Reduced Order Methods for Imaging and Time Series Analysis

A dissertation submitted by

Ege Ozsar

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

Tufts University

May 2025

Adviser: Eric L. Miller

Reduced Order Methods for Imaging and Time Series Analysis

Abstract

This thesis addresses computational and representational challenges associated with large scale problems involving high dimensional data by developing reduced order methods for two critical fields: imaging inverse problems and multivariate time series analysis. The goal is to build effective reduced order methods in large scale systems by leveraging model based and data-driven machine learning and statistical techniques under limited data and computational resources.

First, we introduce PaLEnTIR, a novel parametric level-set (PaLS) method for piecewise-constant image reconstruction. Our key contribution involves a unique PaLS formulation utilizing a single level-set function to restore scenes containing multi-contrast piecewise constant objects without requiring knowledge of the number of objects or their contrasts. Our model integrates anisotropic basis functions to represent a broader class of shapes. PaLEnTIR streamlines the model by reducing redundancy and indeterminacy in the parameterization, improving numerical performance. PaLEnTIR's performance is evaluated via diverse experiments, including linear and nonlinear inverse problems.

Next, PaLEnTIR framework is extended to track evolving objects in spatiotemporal data. Leveraging a ladder function design and temporal smoothness regularization, the proposed method contributes to field by delivering a single level-set formulation capable of capturing multiple contrast levels evolving over time. Simulated diffusion experiments demonstrate that the method recovers key concentration thresholds even under sparse and noisy sensor conditions.

Finally, an AI-driven probabilistic change point detection framework is presented for high dimensional multi-agent time series data. The proposed framework models the collective behavior as a non-stationary stochastic process, where each time step's multi-agent state is treated as a sample from an underlying probability distribution that evolves over time. This representation captures both spatial configurations and temporal dynamics of the group, and it naturally handles varying numbers of agents by producing a consistent feature dimension equal to the number of bins in histograms. We integrate a cascaded dimensionality reduction pipeline combining principal component analysis and autoencoders to manage the massive dimensionality due to probabilistic modeling. This drastic reduction enables efficient non-parametric density estimation on the embedded space and robust detection of distribution shifts. The method is evaluated through synthetic and realistic multi-agent scenarios.

Acknowledgments

Throughout my academic journey at Tufts University, I have been privileged to meet and work with brilliant people whose guidance and support have shaped my experience. I am grateful to all individuals at Tufts and beyond who have contributed to my both professional and personal growth.

I would first like to thank my adviser, Professor Eric L. Miller. His mentorship has been invaluable throughout this thesis. His belief in my work has inspired and encouraged me to think outside the box. I am grateful to have had the opportunity to learn from him. It has been an extremely rewarding and life-changing experience.

I am also very thankful to Professor Misha Kilmer, whose support and guidance have significantly contributed to this thesis. Since my first day at Tufts University, Professor Misha Kilmer has been a key part of my academic career. This thesis would not have taken its current form without her guidance.

I sincerely thank my committee members, Professor Eric L. Miller, Professor Misha Kilmer, Professor Eric de Sturler, and Professor Shuchin Aeron, for their valuable feedback and for generously agreeing to serve on my thesis committee.

Finally, I would like to express my gratitude to my mother, father, fiancée, and beloved cat Pamuk for their support and love throughout this long period of time. Their belief in me during the most difficult times has been my primary source of inspiration. Their support has allowed me to accomplish this, and this thesis is lovingly dedicated to them.

Contents

List of Figures

List of Tables

List of Algorithms

Reduced Order Methods for Imaging and Time Series Analysis

Chapter 1

Introduction

The exponential increase in data generation and collection across various fields has begun an era where handling and analyzing large scale systems, and high dimensional data have become both a necessity and a challenge. Advancements in sensors have led to collection of complex data in fields such as medical imaging [1], environmental monitoring [2], and finance [3]. For instance, medical imaging modalities like magnetic resonance imaging (MRI) and computed tomography (CT) produce high resolution images that are crucial for diagnosis and treatment [4]. Similarly, the advancements in sensors and the Internet of Things (IoT) have led to the accumulation of complex and high dimensional time series data in various applications ranging from climate science to wearable technology [5].

However, the high volume and complexity of this data pose significant challenges for computational methods. High dimensional datasets; such as crowd sourced GPS trajectories, high resolution 3D medical images etc. often require significant computational resources for storage, processing, and analysis [6]. The associated computational cost in large scale systems originates from several factors [7].

First, many applications rely on forward models, which are computer models designed to simulate system behavior given certain input parameters. In fields like geophysics, for example, a forward simulation might start with subsurface properties (e.g., rock densities)

and produce predictions of seismic wave travel times, enabling researchers to test hypotheses about the Earth's interior [8]. In medical imaging, a forward model may describe how electromagnetic or acoustic waves interact with biological tissues to deliver measurable signals [9]. These forward simulations are key to understand many complex physical processes, guide experiment designs, and evaluate the performance of engineering systems. However, when these simulations are run extensively, whether to tune system parameters, explore multiple scenarios, or to generate synthetic data, they rapidly become computationally expensive.

Second, the high dimensionality of data causes the "curse of dimensionality", that increases computational costs [10]. This phenomenon implies that, with growing dimensions, the computational cost and, in most cases, the size of data required to obtain reliable results grow exponentially [11]. Increased computational complexity impacts critical tasks such as sampling and uncertainty quantification [12].

Lastly, storing and processing high-dimensional data often requires expensive resources, including electricity, cloud and server services. These requirements can be problematic due to limited resources and consequently lead to delays. This could have severe impact in time sensitive applications like emergency medical diagnosis or real-time anomaly detection problems [13].

One way of reducing computational burden is through low order representation of a problem. Such methods consider computationally inexpensive mathematical representations of problems to enable efficient optimization [14], uncertainty quantification [15], and sensitivity analysis [16] across various fields such as engineering design, environmental modeling, finance, and machine learning. These methods result in lower dimensional problem through surrogate models, and reduced-order models. Surrogate models are approximation methods used to emulate the behavior of complex and computationally expensive simulations or processes [17]. They are employed when running detailed simulations is impractical due to high computational costs or time constraints. Reduced-order models (ROM) can be divided into two categories; model-based, and data-driven ROMs. Model-based reduced-order meth-

ods rely on a mathematical or physical understanding of the underlying model [18]. Some ROM techniques such as the Craig-Bampton method in structural mechanics are designed for specific PDE-based models [19]. In linear system analysis, linearization, linear parameter-varying models, and techniques such as balanced truncation and pole-zero simplification are often used to simplify the system model [20]. Another option is through the parameterization of the problem. Parametric reduced order methods aim to reduce the search space via a low-order representation of the problem through parameterization. For instance, parameterizing the level set function in level set methods result in a significantly lower dimensional problem, and consequently paves the way for easier use of Newton and quasi-Newton methods [21]. Data-driven ROMs use input/output data from the original high-fidelity first-principles model to construct either a dynamic or static reduced order model that accurately represents the underlying system [22]. Traditionally, Principal Component Analysis (PCA) and factor analysis were performed for dimensionality reduction [23]. However, these methods are only able to represent linear dependencies among the data. Alternative to linear methods, a large number of nonlinear techniques for dimensionality reduction have been proposed [24, 25]. Recently, many machine learning based approaches have emerged as promising tools for reduced order modeling, such as autoencoder and LSTM networks [26].

This thesis focuses on two fundamental and computationally intensive domains, imaging science and time series analysis, and applies the previously discussed frameworks of reduced-order modeling and dimensionality reduction to address their inherent challenges. Imaging science is an interdisciplinary field dedicated to the study and use of technologies for applications in imaging. This ever-evolving branch of study incorporates knowledge and techniques in physics, mathematics, electrical engineering, computer science, and computer vision. By combining these different disciplines, imaging science supports the development of new imaging systems and enhances our ability to analyze and apply visual information in a broad set of applications. This thesis mainly focuses on inverse problems, where, as discussed in greater detail below, the goal is the recovery of an image of objects or scenes of

interest from various forms of measured sensor data. Our work employs parametric level-set methods to achieve a low-order representation of the reconstruction problem. Parametrization of the level-set function reduces the image recovery to a problem defined in terms of a considerably lower number of parameters.

Similarly, time series analysis is a fundamental problem that involves examining sequences of data points collected over time to identify underlying patterns, trends, and anomalies. With the increasing availability of high-frequency and high-dimensional time series data in fields like finance [27], healthcare [28], security [29] and environmental monitoring [30], efficient processing and analyzing the time series data becomes more and more important every day. This thesis mainly focuses on time series analysis of multi-agent simulations where agents interact each other. Examples to such simulations include crowd anomaly detection [31], prey-predator relations [32], collective behavior of agents and groups of animals [33] etc. This thesis leverages both linear and nonlinear data-driven reduced-order techniques. Using Principal Component Analysis (PCA) for linear reduction and Autoencoder architectures for nonlinear reduction, complex multi-agent time series data are projected onto a far smaller, more tractable space. This reduced representation facilitates reliable and efficient density estimation, addressing the previously discussed dimensionality problems and computational costs [11].

1.1 Inverse Problems and Imaging

Inverse problems are fundamental in science and engineering problems. An inverse problem's primary goal is to determine a medium's unknown composition and structure from a set of indirect measurements related to the desired unknowns through a physical model. Reduced order methods have shown great promise in particularly solving inverse imaging problems, such as MRI, CT, and ultrasound [34], where the size of the problem can be a significant concern, especially if high-fidelity high dimensional reconstructions are needed.

A key challenge in many inverse imaging problems is their inherent ill-posedness, which causes the solutions to become extremely sensitive to noise in the data or discrepancies in the sensor's physical model [35]. This problem often arises from the basic physics of the sensing systems, such as in applications like electrical impedance tomography [36], diffuse optical tomography [37], and inverse scattering [38]. Consequently, the accuracy of the reconstructions is affected negatively by this phenomenon. Another reason that leads to ill-posedness is the poor positioning of sensors, leading to limited data that does not contain enough observation to capture fine details across the region of interest. Many practical inverse problems encounter both issues, resulting in reconstructions containing artifacts characterized by high frequency and large amplitude if ill-posedness is not addressed correctly.

