Scalable Online Nonlinear Goal-Oriented Inference with Physics-Informed Maps

by

Harriet Li
M.S., Massachusetts Institute of Technology (2015)
B.S., Massachusetts Institute of Technology (2013)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2019

Author . . . .
Department of Aeronautics and Astronautics
May 23, 2019

Certified by. . . . .
Karen Willcox
Professor of Aerospace Engineering and Engineering Mechanics
The University of Texas at Austin
Thesis Supervisor

Certified by. . . . .
Youssef Marzouk
Associate Professor of Aeronautics and Astronautics
Thesis Committee Member

Certified by. . . . .
Bart van Bloemen Waanders
Distinguished Member of Technical Staff
Sandia National Laboratories
Thesis Committee Member

Accepted by . . . .
Sertac Karaman
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Student Committee

Scalable Online Nonlinear Goal-Oriented Inference with Physics-Informed Maps

by

Harriet Li

Submitted to the Department of Aeronautics and Astronautics
on May 23, 2019, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis develops a physics-informed k-nearest neighbors approach, which draws from both physics-based modeling and data-driven machine learning. In doing so, our method achieves robustness and increased accuracy with small datasets, while being cheap to apply. Our method tackles the challenges of high-dimensional inverse problems governed by complex physical models. Such inverse problems arise in important engineering applications, such as heat transfer, medical and structural imaging, and contaminant control. In particular, we consider the goal-oriented inverse problem setting, where unknown model parameters are inferred from observations in order to calculate some low-dimensional quantity of interest (QoI). When computational resources and/or time are limited, it is infeasible to solve the full inverse problem for inferred parameters to obtain the QoI.

This thesis describes an algorithm that bypasses solving the inverse problem, instead directly giving rapid QoI estimates for observations. We generate a library of physics-informed maps based on local approximations to the goal-oriented inverse problem. Applying tensor decompositions to these approximate problems gives compact multilinear physics-informed maps. These maps are calculated and stored in an offline preparatory phase, and then applied to online observations to give rapid QoI estimates. This thesis also describes tailored active learning algorithms, which efficiently choose training points in observation space at which to generate these physics-informed maps. This improves the online prediction performance given a limited offline computational and/or storage budget.

We demonstrate our rapid QoI estimation and active learning algorithms on a quality-control problem for additive manufacturing. The proposed physics-informed approach achieves 5% relative QoI error in 0.1% of the time to solve the full inverse problem. Our physics-informed mappings give a third of the QoI estimate error that black-box regression methods do for small datasets, and are more robust when the offline dataset does not well represent the online test points. The tailored active learning algorithms produce datasets that reduce maximum QoI error by 25% and misclassification by 15%, compared to randomly chosen datasets.

Thesis Supervisor: Karen Willcox
Title: Professor of Aerospace Engineering and Engineering Mechanics
The University of Texas at Austin

Thesis Committee Member: Youssef Marzouk
Title: Associate Professor of Aeronautics and Astronautics

Thesis Committee Member: Bart van Bloemen Waanders
Title: Distinguished Member of Technical Staff
Sandia National Laboratories

Acknowledgments

This work would not have been possible without the guidance and support of many others. May I one day give back as much as I have received.

I am deeply grateful to Professor Karen Willcox, for being so much more than just an academic advisor; for her patience and encouragement and caring; for scholarly advice when I was stuck with no idea or too many ideas; for seeing value in my work when I did not; for being a role model in more than just academic matters. I count myself very fortunate to have become her undergraduate advisee back in sophomore year, when graduate studies in computational engineering was not even a future possibility I had entertained. Thank you for these past nine years.

I thank Dr. Bart van Bloemen Waanders, for the educational summer internships and for keeping in touch afterwards, for the encouragement when I kept breaking things, and for the insight into the connections between different fields. I thank Professor Youssef Marzouk, for the thoughtful questions and suggestions, and for making UQ less terrifying and more fascinating. Thank you both for serving on my committee.

I thank Professor Tiangang Cui and Dr. Boris Kramer, for reading over the manuscript that I really thought would be under 100 pages. Your comments have helped improve this thesis even before I started writing.

I thank my parents and my sister for their love and support. I am sorry my Chinese vocabulary has shrunk to basic greetings and ‘math is hard (but I like it)’.

I thank my roommates, for making my nest a place to rest, and for eating the products of my stress-baking. I tried to make them healthy and nutritious, really. I thank old friends, for continuing to stay in touch even though teleportation has not yet been made widely accessible.

I thank my peers in the ACDL and the CSE offshoot, for contributing to a welcoming environment, even if we exchanged few words and I kept forgetting your face and/or name every time you got a haircut or went on vacation or changed your shirt. I have a penchant for dropping eaves. Special thanks to Elizabeth Qian and Zheng

Wang, for talking with me, a lot.

I thank the people of the clubs I have had the pleasure of being a part of: the MIT Science Fiction Society (MITSFS), Graduate Women in Aerospace Engineering (GWAE), the Boston Lightsaber Stage Combat Club (BLSCC), and the Alpha Chi chapter of Alpha Phi Omega (AXcAPO).

I thank Jean Sofronas and Joyce Light, for doing so much more than administrative assisting. Thank you for the smiles (mine and yours).

This work was supported in part by the NSF Computational and Data-Enabled Science and Engineering Program grant CNS-050186 and the US Department of Energy Office of Advanced Scientific Computing Research (ASCR) Applied Mathematics Program, awards DE-FG02-08ER2585 and DE-SC0009297, as part of the DiaMonD Multifaceted Mathematics Integrated Capability Center, and the MIT-SUTD International Design Center.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In scientific and engineering contexts, observations about a physical system can be used to inform decisions or actions that affect it. The observations are used to calibrate models of the physical system, which can then be used to predict quantities relevant to the decision. When these decisions need to be made quickly and/or on platforms with limited computational power, the calibration step may be infeasibly expensive. In this thesis, we develop methods that accelerate the process of obtaining these prediction quantities from observations.

In this chapter, we motivate and give context to this problem of informing decisions with observations, and review the literature on relevant approaches. In Section 1.1, we frame the problem of using observations to inform decisions, and motivate the need for algorithms that accelerate this process. In Section 1.2, we discuss physics-based approaches that make strategic approximations to the expensive model calibration. In Section 1.3, we discuss black-box data-based approaches, many of which give cheaper estimates of the prediction quantities for observations, but require large training sets for accuracy. Our proposed algorithm will draw from both physics-based and data-based approaches in order to combine their strengths and ameliorate their drawbacks. We review existing combined approaches in Section 1.4. To improve QoI estimation given a limited offline and/or storage budget, we will draw from active learning approaches, discussed in Section 1.5, to efficiently choose our training sets. In Section 1.6, we introduce our thesis objectives and outline the remaining chapters of the thesis. In Section 1.7, we define notation and terminology that will be used in developing our algorithm.

1.1 Context and Motivation

In scientific and engineering contexts, observations about a physical system can be used to inform decisions or actions regarding it. The observations are used to infer the parameters of a mathematical model of the physical system, and the inferred parameters are then used to compute quantities relevant to the decision or action. For example, one might load a manufactured part and gather observations of its response to infer its internal structure, which can then be used to predict its behavior in extreme cases and decide its acceptability. Another example occurs in imprint lithography for nanofabrication, where measurements from optical sensing techniques are used to infer the presence of contaminant particles on a candidate substrate wafer [61, 125]. The inferred contamination state can be used to predict damage to an expensive template from imprinting on the wafer, and decide whether to proceed with the imprinting process.

Solving the inverse problem for the parameters based on observations presents a major computational challenge, involving repeated evaluations of the mathematical model. For our target engineering and scientific problems of interest, each model evaluation is expensive. In many cases the unknown parameters are spatially and/or temporally distributed, leading to discretized parameter spaces of high dimension, which require many model evaluations to explore.

We frame the process of obtaining a decision-relevant prediction quantity from observations as a *goal-oriented inverse problem*. In the goal-oriented inverse problem, the purpose of inferring parameters is to use them in predicting some low-dimensional prediction quantity of interest (QoI). This prediction quantity may then be used to inform a decision; for example, one may infer characteristics of a system in order to predict the outcome of a potential action under evaluation (e.g., remediation strategy [101], injecting carbon [86]), or to inform a decision target (e.g., future ice mass flux [69]). The relation between observations, parameters, and prediction QoI is depicted in Figure 1-1. In contrast to the inverse problem, the inferred parameters in the goal-oriented inverse problem are only a means to an end (the QoI), and are not themselves of interest.

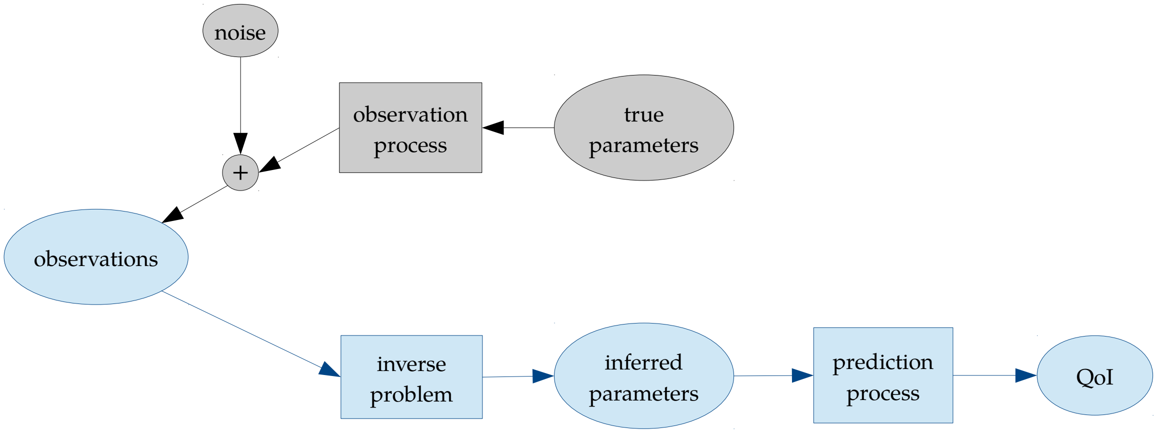


Figure 1-1: The observations are modeled as noisy outputs of the model with a set of true parameters. Given observations, the goal-oriented inverse problem (in blue) seeks to infer parameters (inverse problem) from the observations in order to calculate a prediction QoI (prediction process).

In settings where one has limited time and/or computational resources in which to calculate the QoI from observations, it may not be feasible to solve the inverse problem. Such time- and computation-constrained settings occur, for example, when an engineer wishes to make quick optimization calculations during a manufacturing process using a mobile device [121], or when an unmanned aerial vehicle needs to decide its flight path based on structural health observations obtained during flight [124], using only its onboard computational resources. The issue of limited computational resources is compounded if one needs a QoI for each of multiple observations. For example, one may need to estimate the QoI from observations of a single system at each of many points in time (e.g., structural health monitoring), or observations corresponding to different instances of similar systems (e.g., testing each product of manufacturing process for quality assurance). We seek an algorithm that enables rapid QoI estimation from observations, in settings with limited computational resources. We assume an offline preparatory phase, where one can make expensive calculations and store quantities, followed by an online resource-constrained phase, where one uses these stored quantities to enable rapid QoI estimation for online observations.

1.2 Accelerating Inverse Solution for QoI

To obtain a QoI from observations, one solves an inverse problem for model parameters that best reproduce the observations, subject to any additional user-imposed constraints or penalties. These inferred parameters are then used to compute the QoI. When the unknown parameters are discretized fields that are spatially and/or temporally distributed, the parameter space can become very high-dimensional. The cost of obtaining the QoI is then dominated by the cost of solving a large-scale optimization problem. To accelerate the process of obtaining the QoI from observations, one can use techniques for accelerating the solution of large-scale optimization problems. For example, one can use preconditioned Krylov subspace methods [3,17,75]. If available, cheaper surrogate models can also be used to drastically reduce the need for expensive model evaluations. This approach has been applied to optimization problems [2, 49] and statistical inverse problems [31, 37, 41, 48]; a review of multifidelity model management for outer-loop applications can be found in [106].

Further acceleration can be achieved by taking advantage of the particular structure of the inverse problem. Since the observations are often sparse, noisy, and indirect, they are insufficiently informative to fully determine the parameters. Regularization, in the form of a penalty term or constraint on the solution space, introduces additional information to give unique inferred parameters for observations. Even with regularization, the sparsity of observed information introduces low-dimensional structure in the mapping between observations and the inferred parameters. In Bayesian inference, where one infers a posterior distribution of the parameter instead of a single estimated value, existing approaches have exploited this low-dimensional structure to efficiently sample from the parameter posterior [24, 40, 52, 92]. In the deterministic setting, existing approaches have reduced the cost of solving the inverse problem by approximating it; the forward model, which relates parameters to their predicted observations, is replaced with a cheaper alternative. In [9,25,45,64,66,84,87,100], this cheaper alternative is a reduced-order model created by projecting the state and/or parameter space of the governing equations into a reduced space. The reduced space is chosen to efficiently approximate the relationship between parameters and their predicted observations, and thus efficiently approximate the inferred parameters.

A greater reduction in computational cost can be achieved by considering the ultimate goal when approximating the inverse problem. In the goal-oriented inverse problem, not only are the parameters incompletely informed by the observations, but their full resolution is also unnecessary to preserve the often low-dimensional QoI. Rather than constructing a reduced-order or surrogate model to accurately infer the parameters, one sacrifices accuracy in the inferred parameters in a controlled fashion, so as to minimize error in the QoI. This controlled trade-off can be done, for example, by constraining the inferred parameters to a particular low-dimensional subspace; in the case of a linear goal-oriented inverse problem, [85] show that one can choose such a low-dimensional subspace without sacrificing any accuracy in the QoI. Goal-oriented mesh refinement [10] and goal-oriented model adaptivity [82] derive an error estimate for the QoI from solving the (cheaper) inverse problem with a coarser mesh or lower-fidelity models; this error is relative to the QoI from the (expensive) inverse problem in infinite dimensions, or with the highest-fidelity model. The error estimate is used to guide adaptive mesh and/or model refinement, resulting in an approximate inverse problem that is less expensive to solve, but which gives an accurate QoI estimate.

By approximating the inverse problem in a principled fashion, one is able to obtain an accurate QoI estimate from observations at reduced cost by solving a lower-dimensional and/or less nonlinear optimization problem. However, this approach still requires that an optimization problem, albeit a cheaper one, be solved for each set of online observations for which we wish to estimate a QoI. For settings where one does not have the time and/or computational resources to solve even an approximate inverse problem, it may be appropriate to choose a black-box data-based approach, which can give a cheaper QoI estimate.

1.3 Black-box Data-based Approaches

For a given input, black-box data-based methods can be used to obtain an output estimate by applying a direct mapping that is cheap to evaluate. This mapping is determined ('trained') based on a set of known input-output pairs (training data), with its form and/or parameters chosen to match (exactly or with minimal penalty) the training pairs. The trained mapping is black-box in that it seeks to emulate the input-output relationship that generated the training set without requiring any additional information about its structure. Such black-box data-based methods can be found in various communities. Engineering applications have traditionally utilized a wide range of statistical and machine learning approaches, with methods such as response surfaces [20,21,28,74,114], kriging or Gaussian process regression [78,93,109], radial basis function approximation [22, 107, 146], neural networks [94, 113, 115], k-nearest neighbors [29, 39, 51], and support vector machines (SVMs) [19, 38, 135].

Black-box data-based methods are both flexible (able to represent a wide range of input-output relationships) and cheap to evaluate (give rapid output estimate for given input). These characteristics make them especially useful when the true input-output relationship is unknown or only known implicitly. The flexibility and relatively cheap evaluation costs of data-based methods have been utilized in the setting of computing prediction QoI from observations, where the observations and QoI are connected by models with a shared set of parameters. In [90, 128], an observation-to-QoI mapping is trained in the context of structural health monitoring, mapping strain measurements to coefficients describing the capability or damage state. In these works, the "true" QoI for observations is not obtained through solving an inverse problem. Instead, the training observation-QoI pairs are generated by choosing a set of parameter values, representing possible or expected damage and health states. For each parameter instance, a corresponding pair of observations and QoI is generated through forward model simulations. In contrast, by relating observations to their QoI through a goal-oriented inverse problem, we are able to make use of past observations (e.g., from past products of manufacturing process) as training points containing information about expected online observations.

Due to their flexible representation and relatively cheap evaluation costs, black-box data-based methods are useful for accelerating the process of obtaining QoI estimates from observations informing parameters of expensive models. However, these data-based mappings often require large training sets in order to give accurate QoI estimates. Each training point requires evaluation of the expensive model, so a large training set is expensive to obtain. The need for large training sets can be reduced by introducing information about the structure of the goal-oriented inverse problem, to supplement the information contained in the observation-QoI training pairs.

1.4 Combining Physics- and Data-based Approaches

In Sections 1.2 and 1.3, we discussed two types of approaches to accelerating the process of obtaining a QoI from observations informing the parameters of an expensive model. In Section 1.2, we discussed physics-based approaches that approximate the inverse problem while preserving its goal; the approximate goal-oriented inverse problem can be solved for different observations to obtain their QoI. This contrasts with the black-box data-based approaches discussed in Section 1.3; these methods use no knowledge of the structure of the goal-oriented inverse problem, but give cheap mappings from observations to estimated QoI. These methods have the drawback of requiring large expensive training sets for accuracy. Combining physics- and data-based approaches can produce methods which combine their strengths and ameliorate their respective weaknesses.

Methods combining physics- and data-based approaches have been applied to a variety of problems, with a spectrum of emphasis on either the physics-based or data-based aspects. ‘Theory-guided data-science’ is introduced in [72] as a paradigm for combining data and scientific knowledge to create generalizable and interpretable models, and gives a review of such combined approaches. The loosest coupling can be found in methods where black-box data-based mappings are trained on the results of numerical physics-based model simulations. Results from data-based mappings can also be post-processed with physics-based models; for example, data-based mappings in [62] are used to cheaply generate candidate material structures that are then pruned using more expensive physics-based simulations.

In more tightly coupled methods, physical knowledge can inform the structure and parameters of the data-based mapping, or how it will be trained. For example, domain knowledge has been used to choose a feature space [90, 119, 128], and to encourage (through penalty terms) or enforce (through restrictions on mapping structure and/or parameters) physically meaningful properties and behaviors [8, 83, 89, 91, 127]. Data-based mappings can also be used to augment physics-based models; for example, data-based terms can serve as a correction to projected reduced-order [13, 14, 47, 143, 148], or otherwise simplified [123, 147], physical models, capturing missing dynamics or other phenomena.

We extend this notion of combining physics- and data-based approaches to the goal-oriented inverse problem setting in order to obtain favorable characteristics from both. We construct a library of observation-to-QoI mappings, each based on local approximations to the goal-oriented inverse problem about a 'training' point. Each local mapping is informed by the structure and physics of the goal-oriented inverse problem, but has the low evaluation cost of (data-based) polynomial regression. Compared to purely data-based approaches, the additional physics information in the local mappings results in fewer training points being needed for a desired QoI estimate accuracy. However, the accuracy of the QoI estimate still depends on the quality of the training set. Since each training point requires expensive model evaluations, it is desirable to choose a training set efficiently. We review methods in the literature for efficiently querying expensive information in the next section.

1.5 Efficient Training with Active Learning

When a data-based mapping is trained to emulate an input-output relationship, its prediction accuracy is dependent on the quality of its training data. Large training sets can often give improved performance, but can also be expensive to obtain and utilize. Rather than randomly choose training points, one can iteratively use the mapping, trained on an initial subset of training points, to choose subsequent training points to best achieve the goal of accurate predictions.

Such techniques can be found in the active learning, optimal experimental design (OED), and Bayesian optimization literature. A review of active learning methods can be found in [117]. An introduction to OED [126] can be found in [99]; reviews of OED methods can be found in [46, 53]. Bayesian optimization [71, 79, 95, 152] focuses on the particular case where the emulated input-output relationship is an optimization objective function; a review can be found in [120]. In the contexts of active learning, OED, and Bayesian optimization, mappings are trained (fitted parametrically, or constructed nonparametrically) based on a set of known input-output points; these points are called the 'training set' or 'experimental data'. The true output (or 'label') for a given input point is obtained by 'querying', or performing an 'experiment', at the input point. Large training/data sets can be expensive, both due to the cost of performing many queries/experiments, and due to the cost of fitting the trained mapping to large training/data sets. Thus, choosing the training/data sets efficiently can reduce the costs of both obtaining and using them.

In active learning and OED, one chooses training/data points to (greedily or optimally) minimize some criterion, often a (local or integrated) measure of error or uncertainty in the mapping parameters or outputs. These criteria are calculated using the mapping trained/calibrated on the training/data points thus far available; one iterates between choosing training/data points and updating the mapping. For a class of mappings from which one chooses a member that best fits the known input-output pairs, different criteria can be computed and used for choosing training/data points. One possible criterion is the size of the version space (space of mappings consistent with known training pairs). For some classes of mappings, the size of their version space can be estimated, so they are conducive to active learning approaches that focus on reducing this space [42, 57, 131, 132]. For example, the version space for SVMs corresponds to the region of overlap between half-planes whose separating hyperplanes are defined by the training points; in [131], this property is used to choose subsequent training points that most nearly halve the version space and thus have the greatest potential to reduce the version space.

While halving the version space can lead to attractive prediction error decay rates with training set size [54], directly estimating the size of the version space is only tractable for particular classes of mappings. One can alternatively estimate, and seek to minimize, the error or uncertainty in the mapping output. If one is able to estimate the error or uncertainty at input points, and can characterize how adding a training point will affect the resulting trained mapping, then one can choose training points whose addition will minimize integrated error or variance. For example, in [30, 151], training points are chosen to minimize an integrated output error, using expressions for the asymptotic expansion of the error for certain classes of nonparametric regression. In [88], no direct error expression is available, so the error is estimated by describing possible outputs with a random field, and comparing them to the output estimate of the mapping when trained on potential training sets. In [34], training points are chosen to minimize integrated variance for Gaussian mixtures and locally weighted regression. If one can not characterize how adding a training point will affect the resulting trained mapping, then one can greedily select input points with the greatest error or uncertainty to add to the training set, as is done in [27,33,56,80,149].

1.6 Thesis Objectives and Outline

The process of making a decision based on observations of a physical system appears in many engineering applications, and can be framed as a goal-oriented inverse problem. In time and/or resource constrained settings, it is not feasible to infer for the parameters, especially when the parameters are high-dimensional and/or the model is expensive to evaluate. Both physics-based and black-box methods have been used to accelerate the process of obtaining the QoI for observations. Existing physics-based methods strategically approximate the inverse problem so that the parameters can be more cheaply inferred. However, estimating the QoI for given observations still requires solution of an optimization problem after the observations are obtained.

In contrast, black-box data-based methods produce explicit observation-to-QoI mappings that are cheap to apply, giving rapid QoI estimates for online observations. However, these mappings require many expensive offline training points for accuracy.

We seek a method to give rapid QoI estimates for online observations without requiring large expensive training sets. To this end, we take inspiration from (a) work in combining physics- and data-based approaches, and (b) work in active learning to efficiently select training points. We propose an algorithm that creates a library of multilinear maps based on local approximations to the goal-oriented inverse problem, where these local approximations can be efficiently chosen using our tailored active learning strategies. The maps can be calculated and stored in an offline preparatory phase, and applied to online observations to obtain a rapid QoI estimate.

The objectives of this thesis are to:

- develop an algorithm that bypasses the solution of an expensive, high-dimensional inverse problem to directly give rapid QoI estimates for observations;
- develop tailored active learning strategies for efficiently choosing offline evaluations at training observations;
- demonstrate through numerical experiments that the algorithm gives improved accuracy and robustness when compared to black-box data-based methods;
- demonstrate that the proposed active learning approaches can give improved QoI estimation and classification performance over random sampling.

The remainder of this thesis is organized as follows. Section 1.7 defines tensor terminology and notation that will be used in developing our algorithm. Chapter 2 describes the deterministic nonlinear goal-oriented inverse problem and derives our proposed algorithm. Chapter 3 describes active learning strategies for our algorithm. Chapter 4 describes numerical results from applying our algorithm to a goal-oriented tomography problem. Chapter 5 gives concluding remarks and directions for future work.

1.7 Tensor Terminology and Notation

Let $B \in \mathbb{R}^{p\times\cdots\times p}$ be an order-d symmetric tensor with length p along each of its d dimensions (or modes), with elements $B(i_1,\ldots,i_d) = b_{i_1\ldots i_d}$. A symmetric tensor is invariant under permutation of indices, and a diagonal tensor has nonzero elements only along its superdiagonal, i.e., $b_{i_1\ldots i_d} \neq 0$ only if $i_1 = \cdots = i_d$.

