknesses. Figure 3.7 shows the results of the ML framework for solving the inverse problem for three thermocouples. The ML framework was able to identify 58 possible BC solutions that closely satisfy the process specifications. This work was another example of the effective use of ML-based surrogate models for solving inverse problems.

Through this literature review, it was seen that there have been various studies on inverse modeling in different domains. However, in the case of multi-phase ceramics, inverse modeling is unexplored and this thesis contributes to the ongoing study of these ceramics by implementing effective

forward models using GPRs and using them as surrogate models for inverse design. This work provides a full characterization of GPRs in inverse design for multi-phase ceramics and discusses various best practices that may be applicable and relevant to other domains as well.

Chapter 4

Methodology

4.1 Data Preparation

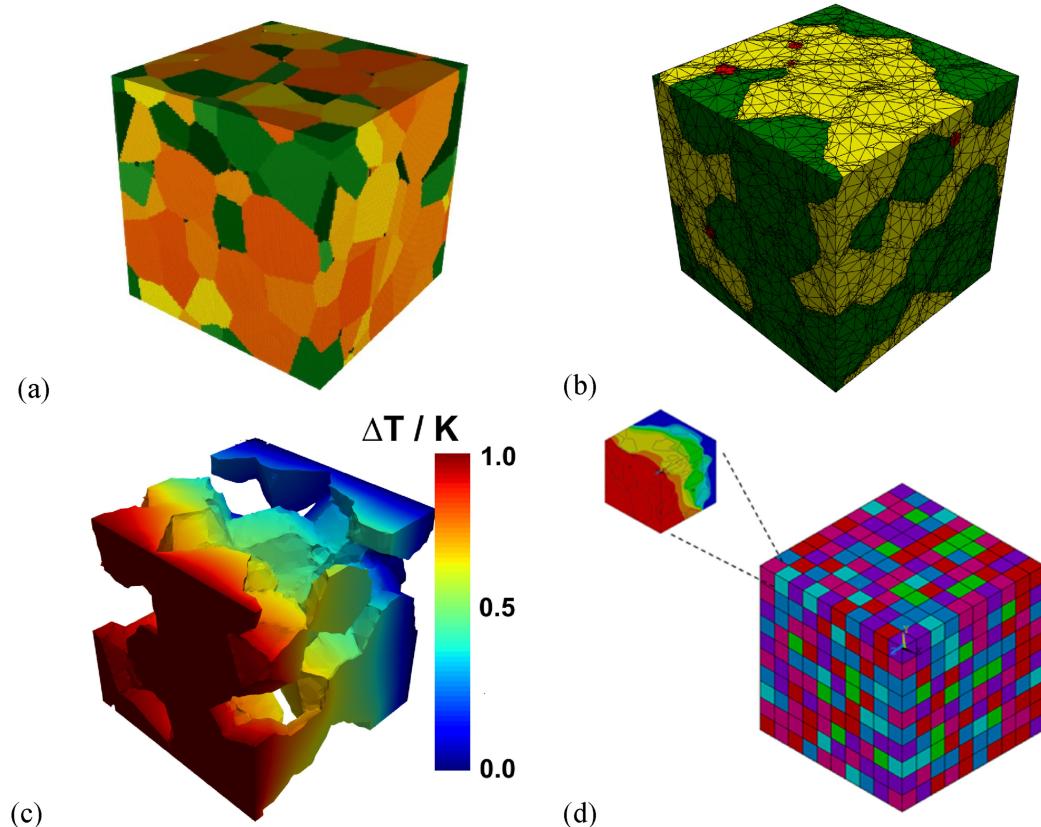


Figure 4.1: (a) RVE generated by GeoVal for a two-phase ceramic composite; (b) Tetrahedral mesh generated with VoxSM; (c) FE result example for thermal conductivity simulations (d) Homogenization technique using a larger cell composed of multiple small RVEs [8].

As described earlier, this work focuses on the example of multi-phase ceramics of Zirconia-Alumina composites. The data was generated through a simulation toolchain developed by Pirkelmann et al. [8]. The representation of such multi-phase ceramics is done through the generation of Representational Volume Elements (RVEs) with periodic boundary conditions, which can be seen as the artificial geometrical representation of the morphology of the multi-phase composites. An example can be seen in Figure 4.1a. This construction is performed using a software named GeoVal, which synthesizes elements of microstructure formation simulation and geometrical constructions. Initially, spheres representing different phases with varying sizes and distributions are positioned randomly within the RVE. These spheres are then redistributed to simulate particle rearrangement and irregularity during forming processes such as spray drying or slip casting and milling and transformed into Voronoi-polyhedra. The geometric objects are then converted into voxel structures, which allows the introduction of porous features. Structural properties of the voxel structure, such as chord length, are quantitatively analyzed for comparison with experimental microstructures.

The voxel mesh is then smoothed and simplified into a triangular surface mesh while retaining detailed information about the particles, phases, and volume fractions using a software called VoxSM. This is done to avoid computational complexity arising from possible artifacts in the original voxel mesh. The surface mesh is then used to create a volume mesh that is compatible with the FE software. Using the volume mesh which can look like the representation in Figure 4.1b, FE simulations are conducted using ANSYS to calculate various macroscopic material properties. Loads such as mechanical strain, electrical potential, or temperature differences are applied in the FE model to obtain material property tensors such as thermal expansion, thermal conductivity and other elastic properties as seen in Figure 4.1c. The materials were assumed to be operating under a temperature of 25°C. For materials exhibiting isotropic properties, a homogenization step is included in the process (illustrated in Figure 4.1d). This involves creating a larger cell composed of multiple RVEs, each representing different orientations. This step is crucial for removing anisotropies caused by the limited size of individual RVEs and yields homogenized macroscopic properties after conducting the final FE analysis.

The entire simulation chain is automated, allowing for the generation and meshing of numerous RVE structures using scripts. To generate a wide range of volume fractions and particle sizes a random sampling strategy was used [8]. Using this, a comprehensive database from the simulation results was constructed for training machine learning models. The structures are char-

acterized by three phases, namely, Zirconia, Alumina, and the pores between the grains. Nine easily quantifiable microstructural descriptors based on the phases are considered:

- volume fraction of each of three phases
- average chord length for grains of each of three phases
- variance of chord length for grains of each of three phases

The chord lengths may also be referred to as grain sizes, or particle sizes.

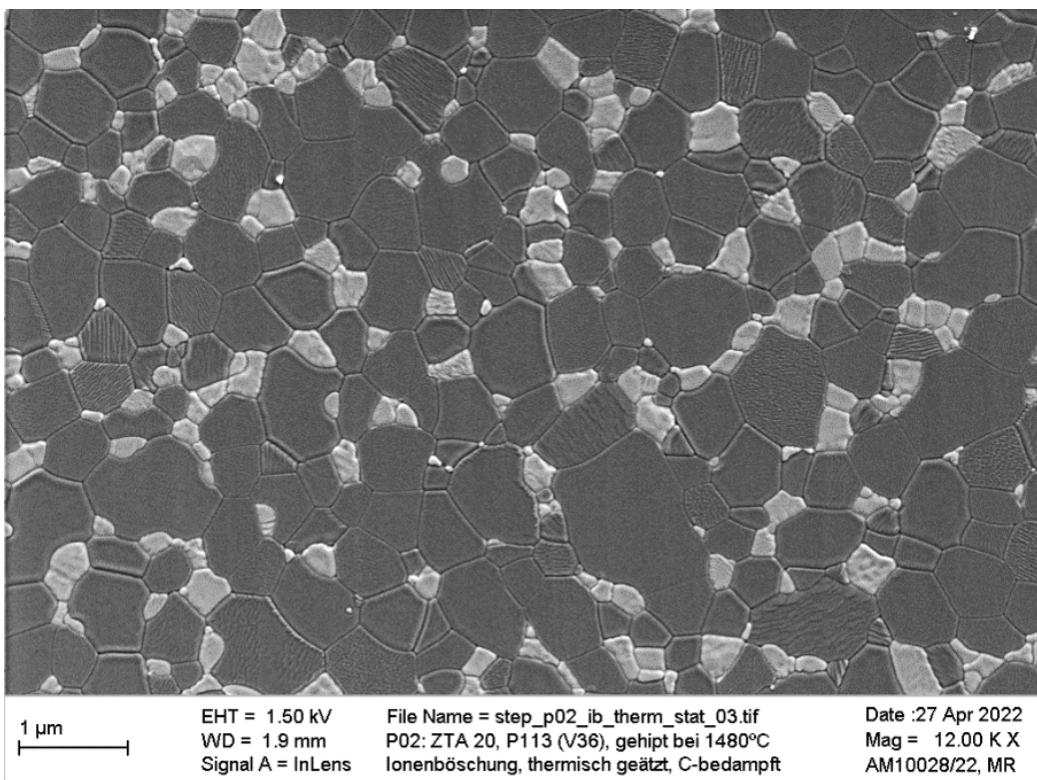


Figure 4.2: A scanning electron microscopy (SEM) image of a ZTA microstructure (dark gray = Alumina, light gray = Zirconia) [8].

As an example, the microstructural descriptors can be seen in the microstructure shown in Figure 4.2. This ZTA ceramic sample was produced by hot isostatic pressing (HIP) and through image analysis the grain boundaries were recognized, from which the chord lengths were derived. For this sample, the volume fractions of Zirconia and alumina were 14.3% and 85.7% respectively, and the average chord lengths of Zirconia and Alumina were $0.42\mu\text{m}$ and $0.2\mu\text{m}$ respectively.

Several associated material properties were generated, but the most relevant properties for the inverse modeling application considered in this work are:

- Coefficient of Thermal Expansion (CTE)
- Coefficient of Thermal Conductivity (CTC)
- Young's Modulus
- Poisson Ratio

The database consists of 1815 entries for CTC, Young's Modulus, and Poisson Ratio. Due to the presence of null values for CTE, it only has 1514 samples.

4.2 Data Preprocessing

Data Cleaning: The first step in data processing was to ensure the integrity of the dataset. Any rows with missing values (NaNs) in the data were identified and removed. This ensures the consistency and completeness of the dataset, eliminating potential biases or inaccuracies that could arise from incomplete data.

Normalization with Standard Scaler: The features were normalized using a Standard Scaler. This normalization is essential for models that are sensitive to the scale of the input features. The Standard Scaler transforms each feature to have a mean of zero and a standard deviation of one, following the formula:

$$x_{\text{scaled}} = \frac{x - \mu}{\sigma} \quad (4.1)$$

Here, x is the original feature value, μ is the mean of the feature, and σ is the standard deviation of the feature. This scaling process ensures that each feature contributes proportionally to the predictive model, preventing features with larger magnitude ranges from dominating the model's behavior.

In the original work [8] that generated this data, all nine microstructural descriptors were used for training the forward models for each property. However, to address the curse of dimensionality in the ill-posed problem of inverse design [18], the dimensionality can be reduced while also maintaining a good trade-off between accuracy and performance. The feature sets considered in this work for the data are:

Feature Sets

Feature Set	Features
Full Feature Set	<ul style="list-style-type: none"> • Volume Fraction of Zirconia (VF_{ZrO_2}) • Volume Fraction of Alumina ($\text{VF}_{Al_2O_3}$) • Volume Fraction of Pores (VF_{pores}) • Average Chord Length of Zirconia (AvgCL_{ZrO_2}) • Average Chord Length of Alumina ($\text{AvgCL}_{Al_2O_3}$) • Average Chord Length of Pores (AvgCL_{pores}) • Variance of Chord Length of Zirconia (VarCL_{ZrO_2}) • Variance of Chord Length of Alumina ($\text{VarCL}_{Al_2O_3}$) • Variance of Chord Length of Pores (VarCL_{pores})

Feature Set	Features
8-feature Set	<ul style="list-style-type: none"> • Volume Fraction of Zirconia (VF_{ZrO_2}) • Volume Fraction of Pores (VF_{pores}) • Average Chord Length of Zirconia (AvgCL_{ZrO_2}) • Average Chord Length of Alumina ($\text{AvgCL}_{Al_2O_3}$) • Average Chord Length of Pores (AvgCL_{pores}) • Variance of Chord Length of Zirconia (VarCL_{ZrO_2}) • Variance of Chord Length of Alumina ($\text{VarCL}_{Al_2O_3}$) • Variance of Chord Length of Pores (VarCL_{pores})

Note: The Volume Fraction of Alumina ($\text{VF}_{Al_2O_3}$) was recognized as redundant for consideration in the feature set and excluded due to the relation: $\text{VF}_{ZrO_2} + \text{VF}_{Al_2O_3} + \text{VF}_{pores} = 1$, as it can be inferred from the other two volume fractions.

Feature Set	Features
3-feature Set	<ul style="list-style-type: none"> • Volume Fraction of Zirconia (VF_{ZrO_2}) • Volume Fraction of Pores (VF_{pores}) • Chord Length Ratio (CLR)

Note: This feature set was considered based on the premise that the role of the chord lengths of the pores and the variance terms of all phases play a very minimal role in model predictions. It is further reduced by considering the ratio of the average chord lengths between Zirconia and Alumina as a single feature instead of both chord lengths. This was based on the hypothesis that this ratio might generalize the effect of these chord lengths in influencing the material's properties. The chord length ratio is given as:

$$\text{CLR} = \frac{\text{AvgCL}_{ZrO_2}}{\text{AvgCL}_{Al_2O_3}}.$$

Feature Set	Features
2-feature Set	<ul style="list-style-type: none"> • Volume Fraction of Zirconia (VF_{ZrO_2}) • Chord Length Ratio (CLR)

Note: Further reduction considered scenarios where porosity was absent (equal to zero). The 3-feature set was filtered with the condition $\text{VF}_{\text{pores}} = 0$. For these data points, the model focused on just two features: the volume fraction of Zirconia and the chord length ratio. This approach was based on the idea that in the absence of porosity, the relationship with the volume fraction of zirconia and the chord length ratio would be mostly indicative of the material's behavior.

4.3 Data Exploration

This section provides a brief exploration of the data used in this work for different material properties, which should enable visual pattern recognition and set expectations for the surrogate models and inverse design results.

Figure 4.3 shows the relationship of the volume fraction with the material

properties. By examining the colors along the parallel coordinates, it can be observed that the material properties are either positively correlated (CTE and Poisson Ratio) or inversely correlated (CTC and Young's Modulus) to the volume fractions of Zirconia. This confirms that the volume fractions play a major role in determining the material properties. Further, the upper and lower bounds for each property can also be seen here. The Poisson Ratio ranges within [0.228, 0.322], the Young's Modulus within [205.51, 387], the CTE within [5.4183, 7.0143], and the CTC within [3.016, 34.138].

Figure 4.4 shows the volume fraction of Zirconia plotted against the material properties with the effect of porosity coded in the colormap using the full feature set. It can be seen that for each material property, there are multiple microstructures that correspond to a material property value. This means there are multiple microstructure solutions that can be found with optimizations for inverse design for a desired material property value. It can be seen that for each property value, there are only small variances in volume fraction. For Young's modulus, large variances can be observed by small changes in porosity. In CTE, and Poisson ratio, the effect of porosity is much lower and they can be determined almost solely with the volume fraction. CTC is also not too strongly affected by porosity but shows a slightly higher variance in the volume fraction of Zirconia.

The 2-feature set is the most studied data subset in this work. The reasoning behind this is to use a simpler dataset that has high performance with both speed and accuracy while making 3D visualizations feasible for further qualitative analysis. This also sets the groundwork for further analysis with higher dimensional feature sets. With this dataset, the effect of porosity can be ignored and the focus can be on other patterns seen in the data. To understand the patterns within this data subset, the volume fraction of Zirconia is again plotted against material property but instead with the particle size ratios coded in the colormap. This is seen in Figure 4.5. Since the porosity is now set to 0, very linear relationships can be observed with Young's Modulus, Poisson Ratio, and CTE. For these three properties, multiple solutions will correspond to different values of particle size ratios with very small or no changes in the volume fractions. However, with CTC, it can be seen that the larger variances in its values can be explained by the particle size ratios as the curve color changes in a visible pattern. This can be explained as the conductivity of the material is dominated by Alumina which is the phase with higher CTC. This means heat flux through the material will predominantly happen through the Alumina particles. At the particle boundaries, the heat flux is inhibited due to imperfections of the crystal lattice at the boundary. Therefore, for larger particle size ratios, the average distance the

heat can flow before being stopped by a grain boundary influences the overall thermal conductivity. For CTC, a higher variance can be expected with both volume fraction and particle size ratios, and multiple solutions will correspond to slightly larger variances in both volume fraction and particle size ratios. This is what should be expected as the outcome of the inverse design methodology.

Visualization of the distribution of microstructures for the 2-feature set can be seen in Figure 4.6. It can be seen that through the random sampling strategy used to generate the data [8], the data is very dense between particle size ratios of 0 and 2. There exists some data for particle size ratios above 4 but it is only for higher volume fractions. It is also somewhat impractical to have microstructures with small volume fractions of Zirconia and large chord length ratios, therefore there is no data in the upper-left section of the data space. It should still be possible to have microstructures between particle size ratios of 3 and 4 for small volume fractions of Zirconia but this data is missing. This may affect the performance of the model for this unseen data. Since the 2-feature set is filtered with the condition of $\text{VF}_{\text{pores}} = 0$, the amount of data samples reduced from 1815 to 450 for CTC, Young's modulus, and Poisson ratio. Due to the presence of null values for CTE, the full dataset had around 1514 samples and after the reduction to the 2-feature set, only 149 samples remained.

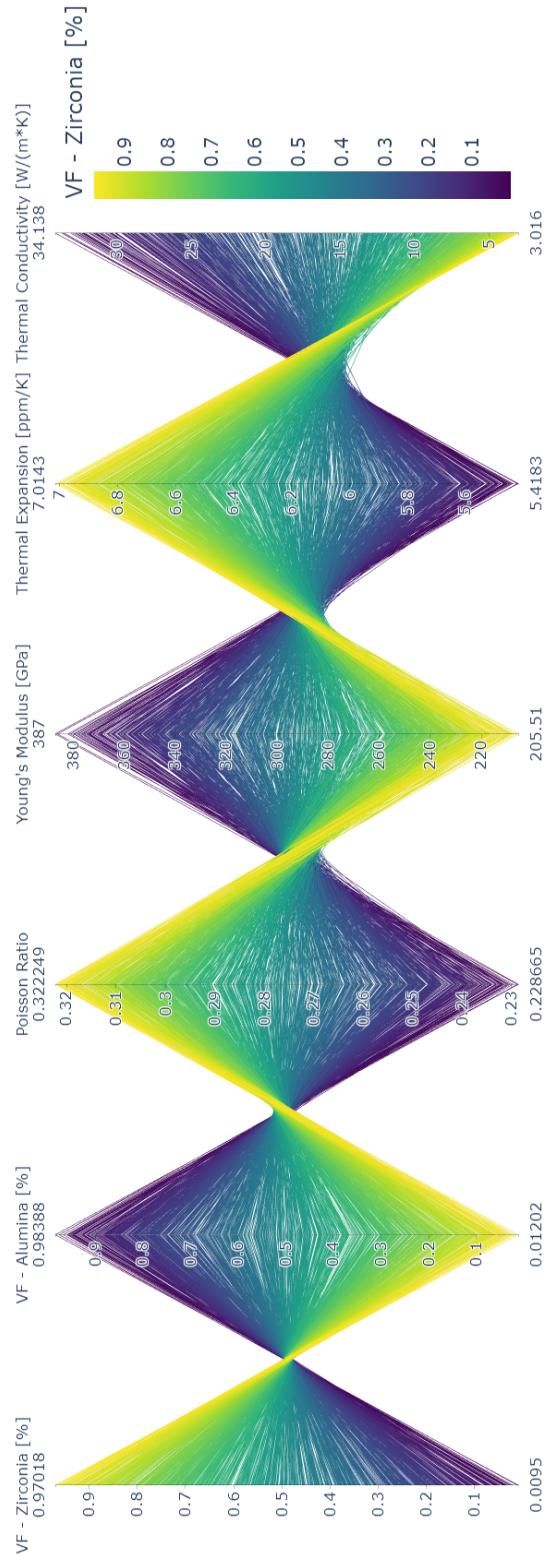


Figure 4.3: Parallel plot of the Volume fractions and Material Properties.