To overcome ill-posedness, regularization techniques are typically implemented [39]. These methods use prior knowledge regarding the unknowns and add constraints to the inverse problem to regularize the solution. In a variational framework, regularization is often done by including terms in the cost function that act as penalties for undesirable characteristics of the reconstruction. Tikhonov regularization [40], which favors smoother solutions, and total variation (TV) regularization [41], which encourages piecewise-constant regions, are two examples of such regularization. While these pixel-based approaches have been practical, they struggle as image data becomes higher in resolution and dimension. The large number of unknowns in high-resolution problems can make pixel-based methods computationally expensive and ill-posed. Under these circumstances, reduced order methods become increasingly crucial as low order representation of a problem can result in a better-posed problem and can reduce computational cost.

1.1.1 Shape Based Inverse Problems

In the pursuit of extracting the unknown composition and structure of a medium from indirect observations governed by physical models, researchers often focus on characterizing "regions of interest" (ROIs). Examples of such ROIs include cancerous tumors in diffuse

optical data [42], subsurface contaminants in hydrological data [43], or buried objects in electromagnetic data [44]. These problems typically involve a computationally intensive image formation step followed by post-processing it to identify the ROIs [45]. When data are limited, the initial image formation stage will require potentially complex regularization methods to overcome ill-posedness. While this two-step approach can be effective, it is computationally intensive. Moreover, when data are limited, the initial image reconstruction stage may require complex regularization techniques to mitigate the ill-posed nature of the problem. Alternatively, direct estimation of ROI geometry and contrasts from data offers a more efficient approach. This approach is known as inverse obstacle or shape-based problems [46, 47]. This methodology typically uses a straightforward reduced-order shape parametrization. Compared to methods that rely on pixel-based imaging, this geometric approach generally leads to better-posed problems due to more accurate obstacle representation. Additionally, regularization in this context can either be implicitly integrated through the shape parametrization or explicitly through geometric constraints on the shape. However, this form of shape representation lacks topological flexibility and requires prior knowledge of the number of shape components. It also struggles with topographical features like holes and regions with high curvature, such as corners. These challenges have driven the development of shape-based inverse methods using level set-type representations because of their ability to naturally recover objects whose topology (i.e., the number of connected components) is not known *a priori* [48].

1.1.2 Level Set Methods

Originally introduced by Osher and Sethian [49], the level set method is a general framework for the computation of evolving interfaces using implicit representations. The implicit representation of the geometry introduces the possibility to handle topological changes such as splitting and merging of connected components in an automatic way, and allows to construct efficient and accurate numerical methods. Santosa [48] was among the first to ap-

ply these models to inverse problems, prompting significant subsequent research utilizing classical level set evolution methods, particularly in image processing applications [50]. In ill-posed inverse problems, regularization of the level set function is required to obtain stable solutions. Various strategies have been employed to address this challenge, including conventional pixel-based regularization methods [51], incorporation of geometric constraints [52], or representing the level set function in a finite-dimensional basis, as in the parametric level set (PaLS) approach first introduced for inverse problems by Aghasi *et al.* [21]. This latter method forms the foundation of the work presented in Chapter 3 of this thesis.

The PaLS model has proven capable of capturing the topological advantages of a level-set methods while avoiding difficulties such as explicit regularization and reinitialization that occur frequently when using traditional level-set methods for inverse problems. Moreover, it was shown in [21] that the low order representation of the inverse problem makes it possible to use Newton and quasi-Newton methods for determining the PaLS parameters and potentially reducing down the size of the problem by reduced-order parameterization. Recently, the PaLS model in [21] and variants have been used across a range of application areas and imaging modalities including geophysics [53], seismology [54] and reservoir monitoring [55], image segmentation [56], tomography problems [57, 58, 59], electromagnetic imaging [60], and multi-modal imaging [61].

Despite the advantages of PaLS, room for improvement remains. Existing PaLS models and most other level-set methods struggle to recover distributions of objects with multiple contrast values. Approaches like colour [62], vector [63], and binary level-sets [64] rely on a number of level-sets proportional to the logarithm of the number of contrast levels [56], leading to scalability challenges as the object count increases. Furthermore, the choice of basis functions for PaLS raises limitations. Traditionally, a weighted superposition of predetermined basis functions, predominantly radial basis functions (RBFs) [65, 66, 67], has been employed. RBFs possess circular cross-sections and thus can limit the range of objects that can be represented using a given number of RBFs. Moreover, despite their

efficacy, many prevailing PaLS models grapple with numerical conditioning issues. Models employing RBFs can encounter non-uniqueness in parameter representation due to infinitely many parameter pairs yielding the same circular cross-sections. As we illustrate empirically in this thesis, this can yield ill-conditioned, even singular, Jacobian matrices — detrimental for quasi-Newton and even for trust-region methods, the performance of which degrades under such conditions [21].

1.2 Time Series Analysis

Among the various types of data, time series is one of the types that has been studied for a long time in academia and in practice. Time series analysis includes a wide range of methods for examining data points collected at successive points in time, including approaches that leverage statistical and probabilistic techniques, one significant class of methods in this field.

Time series analysis aims to build models that capture the underlying mechanisms generating the observations [68]. These models ideally learn the temporal and other dependencies in data and are used in critical problems, including detecting change points and predicting future outcomes based on the previous observations [69]. Different sectors including economics, finance, environmental science, healthcare engineering and control systems make use of time series analysis. Economic time series analysis predicts key indicators including GDP growth rates alongside unemployment patterns and inflation trends [70].

Financial modeling requires accurate prediction techniques for stock prices, and exchange or interest rates [71]. Precise predictions in these domains support strategic planning, investment decisions, and risk management. Researchers can model climate patterns through time series analysis while predicting weather conditions and tracking environmental changes [72]. Studying long-term temperature and precipitation data uncovers patterns of climate change. Healthcare professionals analyze time series data from patient monitoring systems, such as blood pressure and heart rate measurements, to identify anomalies and forecast potential

health problems [73]. Time series analysis is also used in epidemiology to model the spread of diseases over time [74]. Engineers use time series analysis for system identification, control, and fault detection in various systems, including manufacturing processes and robotics. Monitoring time series data from sensors allows for predictive maintenance and optimization of system performance [75].

The developments in techniques for collecting high-frequency data such as sensors, or IoT devices, and rapid growth of digital sources of information have led to the collection and generation of vast amounts of time series data. Efficiently processing and analyzing this data is crucial for most modern-day problems. However, analyzing time series data poses significant challenges. The study of high-dimensional and high-frequency time series data presents multiple difficulties [76]. The main challenges encountered when working with these datasets include computational complexity and the curse of dimensionality. Analyzing large datasets, especially at high temporal resolution, demands substantial computational resources. Furthermore, the complexity of data analysis grows significantly when multiple variables and multi-agent interactions are involved [3]. Another noteworthy challenge associated with time series data is non-stationarity, which refers to changes in the statistical properties of the data over time [77]. Moreover, dynamically sized problems such as crowd behavioral problems where varying numbers of agents enter and leave the scene [78]. Such problems can produce inconsistent data structures that require special processing techniques that can handle dynamic-sized inputs over time. Finally, time series problems often contain missing data and are also affected by measurement errors or noise, which can hide underlying patterns [79].

Addressing these challenges requires advanced techniques that can efficiently process large-scale data while capturing essential temporal dynamics. Several statistical methods are employed in time series analysis. Some of the commonly used techniques include: Autoregressive Integrated Moving Average Models, Exponential Smoothing Models, and State Space Models. Alternatively, machine learning methods such as recurrent neural networks

(RNNs) and long short-term memory (LSTM) networks have been applied to time series forecasting due to their ability to model complex nonlinear dependencies [80]. However, recurrent models use sequential processing, meaning that they process one token at a time from first to last. As a result, they cannot operate in parallel over all tokens in a sequence. Recently, transformers gained significant attraction for time series analysis [81]. Transformers have the advantage of having no recurrent units and overcoming one token at a time limitation, therefore enabling parallel operation and requiring less training time than earlier recurrent neural architectures, RNNs an LSTMs. Selecting the appropriate model depends on the characteristics of the data and the specific objectives of the analysis.

1.2.1 Dimensionality Reduction

High dimensionality stands as one of the primary obstacles in time series data analysis [76]. Real-world datasets, including speech signals, digital images, and fMRI scans, typically exhibit complex high-dimensional structures [82]. Effective management of such data often depends on diminishing its dimensional complexity. Dimensionality reduction processes high-dimensional data and transforms it into a more straightforward representation of fewer dimensions while preserving essential data features. The reduced representation ideally needs to align with the intrinsic dimensionality, which represents the smallest number of parameters that describe data features [83]. Dimensionality reduction plays an essential role across multiple domains since it addresses issues in high-dimensional spaces. Through this procedure we gain assistance in tasks such as classification of data as well as its visualization and compression. Dimensionality reduction has been traditionally accomplished through linear techniques, including Principal Components Analysis (PCA) and factor analysis [23]. These traditional methods fail to detect nonlinear patterns within the dataset. Researchers have introduced various nonlinear dimensionality reduction methods to overcome this shortcoming [24] since the complexity of most real-world data requires nonlinear methods.

1.2.2 Change Point Detection

As we stated previously, time series data are sequences of observations over time describing the behavior of systems. While systems may gradually drift in behavior over long periods due to slow, continuous changes; the class of problems considered in this thesis focuses on sudden shifts in system dynamics [84]. These abrupt changes can occur when external events disrupt the equilibrium or when the internal parameters of the system transition sharply from one regime to another. Change point detection (CPD) refers to the task of identifying these points in time where an underlying property of the time series—such as mean, variance, or underlying model parameters—undergoes a sudden and persistent alteration. Detecting these abrupt transitions is critical in a variety of real-world applications. In medical monitoring, CPD is used to analyze trends in physiological variables like heart rate and EEG to facilitate automated, real-time health monitoring for conditions such as epilepsy and sleep disorders [85, 86]. It plays a crucial role in climate science, helping detect shifts in climate patterns [87]. CPD also covers multi-agent problems such as prey/predator relations, collective crowd and animal behaviors [88, 89].