Following [77], let the n-mode tensor-vector product of tensor B and vector $v \in \mathbb{R}^p$ as $B \mathbin{\bar{\times}_n} v$ be denoted by

$$(B \mathbin{\bar{\times}_n} v)_{i_1\ldots i_{n-1}i_{n+1}\ldots i_d} = \sum_{i_n=1}^{p} b_{i_1\ldots i_d} v_{i_n}, \tag{1.1}$$

which gives an order-$(d-1)$ tensor of length p in each dimension. The n-mode tensor-matrix product of tensor B and matrix $U \in \mathbb{R}^{q\times p}$ is $B \times_n U$, which gives an order-d tensor with length q along its n-th dimension and length p along the others, with

$$(B \times_n U)_{i_1\ldots i_{n-1}ji_{n+1}\ldots i_d} = \sum_{i_n=1}^{p} b_{i_1\ldots i_d} u_{ji_n}. \tag{1.2}$$

Let the m-linear form $B(v^{(1)},\ldots,v^{(m)})$ denote a series of tensor-vector products, written

$$B(v^{(1)},\ldots,v^{(m)}) = B \mathbin{\bar{\times}_1} v^{(1)} \ldots \mathbin{\bar{\times}_m} v^{(m)}, \tag{1.3}$$

where $v^{(j)} \in \mathbb{R}^p$, for $j = 1,\ldots,m$. Note that when $m = d$, the multilinear form $B(v^{(1)},\ldots,v^{(d)})$ gives a scalar. Similarly, let $B(U^{(1)},\ldots,U^{(m)})$, $B^{(j)} \in \mathbb{R}^{q\times p}$ denote a series of tensor-matrix products

$$B(U^{(1)},\ldots,U^{(m)}) = B \times_1 U^{(1)} \ldots \times_m U^{(m)} \tag{1.4}$$

resulting in an order-d tensor of length q in the first m dimensions and of length p in the last $d-m$ dimensions.

The Frobenius norm of a tensor is defined by

$$\|B\|_F = \sqrt{\sum_{i_1=1}^{p} \cdots \sum_{i_d=1}^{p} b_{i_1 \ldots i_d}^2}, \tag{1.5}$$

analogous to the Frobenius matrix norm. Its spectral norm is defined by

$$\|B\|_{\text{spec}} = \sup_{v^{(1)},\ldots,v^{(d)} \neq \vec{0}} \left(\frac{|B(v^{(1)}, \ldots, v^{(d)})|}{\|v^{(1)}\|_2 \cdots \|v^{(d)}\|_2} \right). \tag{1.6}$$

Chapter 2

Nonlinear Goal-Oriented Inference with Physics-Informed Maps

In this chapter, we describe and derive our algorithm for rapidly obtaining a QoI estimate for given observations. Section 2.1 introduces and relates the linear and nonlinear goal-oriented inverse problems. Section 2.2 describes our algorithm for rapidly estimating the QoI corresponding to the inferred parameters of online observations. We then detail the different components of the algorithm. Section 2.3 describes local approximations to the goal-oriented inverse problem. Section 2.4 details different ways to build and store physics-informed maps corresponding to these local approximations. Section 2.5 describes an error estimate for the QoI of the approximate goal-oriented inverse problems. Section 2.6 describes how to choose maps corresponding to different local approximations from a library of maps, and how to combine their QoI estimates, given online observations. Section 2.7 summarizes the chapter and motivates the work in the next chapter.

2.1 Deterministic Goal-Oriented Inverse Problem

In Section 2.1.1, we review the linear goal-oriented inverse problem and the favorable properties of its analytically known observation-to-QoI map. In Section 2.1.2, we introduce the more general nonlinear goal-oriented inverse problem that is the focus of this thesis.

2.1.1 Linear Goal-Oriented Inverse Problem

In this subsection, we describe the linear goal-oriented inverse problem, for which there exists an observation-to-QoI map with desirable properties. We review this map and its favorable properties, which we aim to emulate in the nonlinear case.

The deterministic inverse problem seeks to infer parameters from (noisy, indirect) observations $y \in \mathbb{R}^{n_y}$. In the linear case, one can relate parameters to predicted observations through a linear mapping. When the observations are sparsely informative, the problem of seeking parameters to minimize the mismatch between predicted and observed observations has non-unique solutions. Using Tikhonov regularization to address this ill-posedness, the linear inverse problem is given by

$$x^\star = \arg\min_x \frac{1}{2}\|y - O_e x\|_2^2 + \frac{1}{2}\|Rx\|_2^2, \tag{2.1}$$

where $x^\star \in \mathbb{R}^{n_x}$ are the inferred parameters, $O_e \in \mathbb{R}^{n_y \times n_x}$ maps parameters to expected observations, and $\frac{1}{2}\|Rx\|_2^2$ is the Tikhonov regularization term with Tikhonov regularization matrix $R \in \mathbb{R}^{n_x \times n_x}$. In the *goal-oriented* inverse problem, the purpose of inferring parameters is to use them in predicting an n_z-dimensional QoI. In the linear case, the QoI is also linear in the parameters, and is described by

$$z^\star = O_p x^\star, \tag{2.2}$$

where $O_p \in \mathbb{R}^{n_z \times n_x}$.

For the linear goal-oriented inverse problem, one can combine the analytical expression for inferred parameters $x^\star$ in terms of observations y [129] with Equation (2.2) to obtain a map

$$M = O_p(O_e^T O_e + R^T R)^{-1} O_e^T \tag{2.3}$$

that can be applied to any observations to obtain their corresponding QoI: $z^\star(y) = My$. Since the map $M \in \mathbb{R}^{n_z \times n_y}$ itself is low-dimensional and does not depend on the observations, it can be computed once and stored, and then applied to any future observations to rapidly obtain their QoI predictions.

We seek a similar ability to obtain rapid predictions given observations in the case where O_e and/or O_p are generalized to nonlinear operators, using stored small multilinear mappings from observations to prediction estimates.

2.1.2 Nonlinear Goal-Oriented Inverse Problem

We now describe the nonlinear goal-oriented inverse problem that is the focus of our algorithm. Again, we consider the case where the observations are sparsely informative, and Tikhonov regularization is used to formulate a well-posed inverse problem:

$$x^\star = \arg\min_x \frac{1}{2}\|y - \mathcal{O}_e(x)\|_2^2 + \frac{1}{2}\|Rx\|_2^2, \tag{2.4}$$

where $x^\star \in \mathbb{R}^{n_x}$ are the inferred parameters, $\mathcal{O}_e : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ is a continuous observation operator that maps parameters to expected observations, and $\frac{1}{2}\|Rx\|_2^2$ is the Tikhonov regularization term with Tikhonov regularization matrix $R \in \mathbb{R}^{n_x \times n_x}$. We assume that the regularization is sufficient for Equation (2.4) to have a unique solution.

Remark 1. *Although the inverse problem in Equation (2.4) is regularized about zero, one can use the same form to describe an inverse problem with Tikhonov regularization about a nonzero point $x_R \in \mathbb{R}^{n_x}$. Such an inverse problem can be written as*

$$x^\star = \arg\min_x \frac{1}{2}\|y - \mathcal{O}_e(x)\|_2^2 + \frac{1}{2}\|R(x - x_R)\|_2^2. \tag{2.5}$$

Solving Equation (2.5) is equivalent to solving

$$(\Delta x)^\star = \arg\min_{\Delta x} \frac{1}{2}\|y - \tilde{\mathcal{O}}_e(\Delta x)\|_2^2 + \frac{1}{2}\|R(\Delta x)\|_2^2, \tag{2.6}$$

where

$$\tilde{\mathcal{O}}_e(\cdot) = \mathcal{O}_e(\cdot + x_R), \tag{2.7}$$

and $x^\star = (\Delta x)^\star + x_R$. *Equation* (2.6) *fits the form of Equation* (2.4).

The QoI corresponding to the inferred parameters is described by

$$z^\star = \mathcal{O}_p(x^\star), \tag{2.8}$$

where the continuous prediction operator $\mathcal{O}_p : \mathbb{R}^{n_x} \to \mathbb{R}^{n_z}$ maps the parameters to a n_z-dimensional prediction output. We consider problems where the observations are only sparsely informative (represented by $n_y \ll n_x$) and the QoI is scalar ($n_z = 1$). If the prediction is low-dimensional but not scalar, we can consider each element of the prediction vector as a separate scalar prediction.

2.2 Algorithm for Scalable Online Nonlinear Goal-Oriented Inference

In Section 2.1.1, we reviewed the linear goal-oriented inverse problem, where one can derive a small, linear, observation-independent mapping that can be applied to observations to rapidly obtain their corresponding QoI. In this section, we introduce our algorithm for similarly obtaining rapid QoI estimates for the nonlinear case described in Section 2.1.2, by applying stored multilinear maps to online observations.

Algorithm 1 describes our algorithm for quickly estimating the QoI $z^\star$ corresponding to parameters inferred from observations y. In the offline phase, we choose a set of expansion observation points and solve the full inverse problem for these observations. At each expansion point, the inferred parameters are used to build a local approximation to the goal-oriented inverse problem. The form of the approximation allows one to obtain the QoI for the approximate problem by applying small, observation-independent, multilinear maps to the observations. Since the maps are independent of the online observations they can be applied to, the maps for each expansion point are calculated and stored offline, forming a library of locally accurate maps, similar in spirit to the model reduction approach of [112]. In the online phase, the appropriate maps can be selected and applied to the online observations y to obtain rapid QoI estimates $\tilde{z}^\star(y)$. This algorithm is scalable, in that the online cost of obtaining a QoI estimate does not increase with the dimensionality of the parameters. This is in contrast to solving the full inverse problem to obtain the exact QoI, which becomes more expensive as the dimensionality of the parameters increases.

Algorithm 1 Algorithm for scalable online goal-oriented inference.

```
Define number of expansion points $N_e$ and prediction approximation order $t$.
procedure OfflinePreparation($N_e$,$t$)
    Obtain set of $N_e$ expansion point observations $y^{(i)}$, $i = 1, \dots, N_e$.
    for $i = 1, \dots, N_e$ do
        Solve for parameters $x^\star_{(i)}$ inferred from observations $y^{(i)}$:
            $$x^\star_{(i)} = \arg\min_x \frac{1}{2}\|y^{(i)} - \mathcal{O}_e(x)\|_2^2 + \frac{1}{2}\|Rx\|_2^2.$$
        Linearize observation operator $\mathcal{O}_e$ about $x^\star_{(i)}$; define $G^{(i)} = \frac{\partial \mathcal{O}_e}{\partial x}\Big|_{x=x^\star_{(i)}}$.
        Approximate prediction operator $\mathcal{O}_p$ with its Taylor expansion about $x^\star_{(i)}$,
          truncated at order $t$.
        Calculate and store offline QoI contribution (see Section 2.3).
        Build and save observation-to-QoI maps for expansion point $y^{(i)}$ (see
          Section 2.4).
    end for
end procedure
Obtain observations $y$.
procedure OnlinePredictionEstimate($y$)
    Choose expansion point(s) (see Section 2.6).
    Calculate QoI estimate $\tilde{z}^\star(y)$ using stored observation-to-QoI map(s) and
        offline QoI contribution(s) (see Section 2.6).
end procedure
```

The set of expansion points may correspond to (a subset of) past observations that we expect to be representative of future online observations, or be a set of (noisy or noise-free) synthetically generated observations. Efficient selection of expansion points from a pool of potential candidates is described in Chapter 3. The remaining sections of this chapter describe various aspects of the algorithm in more detail, for a given set of expansion points. Section 2.3 describes the local approximate goal-oriented inverse problems. Section 2.4 describes different ways to build and store maps corresponding to these local approximations, each appropriate for different situations. Section 2.5 describes an estimate for the QoI error from approximating the goal-oriented inverse problem. Section 2.6 describes how to choose and combine maps, and their resulting QoI estimates, from the library in the online phase.

2.3 Approximate Goal-Oriented Inverse Problem

In this section, we derive an approximate goal-oriented inverse problem that can be used to estimate the QoI for given observations. Sections 2.3.1 and 2.3.2 describe two equivalent expressions for the approximate QoI, and their tradeoffs in practice.

2.3.1 Expansion about Inferred Parameters

Recall the nonlinear goal-oriented inverse problem introduced in Equations (2.4) and (2.8), where both the expected observations $\mathcal{O}_e(x)$ and the prediction output $\mathcal{O}_p(x)$ are nonlinear functions of the parameters. We will approximate the nonlinear goal-oriented inverse problem in a way that gives small, observation-independent, multilinear mappings from observations to the approximate QoI.

Assume the observation operator $\mathcal{O}_e$ is first-order differentiable. Approximate $\mathcal{O}_e$ with its linearization about an expansion point $x_0 \in \mathbb{R}^{n_x}$:

$$\mathcal{O}_e(x) \approx \mathcal{O}_e(x_0) + G(x - x_0), \tag{2.9}$$

where

$$G := \left. \frac{\partial \mathcal{O}_e}{\partial x} \right|_{x=x_0} \in \mathbb{R}^{n_y \times n_x}. \tag{2.10}$$

Let G denote the Jacobian of $\mathcal{O}_e$; note that O_e is used (as in Equation (2.1)) when $\mathcal{O}_e(x) = O_e x$ is linear in x and the observation operator is completely determined by its Jacobian. Equation (2.9) leads to the approximate inverse problem

$$\tilde{x}^\star = \arg\min_x \frac{1}{2}\|y - (\mathcal{O}_e(x_0) + G(x - x_0))\|_2^2 + \frac{1}{2}\|Rx\|_2^2. \tag{2.11}$$

Equation (2.11) is a linear inverse problem with perturbed observations $y - \Delta y$, where

$\Delta y = \mathcal{O}_e(x_0) - Gx_0$, with solution

$$\tilde{x}^\star = (G^T G + R^T R)^{-1}(G^T(y - \Delta y)). \tag{2.12}$$

Remark 2. *For fixed y, when the linearization point $x_0 = x^\star$, the approximate inferred parameters $\tilde{x}^\star$ are exactly equal to the inferred parameters $x^\star$ of the original inverse problem. This need not hold for higher-order approximations of $\mathcal{O}_e$. For higher-order approximations of $\mathcal{O}_e$, the solution of the corresponding approximate inverse problem has no direct analytical expression and may not be unique.*

We next approximate the prediction operator $\mathcal{O}_p$. Assume $\mathcal{O}_p$ is t-order differentiable and can be approximated using a truncated Taylor expansion about x_0:

$$\mathcal{O}_p(\tilde{x}^\star) \approx \tilde{z}^\star \coloneqq \mathcal{O}_p(x_0) + \sum_{d=1}^{t} \frac{1}{d!} T_d(\underbrace{\tilde{x}^\star - x_0, \dots, \tilde{x}^\star - x_0}_{d \text{ times}}), \tag{2.13}$$

where $T_d \in \mathbb{R}^{n_x \times \cdots \times n_x}$ is an order-d symmetric tensor representing the d-th derivatives of $\mathcal{O}_p$ with respect to the parameters. Recall from Equation (1.3) that each tensor T_d acts as a multilinear map.

Let $x_0 = x^\star(y_0)$ be the inferred parameters for observations y_0. For the rest of this thesis, we will use 'expansion point' to refer to either x_0 or y_0, depending on context. Then Remark 2 lets us write

$$x_0 = \frac{\partial \tilde{x}^\star}{\partial y}(y_0 - \Delta y), \tag{2.14}$$

where $\partial \tilde{x}^\star / \partial y$ is known from Equation (2.12). Using Equation (2.14) and tensor-matrix multiplication, we rewrite Equation (2.13) in terms of observations, giving

$$\tilde{z}^\star = \mathcal{O}_p(x_0) + \sum_{d=1}^{t} A_d(\underbrace{y - y_0, \dots, y - y_0}_{d \text{ times}}), \tag{2.15}$$

where

$$A_d = \frac{1}{d!} T_d \left(\frac{\partial \tilde{x}^\star}{\partial y}^T, \dots, \frac{\partial \tilde{x}^\star}{\partial y}^T \right) \in \mathbb{R}^{n_y \times \cdots \times n_y}. \tag{2.16}$$

Equation (2.15) gives the approximate QoI $\tilde{z}^\star$ from applying multilinear A_d to the perturbed observations $y - y_0$.

Equation (2.15) assumes Equation (2.14) holds. In practice, x_0 is rarely exactly equal to the solution of the inverse problem for observations y_0. Having a large gradient at x_0 can induce significant additional error in the approximate QoI from violating Equation (2.14), so it is important to solve the inverse problem for y_0 to near optimality. Section 2.3.2 describes a different grouping of terms to obtain equivalent expressions for the approximate QoI $\tilde{z}^\star$ that avoid assuming Equation (2.14), at the cost of potentially less favorable error behavior when reducing the rank of maps to meet cost constraints (see Sections 2.4.2 and 2.4.3).

2.3.2 Expansion about Arbitrary Parameters

The approximate QoI described by Equation (2.15) can be written in terms of y rather than $y - y_0$, avoiding the assumption that x_0 is the solution to the inverse problem for observation y_0.

Recall Equation (2.13), which expresses the approximate QoI as a function of the solution to the linearized inverse problem. Recall from Equation (1.3) that each tensor T_d acts as a multilinear map. Using this multilinearity and the symmetry of T_d to expand each term of the summation in Equation (2.13) gives

$$\tilde{z}^\star = \mathcal{O}_p(x_p) + \sum_{d=1}^{t} \frac{1}{d!} \sum_{l=1}^{d} (-1)^{d-l} \binom{d}{l} T_d(\underbrace{\tilde{x}^\star, \dots, \tilde{x}^\star}_{l \text{ times}}, \underbrace{x_0, \dots, x_0}_{d-l \text{ times}}). \tag{2.17}$$

Rather than keep terms grouped by their combined order in x_0 and $\tilde{x}^\star$, we can rewrite the summation in Equation (2.17) by grouping together terms that share the same order in $\tilde{x}^\star$, giving

$$\tilde{z}^\star = z_{o,\hat{T}} + \sum_{d=1}^{t} \hat{T}_d(\underbrace{\tilde{x}^\star, \dots, \tilde{x}^\star}_{d \text{ times}}), \tag{2.18}$$

where

$$\hat{T}_d = \sum_{j=d}^{t} \frac{(-1)^{j-d}}{j!} \binom{j}{d} T_j(\underbrace{x_0, \dots, x_0}_{j-d \text{ times}}) \tag{2.19}$$

is an order-d symmetric tensor and

$$z_{o,\hat{T}} = \mathcal{O}_p(x_0) + \sum_{d=1}^{t} \frac{(-1)^d}{d!} T_d(x_0, \dots, x_0) \tag{2.20}$$

denotes terms that do not depend on $\tilde{x}^\star$ and thus do not depend on observations y.

Since $\tilde{x}^\star$ is the solution to a linearized inverse problem, $\hat{T}_d(\tilde{x}^\star, \dots, \tilde{x}^\star)$ can be written in terms of the observations. Let G be the linearized observation operator in Equation (2.10). Using Equation (2.12), we can rewrite Equation (2.18) to express the approximate QoI $\tilde{z}^\star$ written as

$$\tilde{z}^\star(y) = z_{o,\hat{A}} + \sum_{d=1}^{t} \hat{A}_d(y, \dots, y), \tag{2.21}$$

where

$$\hat{A}_d = \sum_{j=d}^{t} (-1)^{j-d} \binom{j}{d} B_j(\underbrace{\Delta y, \dots, \Delta y}_{j-d \text{ times}}), \quad B_j = \hat{T}_j\left(\frac{\partial \tilde{x}^{\star T}}{\partial y}, \dots, \frac{\partial \tilde{x}^{\star T}}{\partial y}\right) \tag{2.22}$$

and

$$z_{o,\hat{A}} = z_{o,\hat{T}} + \sum_{d=1}^{t} (-1)^d B_d(\Delta y, \dots, \Delta y) \tag{2.23}$$

encompasses terms that do not depend on $\tilde{x}^\star$ and thus do not depend on the observations y.

Equation (2.21) gives the QoI from an approximate goal-oriented inverse problem, where the observation and prediction operators are expanded about parameters x_0. This approximation does not require the expansion parameters x_0 to correspond to inferred parameters, so x_0 can be arbitrarily chosen from $\mathbb{R}^{n_x}$. However, if x_0 is chosen without relation to any offline observations y_0, it is then difficult to judge the suitability of the resulting approximate goal-oriented inverse problem for estimating the QoI of online observations y. Thus, for the sake of interpretability, we will assume x_0 is the solution to the inverse problem for observations y_0, rather than arbitrary parameters.

Equation (2.15) and Equation (2.21) are equivalent expressions for the approximate QoI in terms of the (perturbed) observations, assuming that x_0 is the inferred parameters for observations y_0. As noted in Section 2.3.1, this assumption does not exactly hold in practice when the inverse problem is solved numerically. As Equation (2.21) does not rely on this assumption, it may be preferable when x_0 is only coarsely resolved and not very close to optimality. However, numerical experiments (see Section 4.3.2) suggest that Equation (2.21) gives less favorable error behavior when the tensors A_d and $\hat{A}_d$ must be approximately stored due to cost constraints (see Sections 2.4.2 and 2.4.3).

2.4 Observation-to-Approximate-QoI Maps

In Sections 2.3.1 and 2.3.2, we derived two equivalent expressions for an approximate QoI $\tilde{z}^\star(y)$ of online observations y. This approximate QoI is multivariate polynomial in the observations, with coefficients represented by tensors A_d or $\hat{A}_d$. Although they give the same QoI estimate, different representations of A_d and $\hat{A}_d$ give different observation-to-QoI maps with different storage and application costs. These maps are appropriate for different cases, depending on the number of observations n_y, the order t of the prediction Taylor expansion, and the structure of the prediction operator $\mathcal{O}_p$. Three different cases and their appropriate maps are discussed in the following Sections 2.4.1 to 2.4.3, and summarized in Table 2.1. We will refer to the QoI estimate described by Equation (2.15); the analysis is similar for Equation (2.21).

2.4.1 Multilinear Observation-to-QoI Maps

Equation (2.15) shows that the approximate QoI $\tilde{z}^\star$ can be obtained by applying tensors A_d as multilinear maps to perturbed observation $y - y_0$. For small number of observations n_y and small order d, one can directly store A_d at a storage cost of $(n_y)^d$. To obtain $\tilde{z}^\star$ for observations y requires tensor-vector multiplication of these derivative tensors with perturbed observations $y - y_0$ for each non-constant term of the Taylor expansion, at an online application cost of $\mathcal{O}((n_y)^d)$.

Table 2.1: Comparison of observation-to-QoI maps for order-d term of approximate QoI $\tilde{z}^\star(y)$.

	Storage	**Online Application**	**Notes**
Multilinear maps (See Section 2.4.1)	$(n_y)^d$	$\mathcal{O}((n_y)^d)$	
Factored tensor (See Section 2.4.2)			
Canonical	$r(n_y+1)$	$\mathcal{O}(r(n_y+d+1))$	$r \leq$ symmetric canonical rank
Tucker	r^d+rn_y	$\mathcal{O}(r^d+rn_y)$	$r \leq n_y$
Tensor-train	dn_yr+ $(d-2)r^3$	$\mathcal{O}(dn_yr+dr^3)$	$r \leq$ canonical rank, $r \leq (n_y)^{d/2}$
Intermediate QoI (See Section 2.4.3)			additively separable $\mathcal{O}_p$
Truncated intermediate QoI	$r(n_y+1)$	$\mathcal{O}(r(n_y+d+1))$	$r \leq n_x$
Reduced rank	r^d+rn_y	$\mathcal{O}(r^d+rn_y)$	$r \leq n_y$

2.4.2 Factored Tensor Observation-to-QoI Maps

If n_y and/or d are large enough that $\mathcal{O}((n_y)^d)$ storage and/or online application costs are prohibitive, these costs can be reduced by replacing A_d, $d \geq 2$, with a rank-r factorization. Different tensor decompositions correspond to different generalizations of matrix rank and have different storage requirements for given r; unless otherwise specified, we will use 'rank' to refer to any of these generalizations. Factorization alone may sufficiently reduce storage and/or application costs. Otherwise, if we must choose r too small to recover A_d exactly, we then seek to minimize the QoI error incurred by approximating A_d with a rank-r tensor $\tilde{A}_{d,r}$, defined by

$$\tilde{A}_{d,r} = \operatorname*{arg\,min}_{\operatorname{rank}(A)\leq r} \left(\sup_{\|v\|_2=1} |A(v,\ldots,v) - A_d(v,\ldots,v)| \right), \tag{2.24}$$

where r is the chosen maximum rank. This is equivalent to

$$\tilde{A}_{d,r} = \underset{\text{rank}(A) \leq r}{\arg\min} \|A - A_d\|_{\text{spec}} \tag{2.25}$$

by definition of the spectral norm $\|\cdot\|_{\text{spec}}$ and a result by Banach [7,65,104,130]. For $d = 2$, the optimal decomposition is the truncated SVD or eigendecomposition. For $d > 2$, minimizing the spectral norm directly is difficult since calculating the spectral norm of a general tensor is NP-hard [65]; we can instead minimize the error in the Frobenius norm, which is an upper bound of the spectral norm (due to the spectral norm being a special case of the induced norm [142] and by the Cauchy-Schwarz inequality).