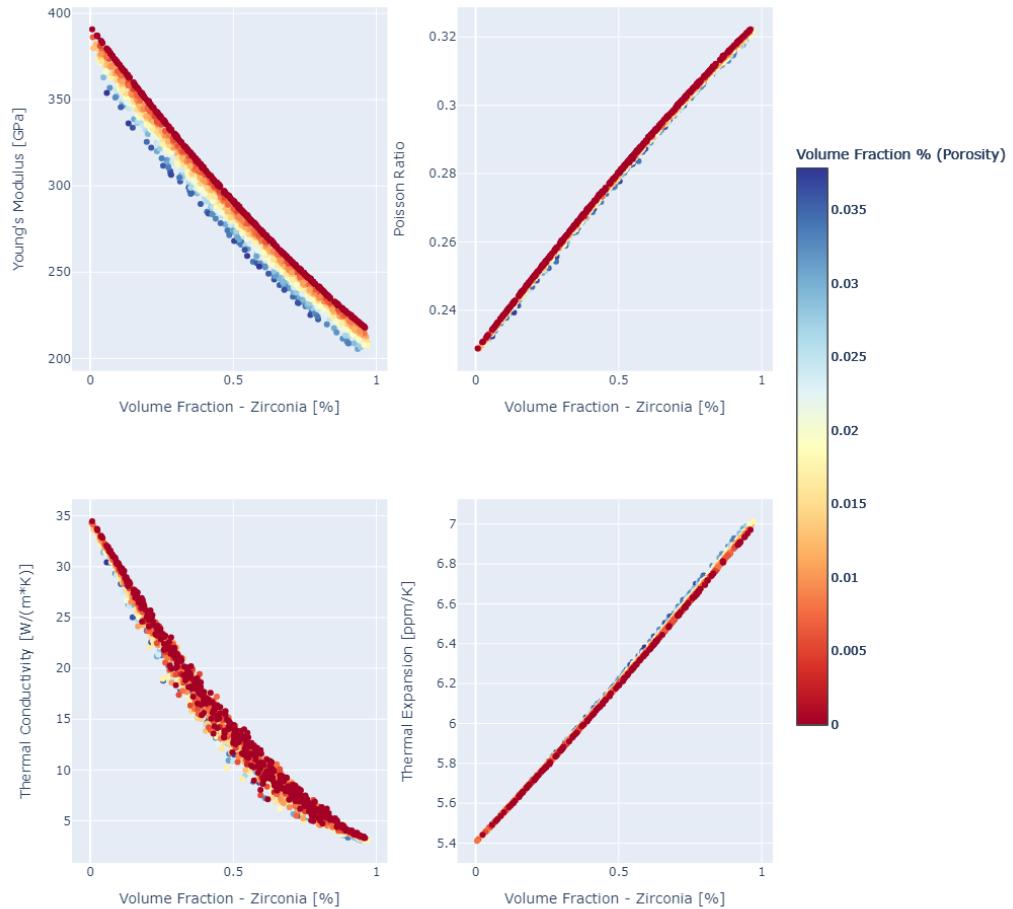


Figure 4.4: Volume Fraction (Zirconia) vs Material Properties with the effect of Porosity using the full feature set.

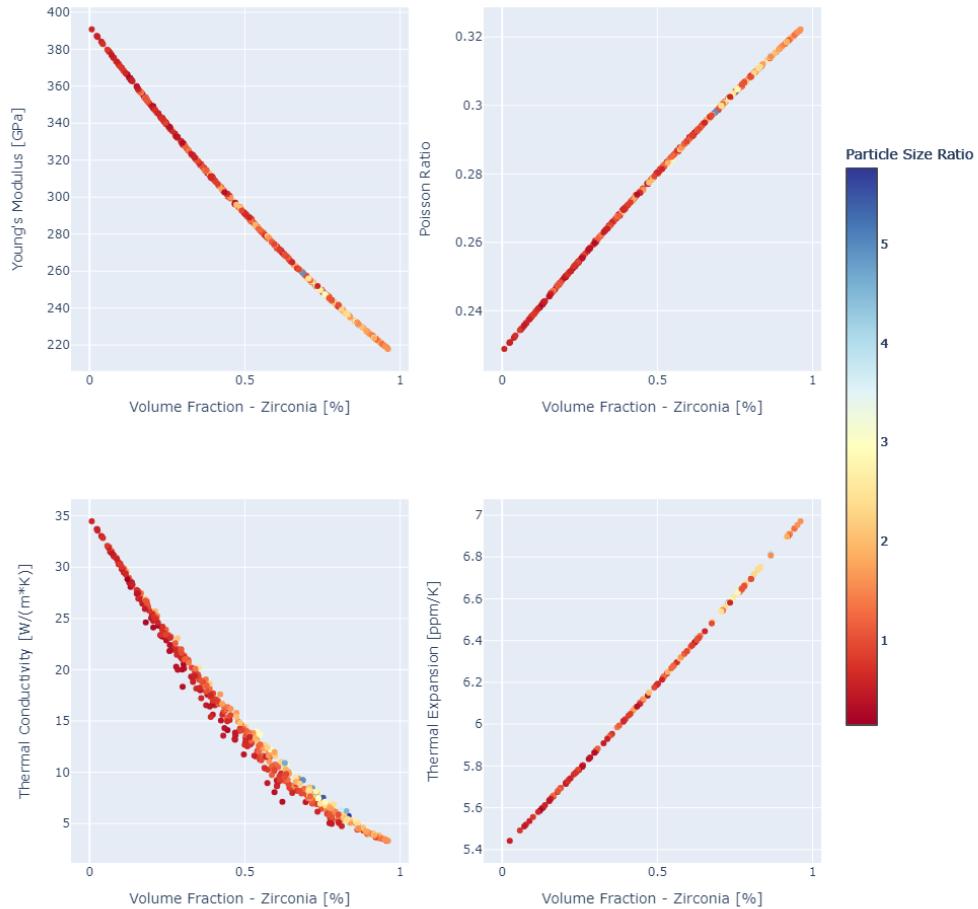


Figure 4.5: Volume Fraction (Zirconia) vs Material Properties with the effect of particle size ratio using the 2-feature set.

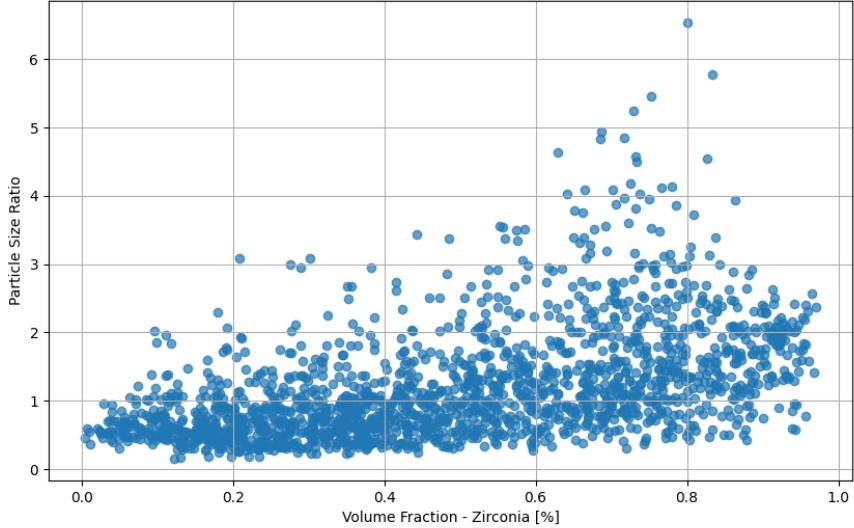


Figure 4.6: Distribution of microstructures in the 2-feature design space.

4.4 Surrogate Modeling with GPR

One of the preliminary goals of this work is to train forward models that can accurately predict material properties in ZTA/ATZ ceramics from their microstructural features. These models are intended to replace expensive FEM simulations, thereby streamlining the design process. As noted by Pirkemann et al. [8], the generation of 1815 data points through the automated simulation tool required around 320 hours, which means approximately 10 minutes per data point. By training an effective forward model on the FE simulations, it is aimed to use the model as a surrogate function that allows closely approximated evaluations of the true material relationships in less than a second. Achieving high accuracy and efficiency in these predictive models is crucial as they form the foundation for the subsequent inverse design process.

Since the data from the FEM simulations is a numerical dataset, with various microstructural descriptors (feature set X) and material properties (target set Y), forward modeling is essentially a regression task to train a surrogate function $f : X \rightarrow Y$. For the purpose of this work, the GPR is chosen as the modeling approach. In the original work [8] that generated the data, several other regressors were trained and evaluated. The Gradient Boosting Regres-

sor (GBR) was indicated to perform the best out of all models. However, for the purpose of this work, GBRs are not suitable models for inverse design as they are prone to having regions of zero-value gradients. However, since the accuracy of the GBR was high, it is used as the baseline performance to compare the performance characteristics of the GPR with it.

For implementing the GPR, the Scikit-optimize (skopt) [22] library wrapper for the GP implemented in Scikit-learn (sklearn) [23] Python library is used. One of the key characteristics of this implementation of GPR is that it assumes the prior mean $m(X)$ to be zero. This simplifies the mean prediction function described earlier (Equation 2.2) as:

$$\bar{\mathbf{f}}_* = K(\mathbf{X}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{y} \quad (4.2)$$

The sklearn implementation also gives a measure of the uncertainty as the standard deviation at the input point which is the square root of the variance function (Equation 2.3) as:

$$\begin{aligned} \text{std}(\mathbf{f}_*) &= \sqrt{\text{cov}(\mathbf{f}_*)} \\ &= \sqrt{K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1} K(\mathbf{X}, \mathbf{X}_*)} \end{aligned} \quad (4.3)$$

In the development of predictive models using Gaussian Process Regression (GPR), extensive hyperparameter tuning and kernel selection are performed. These factors affect the model's ability to capture non-linear patterns and complexities in data. This ensures a balance is found between model complexity and model generalizability.

Sklearn's GPR offers flexibility in the use of kernels and their combinations. This is beneficial for complex data where a single kernel may not be able to capture the non-linearity seen in the data. In this study, various kernels such as the *RBF*, *Matern*, *DotProduct*, and *RationalQuadratic* kernels were experimented with. Furthermore, combinations of these kernels as a sum-kernel with the *White* kernel (such as *RBF+White*), which models the noise in the data, were explored as well. Additionally, the product of the *RBF* and *DotProduct* kernels, added with a *White* kernel (*RBF * DotProduct + White*), was tested as well to evaluate its ability to capture a broader range of data patterns.

Despite sklearn's implementation of GPR having the capability for automatic noise estimation, a comprehensive evaluation across a spectrum of alpha (α) values, ranging from $1e - 10$ to 10 , was performed in this study. The α

parameter, or the noise level in GPR represents the variance as additional datapoint-dependent noise on the training observations. Adjusting this parameter helps in managing the trade-off between fitting the noise in the data versus capturing the underlying trend. This is equivalent to adding a constant noise to the data. This is slightly different from the use of the White kernel introduced in Section 2.2.2, which is usually combined with another kernel as a sum-kernel and allows for the noise to be learned from the observed data during the model-fitting process as i.i.d. Gaussian noise.

Normalization of the target variable was also tested as a part of the hyperparameter tuning. Normalizing the target variable can be advantageous in certain contexts, especially when the data spans several orders of magnitude. Normalization aids in stabilizing the numerical computations and often leads to better model performance, as it brings all outputs to a similar scale.

The selection of the best hyperparameters was guided by a systematic evaluation using GridSearchCV, another tool in Scikit-learn [23] that automates the process of hyperparameter tuning. It methodically searches through a predefined grid of parameters, evaluating the model’s performance at each point. The grid included the diverse array of kernels and the range of alpha values mentioned earlier.

Validation is performed using 5-fold cross-validation. The performance of each model configuration was assessed with cross-validation based on metrics such as root mean squared error (RMSE), and the coefficient of determination (R^2). These metrics are described further in Section 5.1.

4.5 Adaptive GPRs

So far, in this work, surrogate GP models are constructed using pre-generated offline data through a random sampling strategy with FEM simulations to densely populate the design space, a process that is both time-consuming and computationally expensive. However, it should be possible to train GPRs with a smaller amount of data, which would make the GPR computationally less intensive, by decreasing prediction times (which is proportional to the size of the training data (Equation 2.2)) and training times. An efficient approach would be to adaptively train the GP by iteratively selecting sample points that would minimize a global error estimate using online simulations in a greedy-like fashion.

The methodology used for training an adaptive GP in this thesis is an application and generalization of the adaptive GP approach presented by Semler

et al. [5], where two models were used:

- an accuracy model focusing on choosing the best points in a given iteration based on a greedy heuristic
- a work model that allocates resources to the simulation tool while constrained by the remaining computational budget for improving the accuracy and resolution of the simulations

This thesis only considers the accuracy model for the development of the adaptive GP without integrating the work model. Future work has to be done to integrate the work model which can potentially improve the resolution of simulations when required without sacrificing too much performance.

The formulation of the adaptive GP used in this approach is given as the minimization of the accuracy model $E(\mathcal{D})$ over all possible training data designs D :

$$\min_{\mathcal{D} \in D} E(\mathcal{D}) \quad (4.4)$$

The accuracy model is quantified as the estimate of the parameter reconstruction error over the training data design (\mathcal{D}), which is the measure of the discrepancy between the parameters reconstructed by inverse design using the surrogate model as $p_{\mathcal{D}}(y^m)$, and those obtained from the true forward function (FE simulations) as $p(y^m)$, that is, the term $p_{\mathcal{D}}(y^m) - p(y^m)$. A heuristic like this ensures that the resulting model is not only accurate and well-generalized but also performs well in the context of inverse design. The methodology by Semler et al. [5] provides a framework for this, which this thesis adapts and extends to suit the specific needs of multi-phase ceramics simulations. The accuracy model [5] (or the global reconstruction error) is estimated as:

$$E(\mathcal{D}) \approx \frac{\text{vol}(X)}{N_{MC} - 1} \left(\sum_{i=0}^{N_{MC}-1} w^q(p_i) \epsilon(p_i)^q \right)^{\frac{1}{q}} \quad (4.5)$$

where, p_i is a point in the design space X . N_{MC} is the number of sample points for $1 \leq q < \infty$.

$w(p)$ is the weighing factor that assigns importance to a point p , given as:

$$\tilde{w}(p) := \left\| (f'^T \Sigma_l^{-1} f' + \lambda I)^{-1} f'^T \Sigma_l^{-1} \right\|_2 \quad (4.6)$$

where, f is the forward model, f' is the Jacobian matrix of f at p . Σ_l is the covariance matrix (or kernel matrix), λI is the regularization term with λ being a positive real number and I the identity matrix.

And $\epsilon(p)$ is the local error estimate at point p given as:

$$\epsilon(p) = \text{tr}(\sigma_D(p)^q) \quad (4.7)$$

where the term $\sigma_D(p)$ denotes the standard deviation of the prediction errors at point p in the GP model, and $\text{tr}(\sigma_D(p)^q)$ represents the trace of the GP model variance at a point p .

The surrogate model is constructed iteratively. The set of sampling points are selected using techniques like grid sampling or Latin Hypercube Sampling (LHS) [24]. The error model is then developed by calculating weight factors (Equation 4.6) and local error estimates (Equation 4.7) on these samples. The acquisition function finds the local maximizers of the local error estimate density over the samples which are then chosen as the candidate points for a given iteration. The acquisition function is given as:

$$g(p) := \tilde{w}(p)\epsilon(p) = \tilde{w}(p)\sqrt{\text{tr}(\tilde{\sigma}_D(p)^2)} \quad (4.8)$$

Choosing candidate points that maximize the acquisition function (Equation 4.8) in every iteration in a greedy manner is the basis of optimizing the accuracy model (Equation 4.4 and Equation 4.5). The framework also ensures points that were previously added to the training data are not re-added and the next best candidate is selected. The training ends when the global reconstruction error estimate starts to converge.

An illustration of the local error estimate density function over the parameter space for an example function during an iteration of the adaptive phase can be seen in Figure 4.7 (left). The red crosses represent the initial points which are placed at the boundaries of the design space, the black points indicate points added through the adaptive strategy, and the green point indicates the next candidate point to be added for the current iteration. The contours represent the local reconstruction error estimate densities at different regions in the design space. The acquisition function looks for regions where the error estimate density is high and selects these as candidate points to be added during the iteration, In Figure 4.7 (right), it can be seen that once the candidate point is added, the error estimate density in that region is lowered. This is the core working of the adaptive GP approach.

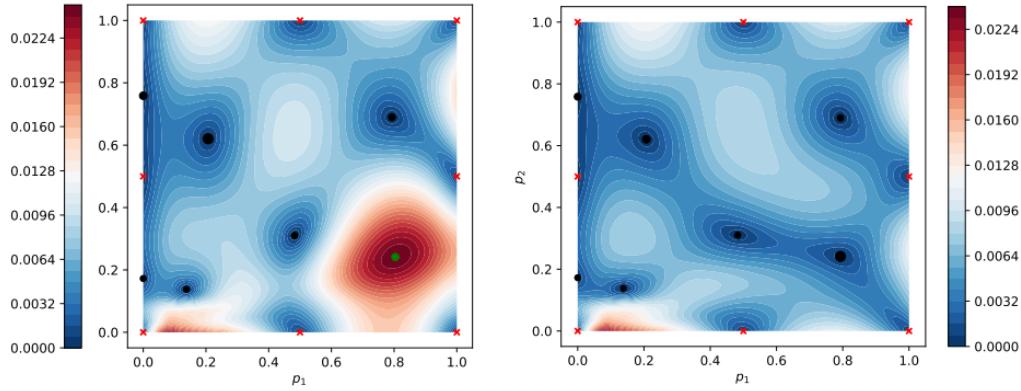


Figure 4.7: Local error estimate density for a parameter space during adaptive training for the function

$$f(p) = (\cos(\phi)(p_1 + p_2) + \sin(\phi)(p_2 - p_1))^2, \quad \text{for } p \in X = [0, 1]^2, \phi \in \mathbb{R}_{>0} [5].$$

For the purpose of this work, adaptive GPs have been implemented with online FEM simulations, where microstructures as requested by the adaptive sampling approach are generated and simulations are run to determine their material properties, in similar steps as described in Section 4.1. However, the generated structures from the simulation programs are not always the exact microstructures as requested microstructures but are close enough to be considered acceptable. In cases where the deviation is too large, for example, if the adaptive sampling approach requests infeasible microstructures, the adaptive sampling approach has been modified in this work to iteratively exclude these microstructures from future sampling. This is done by defining a small radius around the infeasible microstructure and excluding this zone from future sampling as candidates. After repeated attempts, this will eventually result in a well-defined design space sampling that avoids infeasible regions even if they exist within the design space.

The adaptive GP also requires that the variance of the sample be given along with its target (Y) value. This is a form of prior knowledge about known variance in the data that is used by the GP for improving its predictions. For example, if a point is sampled repeatedly through FEM simulations, some minor differences will be seen in the resulting properties of the structure. This variance has to be quantified and provided directly to the GP for each adaptively added sample. It is also used by the work model to decide whether to spend some computational budget from the remaining work budget to improve the variance of the sample. The computational budget here would

correspond to increasing the resolution of the structure generator and FEM simulations which increase the computational load and computational time. For this work, the work model has not been implemented as described and a fixed variance of $1e - 4$ is assigned to all adaptively added points. Determination of the true variance of a sample and implementing the work model would be possible directions for future study.

4.6 Inverse Modeling

Once the GPR surrogate models are trained, the inverse design workflow can be carried out. First, the optimization problem needs to be defined with the objective function that has to be minimized, and then the gradients of this objective function have to be derived. The optimization problem is given as:

$$\begin{aligned} & \underset{\mathbf{x} \in \mathcal{X}}{\text{minimize}} \quad J(\mathbf{x}) \\ & \text{with} \quad J(\mathbf{x}) < \theta \end{aligned} \tag{4.9}$$

where \mathcal{X} is the design space of the input features, in this case, all possible microstructure designs, and θ is the tolerance for the acceptance of the solution, which is required to discard solutions that are part of some unacceptable local minima. In this work, $\theta = 0.01$ was used in most cases.

The objective function $J(\mathbf{x})$ is defined similarly to Equation 2.1, but for ease of differentiability, it is opted to use the squared error instead of absolute differences as follows:

$$J(\mathbf{x}) = (\bar{\mathbf{f}}_* - y^*)^2 \tag{4.10}$$

where $\bar{\mathbf{f}}_*$ is the GPR prediction at microstructural descriptor \mathbf{x} (given by Equation 4.2 with $\mathbf{X}_* = \mathbf{x}$) and y^* is the desired material property. The gradient of this function is then given as:

$$\nabla J(\mathbf{x}) = 2(\bar{\mathbf{f}}_* - y^*)\nabla\bar{\mathbf{f}}_* \tag{4.11}$$

One of the benefits of the GPR is that, along with the predictive mean function (Equation 4.2), it also provides an estimate for the uncertainty at different input points \mathbf{x} using the predictive standard deviation (given by Equation 4.3). This can be incorporated in the objective function to prefer microstructures that are known to exist as part of the training points, to avoid

the suggestion of microstructures that are unknown and possibly unrealistic in practical applications. This modified objective function is formulated as:

$$J(\mathbf{x}) = (\bar{\mathbf{f}}_* - y^*)^2 + \text{std}(\mathbf{f}_*) * s \quad (4.12)$$

and the gradient of this modified objective function is given as:

$$\nabla J(\mathbf{x}) = 2(\bar{\mathbf{f}}_* - y^*)\nabla\bar{\mathbf{f}}_* + \nabla\text{std}(\mathbf{f}_*) * s \quad (4.13)$$

where, $\text{std}(\mathbf{f}_*)$ is the predictive standard deviation function defined in Equation 4.3 (with $\mathbf{X}_* = \mathbf{x}$) and s is a scaling factor to allow a level of generalizability in the optimizations. A high scaling factor will rescale the objective function to find local minima only around regions where known data is available, which can be considered as overfitting for the inverse models, and a very small scaling factor will have no effect and can suggest unrealistic microstructures. In this work, $s = 0.01$ was used in most cases.