Despite the availability of a wide array of CPD algorithms, many of the existing methods suffer from notable limitations that could impact their performance. First, many standard approaches rely on the assumption of data independence among dimensions to solve simpler problems with smaller dimensions [90]. In reality, time series data typically exhibit temporal correlations or other dependencies (e.g., collective behavior and inter-agent dependencies in multi-agent problems), which, if ignored, may lead to inaccurate change point estimates. Second, in high dimensional datasets, the computational complexity can grow exponentially as the dimensionality increases, leading to the curse of dimensionality [11]. Traditional CPD models, designed for moderately sized data dimensions, may become impractical for handling numerous variables simultaneously [84]. Third, modern multi-agent systems, ranging from fleets of autonomous vehicles to networks of sensors, produce dynamic data streams where

the number of agents (and hence the dimensionality of data) can change over time [91]. Most CPD techniques assume a fixed data dimension and thus struggle with missing or variable-sized data [92].

1.3 Thesis Contributions

This thesis contributes to the field of imaging and time series analysis. In particular, it contributes reduced order methods that consider high dimensional complex imaging and time series problems, including multi-contrast imaging, spatiotemporal monitoring of evolving objects, and change detection in multi-agent time series.

In Chapter 3, we employ an enhanced image reconstruction model for inverse problems in imaging. For inverse problems where one is concerned with recovering the shape and contrast of an unknown number of objects embedded in a medium, parametric level set (PaLS) methods provide much of the flexibility of traditional level set methods while avoiding many of the difficulties such as regularization, re-initialization. In this thesis, we consider the restoration and reconstruction of piecewise constant objects in two and three dimensions using a significantly enhanced PaLS model relative to the current state-of-the-art. We develop Parametric Level-sets Enhanced to Improve Reconstruction (PaLEnTIR) for piecewise constant image reconstruction to address the issues related to PaLS methods.

In Chapter 4, we extend the PaLEnTIR framework developed in Chapter 3 to track objects evolving over time. The proposed approach addresses significant practical challenges encountered in environmental monitoring. The primary motivation is the difficulty and expense associated with embedding sufficient number of sensor networks in large, inhospitable environments, which typically results in sparse, noisy measurements. Our model specifically targets scenarios involving the spatial and temporal monitoring of critical concentration levels of substances, such as nutrients or contaminants, evolving through diffusion processes.

Chapter 5 focuses on detecting abrupt shifts in the dynamics of multivariate systems

concerning multi-agent systems. These systems possess complex dynamics, including temporal dependencies, collective behavior, and inconsistent input data structure due to variable number of agents, which often pose difficulties to statistical and machine learning techniques, including change point detection techniques. Many CPD techniques rely on the assumption that the data observations are independently and identically distributed (iid). However, this assumption rarely holds in realistic, high-dimensional systems where observations come with strong temporal dependencies and non-stationary behavior. Another commonly made assumption is a fixed size input data. This assumption does not apply to multi-agent systems, since agents can enter/exit the system frequently and causing inconsistency in succesive observations. Lifting these assumptions significantly complicates the processing and modeling of the problem. These challenges are addressed by using the principles of probabilistic methods and modern AI-based tools.

1.3.1 Parametric Level-sets Enhanced To Improve Reconstruction

PaLEnTIR significantly advances previous parametric level set methods by coping with several key challenges:

Multi-Contrast Representation

We introduce a smooth transition function to replace the binary Heaviside function traditionally used in PaLS models. This replacement allows our model to represent multiple objects with different contrasts using a single level-set function. To our knowledge, this capability is unique to our proposed method. A major advantage of this approach is the stable number of parameters regardless of the number of contrasts. The efficacy of this multi-contrast feature is contingent upon its capacity to specify space-varying bounds on the contrasts at relatively coarse scales. The new model achieves this by parametrically characterizing the spatially varying contrast limits on a sparse grid of points.

Enhanced Shape Expressiveness

To increase geometric flexibility, we replace radial basis functions (RBF) with anisotropic basis functions (ABF) which are capable of forming rotated ellipsoidal cross-sections. This choice provides enhanced shape expressiveness and greater geometric flexibility. This is particularly important in challenging scenarios like representing long, flat or sharp objects where RBFs have difficulties while representing. The ABF-based model captures more detail while employing fewer number of basis functions.

Improved Numerical Performance

The streamlined parameterization of PaLEnTIR addresses issues related to non-uniqueness which is commonly seen in PaLS-like models. Empirical analysis reveals that the new parameterization and enhancements in the proposed PaLS approach enhance numerical stability and performance. PaLEnTIR reduces redundancy and indeterminacy in the parameterization, and results in better-conditioned Jacobian matrices. This contributes to faster and more reliable convergence during the optimization process for the estimation of the PaLS parameters.

1.3.2 Parametric Level-Set Methods for Tracking Evolving Objects

In Chapter 4, we introduce a PaLS approach for tracking evolving objects over time. The main contributions of this work are as follows:

Multi-concentration Representation

We develop a novel parametric level-set formulation for tracking time-evolving piecewise constant concentration levels. This formulation utilizes a *ladder function*, replacing the binary Heaviside function. The ladder function is defined by the summation of shifted

binary Heaviside functions to efficiently represent multiple concentration levels using only a single level-set function, and thus the proposed method significantly reduces the complexity of the multi-contrast reconstruction problem.

Temporal Smoothness Regularization

To promote smooth transitions of concentration levels in reconstructions of successive time points, we introduce a temporal smoothness regularization term into the PaLEnTIR framework. This modification results in enhanced temporal coherence of reconstructions, and significantly improving the stability and accuracy of recovered sequences from noisy and sparse sensor data.

1.3.3 AI-Driven Probabilistic Change Point Detection in High Dimensional Time Series Data

In Chapter 5, we employ an AI-driven probabilistic change point detection method for time series analysis. The proposed approach targets detecting abrupt shifts in the dynamics of multivariate systems concerning multi-agent systems. The major contributions presented in Chapter 5 are as follows:

Probabilistic modeling

The proposed approach models a reduced order representation (through dimensionality reduction techniques) of the data as a non-stationary stochastic process, where a change point marks a shift in the underlying generative distribution. Instead of analyzing individual time steps, the method employs a sliding window strategy, capturing short-range dependencies in time series data. In the context of crowd behavior, the model treats the collective agent states at each time as samples from a population distribution and represents them using histograms over a discretized state space. By tracking sequences of these histograms within

sliding windows, the framework aims to capture inter-agent and temporal dependencies by probabilistic modeling of time series data.

Cascade dimensionality reduction:

In the proposed CPD approach, a probabilistic approach is considered to model the crowd behavior as a stochastic process discretized over time. The resultant problem becomes huge in dimensionality. The proposed approach integrates linear and nonlinear dimensionality reduction methods (PCA and autoencoders) to simplify the probabilistic characterization of the underlying stochastic process of the crowd behavior, reducing the dimension of the data up to a factor of 3750. This dimensionality reduction step significantly reduces computational complexity and enables the application of efficient and effective density estimation and change point detection.

Fixed-sized representation of multi-agent data

Active agents can enter or exit the media, or they can disconnect over time. One of the challenges of dealing with time series problems with multiple active agents is to deal with varying-sized, or missing input data. The proposed approach addresses both varying-sized and missing input data by histogram-based representation of the agent data. The proposed model maps the raw agent data into fixed-sized histograms, ensuring consistent dimensionality irrespective of agent entry, exit, or missing observations.

1.3.4 Thesis Publications

The publications related to the work presented in this thesis are as follows,

- Ege Ozsar, Misha Kilmer, Eric de Sturler, Arvind K Saibaba, and Eric Miller. "Parametric level-sets enhanced to improve reconstruction (PaLEnTIR)". in: *Inverse Problems* 41.2 (Jan. 2025), p. 025004. DOI: 10.1088/1361-6420/ada798. URL: https://dx.doi.org/10.1088/1361-6420/ada798

1.4 Project Background

This research has received support from the **National Science Foundation** under NSF Grant 1720291, NSF Grant 1935555, NSF Grant 1934553, NSF Grant 1720398, NSF Grant 1720305.

The material in Chapter 5 is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 22STESE00001 01 01. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

Chapter 2

Background

In this chapter we will review forward and inverse problems, parametric level set methods, optimization algorithms, dimensionality reduction, and density estimation.

2.1 Forward and Inverse Problems

Consider a region of space to be imaged, $\Omega \subset \mathbb{R}^d$. For $\mathbf{r}$ a point in Ω, let us define a spatially-dependent property $f(\mathbf{r})$ of the medium (e.g., electrical conductivity, optical absorption, sound speed, etc.). We denote with $\mathcal{M}$ the map which takes $f(\mathbf{r})$ to a vector of noise free data. Typically, $\mathcal{M}$ encompasses the physics of the sensing modality and the engineering details of the associated sensors. The data available for processing is equal to $\mathcal{M}(f)$ corrupted by noise. Unless specified otherwise, we assume that the noise is additive Gaussian. In summary, the forward model can be written

$$\mathbf{d} = \mathcal{M}(f) + \mathbf{w}, \tag{2.1}$$

where $\mathbf{w}, \mathbf{d} \in \mathbb{R}^{N_{pts}}$ represent the additive noise and the data available for processing, respectively.

The inverse problem requires determination of the unknown function f from $\mathbf{d}$. Following

the penalized likelihood approach, (2.1) leads to the following minimization problem as the basis for recovering $f(\mathbf{r})$,

$$\min_f \frac{1}{2}\|\mathcal{M}(f) - \mathbf{d}\|_2^2 + \xi(f). \tag{2.2}$$

The first term in (2.2) arises from a Gaussian assumption on the noise and quantifies the mismatch error between model prediction and the data, while $\xi(f)$ is the regularization functional which is usually used when the problem is ill-posed and is chosen based on prior knowledge concerning, e.g., the degree of smoothness associated with f [94]. It is also possible to regularize the problem using a geometric parameterization of the unknown property. This is done by either embedding the regularization implicitly in the parameterization of the unknown property, in which case no explicit ξ may be needed, or by expressing it as geometric constraints on the shape of the unknown [38]. In this work, we follow the former approach using a parametric level-set type of model. Next section, we will define and review parametric level-set (PaLS) methods.