We compare three potential options for direct factorization of A_d for $d > 2$:

1. **Canonical**: The canonical (or tensor-rank) decomposition gives the greatest reduction in storage costs; constraining the decomposition to be symmetric, the tensor A_d can be represented exactly with storage costs of $R_S(n_y + 1)$, where R_S is the symmetric tensor rank. However, the cost of exact representation may be difficult to gauge, as R_S may be large (exceed n_y) but is NP-hard to calculate [65]. A_d may also be degenerate, a non-rare event [36, 44] where Equation (2.25) is ill-posed; this can cause problems for iterative algorithms, such as those reviewed in [60, 77]. Using the canonical decomposition gives online application costs of $\mathcal{O}(r(n_y + 1 + d))$ for the d-th non-constant term of the expansion of $\tilde{z}^\star$.

2. **Tucker**: The Tucker decomposition [133] (or multilinear SVD [43]) can be used to represent A_d with an order-d symmetric core tensor of length r along each dimension and a r-by-n_y factor matrix, for storage cost of $r^d + rn_y$; here r is the multilinear rank, which need not exceed n_y to exactly represent A_d. One can further reduce storage costs by seeking sparser core tensors [4, 63, 133]. Equation (2.25) is well-defined for the multilinear rank; there are algorithms for computing an exact [43] decomposition, an optimal reduced multilinear rank decomposition [60,77], and a suboptimal but error-bounded decomposition [43,

134]. Using the Tucker decomposition gives online application costs of $\mathcal{O}(rn_y + r^d)$ for the d-th non-constant term of the expansion of $\tilde{z}^\star$.

3. **Tensor-Train**: The tensor-train decomposition [102] is most appropriate in cases where d is large but calculating the canonical decomposition encounters numerical difficulties. If the tensor-train ranks are chosen to be r, then the storage cost is $dn_yr + (d-2)r^3$ after applying a Tucker decomposition to further compress the core tensors. An algorithm presented in [102] can be used to obtain a quasi-optimal tensor-train decomposition. Using the tensor-train decomposition gives application costs of $\mathcal{O}(dn_yr + dr^3)$ for the d-th non-constant term of the expansion of $\tilde{z}^\star$.

2.4.3 Observation-to-Intermediate-QoI Maps

When n_y or d are large, obtaining maps for the approximate QoI $\tilde{z}^\star$ by factoring tensors as in Section 2.4.2 can be costly in the offline phase. Iterative algorithms for the canonical decomposition are not guaranteed to give good approximations [77, 102], and computing the Tucker and tensor-train decompositions using [43, 102] requires matrix factorization of unfolded tensors whose number of elements grows exponentially in the tensor order.

In the case where $\mathcal{O}_p$ is additively separable ($\mathcal{O}_p(x) = \sum_{i=1}^{n_x} f_i(x_i)$ for some functions f_i), one can obtain maps for $\tilde{z}^\star$ while avoiding this costly tensor decomposition. Such structure can occur when, for example, the parameters correspond to a discretized field and the QoI is an integral of a function of the field, or when $\mathcal{O}_p$ corresponds to a truncated high-dimensional model representation (HDMR) [81].

Recall the expression for the approximate QoI given in Equation (2.13). The first term $\mathcal{O}_p(x_0)$ is independent of the observations y. For $d = 1$, the term $T_1(\tilde{x}^\star - x_0)$ can be obtained by directly applying Equation (2.3) with $O_p = T_1$ to obtain an observation-to-QoI map M_1 such that $M_1(y - y_0) = A_1(\tilde{x}^\star - x_0)$.

For $d > 1$, we use the symmetric tensor rank decomposition (over $\mathbb{R}$) for an order-d

symmetric tensor [67,68], writing

$$T_d = \sum_{i=1}^{R_S} s_i \underbrace{a_i \otimes \cdots \otimes a_i}_{d \text{ times}} = \sum_{i=1}^{R_S} s_i a_i^{\otimes d}, \tag{2.26}$$

where $a_i \in \mathbb{R}^{n_x}$, $s_i \in \mathbb{R}$, and the symmetric tensor rank (over $\mathbb{R}$) of T_d is the smallest R_S for which Equation (2.26) holds with equality. $T_d(\tilde{x}^\star - x_0, \ldots, \tilde{x}^\star - x_0)$ can be written as a simple function of intermediate linear outputs of $\tilde{x}^\star - x_0$; we may use Equations (2.3) and (2.14) to calculate observation-to-intermediate-QoI maps for these intermediate linear outputs. Construct a factor matrix $P \in \mathbb{R}^{n_x \times R_S}$ so that its i-th column is a_i. We generate the observation-to-QoI map M_d for the intermediate linear output $z_d = P^T(\tilde{x}^\star - x_0)$ so that once observations y are obtained online, we apply the observation-to-QoI map to obtain $z_d = M_d(y - y_0)$; the d-th non-constant term in the approximate prediction output can then be obtained by

$$T_d(\underbrace{\tilde{x}^\star - x_0, \ldots, \tilde{x}^\star - x_0}_{d \text{ times}}) = \sum_{i=1}^{R_S} s_i (\underbrace{z_d \circ \cdots \circ z_d}_{d \text{ times}})_i, \tag{2.27}$$

where $\circ$ denotes the Hadamard (element-wise) product.

When $\mathcal{O}_p$ is additively separable, its Hessian and higher-order derivative tensors T_d, $d \geq 2$, are diagonal. Their symmetric tensor rank decomposition in Equation (2.26) is immediately available, with $R_S = n_x$; note that this diagonal structure is lost if T_d is transformed into A_d as in Equation (2.15). Combining Equation (2.13) and Equation (2.27), we can generate observation-to-QoI maps M_d for intermediate QoI z_d and obtain approximate QoI

$$\tilde{z}^\star(y) = \mathcal{O}_p(x_0) + \sum_{d=1}^{t} \sum_{j=1}^{n_x} s_j \left((M_d(y - y_0)) \circ \cdots \circ (M_d(y - y_0)) \right)_j \tag{2.28}$$

at a storage cost of $n_x(n_y + 1)$ and online application cost of $\mathcal{O}(n_x(n_y + 1 + d))$ for the d-th non-constant term of the expansion of $\tilde{z}^\star$.

To avoid storage and application costs scaling with n_x, one can control the size of the maps M_d by truncating the intermediate QoI or by using a low-rank approximation of M_d. The former is cheapest to calculate and, for a given r, gives smaller storage and application costs, but potentially more error.

To truncate the intermediate QoI, scale the decomposition in Equation (2.26) so that $\|a_i\|_2 = 1$ and truncate it, keeping only the r terms with the largest $|s_i|$. This reduces the dimension of the intermediate QoI z_d from n_x to r, thus reducing the size of M_d from n_x-by-n_y to r-by-n_y, and the storage costs to $r(n_y + 1)$ and application costs to $\mathcal{O}(r(n_y + 1 + d))$, for the d-th non-constant term of the expansion of $\tilde{z}^\star$. The elements of the intermediate QoI most sensitive to the parameters may be insensitive to the observations, so keeping them may do little to minimize the error induced by reducing the size of the map.

A low-rank approximation of M_d can induce less error for a given r, at the expense of greater storage and application costs. Let the decomposition in Equation (2.26) be scaled so that $|s_i| = 1$. The simplest rank reduction corresponds to approximating M_d with its rank-r SVD reconstruction; this minimizes worst-case error in the intermediate QoI. With enough information about the distribution of online observations, one can instead minimize an average error. We seek a low-rank approximation $\tilde{M}_{d,r}$ satisfying

$$\tilde{M}_{d,r} \in \underset{\operatorname{rank}(M)\leq r}{\arg\min}\ \mathbb{E}\left[\|M_d y - M y\|_2^2\right], \tag{2.29}$$

where the expectation is over the distribution of online observations. Equivalently,

$$\tilde{M}_{d,r} \in \underset{\operatorname{rank}(M)\leq r}{\arg\min}\ \|(M_d - M)\Gamma_Y^{1/2}\|_F, \tag{2.30}$$

where $\Gamma_Y^{1/2}(\Gamma_Y^{1/2})^T = \Gamma_Y$ and $\Gamma_Y \succ 0$ is the covariance of the online observations. Using the results of [55], we can write

$$\tilde{M}_{d,r} = (M_d \Gamma_Y^{1/2})_r \Gamma_Y^{-1/2}, \tag{2.31}$$

where $(\cdot)_r$ indicates the rank-r SVD reconstruction.

Approximating M_d with either its rank-r SVD or the weighted SVD in Equation (2.31) gives a rank-r map of size n_x-by-n_y. In both cases, we can avoid storage and online application costs scaling with n_x. We form and store a tensor $S(U_r, \ldots, U_r)$, where S is a diagonal tensor whose diagonal elements are s_i, and U_r are the left singular vectors of M_d or $M_d\Gamma_Y^{1/2}$, respectively; the product of the corresponding singular values and right singular vectors are also stored. This gives storage and application costs of $\mathcal{O}(r^d + rn_y)$ for the d-th non-constant term of the expansion of $\tilde{z}^\star$.

2.5 Error Estimate for Approximate QoI

In this section, we describe a first-order estimate for the error $|z^\star(y) - \tilde{z}^\star(y)|$ of the QoI estimate $\tilde{z}^\star(y)$ for online observations y. Section 2.5.1 describes the construction of this estimate by viewing $\tilde{z}^\star$ as a function of its expansion point, given fixed online observations y. Section 2.5.2 gives a finite difference method to estimate the necessary gradient. Section 2.5.3 introduces further approximations to the error estimate expression that facilitate more rapid online error estimation.

2.5.1 Approximate QoI as a Function of Expansion Point

Recall from Sections 2.3.1 and 2.3.2 that the QoI estimate $\tilde{z}^\star$ corresponds to the Taylor series approximate output of a linearized inverse problem. Recall from Remark 2 that the inferred parameters of the linearized inverse problem are exact when linearization occurs about the solution of the original inverse problem. A Taylor series expansion is also exact when evaluated at its expansion point. Thus, the approximate QoI is exact ($\tilde{z}^\star = z^\star$) when its expansion point is the exact inferred parameters ($x_0 = x^\star$, equivalent to $y_0 = y$); the exactness of the approximate QoI in this case is most obvious in Equation (2.15).

It follows that $\tilde{z}^\star$ may not be exact when constructed by expanding about $y_0 \neq y$. For the rest of Section 2.5, we will use y to refer to the online observations, and $v \in \mathbb{R}^{n_y}$ to refer to the expansion point observations when viewed as a variable, with v_0 referring to a particular fixed expansion point. In the offline stage, we solve the inverse problem (Equation (2.4)) for expansion point observations v_0. Expanding $\mathcal{O}_e$ and $\mathcal{O}_p$ about $x^\star(v_0)$ (as in Equations (2.9) and (2.13)), we obtain an expression (see

Equations (2.15) and (2.21)) for the approximate QoI $\tilde{z}^\star(y)$ of online observations y. We define a parameterized function f so that $f(v_0; y)$ denotes this approximate QoI from expanding about $x^\star(v_0)$ and applied to online observations y. Choosing different expansion point observations v gives different QoI estimates $f(v; y)$ for fixed online observations y; written this way, we emphasize the QoI estimate as a function of the expansion point, parameterized by the online observations. For given y, the error in the QoI estimate is a result of evaluating f at the wrong v; the error is a result of expanding about $v \neq y$. When $v = y$, we have $f(y; y) = z^\star(y)$.

We wish to estimate the error in the QoI estimate from expanding about expansion point v_0 and applying its maps to online observations y; we seek $|z^\star(y) - \tilde{z}^\star(y)| = |f(y; y) - f(v_0; y)|$. Viewing the QoI estimate as a function of the expansion point, we use the linear Taylor expansion of $f(v; y)$ about v_0 to obtain the first-order error estimate

$$|z^\star(y) - \tilde{z}^\star(y)| = |f(y; y) - f(v_0; y)| \approx \left| \left.\frac{\partial f(v; y)}{\partial v}\right|_{v_0}^T (y - v_0) \right|. \tag{2.32}$$

There is generally no analytical expression for the gradient $\partial f(v; y)/\partial v$. In Sections 2.5.2 and 2.5.3, we describe two finite difference approaches to numerically estimate the gradient, with different tradeoffs in terms of accuracy and online expense.

2.5.2 Finite Difference Gradient Approximation

Fix online observations y for which we are calculating the QoI error estimate. To obtain a numerical estimate for the gradient $\partial f(v; y)/\partial v$, we use the finite difference method described in [11], which requires evaluation of $f(v^{(i)}; y)$ at n_g other finite difference expansion points $v^{(i)}$, $i = 1, \dots, n_g$, where $n_g \geq n_y$ for a first-order gradient estimate and $n_g \geq n_y(n_y + 1)$ for a second-order gradient estimate. The gradient estimate g is the least-squares solution to the (potentially overdetermined) system

$$Dg = (\delta f), \tag{2.33}$$

where difference matrix $D \in \mathbb{R}^{n_g \times n_y}$ has elements $D_{ij} = v_j^{(i)} - (v_0)_j$ and the right-hand side $(\delta f) \in \mathbb{R}^{n_g}$ has elements $(\delta f)_i = f(v^{(i)}; y) - f(v_0; y)$.

Each $f(v^{(i)}; y)$ corresponds to the QoI estimate for online observations y, where maps for the QoI estimate are obtained from expanding $\mathcal{O}_e$ and $\mathcal{O}_p$ about the inferred parameters of expansion point $v^{(i)}$ (see Section 2.5.1). If the number of expansion points $N_e \geq n_g + 1$, then we can choose $v^{(i)}$ from among the expansion points already in the library, avoiding the need to solve new inverse problems and generate new maps in the online phase. We can use existing maps in the library to compute $f(v^{(i)}; y)$ and estimate the gradient at only the cost of applying the maps and solving Equation (2.33).

The quality of the gradient estimate depends on the spread of the expansion points. If the library is small, then neighboring expansion points may be far apart and give a poor finite difference estimate of the gradient. Expansion points that are closely clustered together may be useful for gradient estimation, but not efficient (in terms of QoI estimation accuracy for a given number of expansion points).

Estimating the gradient using Equation (2.33) requires evaluation of $n_g + 1$ QoI estimates by applying maps from $n_g + 1$ expansion points in the library, which may become expensive for large n_y. In Section 2.5.3, we describe how one can make further approximations to the error expression in Equation (2.32) to obtain a more rapid online QoI estimate, at the cost of its accuracy being further locally restricted about v_0.

2.5.3 Approximations for Rapid Online QoI Error Estimation

Computing the error estimate in Equation (2.32) by solving Equation (2.33) can be expensive in the online phase because the gradient depends on the online observations y. For online observations close to the expansion point (small $\|y - v_0\|$), we can approximate y with v_0 in the gradient in Equation (2.32). This gives the following estimate for the QoI error from expanding about v_0 for online observations y:

$$|z^\star(y) - \tilde{z}^\star(y)| = |f(y;y) - f(v_0;y)| \approx \left| \left.\frac{\partial f(v;v_0)}{\partial v}\right|_{v_0}^T (y - v_0) \right|. \tag{2.34}$$

This new gradient $\partial f(v;v_0)/\partial v$ can also be estimated by using other expansion points in the library to obtain a finite difference estimate, as described in Section 2.5.2.

Since $\partial f(v;v_0)/\partial v$ is independent of the online observations, it can be estimated in the offline phase. Rather than choose finite difference points from among existing expansion points, we can choose finite difference points that are better placed for an accurate finite difference gradient estimate. If the majority of the cost of obtaining the maps for each expansion point comes from solving the inverse problem, rather than from expanding $\mathcal{O}_e$ and $\mathcal{O}_p$ and computing the maps, then these finite difference expansion points can be obtained more cheaply by avoiding solution of the full nonlinear inverse problem. The maps of these finite difference points are not stored for use in the online phase, since they are usually close to v_0, but of lower quality from bypassing the full inverse problem.

Let $v^{(i)}$, $i = 1, \ldots, n_g$ be our set of finite difference points, chosen randomly from a ball of some user-specified radius and centered at v_0. Instead of solving Equation (2.33), we solve for gradient estimate $\tilde{g}$ as the least-squares solution to

$$\tilde{D}\tilde{g} = (\tilde{\delta f}), \tag{2.35}$$

where the approximate difference matrix $\tilde{D} \in \mathbb{R}^{n_g \times n_y}$ has elements $\tilde{D}_{ij} = \tilde{v}_j^{(i)} - (v_0)_j$ and the right-hand side $(\tilde{\delta f}) \in \mathbb{R}^{n_g}$ has elements $(\tilde{\delta f})_i = \tilde{f}(v^{(i)};y) - f(v_0;y)$. The elements of $\tilde{D}$ and $\tilde{\delta f}$ correspond to approximately solving the inverse problem at the finite difference points, in the manner described below.

For each $v^{(i)}$, we solve the linearized inverse problem (see Equation (2.11)), with $\mathcal{O}_e$ linearized about $x^\star(v_0)$, for observations $v^{(i)}$:

$$(\tilde{x}^\star_{\tilde{g}})^{(i)} = \arg\min_x \frac{1}{2}\|v^{(i)} - (\mathcal{O}_e(x^\star(v_0)) + G(x - x^\star(v_0)))\|_2^2 + \frac{1}{2}\|Rx\|_2^2, \tag{2.36}$$

where G is the Jacobian of $\mathcal{O}_e$ at $x^\star(v_0)$. We define $\tilde{f}(v^{(i)}; y)$ to be the QoI estimate for online observation y, constructed by expanding $\mathcal{O}_e$ and $\mathcal{O}_p$ about $(\tilde{x}^\star_{\tilde{g}})^{(i)}$, as in Section 2.3, to obtain an approximate goal-oriented inverse problem.

To partially compensate for the fact that $(\tilde{x}^\star_{\tilde{g}})^{(i)}$ is only an approximation to $x^\star(v^{(i)})$, we use a perturbed step matrix $\tilde{D}$ based on $\tilde{v}^{(i)}$, which we define as the observations for which $(\tilde{x}^\star_{\tilde{g}})^{(i)}$ is closest to being the exact inferred parameters. Let $J(x, v)$ denote the objective function of the full inverse problem (see Equation (2.4)) evaluated at parameters x and observations v:

$$J(x, v) = \frac{1}{2}\|v - \mathcal{O}_e(x)\|_2^2 + \frac{1}{2}\|Rx\|_2^2. \tag{2.37}$$

Then we define $\tilde{v}^{(i)}$ as

$$\tilde{v}^{(i)} = \arg\min_{\tilde{v}} \left\| \left. \frac{\partial J(x, \tilde{v})}{\partial x} \right|_{(\tilde{x}^\star_{\tilde{g}})^{(i)}} \right\|_2, \tag{2.38}$$

which corresponds to the solution of an overdetermined linear system with n_x equations in n_y variables:

$$\left(\left. \frac{\partial \mathcal{O}_e}{\partial x} \right|_{(\tilde{x}^\star_{\tilde{g}})^{(i)}} \right)^T \tilde{v}^{(i)} = R^T R (\tilde{x}^\star_{\tilde{g}})^{(i)} + \left(\left. \frac{\partial \mathcal{O}_e}{\partial x} \right|_{(\tilde{x}^\star_{\tilde{g}})^{(i)}} \right)^T \mathcal{O}_e. \tag{2.39}$$

The error estimate in Equation (2.34) can be rapidly computed in the online phase, requiring only the dot product of a stored gradient estimate with the difference $y - v_0$, where v_0 corresponds to an expansion point. The stored gradient estimate can be computed in the offline phase without additional expensive solutions of the full nonlinear inverse problem, either by using existing maps in the library, or by expanding about the solutions to linearized inverse problems. The latter option is more costly, but potentially more accurate due to the ability to choose finite difference points. Note that the latter option induces additional error in the gradient estimate when the approximate QoI $\tilde{z}^\star$ corresponds to reduced-rank representations of A_d, due to the violation of Equation (2.14), as discussed in Section 2.3.1.

2.6 Choosing and Combining Local Approximations

In the offline phase of Algorithm 1, we approximate goal-oriented inverse problems (see Section 2.3) and generate observation-to-QoI maps to map observations to their approximate QoI (Section 2.4). These approximations are localized about the inferred parameters $x^\star_{(i)}$ of expansion point observations $y^{(i)}$, $i = 1, \dots, N_e$. For given online observations y, each of the N_e expansion points in the library corresponds to its own QoI estimate, with the i-th expansion point having QoI estimate denoted by $\tilde{z}^\star_{(i)}(y)$. Given this library of approximate problems, we wish to minimize the error in the QoI estimate, either by choosing the QoI estimate of a single best expansion point or by synthesizing the estimates corresponding to multiple expansion points.

For given online observations y, we write the final QoI estimate $\tilde{z}^\star(y)$ as a weighted sum of the QoI estimates $\tilde{z}^\star_{(i)}(y)$ corresponding to individual expansion points:

$$\tilde{z}^\star(y) = \sum_{i=1}^{k} \tilde{z}^\star_{(i)}(y) \frac{\phi(\delta_i(y))}{\sum_{j=1}^{k} \phi(\delta_j(y))}, \tag{2.40}$$

where the weighting (or kernel) ϕ is some function of the distance $\delta_i(y)$ between the current evaluation point and the i-th expansion point. The k nearest neighbors over which we average are chosen based on $\delta_i(y)$. We note that Equation (2.40) has similarities to k-Nearest Neighbors (kNN), also referred to as 'kernel regression' or 'distance weighted averaging', in that both estimate the unknown output at a query point using a weighted sum of values at neighboring points [98, 122, 144]. The black-box kNN estimate $\hat{z}^\star(y)$ can be written

$$\hat{z}^\star(y) = \sum_{i=1}^{k} z^\star(y^{(i)}) \frac{\phi(\delta_i(y))}{\sum_{j=1}^{k} \phi(\delta_j(y))}, \tag{2.41}$$

where k is the number of nearest neighbors over which the average is taken, and the weighting ϕ is some function of the distance $\delta_i(y) = \|y - y^{(i)}\|_2$ between the current evaluation point and the i-th expansion point. In contrast, we will refer to our approach in Equation (2.40) as *physics-informed kNN*, since the values being combined are based on outputs of maps built using knowledge about the physics and structure of the goal-oriented inverse problem. We note that Equation (2.41) is equivalent to Equation (2.40) when $\mathcal{O}_p$ is approximated with a zero-order Taylor expansion. Given this equivalence, we will interchangeably refer to our expansion points as 'training points' for the rest of this thesis.

This similarity between physics-informed and black-box kNN allows us to borrow weighting functions used in black-box kNN. One can weight all neighbors equally ($\phi(\delta) = 1$), although one usually weights nearby points more favorably. A common choice of ϕ is the inverse distance weighting $\phi(\delta) = \delta^{-1}$; another option is a Gaussian weighting ($\phi(\delta) = \exp(-\delta^2)$). The choice of weighting kernel ϕ reflects the relative importance of nearby and distant points.

In regression, the distance $\delta_i(y)$ reflects one's beliefs about the similarity of the function value at different points, with nearby points more likely to have similar values. Recall from Section 2.5.1 that for a given online y, we can view the QoI estimate as a function of the expansion point. In our case, the distance $\delta_i(y)$ reflects our beliefs about the similarity of the QoI estimates from using either $y^{(i)}$ or y as the expansion point. The simplest distance function is $\delta_i(y) = \|y - y^{(i)}\|_2$.

One can potentially obtain a more accurate QoI estimate $\tilde{z}^\star(y)$ by using an error estimate to help choose neighboring points to average over. This error estimate should be rapidly obtainable in the online phase. To estimate the error from using the maps of expansion point $y^{(i)}$ to calculate the QoI for online observations y, we use Equation (2.34) to obtain the error estimate $\tilde{\epsilon}_i(y)$, defined as

$$\tilde{\epsilon}_i(y) = \left| \left.\frac{\partial f(v; y^{(i)})}{\partial v}\right|_{y^{(i)}}^{T} (y - y^{(i)}) \right|. \tag{2.42}$$

Since the error estimate is only locally accurate, we use it to choose and weight nearest neighbors by defining $\delta_i(y) = \|y - y^{(i)}\|_2 + \alpha\tilde{\epsilon}_i(y)$. The tuning factor α is a conservative way to balance the error estimate with its accuracy, avoiding the selection of faraway points whose error is greatly underestimated. Although Equation (2.42) gives a less accurate error estimate than Equation (2.32), it is more rapidly obtainable in the online phase and can still be useful for choosing and combining library estimates. Such an example is given in Section 4.4.

2.7 Chapter Summary

In this chapter, we described and derived our algorithm for obtaining rapid QoI estimates given observations. The algorithm computes and stores compact multilinear maps in an offline preparatory phase, and applies these maps to online observations to obtain rapid estimates of their QoI. The maps are based on local approximations to the full nonlinear goal-oriented inverse problem. Since the online observations are unknown in the offline phase, and the maps are only locally accurate, we generate a library of maps, each localized about a training (expansion) point.