To, obtain the gradient of the objective functions (Equations 4.11 and 4.13), the gradient of the predictive mean and the predictive standard deviation has to be derived. This gradient of the predictive mean function $\nabla\bar{\mathbf{f}}_*$ can be obtained by partially differentiating Equation 4.2 (with respect to \mathbf{X}_*) to obtain:

$$\nabla\bar{\mathbf{f}}_* = \nabla K(\mathbf{X}_*, \mathbf{X}) [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{y} \quad (4.14)$$

The gradient of predictive standard deviation $\nabla\text{std}(\mathbf{f}_*) * s$ is derived by partially differentiating Equation 4.3 (with respect to \mathbf{X}_*) to obtain:

$$\nabla\text{std}(\mathbf{f}_*) = -\frac{1}{\text{std}(\mathbf{f}_*)} K(\mathbf{X}_*, \mathbf{X}) [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1} \nabla K(\mathbf{X}_*, \mathbf{X})^T \mathbf{y} \quad (4.15)$$

This means that the GPR prediction function's gradient can be computed if the employed kernel function's gradient can be computed. Since multiple kernels are explored in this study, the gradients are briefly stated for each of them. For simplification, length scale parameters are set to 1.

Gradient of RBF kernel (Equation 2.4):

$$\nabla K_{\text{RBF}}(\mathbf{x}, \mathbf{X}) = -\exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{X}\|^2\right) (\mathbf{x} - \mathbf{X}) \quad (4.16)$$

Gradient of Matérn kernel for $\nu = 1.5$ (Equation 2.5):

$$\nabla K_{\text{Matern}}^{1.5}(\mathbf{x}, \mathbf{X}) = -3 \exp\left(-\sqrt{3}\|\mathbf{x} - \mathbf{X}\|\right) (\mathbf{x} - \mathbf{X}) \quad (4.17)$$

Gradient of White kernel (Equation 2.6):

Since White Kernel (Equation 2.6) can be considered constant, its gradient is always 0 given as

$$\nabla K_{\text{White}}(\mathbf{x}, \mathbf{X}) = 0 \quad (4.18)$$

Gradient of Dot Product kernel (Equation 2.7):

$$\nabla K_{\text{DotProduct}}(\mathbf{x}, \mathbf{X}) = \mathbf{X} \quad (4.19)$$

Gradient of Rational Quadratic kernel (Equation 2.8):

$$\nabla K_{\text{RatQuad}}(\mathbf{x}, \mathbf{X}) = -\left(1 + \frac{\|\mathbf{x} - \mathbf{X}\|^2}{2\alpha}\right)^{-\alpha-1} (\mathbf{x} - \mathbf{X}) \quad (4.20)$$

It should be noted that in cases where a combination of kernels as sum-kernels or product kernels are used as described in Section 4.4, the gradients follow the sum-rule and product-rule of differentiation respectively.

These gradients are pre-implemented with skopt [22] in its wrapper for sklearn's [23] implementation of GP, which was used in this work. The correctness of both the objective function gradients (Equations 4.11 and 4.13) using different implemented kernels was confirmed using a 2-point finite differences approximation at various points in the design space, which returned an average error of 1e-06 across most kernels and material properties.

To overcome the limitations of gradient-based optimizers getting stuck in local optima, which is often due to the choice of the starting point, multiple solutions are generated such that each solution corresponds to a unique minimum, and the solutions that satisfy $J(x) < \theta (= 0.01)$ (Equation 4.9) are chosen as part of the solutions. All possible microstructure design solutions can be presented for practical applications, as used in qualitative and quantitative analysis in this work. To obtain multiple solutions, optimizations were performed using multiple starting points, known traditionally as multi-start optimization (MSO). The multiple starting points were chosen from the existing data using points that are closest to an initial LHS sampling of the design space, to ensure uniformly spaced and realistic data points.

With the objective function and the gradients defined, the optimization problem now can be minimized for inverse modeling. This is done using optimization routines. Optimization routines aim to minimize (or maximize) an objective function iteratively. For the choice of the optimization method, the work by Varoquaux [25] in the Scipy Lecture Series recommends BFGS and L-BFGS-B [26] in cases where gradients are both known and unknown (substituted with approximated gradients) for gradient-based optimizers. If the problem is well-conditioned (small changes in inputs only result in small changes in output), then the Powell [27] or Nelder-Mead [28], which are both gradient-free techniques, can be employed. The inverse design problem also requires bounded-box constraints to be defined to ensure the optimizer only looks for microstructures within known ranges. This is not natively supported by BFGS, and for this reason, the L-BFGS-B method and another method SLSQP [29] are considered in this work for gradient-based optimizers. The Powell and Nelder-Mead methods are also presented for comparison as the gradient-free technique to emphasize the power of gradient-based optimizers.

4.7 Direct Inverse Training

Direct Inverse Training (DIT) presents an alternative approach to tackling inverse design problems. The methodology used so far trains a surrogate model that predicts outputs based on given inputs, which is used to find optimal inputs through optimization routines. However, DIT flips this concept, treating labels (desired outcomes) as input features and learning to predict the original inputs. This way one can realize direct predictions for the inverse problem, completely skipping the formulation of an objective function and iterative optimization.

While it's not widely used as an inverse design methodology, the purpose of its implementation is to explore whether it is possible to train models directly in the reverse direction and how the performance compares to the surrogate-based inverse design methodology. In theory, reverse models of this manner would be able to make instantaneous predictions directly for optimal design.

So far, with forward modeling with surrogates, the task was modeled using single-output regression where microstructure descriptors are considered as features and a material property (thermal conductivity, thermal expansion, etc) is considered as the target. With DIT, this changes to a multi-output regression task where now material properties (Y) are treated as features and the microstructural descriptors (X) are treated as targets, that is, to directly train an inverse function $f : Y \rightarrow X$. The data is similarly preprocessed and

normalized for effective training using the preparation and pre-processing techniques described in Sections 4.1 and 4.2. Different models such as the Multilayer Perceptron (MLP), GPR, and Random Forests (RF) Regressors are then trained with inverse mapping, i.e., from properties to structural features.

The NN employed here was a 4-layer Multi-layer Perceptron (MLP) with two hidden layers of size (100, 100). The GP employed used the *RBF + White* kernel. The GP employed here should not be confused with the GP work discussed so far with the forward modeling of the surrogate function, as this is strictly for DIT. And finally, the RF was employed with default settings as defined in the sklearn library. These models are chosen due to their inherent nature of supporting multi-target regression. The research question for this study focuses on answering how the optimization-based inverse design methodology, using GPR as surrogates, compares to directly training ML models in reverse with DIT.

Chapter 5

Results and Discussion

This chapter will examine and discuss the results of the application of methods described in Chapter 4.

5.1 GPR Training

Regression Metrics

R-squared (R^2)

R-squared (or R^2), known as the coefficient of determination, is a metric in assessing the performance of regression models. It quantifies how well the model's predictions fit the actual data. R^2 is the proportion of the variance in the dependent variable that is predictable from the independent variables. It is expressed as:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (5.1)$$

where, n is the number of values, y_i are the actual values, \hat{y}_i are the values predicted by the model, \bar{y} is the observed mean of actual values.

R^2 values of 1 imply perfect predictions and a value below 0 indicates that the model performs worse than the mean function of the dependent variables.

Root Mean Squared Error (RMSE)

RMSE measures the differences between values predicted by a model and the actual values. It provides an indication of the model's prediction accuracy on the scale of the actual data and tells how much the model's prediction

deviates from the actual data on average. A low RMSE value therefore means better model performance. RMSE is given as the square root of the mean squared error:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (5.2)$$

In the context of building surrogate models with regression, these metrics provide an interpretation of the model's predictive ability. R^2 suggests the overall fit of the model, and RMSE provides a scale-relevant measure of prediction accuracy.

Baseline Performance

To establish baseline performance of the surrogate models, this work reproduced the results of the work by Pirkelmann et al. [8]. The GBR was claimed to have performed the best out of many regression models. For these initial results, the default kernels and hyperparameter settings with the sklearn library [23] were used to replicate the same results as the work by Pirkelmann et al. [8]. However, for the sake of avoiding redundancy, the initial full feature set was ignored and the volume fraction of alumina was excluded from the features, and the 8-feature set described in Section 4.2 was used. The omitted results for the full feature set are comparable to the 8-feature set.

The tabulated results can be seen in Table 5.1. The scores are computed using 5 cross-validation splits in both cases (GBR and GPR). The default kernel and hyperparameter settings for GPR correspond to an *RBF* kernel with *lengthscale* = 1 and noise term $\alpha = 1e - 10$. Unsurprisingly, the GPR with these settings performs poorly on the multi-phase ceramics data when compared to the GBR.

The GBR achieves a near-perfect R^2 fit on the validation data on all properties (≈ 1) with significantly small RMSEs and MSEs. This is the baseline performance that was hoped to achieve with the GPR. However, the formulation of an effective GPR requires care in choosing the right kernel that can represent the non-linearity (or linearity) in the given data. Therefore, the kernels were treated as a hyperparameter to be optimized through the grid search described in Section 4.4.

Property	R^2	RMSE
Gradient Boosting Regressor		
CTC	0.997	4.28e-01
CTE	0.999	5.08e-03
Young's Modulus	0.999	1.3e+00
Poisson Ratio	0.999	3.87e-04
Gaussian Process Regressor		
CTC	0.64	4.84e+00
CTE	0.561	2.82e-01
Young's Modulus	0.901	1.41e+01
Poisson Ratio	0.862	9.07e-03

Table 5.1: Comparison of GBR and GPR Performance Metrics for 8-feature set.

Hyperparameter grid-search

For the purpose of the grid-search, multiple kernels, noise levels (α), and the effects of normalizing the target variable for each feature set were explored. To keep the discussion concise, the results of the grid search across the kernels and noise levels are first presented with the normalization of the target variables which was not done in the baseline results. Then the effect of not normalizing the targets for the material properties is discussed.

The R^2 scores of each component of the grid search can be seen in Figure 5.1. These R^2 scores are calculated by the average of the R^2 score of the 5 cross-validation splits over each combination (Kernel, Noise Level) of the grid. A big improvement from the default GPR model can be seen already. For CTC (thermal conductivity), the *Matern+White*, *RBF+White*, *RelationalQuadratic+White* and *RBF*DotProduct+White* perform the best over a wide range of noise levels ($1e-10$ to 0.001) with an R^2 score of 0.9928. Though this best score is slightly worse than the baseline GBR model's R^2 score of 0.997 for CTC (Table 5.1), it is sufficiently close to be considered acceptable, given the compatibility of GPRs for use in inverse design. The *RBF* kernel and *Matern* kernel perform very poorly for very low noise levels ($1e-10$ to $1e-05$) with an R^2 score of 0.1 and 0.4 respectively.

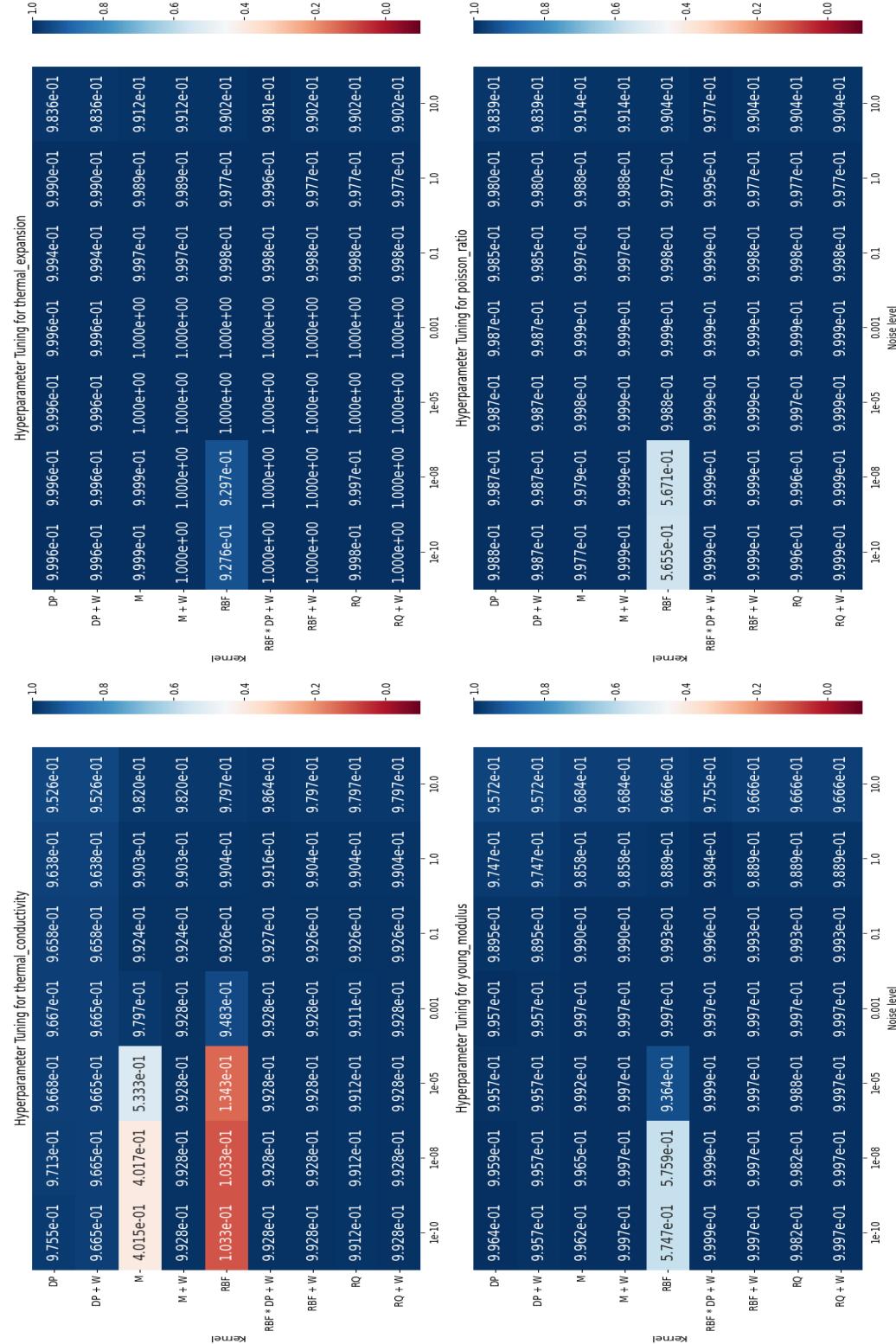


Figure 5.1: R^2 scores for each combination of the grid-search; Kernels (Y-axis): RBF, M: Matern, DP: Dot Product, RQ: Relational Quadratic, W: White kernel, and Noise Levels (X-axis) between $1e - 10$ to 10 .

For CTE (thermal expansion), perfect fits ($R^2 = 1$) are seen for a wide range of kernels and noise levels, which is a small improvement over the baseline GBR model's R^2 score of 0.999 for CTE as shown in Table 5.1. Only the *DotProduct* and *DotProduct + White* kernels don't achieve a perfect R^2 fit. The kernels without a *White* kernel only achieve perfect fits for noise levels between 1e-05 and 0.001. Overall, this suggests a lower variance in the data and less complex non-linearity.

For Young's modulus, the *RBF * DotProduct + White* performs the best with an R^2 of 0.9999 across all noise levels. A good effect is also seen by adding a *White* kernel. In its absence, the kernels always have a lower score than their *White* sum-kernel counterpart. This suggests the presence of some noise in the data that can only be fit using a *White* kernel. Adding a *White* kernel results in high R^2 scores of 0.9997 over a wide range of noise levels. This score is a very minute improvement over the baseline GBR model's R^2 score of 0.999 for Young's modulus (Table 5.1).

For Poisson ratio, most of the kernel tuples perform rather well, except the *RBF* kernel at very low noise levels with R^2 scores of 0.5655. However, in general, either a *White* Kernel that can model noise inherently or a higher value of noise level is required to achieve the best possible R^2 scores for Poisson ratio of 0.9999. This score is similar to the baseline GBR model's R^2 score of 0.999 for Poisson ratio (Table 5.1). Here, as well, the *DotProduct* kernel with and without an added *White* component does not realize this highest possible R^2 score.

These scores are comparable to the baseline GBR model shown in Table 5.1. The GPR, therefore, is shown to be able to achieve similar, if not better scores to the GBR model after an extensive grid search for kernels and hyperparameters for all properties.

The biggest takeaway here is that in general, if one can observe inherent noise or variance in the data, which was seen for all properties in Section 4.3, adding a *White* kernel is the simplest way to address this to achieve an effective GPR model. Tuning the noise level α can help to an extent, but this is a fixed and constant value of noise that is added to every point in the training set primarily used for numerical stability, ensuring that the kernel matrix is positive definite. But for modeling inherent noise in the data, the use of a *White* Kernel is far more suitable.

Additionally, the *RBF * DotProduct + White* Kernel is the most complex kernel that was tested, which performs well for all properties as expected, but it remains to be seen whether it's viable to trade off computational speed

in favor of using a complex kernel for high accuracy. On the other hand, the *DotProduct* kernel fails to reach the level of accuracy that can be modeled by other kernels, although usually only by a small margin in the range of 1e-02. An argument can be made for its use as well in terms of its trade-off in losing some accuracy if the exhibited computational speed is significantly higher. These aspects will be explored in the context of inverse design in ceramics in further sections.

An important consideration to factor when formalizing an effective GPR model is whether the targets require normalization. It was chosen to normalize the target variables for all properties for consistency in the framework as described in Section 4.4, though, it is generally only required for target variables that have parameter ranges in the order of 100 and above. As described earlier in Section 4.3, Young's modulus in the data varies from 205.51 to 387. In Figure 5.2, it can be seen that in the absence of normalized targets for Young's Modulus, all kernels with an added *White* component perform poorly, with most R^2 scores being negative. It can be concluded that normalization was required for Young's modulus to be modeled effectively by the GPR. For other material properties, normalizing the targets did not make a major difference but it also did not reduce their accuracy.

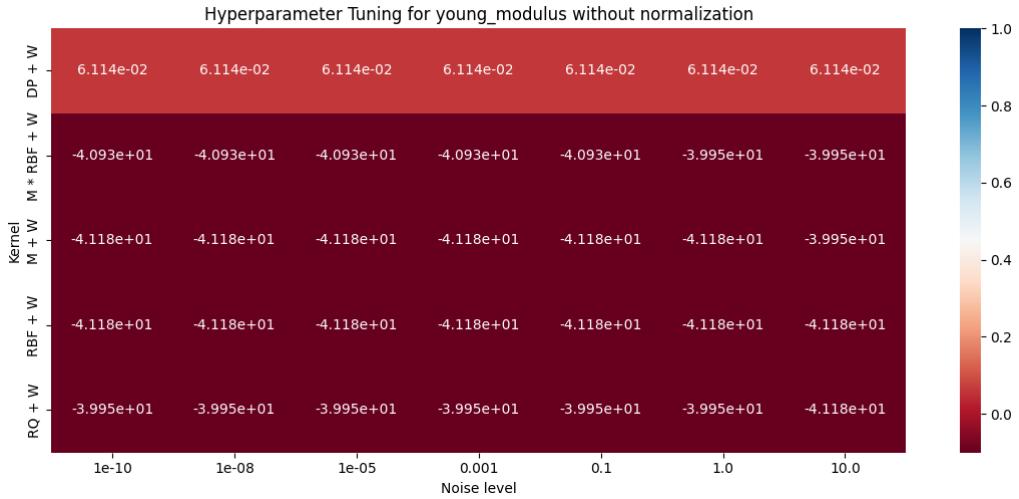


Figure 5.2: Grid Search results for Young's Modulus without normalization.

Feature set analysis

So far an initial baseline model based on most of the features present that describe the microstructures has been shown. A primary challenge was whether

the dimensionality of the design space can be reduced which can potentially ease computation times in the inverse design step without sacrificing regression accuracy.

An extensive analysis of the R^2 scores of the different feature sets using different kernels is illustrated in Figure 5.3. This analysis only considers kernels added with a *White* component, which is now known to be sufficient for achieving high accuracy through the grid search in the previous analysis.