2.2 Parametric Level-set methods

Under a PaLS model, $f : \mathbb{R}^d \to \mathbb{R}$ consists of object, O, and background, $\Omega \backslash O$, terms and is written as

$$f(\mathbf{r}) = f_O(\mathbf{r})\chi_O(\mathbf{r}) + f_B(\mathbf{r})(1 - \chi_O(\mathbf{r})), \tag{2.3}$$

where $f_O(\mathbf{r})$ and $f_B(\mathbf{r})$ are the generally spatially dependent property values of the object and background, respectively, and $\chi_O(\mathbf{r}) = 1$ for $\mathbf{r} \in O$ and 0 elsewhere is the characteristic function of the region O. Under this model, we seek O (or the boundary, ∂O), $f_O(\mathbf{r})$, and $f_B(\mathbf{r})$. As O can be multiply connected with each component having no specific (i.e., easily parameterized) shape, level-set methods have proven convenient [21]. Mathematically, the

level-set representation of O satisfies

$$\phi_O(\mathbf{r}) \begin{cases} > c & \mathbf{r} \in O, \\ = c & \mathbf{r} \in \partial O, \\ < c & \mathbf{r} \in \Omega \backslash O, \end{cases} \tag{2.4}$$

where c is a constant determining the level-set[1] [21]. In terms of ϕ_O, $\chi_O(\mathbf{r}) = H(\phi_O(\mathbf{r}) - c)$ where $H(x)$ is the Heaviside function. Thus, (2.3) becomes

$$f(\mathbf{r}) = f_O H(\phi_O(\mathbf{r}) - c) + f_B(1 - H(\phi_O(\mathbf{r}) - c)) \tag{2.5}$$

where f_O and f_B are, for now, taken to be constants.

Many level-set methods follow a finite difference discretization of the level-set function, which requires a dense collection of nodes. The difficulty of implementing this approach as well as the numerical considerations of its discrete computation overshadow the advantages of the level-set function, especially in the case of ill-posed inverse problems [21, 95]. Alternatively, a parametric form for the level-set function expands $\phi_O(\mathbf{r})$ as a linear superposition of a set of basis functions (e.g., polynomial, radial basis function, trigonometric, etc). Specifically, the original PaLS model takes the form

$$\phi_{\text{rbf}}(\mathbf{r}; \mathbf{p}) = \sum_{j=1}^{N} \alpha_j \psi(\beta_j(\mathbf{r} - \boldsymbol{\chi}_j)), \;\; \mathbf{p} := \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \\ \boldsymbol{\chi} \end{bmatrix}. \tag{2.6}$$

In this case, the PaLS function is formed as a weighted summation of N basis functions $\psi_j(\mathbf{r}) := \psi(\beta_j(\mathbf{r} - \boldsymbol{\chi}_j))$ for $j = 1, 2, \ldots, N$. The basis functions $\psi_j(\mathbf{r})$ are often taken to be RBFs [96, 97, 98, 58]. We will refer to use of such basis functions as "RBF PaLS", and it

[1] This could be taken to be the zero level-set, though in [21], the case was made for c slightly larger.

is against such representations that we compare our new PaLEnTIR representation.[2] Each basis function $\psi_j(\mathbf{r})$ is associated with its own dilation coefficient β_j, and center location $\boldsymbol{\chi}_j$. Ignoring the contrasts f_B and f_O for a moment, $\mathbf{p}$ contains the geometric unknowns to be determined, including $\boldsymbol{\alpha}, \boldsymbol{\beta}$, and $\boldsymbol{\chi}$, containing the expansion coefficients α_j, the dilation values β_j, and PaLS centers $\boldsymbol{\chi}_j \in \mathbb{R}^d$, respectively. In general, the length of this vector is $(d+2)N$, which results in length $4N$ (for 2D problems) or $5N$ (for 3D).

Combining (2.6) with (2.5), we obtain $f = f(\mathbf{r}; \mathbf{p})$. The goal of a PaLS-based inverse problem is to recover the unknown $\mathbf{p}$ based on the observed data $\mathbf{d}$ and the model $\mathcal{M}$. In practice, this requires replacing the exact Heaviside function with a differentiable approximation (we discuss this more in section 3.1). If the contrast coefficients f_O and f_B are known, the resultant inverse problem formulation using PaLS recovers f as $f(\mathbf{r}; \hat{\mathbf{p}})$ where $\mathbf{p}$ is a minimizer of a nonlinear least squares problem with the objective function

$$\mathrm{F}(\mathbf{p}) = \frac{1}{2}\|\mathcal{M}(f(\mathbf{r}; \mathbf{p})) - \mathbf{d}\|_2^2. \tag{2.7}$$

When the contrasts are not known, a cyclic descent method is often employed [100], in which one alternately estimates $\mathbf{p}$ for the current estimates of f_O and f_B and then updates the contrasts using the just-computed PaLS parameters. For linear inverse problems, computing the contrasts is a linear least squares problem which can be solved efficiently, and the bulk of the computational effort is focused on estimating $\mathbf{p}$.

Since the number of PaLS basis functions is always much smaller than the size of the grid obtained from discretizing Ω, the PaLS parameterization acts as a form of regularization, so we can drop the regularization term $\xi(f)$ from the objective function.

[2] In [21], the norm in the argument is replaced by a pseudo-norm and $\psi(\cdot)$ is taken to be a compactly supported radial basis function. The compactness of the basis functions may be advantageous in terms of yielding a sparse Jacobian [21]. Here, we will use Gaussian RBFs instead of CSRBFs and forgo compactness in favor of retaining a norm, rather than a pseudo-norm. This choice facilitates the comparative analysis of the expressiveness and numerical properties of the proposed approach over RBF PaLS. The parameterization developed here and the analysis, can be utilized in a CSRBF framework. We acknowledge that the Gaussian RBFs do not guarantee the same level of sparsity as seen in [21]. However, we use the TREGS algorithm [99] as our optimizer, and since TREGS involves performing an SVD on the Jacobian, it does not take advantage of any potential sparsity.

2.3 Optimization Algorithms

2.3.1 TREGS (Trust Region Algorithm with Regularized Model Solution)

In this work, we use Trust region algorithm with REGularized model Solution, `TREGS`, in [99] to minimize (2.7) and solve for the PaLS parameters. The `TREGS` algorithm combines the Basic Trust Region Algorithm from [101] with solution update algorithm, Algorithm SU in [99] to construct a globally convergent optimization algorithm for solving (2.7). `TREGS` is designed to address nonlinear least squares problems encountered in parameterized imaging scenarios. By analyzing which spectral components of the Gauss–Newton direction should be discarded or damped, `TREGS` aims to minimize the total number of function and Jacobian evaluations. Additionally, following [14], the algorithm reduces the number of Jacobian evaluations and reduced SVD computations by doubling the trust region and trying a larger step from the current solution iterate after a *very successful* step ($\rho \geq \eta_2$; as shown in the algorithm below.). The pseudocode for the `TREGS` algorithm is given in Algorithm 1. The algorithm requires the initial approximate solution p_c, the initial residual r_c, the initial objective F_c, initial Jacobian and a starting δ. $newSVD$ is a Boolean variable that determines whether the SVD of the Jacobian should be calculated or not. $newJAC$ is another Boolean variable which determines whether a new Jacobian needs to be calculated or not. The last Boolean variable *doublestep* determines whether the algorithm should double the size of the trust region and try a larger step at the same solution iterate. If the larger step fails then algorithm proceeds with the previous smaller step from the current iterate. Initial values, parameter relations and more details about the `TREGS` algorithm can be found in [99].

To prove convergence for a specific objective function, two sets of properties need to be satisfied, one set concerns the method/algorithm, and the second relates to the objective function being minimized [101]. The algorithmic properties are proved in [99]. For

Algorithm 1 TREGS: Trust region algorithm with regularized model solution

Choose initial approximate solution p_c; $r_c = r(p_c)$; $F_c = F(p_c)$; $m_c = F_c$; $J_c = J(p_c)$
Choose starting δ
$newSVD \leftarrow 1$; $newJAC \leftarrow 0$; $doublestep \leftarrow 0$
while not converged **do**
    if $newSVD$ **then**
        $[U, S, V] \leftarrow red\_svd(J_c)$
        $newSVD \leftarrow 0$
    end if
    Compute trial solution update s using Algorithm SU with r_c, U, Σ, V, τ, and δ
    Compute trial solution $p_t = p_c + s$, $r_t = r(p_t)$, $F_t = F(p_t)$
    Use SU output to compute (efficiently) $m_t = m_{GN}(p_t)$ from (3.1)
    $\rho = \frac{F_c - F_t}{m_c - m_t}$
    if $\rho \geq \eta_2$ **and** s is not a Gauss–Newton step **then**
        very successful step
        double trust region, $\delta \leftarrow 2\delta$, but continue to use p_c, F_c, SVD of J_c, r_c
        save current trial solution values in case larger step fails
        $doublestep \leftarrow 1$
    else if $\eta_1 \leq \rho < \eta_2$ **then**
        successful step
        accept the trial solution: $p_c \leftarrow p_t$; $F_c \leftarrow F_t$; $r_c \leftarrow r_t$
        $newSVD \leftarrow 1$; $newJAC \leftarrow 1$
    else if $\rho < \eta_1$ **then**
        unsuccessful step
        reject the trial solution and reduce trust region: $\delta \leftarrow \gamma_1 \delta$
        if $doublestep$ **then**
            set p_c, J_c, r_c, F_c to saved trial solution values
            $newSVD \leftarrow 1$; $newJAC \leftarrow 1$
        else
            $newJAC \leftarrow 0$; $newSVD \leftarrow 0$
        end if
    else if $\rho \geq \eta_2$ **and** s is the Gauss–Newton step **then**
        no need to try a larger step
        accept trial solution: $p_c \leftarrow p_t$; $F_c \leftarrow F_t$; $r_c \leftarrow r_t$
        $newJAC \leftarrow 1$; $newSVD \leftarrow 1$
    end if
    if $newJAC$ **then**
        $J_c \leftarrow J(p_c)$
        $newJAC \leftarrow 0$
    end if
end while

the PaLEnTIR problem, we prove the second set of properties in Appendix B. The (sufficient) requirements on the objective function in (2.7) are; (1) F($\mathbf{p}$) is twice continuously differentiable, (2) F($\mathbf{p}$) is bounded from below, and (3) the norm of its Hessian is bounded [101]. For the problems in this paper, the algorithm employs a discrepancy principle–based stopping criterion which states that one should stop iterating when the norm of the residual reaches the norm of the (weighted) noise vector [102]. In our experimental setup, an additional stopping criterion is implemented to prevent unnecessary computational expenditure. Specifically, the algorithm concludes its iterations when the relative decrease in the residual falls below a predefined threshold. However, these additional stopping criteria are not used in the proof that TREGS is guaranteed to converge to a stationary point [99].