A large library is more likely to have a training point close to the online observations, and thus likely to give a more accurate QoI estimate. However, generating and/or utilizing large training sets may not be feasible, since each training point requires an expensive solution of the full inverse problem, and each training point corresponds to multilinear maps that must be stored. In the next chapter, we examine approaches for efficiently choosing members of the training set, in order to increase online estimation accuracy given a limited offline computational budget.

Chapter 3

Active Learning to Efficiently Generate Physics-Informed Maps

Algorithm 1 enables rapid QoI estimates for online observations by generating a library of observation-to-QoI maps in the offline stage. Each map in the library corresponds to a local approximation to the goal-oriented inverse problem, constructed by expansion about the solution to a full inverse problem. In this chapter, we propose methods for efficiently choosing these expansion points, so as to reduce errors from estimating the QoI with physics-informed maps given a limited offline computational budget and/or map storage budget. In Section 3.1, we define the scenarios in which we develop our active learning approaches. Section 3.2 describes an active learning approach that seeks to efficiently minimize the largest QoI error. We then consider the specific case where our QoI is used for classification. Section 3.3 describes an uncertainty-based greedy sampling approach for this classification case. Section 3.4 describes a more expensive variation that uses a more complex notion of utility to potentially choose more efficient training sets. We summarize the chapter in Section 3.5.

3.1 Active Learning Scenarios

In this section, we describe the scenarios in which we develop our active learning approaches. Section 3.1.1 describes assumptions about the observations available in the offline phase. Section 3.1.2 describes how we extend our continuous QoI to a classification goal.

3.1.1 Pool-based Sampling

We focus on the pool-based sampling scenario, where a large, static pool of offline observations is available at once. We let $\mathcal{U}$ denote the total pool of offline observations. Our goal in active learning is to efficiently choose N_e points from $\mathcal{U}$ to form the training set $\mathcal{T} \subset \mathcal{U}$. In cases where the learned input-output map must be (expensively) updated each time the training set changes, there may be a benefit to adding a batch of points to the training set at a time. In our case, there are no savings to be gained in this batch setting, so we will sequentially choose single points to add to the training set by solving their inverse problem and create maps from their local expansions. We assume that the distribution of points in this offline pool is representative of the expected distribution of observations in the offline phase.

Given the parallels with black-box kNN and the borrowing of ideas from the machine learning literature, we will use similar terminology to refer to parallel concepts. Specifically, we will use 'labeling' or 'querying' an observation point to refer to the act of solving its full inverse problem and computing the corresponding physics-informed maps, as described in the offline portion of Algorithm 1.

3.1.2 Classification from Regression

Thus far, we have considered the goal-oriented inverse problem where the QoI is a continuous, differentiable function of the parameters (see Section 2.1.2). We can extend this continuous regression setting, and our Algorithm 1, to some cases where the ultimate goal is classification. We describe two scenarios where a continuous QoI can be simply processed to obtain a classification. We will continue to use 'Quantity of Interest' to refer to the continuous output of the inferred parameters, even when the classification is the ultimate goal.

A common way to extend continuous regression to classification is to define a 'plug-in' classifier [29], where the vector QoI $\vec{z}(x)$ is a vector of class probabilities; each element $z_i(x)$ is equal or proportional to the probability of parameters x belonging to the i-th class. The estimated class is then the one with greatest probability. The case of a plug-in classifier can arise when the prediction operator $\mathcal{O}_p$ corresponds to a statistical model. For example, $\mathcal{O}_p$ might correspond to a multinomial logistic regression model, where the weightings have previously been tuned based on a training set of parameter-class pairs. The model gives $z_i(x) = \exp(\sum_j f_j^{(i)}(x))$ proportional to the probability of x belonging to the i-th class, where $f_j^{(i)}$ is the j-th feature of the i-th class. If $f_j^{(i)}$ are continuous and differentiable, then we can generate physics-informed maps using Algorithm 1 to rapidly classify observations without having to first infer for their corresponding parameters.

We focus our active learning approaches on a simpler extension to classification, where a scalar continuous QoI is mapped to a class through a (combination of) thresholding function(s). For example, suppose the QoI corresponds to the predicted performance of an object with the inferred material parameters. Then the object may be classified as acceptable or unacceptable based on whether the QoI exceeds some threshold. A more complex classification might use multiple thresholds to define multiple grading classes for the object ('poor', 'good', 'excellent'). Another example is where one has a support vector machine (SVM) trained on parameter-class pairs to give binary classifications for parameters. In this case, one can define a QoI equal to the dot product of the parameters with the separating hyperplane normal (in the transformed feature space); for given observations, whether the QoI exceeds some threshold determines the class of the observations. In both of these examples, we can obtain rapid QoI estimates $\tilde{z}^\star$ for online observations with Algorithm 1, then apply simple thresholding functions to the QoI to classify the online observations.

Compared to the general regression case, the goal of an active learning approach in the classification case is to efficiently reduce misclassification rates, rather than efficiently reduce QoI errors. For the QoI thresholding case, it is more important for the QoI estimate and exact QoI to be on the same side of the threshold(s) than for the QoI estimate to closely match the exact QoI.

3.2 Minimizing Maximum Error

In this section, we present an active learning approach that seeks to minimize the largest online QoI error. We will describe a min-max error formulation and the obstacles to solving it directly for our training set, and then give an approximate formulation that can be optimally solved with an existing method for which we can compute the necessary quantities.

Assume the offline pool of observations is representative of the spread of observations that will be obtained in the online phase. In the online phase, we will choose one expansion which gives the least QoI error ($k = 1$). For candidate training set T and observations u, this choice gives the minimal QoI error $\min_{\tau \in T} |f(u; u) - f(\tau; u)|$, where $f(\cdot; \cdot)$ is as defined in Section 2.5.1. The maximum online QoI error, estimated using the observation pool $\mathcal{U}$, is described by $\max_{u \in \mathcal{U} \setminus T} \min_{\tau \in T} |f(u; u) - f(\tau; u)|$. To minimize the maximum online QoI error, we choose N_e points from $\mathcal{U}$ to form the training set $\mathcal{T}$ such that

$$\mathcal{T} = \underset{T \subset \mathcal{U},\, |T| \leq N_e}{\arg\min} \left(\max_{u \in \mathcal{U} \setminus T} \min_{\tau \in T} |f(u; u) - f(\tau; u)| \right). \tag{3.1}$$

Define $d(u, v) = |f(u; u) - f(v; u)|$. If we assume $d(\cdot, \cdot)$ satisfies the directed triangle inequality $d(u, v) + d(v, w) \geq d(u, w)$, then Equation (3.1) is an asymmetric N_e-center problem. Greedy algorithms for solving the asymmetric N_e-center problem are described in [5, 103]. These algorithms are asymptotically optimal [32]. Although we have a QoI error estimate (see Equation (2.32)), we cannot feasibly apply any of these algorithms to our active learning problem. We can only evaluate $d(u, v)$ for v already in the training set, whereas the aforementioned algorithms require the ability to evaluate $d(u, v)$ for any pairs $u, v \in \mathcal{U}$.

Instead, we use the distance between pairs of observation points as a proxy for their QoI error, from expanding about one to estimate the QoI at the other. Assuming the error from expanding about training point τ to estimate the QoI for observations u increases with the distance between τ and u, we can replace the QoI error $|f(u; u) - f(\tau; u)|$ in Equation (3.1) with the distance $\|u - \tau\|_2$. Then the active learning problem becomes one of choosing N_e points from $\mathcal{U}$ to form the training set $\mathcal{T}$ such that

$$\mathcal{T} = \underset{T \subset \mathcal{U},\, |T| \leq N_e}{\arg\min} \left(\max_{u \in \mathcal{U} \setminus T} \min_{\tau \in T} \|u - \tau\|_2 \right), \tag{3.2}$$

where $\min_{\tau \in T} \|u - \tau\|_2$ describes the distance between observations u and the candidate training set T. Equation (3.2) corresponds to the symmetric N_e-center problem, which can be solved approximately in a greedy fashion using the algorithm described in [59]. This algorithm reduces the objective function to within a factor of two of the optimal value, which is the best that can be achieved by a polynomial time approximation algorithm [50].

Solving Equation (3.2) using the method in [59] to obtain a training set has similarities to the active learning approach in [139]. In [139], a variation of this greedy algorithm is used to select training points for a nearest-neighbors classifier. The projected distance of a query point to its nearest group of training points, representing a shared class, is used as a proxy for the uncertainty in the query point belonging to that class. The query point with the greatest uncertainty is next added to the training set.

In our case, we use the distance between a query point and its nearest training point as a proxy for the QoI error in that query point given the current training set. At each iteration, we greedily add the point with the greatest error proxy (greatest distance to the current training set) to the training set. Solving the distance-based formulation in Equation (3.2) with the greedy algorithm in [59] is feasible in that it only requires the evaluation of the distance between observations, rather than error estimates between arbitrary pairs of points. We note that there exist other active learning methods and clustering approaches which use only characteristics of the pool of input points, such as [73, 105, 150], which may also be useful for active learning in an approximate error minimization framework.

3.3 Physics-Informed Uncertainty Sampling

In this section, we describe an uncertainty sampling approach to active learning for threshold-based classification, extending ideas from uncertainty sampling for kNN classification to our physics-informed maps. In an uncertainty sampling approach, the training set is usually seeded with a random initial selection of points. Each unlabeled point is assigned a score related to the uncertainty in its classification, given the current input-output mapping trained on the current training set. A point with the greatest uncertainty score is then chosen as the next query point to add to the training set. The trained mapping is updated, the uncertainty scores of the remaining unlabeled points are re-evaluated where necessary, and then the process repeats.

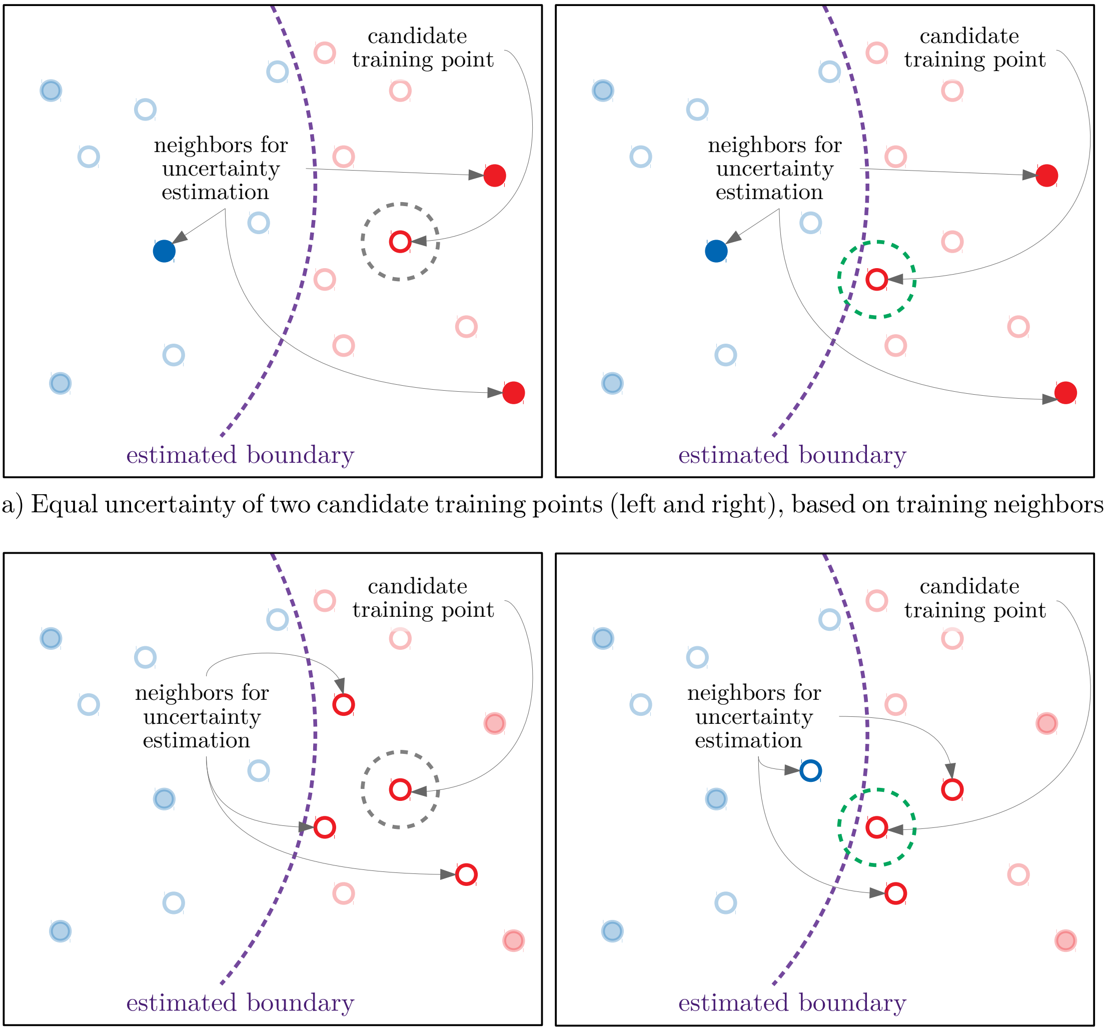
Uncertainty sampling approaches for different classes of trained mappings use different ways of estimating the probability of each class at a point, depending on the structure of the trained mapping. For all classes of trained mappings, when there are more than two classes, one also has different choices of uncertainty measure with which to map a distribution over classes to a scalar uncertainty score. The choice of uncertainty measure, such as information entropy or margin (gap between probabilities of two most likely classes), affects the behavior of the uncertainty sampling; some choices of uncertainty measure are compared in [118]. We will focus on tailoring estimation of the probability distribution of classes at a point to our physics-informed kNN. Once the probability of different classes at a point is estimated, any of the uncertainty measures can be applied to map the distribution to an uncertainty score.

For classification kNN, the class of a point is estimated as the most common class among its k nearest neighbors in the training set. Similarly, when performing uncertainty sampling, the probability distribution over the classes at a point can be estimated with the empirical distribution of the classes of its k_a nearest neighbors in the training set. We will refer to uncertainty sampling with this estimate of classification probabilities as uncertainty sampling with training neighbors (USTN). Described this way, greater uncertainty tends to correspond to points near the boundaries between classes, as estimated by the classification kNN mapping trained on the current training set. Placing more training points near the estimated boundary helps reduce misclassification rates, since the accuracy of a classifier depends on accurately capturing the boundaries between classes. Points near the estimated boundary have more uncertainty, as their class is more prone to change with small changes in the boundary.

For our physics-informed kNN, we propose a different method of computing an uncertainty score to use in uncertainty sampling. Rather than computing the uncertainty of a point based on the classes of its nearest training points, we compute uncertainty based on the estimated classes of its nearest neighbors in the pool $\mathcal{U}$. If these neighbors are also in the training set $\mathcal{T} \subset \mathcal{U}$, then their estimated class is exact (see Remark 2). By computing uncertainty based on the estimated classes of neighbors, we are better able to find points close to the boundary between classes, as estimated by physics-informed kNN. We illustrate this with a schematic in Figure 3-1, which compares estimating uncertainty based on either training or estimated neighbors.

Algorithm 2 describes an uncertainty sampling approach for threshold-based classification with physics-informed maps, using this idea of estimating uncertainty based on estimated neighboring classes. We first seed the training set with n_s randomly chosen points from the pool. Then, at each iteration, we estimate the class of each point in the pool using the physics-informed maps in the current iteration of the library. Each point not in the training set is then given an uncertainty score based on the empirical distribution of the estimated classes of its k_a nearest neighbors in $\mathcal{U}$. At each iteration, we choose a point with the greatest uncertainty score to query. In using this method, we associate uncertainty with proximity to the estimated boundary, and seek query points close to this estimated boundary.

An expensive step in Algorithm 2 is finding the k_a nearest neighbors in $\mathcal{U}$ for each point in $\mathcal{U}$, although this step only has to be performed once. For large pools, one can reduce the expense of this step by approximately finding nearest neighbors [15,58,97]. At each iteration, one also has to search over $|\mathcal{U} \backslash \mathcal{T}|$ points to find one with the largest



(a) Equal uncertainty of two candidate training points (left and right), based on training neighbors

(b) Green candidate training point (right) more uncertain, based on estimated neighbors

Figure 3-1: Example pool $\mathcal{U}$, divided into two estimated classes (red and blue) with estimated boundary (purple dotted line) based on training points (filled-in circles). For two candidate training points (circled in grey (left) or green (right)), the uncertainty score is computed based on classes of indicated (opaque) points. In (a), the classes of the three nearest training neighbors is used to estimate uncertainty, and the two candidate training points have equal uncertainty score. In (b), the estimated classes of the three nearest neighbors in the pool is used to estimate uncertainty; the green-circled candidate (right), which is closer to the estimated boundary, has greater uncertainty.

Algorithm 2 Uncertainty sampling with estimated neighbors (USEN) for threshold-based classification with physics-informed maps.

```
Define number of training (expansion) points $N_e$, number of nearest neighbors $k_a$ for uncertainty estimate, number of randomly selected seed training points $n_s$.
Define pool of offline observations $\mathcal{U} = \{u^{(i)}\}_{i=1}^{|U|}$ and training set of expansion points $\mathcal{T} \subset \mathcal{U}$.
procedure USENTrainingSet($N_e$,$k_a$,$n_s$)
    for $u^{(i)} \in \mathcal{U}$ do
        Find and store indices of its $k_a$ nearest neighbors in $\mathcal{U}$
    end for
    Select $n_s$ points randomly from $\mathcal{U}$ to query and add to $\mathcal{T}$.
    for $i = 1, \dots, |\mathcal{U}|$ do                          ▷Store distances and estimated classes.
        Store distances $\delta^{(i)} = \min_{\tau \in \mathcal{T}} \|u^{(i)} - \tau\|_2$.
        Estimate the class $\tilde{c}^{(i)}$ of observations $u^{(i)}$ using physics-informed maps in
            $\mathcal{T}$ with $k = 1$.
    end for
    for $j = (n_s + 1), \dots, N_e$ do                           ▷Choose one training point at a time.
        for $u^{(i)} \in (\mathcal{U} \backslash \mathcal{T})$ do                     ▷Compute uncertainty scores.
            Compute uncertainty score for $u^{(i)}$ based on the empirical
              distribution of the estimated class of its $k_a$ nearest neighbors.
        end for
        Choose $u^\star \in (\mathcal{U} \backslash \mathcal{T})$ with greatest uncertainty measure to query and
            add to $\mathcal{T}$.                                ▷Add most uncertain point to training set.
        Update distance $\delta^{(\star)}$ and class $\tilde{c}^{(\star)}$ of newest training point $u^\star$.
        for $u^{(i)} \in (\mathcal{U} \backslash \mathcal{T})$ do                 ▷Update distances and estimated classes.
            Update $\delta^{(i)}$ and $\tilde{c}^{(i)}$ if current $\delta^{(i)} > \|u^{(i)} - u^\star\|_2$.
        end for
    end for
end procedure
```

uncertainty score, and to check whether their estimated classifications and distances to the training set need to be updated. These searches can also become expensive for large $|\mathcal{U}|$. One way to reduce this expense is to produce a smaller effective pool using a preprocessing step. For example, in [140], similar points are clustered and a point is chosen to represent them; the collection of the representative points of these clusters gives a compact pool from which to choose query points.

The method described in [140] can also be used directly for active learning in our case, as it requires only $\mathcal{U}$ and the classes of points in $\mathcal{T}$. The method uses query synthesis to avoid searching over the entire pool $\mathcal{U}$, and may be more appropriate when the cost of active learning is dominated by searching and updating over $\mathcal{U}$ rather than querying points (solving the full inverse problems). When the inverse problem solves dominate the cost, Algorithm 2 is better able to explore distant regions of the boundary with small training set sizes N_e.

3.4 Physics-Informed Uncertainty-based Utility Sampling

Rather than focusing on the uncertainty of single points, we can use uncertainty scores to calculate a measure of utility that suggests how querying a point might impact the total uncertainty of the pool of points. A point whose class is uncertain but that is close to few other points may occur in an area of low density, and thus be less likely to impact the classification accuracy of online points that rarely fall nearby. This idea is the motivation behind the uncertainty-based utility sampling approach in [56].

Algorithm 3 describes a similarly motivated uncertainty-based utility sampling approach for threshold-based classification with physics-informed maps. After seeding the training set with a few randomly chosen initial points, we then iteratively add to $\mathcal{T}$ the unlabeled point with the greatest utility. We let the utility of a potential query point $u^{(i)}$ be the total uncertainty of all points in $\mathcal{U}$ that are closer to $u^{(i)}$ than to $\mathcal{T}$. Equivalently, the utility of a point $u^{(i)}$ is the total uncertainty of all points whose nearest training neighbor (and estimated class) would (or could) change if $u^{(i)}$ were added to the training set. The uncertainty of a point is calculated using estimated neighbors, as described in Section 3.3, and the uncertainty of any points in $\mathcal{T}$ is zero. Defining utility in this manner lets us choose training points that may most reduce the total uncertainty of unlabeled points in the pool. In the case where all potential query points have a zero utility score, we recompute the utility scores with the assumption that all unlabeled points have equal, nonzero uncertainty score. This is equivalent to choosing the point $u^{(i)}$ for which there are the most points whose nearest training neighbor (and estimated class) would (or could) change if $u^{(i)}$ were added to the training set.

When the pool $|\mathcal{U}|$ is large, we can use the approaches referenced in Section 3.3 to reduce the cost of searching for nearest neighbors or reduce the size of the effective pool. As the utility measure is more expensive to compute and update than the uncertainty measure, Algorithm 3 is more expensive than Algorithm 2, but potentially achieves a smaller misclassification rate for a given training set size N_e. The two are compared in a numerical example in Section 4.7.1.

3.5 Chapter Summary

In this chapter, we described active learning algorithms that can be used to efficiently choose training points for our physics-informed kNN (see Algorithm 1). For the regression case, we described an active learning approach aimed at minimizing the worst-case error in the QoI estimate. For the classification case, we described two active learning approaches which seek to reduce the misclassification rate by greedily choosing training points near the estimated boundary between classes. In the next chapter, we demonstrate our physics-informed mapping and its active learning algorithms on a nonlinear goal-oriented tomography problem.

Algorithm 3 Uncertainty-based utility sampling with estimated neighbors (utilEN) for threshold-based classification with physics-informed maps.

```
Define number of training (expansion) points $N_e$, number of nearest neighbors $k_a$ for uncertainty estimate, number of randomly selected seed training points $n_s$.
Define pool of offline observations $\mathcal{U} = \{u^{(i)}\}_{i=1}^{|U|}$ and training set of expansion points $\mathcal{T} \subset \mathcal{U}$.
procedure UTILENTRAININGSET($N_e$,$k_a$,$n_s$)
    for $u^{(i)} \in \mathcal{U}$ do
        Find and store indices of its $k_a$ nearest neighbors in $\mathcal{U}$
    end for
    Select $n_s$ points randomly from $\mathcal{U}$ to query and add to $\mathcal{T}$.
    for $i = 1, \ldots, |\mathcal{U}|$ do                    ▷Store distances and estimated classes.
        Store $\delta^{(i)} = \min_{\tau \in \mathcal{T}} \|u^{(i)} - \tau\|_2$.
        Estimate the class $\tilde{c}^{(i)}$ of observations $u^{(i)}$ using physics-informed maps in
            $\mathcal{T}$ with $k = 1$.
    end for
    for $j = (n_s + 1), \ldots, N_e$ do                    ▷Choose one training point at a time.
        for $u^{(i)} \in \mathcal{U}$ do                    ▷Compute uncertainty scores.
            if $u^{(i)} \notin \mathcal{T}$ then
                Compute uncertainty score $\beta^{(i)}$ for $u^{(i)}$ based on the empirical
                 distribution of the estimated class of its $k_a$ nearest neighbors.
            else
                Set $\beta^{(i)} = 0$.
            end if
        end for
        for $u^{(i)} \in (\mathcal{U} \setminus \mathcal{T})$ do      ▷Compute utility scores from uncertainty scores.
            Calculate utility $\gamma^{(i)} = \sum_{k=1}^{|\mathcal{U}|} \beta^{(k)} \mathbb{1}(\delta^{(k)} > \|u^{(i)} - u^{(k)}\|_2)$.
        end for
        if $\max_{u^{(i)} \in (\mathcal{U} \setminus \mathcal{T})} \gamma^{(i)} = 0$ then      ▷Compute alternate utility scores, if necessary.
            for $u^{(i)} \in (\mathcal{U} \setminus \mathcal{T})$ do
                Calculate utility $\gamma^{(i)} = \sum_{k=1}^{|\mathcal{U}|} \mathbb{1}(\delta^{(k)} > \|u^{(i)} - u^{(k)}\|_2)$.
            end for
        end if
        Choose $u^{\star} \in (\mathcal{U} \setminus \mathcal{T})$ with greatest utility to query and add to $\mathcal{T}$. Update
            its corresponding $\delta^{(\star)}$ and $\tilde{c}^{(\star)}$.      ▷Add highest utility point to training set.
        for $u^{(i)} \in (\mathcal{U} \setminus \mathcal{T})$ do      ▷Update distances and estimated classes.
            Update $\delta^{(i)}$ and $\tilde{c}^{(i)}$ if current $\delta^{(i)} > \|u^{(i)} - u^{\star}\|_2$.
        end for
    end for
end procedure
```

Chapter 4

Numerical Experiments and Results

In this chapter, we give numerical results from applying our physics-informed kNN algorithm to an example of goal-oriented tomography. Section 4.1 describes the setup for our numerical experiments. We first consider the case where the single nearest training expansion point is chosen for online QoI estimates. For this case, we present relative timing and error performance in Section 4.2, and error behavior for different methods of map rank reduction in Section 4.3. We then consider the more general case where the QoI estimates from multiple neighboring training points are combined to give a final QoI estimate. In Section 4.4, we explore the effects of choosing and weighting these neighboring estimates with different kernels and distance scores. In Section 4.5, we explore the accuracy of online QoI estimates for different training set sizes when using our physics-informed kNN, compared to several black-box regression approaches. Finally, we explore the effects of generating training sets with active learning approaches. In Section 4.6, we compare QoI errors from using physics-informed kNN with either randomly or actively sampled training sets. In Section 4.7, we compare misclassification rates from using our physics-informed or a black-box approach, with randomly or actively sampled training sets.