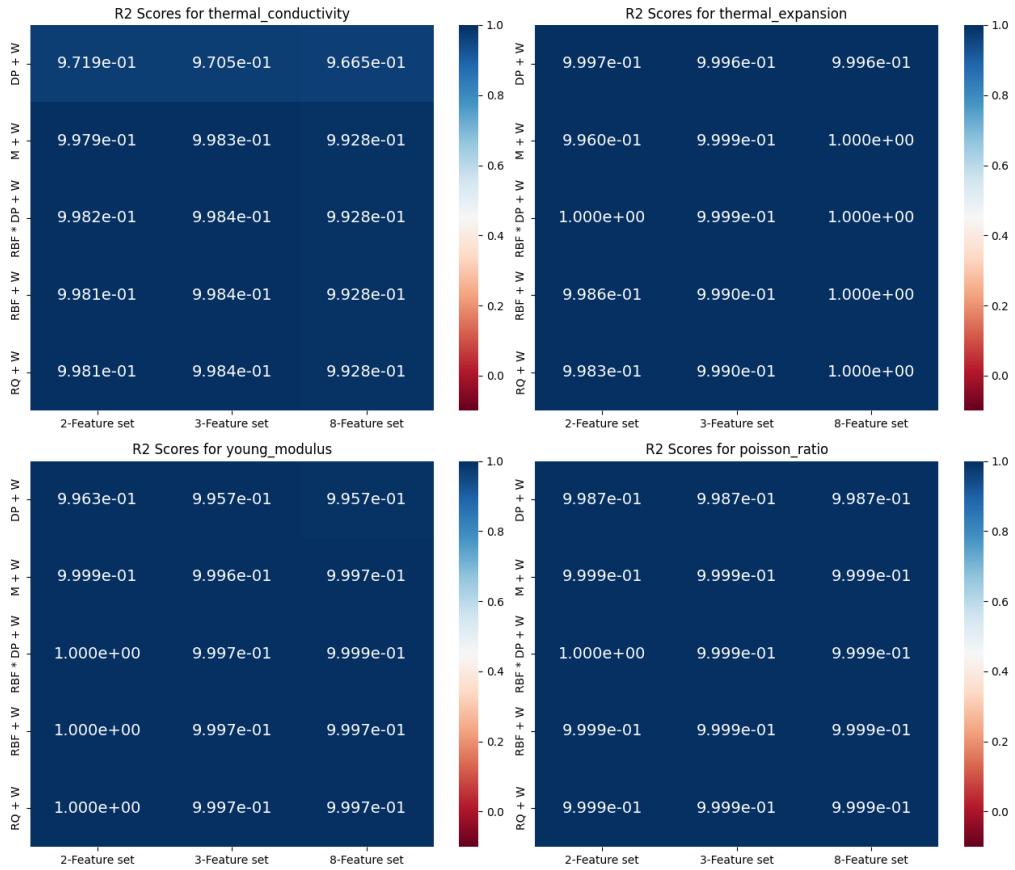


Figure 5.3: R^2 scores of the different feature sets.

In CTC, for the most part, a minute increase of ≈ 0.005 in R^2 score can be observed in the 3-feature set and 2-feature set in most kernels compared to the 8-feature set.

In CTE, the model accuracy goes down with the smaller feature sets for most kernels except for the *RBF * DotProduct + White* kernel in the 2-feature set. For the remaining kernels, while the R^2 scores for the 3-feature set and

2-feature set do not exhibit a perfect fit like the 8-feature set, the difference is very negligible between $0.0001 - 0.0013$. It can be suspected that this is due to the increased presence of some null values in the CTE data and when the 2-feature set is constructed, there is even lesser data available for training as shown in Section 4.3.

In Young's modulus, the 8-feature set and 3-feature set have similar accuracies, but the 2-feature set starts to see perfect fits in the *RBF + White*, *RBF * DotProduct + White* and *RationalQuadratic + White*. This can be explained due to the decreased variance in material property targets when considering the 2-feature sets where the porosity is equal to zero. In the case of Young's modulus, this is especially exaggerated where the variance much lower (as seen in Figure 4.5), due to which the GPR is able to achieve an $R^2 = 1$.

In Poisson ratio, the R^2 scores are similar across all kernels and feature sets, where it shows an R^2 score of 0.9999. With the *RBF * DotProduct + White*, a perfect fit is seen.

From the results, it can be seen that the accuracy either mostly remains the same across feature sets for every kernel, or improves slightly. This confirms the premise that the material properties for multi-phase ceramics can be represented by a smaller set of microstructural features without sacrificing much accuracy.

The *RBF * DotProduct + White* Kernel seems to have a small edge in accuracy in most material properties. While this is true, its increased complexity might be unsuitable for inverse design and remains to be seen in experiments in upcoming sections.

For the most part, the 8-feature set can be substituted with the 3-feature set for all properties without loss of generality. The 2-feature set, however, is a special case where the porosity is absent. For theoretical purposes, with its ease of visualization and fast computational speed, the 2-feature set will be considered in upcoming experiments. This sets the groundwork for future work in higher dimensional inverse modeling for multi-phase ceramics.

Recomparison to Baseline

Finally, to generalize the results of the extensive grid-search and feature set analysis and compare the GPR results to the baseline GBR, an *RBF + White* kernel is used. Here, both the models are trained on the 2-feature sets. The results, seen in Table 5.2, show that the scores are similar if not better

for all material properties in terms of both the R^2 and the RMSE. This validates the choice of GPRs as surrogate models as they can, with thorough hyperparameter testing, outperform most ML models for this purpose (since the GBR outperformed most ML models [8]).

Property	R^2	RMSE
Gradient Boosting Regressor		
CTC	0.997	3.26e-01
CTE	0.99	7.29e-03
Young's Modulus	1.0	4.98e-01
Poisson Ratio	1.0	2.53e-04
Gaussian Process Regressor		
CTC	0.998	2.42e-01
CTE	0.99	9.07e-04
Young's Modulus	1.0	1.97e-01
Poisson Ratio	1.0	1.06e-04

Table 5.2: Comparison of GBR and GPR Performance Metrics for 2-feature set.

5.2 Inverse Modeling

In this section, the results of the inverse microstructure design methodology using the GPR surrogate models and an analysis of the effectiveness of the prescribed methods and related metrics are presented.

Inverse Modeling Metrics

All tests for this section were performed using the 2-feature set for simplicity. A comparison of the different optimizers is done in upcoming sections, but for the purpose of this case study, the L-BFGS-B optimizer [26] was used as the optimization routine here. The optimizer uses bounded-box constraints set to the parameter ranges seen in the ceramics data. This ensures the optimization does not explore regions of the parameter space that are too different from known data. The main advantage of using GPRs in inverse design was that the gradients for the surrogate model-based objective function can be mathematically derived as described in Section 4.6. The objective function $J(x)$ used in these initial results is given by Equation 4.10.

As portrayed earlier in Section 2.1, inverse design has to find microstructures x^* using a surrogate function $f(x) = y$ for a given desired property

value $y = y^*$. For a given material property p , belonging to one of the four material properties {CTE, CTC, Young's Modulus, Poisson Ratio}, the given surrogate function is $f_p(x) = y$. To evaluate the performance of the inverse design, 20 property values y_i^* ($1 \leq i \leq 20$) evenly distributed over the range $[y_{min}, y_{max}]$, are evaluated through the inverse design process. This process then seeks to minimize the objective function $J(x)$ for each y_i^* , to find their optimal microstructures x_i^* . For each y_i^* , the optimizer yields a set of optimal microstructures, each corresponding to different starting points. These microstructures are then re-evaluated using the surrogate model to quantify the error between the desired property value y_i^* and obtained property value $f_p(x_i^*)$ as $\epsilon_i = f_p(x_i^*) - y_i^*$ for error metrics such as R^2 and RMSE. These error metrics are performed over the entire range of property values to quantify an overall error for the inverse design methodology. To avoid confusion with regression metrics, they will be referred to as Reconstruction R-squared RR^2 , and Reconstruction Root Mean Squared error (RRMSE). The RR^2 in this sense suggests a measure (with 1 being a perfect fit) of whether the solution microstructures truly exhibit the desired property value (from the perspective of the surrogate model) and the RRMSE quantifies in the units of the property how close the exhibited property value is to the desired value. Additionally, qualitative analysis via visualization of the objective function, projection of optimized microstructures over existing data, etc., can help to visualize any inconsistencies and limitations of the surrogate models.

5.2.1 Initial Results with Gradient-Based Optimizer: A case study

For this early analysis, GPR surrogates using *RBF + White* kernel and noise level of 1e-10 were constructed for all material properties.

Property	RR^2	RRMSE
CTC	0.999	1e-02
CTE	0.999	1e-04
Young's Modulus	0.999	2e-02
Poisson Ratio	0.999	1e-04

Table 5.3: Inverse Design Performance Metrics with GPR surrogate model.

The advantage of using the 2-feature set for these optimizations is that the microstructure design space can be visualized with the material properties in the resulting 3D space. Visualizations in this section take advantage of this to show the fit of the observed optimized microstructures from inverse

design over the known ground truth data, i.e., training data, in terms of the volume fraction of Zirconia (since it is the most correlated feature for all material properties) as a 2D plot (such as Figure 5.4a), and in terms of the volume fraction and particle size ratio as a 3D plot (such as Figure 5.4b). The shaded plane in the 3D plots also shows the GPR prediction function ($\tilde{\mathbf{f}}_*$) plane, which is the plane that models the forward function using the GPR surrogate model (trained in Section 5.1), which is naturally also the plane on which the ground truth data and the optimized microstructures should lie. The rationale behind these visualizations is to see whether the suggested microstructures align within known ranges for volume fraction from the training data and whether they generalize well for unseen particle size ratios. The latter means that for all material properties, if multiple solutions are requested for a single property value, they would correspond to microstructures with different values of particle sizes but with only very small variations in volume fraction, which upon further qualitative analysis by domain experts, would allow ideal microstructures to be identified.

As the data exploration from Section 4.3 suggested, CTE and Poisson's ratio for ZTA/ATZ ceramics are positively correlated to the volume fraction of Zirconia, with multiple solutions corresponding to different particle size ratios with small changes in volume fraction. This should make it a rather simple task for inverse design.

Using the quantitative analysis described in this section, Table 5.3 shows the error metrics for the inverse modeling optimizations for CTE and Poisson's ratio. The RR^2 fit shown for both models is very high (≈ 1), showing that the optimizations work as intended and suggest valid microstructures for obtaining desired material properties, from the perspective of the surrogate model. The RRMSE is in the order of 1e-04 for both CTE and Poisson's ratio respectively, which is a very acceptable amount of error for their material property ranges of [5.4183, 7.0143] and [0.228, 0.322] respectively (described in 4.3).

Using the qualitative analysis described in this section, Figure 5.4a shows the fit of the observed optimized microstructures over the known ground truth data (training data points) in the 2D space for CTE. Here, it can be seen that for CTE, due to low variance, most of the optimized structures are tightly within the volume fraction ranges seen in the ground truth data. Figure 5.4b shows the fit of the observed optimized microstructures over the ground truth in the 3D space for CTE. The multiple solutions can be seen to generalize well for unseen particle size ratios along the prediction plane. For the most part, the inverse design for CTE using GPRs as surrogates is shown to be

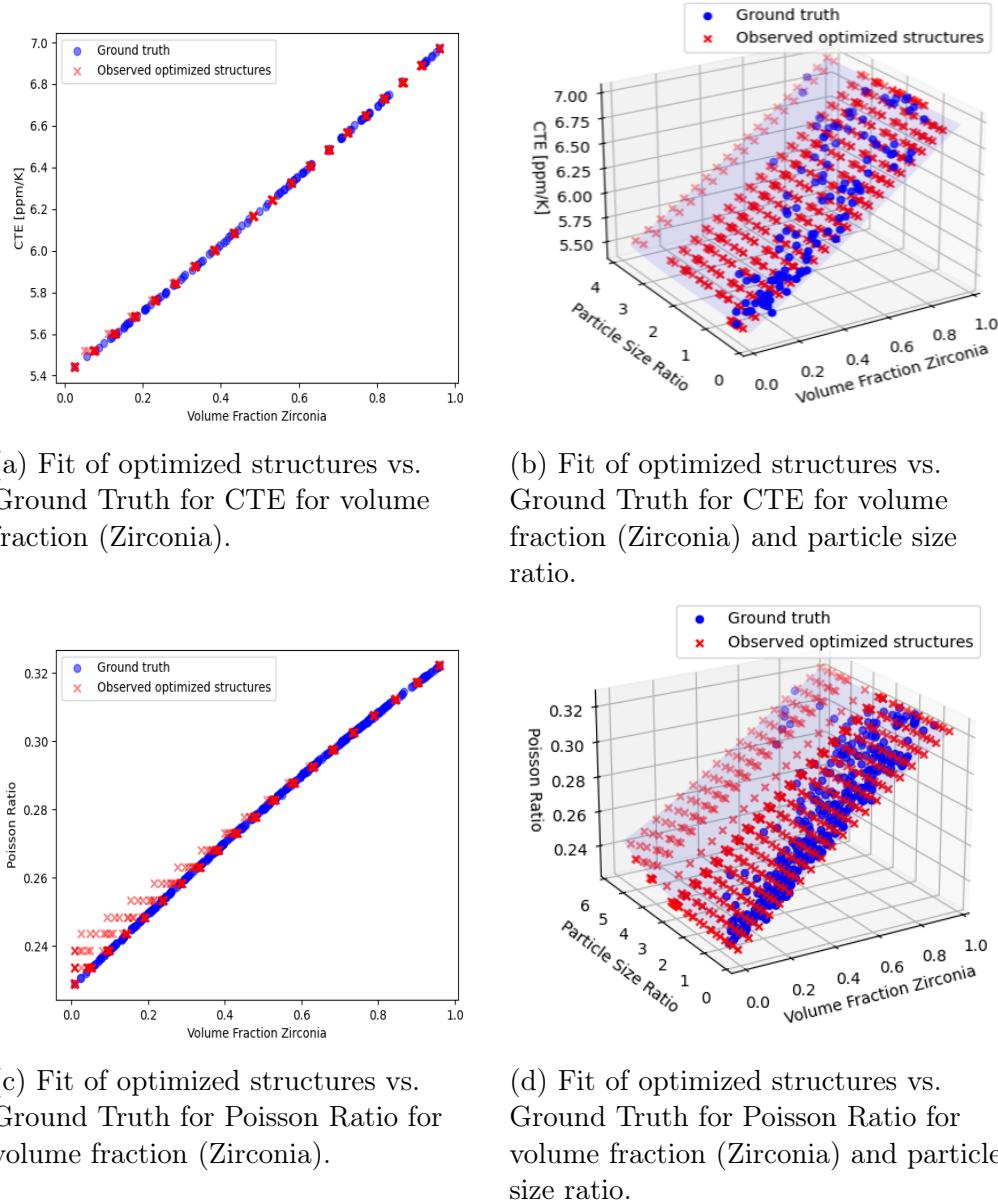


Figure 5.4: Visualisation of optimized microstructures for CTE and Poisson Ratio. The shaded region indicates the GPR surrogate prediction function (\hat{f}_*) plane over the design space, blue points are known real data (also the training data), and red crosses show optimized structures.

promising.

For Poisson's ratio, as seen in Figure 5.4c, a lot of the optimized microstructures suggested do not fit well with the ground truth with known volume fractions, unlike the optimizations for CTE where most optimized structures were within the bounds of volume fraction. In the 3D plot for Poisson's ratio (Figure 5.4d), the multiple solutions returned for each value of Poisson's ratio appear valid, but it can be seen that the shaded GPR prediction surface does not generalize well in the bottom-left corner, where there is no known training data (blue points). The region seems to fold over slightly and mispredicts the correlation against the volume fraction. This is a limitation of the GPR and will be elaborated further.

On the other hand, CTC and Young's Modulus for ZTA/ATZ ceramics are inversely correlated to the volume fraction of Zirconia. The variances for CTC are explained by particle size ratios and volume fraction, and multiple solutions can see larger variances in both the features. For Young's Modulus, multiple solutions will correspond to different values of particle size ratios with small variances in volume fraction similar to CTE and Poisson Ratio.

The quantitative metrics in Table 5.3 show that the RR^2 fit is also very high (≈ 1) for both CTC and Young's Modulus, with an RRMSE of 1e-02 for CTC which is acceptable for its range ([3.016, 34.138]) and very low RRMSE in the order of 2e-02 for Young's Modulus, which is surprising since it has a very high range ([205.51, 387]). This shows that the inverse design methodology is robust even when dealing with material properties with high numerical ranges.

Figures 5.5a and 5.5c show the fit of the observed optimized microstructures over the known ground truth data in terms of the volume fraction of Zirconia for CTC and Young's Modulus respectively. It can already be seen that some optimized microstructures suggested appear as outliers when compared to the ground truth. This is unexpected (similarly to Poisson's ratio) as a high RR^2 score was witnessed for inverse design in these properties. However, when looking at Figure 5.5b and 5.5d, it can be seen that the GPR prediction surface for these models do not generalize well either (top-right corner) where there is no known training data (blue points).

These regions with missing data as seen in Poisson Ratio, CTC, and Young's Modulus are regions of infeasible microstructures, notably microstructures with very low volume fractions of Zirconia and high particle size ratios. These structures are not very realistic, and thus the training data in these regions is naturally missing. The GPR, by virtue of being a prediction model, must

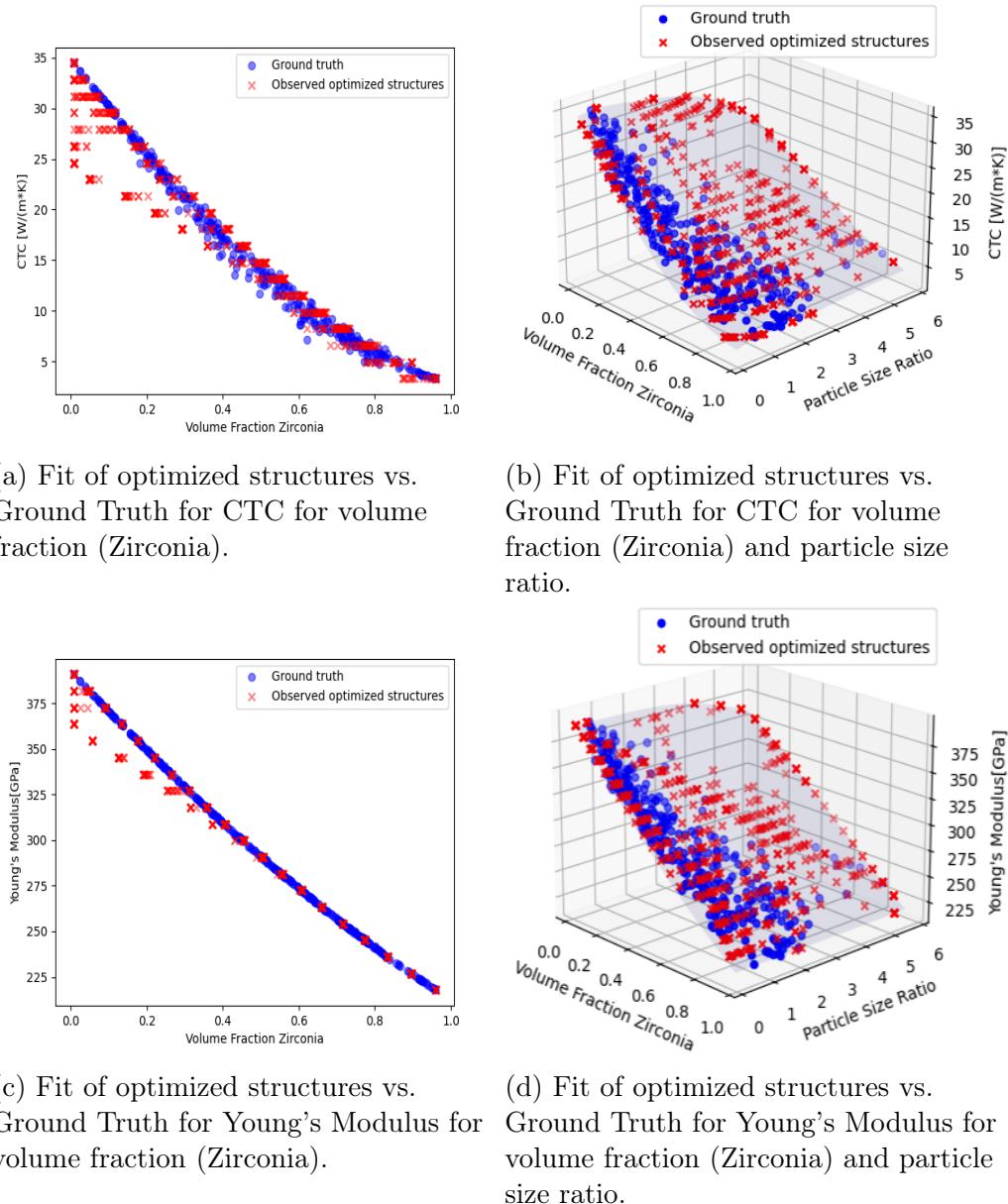
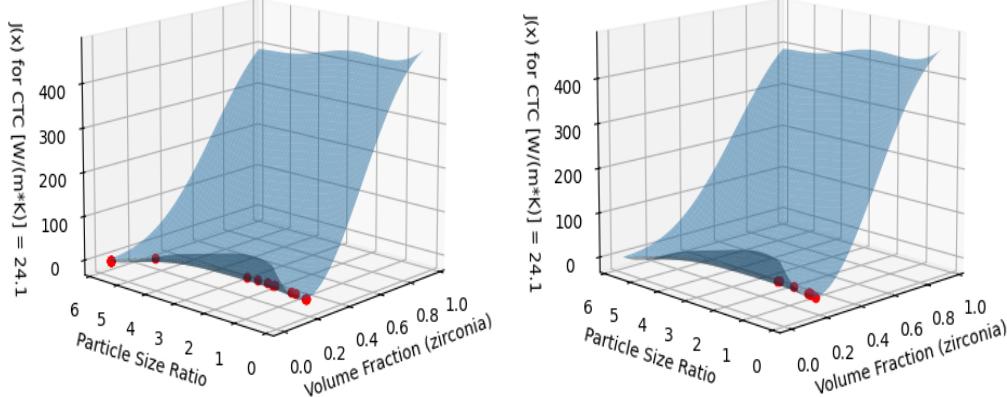


Figure 5.5: Visualisation of optimized microstructures for CTC and Young's Modulus.