2.3.2 ADAM (Adaptive Moment Estimation)

In this work, we use Adaptive Moment Estimation (ADAM) algorithm as an optimizer for training neural network models [103]. ADAM is a first-order gradient-based optimization algorithm that combines the advantages of two popular methods: AdaGrad [104] and RMSProp [105] . ADAM is a preferred optimizer in most deep learning tasks due to it's speed, and efficiency in working with large problems involving a lot of data or parameters. ADAM offers faster convergence compared to traditional gradient descent-based optimizers such as Stochastic Gradient Descent (SGD) by leveraging past gradient information. It computes each individual adaptive learning rates of parameters from estimates of first and second moments of the gradients. Some of the advantages are that magnitudes of parameter updates are invariant to rescaling of the gradient, step-sizes are approximately bounded by the step-size hyperparameter, it does not require a stationary objective, allows sparse gradients, and naturally performs step size annealing [103].

The pseudo-code of ADAM in [103] is given in Algorithm 2. At each time step t, ADAM updates the parameters by taking the following actions: First, it first computes the gradient g_t of the stochastic objective function with respect to the parameters θ_t. Then, it updates

the biased first moment estimate m_t and the biased second moment estimate v_t where the hyper-parameters β_1 and β_2 control the exponential decay rates of these moving averages respectively. To address the initialization bias that appears from initializing the moments with 0, ADAM applies bias-correction terms to these moment estimates, which yield $\hat{m}_t$ and $\hat{v}_t$. At the end of each step, the parameter updates are defined as:

$$\theta_t = \theta_{t-1} - \frac{\alpha \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon},$$

where α is the learning rate, and ϵ is a small constant added for numerical stability.

Algorithm 2 Adam: Stochastic optimization algorithm

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector

$m_0 \leftarrow 0$ ▷ Initialize 1st moment vector
$v_0 \leftarrow 0$ ▷ Initialize 2nd moment vector
$t \leftarrow 0$ ▷ Initialize timestep
while θ_t not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ ▷ Get gradients w.r.t. stochastic objective at timestep t
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ ▷ Update biased first moment estimate
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ ▷ Update biased second raw moment estimate
    $\hat{m}_t \leftarrow \frac{m_t}{1-\beta_1^t}$ ▷ Compute bias-corrected first moment estimate
    $\hat{v}_t \leftarrow \frac{v_t}{1-\beta_2^t}$ ▷ Compute bias-corrected second raw moment estimate
    $\theta_t \leftarrow \theta_{t-1} - \frac{\alpha \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$ ▷ Update parameters
end while
return θ_t ▷ Resulting parameters

2.4 Dimensionality Reduction

Dimensionality reduction is the process which transforms high-dimensional data into a lower-dimensional space while largely maintaining the key structures and relationships from the original data. Formally, given a dataset of N points $\{\mathbf{x}_i\}_{i=1}^N$ with each $\mathbf{x}_i \in \mathbb{R}^D$, a dimen-

sionality reduction method seeks a transformation

$$g : \mathbb{R}^D \to \mathbb{R}^d, \quad d \ll D,$$

such that the resulting set of points $\{\mathbf{y}_i\}_{i=1}^N$ with $\mathbf{y}_i = g(\mathbf{x}_i) \in \mathbb{R}^d$ keeps the key information and structure of the original data.

Dimensionality reduction exists primarily as a solution to the problems that high-dimensional datasets create. With growing dimension D, the computational cost and, in most cases, the size of data required to obtain reliable results grow exponentially, also known as the curse of dimensionality [10]. Conversely, reducing the degree of dimensionality has the potential to ease computational costs while improving the efficiency of data-driven algorithms and the discovery of meaningful patterns.

Dimensionality reduction not only addresses computational aspects but also eliminates noise and redundant features to improve performance of tasks such as classification, regression, and clustering [106]. In addition, low-dimensional representations are essential to visualize and interpret high-dimensional data and display patterns or groupings that would remain obscure otherwise [107].

The design of a dimensionality reduction procedure requires the knowledge of the intrinsic dimensionality and geometry of the data. These properties are often unknown, and therefore the dimensionality reduction is usually considerred as an ill-posed problem [108]. Therefore, these methods typically require to make assumptions regarding the properties of the data such as model complexity or geometry of the data based on error tolerances, available resources, and application requirements.

There are two primary groups of dimensionality reduction methods which are linear methods and nonlinear methods. Linear methods aim to find a linear subspace within the high-dimensional space that retains most of the variance or other essential statistical characteristics. Nonlinear methods operate without the linearity constraint in mind and work

on the assumption that the high-dimensional data contains complex structures. Nonlinear methods demonstrate superior representational power over linear methods because they can map latent and observed variables through complex transformations that extend beyond linear mappings. The increased flexibility necessitates a significantly larger number of parameters which demands substantially more data for accurate parameter estimation. For example, when PCA projects data from a D-dimensional space onto a P-dimensional subspace, the resulting model involves a number of parameters on the order of $O(PD)$. Notably, as few as $P + 1$ data points may suffice to fully determine these parameters. By contrast, $3^P D$ parameters is hardly enough to describe a basic nonlinear model [109].

In this work, we use both linear and nonlinear methods for the dimensionality reduction. For the linear dimensionality reduction method, this thesis considers PCA. For the nonlinear method, we consider autoencoders. In the rest of this section, we will focus on these two approaches.

2.4.1 Principal Component Analysis (PCA)

Linear methods achieve dimensionality reduction by mapping the data to a lower-dimensional linear subspace. Among numerous dimensionality reduction techniques PCA stands out as one of the dominant unsupervised linear method [110]. This thesis applies PCA as a method for linear dimensionality reduction.

PCA transforms high-dimensional data into a lower-dimensional space while preserving the highest possible amount of data variance [111]. Let us define a dataset that consists of N observations represented by $\{\mathbf{x}_i\}_{i=1}^{N}$ where each observation $\mathbf{x}_i \in \mathbb{R}^D$ has D features. Before PCA analysis takes place the dataset is centered to encourage equal influence of each feature in the analysis. The data centering process involves subtracting the sampled mean from all observations. For the dataset $\mathbf{X}$, elements of the sample mean vector $\{\mu_j\}_{j=1}^{D}$ can be calculated as:

$$\mu_j = \sum_{i=1}^{N} x_{i,j}. \tag{2.8}$$

Then, for a sample mean vector $\boldsymbol{\mu}$, the centered data $\mathbf{Z}$ takes the form:

$$\mathbf{Z} = \begin{bmatrix} (\mathbf{x}_1 - \boldsymbol{\mu})^\top \\ (\mathbf{x}_2 - \boldsymbol{\mu})^\top \\ \vdots \\ (\mathbf{x}_N - \boldsymbol{\mu})^\top \end{bmatrix} \in \mathbb{R}^{N \times D}. \tag{2.9}$$

The next step involves the calculation of the sample covariance matrix $\mathbf{C}$ using the previously centered dataset. The covariance matrix tells us how each individual feature relates to all each other within the dataset. Redundant features can be identified through detection of highly correlated attributes. For a centered dataset, the sample covariance matrix is defined as:

$$\mathbf{C} = \frac{1}{n-1}\mathbf{Z}^T\mathbf{Z}, \tag{2.10}$$

where $\mathbf{Z}^T$ is the transpose of the standardized data matrix $\mathbf{Z}$. Note that $\mathbf{C} \in \mathbb{R}^{D \times D}$ is a symmetric, positive semi-definite matrix.

Once the covariance matrix is computed, the next step is to calculate its eigenvalues and eigenvectors. After calculating the covariance matrix, we proceed to find its eigenvalues and eigenvectors. The eigenvectors of the covariance matrix represent the directions of the principal components. The eigenvalues provide a measure of the variance contained within each principal componentb [112]. A vector $\mathbf{v}$ with dimension N is an eigenvector of the matrix $\mathbf{C}$ if it satisfies a linear equation of the form:

$$\mathbf{C}\mathbf{v} = \lambda\mathbf{v}, \tag{2.11}$$

where the scalar λ is the eigenvalue. Since $\mathbf{C}$ is symmetric and positive semi-definite, it admits an eigen-decomposition of the form:

$$\mathbf{C} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top, \tag{2.12}$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_D]$ is the matrix of orthonormal eigenvectors. $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_D)$ is the diagonal matrix of eigenvalues ordered in descending order ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$). Note that, $\mathbf{C}$ is a real symmetric matrix. Every real symmetric matrix is Hermitian, and therefore all its eigenvalues are real. The k-th principal component direction is given by the k-th eigenvector $\mathbf{v}_k$. The corresponding eigenvalue λ_k indicates the amount of variance captured by that principal component [110].

The next step involves identifying principal components that show the highest variance levels in the data. We select a subset of principal components based on the largest the eigenvalues to reduce the dataset's dimensionality. Typically a threshold is determined to keep components that account for a major amount of variance like 90%. If we choose to keep the top k principal components $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ where $k < D$, we can form the matrix $\mathbf{V}_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k] \in \mathbb{R}^{D \times k}$. Then each centered observation $\mathbf{z}_i = \mathbf{x}_i - \boldsymbol{\mu}$ is projected onto this subspace as:

$$\mathbf{y}_i = \mathbf{V}_k^\top \mathbf{z}_i \quad \in \mathbb{R}^k. \tag{2.13}$$

The vector $\mathbf{y}_i$ is the coordinate vector of $\mathbf{x}_i$ in the k-dimensional principal component space [110].

2.4.2 Autoencoders

The previous subsection examined PCA which serves as a linear approach to reducing data dimensions. Our next topic of study is a nonlinear dimensionality reduction technique known as autoencoders.