4.1 Setup for Goal-Oriented Tomography

We apply Algorithm 1 to non-destructive inspection of objects, such as parts created via additive manufacturing techniques. Parts that are nominally the same, in that they were made with the same materials and patterns, can show significant variability in their material properties [23]. Non-destructive tests can help find imperfections among these parts without damaging them; techniques include tomography with X-rays or ultrasound [35,70,141]. Rapid assessment of the quality of a part soon after or while it is being manufactured can inform decisions as to whether to finish or discard the part, or whether to modify how the next part is made. In our numerical experiments, we simulate the case where observations from past products of a manufacturing process are available in the offline phase; in the online phase, we evaluate products from the same manufacturing process by computing QoI for their observations.

In X-ray computed tomography, one examines the interior of an object by passing an X-ray beam through it; by comparing the emitted and received intensities, one can infer the attenuation coefficient function f of the object. Different materials have different attenuation coefficients for different energy levels. We consider two models, describing monochromatic and bichromatic cases.

In the monochromatic case, the beam is composed of photons of a single energy level. Following [96], we consider f over a rectangular domain and discretize it into n_x piecewise-constant pixels, so that f_j is the value in pixel j. Then the observed log ratio of received-to-emitted intensities for ray i is

$$y_i = \sum_{j=1}^{n_x} a_{ij} f_j, \tag{4.1}$$

where a_{ij} is the distance that ray i travels through pixel j. Although inferring directly for f would give a linear inverse problem, doing so would allow for unrealistic negative attenuation coefficients. Since Algorithm 1 does not produce inferred parameters which we can examine for reasonableness, we enforce positivity in f by inferring for $x = \log(f)$. This setup gives a nonlinear observation operator defined by $\mathcal{O}_e(x) =$

$\{a_{ij}\} \exp(x)$.

Most X-ray sources emit a spectrum of energies; we consider the simplest multi-energetic case with a bichromatic beam of two energy levels. Let f be discretized as in the monochromatic case, and let the emitted intensity be equal for both energy levels. Following [116], the observed ratio of received to emitted intensities for ray i is

$$y_i = \exp\left(-\sum_{j=1}^{n_x} a_{ij} f_j^{(1)}\right) + \exp\left(-\sum_{j=1}^{n_x} a_{ij} f_j^{(2)}\right), \tag{4.2}$$

where $f_j^{(1)}$ and $f_j^{(2)}$ are the attenuation coefficient in pixel j at the two energy levels. To enforce positivity in the attenuation coefficients and to have different attenuation at different energy levels, we let $f^{(1)} = \exp(x)$ and $f^{(2)} = 2\exp(x)$, where x are the inferred parameters.

The domain Ω is a 2×2 square centered at the origin and discretized into a square grid. We have n_{src} photon sources on one side of the object, evenly spaced along an origin-centered arc of radius 4 to give 90 degrees of illumination; each source emits a fan of n_{rays} equally spaced rays which pass through the domain and are observed by receivers on the other side. This setup is illustrated in Figure 4-1.

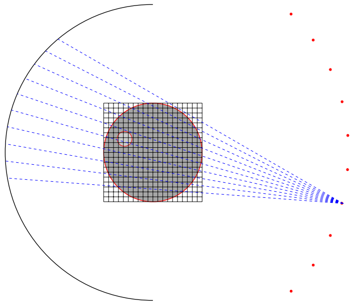


Figure 4-1: Setup of X-ray tomography example. Sources are represented by red stars, detectors by black arc, and rays from one source by blue dotted lines. Discretized domain contains circular object with a randomly placed and sized circular anomaly with random reduced attenuation coefficient.

Training and test observations are generated analytically from non-discretized images containing a unit circle of uniform attenuation coefficient. Each circular object contains a circular anomaly whose location (polar coordinates) and size are sampled from scaled uniform distributions; each anomaly has density distributed as $\mathcal{U}(0, 0.2)$ and is contained completely within the unit circle. We then add Gaussian white noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, where $\sigma = 10^{-4}$, to simulate noisy observations from previously available objects.

Common choices of regularization matrix R include the identity (to encourage small parameter values), the first difference operator (to encourage small spatial changes in parameter), and the second difference operator (to encourage smoothness in parameter values) [136]. We use a diagonal regularization matrix, with magnitude 0.05 for pixels inside the unit circle, and heavier weightings (up to 0.5) for pixels outside the unit circle to represent the belief that these pixels correspond to the air around the object.

We consider two QoI prediction outputs of the form

$$z = \int_{\Omega} q(x(\xi, \zeta), \xi, \zeta) \, \mathrm{d}\xi \mathrm{d}\zeta \tag{4.3}$$

where ξ and ζ are spatial coordinates and q is some continuous, sufficiently differentiable function; this gives a prediction output that is additively separable in terms of the discretized parameters. First, we consider the case where one has a target pattern x_{target} to match, where x_{target} corresponds to the pristine circle. We use

$$q(x, \xi, \zeta) = w(\xi, \zeta) \exp\left(-a(x - x_{\text{target}})^2\right), \tag{4.4}$$

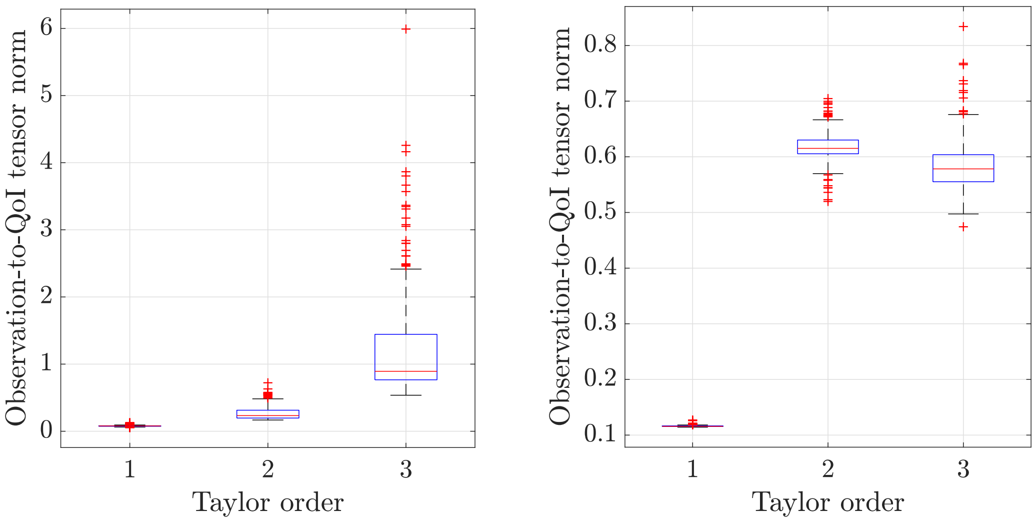
where weighting $w(\xi, \zeta)$ reflects the increased importance of the region inside the unit circle compared to the rest of the domain, and $a = 2$ controls how heavily deviations from the target are penalized.

As a second prediction output, we consider the case where a part is evaluated based on its predicted performance; anomalies may be acceptable if they are in areas not critical to the performance of the part. Suppose the domain represents the cross-section of a slender beam with uniform cross-section throughout its length. Slender beam theory predicts the beam's deflection given some specified loading configuration using only its length, details of the loading configuration, and the area moment of inertia of the cross-section. Assuming small anomalies, the area moment of inertia about the neutral axis is approximated with the area moment of inertia about the origin, giving

$$q(x, \xi, \zeta) = E(x)\xi^2, \tag{4.5}$$

where $E(x)$ is the modulus of elasticity. For the purpose of testing Algorithm 1, we let $E(x) = f(x)$; in practice, numerical simulations or experiments can be used to better determine the relationship between attenuation coefficient and modulus of elasticity for the objects of interest (for example, tomography can be used to infer bone mineral density, which is experimentally correlated with Young's modulus [138]).

In the following numerical experiments, we consider the bichromatic inverse problem with the target mismatch QoI, and/or the monochromatic inverse problem with the area moment of inertia QoI. While both prediction outputs are nonlinear in the observations, the nonlinearity of the former is stronger, as suggested by the Frobenius norms of the observation-to-QoI tensors A_d (defined in Equation (2.16)). These norms for the two goal-oriented inverse problems, for the setup in Section 4.3.1, are shown in Figure 4-2.



(a) Bichromatic, target mismatch

(b) Monochromatic, area moment of inertia

Figure 4-2: Distribution over training expansion points of Frobenius norm of A_d for various Taylor orders d, for (a) target mismatch QoI of bichromatic inverse problem and (b) area moment of inertia QoI of monochromatic inverse problem.

Unless otherwise noted, we use the approximate QoI expression given by Equation (2.15). In all sections other than Section 4.3, we use the additive separability of the QoI to generate observation-to-QoI maps as described in Section 2.4.3, without any additional map rank reduction. Tensor operations are performed using the Tensor Toolbox [6].

4.2 Timing and Error Comparison

Consider the bichromatic inverse problem with the target mismatch QoI. We discretize the domain with a 32 × 32 grid ($n_x = 1024$) and model $n_{\text{src}} = 10$ sources with $n_{\text{rps}} = 10$ rays per source ($n_y = 100$). In the offline stage, we generate observation-to-QoI maps for 256 expansion points corresponding to training observations, generated as described in Section 4.1. We test the algorithm on 500 similarly generated test observations.

Figure 4-3 compares solving the full inverse problem for the test observations to applying our observation-to-QoI maps for different Taylor orders t in approximating $\mathcal{O}_p$; we choose the nearest expansion point based on distance in y. For each test observation y, the relative QoI error is computed as $|(\tilde{z}^\star(y) - z^\star(y))/z^\star(y)|$, where $\tilde{z}^\star(y)$ and $z^\star(y)$ are the estimated and exact QoI, respectively. The full inverse problem is solved using Matlab's `fminunc`; we use the subspace trust-region method, where the subspace in each iteration is approximately obtained using preconditioned conjugate gradients. The gradients and Hessian-vector products needed are calculated analytically and without forming the full Hessian.

Increasing the Taylor order t generally gives reduced time savings, but potentially less error in the QoI, compared to solving the full inverse problem. In this case, increasing the order t from $t = 1$ to $t = 2$ gives a 30% reduction in average relative QoI error; increasing the order further from $t = 2$ to $t = 3$ gives another 10% reduction in average relative QoI error. Note that there is no guarantee that using a higher-order representation will monotonically decrease QoI error for all QoI outputs or for a particular online observation y.

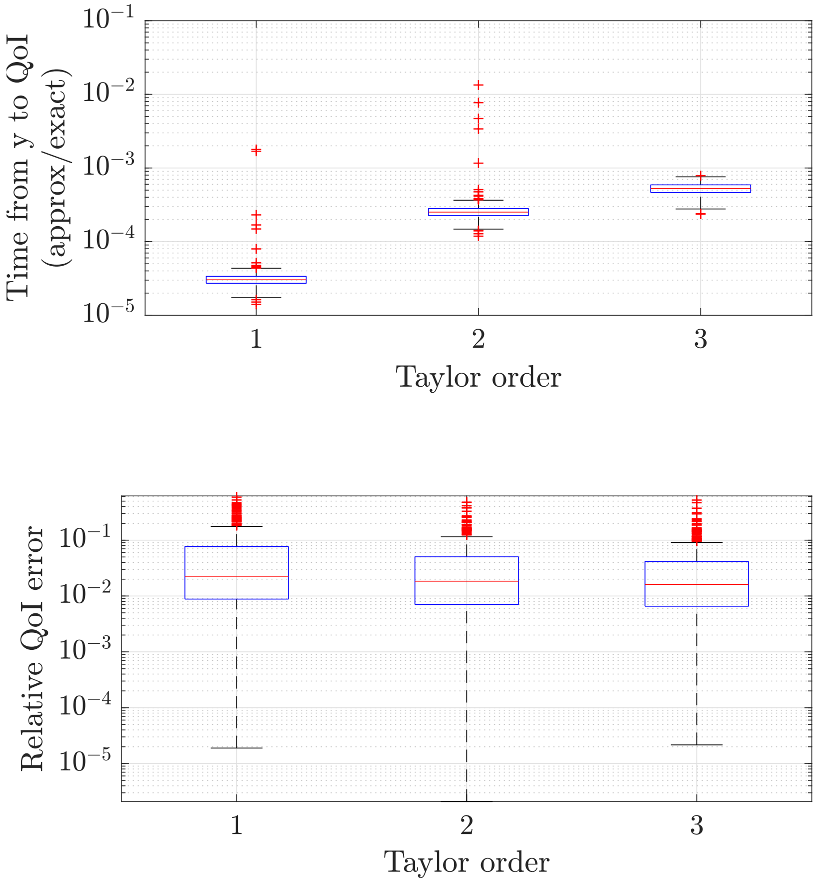


Figure 4-3: Top: Ratio of time needed to obtain QoI (approximate (using observation-to-QoI maps) to exact (using full inverse solve)) for different orders of Taylor approximation of $\mathcal{O}_p$. Bottom: QoI error (relative to exact values) from using observation-to-QoI maps for different orders of $\mathcal{O}_p$ approximation.

On average, for $t = 3$, the relative QoI error is less than 4%, obtained in less than 0.1% of the time necessary to solve the full inverse problem and apply $\mathcal{O}_p$ to the inferred parameters. A QoI estimate with $t = 3$ takes on average the time of three evaluations of the inverse problem objective function. Fully solving the inverse problem requires an average of 24 iterations and the time of 4000 objective function evaluations. In this particular example, using expansion order $t = 1$ may best balance online evaluation time and QoI estimation accuracy. We will approximate $\mathcal{O}_p$ with $t = 3$ for the numerical experiments in the rest of this chapter.

4.3 Reduced Rank Maps

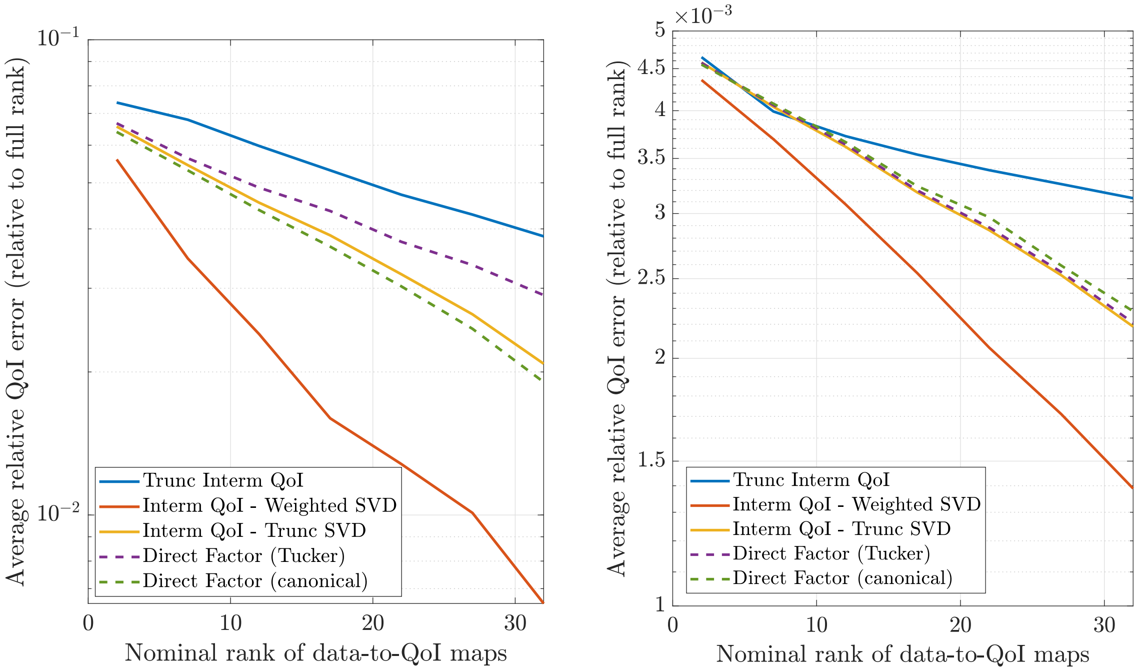
As discussed in Sections 2.4.2 and 2.4.3, when n_y and/or t are large, storage and/or online application cost limitations may require that we use lower-rank approximations of the observation-to-QoI maps. In Section 4.3.1, we compare different rank-reduction approaches when applied to the tensors A_d of the approximate QoI expression given by Equation (2.21). In Section 4.3.2, we give numerical evidence that reducing the rank of the tensors in Equation (2.15) gives better behavior than similarly approximating the tensors of the equivalent expression in Equation (2.21).

4.3.1 Comparison of Factorization Approaches

In this example, we discretize the domain with a 32×32 grid ($n_x = 1024$), and model $n_{\text{src}} = 10$ photon sources with $n_{\text{rps}} = 10$ rays per source ($n_y = 100$). We have 256 expansion points from the offline stage, and consider the average error over 500 test observations. For the each of the test observations, we choose the nearest expansion point ($k = 1$). For a chosen reduced rank r, one can directly factor the tensors A_d of derivatives of the QoI with respect to observations (see Section 2.4.2), or, since $\mathcal{O}_p$ is additively separable, one can either truncate the dimension of the intermediate QoI or reduce the rank of the observation-to-intermediate-QoI maps (see Section 2.4.3).

In Figure 4-4, we compare the average QoI error (relative to the QoI estimate without any rank reduction) when reducing rank with these approaches, for rank

$r = 2$ to $r = 32$. We compute a symmetric Tucker decomposition using an iterative method based on [110] and provided by the Tensor Toolbox [6], and compute an optimal non-symmetric canonical decomposition using a nonlinear least-squares algorithm provided by the Tensorlab toolbox [137]; we use the default parameter settings. When used for rank reduction, the observation covariance Γ_Y is calculated from the offline training observations.



(a) Bichromatic, target mismatch

(b) Monochromatic, area moment of inertia

Figure 4-4: Average relative QoI error for different approaches (truncated intermediate QoI; intermediate QoI with weighted SVD or truncated SVD rank reduction; direct factorization (canonical or Tucker)) to reducing storage of observation-to-QoI maps, for different values of reduced rank r.

For fixed r, directly factoring A_d with a canonical factorization gives comparable storage and online application costs to truncating the dimension of the intermediate QoI. When $\mathcal{O}_p$ is additively separable, truncating the intermediate QoI gives equivalent observation-to-QoI maps to a truncated canonical decomposition of A_d, which is generally suboptimal for given rank when $d > 2$; this inefficiency is visible in Figure 4-4. When $\mathcal{O}_p$ is additively separable, both approaches will give exact recovery for some $r \leq n_x$. In general, the canonical decomposition may require rank larger than

n_x for exact recovery, and may also encounter convergence issues with computing the factorization, depending on the algorithm used and the degeneracy of A_d.

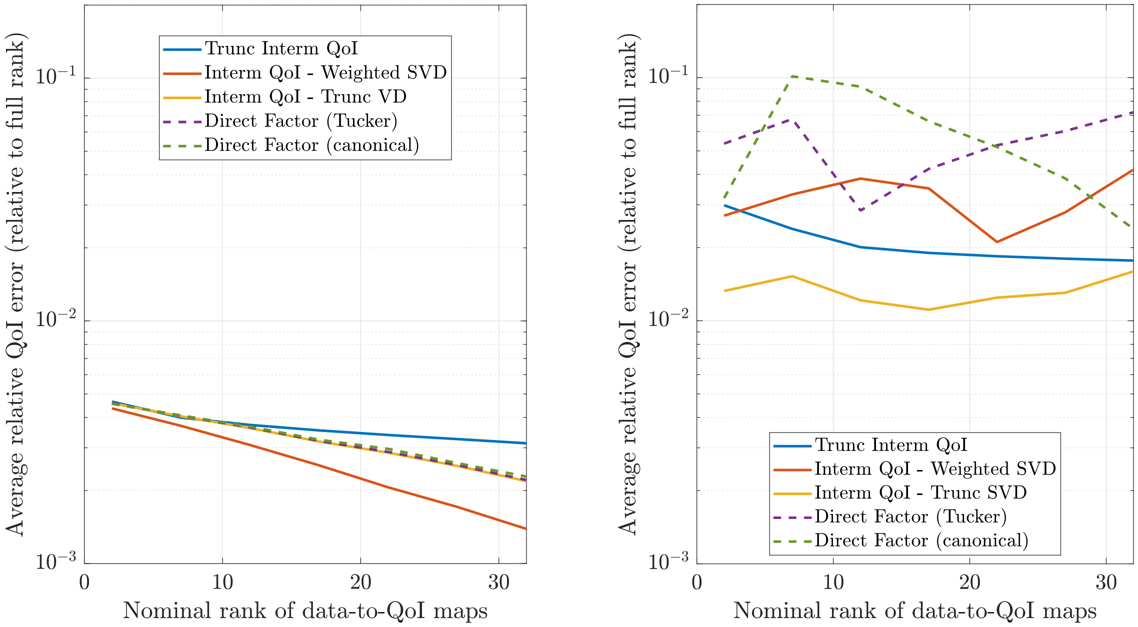
For fixed r, factoring A_d with the Tucker decomposition gives comparable storage requirements to reducing the rank of the observation-to-intermediate-QoI maps. Compared to direct canonical factorization and truncation of the intermediate QoI, these approaches have greater storage and online application costs for given r ($\mathcal{O}(r^d + kn_y)$ compared to $\mathcal{O}(r + rn_y + d)$), but induce no additional error from rank reduction by $r = n_y$. If this extra cost for fixed r is acceptable and $\mathcal{O}_p$ is additively separable, Figure 4-4 suggests that rank reduction using the weighted SVD is both most efficient and cheaper to compute than direct Tucker factorization of A_d.

4.3.2 Factoring Tensors in Equivalent Expressions

Equation (2.15) and Equation (2.21) give equivalent expressions for the approximate QoI in terms of the observations, using different groupings that result in different tensors A_d and $\hat{A}_d$. For given order d, both A_d and $\hat{A}_d$ are symmetric and of the same size, and thus amenable to the same approaches for reducing their storage and application costs using lower-rank approximations. We previously noted in Section 2.3 that Equation (2.21) is more robust to coarse solutions of the inverse problem at expansion points. Here, we give numerical evidence that this robustness comes at the cost of less favorable error behavior when its tensors $\hat{A}_d$ are approximated with reduced-rank approximations. We use the setup described in Section 4.3.1 and consider the monochromatic inverse problem with the area moment of inertia QoI.

In Figure 4-5, we compare the average QoI error (relative to the QoI estimate without any rank reduction) when reducing rank with the approaches described in Sections 2.4.2 and 2.4.3, for rank $r = 2$ to $r = 32$. We see that, for a given r, factoring A_d from Equation (2.15) gives less relative error than factoring $\hat{A}_d$ from Equation (2.21). Factoring A_d also gives better decay in error as r increases.

Regardless of which equivalent expression we are factoring the coefficient tensors of, there is no guarantee that a higher r will give less error. Additionally, although their error behaviors differ at small r, the tensors A_d and $\hat{A}_d$ have the same size,



(a) Separate derivative terms

(b) Combined derivative terms

Figure 4-5: Average relative QoI error versus reduced rank r for different approaches to reducing storage of observation-to-QoI maps, when factor expression uses (a) separate or (b) combined derivative terms.

and thus share an upper bound on the r necessary for exact recovery, for any of the approaches besides direct canonical decomposition. For this experiment, the error from reducing the rank of the observation-to-intermediate-QoI maps for $\hat{A}_d$ does not decay consistently below the error at $r = 2$ until after about $r = 70$ (not shown in range of Figure 4-5). Experiments with smaller n_y empirically confirm that the decomposition approaches are exact at ranks matching the theoretical upper bound (reviewed in Sections 2.4.2 and 2.4.3). They also support the observation that factoring A_d gives less relative error than factoring $\hat{A}_d$ for given r, although the error decay at smaller r when factoring $\hat{A}_d$ is less stagnant than in Figure 4-5.