(a) For outlier CTC = 24.1 W/m*K with standard objective function (equation 4.10).

(b) For outlier CTC = 24.1 W/m*K with modified objective function (equation 4.12).

Figure 5.6: Objective function for CTC; red points indicate local minima found through multi-starts.

make predictions for the given design space but fails to capture the consistency of the correlation between the material property and volume fraction in this infeasible region. This explains why the RR^2 score and RRMSE appear good, because in reality, due to the limitation in generalizability, the GPR surrogate predicts incorrect material property values for some regions, and this surrogate is also used to validate the error metrics. More precisely, the defined error $\epsilon_i = f_p(x_i^*) - y_i^*$ uses $f_p(x_i^*)$, which is the GPR surrogate function, instead of the true forward function (FEM simulations) as it would be too computationally expensive for validation. The error metrics only show how well the optimized structures are in perspective of the surrogate function $f_p(x_i^*)$, and is a good starting point for validation, but for a full characterization, visualizing the optimizations in this manner is helpful. In Figure 5.5b and 5.5d, the prediction surface in these regions (top-right corners) can be seen to fold over, showing lower property values than expected for very low volume fractions of Zirconia. These outliers are not within typical tolerance limits to be acceptable.

A look at the objective function for one of the outliers (CTC = 24.1 W/m*K) in Figure 5.6a shows that the optimizer reaches a local minimum in the infeasible region of high particle size ratio and low volume fraction near the far bottom-left corner of the graph. From a known perspective, typically, it should only find the local minima in the trough that can be located roughly

around a volume fraction of 0.24. The infeasible region should normally represent a full crest along the length of the range of particle size ratio in the objective function, but due to the limitation of the GPR model, this region drops and appears flat, and as a consequence includes a local minima in the objective function for some of the CTC values.

To address this, the unwanted valleys in the objective function have to be raised by modifying the objective function. The modified objective function described in Equation 4.12 now includes the uncertainty from the GPR model. Since the amount of training data in the infeasible regions is low, the uncertainty of predictions in these regions must be higher as well. This fact is exploited to modify the objective function and raise the unwanted local minima.

Various values of scaling factor s in the modified objective function (Equation 4.12) were tested ranging from 1 to $1e - 20$. A value of 1 will limit the solutions of the optimization to known data too harshly with less room for generalizability and a value of $1e - 20$ does not raise the regions of unwanted local minima. Good generalizability was observed at $s = 0.01$, and the modified objective function for the outlier observed from earlier (Figure 5.6a) can now be seen in Figure 5.6b. The local minima from the infeasible region now disappear and are not considered a part of the solutions of the optimizer.

With the modified objective function, the fit of the optimized microstructures now aligns better with the ground truth with no outliers for CTC and Young's Modulus as seen in Figure 5.7a and 5.7c respectively. The multiple solutions lie within the variance observed in training data as well. In the 3D visualization in Figure 5.7b and 5.7d, it can be seen that the suggested microstructures now avoid the infeasible region (top right corner) where the training data density is lower, indicating higher uncertainty for these regions as hypothesized.

The effect of the modified objective function on CTE and Poisson's Ratio can be seen in 2D and 3D plots in Figure 5.8. These models now show tighter fits to the ground truth and maintain the known correlation between the material properties and the volume fraction similar to CTC and Young's Modulus.

In all cases, while the regions with infeasible microstructures are avoided, the inverse design with the modified objective function still maintains a good level of exploration to unseen microstructures from training data, showing acceptable generalizability in inverse design.

The performance metrics are very similar for the modified objective function

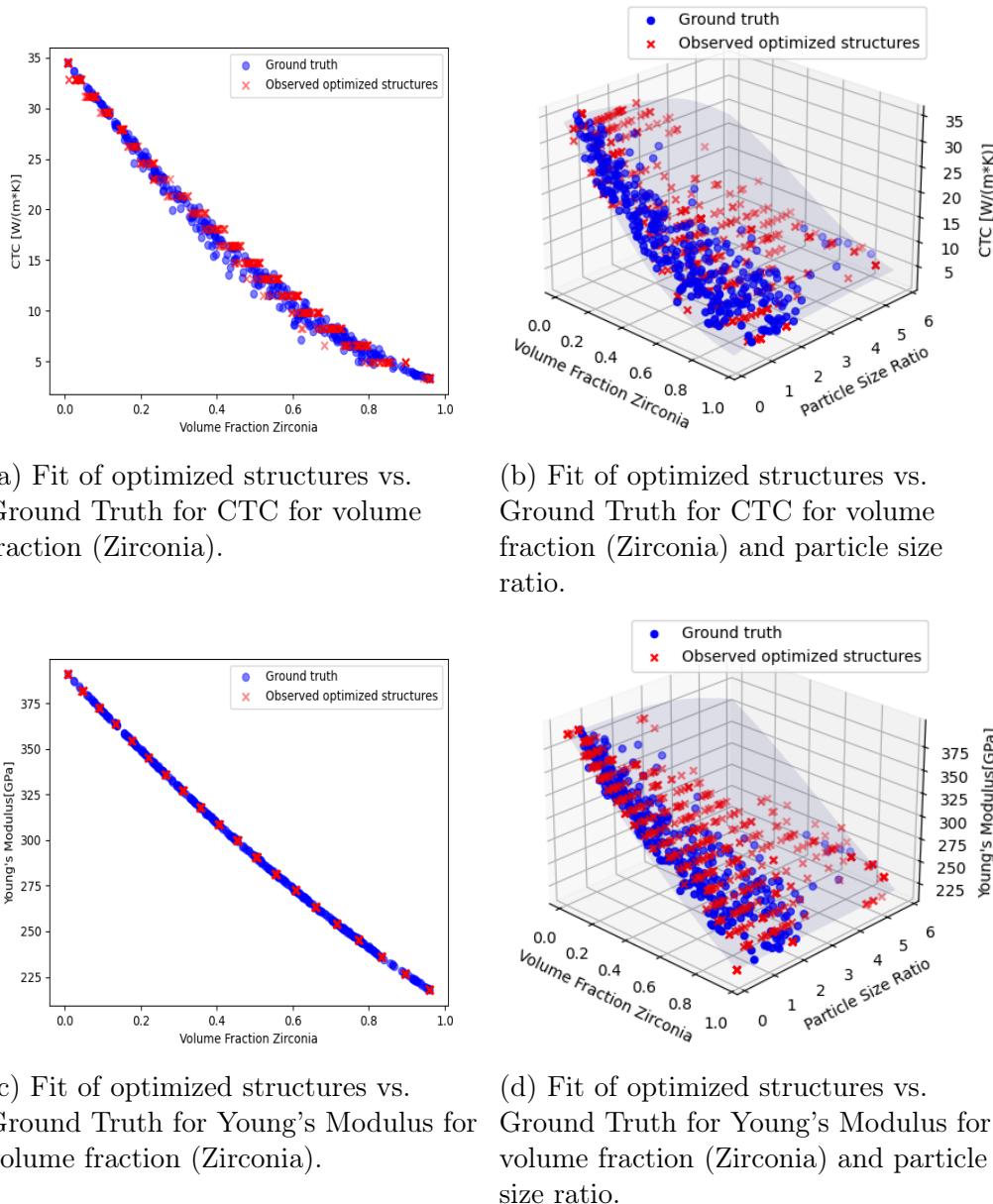


Figure 5.7: Visualisation of optimized microstructures for CTC and Young's Modulus using the modified objective function (Equation 4.12).

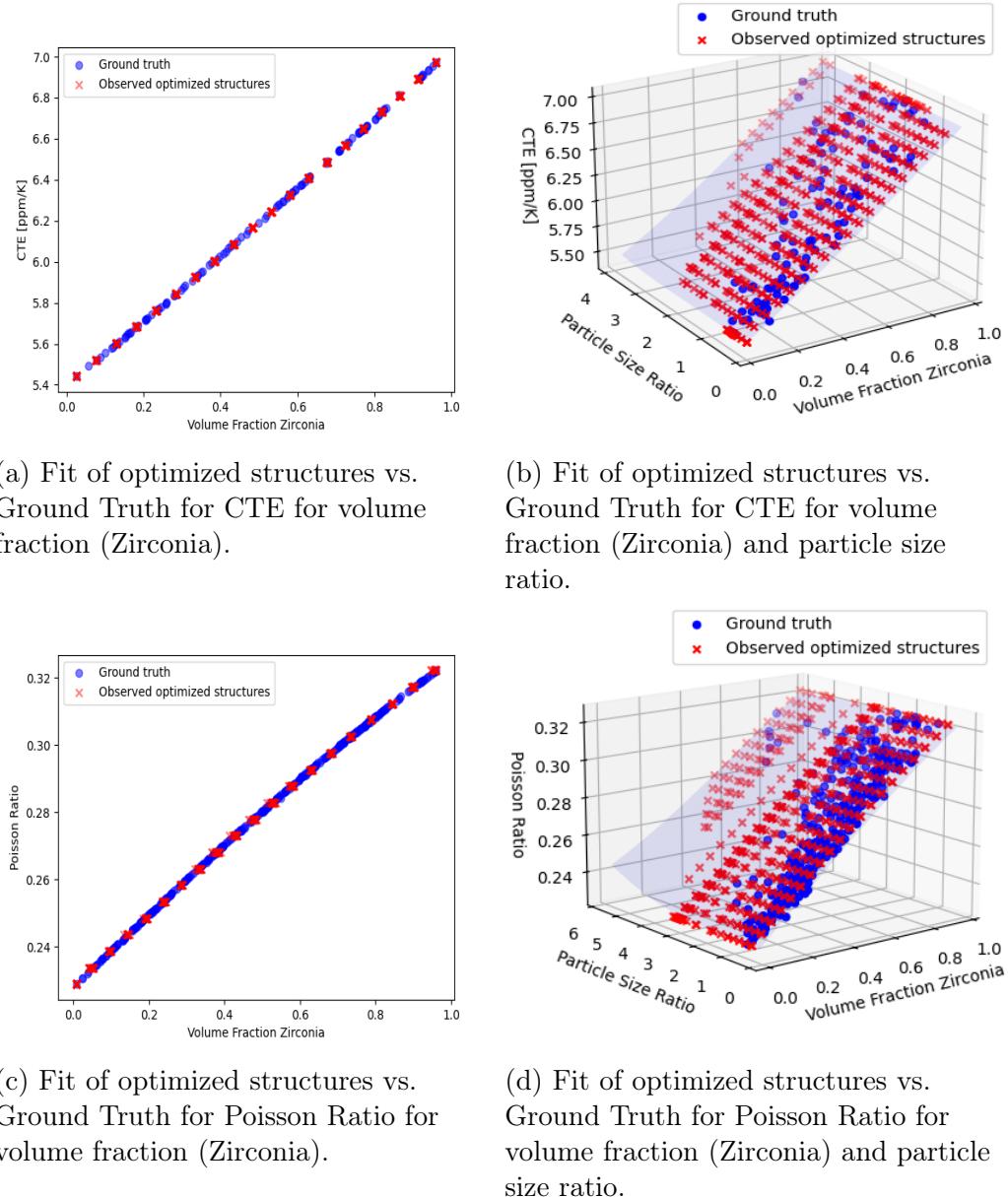


Figure 5.8: Visualisation of optimized microstructures for CTE and Poisson Ratio using the modified objective function (Equation 4.12).

as described in Table 5.3, which were already well-performing. Through this case study, a robust gradient-based optimization approach was developed which overcomes some limitations of generalizability in GPRs as observed for the inverse design of ZTA/ATZ ceramics. For experiments in further sections, unless specified, inverse design for all properties will use the modified objective function.

5.2.2 Gradient Based Optimizers vs. Gradient Free Optimizers

Model	RR^2	RRMSE	C.Time
CTC			
L-BFGS-B	1.00	1.12e-02	12.3s
SLSQP	1.00	1.13e-02	6.1s
Powell	1.00	1.15e-02	53s
Nelder-Mead	1.00	1.68e-02	52.5s
CTE			
L-BFGS-B	1.00	1.83e-04	0.99s
SLSQP	1.00	1.38e-03	0.51s
Powell	1.00	2.15e-04	22.1s
Nelder-Mead	0.99	9.99e-03	8.8s
Young's Modulus			
L-BFGS-B	1.00	2.54e-02	10.9s
SLSQP	1.00	2.94e-02	6.7s
Powell	1.00	4.11e-02	69.9s
Nelder-Mead	1.00	4.6e-02	29.4s
Poisson Ratio			
L-BFGS-B	1.00	1.94e-04	0.92s
SLSQP	0.99	1.746e-03	0.88s
Powell	1.00	1.06e-04	24.8s
Nelder-Mead	0.99	1.833e-03	13.5s

Table 5.4: Performance Metrics of different optimization techniques. Metrics include RR^2 , RRMSE, and Computational Time in seconds (C.Time).

This section shows a comparison of the different optimization techniques for the inverse design methodology. The experiments were performed on a system with an AMD Ryzen 5 7600X CPU, Nvidia RTX 4070 GPU, and 32 GB of RAM. Table 5.4 shows the performance of two gradient-based optimizers (L-BFGS-B and SLSQP) and two gradient-free optimizers (Powell

and Nelder-Mead). The metrics were performed for 20 test designs over the range of the respective material property with the modified objective function, similar to the tests in the previous section. While the RRMSE is fairly close in all methods, SLSQP and Nelder-Mead perform slightly poorer for CTE and Poisson Ratio. Other than this, L-BFGS-B shows the best RRMSE in all properties except for Poisson Ratio, where the Powell method performs slightly better by an error that is smaller only by 8.8e-05. This is overshadowed by fact that the computation times for the gradient-free techniques (Powell and Nelder-Mead) are significantly higher than gradient-based techniques (L-BFGS-B) for every property, ranging from approximately 4.3 times faster for CTC to nearly 25 times faster for Poisson Ratio. This brief comparison shows the power of gradient-based techniques and why they were chosen for the inverse design methodology. From this comparison, L-BFGS-B can be seen to perform consistently well across all material properties and will be recommended for most uses in the inverse design methodology (consistent with the recommendations by Varoquaux [25]).

5.2.3 Comparison to Baseline GBR

Property	RR^2	RRMSE
Gradient Boosting Regressor		
CTC	-0.64	1.639
CTE	-0.893	0.639
Young's Modulus	-0.745	2.639
Poisson Ratio	-0.862	0.521

Table 5.5: Inverse Design Performance Metrics with GBR surrogate model.

The GPR was chosen for this study because it was hypothesized that the GBR implemented in the foundational study by Pirkelmann et al. [8] would perform poorly as a surrogate model in an optimization routine. A brief look at how the GBR functions with the inverse design methodology is shown for CTE in this section. This comparison only uses the standard objective function, since the GBR does not provide a measure of uncertainty. The resulting conclusions are generalizable for the other material properties as well. For the purpose of this experiment, the condition $J(x) < \theta$ (from Equation 4.9) was ignored since the GBR could not find solutions for many property values. Therefore, all minima were accepted as solutions for further investigation. This does not affect the fairness of this experiment by a lot, since the optimization for CTE returned 29.1 solutions on average satisfying $J(x) < \theta$ for each desired value. This already shows the limitation of GBRs.

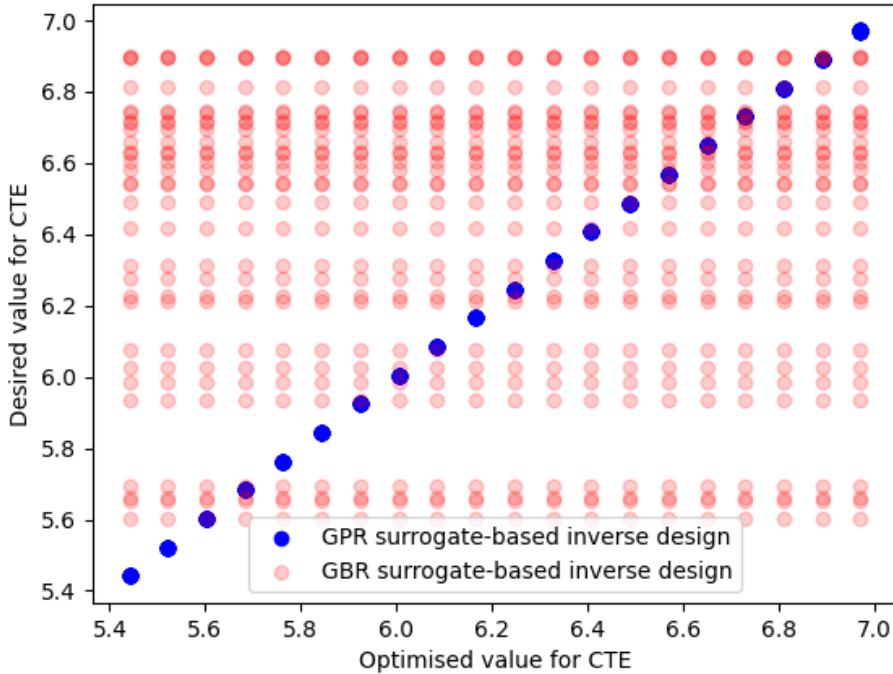
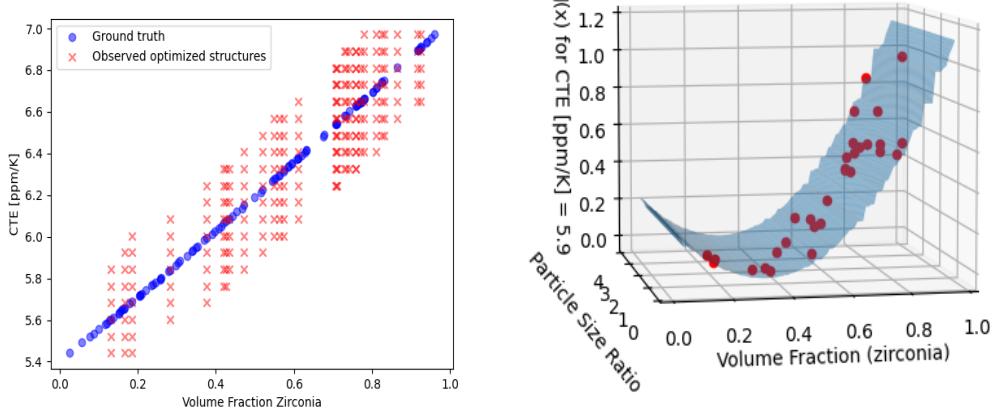


Figure 5.9: Fit of the property values of the optimized structures and the initially desired structures with different surrogate models.

A comparison of the GBR and the GPR model for inverse design, in terms of the fit of the property values of the optimized structures and the initially desired structures, can be seen in Figure 5.9. The GBR surrogate-based inverse design performs poorly, suggesting multiple solutions for each desired value across the entire range of CTE. As shown in Table 5.5, the GBR model generated an RRMSE of 0.639 and an RR^2 score of -0.893 for CTE. This is a high RRMSE for CTE when compared to the results seen earlier with GPRs (from Table 5.3), where the RRMSE was as low as $1e - 4$. The RR^2 fit is very weak as well, which means that many of the predicted property values are offset from desired property values by a large margin, which is seen in Figure 5.9.