Autoencoders use neural networks to create data representations from unlabeled sources for various applications such as dimensionality reduction [113]. Autoencoders works as follows: Initially the input data undergoes compression to form a lower-dimensional representation in the latent space and then it is reconstructed from this compressed form with high accuracy. Autoencoders serve as a robust unsupervised learning method as they learn the

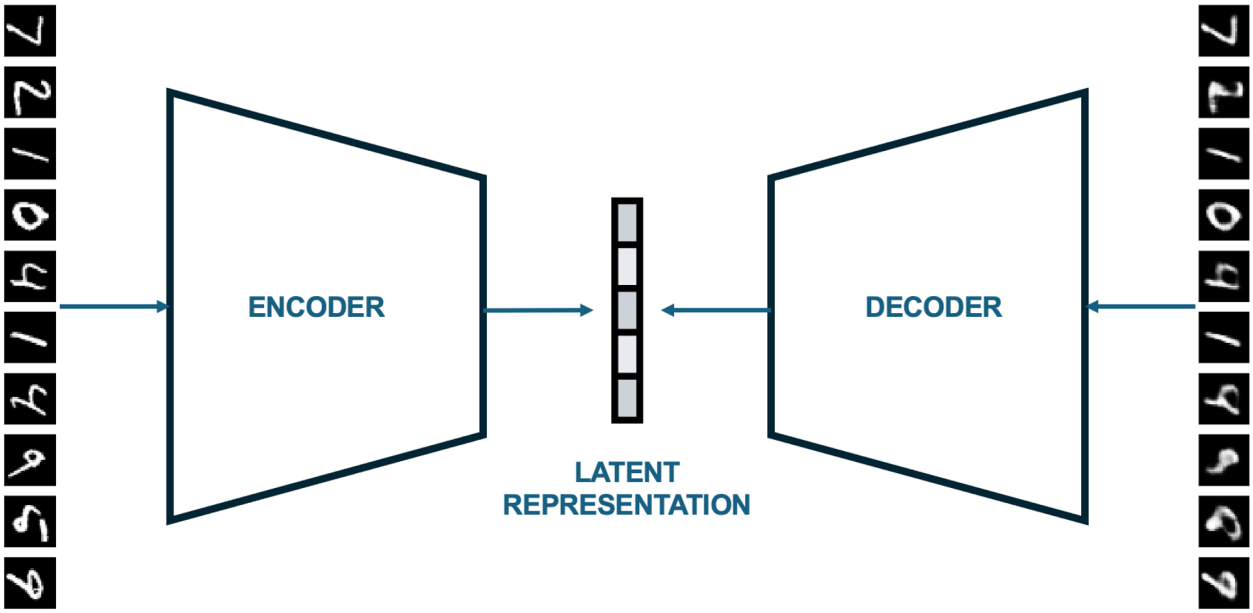


Figure 2.1: General structure of an autoencoder.

fundamental properties of data without requiring a labeling process [114].

Autoencoders have three main components: the encoder, the latent space, and the decoder. Each of the three parts has a specific role in the overall autoencoder framework. A typical autoencoder architecture is illustrated in Figure 2.1. Each of the three component isdescribed below in detail.

The encoder in an autoencoder is a function $f_a : \mathcal{X} \to \mathcal{Z}$ that maps the input data $x \in \mathcal{X}$ into a lower-dimensional, compressed representation in the latent space $\mathcal{Z}$. Mathematically, this is achieved by the mapping $z = f_a(x; \theta_{f_a})$, where θ_{f_a} are the learnable parameters of the encoder network. This function is typically realized through a series of neural network layers [115]. The parameters of these layers are learned in such a way that the encoder reduces the dimensionality of the data while maintaining as much of the relevant information as possible [116].

The latent representation, $z \in \mathcal{Z}$, functions as the encoded version of the input data. The dimensionality of $\mathcal{Z}$ is desirably much lower than that of $\mathcal{X}$ [113]. This compressed version is a lower dimensional representation of the input, which ideally captures the most essential features required reconstruct the original data with minimal loss.

The decoder in an autoencoder is a function $g_a : \mathcal{Z} \to \mathcal{X}$ that attempts to reconstruct the original input data x from the compressed representation z in the latent space. Mathematically, the reconstruction $\hat{x} = g_a(z; \theta_{g_a})$, where θ_{g_a} are the learnable parameters of the decoder network. The architecture of the decoder often mirrors that of the encoder, utilizing a series of neural network layers that progressively increasing the dimensionality until reaching the size of the original input from the size of the latent representation [115]. During the training of the autoencoder, the parameters θ_{g_a} are learned in a way that the decoder reconstructs $\hat{x}$ from z with the minimal difference between the input x and its reconstruction $\hat{x}$, usually measured through a loss function such as mean squared error or cross-entropy [117].

Both the encoder and the decoder components of the autoencoder architecture used in this thesis are built from artificial neural networks [118]. In the following section, we will delve deep into artificial neural networks, to understand their structure and functionality in detail.

2.5 Artificial Neural Networks

Neural networks are computational models inspired from the human brain's structure and function [119]. They consist of layers of multiple interconnected units or nodes called neurons, which are capable of collectively performing complex and difficult tasks that would be difficult or sometimes impossible to solve using traditional methods. Neural networks are the main component of many modern machine learning algorithms, particularly in deep learning, where they learn to perform tasks without being explicitly programmed with task-specific rules [120].

A neural network typically includes an input layer followed by one or several hidden layers which then lead into an output layer. The network structure includes multiple neurons per layer which establish weighted connections to neurons in the subsequent layer. Weights represent connection strength between neurons which determines the influence one neuron

has over another. During neural network learning, weights get updated by measuring the difference between predicted output and actual output [121].

There are many different neural network models available in the literature [122]. One of the key ways of characterizing the neural network models is by direction of the flow of information between its layers. In a feedforward neural network model, the flow of the information is uni-directional, meaning that the information in the model flows in only forward direction from the input nodes, passing through the hidden nodes and reaching to the output nodes, without any cycles or loops [121]. In contrast, a recurrent neural network (RNN) have a bi-directional flow where the building block of RNNs, called recurrent unit, maintains a hidden state, a form of memory, which is updated at each time step based on the current input and the previous hidden state [123].

The primary difference among neural network models emerges from how their layers interconnect. In fully-connected network layers each neuron connects to every neuron in the subsequent layer so that all neurons from the preceding layer affect every neuron in the current layer [124]. A convolutional layer architecture allows each neuron to receive input from only select neurons from the preceding layers of the network [125].

This thesis considers feedforward neural networks in fully connected layers. In feedforward fully connected neural networks (FCNNs), neurons are structured such that, each neuron is connected to every neuron in the following layer.This type of network structure can be expressed mathematically as follows:

$$\mathbf{a}^{(l+1)} = \sigma(\mathbf{W}^{(l)}\mathbf{a}^{(l)} + \mathbf{b}^{(l)})$$

where:

- $\mathbf{a}^{(l)}$ is the activation vector of layer l, of dimension $n_l \times 1$,
- $\mathbf{W}^{(l)}$ is the weight matrix connecting layer l to layer $l+1$, of dimension $n_{l+1} \times n_l$,
- $\mathbf{b}^{(l)}$ is the bias vector for layer l, of dimension $n_{l+1} \times 1$,

- σ represents the activation function.

The activation function σ introduces non-linearity into the network, enabling it to learn complex functions. Neural networks commonly use sigmoid, tanh, and ReLU functions as activation functions σ.

The training process for feedforward fully connected neural networks requires modifying weights $\mathbf{W}$ and biases $\mathbf{b}$ to reduce the loss function values. The optimization process typically employs algorithms such as stochastic gradient descent or Adam [103]. In section 2.3, we reviewed the Adam optimizer as we use Adam optimizer for training neural network models in this thesis.

2.6 Probability Theory

Probability theory forms the mathematical foundation for understanding uncertainty and variability in data [126]. This section briefly reviews the key concepts of probability theory relevant to the methodologies employed in this thesis.

Probability Spaces

A probability space is defined as a $W = \{\Omega, \mathcal{F}, P\}$, where:

- Ω is the sample space, representing all possible outcomes.
- $\mathcal{F}$ is an event space, where and event is a set of outcomes in Ω.
- $P : \mathcal{F} \to [0, 1]$ is a probability measure satisfying:
  1. $P(A) \geq 0$ for all $A \in \mathcal{F}$
  2. $P(\Omega) = 1$,
  3. $P(A \cup B) = P(A) + P(B)$ for disjoint $A, B \in \mathcal{F}$.

2.6.1 Random Variables

A random variable X is a measurable function $X : \Omega \to \mathbb{R}$, mapping set of possible outcomes in a sample space to real numbers.

Types of Random Variables

Random variables can be categorized into two main types:

- **Discrete Random Variables:** Take values from a finite or countable set. The probability mass function (PMF), $p_X(x)$, specifies the probability of each outcome:

$$p_X(x) = P(X = x), \quad \sum_x p_X(x) = 1. \tag{2.14}$$

- **Continuous Random Variables:** Take values from an uncountable set, typically intervals in $\mathbb{R}$. Their distribution is described by a probability density function (PDF), $f_X(x)$, such that:

$$P(a \leq X \leq b) = \int_a^b f_X(x)\,dx, \quad \int_{-\infty}^{\infty} f_X(x)\,dx = 1. \tag{2.15}$$

The cumulative distribution function (CDF) of random variable X is defined as

$$F_X(x) = P(X \leq x). \tag{2.16}$$

Expectation and Variance

The expectation of a continuous random variable X is given by:

$$\text{For discrete } X: \quad \mathbb{E}[X] = \sum_x x\, p_X(x), \tag{2.17}$$

$$\text{For continuous } X: \quad \mathbb{E}[X] = \int_{-\infty}^{\infty} x\, f_X(x)\,dx. \tag{2.18}$$

and its variance is:

$$\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2]. \tag{2.19}$$

Multiple Random Variables

When dealing with multiple random variables, their relationships and dependencies are described using joint, marginal, and conditional distributions:

- **Joint Distribution:** For random variables X and Y, the joint PDF $f_{X,Y}(x, y)$ describes the probability density over pairs (x, y).
- **Marginal Distribution:** Derived by integrating the joint PDF over one variable:

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y)\, dy. \tag{2.20}$$

- **Conditional Distribution:** Describes the probability of one variable given another:

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x, y)}{f_Y(y)}, \quad f_Y(y) > 0. \tag{2.21}$$

- **Independence:** Random variables X and Y are independent if and only if:

$$f_{X,Y}(x, y) = f_X(x) f_Y(y). \tag{2.22}$$

2.6.2 Gaussian Random Variables

PDF of a Gaussian random variable, referred to as $\mathcal{N}(x \mid \mu, \sigma^2)$, is defined as:

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{2.23}$$

where:

- μ is the mean of X.

- σ^2 is the variance of X.

Gaussian random variables have several key properties:

- **Symmetry:** The distribution is symmetric about the mean μ.
- **Expectation and Variance:** $\mathbb{E}[X] = \mu$ and $\text{Var}(X) = \sigma^2$.
- **Standard Normal Distribution:** A Gaussian random variable with $\mu = 0$ and $\sigma^2 = 1$ is called the standard normal random variable, denoted $Z \sim \mathcal{N}(0, 1)$.