The behavior observed in Figure 4-5 (and in numerical experiments with smaller n_y) suggests that if cost constraints require the tensor be stored approximately with a reduced rank, then using Equation (2.15) is preferable. Otherwise, Equation (2.21) is preferable in providing robustness to possible convergence issues when solving the full offline inverse problems.

4.4 Choosing and Combining Local Approximations

In this section, we numerically explore different approaches to weighting QoI estimates from different expansion points in the library (see Section 2.6). Section 4.4.1 explores the use of different kernels to weight the QoI estimates from different local approximations in the library. Section 4.4.2 explores the use of an error estimate in choosing and weighting local approximations.

4.4.1 Kernel Comparison

As noted in Section 2.6, there are many kernel functions that can be used to encode the relative importance of nearby and distant points. We consider three common choices: inverse distance ($\phi(\delta_i) = (\delta_i)^{-1}$), Gaussian ($\phi(\delta_i) = \exp(-\delta_i^2)$), and equal ($\phi(\delta_i) = 1$), where $\delta_i(y) = \|y - y^{(i)}\|_2$ is the distance between the online observations y and the expansion point $y^{(i)}$. We also consider an inverse distance kernel where the distance between the observations and expansion point is measured with a fractional norm:

$$\phi(\delta_i) = (\delta_i)^{-1}, \quad \delta_i(y) = \left(\sum_{j=1}^{n_y} |y_j - y_j^{(i)}|^p\right)^{1/p}, \tag{4.6}$$

with $p = 0.5$. Using a fractional norm can reduce the error from choosing and interpolating nearest neighbors in high-dimensional spaces [1,16].

In this example, we discretize the domain with a 16 × 16 grid ($n_x = 256$), and model $n_{\text{src}} = 10$ photon sources with $n_{\text{rps}} = 5$ rays per source ($n_y = 50$). We compute the average QoI error over 500 test observations. We consider the monochromatic inverse problem with the area moment of inertia QoI. Figure 4-6 shows the average relative QoI for different training set sizes and choices of kernel. For each training set size and kernel, we plot the best average error among four choices of number of nearest neighbors to average over ($k = 1, 2, 4, 8$).

We see that different choices of kernel give little, if any, improvement over the inverse 2-norm distance kernel. In this experiment, using a fractional norm to find and weight nearest neighbors does not give improved performance over the Euclidean

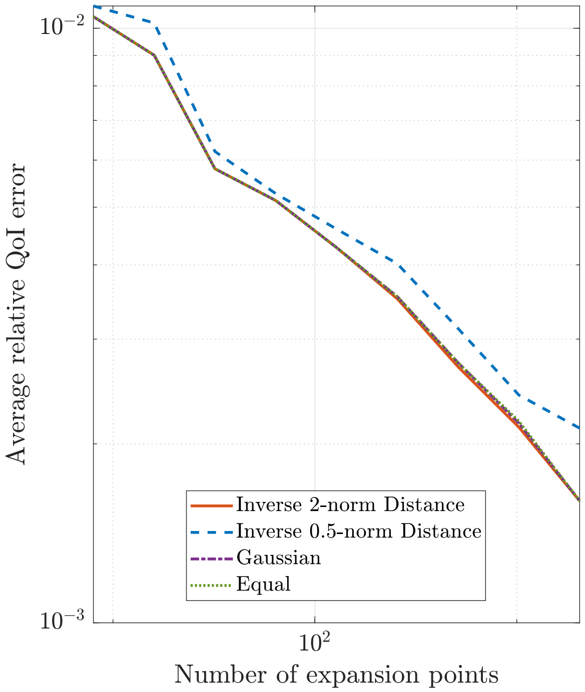


Figure 4-6: Average relative QoI error for different training set sizes and kernels.

norm. It may be possible to obtain different relative performance with more intensive tuning of k and the addition of a bandwidth term to scale the distances for different kernels. For the numerical experiments presented in the rest of this chapter, we will use the inverse distance kernel with the Euclidean distance in observation space, unless otherwise stated.

4.4.2 Choosing Points with Error Estimate

For a given kernel choice, one can potentially obtain a more accurate QoI estimate by using a more general ‘Kdistance score’ to choose nearest neighbors and to weight their QoI estimates. As suggest in Section 2.6, this distance score can include an estimate of the error from using different expansion points, so that expansion points giving more accurate QoI estimates are chosen and weighted more heavily. Our suggested first-order error estimate can also be interpreted as the projected distance between observations, in a subspace defined by the gradient.

Using the same setup as in Section 4.4.1, we compare the QoI error from using the inverse distance kernel, using two types of distance score to choose and weight the QoI estimates from neighboring expansion points. First, we let the distance score for online observations y and an expansion point $y^{(i)}$ simply be the Euclidean norm of their difference. Second, we let the distance score be a weighted sum of the Euclidean distance and the estimated error, as described in Section 2.6; we use the weighting $\alpha = 50$. We compute the offline gradient for each training point by solving Equation (2.35) with $n_g = 100$ finite difference points, chosen randomly from a sphere of radius 10^{-4} centered at the training point.

Figure 4-7 compares the average relative QoI error from using these two notions of distance scores. We see that using the error estimate to choose and weight neighboring estimates is most helpful when the training set is large. For the largest training set, including the error estimate gives a 20% reduction in error compared to using only the Euclidean distance between observations. The error estimate is most accurate locally, and is most useful for choosing between expansion points which are similarly close to the online observations in the Euclidean sense, but which lie in directions where the QoI error is differently sensitive to changes in the observations. When the training set is small, it is unlikely that there will be multiple expansion points with similar distances to the observations in the Euclidean sense; in this case, including the error estimate in choosing nearest neighbors will make little difference.

We note that not all values of α will give improved error compared to using only the Euclidean distance. If α is too small, the inclusion of the error estimate will not noticeably affect the choice and weighting of nearest neighbors. If α is too large, the error estimate may drastically underestimate the error from distant expansion points and choose them as nearest neighbors, in which case including the error estimate may give more QoI error than just using the Euclidean distance. Tuning of α can be performed with cross-validation in the offline phase. Cross-validation generalizes poorly for small training sets, but in this case, including the error estimate is unlikely to improve the QoI for any α anyways.

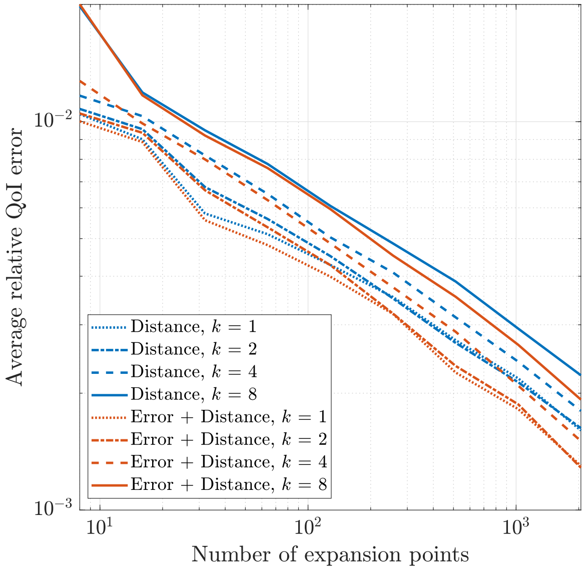


Figure 4-7: Average relative QoI error when error estimate is or is not considered when choosing and weighting estimates from nearest neighboring expansion points.

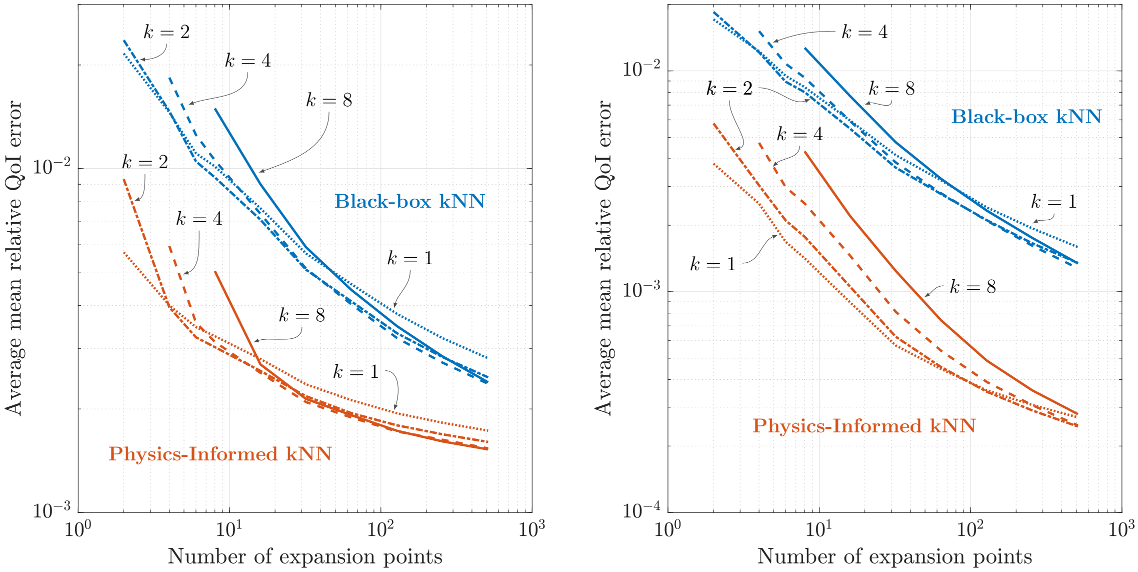
4.5 Comparison with Black-Box Interpolation

Viewing the QoI $z^\star(y)$ as a function of observations y, we can apply black-box regression techniques to estimate $z^\star(y)$ based on training pairs $(y^{(i)}, z^\star(y^{(i)}))$. Section 4.5.1 compares our approach to kNN for different training set sizes, and Section 4.5.2 compares our approach to several black-box regression techniques in the case where training observations do not well represent online test observations.

4.5.1 Physics-Informed vs Black-Box k-Nearest-Neighbors

As noted in Section 2.6, our approach bears similarities to kNN, a black-box approach that estimates the unknown value of a function $(z^\star(y))$ at a query point (y) as a weighted average of the values at neighboring points $(z^\star(y^{(i)}))$, where the weighting is usually based on the distance between the query point (y) and its neighbors $(y^{(i)})$.

In this example, we compare physics-informed kNN (Equation (2.40)) to black-box kNN (Equation (2.41)). We discretize the domain with a 16×16 grid ($n_x = 256$), and model $n_{\text{src}} = 5$ photon sources with $n_{\text{rps}} = 3$ rays per source ($n_y = 15$). From a pool of 2048 potential training points, we generate training sets by randomly selecting subsets of up to 512 training points. For each training set, we calculate the mean QoI error over 500 test points; we then average this mean error over 40 random training sets. In Figure 4-8, we plot this average mean QoI error for our physics-informed kNN and for black-box kNN.



(a) Bichromatic, target mismatch

(b) Monochromatic, area moment of inertia

Figure 4-8: Average mean QoI error over test observations for (a) target mismatch and (b) area moment of inertia QoI outputs using physics-informed or black-box kNN for different choices of number of nearest neighbors k.

In the examples considered, our physics-informed kNN requires fewer training expansion points than black-box kNN to achieve a given level of accuracy; in order to obtain the level of accuracy achieved by physics-informed kNN with eight training points, black-box kNN requires at least 128 training points. Physics-informed kNN takes averages over QoI estimates that are based on local approximate problems, built using knowledge of the physics and structure of the full goal-oriented inverse problem. Using this additional knowledge allows physics-informed kNN to achieve reduced QoI error compared to black-box kNN with fewer training points.

4.5.2 Training-Test Mismatch

We compare the performance of our Algorithm 1 to black-box interpolation approaches when the online test observations and offline training observations are generated by different processes. In addition to black-box kNN, we will also consider two other linear smoothers [26] for black-box regression. One comparison case is radial basis function interpolation, in which $z^\star(y)$ is approximated by

$$\tilde{z}^\star(y) = \sum_{i=1}^{N_e} w_i \phi(\|y - y^{(i)}\|_2), \tag{4.7}$$

where ϕ is a radial basis function and the weights w_i are chosen to satisfy

$$z^\star(y^{(j)}) = \sum_{i=1}^{N_e} w_i \phi(\|y^{(j)} - y^{(i)}\|_2), \quad \forall j = 1, \ldots, N_e. \tag{4.8}$$

Solving for the weights requires a linear solve of a size N_e system, but this can be performed in the offline stage. The radial basis function ϕ describes how much influence the QoI at some observation has on nearby observation values. A common choice for ϕ is the multiquadratic $\phi(\delta) = (\delta^2 + \delta_0^2)^{1/2}$ [108], where δ_0 is a scale factor we choose to be equal to the average nearest-neighbor distance.

A second comparison case is regression with basis functions fitted using the procedure described in [111]. Let k be a chosen number of nearest neighbors to interpolate over, and n_q be the number of nearest neighbors with which to build a quadratic basis function. The QoI is approximated by

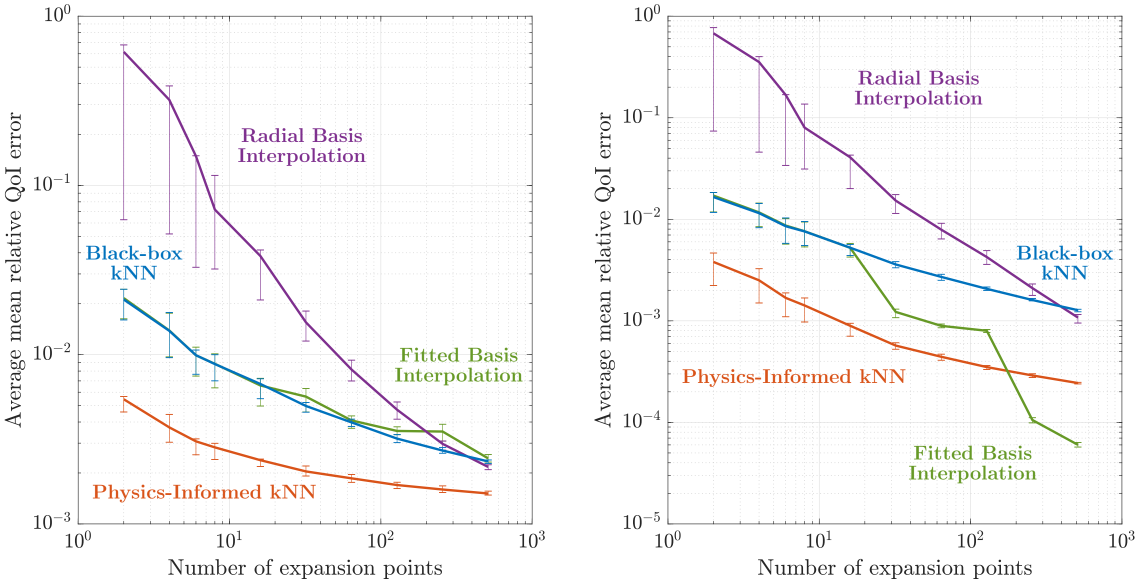
$$\begin{aligned} \tilde{z}^\star(y) &= \sum_{i \in \mathcal{K}_i} \frac{w_i(y)}{\sum_{j \in \mathcal{K}_i} w_j(y)} \phi_i(y), & (4.9)\\ w_i(y) &= \left(\frac{R - \|y - y^{(i)}\|_2}{R\|y - y^{(i)}\|_2} \right)^2, \quad R = \max_{i \in \mathcal{K}_i} \|y - y^{(i)}\|_2, & (4.10)\\ \phi_i(y) &= y^T A_{(i)} y + b_{(i)}^T y + c_{(i)}, & (4.11) \end{aligned}$$

where $\mathcal{K}_i$ are the k nearest neighbors of $y^{(i)}$. The basis functions are fitted in the sense that $A_{(i)} \in \mathbb{R}^{n_y \times n_y}$, $b_{(i)} \in \mathbb{R}^{n_y}$, $c_{(i)} \in \mathbb{R}$ are computed for each of the N_e

known $(y^{(i)}, z^\star(y^{(i)}))$ pairs by means of a least-squares fit on their n_q nearest neighbors while enforcing $\phi_i(y^{(i)}) = z^\star(y^{(i)})$. Based on the recommendations of [18], we use $n_q = 1.5 n_{\text{DoF}}$ and $k = \min(1.5 n_q, N_e)$, where n_{DoF} is the number of degrees of freedom in the basis functions being fitted; we fit linear or constant basis functions instead when there are not enough training points.

We compare the performance of Algorithm 1 and the various black-box interpolation methods in the case where the training and test observations are generated by the same process, and in the case where they are not. The latter simulates a case where the manufacturing process has changed from that used to generate the training objects. We use the same experimental setup as in Section 4.5.1. In the case of training-test mismatch, the training observations are generated from circular objects with one random anomaly, but the test observations are generated from circular objects with two random anomalies. The results for the case where the training and test observation distributions do and do not match are shown in Figure 4-9 and Figure 4-10, respectively. For physics-informed and black-box kNN, for each training set, we select from among $k = 1, 2, 4, 8$ to obtain the lowest mean QoI error over the test points. For each approach considered, at each training set size, we plot the average, and 0.25 and 0.75 quantiles, of these mean QoI errors over the 40 random training sets.

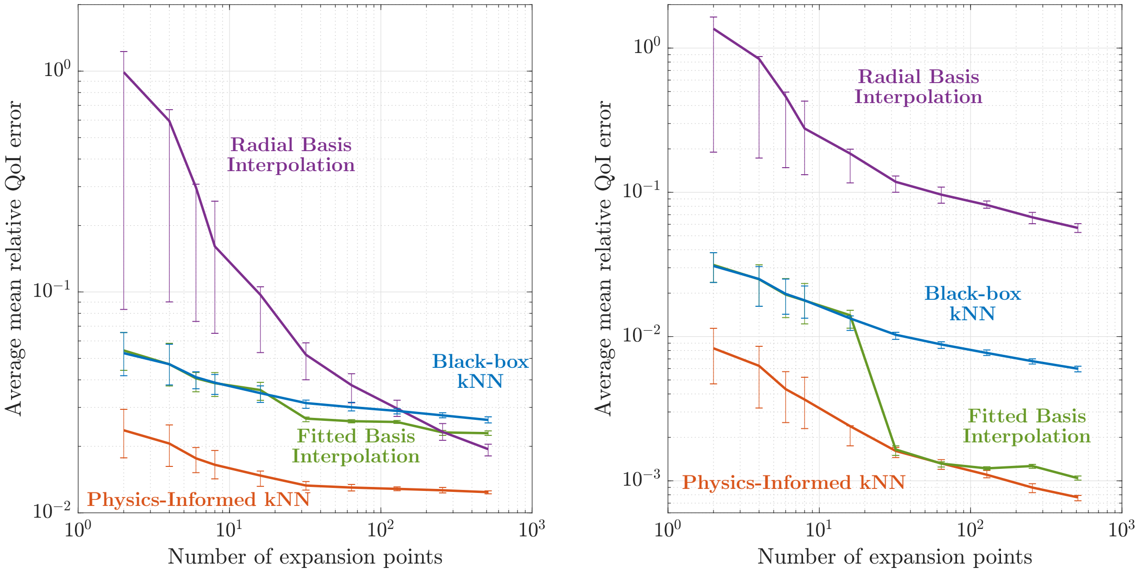
In both cases, the physics-informed approach of Algorithm 1 is more accurate than the black-box approaches with small numbers of training points. When the training observations well represent the test observations, the basis interpolation methods are able to achieve greater accuracy than Algorithm 1 when there are many training points available. However, when the training and test distributions do not match, the improved error of the basis interpolation methods over Algorithm 1 for large training sets is greatly reduced or disappears. Here, physics-informed kNN is more robust to a mismatch in the distribution of training and test observations.



(a) Bichromatic, target mismatch

(b) Monochromatic, area moment of inertia

Figure 4-9: Average mean QoI error over test observations for (a) target mismatch and (b) area moment of inertia QoI outputs using physics-informed kNN or black-box interpolation methods. Vertical bars indicate 0.25 and 0.75 quantiles over random training sets.



(a) Bichromatic, target mismatch

(b) Monochromatic, area moment of inertia

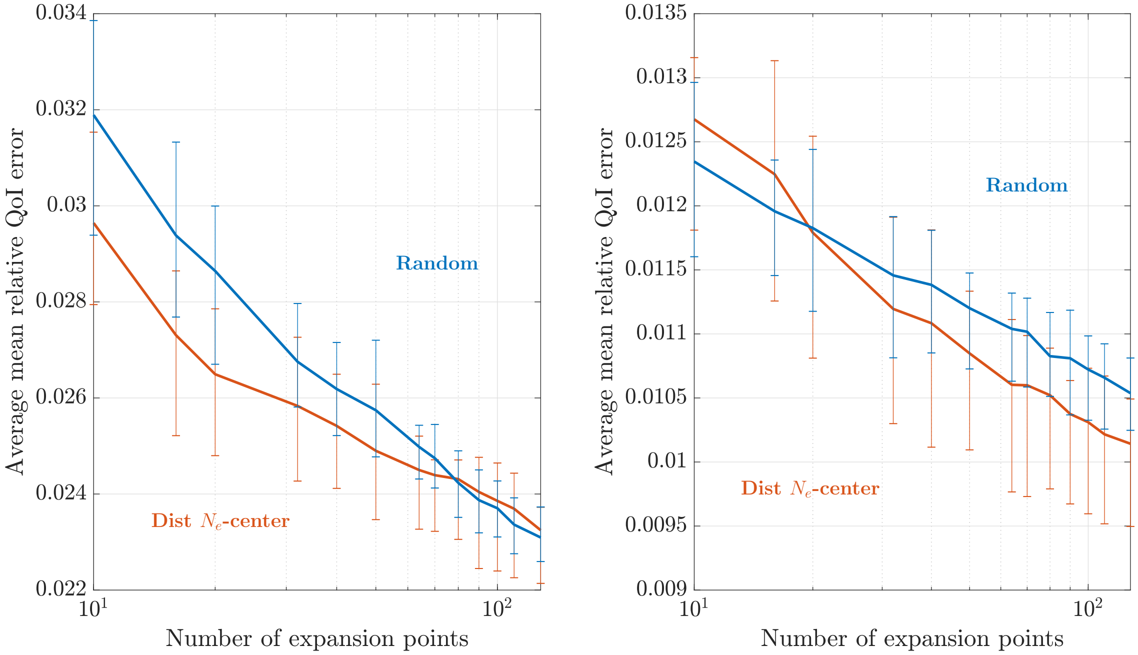
Figure 4-10: Average mean QoI error over test observations for (a) target mismatch and (b) area moment of inertia QoI outputs using physics-informed kNN or black-box interpolation methods when training and test observations are generated from different processes. Vertical bars indicate 0.25 and 0.75 quantiles over random training sets.

4.6 Active Learning for Minimizing Worst-Case Regression Error

For a given training set size, one can obtain more accurate QoI estimates with physics-informed KNN if the expansion points are chosen in a principled fashion, rather than at random. In this example, we discretize the domain with a 16×16 grid ($n_x = 256$), and model $n_{\text{src}} = 10$ photon sources with $n_{\text{rps}} = 5$ rays per source ($n_y = 45$). We start with a pool of 2048 unlabeled points, and randomly select eight initial points to seed the training set. We then pick the remaining training expansion points either at random, or with an active learning approach by greedily solving the N_e-center problem (see Section 3.2)).

In Figure 4-11, the QoI error from using these training sets is averaged over 500 test points; this average QoI error is then averaged for 45 randomly seeded training sets. For each training set, we select from among $k = 1, 2, 4, 8$ to obtain the lowest mean QoI error over the test points. We see that for given training set size N_e, our proposed active learning approach usually gives small improvements in average QoI error, compared to random selection. The performance of the active learning approach varies with the random seed training points, especially at small training set sizes. If instead we examine maximum error, which our active learning approach is designed to minimize, we see a larger gap in the performance of the active and random approaches. In Figure 4-12, we take the largest QoI error incurred over the test points, then average this maximum error over the 45 randomly seeded training sets. For each training set, we select from among $k = 1, 2, 4, 8$ to obtain the lowest maximum QoI error over the test points. For most training set sizes, using the active learning approach does indeed give better average maximum QoI error than random selection. The active learning approach also gives less variability in the maximum QoI error over the random training set seeds.