Figures 5.10a display the fit of optimized microstructures over the ground truth data using GBR surrogates for inverse modeling. Figure 5.10a shows that the optimization for different property values does not return microstructures known from ground truth. A lot of desired values suggest solution structures that should belong to a different value of CTE. A further look into the



(a) Design space wide inverse design for CTE using GBR-based surrogate. (b) Objective function for $CTE = 5.9 \text{ ppm}/K$.

Figure 5.10: Inverse design using GBR.

shape of the objective function for a CTE value equal to 5.9 ppm/K in Figure 5.10b shows that this objective function resembles a sort of step function, and a lot of solutions are stuck in the horizontal edges which should not normally be part of the minima. This, as hypothesized, was due to the nature of GBR modeling the surrogate function in a staggered manner. GBRs are essentially an ensemble of decision trees, which can often at decision boundaries become staggered and lead to regions of zero-value gradients. When such a region is encountered in the objective function, the gradient-based optimizer must travel in a step based on the gradient, which is zero, and therefore is unable to leave this region even if this is not a minimum in the true sense.

These observations highlight the limitations of using GBRs as surrogate models in inverse design and encourage the use of ML techniques that model data in a smooth manner, such as the GPR, for multi-phase ceramics.

5.2.4 Comparison of Kernels

In this section, a full comparative study into the performance of each kernel in inverse design is presented. The performance is compared using quantitative metrics such as RR^2 , RRMSE, and computation time (in seconds).

Table 5.6 shows the property-wise performance of each kernel with analytical gradients and approximated gradients. The approximated gradients are performed using a 2-point finite difference with an absolute step size. The discussion of this study is two-fold, i.e., to identify the best-performing kernels

Kernel	Analytical Gradient			Approximated Gradient		
	RR ²	RRMSE	C.Time	RR ²	RRMSE	C.Time
CTC						
DotProduct	1.000	6.59e-02	3.79s	1.000	3.69e-05	1.82s
RationalQuadratic	1.000	1.08e-02	12.7s	1.000	1.09e-02	16.2s
RBF	1.000	1.12e-02	12.3s	1.000	1.15e-02	16.06s
Matern	1.000	1.88e-02	8.26s	1.000	1.88e-02	16.13s
RBF * DotProduct	1.000	1.68e-01	4.5s	1.000	2.07e-02	10.58s
CTE						
DotProduct	0.805	2.05e-01	1.5s	1.000	2.96e-03	2.43s
RationalQuadratic	1.000	1.81e-04	1.03s	1.000	2.27e-03	3.82s
RBF	1.000	1.83e-04	0.99s	1.000	2.12e-03	3.84s
Matern	1.000	3.37e-04	1.6s	1.000	2.28e-03	3.82s
RBF * DotProduct	0.031	4.21e-01	2.25s	0.9993	1.23e-02	4.65s
Young's Modulus						
DotProduct	1.000	9.27e-02	2.5s	1.000	1.53e-04	3.78s
RationalQuadratic	1.000	2.79e-02	11.13s	1.000	5.61e-02	15.0s
RBF	1.000	2.54e-02	10.9s	1.000	5.49e-02	14.35s
Matern	1.000	2.6e-02	8.2s	1.000	4.77e-02	12.08s
RBF * DotProduct	1.000	3.37e-01	11.1s	1.000	1.08e-01	24.51s
Poisson Ratio						
DotProduct	-0.788	3.79e-02	2.59s	0.9998	4.34e-04	3.88s
RationalQuadratic	1.000	1.91e-04	0.94s	0.9987	1.02e-03	7.51s
RBF	1.000	1.94e-04	0.92s	0.9987	1.02e-03	7.22s
Matern	0.999	2.02e-04	1.18s	0.9982	1.19e-03	7.24s
RBF * DotProduct	-0.814	3.81e-02	3.12s	0.5274	1.94e-02	7.74s

Table 5.6: Performance metrics of different kernels in inverse design using analytical and approximated gradients. Metrics include RR², RRMSE, and Computational Time in seconds (C.Time).

for each property that are the most accurate without losing much performance in computational speed (determined in conjunction with the results from Section 5.1 with Figure 5.3), and to determine whether it really is beneficial to use analytical gradients that are computed using potentially intensive matrix operations over approximated gradients which is simply requires the evaluation of the surrogate model at two points separated by a very small distance. The former ensures that by selecting kernels that already show high regression accuracy (R^2 , RMSE) in surrogate models and also show high inverse modeling accuracy (RR², RRMSE) while using those surrogate models, a robust and highly reliable inverse design is achieved. This also determines whether acceptable tradeoffs between speed and accuracy exist in kernels. Note that the regression metrics used in Section 5.1 such as the R^2 are dif-

ferent from the inverse design metrics used, such as the RR^2 , and should not be confused with each other, as they will both be simultaneously discussed in this section. Additionally, while not explicitly stated in this section, each considered kernel has an added *White* kernel component. For example, the *RBF* kernel considered is the *RBF + White* kernel.

Based on a look at the gradients of each kernel in Section 4.6, it can be expected that in most cases, each evaluation of the *DotProduct* kernel and its gradients can be evaluated the quickest, as it always returns a constant matrix of the training features \mathbf{X} (Equation 4.19). This should be followed by closely similar performance between *RBF* (Equation 4.16), *RationalQuadratic* (Equation 4.20), and *Matern* (Equation 4.17) kernels. The *RBF * DotProduct* kernel should take the longest as it is the most complex kernel. The effect of the *White* kernel can be considered minimal on computational times as its gradient is a constant matrix of zeros. It can also be naturally expected that analytical gradients perform better than approximated gradients. However, due to the nature of the inverse problem and gradient calculations, the use of a standardization factor which must also unscale the gradients, and internal divisions by very small numbers, there is always a possibility of unexplainable numerical instability issues.

One of the notable observations is that the *DotProduct* kernel does indeed converge the fastest in all properties. The approximated gradients for the *DotProduct* kernel seem to perform better than the analytical gradients in almost all cases in terms of RR^2 and RRMSE with only slightly worse computation times ($\pm 1s$). The *DotProduct* kernel shows a weak RR^2 fit for CTE and Poisson ratio with analytical gradients, but seems to function well with approximated gradients. This is unusual and some further investigation suggested that the gradients of the uncertainty (Equation 4.15) found analytically showed large errors against approximated gradients with the *DotProduct* kernel. This might be perhaps due to some numerical stability issues but an exact cause could not be determined. Due to this, it is recommended to rely on approximated gradients when gradients of the uncertainty are required when using the *DotProduct* kernel. However, even with approximated gradients, though CTC and Young's modulus show very low RRMSE (3.69e-05 and 1.53e-04 respectively), it still remains from Figure 5.3 (from Section 5.1) that the surrogate model with the *DotProduct* kernel has a weaker R^2 fits, which already discourages its use. The *DotProduct* kernel for CTE and Poisson Ratio achieves RRMSE values that are weaker than other kernels as well. Additionally, while the *RBF * DotProduct* kernel had performed very well in the regression results in Figure 5.3 (from Section 5.1) where it had an edge in accuracy over other kernels, the kernel combi-

nation performs poorly here across the board in all properties in terms of its RRMSE score. This is seen with both the analytical and approximated gradients when compared to the better-performing kernels, where it is worse by an order of 1e-01 for CTC and Young's Modulus and by an order of 1e-02 for CTE and Poisson Ratio. Furthermore, this kernel combination shows poor R^2 fits of 0.031 and -0.814 for CTE and Young's Modulus, respectively. Even with approximated gradients for $RBF * DotProduct$, the R^2 and the RRMSE shown is not as good as seen in other kernels. For these reasons, the effectiveness of the *DotProduct* kernel is questionable and not recommended for any material property, even with kernel combinations, unless fast computation time is a prerequisite and a higher error margin is acceptable.

Disregarding the *DotProduct* kernels, an overview of the remaining kernels can be considered. For CTC and Young's Modulus, *RationalQuadratic*, *RBF*, and *Matern* kernels show very similar RRMSE scores with the *Matern* kernels being slightly faster by 2-4 seconds. For CTC, even though *Matern* kernel shows faster computation, the *RationalQuadratic* and the *RBF* kernel showed slightly higher R^2 regression scores of 0.9981 over the *Matern* kernel's score of 0.9979 (2-feature set results for CTC in Figure 5.3). Since the purpose of this section is to find the most accurate kernels (in terms of both regression accuracy and inverse modeling accuracy) without losing too much computational speed, either of these two kernels (*RBF* and *RationalQuadratic*) should be preferred over the *Matern* kernel. The computation time tradeoff of approximately 4s over the *Matern* kernel is minimal since this time is aggregated over the 20 different test designs and in real-time applications only a single design is sought. In this scenario, the time difference would be unnoticeable. Similarly for Young's Modulus, the *RationalQuadratic* and the *RBF* kernel showed perfect R^2 scores of 1.0 (2-feature set results for Young's Modulus in Figure 5.3) and should be preferred instead while accepting a small computation time overhead.

For CTE and Poisson Ratio, the *RationalQuadratic* and *RBF* kernel with analytical gradients outperform the other kernels in terms of both speed and accuracy. For CTE, an RRMSE of approximately 1.8e-04 and a computation time of approximately 1s are shown by both kernels. Since the 2-feature set regression results from Section 5.1 for CTE (Figure 5.3) showed a slightly better R^2 score with *RBF* kernel (0.986) over the *RationalQuadratic* kernel (0.983), the *RBF* kernel can be preferred for CTE. For Poisson Ratio, an RRMSE of 1.9e-04 and a computation time of approximately 0.9s are shown by both *RationalQuadratic* and *RBF* kernels. The 2-feature set regression results for Poisson Ratio (Figure 5.3) show similar R^2 scores with all kernels, thus, either the *RBF* or the *RationalQuadratic* are equally acceptable for

Poisson Ratio.

For all properties, it can be seen that the computation time for the 20 test design cases can be computed in seconds, making it highly viable for real-time applications in the design of composite ceramic materials. The *RBF* kernel and *RationalQuadratic* kernel are also seen to be a viable choice for all properties in terms of accuracy in surrogate models, inverse design (reconstruction) accuracy, and computation speeds. For a choice between the two, for its known simplicity, the *RBF* kernel can be chosen as the kernel for modeling GP surrogates in future applications.

On the topic of analytical gradients and approximated gradients, the analytical gradients can be seen to vastly outperform approximated gradients with *RationalQuadratic*, *RBF*, and *Matern* kernels. In CTC and Young's Modulus, the analytical gradients for these kernels are approximately 30% faster than approximated gradients with similar, if not better, RRMSE. For CTE, the analytical gradients are around 2-3 times faster than approximated gradients and for Poisson Ratio, the analytical gradients are more than 7 times faster than approximated gradients. Furthermore, in both CTE and Poisson Ratio, analytical gradients achieved an RRMSE of the order 1e-04 while approximated gradients only achieved an RRMSE of the order 1e-03. These observations show that even though approximated gradients are good enough to achieve good results in inverse design, they are almost always slower to compute and cannot find minima as well as analytical gradients. This shows that the choice of GPR as a surrogate model, which can model smooth functions whose gradients can be calculated analytically, is apt for an inverse design application for both speed and accuracy.

5.3 Direct Inverse Training

In this section, a study into the performance characteristics of DIT is presented. This study aims to compare the inverse modeling methodology used so far with a direct ML model trained in the reverse direction.

DIT Metrics

This is essentially a multi-target regression task for an inverse design application where, instead of finding microstructure x^* for a desired property value $y = y^*$ using a surrogate function $f(x) = y$, a function $f(y) = x$ is trained directly by treating the material property (Y) as features and microstructure (X) as targets to find x^* directly. This causes an issue with standard regres-

sion metrics as described in Section 5.1. In an inverse design application, a desired property can have multiple design solutions. This means that for a desired $y = y^*$, there can be many design solutions $x^* \in X^*$, where X^* is a set of all such solutions. When using standard regression metrics with cross-validation, a one-to-one mapping of the features to the targets is expected for the test data. However, in DIT, the model can predict targets, microstructures x^* , that are different from the microstructure targets in the test data x_{test}^* for a given y_{test}^* where $x_{test}^* \in X^*$ and X^* is the set of all possible microstructure solutions for $y = y_{test}^*$. It might still remain that the proposed x^* is a different solution that satisfies $x^* \in X^*$ for $y = y_{test}^*$. Therefore one cannot expect uniqueness in solutions and regression metrics that expect one-to-one mappings for validation data would be an unfair manner to judge DIT.

A better way to quantitatively analyze DIT would be to quantify the error in the material property y exhibited by the predicted microstructure x^* and the initially desired material property y^* . The material property y exhibited by the predicted microstructure x^* can be evaluated using a known forward function $g(x) = y$. Ideally, the forward function for this case would be the FEM simulations, but due to its computational demand and time constraints, a surrogate function with high accuracy is substituted instead. The surrogate model used to validate the designs of DIT was the GBR described in Section 5.1, which had an accuracy of 0.99 in most material properties. This error can now be quantified as $\epsilon = g(x^*) - y^*$ with $x^* = f(y^*)$ and substituted in metrics such as R^2 and RMSE. These metrics are now semantically equivalent to the metrics defined for inverse modeling in Section 5.2 and will be similarly referred to as RR^2 and RRMSE.

Results

Table 5.7 shows the performance metrics of DIT using NN, GP, and RF. Across all properties, NN shows relatively the weakest RR^2 and RRMSE despite being the most complex model. The GP performs slightly better and the RF performs the best across all properties with respect to the RR^2 and RRMSE. Note that this GP should not be confused with the GP used in the optimization-based inverse design methodology.

Comparing the metrics here to the earlier inverse design methodology (Table 5.6) shows that the inverse design methodology with optimization routines outperforms the DIT approach. Considering the *RBF* kernel in Table 5.6 for comparisons to DIT with the RF model: for CTC an RRMSE of 1.12e-02 is seen with the optimization-based approach compared to 2.83e-01 for

DIT; for CTE an RRMSE of 1.83e-04 is seen with the optimization-based approach compared to 5.96e-03 for DIT; for Young’s Modulus an RRMSE of 2.54e-02 is seen with the optimization-based approach compared to 3.69e-01 for DIT; and for Poisson Ratio an RRMSE of 1.9e-04 is seen in both approaches. Except for Poisson Ratio, the margin of error is a lot higher in other properties. The one advantage DIT has over the optimization-based inverse design workflow is its instantaneous predictions, and considering the ranges of each material property (described in Section 4.3), the higher margin of error may be acceptable if the inverse problem is empirically solved well.

Model	RR^2	RRMSE
CTC		
NN	0.9941	5e-01
GP	0.9970	3.8e-01
RF	0.9979	2.83e-01
CTE		
NN	0.9953	1.6e-02
GP	0.9970	1.5e-02
RF	0.9994	5.96e-03
Young’s Modulus		
NN	0.9988	1.2
GP	0.9990	1.02
RF	0.9999	3.69e-01
Poisson Ratio		
NN	0.9988	6.6e-04
GP	0.9990	5.2e-04
RF	0.9999	1.97e-04

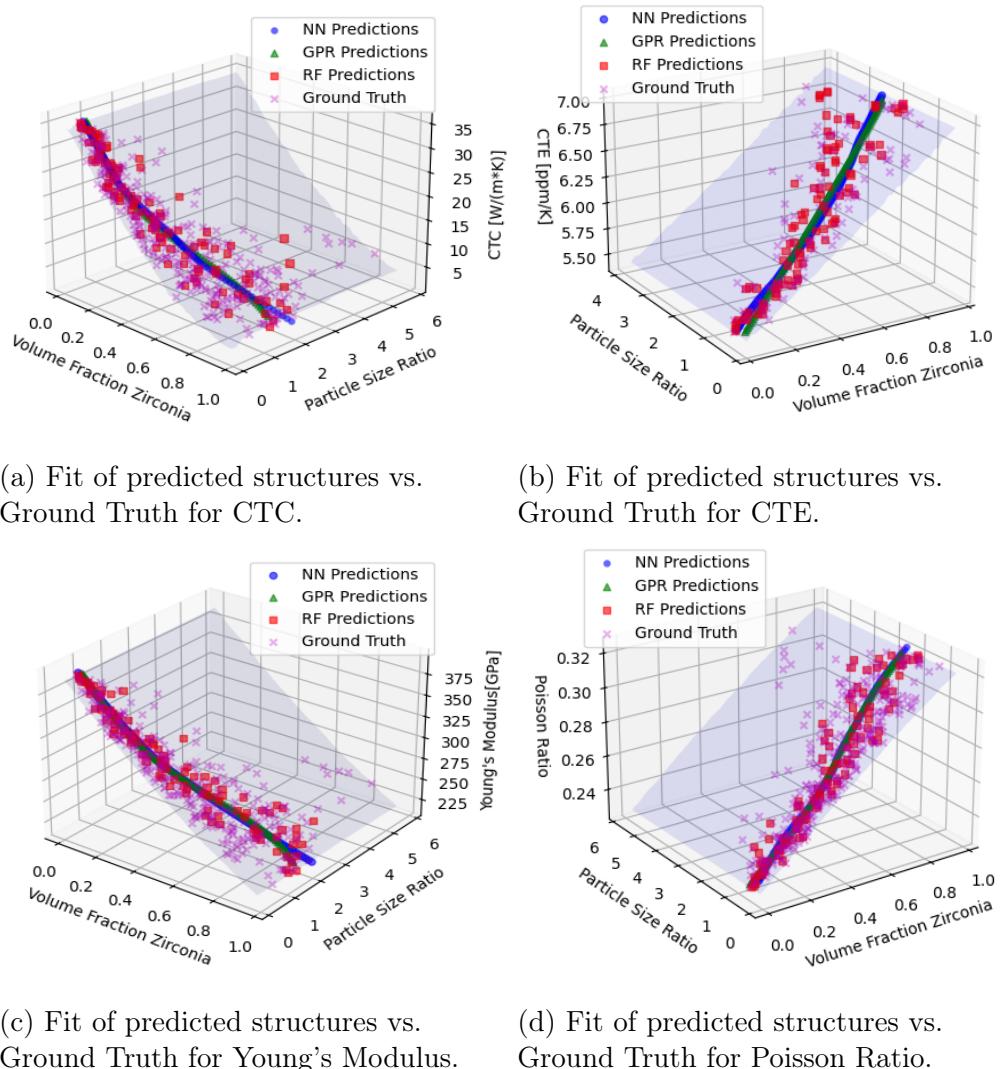
Table 5.7: Performance metrics of DIT for different properties with Neural Networks (NN), GP, and Random Forests (RF).

Figure 5.11 shows the fit of the microstructures predicted using DIT across the range for each material property. The shaded plane represents a forward function (in this case a GBR surrogate) for each material property. It can be seen that the NN and GPR predict a smooth curve in each material property. The microstructures appear correct as they lie on the plane of the forward function but show very little generalizing power and predict microstructures for a very small range of particle size ratios. The RF on the other hand predicts more evenly distributed microstructures over a wider range of particle size ratios. Moreover, these microstructures seem to be constrained around known ground truth (training data), which is acceptable.

From this, it can be seen why the RF performs better than the NN and the GPR with DIT.

The advantages of DIT lie in its simplicity of training and instantaneous predictions. DIT does not require real-time optimization routines for the suggestion of microstructures for desired properties, reducing the overall time complexity of the method. The model training is simple, one simply flips the microstructure features as the targets and the material properties as features. If this is attractive and favorable, given that domain experts can approximate additional solutions based on a single solution, the RF appears to be a good choice. It may also be possible to train multiple RF-based DIT models that each predict different solutions. This would require well-defined training data subsets for each model and could be a possible avenue for future work.

Despite the advantages of DIT, the primary drawback of this approach is that it approaches the inverse problem without addressing its ill-posed nature. DIT can only predict a single microstructure for a desired property, whereas the optimization-based inverse design workflow is able to find multiple solution microstructures (as shown in Section 5.2). This is important because in a real-world scenario feasibility of microstructures may only be possible under certain constraints with techniques such as milling and sintering. When multiple possible microstructures are known, domain experts can choose the right microstructure that is known to be feasible with their existing real-world constraints. In this regard, DIT is limited. The computation time for the suggestion of a single desired property with the optimization-based inverse design workflow was shown to be almost instantaneous as well in an experimental scenario despite the higher computational complexity over DIT (in Subsection 5.2.4). This furthers the appeal of using the optimization-based inverse design workflow over DIT and shows that it is a much more elegant approach to the inverse problem.



(a) Fit of predicted structures vs. Ground Truth for CTC.

(b) Fit of predicted structures vs. Ground Truth for CTE.

(c) Fit of predicted structures vs. Ground Truth for Young's Modulus.

(d) Fit of predicted structures vs. Ground Truth for Poisson Ratio.

Figure 5.11: Visualisation of predicted microstructures for material properties with DIT using NNs, GPR and RF, shaded region shows the prediction plane of a forward GBR surrogate.