Multivariate Gaussian Distributions

The multivariate Gaussian distribution extends the univariate case to d dimensions. A random vector $\mathbf{X} \in \mathbb{R}^d$ is a multivariate Gaussian distribution, referred to as $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$, if its PDF is:

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right), \tag{2.24}$$

where:

- $\boldsymbol{\mu} \in \mathbb{R}^d$ is the mean vector.
- $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ is the covariance matrix, which is symmetric and positive semi-definite.

2.7 Density Estimation

Density estimation involves reconstructing the probability density function that defines a dataset using samples that have been observed [127]. The objective with dataset $\{x_1, x_2, \ldots, x_n\}$ is to create an approximation of the PDF $p_X(x)$ which produced the data.

There are two main classes of density estimation techniques which are divided into parametric methods and non-parametric methods. Parametric methods assume that we have a prior knowledge about the form of the distribution. Gaussian mixture models is one of the most commonly used parametric density estimation approaches [128]. However, we do not

always have a prior knowledge about the probability distribution. Non-parametric methods can be used with arbitrary distributions and without the assumption that the form of the underlying densities are known [129]. Histogram, kernel density estimator and k-nearest neighbor estimator are some of the commonly used non-parametric density estimators.

In this work we use histograms and Gaussian mixture models for the purposes of data processing and density estimation. In the rest of the section, we will review these two density estimation methods.

2.7.1 Histograms

Histograms function as fundamental non-parametric statistical tools that enable researchers to estimate and depict the probability distribution of datasets [130]. The data range gets divided into separate bins with each bin containing the number of data points that belong to it. Histograms serve as valuable tools in exploratory data analysis because they enable fast evaluation of data distribution features including modality, symmetry, skewness and outliers.

Consider a univariate dataset comprising observations $\{x_i\}_{i=1}^n$ drawn from an unknown probability distribution. To construct a histogram, the data range is divided into k bins:

$$B_j = [a_{j-1}, a_j), \quad \text{for} \quad j = 1, 2, \ldots, k,$$

where $a_0 < a_1 < \cdots < a_k$ represent bin edges, typically chosen to be equally spaced. The width of the bin, denoted as Δ, is given by:

$$\Delta = \frac{a_k - a_0}{k}.$$

The histogram is then defined by counting the number of observations n_j falling into each bin B_j:

$$n_j = \sum_{i=1}^{n} \mathbf{1}_{B_j}(x_i), \quad j = 1, 2, \ldots, k,$$

where $\mathbf{1}_{B_j}(x)$ is the indicator function defined as:

$$\mathbf{1}_{B_j}(x) = \begin{cases} 1, & \text{if } x \in B_j \\ 0, & \text{otherwise} \end{cases}.$$

The empirical density function is approximated by the normalized histogram as:

$$\hat{f}(x) = \frac{n_j}{n\Delta}, \quad \text{for} \quad x \in B_j,$$

where it holds that $\sum_{j=1}^{k} n_j = n$.

Histogram assigns an equal density value to every point within the bin, effectively treating all observations in a bin as equally likely to occur. The bins are often chosen to have equal width, but adaptive binning can be used for better representation of the data [131]. In this work, we treat all observations as equally likely and choose all the bins to have equal width.

2.7.2 Gaussian Mixture Models

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of multiple Gaussian component densities. It is widely used to model the distribution of continuous data [132].

Mathematically, a GMM is expressed as:

$$p_X(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k), \tag{2.25}$$

where:

- K is the number of Gaussian components,
- π_k are the mixture weights satisfying $\sum_{k=1}^{K} \pi_k = 1$,
- $\mathcal{N}(x|\mu_k, \Sigma_k)$ represents the Gaussian component density with mean μ_k and covariance

matrix Σ_k.

Each component density is given by:

$$\mathcal{N}(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right), \tag{2.26}$$

where D is the dimensionality of the data.

The GMM parameters $\{\pi_k, \mu_k, \Sigma_k\}$ are typically estimated using the Expectation-Maximization (EM) algorithm, which iteratively maximizes the likelihood of the observed data [133].

2.7.3 Change Point Detection

Change point detection (CPD) is the task of identifying time points at which the underlying distribution of a time series changes abruptly. More formally, let

$$\{\mathbf{x}_t\}_{t=1}^T$$

be a sequence of observations in a time series of length T, where $\mathbf{x} \in \mathbb{R}^d$. If the data-generating process is piecewise stationary, there exist time indices $\{\tau_1, \tau_2, \ldots, \tau_k\}$ such that

$$\mathbf{x}_1, \ldots, \mathbf{x}_{\tau_1} \sim \mathcal{D}_1, \quad \mathbf{x}_{\tau_1+1}, \ldots, \mathbf{x}_{\tau_2} \sim \mathcal{D}_2, \quad \ldots, \quad \mathbf{x}_{\tau_k+1}, \ldots, \mathbf{x}_T \sim \mathcal{D}_{k+1},$$

where each $\mathcal{D}_i$ denotes a (unknown) distribution. The set of time indices $\{\tau_1, \tau_2, \ldots, \tau_k\}$, are referred to as the change points. The problem of unsupervised CPD is to estimate the set of change points from the sequence of observations based on the previous observations.

In this work, we focus on CPD in high-dimensional time series data arising from *multi-agent* systems, where each observation x_t may itself represent the combined state of N_a agents at time t. Such multi-agent systems are pervasive in real-world scenarios: from sheep herds [134] ; to crowd dynamics [135, 136]. The essential feature of these systems is *collective behavior* which develops through simultaneous local interactions between multiple

agents [137]. Researchers investigate collective behavior by studying scenarios like flocking and swarming movements and crowd motion dynamics which can undergo sudden significant transformations as a result of environmental stimuli or through decision-making processes and external interventions.

Chapter 3

Parametric Level-sets Enhanced To Improve Reconstruction (PaLEnTIR)

This chapter is based on [93]

In chapter 2, we stated that previous PaLS models, despite their advantages over traditional level-set methods, still have room for improvements in terms of three main categories; multi-contrast recovery, shape expressiveness, and redundancy and indeterminacy in the parameterization. In this chapter we present PaLEnTIR, a significantly enhanced parametric level-set (PaLS) method to improve the restoration and reconstruction of piecewise constant objects by addressing the above mentioned problems observed in previous PaLS models. The proposed PaLEnTIR model is defined as follows

$$f(\mathbf{r};\mathbf{p}) = C_H(\mathbf{r})T\left(\phi(\mathbf{r};\mathbf{p})\right) + C_L(\mathbf{r})\left(1 - T\left(\phi\left(\mathbf{r};\mathbf{p}\right)\right)\right), \tag{3.1}$$

$$\phi(\mathbf{r};\mathbf{p}) = \sum_{j=1}^{N} \sigma_h\left(\alpha_j\right)\psi\left(\mathbf{R}_j(\mathbf{r} - \boldsymbol{\chi}_j)\right), \tag{3.2}$$

$$\psi(\mathbf{r},\mathbf{p}) = e^{-\|(\mathbf{R}(\mathbf{r}-\boldsymbol{\chi}))\|_2^2}. \tag{3.3}$$

Where $\mathbf{p}^T := \begin{bmatrix} \boldsymbol{\alpha}^T & \boldsymbol{\beta}^T & \boldsymbol{\gamma}^T \end{bmatrix}$ and as described below $\mathbf{R}_j$ depends on $\boldsymbol{\beta}_j$ and $\boldsymbol{\gamma}_j$. In this model, T is the new transition function, replacing the Heaviside function, $C_H(\mathbf{r})$ and $C_H(\mathbf{r})$ are the new "contrast coefficients", replacing f_O and f_B in (2.5). The weight coefficient $\sigma_h(x)$ is defined as

$$\sigma_h(x) = \frac{1 - e^{-x}}{1 + e^{-x}}. \tag{3.4}$$

The new parametric level-set function of PaLEnTIR, $\phi(\mathbf{r}; \mathbf{p})$ defined in (3.2) is formed by the weighted summation of anisotropic basis functions (ABFs), $\psi(\mathbf{r}, \mathbf{p})$, that are defined in (3.3). The N matrices, $\mathbf{R}_j := \mathbf{R}(\boldsymbol{\beta}_j, \boldsymbol{\gamma}_j)$, for $j = 1, 2 \ldots, N$, depend on the $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ sub-vectors of $\mathbf{p}$, respectively. Suppressing the explicit dependence on β and γ, in 2D $\mathbf{R}$ is defined as follows

$$\mathbf{R} = \mu \begin{bmatrix} e^{\beta} & \gamma \\ 0 & e^{-\beta} \end{bmatrix} \tag{3.5}$$

where, as detailed in Section 3.2 β and γ define the eccentricity and orientation of the elliptical level-sets and μ is a constant that controls the maximum allowable size of the ABFs. We call (3.5) as the *Stretch and Slide Matrix*. The 3D version of the *Stretch and Slide Matrix* is provided in Section 3.2

Though structurally similar to the original RBF-PaLS modeling (2.6), the PaLEnTIR model in (3.1) enhances PaLS in the following ways, each of which is discussed in depth in the following sections:

A) **Multi-contrast, single level-set** reconstructions are obtained by: (a) Replacing the Heaviside function with the smooth transition function $T(x) : \mathbb{R} \to (0, 1)$; and (b) Relabeling and reinterpreting $f_O(\mathbf{r})$ and $f_B(\mathbf{r})$ as upper and lower contrast bounds, $C_H(\mathbf{r})$ and $C_L(\mathbf{r})$. [See section 3.1 for detailed discussion.]

B) **Shape-expressiveness** is expanded by replacing the scalar dilation coefficient, β_j in PaLS, by a matrix $\mathbf{R}_j$, implementing what we call stretching and sliding. [See section 3.2 for detailed discussion.]