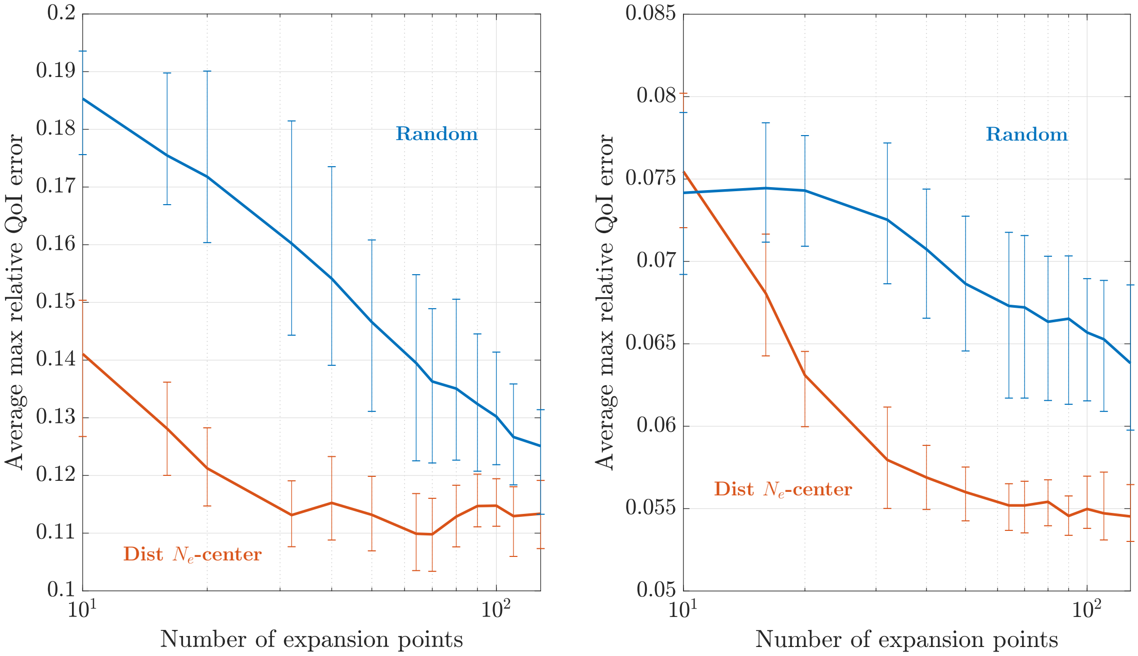
As with all active learning approaches, there is no guarantee that the proposed active learning approach will outperform random selection for all goal-oriented inverse problems and sets of offline and online observations. In particular, the performance of



(a) Bichromatic, target mismatch

(b) Monochromatic, area moment of inertia

Figure 4-11: Average mean QoI error over test observations for (a) target mismatch and (b) area moment of inertia QoI outputs using physics-informed kNN with training sets generated by random selection or active learning by solving the distance-based N_e-center problem. Vertical bars indicate 0.25 and 0.75 quantiles over training sets with different random seeds.



(a) Bichromatic, target mismatch

(b) Monochromatic, area moment of inertia

Figure 4-12: Average maximum QoI error over test observations for (a) target mismatch and (b) area moment of inertia QoI outputs using physics-informed kNN with training sets generated by random selection or active learning by solving the distance-based N_e-center problem. Vertical bars indicate 0.25 and 0.75 quantiles over training sets with different random seeds.

the proposed active learning approach depends on the quality of the offline observation pool and the nonlinearity of the goal-oriented inverse problem. The proposed greedy active approach benefits from an offline observation pool whose spread well represents that of the online test observations, although larger pools will increase the cost of choosing each training point. Recall that in deriving the active learning approach (see Section 3.2), the distance between observations is used as a proxy for the QoI error incurred by using one as an expansion point for the other. Approximating the error minimization problem with the distance minimization problem may not work well for highly nonlinear predictions and/or inverse problems.

Even when the offline pool is representative of the online test points, and the distance minimization problem well approximates the error minimization problem, the greedy algorithm may not optimally solve the distance minimization problem. However, compared to random selection, the greedy algorithm is more consistent, and its performance relative to the optimum has a known bound.

4.7 Active Learning for Classification

In this section, we give numerical results comparing misclassification rates from using our physics-informed or a black-box approach, with randomly or actively sampled training sets, for threshold-based classification. Section 4.7.1 compares different active learning approaches for generating the training set of physics-informed kNN. Section 4.7.2 compares the performance of combinations of physics-informed and black-box offline sampling and online classification approaches.

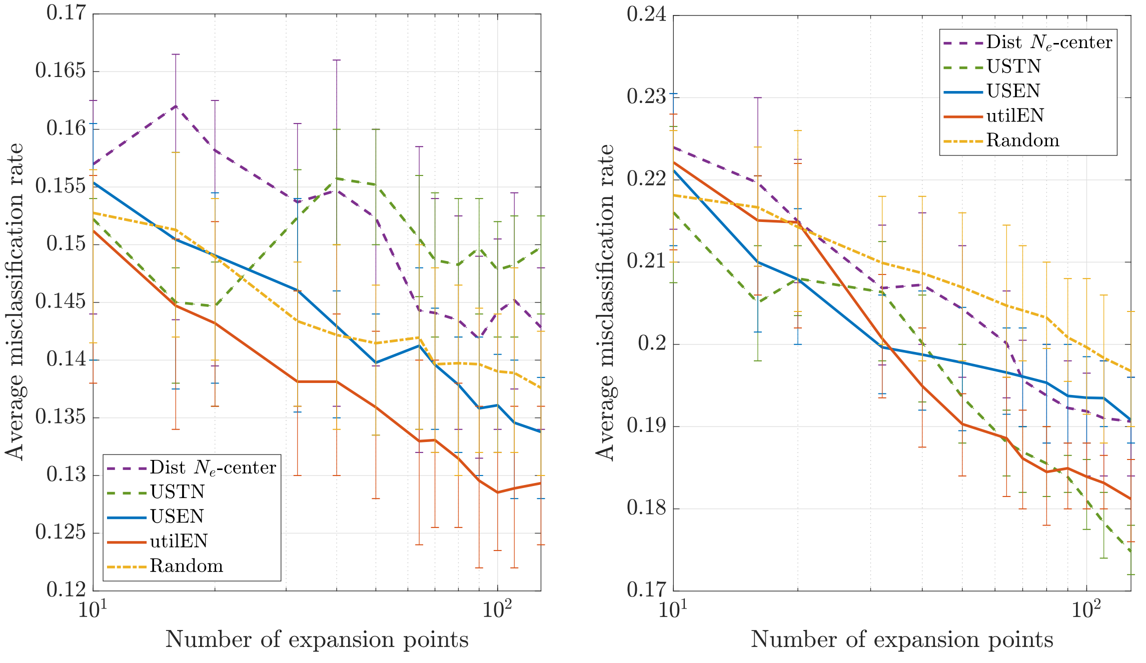
4.7.1 Active Learning for Physics-Informed Classification

As described in Section 3.3, when the continuous QoI of the inverse problem is used for classification by thresholding, one can perform active learning by choosing training points close to the estimated boundary between classes. In this example, we use the same setup as in Section 4.6, with the same test points. We then pick the remaining training expansion points either at random, or with one of the active learning approaches described in Sections 3.2 to 3.4. We use $k_a = 5$ neighbors to determine uncertainty for the USEN and utilEN active sampling approaches. For each training set, we estimate the class of each of the 500 test points by estimating its QoI using physics-informed kNN with that training set, and then calculate the misclassification rate. Misclassification from using our physics-informed maps is determined relative to the class from solving the full inverse problem, not from the truth parameters used to generate the observations. We then average the misclassification rate over 45 randomly seeded training sets.

We consider two types of thresholding to map the continuous QoI to a class. First, we consider a single QoI threshold that divides the offline pool into two classes based on whether their QoI is greater or less than the threshold. We can interpret this as an object being acceptable if it sufficiently matches the target (target mismatch QoI), or if it sufficiently resists bending (area moment of inertia QoI). We choose the threshold so that 40% of the offline pool points belongs to one class, and 60% to the other. Figure 4-13 describes the average misclassification rate from applying a single threshold to the QoI estimate of physics-informed kNN, with randomly or actively sampled training sets; we show the lowest misclassification rate among online number of nearest neighbors $k = 1, 2, 4, 8$.

Second, we consider a classification mapping with two thresholds, so that the QoI values between them correspond to one class, and other QoI values are mapped to the other class. We apply this mapping to the area moment of inertia QoI, which can be interpreted as designating an object acceptable if it is neither too stiff nor too pliable (QoI values between the thresholds). We choose the thresholds so that 20% of the QoI values of the offline pool points fall between them. We consider two pairs of thresholds: the 0.3 and 0.5 quantiles of the offline pool QoI, and the 0.5 and 0.7 quantiles. Figure 4-14 shows the average misclassification rate from applying this double threshold to the QoI estimate of physics-informed kNN, with randomly or actively sampled training sets; again, we show the lowest misclassification rate among online number of nearest neighbors $k = 1, 2, 4, 8$.

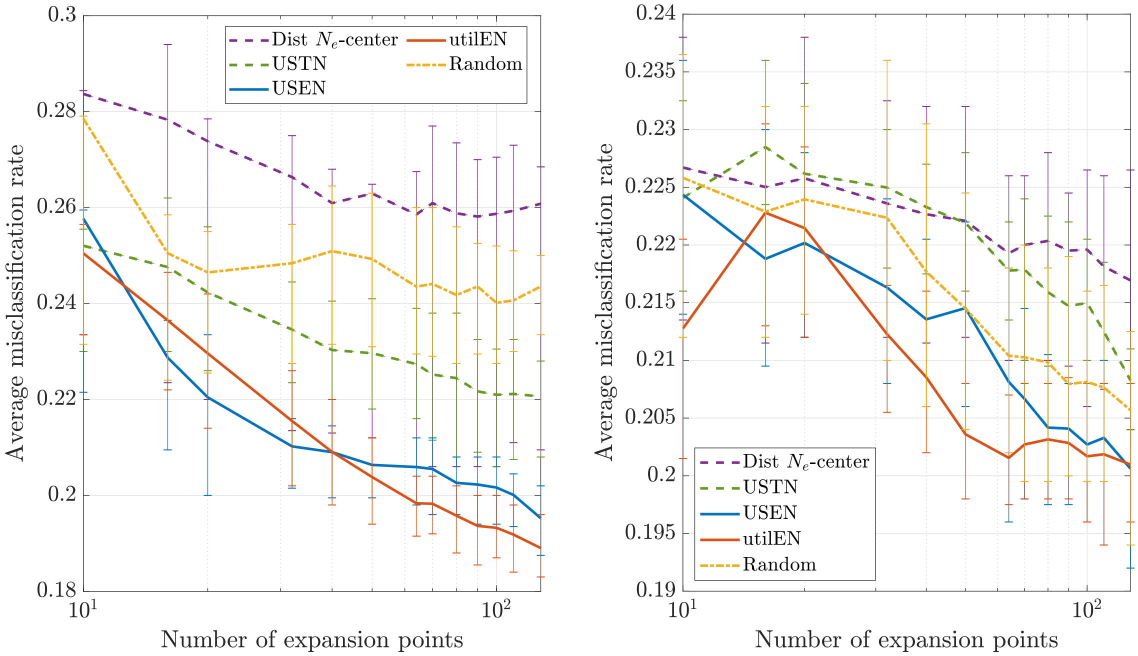
As seen in Figures 4-13 and 4-14, greedily solving the distance-based N_e-center



(a) Bichromatic, target mismatch

(b) Monochromatic, area moment of inertia

Figure 4-13: Average misclassification rate for thresholding QoI when training sets for physics-informed kNN generated through different active learning approaches (binary classification with single threshold). Vertical bars indicate 0.25 and 0.75 quantiles over training sets with different random seeds.



(a) thresholds at 0.3 and 0.5 quantiles

(b) thresholds at 0.5 and 0.7 quantiles

Figure 4-14: Average misclassification rate for double-thresholding QoI from physics-informed kNN, with training sets generated through different active learning approaches, for monochromatic case with area moment of inertia QoI. Vertical bars indicate 0.25 and 0.75 quantiles over training sets with different random seeds.

problem in order to reduce worst-case QoI error (see Section 3.2) is not as efficient as the other approaches in the classification setting. Large QoI errors do not matter so long as the estimate and truth are on the same side of the threshold(s). Note that this active learning approach depends only on the location of the pool points and not their QoI values; the training sets chosen using this approach are the same between Figure 4-11b and Figure 4-14. The QoI errors shown in Figure 4-11b, for both the active and random offline sampling, correspond to the misclassification rates shown in Figure 4-14. In Figure 4-11b the active sampling approach is able to give better average mean QoI error than random sampling at larger training set sizes; however, in Figure 4-14, random sampling gives lower average misclassification rates at all training set sizes.

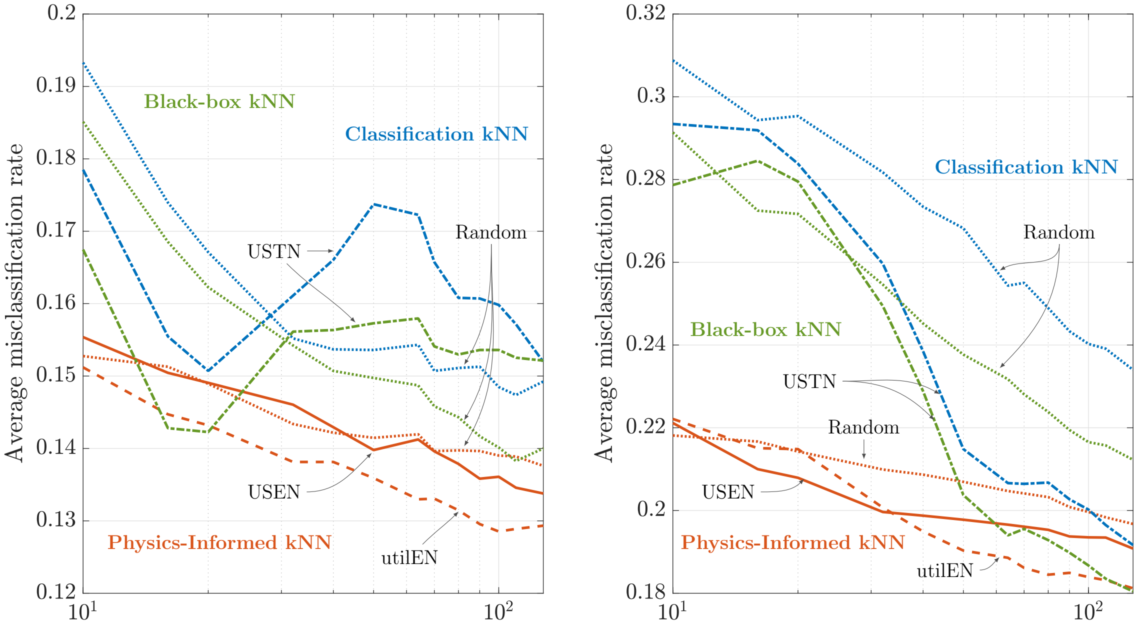
In Figures 4-13 and 4-14, none of the active learning approaches consistently give the lowest misclassification rate for any given training set size. However, random selection does not give the lowest average misclassification rate for any of the training set sizes, suggesting that some form of active learning is advisable. The lowest average misclassification rate is often obtained by one of the active sampling approaches (USEN or utilEN) tailored to physics-informed kNN, although neither consistently performs better than the other. There is no guarantee that either USEN or utilEN will be the best active learning approach for all threshold-based classification inverse problems. The tailored approaches are also more expensive to compute than the other approaches considered, since they require using the physics-informed maps to estimate the classes of the pool points, with utilEN being more expensive than USEN due to the more complex calculation of utility.

4.7.2 Physics-Informed vs Black-Box Active Learning and Online Classifier Combinations

Using the same setup as was used for the results in Figure 4-13, we compare the performance of physics-informed active learning (see Sections 3.3 and 3.4) and online QoI estimation approaches with that of black-box active learning and online QoI estimation approaches. For physics-informed classification, we threshold the QoI estimated using physics-informed kNN, as described in Section 3.1.2. For black-box online classification approaches, we consider two approaches: (a) thresholding the black-box kNN QoI estimate (see (Equation (2.41)), and (b) using classification kNN, where the estimated class of an online point is the most popular class among its k nearest neighbors in the training set. For these black-box online classification approaches, we compare uncertainty sampling (USTN, see Section 3.3) with random sampling.

Figure 4-15 describes the average misclassification rate for different combinations of training set selection and online classification approaches; we show the lowest misclassification rate among $k = 1, 2, 4, 8$. Greedily solving the distance-based N_e-center problem (see Section 3.2) can also be considered a black-box active sampling approach, but gives inferior classification accuracy compared to uncertainty sampling, so we do not plot it. Overall, both physics-informed and black-box classification approaches benefit from active sampling in the offline phase, although no active sampling approach is guaranteed to give better classification for all training set sizes. For this example, one is generally able to achieve the lowest misclassification rate for a given training set size by using a combination of physics-informed active sampling and physics-informed kNN for online QoI estimation. For small randomly selected training sets, classification by thresholding a physics-informed QoI estimate is most accurate. This parallels the behavior seen in Section 4.5.1, though we note that reducing QoI error does not always reduce misclassification rate, since a QoI with a large error can still be on the correct side of the threshold (as discussed in Section 4.7.1).

As in Section 4.5.2, we also explore the performance of different offline sampling and online classification approaches when the online and offline observation distributions do not match. We again consider online test observations generated from objects with two random anomalies, compared to offline training observations generated from objects with just one random anomaly. Figure 4-16 describes the performance of the different offline sampling and online classification approaches when applied to this mismatched test set. We see that overall, both physics-informed and black-box clas-

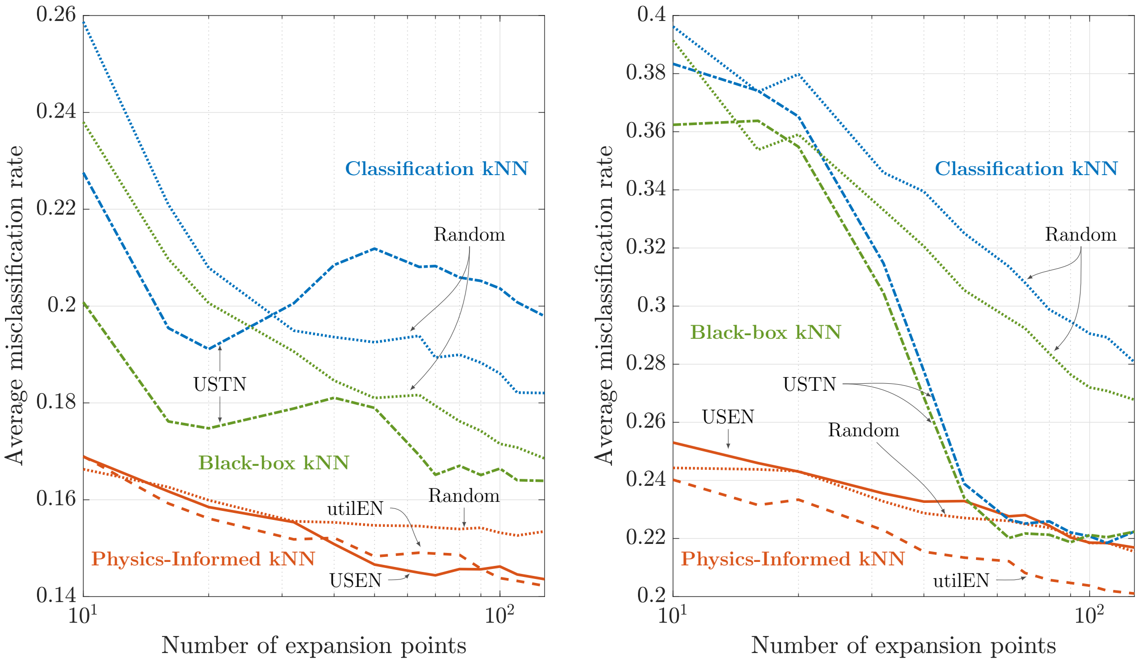


(a) Bichromatic, target mismatch

(b) Monochromatic, area moment of inertia

Figure 4-15: Average misclassification rate for (a) target mismatch and (b) area moment of inertia QoI outputs, by thresholding estimates from physics-informed kNN or black-box kNN, or by classifying using classification kNN.

sification approaches still benefit from boundary-seeking active sampling in the offline phase, even though different portions of the boundary between classes are relevant for the different offline and online test observation distributions. We also see a larger gap in the performance of the best physics-informed and black-box offline-online combinations; this parallels the behavior seen in Section 4.5.2.



(a) Bichromatic, target mismatch

(b) Monochromatic, area moment of inertia

Figure 4-16: Average misclassification rate for (a) target mismatch and (b) area moment of inertia QoI outputs, by thresholding estimates from physics-informed kNN or black-box kNN, or by classifying using classification kNN, when training and test observations are generated from different processes.

Chapter 5

Conclusion

In this concluding chapter, we summarize the work and contributions of this thesis (Section 5.1), and point to directions for future work (Section 5.2).

5.1 Thesis Summary

In this thesis, we sought to address the problem of rapidly estimating a QoI from observations informing the parameters of an expensive model, in a resource-constrained setting where solving the full nonlinear goal-oriented inverse problem is not feasible. To this end, we have developed an algorithm that computes and stores a library of compact, observation-independent, multilinear maps in an offline preparatory phase, and applies these locally accurate maps to online observations to obtain rapid estimates of their QoI. We have developed methods to reduce the storage and application costs of these maps, and an error estimate that can be quickly calculated in the online phase. Our algorithm, which we refer to as 'physics-informed kNN', draws from and shares characteristics with both data-based and physics-based approaches. Like black-box data-based methods, our mappings are cheap to apply (similar application costs to polynomial regression) and are based on a training set of points where the true QoI (from solving the full inverse problem for inferred parameters) for a set of training observations is known. Unlikely purely data-based methods, our mappings also contain information about the structure and physics of the goal-oriented inverse problem. The observation and prediction operators include physical models, and are approximately represented by their truncated expansions when calculating the mappings. Incorporating physics information into the mappings enables rapid QoI estimation, with fewer training points needed than black-box data-based methods to achieve a similar accuracy.

We have developed active learning methods tailored to our physics-informed kNN, enabling the efficient choice of training points from a potential pool. Each training point requires solution of the full inverse problem, and the calculation and storage of its corresponding observation-to-QoI maps, so efficiently choosing training points can reduce errors from estimating the QoI, given a limited offline computational budget and/or map storage budget. We developed active learning methods for both regression and classification. For the regression case, where one seeks to reduce error in the QoI estimate, we formulated the problem of choosing training points to greedily reduce the worst-case QoI estimation error, and made approximations to allow for a feasible solution. For the classification case, where the continuous QoI is mapped to a discrete class, we developed two active learning approaches which seek to reduce the misclassification rate by greedily choosing training points with greatest uncertainty in their classification. In the first, we greedily choose points with greatest uncertainty, where the estimated uncertainty is associated with proximity to the estimated boundary between classes. In the second, we greedily choose points based on a more expensive utility score, where the utility score suggests how querying a point might impact the total uncertainty of the pool of points.

We have demonstrated our physics-informed kNN and its associated active learning algorithms on a nonlinear goal-oriented tomography problem. For this problem, our proposed physics-informed approach achieved small QoI errors in a fraction of the time to solve the full inverse problem. For the same random training sets, the physics-informed mappings achieved greater accuracy in the QoI estimate for small training sets compared to the black-box regression methods tested. The physics-informed estimates were also more robust in the case where the training set did not well represent the test set. Similar behavior was observed when using active learning approaches to choose different training sets for the physics-informed and black-box data-based mappings, and when the QoI estimate was used for classification. When physics-informed kNN was used to estimate the QoI, the proposed active learning approaches were able to achieve reduced maximum QoI errors and misclassification rates compared to randomly choosing training sets. Neither of the two active learning approaches developed for classification with the physics-informed maps was consistently better than the other. However, in the numerical experiments performed, one or both of the active learning approaches gave lower misclassification rates than random sampling, suggesting that some form of active learning is advisable.

The contributions of this thesis are:

- development of a physics-informed kNN algorithm that bypasses the solution of a high-dimensional inverse problem to directly give rapid QoI estimates for observations, using a library of physics-informed maps;
- development of active learning strategies for efficiently choosing offline evaluations with which to generate the library of maps;
- numerical evidence that physics-informed kNN gives improved accuracy and robustness, compared to black-box methods;
- numerical evidence that the proposed active learning approaches can give improved QoI estimation and classification performance over random sampling.

5.2 Directions for Future Work

5.2.1 Large-scale Problems

Additional work may be needed to apply our physics-informed kNN to large-scale and/or highly nonlinear problems. The cost of applying our physics-informed maps does not scale with parameter dimension, in contrast to solving the full inverse problem. Storage and online application costs do, however, increase with observation dimension. Although we assume the observations to be much lower dimensional than the parameters, the observation dimension can still be large for large-scale problems with millions of parameters. High-dimensional observations will further increase the need for reduced-rank representations (see Sections 2.4.2 and 2.4.3). In addition, if one can afford larger training sets, choosing maps based on the Euclidean distance between their training point and online observations may not work well for high-dimensional observation space [16]. Using different norms (e.g., a fractional norm [1]) to measure distance may help. Seeking low-dimensional structure in the relationship between observations and QoI may also help. Global low-dimensional structure may be useful in finding a reduced effective observation space in which to store and choose maps. Local low-dimensional structure may be useful in choosing appropriate maps to apply in the online phase, in a similar fashion to our error estimate (see Sections 2.6 and 4.4.2).