5.4 Adaptive GPRs

The limitations of the GPR in this study revolve around its observed inability to cope with missing data which causes failure in generalizing the forward function well for regions of very low data density. In the case of this study, this was seen in the results for the inverse modeling (Section 5.2) where the region of low data density was the region of infeasible microstructures (high particle size ratios and low volume fraction of Zirconia). In most material properties, it was seen that the GPR prediction function \bar{f}_* plane behaved strangely and showed exaggerated folding of the prediction plane in these regions (Figures 5.4d, 5.4b, and 5.5d), which affects the overall inverse modeling as well when using the standard objective function (Equation 4.10). While using the modified objective function (Equation 4.12) has shown to be an effective way to address this, a more sophisticated solution would be to improve the GPR surrogate model generalizability by adding data points in regions of low data density. This is where the adaptive sampling approach described in Section 4.5 can potentially be useful.

Another main reason behind using the adaptive sampling approach as described was to reduce the overall size of the data. It is a well-known fact that the time complexity of GPRs is $\mathcal{O}(n^3)$ [9] where n is the size of the training data. This is due to the inversion of the $n \times n$ matrix in Equation 2.2. The offline data that was used for training the GPRs so far had around 450 training samples for CTC, Young's Modulus and Poisson Ratio (described in Section 4.3), and around 149 samples for CTE. Using the adaptive GP methodology described in 4.5, where the parameter reconstruction error (or the expected error in inverse design) is quantified through the accuracy model and minimized, it is hypothesized that less data than this is required to train the models for these material properties.

The adaptive training was run for around 100 adaptive steps for CTE and Poisson ratio, and around 200 steps for CTC and Young's Modulus. For all properties, *RBF + White* was used as the kernel. Training typically took around 12 hours for 200 adaptive steps with online simulations and 6 hours for 100 adaptive steps.

For CTC, the accuracy evolution over each step is shown in Figure 5.12a. It is seen that the GP model reaches a high R^2 accuracy (approximately 0.998) at a very low number of samples (around 25 samples). The accuracy then fluctuates slightly near 50 samples and starts to converge back to 0.998 at around 100 samples. The fluctuation is possibly due to the fact that a fixed variance is assigned to all adaptively added points as described in Section

4.5 which can affect the quality of predictions. This behavior can possibly be addressed by determination of the true variances of added samples and improving them using the work model (also described in Section 4.5), but for the purpose of this work, simply adding more samples eventually remedies this.

For CTE (Figure 5.12b), Young’s modulus (Figure 5.12c) and Poisson Ratio (Figure 5.12d), the accuracy starts to converge to an R^2 of nearly 1 at just 40 samples. These R^2 scores are similar if not better than the scores seen from the 2-feature set results from Figure 5.3 for all material properties.

This shows that the same level of accuracy can be obtained with a much smaller number of training samples when using the adaptive approach compared to offline data with random sampling: from 450 samples to 100 for CTC; from 450 to 40 training samples for both Young’s Modulus and Poisson Ratio respectively; and from 149 to 40 samples for CTE.

The evolution of the Volume Fraction (Zirconia) vs Material Property curve can be visualized with a fixed particle size ratio of 1 for the adaptive approach as more training samples are added to the GP with different material properties. This can be seen in Figure 5.13. For CTC (Figure 5.13a) the curve deviates between 10 samples and 50 samples (where lower accuracy was also seen in Figure 5.12a) but for the remaining number of samples plotted, it converges very tightly. For the remaining material properties (Figures 5.13b, 5.13c, 5.13d), the curves have very little deviation after just 20 samples. This shows the adaptive GP’s ability to estimate the function with very small amounts of data.

Showing the evolution of the design space would be exhaustive for all material properties, instead, the design space and the local reconstruction error (described in Section 4.5) quantified as contours are shown at different iterations for the example of CTC. This gives an intuition of how the adaptive GP diminishes the local errors throughout the feature space to obtain a robust surrogate model. Figure 5.14 shows the visualized design space as described at four different iterations. At the first iteration (Figure 5.14a), the initial data points are shown. Usually, the initial data samples are found at boundaries and the mid-point of the design space before the adaptive phase. Here, they are not perfectly at the boundaries due to the variance in microstructures generated by the structure generator but the small deviation is acceptable since the generated microstructures are usually very close in the design space. If a larger deviation is seen, the microstructures are excluded using the exclusion strategy described in Section 4.5. Due to this exclusion strategy of iteratively excluding structures that are unable to be generated

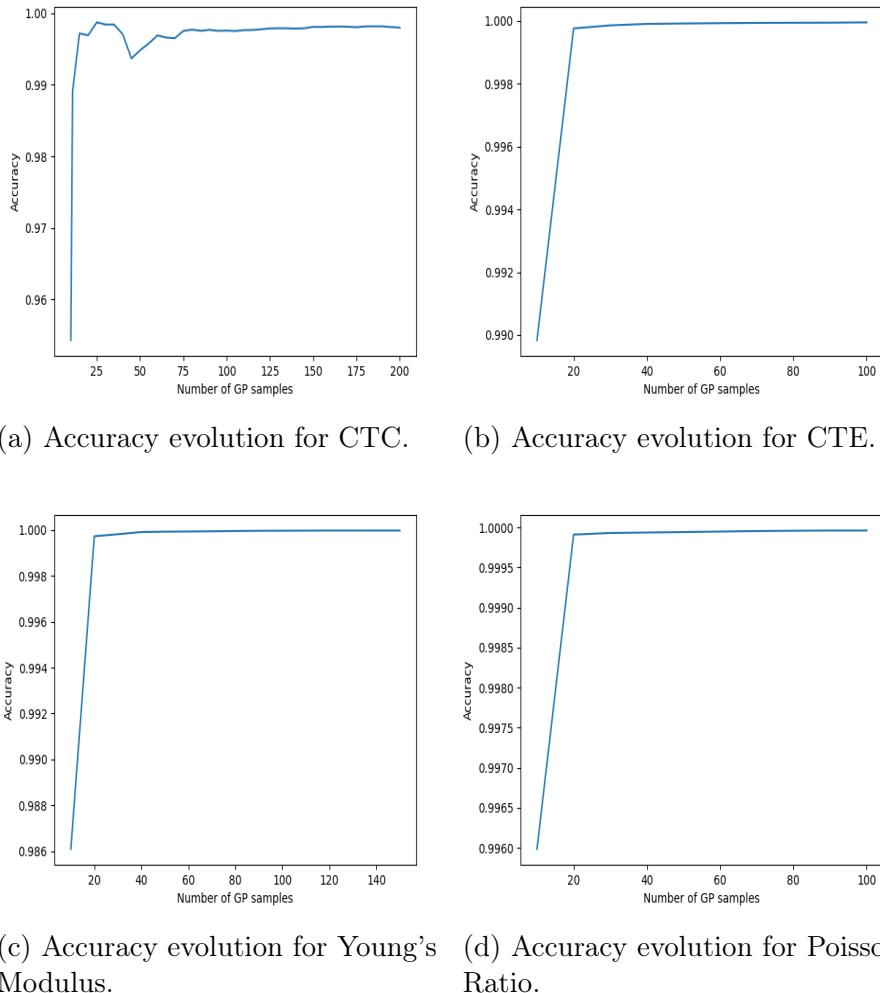


Figure 5.12: Evolution of the R^2 accuracy as training samples are added with adaptive GP for each material property.

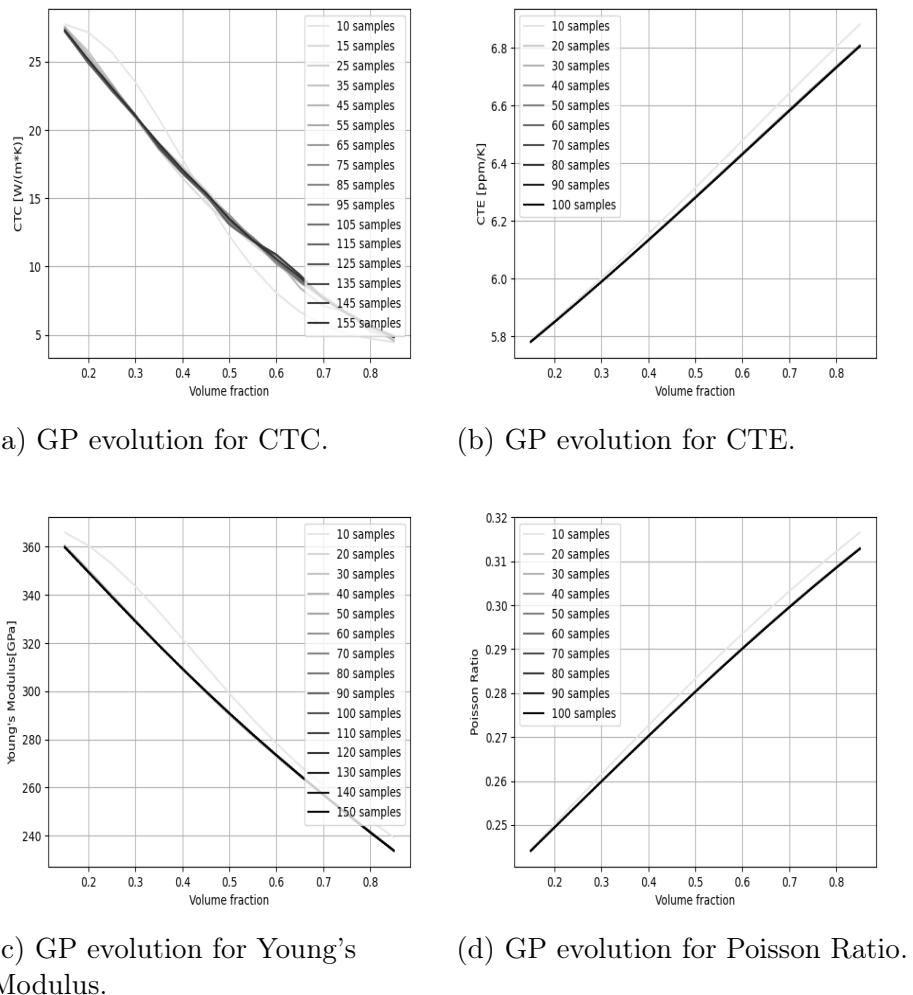


Figure 5.13: Evolution of Volume Fraction (Zirconia) vs. Material Property for fixed particle size ratio = 1 with adaptive approach.

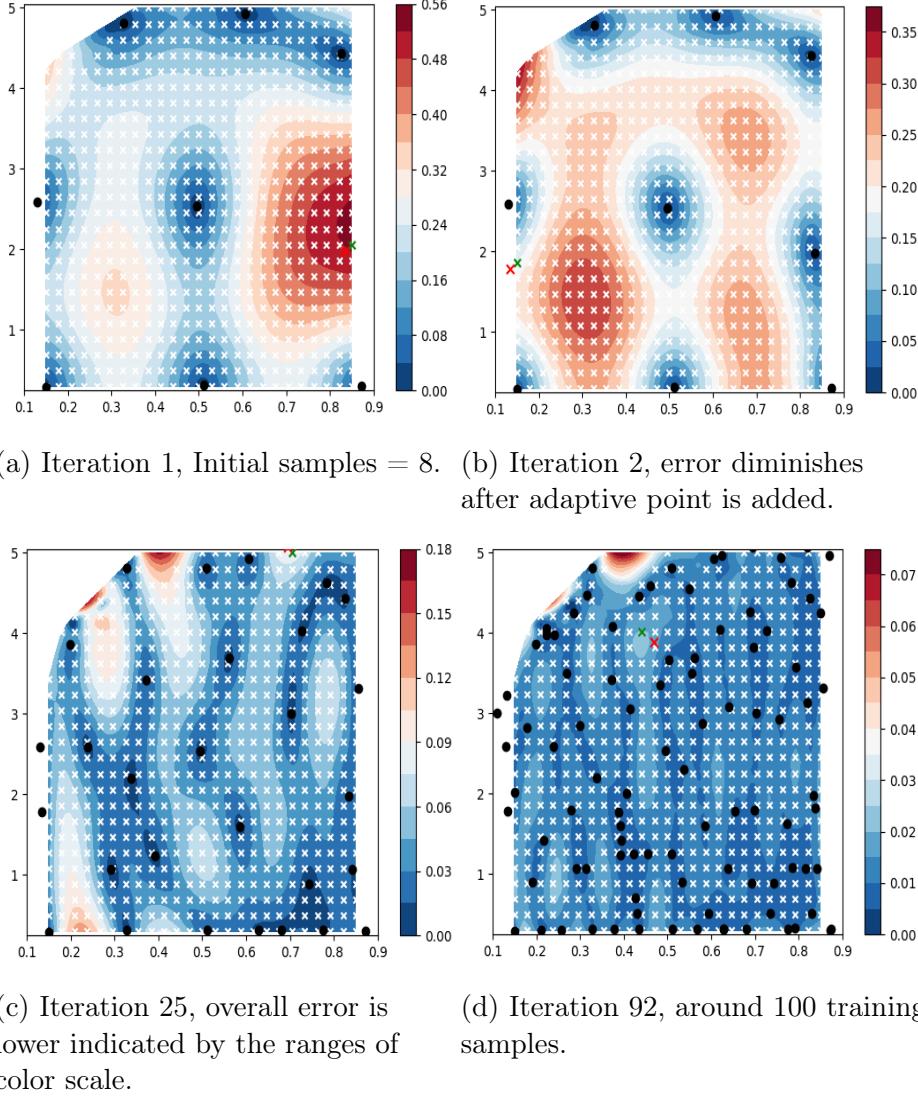


Figure 5.14: Design space at different iterations in the adaptive GPR training for CTC. Black points show training samples added to the GP, green crosses show the candidate point requested by the adaptive GP for the current iteration, red crosses show the points returned by the FEM simulations and adaptively added in the current iteration, white crosses represent the sampled grid points that are considered for the adaptive step, and the contours represent the isolines of the local reconstruction error.

or show too much deviation, the adaptive sampling already excludes the infeasible microstructure region in the top-left (low volume fractions and high particle size ratios) in Figure 5.14a where there are no white crosses. It can be seen that the region at the middle-right (in Figure 5.14a) has a high local error (red contours). The next candidate microstructure to be added is within the high local error region as well, indicated as the green cross. After the generated point is simulated with FEM and added at the red cross, it can be seen the error in that region where the adaptive point was added has now diminished in the next iteration (Figure 5.14b). Subsequent new high local error regions color mapped as red are revealed at the regions that were previously color mapped as white in the first iteration since those regions are now the highest error regions. This can be seen from the color map ranges of the errors which decreases from $[0, 0.56]$ to $[0, 0.35]$. Iteration 25 (Figure 5.14c) shows that the overall error is even lower now as indicated by the ranges of the color scale. Finally, in iteration 92 (Figure 5.14d), where the global error starts to converge, the overall local error has now further diminished and most regions are color-mapped as blue. The adaptive training samples cover the parameter space much more uniformly compared to the coverage seen in the samples in the offline data (Figure 4.6).

This process of adaptively adding points and subsequently diminishing local reconstruction errors in a greedy fashion can be performed for many iterations until the global reconstruction error (Equation 4.5) starts to converge, that is, when no significant change is seen in the global error by adding more training samples. For a full evaluation, the adaptive GP was allowed to run for more than 200 samples. The global reconstruction error per iteration for CTC can be seen in Figure 5.15a which is in agreement with the change of local errors seen in the design space plots (Figure 5.14). The global error gradually decreases with some fluctuations around 40 iterations (approximately 50 training samples). The global errors then fall to a low value (smaller than 0.004) at around 90 iterations (approximately 100 training samples) and start to converge from there. This is similar to the results seen in the accuracy evolution plots for CTC (Figure 5.12a) as well and there is some noticeable correlation between the quantified global reconstruction error and the R^2 accuracy of the model. This shows that in the absence of validation data for accuracy measures, the global reconstruction error estimate is a reliable metric for evaluating model quality. Subsequent adaptively added points do not reduce the global error estimate by much even after 200 iterations. This shows that the GP does not need further adaptive sampling, and adaptive sampling could have been stopped around the 100th iteration. It also indicates that the adaptive GP is ready for application in inverse de-

sign as a surrogate model. The global error estimate plots for other properties are also shown in Figures 5.15b, 5.15c, and 5.15d. For the other properties which are known to have less variance than CTC, the global error estimate is a much smoother curve and quickly drops to a low value and starts to converge within the first 20 iterations (similar to their respective accuracy evolution plots).

To demonstrate the power of adaptive GPs, an inverse design validation run is performed using simply the standard objective function (Equation 4.10) with the adaptive GP as the surrogate model and is shown in Figure 5.16. As described earlier in Section 5.2, this is performed using 20 validation designs with multi-start optimization using the L-BFGS-B optimizer. The results are only discussed for the example of CTC since it is the most non-linear and has the highest variance. The results for the other material properties are omitted from the discussions but their respective results can be seen in Figure 5.17 and generalized from the discussions for CTC.

Figure 5.16a shows that the optimized microstructures now tightly fit within the known ground truth data for volume fractions. When the standard objective function was used with the GPR trained on the offline data for CTC (Figure 5.5a), a lot of the structures were out of the known ranges. This has been solved with the adaptive GP. Figure 5.16b shows the fit of the optimized microstructures in the 3D space with the adaptive GP. It is seen that the surrogate function plane now generalizes the region of infeasible microstructures well (top-right part of the plane). This was the main issue with the offline data-based GP (Figure 5.5b) where the region was seen to fold under and the resulting optimized microstructures fell out of known volume fractions from the ground truth. Additionally, the optimized microstructures are uniformly well-distributed throughout the parameter space for the adaptive GP (Figure 5.16b). With the modified objection function (Equation 4.12) using the offline data-based GP surrogate, the optimized microstructures for CTC (from Figure 5.7b) were less well-distributed and showed weaker exploration. This shows that a better quality and possibly a higher number of multiple solutions were found with the adaptive GP when compared to results for the modified objective function. It may be arguable that the suggestion of some infeasible microstructures occurs with the adaptive GPs, but this is acceptable since these microstructures now do not misfit on the volume fractions of the ground truth and show misleading designs. When analyzed in real-time applications, domain experts can identify infeasible structures and discard them from solution sets. If it is really required to discard infeasible microstructures during optimizations, the modified objective function can be used with the adaptive GP surrogate model.

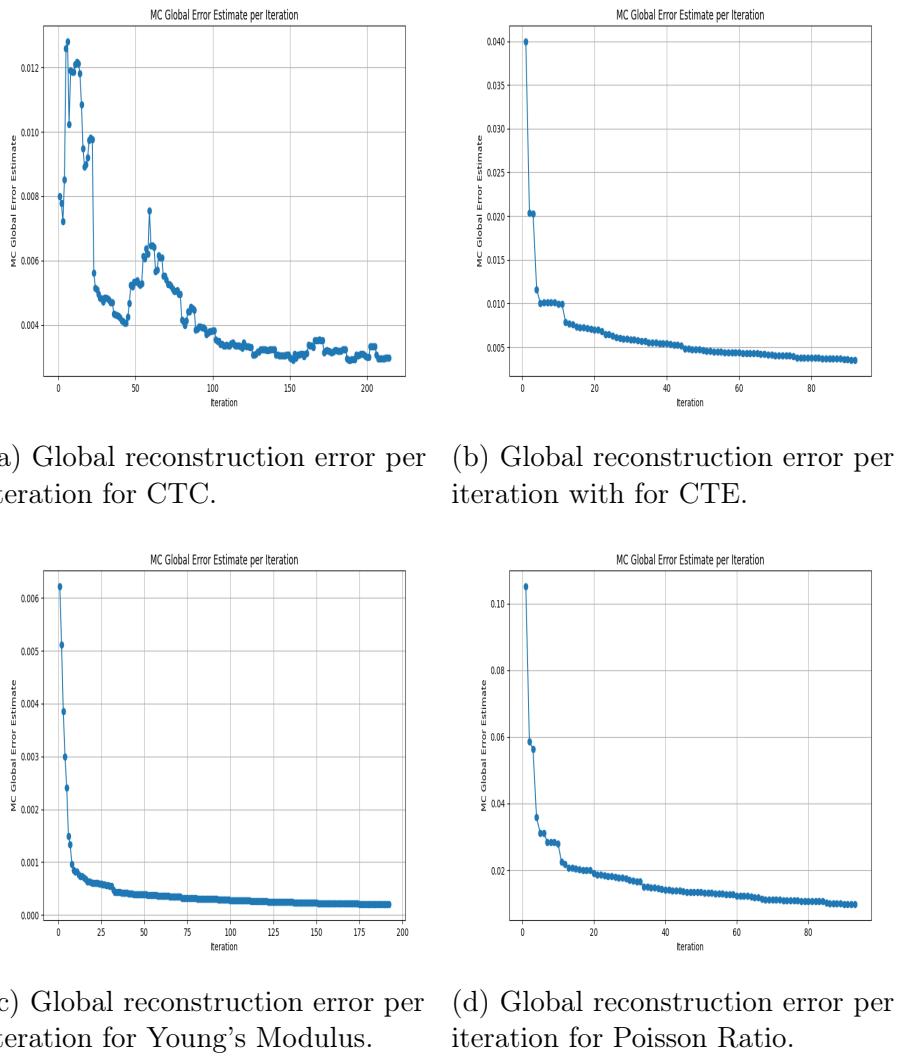


Figure 5.15: Global reconstruction error per iteration with adaptive GP.