We have also thus far assumed the QoI to be a scalar or very low-dimensional vector. Considering each element as a separate scalar QoI and generating a library of maps for each can become expensive. For vector-valued QoI, rather than consider each element as a separate QoI, one may more effectively choose training points and compute factored reduced-rank map representations by considering the QoI as a whole. To reduce storage and online application costs for vector-valued QoI, one may wish to choose a single training set and use a single library of maps for the QoI as a whole; developing active learning strategies suited to vector-valued QoI would be useful in this regard. Having a single training set may also allow for more efficient factored (and potentially reduced-rank) representations of our physics-informed maps. In addition to the non-symmetric versions of the tensor decompositions discussed in Section 2.4.2, one may also make use of tensor decompositions that treat one dimension differently from the others, such as the T-SVD [76]. Techniques for handling vector-valued QoI would be useful for extending continuous QoI to classification with many classes by defining a plug-in classifier (see Section 3.1.2).

Since our physics-informed maps are based on local approximations to the goal-oriented inverse problem, one may expect our algorithms to encounter difficulties in highly nonlinear cases. In particular, we use a linear approximation of the observation operator; the resulting approximate inverse problem is more analytically tractable than when higher-order approximations to the observation operator are used. The order of the Taylor approximation to the prediction operator can be increased for greater accuracy, at the cost of greater expenses in storage and online application. However, increased order is not guaranteed to give better accuracy, and not all continuous, sufficiently differential functions have convergent Taylor approximations. For highly nonlinear goal-oriented inverse problems, the accuracy of our local physics-informed maps may be further restricted to regions near training points. Thus, highly nonlinear problems have increased need for effective tailored active learning approaches in the offline phase, and methods for choosing appropriate maps in the online phase.

5.2.2 Parameterized Family of Predictions

The work in this thesis was motivated by the need to rapidly estimate the QoI corresponding to observations in resource-constrained settings, such as when the QoI is needed to inform a decision. In deriving our algorithm, we assumed a fixed prediction operator (mapping parameters to QoI) in the online phase that is known in the offline preparatory phase. When the effect of the inferred parameters can be summarized by a low-dimensional quantity that does not depend on the decision variables, then it is fair to assume such a known and fixed prediction operator.

For more complex decisions, this assumption may not hold. For example, consider a contaminant control problem where one wishes to use observations of past concentration levels to decide a remediation strategy, with the aim to control future concentrations in some sensitive area. Suppose one infers for the strengths of sources that can not be controlled, and then seeks the optimal strengths of controllable sources in order to counteract the non-controllable sources. If the concentrations are nonlinearly related to the sources, then the effects of the inferred parameters and the control variables on the future concentration can not be separated. The effect of the inferred parameters can not be encapsulated in a way that is independent of the control variables, which are not known in the offline stage.

Enabling rapid decision-making from observations in these more complex, coupled problems motivates extending our physics-informed mappings to the case where the goal-oriented inverse problem has a family of prediction outputs. An immediate path forward would be to sample prediction operators from this family, and generate a library of physics-informed maps for each. Techniques would need to be developed to efficiently choose both prediction operators and observations for which to generate maps in the offline phase. Potential inspiration can be drawn from parametric model-order reduction (surveyed in [12]), which seeks to build approximations for a parameterized family of models. Another potential source of inspiration is transfer learning (surveyed in [145]) for black-box data-based methods, where one seeks to 'transfer' information from one input-output relationship to another, through the training points and/or the trained data-based mappings.

5.2.3 Robustness to Training-Test Mismatch

In our numerical experiments, we saw that physics-informed kNN was more robust to the case where the training observation points did not well match the online observations (see Sections 4.5.2 and 4.7.2). If one is aware of the mismatch, one can potentially take steps to use the physics-informed maps differently, so as to obtain better QoI estimation accuracy. One may again take inspiration from transfer learning methods, which have also been developed for the case where the true input-output (observation-QoI) relationship remains constant, but there is a mismatch in the distribution of input points in the offline training and online test phases. Such methods can potentially be extended to our physics-informed mappings, to further improve its robustness to a mismatch in the distribution of training and test observations.

Bibliography

[1] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. In J. Van den Bussche and V. Vianu, editors, *Database Theory — ICDT 2001*, pages 420–434, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[2] N. M. Alexander, J. E. Dennis, Jr., R. M. Lewis, and V. Torczon. A trust-region framework for managing the use of approximation models in optimization. *Structural Optimization*, 15(1):16–23, 1998.

[3] N. Alger, U. Villa, T. Bui-Thanh, and O. Ghattas. A data scalable augmented Lagrangian KKT preconditioner for large scale inverse problems. *SIAM Journal on Scientific Computing*, 39(5):A2365–A2393, 2017.

[4] C. A. Andersson and R. Henrion. A general algorithm for obtaining simple structure of core arrays in N-way PCA with application to fluorometric data. *Computational Statistics and Data Analysis*, 31(3):255–278, 1999.

[5] A. Archer. Two O(log*k)-approximation algorithms for the asymmetric k-center problem. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 1–14. Springer, 2001.

[6] B. W. Bader, T. G. Kolda, et al. MATLAB Tensor Toolbox version 2.6. Available online, February 2015.

[7] S. Banach. Über homogene polynome in (l^2). *Studia Math*, 7(1):36–44, 1938.

[8] E. Baseman, N. Debardeleben, S. Blanchard, J. Moore, O. Tkachenko, K. Ferreira, T. Siddiqua, and V. Sridharan. Physics-informed machine learning for DRAM error modeling. In *2018 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pages 1–6, 2018.

[9] O. Bashir, K. Willcox, O. Ghattas, B. van Bloemen Waanders, and J. Hill. Hessian-based model reduction for large-scale systems with initial-condition inputs. *International Journal for Numerical Methods in Engineering*, 73(6):844–868, 2008.

[10] R. Becker and B. Vexler. Mesh refinement and numerical sensitivity analysis for parameter calibration of partial differential equations. *Journal of Computational Physics*, 206:95–110, 2005.

[11] J. A. Belward, I. W. Turner, and M. Ilić. On derivative estimation and the solution of least squares problems. *Journal of Computational and Applied Mathematics*, 222(2):511–523, 2008.

[12] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.

[13] M. Benosman, J. Borggaard, O. San, and B. Kramer. Learning-based robust stabilization for reduced-order models of 2D and 3D Boussinesq equations. *Applied Mathematical Modelling*, 49:162–181, 2017.

[14] M. Benosman, B. Kramer, P. Boufounos, and P. Grover. Learning-based reduced order model stabilization for partial differential equations: Application to the coupled Burgers' equation. In *American Control Conference (ACC)*, pages 1673–1678. IEEE, 2016.

[15] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[16] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In C. Beeri and P. Buneman, editors, *Database Theory — ICDT'99*, pages 217–235, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[17] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part II: The Lagrange-Newton solver and its application to optimal control of steady viscous flows. *SIAM Journal on Scientific Computing*, 27(2):714–739, 2005.

[18] S. Bochkanov. ALGLIB. http://www.alglib.net.

[19] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *COLT '92: Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 1992.

[20] G. E. Box, W. G. Hunter, and J. S. Hunter. *Statistics for Experimenters*. John Wiley & Sons, 1978.

[21] G. E. Box and K. B. Wilson. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 13(1):1–45, 1951.

[22] D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2, 1988.

[23] C. U. Brown, G. Jacob, M. Stoudt, S. Moylan, J. Slotwinski, and A. Donmez. Interlaboratory study for Nickel Alloy 625 made by Laser Powder Bed Fusion to quantify mechanical property variability. *Journal of Materials Engineering and Performance*, 25(8):3390–3397, 2016.

[24] T. Bui-Thanh, O. Ghattas, J. Martin, and G. Stadler. A computational framework for infinite-dimensional Bayesian inverse problems Part I: The linearized case, with application to global seismic inversion. *SIAM Journal on Scientific Computing*, 35(6):A2494–A2523, 2013.

[25] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 30(6):3270–3288, 2008.

[26] A. Buja, T. Hastie, and R. Tibshirani. Linear smoothers and additive models. *The Annals of Statistics*, 17(2):453–510, 1989.

[27] C. Campbell, N. Cristianini, and A. J. Smola. Query learning with large margin classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 111–118, 2000.

[28] K. J. Chang, R. T. Haftka, G. L. Giles, and P.-J. Kao. Sensitivity-based scaling for approximating structural response. *Journal of Aircraft*, 30(2):283–288, 1993.

[29] G. H. Chen and D. Shah. Explaining the success of nearest neighbor methods in prediction. *Foundations and Trends in Machine Learning*, 10(5-6):337–588, 2018.

[30] M.-Y. Cheng, P. Hall, and D. M. Titterington. Optimal design for curve estimation by local linear smoothing. *Bernoulli*, 4(1):3–14, 1998.

[31] J. A. Christen and C. Fox. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical Statistics*, 14(4):795–810, 2005.

[32] J. Chuzhoy, S. Guha, E. Halperin, S. Khanna, G. Kortsarz, R. Krauthgamer, and J. S. Naor. Asymmetric k-center is log*n-hard to approximate. *Journal of the ACM (JACM)*, 52(4):538–551, 2005.

[33] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, May 1994.

[34] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4(1):129–145, 1996.

[35] D. Colton and R. Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*, volume 93. Springer Science and Business Media, 2012.

[36] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1254–1279, 2008.

[37] P. R. Conrad, Y. M. Marzouk, N. S. Pillai, and A. Smith. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *Journal of the American Statistical Association*, 111(516):1591–1607, 2016.

[38] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[39] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 09 1967.

[40] T. Cui, J. Martin, Y. M. Marzouk, A. Solonen, and A. Spantini. Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Problems*, 30(11):114015, 2014.

[41] T. Cui, Y. M. Marzouk, and K. E. Willcox. Data-driven model reduction for the Bayesian solution of inverse problems. *International Journal for Numerical Methods in Engineering*, 102(5):966–990, 2015.

[42] S. Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems*, pages 337–344, 2005.

[43] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[44] V. De Silva and L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.

[45] E. de Sturler, S. Gugercin, M. Kilmer, S. Chaturantabut, C. Beattie, and M. O'Connell. Nonlinear parametric inversion using interpolatory model reduction. *SIAM Journal on Scientific Computing*, 37(3):B495–B517, 2015.

[46] A. Donev, A. Atkinson, and R. Tobias. *Optimum Experimental Designs, with SAS*. Oxford Statistical Science Series. Oxford University Press, 2007.

[47] M. Drohmann and K. Carlberg. The ROMES method for statistical modeling of reduced-order-model error. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):116–145, 2015.

[48] Y. Efendiev, T. Hou, and W. Luo. Preconditioning Markov Chain Monte Carlo simulations using coarse-scale models. *SIAM Journal on Scientific Computing*, 28(2):776–803, 2006.

[49] M. S. Eldred, A. A. Giunta, and S. S. Collis. Second-order corrections for surrogate-based optimization with model hierachies. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, 2004.

[50] T. Feder and D. Greene. Optimal algorithms for approximate clustering. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 434–444. ACM, 1988.

[51] E. Fix and J. Hodges, Jr. Discriminatory analysis - nonparametric discrimination: Consistency properties. Technical Report 21-29-004, USAF School of Aviation Medicine, 1951.

[52] H. Flath, L. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, and O. Ghattas. Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations. *SIAM Journal on Scientific Computing*, 33(1):407–432, 2011.

[53] I. Ford, D. M. Titterington, and C. P. Kitsos. Recent advances in nonlinear experimental design. *Technometrics*, 31(1):49–60, 1989.

[54] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2):133–168, Aug 1997.

[55] S. Friedland and A. Torokhti. Generalized rank-constrained matrix approximations. *SIAM Journal on Matrix Analysis and Applications*, 29(2):656–659, 2007.

[56] A. Fujii, T. Tokunaga, K. Inui, and H. Tanaka. Selective sampling for example-based word sense disambiguation. *Computational Linguistics*, 24(4):573–597, 1998.

[57] R. Gilad-Bachrach, A. Navot, and N. Tishby. Query by committee made real. In *Advances in Neural Information Processing Systems*, pages 443–450, 2006.

[58] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th VLDB Conference*, pages 518–529, 1999.

[59] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[60] L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013.

[61] T. Hattori. *Ultraclean Surface Processing of Silicon Wafers: Secrets of VLSI Manufacturing.* Springer Science & Business Media, 2013.

[62] G. Hautier, C. C. Fischer, A. Jain, T. Mueller, and G. Ceder. Finding nature's missing ternary oxide compounds using machine learning and density functional theory. *Chemistry of Materials*, 22:3762–3767, 06 2010.

[63] R. Henrion. Body diagonalization of core matrices in three-way principal components analysis: Theoretical bounds and simulation. *Journal of Chemometrics*, 7(6):477–494, 1993.

[64] J. S. Hesthaven, G. Rozza, B. Stamm, et al. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations.* Springer, 2016.

[65] C. J. Hillar and L.-H. Lim. Most tensor problems are NP-hard. *Journal of the ACM*, 60(6):Article 45, 2013.

[66] C. Himpe and M. Ohlberger. Data-driven combined state and parameter reduction for inverse problems. *Advances in Computational Mathematics*, 41:1343–1364, 2015.

[67] F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927.

[68] F. L. Hitchcock. Multiple invariants and generalized rank of a p-way matrix or tensor. *Journal of Mathematics and Physics*, 7(1-4):39–79, 1928.

[69] T. Isaac, N. Petra, G. Stadler, and O. Ghattas. Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the Antarctic ice sheet. *Journal of Computational Physics*, 296:348–368, 2015.

[70] Y. Javadi, O. Afzali, M. H. Raeisi, and M. A. Najafabadi. Nondestructive evaluation of welding residual stresses in dissimilar welded pipes. *Journal of Nondestructive Evaluation*, 32(2):177–187, 2013.

[71] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, Dec 1998.

[72] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2318–2331, 2017.

[73] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*, volume 344. John Wiley & Sons, 2009.

[74] A. I. Khuri and S. Mukhopadhyay. Response surface methodology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(2):128–149, 2010.

[75] M. E. Kilmer and E. De Sturler. Recycling subspace information for diffuse optical tomography. *SIAM Journal on Scientific Computing*, 27(6):2140–2166, 2006.

[76] M. E. Kilmer and C. D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011.

[77] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[78] D. G. Krige. A statistical approach to some mine valuations and allied problems at the witwatersrand. Master's thesis, University of Witwatersrand, 1951.

[79] H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

[80] D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings 1994*, pages 148–156. Elsevier, 1994.

[81] G. Li, C. Rosenthal, and H. Rabitz. High dimensional model representations. *The Journal of Physical Chemistry A*, 105(33):7765–7777, 2001.

[82] H. Li, V. V. Garg, and K. Willcox. Model adaptivity for goal-oriented inference using adjoints. *Computer Methods in Applied Mechanics and Engineering*, 331:1–22, 2018.

[83] L. Li, J. C. Snyder, I. M. Pelaschier, J. Huang, U.-N. Niranjan, P. Duncan, M. Rupp, K.-R. Müller, and K. Burke. Understanding machine-learned density functionals. *International Journal of Quantum Chemistry*, 116(11):819–833, 2016.

[84] C. Lieberman, K. Fidkowski, K. Willcox, and B. Bloemen Waanders. Hessian-based model reduction: Large-scale inversion and prediction. *International Journal for Numerical Methods in Fluids*, 71(2):135–150, 2013.

[85] C. Lieberman and K. Willcox. Goal-oriented inference: Approach, linear theory, and application to advection diffusion. *SIAM Review*, 55(3):493–519, 2013.

[86] C. Lieberman and K. Willcox. Nonlinear goal-oriented Bayesian inference: Application to carbon capture and storage. *SIAM Journal on Scientific Computing*, 36(3):B427–B449, 2014.

[87] C. Lieberman, K. Willcox, and O. Ghattas. Parameter and state model reduction for large-scale statistical inverse problems. *SIAM Journal on Scientific Computing*, 32(5):2523–2542, 2010.

[88] M. Lindenbaum, S. Markovitch, and D. Rusakov. Selective sampling for nearest neighbor classifiers. In *AAAI/IAAI*, pages 366–371. Citeseer, 1999.

[89] J. Liu, K. Wang, S. Ma, and J. Huang. Accounting for linkage disequilibrium in genome-wide association studies: A penalized regression method. *Statistics and Its Interface*, 6(1):99, 2013.

[90] L. Mainini and K. Willcox. Surrogate modeling approach to support real-time structural assessment and decision making. *AIAA Journal*, 53(6):1612–1626, 2015.

[91] A. J. Majda and J. Harlim. Physics constrained nonlinear regression models for time series. *Nonlinearity*, 26(1):201, 2012.

[92] J. Martin, L. C. Wilcox, C. Burstedde, and O. Ghattas. A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion. *SIAM Journal on Scientific Computing*, 34(3):A1460–A1487, 2012.

[93] G. Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 12 1963.

[94] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, Dec 1943.

[95] J. Močkus. *Bayesian Approach to Global Optimization: Theory and Applications*. Springer Netherlands, 1989.

[96] J. L. Mueller and S. Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. SIAM, 2012.

[97] M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, 2014.

[98] E. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

[99] T. E. O'Brien and G. M. Funk. A gentle introduction to optimal design for regression models. *The American Statistician*, 57(4):265–267, 2003.

[100] M. Ohlberger and M. Schaefer. A reduced basis method for parameter optimization of multiscale problems. In *Proceedings of ALGORITMY*, volume 2012, pages 1–10, 2012.

[101] D. O'Malley and V. Vesselinov. A combined probabilistic/nonprobabilistic decision analysis for contaminant remediation. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):607–621, 2014.

[102] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[103] R. Panigrahy and S. Vishwanathan. An O(log*n) approximation algorithm for the asymmetric p-center problem. *Journal of Algorithms*, 27(2):259–268, 1998.

[104] A. Pappas, Y. Sarantopoulos, and A. Tonge. Norm attaining polynomials. *Bulletin of the London Mathematical Society*, 39(2):255–264, 2007.

[105] H.-S. Park and C.-H. Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341, 2009.

[106] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018.

[107] M. J. D. Powell. Radial basis functions for multivariable interpolation: A review. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation*, pages 143–167. Clarendon Press, New York, NY, USA, 1987.

[108] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, 2007.

[109] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, 2006. http://www.gaussianprocess.org/gpml/.

[110] P. A. Regalia. Monotonically convergent algorithms for symmetric tensor approximation. *Linear Algebra and its Applications*, 438(2):875–890, 2013.

[111] R. J. Renka. Multivariate interpolation of large sets of scattered data. *ACM Transactions on Mathematical Software*, 14(2):139–148, June 1988.

[112] M. Rewienski and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(2):155–170, 2003.

[113] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Washington DC: Spartan Books, 1962.

[114] W. Roux, N. Stander, and R. T. Haftka. Response surface approximations for structural optimization. *International Journal for Numerical Methods in Engineering*, 42(3):517–534, 1998.

[115] J. Schmidhuber. Deep learning in neural networks: An overview. Technical Report IDSIA-03-14, The Swiss AI Lab IDSIA, 2014.

[116] J. K. Seo and E. J. Woo. *Nonlinear Inverse Problems in Imaging.* John Wiley & Sons, 2013.

[117] B. Settles. Active learning literature survey. Technical Report Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2010.

[118] B. Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[119] M. Shahnas, D. Yuen, and R. Pysklywec. Inverse problems in geodynamics using machine learning algorithms. *Journal of Geophysical Research: Solid Earth*, 123(1):296–310, 2018.

[120] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[121] B. Sharma, R. V. Harikrishnan, S. B. McCrorie, M. C. Conner, M. Salahshoor, A. V. Deshmukh, and M. D. Sangid. Incorporation of physics-based machining models in real-time decision making via metamodels. *Procedia Manufacturing*, 10:1109–1117, 2017. 45th SME North American Manufacturing Research Conference, NAMRC 45, LA, USA.

[122] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, pages 517–524. ACM, 1968.

[123] A. P. Singh, S. Medida, and K. Duraisamy. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA Journal*, pages 2215–2227, 2017.

[124] V. Singh and K. E. Willcox. Methodology for path planning with dynamic data-driven flight capability estimation. *AIAA Journal*, pages 2727–2738, 2017.

[125] S. Singhal, M. Grigas, and S. Sreenivasan. Mechanics-based approach for detection and measurement of particle contamination in proximity nanofabrication processes. *Journal of Micro and Nano-Manufacturing*, 4(3):031004, 2016.

[126] K. Smith. On the standard deviations of adjusted and interpolated values of an observed polynomial function and its constants and the guidance they give towards a proper choice of the distribution of observations. *Biometrika*, 12(1/2):1–85, 1918.

[127] R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox. Projection-based model reduction: Formulations for physics-based machine learning. *Computers & Fluids*, 179:704–717, 2018.

[128] T. Taddei, J. D. Penn, M. Yano, and A. T. Patera. Simulation-based classification; a model-order-reduction approach for structural health monitoring. *Archives of Computational Methods in Engineering*, 25(1):23–45, Jan 2018.

[129] A. N. Tikhonov and V. Y. Arsenin. *Methods for Solving Ill-posed Problems*. John Wiley and Sons, Inc, 1977.

[130] R. Tomioka and T. Suzuki. Spectral norm of random tensors. *ArXiv e-prints*, July 2014. https://arxiv.org/pdf/1407.1870.pdf.

[131] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2(Nov):45–66, 2001.

[132] C. Tosh and S. Dasgupta. Diameter-based active learning. In *International Conference on Machine Learning*, pages 3444–3452, 2017.

[133] L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[134] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen. A new truncation strategy for the higher-order singular value decomposition. *SIAM Journal on Scientific Computing*, 34(2):A1027–A1052, 2012.

[135] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, Berlin, Heidelberg, 1995.

[136] M. Vauhkonen, D. Vadász, P. A. Karjalainen, E. Somersalo, and J. P. Kaipio. Tikhonov regularization and prior information in electrical impedance tomography. *IEEE Transactions on Medical Imaging*, 17(2):285–293, 1998.

[137] N. Vervliet, O. Debals, and L. D. Lathauwer. Tensorlab 3.0 – numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization. In *2016 50th Asilomar Conference on Signals, Systems and Computers*, pages 1733–1738, Nov 2016.

[138] N. Wachter, G. Krischak, M. Mentzel, M. Sarkar, T. Ebinger, L. Kinzl, L. Claes, and P. Augat. Correlation of bone mineral density with strength and microstructural parameters of cortical bone in vitro. *Bone*, 31(1):90–95, 2002.

[139] J.-G. Wang, E. Sung, and W.-Y. Yau. Active learning for solving the incomplete data problem in facial age classification by the furthest nearest-neighbor criterion. *IEEE Transactions on Image Processing*, 20(7):2049–2062, 2011.

[140] L. Wang, X. Hu, B. Yuan, and J. Lu. Active learning via query synthesis and nearest neighbour search. *Neurocomputing*, 147:426–434, 2015.

[141] M. Wang. *Industrial Tomography: Systems and Applications*. Elsevier, 2015.

[142] M. Wang, K. D. Duc, J. Fischer, and Y. S. Song. Operator norm inequalities between tensor unfoldings on the partition lattice. *Linear Algebra and its Applications*, 520:44–66, 2017.

[143] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. *Computer Methods in Applied Mechanics and Engineering*, 237-240:10–26, 2012.

[144] G. S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 359–372, 1964.

[145] K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, 2016.

[146] H. Wendland. *Scattered Data Approximation.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004.

[147] J.-L. Wu, H. Xiao, and E. Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Physical Review Fluids*, 3:074602, Jul 2018.

[148] X. Xie, M. Mohebujjaman, L. G. Rebholz, and T. Iliescu. Data-driven filtered reduced order modeling of fluid flows. *SIAM Journal on Scientific Computing*, 40(3):B834–B857, 2018.

[149] H. Xu, X. Wang, Y. Liao, and C. Zheng. An uncertainty sampling-based active learning approach for support vector machines. In *International Conference on Artificial Intelligence and Computational Intelligence*, volume 3, pages 208–213, 2009.

[150] L. Zhang, C. Chen, J. Bu, D. Cai, X. He, and T. S. Huang. Active learning based on locally linear reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):2026–2038, 2011.

[151] Z. Zhao and W. Yao. Sequential design for nonparametric inference. *Canadian Journal of Statistics*, 40(2):362–377, 2012.

[152] A. G. Zhilinskas. Single-step Bayesian search method for an extremum of functions of a single variable. *Cybernetics*, 11(1):160–166, 1975.