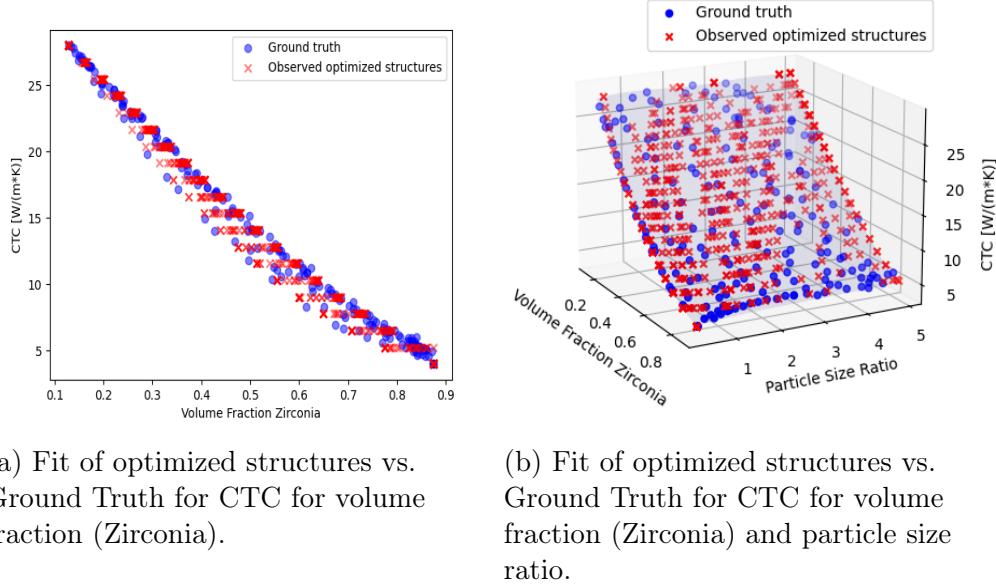


Figure 5.16: Visualisation of optimized microstructures for CTC with the GP trained with adaptive approach, shaded region shows the GP prediction plane \bar{f}_* .

The inverse design metrics for CTC are shown in Table 5.8. In this case, optimizations took 0.7 seconds and showed an RRMSE of $1.1e - 2$. This can be compared to the kernel comparison metrics for CTC from Table 5.6 which was performed with the GPR with offline data, where *RBF + White* kernel had shown a computation time of 12.3s and an RRMSE of $1.12e - 2$. The adaptive GPs greatly improve the computation times. This is due to the use of a smaller amount of data (from 450 to 100 training samples) and the use of the standard objective function and its gradient function (Equation 4.11) which are computationally less intensive. The same can be seen in Table 5.8 for the other material properties as well, where the computational times are much faster compared to the GPR with offline data.

Through the results in this section, it can be seen that the inverse modeling with adaptive GPs using the standard objective function performs better than the inverse modeling with both standard and modified objective functions for offline data-based GPs in terms of inverse design quality and computational speeds. Therefore, its use in future real-time applications is highly recommended.

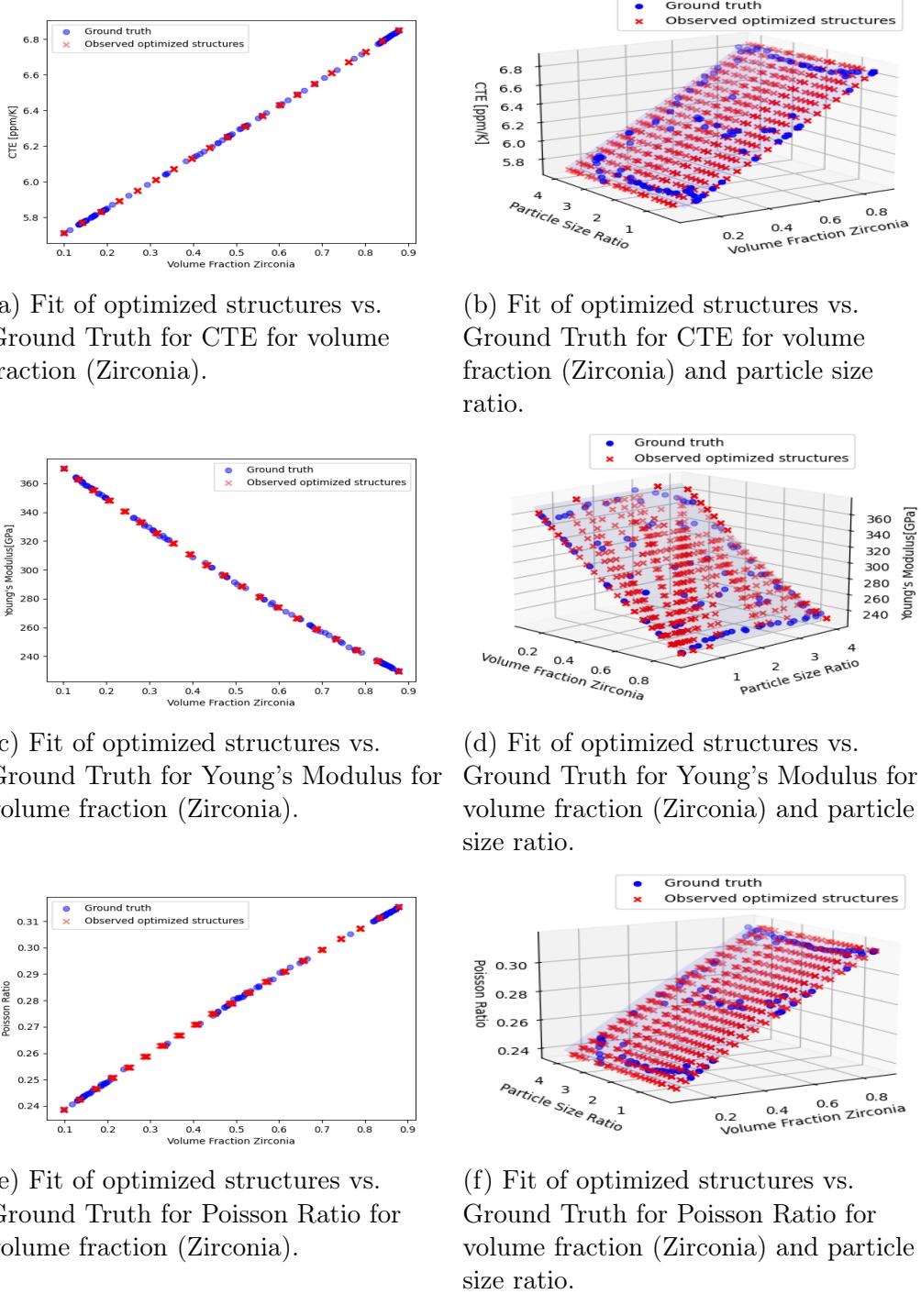


Figure 5.17: Visualisation of optimized microstructures for material properties with the GP trained with adaptive approach, shaded region shows the GP prediction plane \bar{f}_* .

Property	RR^2	RRMSE	C.Time
Adaptive GP			
CTC	1.00	1.06e-02	0.7s
CTE	1.00	1.85e-04	0.5s
Young's Modulus	1.00	2.8e-02	0.7s
Poisson Ratio	1.00	6.7e-05	0.5s

Table 5.8: Inverse Design Performance Metrics with Adaptive GP.

Chapter 6

Conclusions and Future Work

This thesis showed the effective study and application of inverse design in microstructures using GP surrogates that model the forward simulations developed by Pirkelmann et al. [8] for multi-phase ceramics. It was shown that effective GPRs can be constructed to model surrogate functions for various material properties such as CTE, CTC, Young's Modulus, and Poisson Ratio. It was shown that various factors need to be considered when modeling GP surrogates. Through extensive hyperparameter testing, it was found that for data that is known to have inherent noise, the use of a *White* kernel is necessary to achieve high accuracies. Additionally, normalization of the target variables can help when the ranges of the target values are high, such as for Young's Modulus.

Through a case study in inverse modeling, it was seen that GP surrogate functions face some limitations in generalization for regions of low data density. This causes the standard objective function (Equation 4.10) to inaccurately suggest infeasible microstructure designs for some desired material properties that are misaligned from the known ground truth. To remedy this, the modified objective function (Equation 4.12) can be used instead that avoids unknown regions of the design space for finding solutions to inverse design problems. Through this, a reliable inverse modeling methodology is obtained that returns highly accurate microstructure designs. It was also shown that the L-BFGS-B optimizer works reliably for applications in inverse designs as a gradient-based optimizer compared to other gradient-based and gradient-free optimizers.

Ensemble methods such as GBRs that were used in previous works were shown to be ineffective for inverse design due to their inherent nature of mod-

eling surrogates for forward functions in a staggered manner, which leads to regions of zero-value gradients and causes them to be unsuitable for gradient-based optimizations. Instead, the use of GPRs, which can model smooth surrogate functions whose analytical gradients can be easily derived is far more suitable for inverse design applications in ZTA/ATZ ceramics.

It was also shown through extensive analysis of kernels that the *DotProduct* kernel and its combinations with other kernels are unsuitable in inverse design applications. Instead, a safe and reliable option for applications in inverse design that shows high accuracy and computation speeds is the *RBF + White* kernel. The *RationalQuadratic* and *Matern* kernels as sum-kernels with the *White* kernel also show similar performance, and it may be worthwhile to perform an extensive analysis on a case-by-case basis for applications in other domains. It was also shown that the use of analytical gradients significantly outperforms approximated gradients as they are computationally much faster and more accurate as shown with inverse design metrics. This suggests that it is always better to choose surrogate models whose gradients can be analytically calculated, such as the GPR, as it can lead to more accurate inverse designs.

Direct inverse training (DIT) methodology was shown to be a possible and simpler alternative to the optimization-based inverse design method by training surrogate functions directly from material properties to microstructural features. The RF regressor was shown to be a good ML method from comparisons to NNs and GPs for DIT. However, its drawback lies in its inability to suggest multiple solutions for inverse design. A possible future work to remedy this would be to build multiple DIT models that each suggest different microstructures by intelligently partitioning the available data in non-overlapping subspaces within the microstructural feature space.

It was also shown that adaptive GPR training is an elegant approach to leverage online FEM simulations for building highly effective surrogate models for multi-phase ceramics. GP surrogates built through the adaptive approach usually require less than 100 training samples to achieve very high accuracies. The use of adaptive GP as a surrogate model in an optimization-based inverse design method shows that it performs significantly faster than GP surrogates trained with larger offline data from random sampling with similar accuracies. It also addresses limitations of generalizability in GPRs with inverse design by selecting training samples that minimize a global reconstruction error metric. This ensures that microstructures suggested through optimizations agree strongly with the true forward function. It was shown that inverse design with the adaptive GP shows a wider range of optimal mi-

crostructure design solutions with just the standard objective function and is recommended for use in future real-time applications.

A further improvement to the adaptive GP would be to incorporate the work model [5] into the training process along with determination of the true variances of adaptively added microstructural training samples. The work model will then be able to improve microstructures that show high variances by spending some computational budget from an overall budget to improve the resolution of the simulations and decrease the variance seen in the microstructural training samples. This can further improve the performance and quality of the GPR surrogates and its subsequent inverse design applications.

References

- [1] Jannick Kuhn et al.: Identifying material parameters in crystal plasticity by Bayesian optimization. In: *Optimization and Engineering* 23.3 (2022), pp. 1489–1523. ISSN: 1573-2924. DOI: 10.1007/s11081-021-09663-7. URL: <https://doi.org/10.1007/s11081-021-09663-7>.
- [2] Y. Liu, X. Peng, and Y. Qin: FE simulation for concurrent design and manufacture of automotive sheet-metal parts. In: *Journal of Materials Processing Technology* 150.1 (2004), pp. 145–150. ISSN: 0924-0136. DOI: <https://doi.org/10.1016/j.jmatprotec.2004.01.050>. URL: <https://www.sciencedirect.com/science/article/pii/S092401360400086X>.
- [3] Baekjun Kim, Sangwon Lee, and Jihan Kim: Inverse design of porous materials using artificial neural networks. In: *Science Advances* 6.1 (2020), eaax9324. DOI: 10.1126/sciadv.aax9324. eprint: <https://www.science.org/doi/pdf/10.1126/sciadv.aax9324>. URL: <https://www.science.org/doi/abs/10.1126/sciadv.aax9324>.
- [4] Colin Fyfe, Tzai Der Wang, and Shang Jen Chuang: Comparing Gaussian Processes and Artificial Neural Networks for Forecasting. In: *Proceedings of the 9th Joint International Conference on Information Sciences (JCIS-06)*. Atlantis Press, (2006/10), pp. 29–32. ISBN: 978-90-78677-01-7. DOI: 10.2991/jcis.2006.7. URL: <https://doi.org/10.2991/jcis.2006.7>.
- [5] Phillip Semler and Martin Weiser: *Adaptive Gaussian Process Regression for Efficient Building of Surrogate Models in Inverse Problems*. (2023). arXiv: 2303.05824 [math.NA].
- [6] Thomas M. Müller: Computergestütztes Materialdesign: Mikrostruktur und elektrische Eigenschaften von Zirkoniumdioxid-Aluminiumoxid Keramiken. doctoral thesis. Universität Würzburg, (2013).
- [7] M. Iuga and F. Raether: FEM simulations of microstructure effects on thermoelastic properties of sintered ceramics. In: *Journal of the European Ceramic Society* 27.2 (2007). Refereed Reports IX Conference

- Exhibition of the European Ceramic Society, pp. 511–516. ISSN: 0955-2219. DOI: <https://doi.org/10.1016/j.jeurceramsoc.2006.04.072>. URL: <https://www.sciencedirect.com/science/article/pii/S095522190600255X>.
- [8] S. Pirkelmann, F. Raether, and G. Seifert: Top-down material design of multi-phase ceramics. In: *Open Ceramics* 9 (2022), p. 100211. ISSN: 2666-5395. DOI: <https://doi.org/10.1016/j.oceram.2021.100211>. URL: <https://www.sciencedirect.com/science/article/pii/S2666539521001577>.
- [9] Carl Edward Rasmussen and Christopher K. I. Williams: *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, (2006), pp. I–XVIII, 1–248. ISBN: 026218253X.
- [10] Jie Wang: *An Intuitive Tutorial to Gaussian Processes Regression*. (2022). arXiv: 2009.10862 [stat.ML].
- [11] Zoubin Ghahramani: A Tutorial on Gaussian Processes (or why I don't use SVMs). In: (2011). URL: <https://api.semanticscholar.org/CorpusID:61496011>.
- [12] Philipp-Immanuel Schneider et al.: Using Gaussian process regression for efficient parameter reconstruction. In: *Metrology, Inspection, and Process Control for Microlithography XXXIII*. Ed. by Ofer Adan and Vladimir A. Ukraintsev. SPIE, (2019). DOI: 10.1117/12.2513268. URL: <https://doi.org/10.1117%2F12.2513268>.
- [13] Akshay Iyer et al.: *Data-Centric Mixed-Variable Bayesian Optimization For Materials Design*. (2019). arXiv: 1907.02577 [physics.comp-ph].
- [14] Yuki Sato, Kenji Kawano, and Hajime Igarashi: Direct Inverse Modeling for Electromagnetic Components Using Gaussian Kernel Regression. In: *IEEE Transactions on Magnetics* 58.5 (2022), pp. 1–8. DOI: 10.1109/TMAG.2022.3152024.
- [15] Hanita Daud et al.: A Novel Methodology for Hydrocarbon Depth Prediction in Seabed Logging: Gaussian Process-Based Inverse Modeling of Electromagnetic Data. In: *Applied Sciences* 11.4 (2021). ISSN: 2076-3417. DOI: 10.3390/app11041492. URL: <https://www.mdpi.com/2076-3417/11/4/1492>.
- [16] Robert Saunders et al.: Mechanical behavior predictions of additively manufactured microstructures using functional Gaussian process surrogates. In: *npj Computational Materials* 7 (2021). DOI: 10.1038/s41524-021-00548-y.
- [17] K.G.S. Gopinath, Soumen Pal, and Pankaj Tambe: Prediction of Hardness and Fracture Toughness in Liquid-Phase-Sintered Alumina System Using Gaussian Process Regression and Minimax Probability Machine Regression. In: *Materials Today: Proceedings* 5.5, Part 2 (2018).

- International Conference on Materials Manufacturing and Modelling, ICMMM - 2017, 9 - 11, March 2017, pp. 12223–12232. ISSN: 2214-7853. DOI: <https://doi.org/10.1016/j.matpr.2018.02.199>. URL: <https://www.sciencedirect.com/science/article/pii/S2214785318304024>.
- [18] Ruoqian Liu et al.: A predictive machine learning approach for microstructure optimization and materials design. In: *Scientific Reports* 5 (2015). URL: <https://api.semanticscholar.org/CorpusID:11267237>.
 - [19] Ren Kai Tan, Nevin L. Zhang, and Wenjing Ye: A Deep Learning-based Method for the Design of Microstructural Materials. In: *Struct. Multi-discip. Optim.* 61.4 (2020), pp. 1417–1438. ISSN: 1615-147X. DOI: [10.1007/s00158-019-02424-2](https://doi.org/10.1007/s00158-019-02424-2). URL: <https://doi.org/10.1007/s00158-019-02424-2>.
 - [20] L. Cai et al.: Surrogate models based on machine learning methods for parameter estimation of left ventricular myocardium. In: *Royal Society Open Science* 8 (1 2021), p. 201121. DOI: [10.1098/rsos.201121](https://doi.org/10.1098/rsos.201121).
 - [21] Keith D. Humfeld et al.: A machine learning framework for real-time inverse modeling and multi-objective process optimization of composites for active manufacturing control. In: *Composites Part B: Engineering* 223 (2021), p. 109150. ISSN: 1359-8368. DOI: <https://doi.org/10.1016/j.compositesb.2021.109150>. URL: <https://www.sciencedirect.com/science/article/pii/S135983682100531X>.
 - [22] Tim Head et al.: *scikit-optimize/scikit-optimize*. Version v0.9.0. (2021). DOI: [10.5281/zenodo.5565057](https://doi.org/10.5281/zenodo.5565057). URL: <https://doi.org/10.5281/zenodo.5565057>.
 - [23] F. Pedregosa et al.: Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
 - [24] Ronald Iman: Latin Hypercube Sampling. In: (1999). DOI: [10.1002/9780470061596.risk0299](https://doi.org/10.1002/9780470061596.risk0299).
 - [25] Gaël Varoquaux: *Mathematical optimization: finding minima of functions*. Scipy Lecture Notes. Accessed: 2024-01-05. URL: https://scipy-lectures.org/advanced/mathematical_optimization/.
 - [26] Dong C. Liu and Jorge Nocedal: On the limited memory BFGS method for large scale optimization. In: *Mathematical Programming* 45.1 (1989), pp. 503–528. ISSN: 1436-4646. DOI: [10.1007/BF01589116](https://doi.org/10.1007/BF01589116). URL: <https://doi.org/10.1007/BF01589116>.
 - [27] M. J. D. Powell: An efficient method for finding the minimum of a function of several variables without calculating derivatives. In: *The Computer Journal* 7.2 (1964), pp. 155–162. ISSN: 0010-4620. DOI: [10.1093/comjnl/7.2.155](https://doi.org/10.1093/comjnl/7.2.155). eprint: <https://academic.oup.com/comjnl/>

- article-pdf/7/2/155/959784/070155.pdf. URL: <https://doi.org/10.1093/comjnl/7.2.155>.
- [28] J. A. Nelder and R. Mead: A Simplex Method for Function Minimization. In: *The Computer Journal* 7.4 (1965), pp. 308–313. ISSN: 0010-4620. DOI: 10.1093/comjnl/7.4.308. eprint: <https://academic.oup.com/comjnl/article-pdf/7/4/308/1013182/7-4-308.pdf>. URL: <https://doi.org/10.1093/comjnl/7.4.308>.
- [29] D. Kraft: *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, (1988). URL: <https://books.google.de/books?id=4rKaGwAACAAJ>.

Declaration

I hereby declare that I have completed this thesis independently and without using any tools other than those specified. All passages taken literally or analogously from published or unpublished writings are marked as such. I have not already submitted the work in the same or a similar form to obtain an academic degree.

Bayreuth, 15.01.2024

Rayan Hamid Mohiuddin

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit habe ich in gleicher oder ähnlicher Form nicht bereits zur Erlangung eines akademischen Grades eingereicht.

Bayreuth, den 15.01.2024

Rayan Hamid Mohiuddin