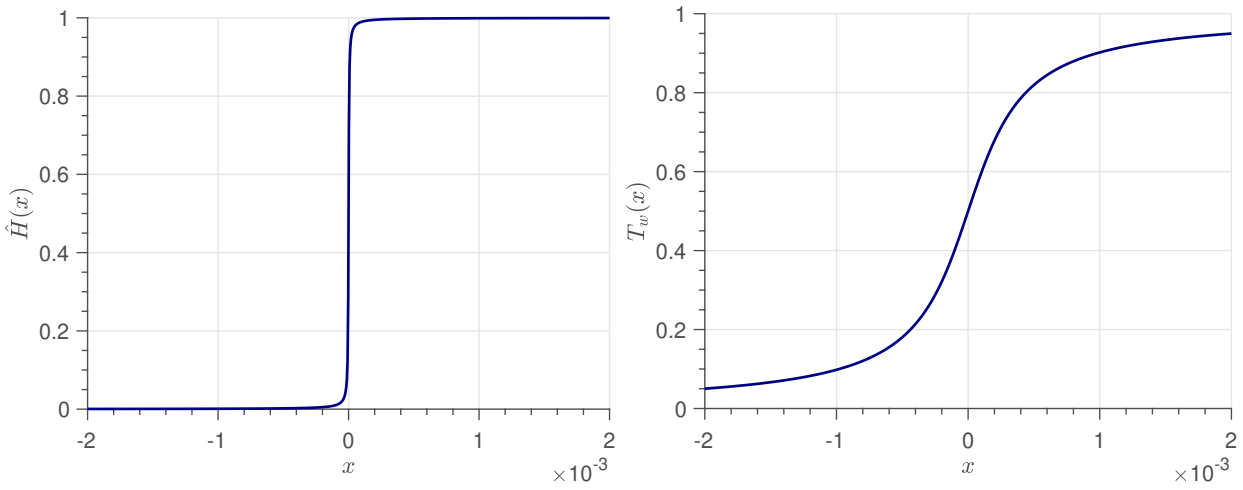
C) **Numerical performance** is improved by (a) Fixing the basis function centers on a grid of pre-specified points, $\boldsymbol{\chi}_j$; (b) Constraining the size of the expansion coefficients using a function $\sigma_h(\alpha_i) \in (-1, 1)$; and (c) Replacing RBFs, that have non-unique mapping of parameters to c-level-sets, with the new anisotropic basis functions (ABFs). [See section 3.3 for detailed discussion.]

3.1 Multi-contrast, Single Level-set Reconstructions

In the previous section, we assumed that the property function $f(\mathbf{r})$ had a binary structure, i.e., each point in Ω either belongs to the region O or the background. However, it may well be the case that we have to represent regions containing objects with more than two contrast values [138, 139, 140]. RBF PaLS uses approximate Heaviside function designed for distinguishing between two contrasts and thus requires a number of level-set functions proportional to the logarithm of the number of contrast levels in the problem. To solve this problem, we replace the approximate Heaviside function in the RBF PaLS model with a transition function $T(x)$ which smoothly, and monotonically varies between zero and one. Specifically, in this work we take

$$T(x) = \frac{1}{2}\left[1 + \frac{2}{\pi}\tan^{-1}\left(\frac{\pi(x-c)}{w}\right)\right], \tag{3.6}$$

where w determines the steepness of the transition region. The approximate Heaviside function used in the RBF PaLS work also monotonically increased from zero to one [21] and thus is similar to $T(x)$. The difference lies in the width of the transition. As seen in Figure 3.1a, for the approximate Heaviside function, the transition region is by construction very narrow, as the goal was to represent binary valued objects, whose "phases" are separated by a narrow, smooth interface. In the new formulation, we stretch the width of this transition region, so that there is a far broader range of $\mathbf{r}$ for which $f(\mathbf{r})$ can assume values between C_L and C_H, as seen in Figure 3.1b.



(a) (b)

Figure 3.1: Plots of (a) the approximate Heaviside function $\hat{H}(x)$ and (b) the new transition function $T(x)$ for the zero level-set ($c = 0$).

To illustrate the impact of these changes to the PaLS model, in Figure 3.2a, we display an image of size 256×256 pixels, comprised of five piecewise constant objects on a zero-contrast background. Here, we took the forward model to be the identity, so that $\mathbf{d} = \mathbf{f}(\mathbf{p}) + \mathbf{w}$, where $\mathbf{f}(\mathbf{p})$, a discrete representation of $f(\mathbf{r}; \mathbf{p}, C_H, C_L)$, is a vector of length $N_{pts} = 256^2 = 65536$ obtained using the discretization process described in section 3.4. The vector $\mathbf{w}$ contains independent, identically distributed Gaussian random variables, with variance such that the signal to noise ratio (SNR) of the corrupted image is 22 dB. The corrupted image is shown in Figure 3.2b, along with our PaLEnTIR reconstruction in Figure 3.2c. We used 225 basis functions centered on an equally spaced 15×15 grid. Each basis function requires the parameters α, β and γ, resulting in total of $3N$ (675) parameters for N (225) basis functions. Contrast coefficients are chosen as $C_H = 1$ and $C_L = 0$, since the maximum contrast in the image is equal to 1 and the minimum is equal to 0.

In Figure 3.2c, we see that the bright yellow region, whose contrast is equal to C_H, is recovered most accurately among the objects. The α_i of the basis functions in this region take on relatively large values which are truncated by $T(x)$. The regions without objects are also recovered very accurately. In those areas the coefficients are relatively small, so that

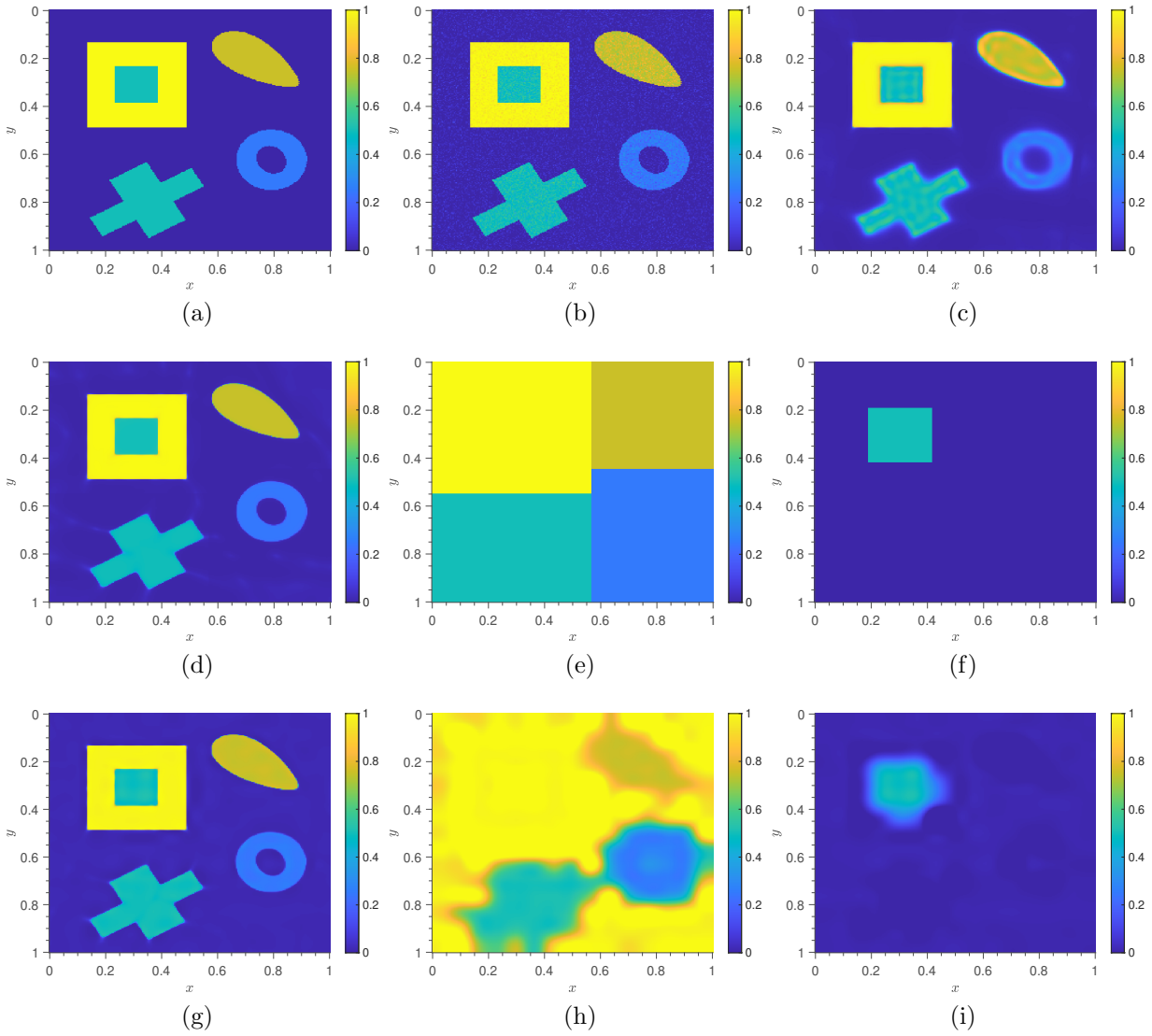


Figure 3.2: (a) Image of five piecewise-constant objects with varying shapes and contrasts. (b) The image corrupted with additive Gaussian noise. (c) PaLEnTIR reconstruction with fixed contrast limits. (d) PaLEnTIR reconstruction using manually adjusted vectors (e) $\mathbf{C}_H$ and (f) $\mathbf{C}_L$. (g) PaLEnTIR reconstruction with parameterized contrast limits, $\mathbf{p}_c \in \mathbb{R}^{2N}$, included in the estimation problem. Estimated parameterized contrast limits: (h) $C_H(\mathbf{r})$ and (i) $C_L(\mathbf{r})$.

the transition function $T(x)$ creates a constant, zero-background. For the remainder of the structures whose contrasts are between 0 and 1, we see blurred edges and oscillations similar to what we would expect from least squares denoising. Overall, the use of $T(x)$ resulted in accurate recovery of regions with values not between C_H and C_L, and rough recoveries of both shapes and contrasts of regions with values in between C_H and C_L. This is achieved

through the utilization of a single level-set, requiring only 675 unknowns, despite dealing with an 65536-pixel image.

The strong results in Figure 3.2c for the background and the yellow object in the upper left corner arise from the ability of the transition function to truncate the values of ϕ which are not between C_H and C_L. More generally, PaLEnTIR can achieve similar results across the entire scene by varying C_H and C_L in space. For example, consider the case where we set $C_H(\mathbf{r})$ and $C_L(\mathbf{r})$ as shown in Figures 3.2e and 3.2f. Re-estimating $\mathbf{p}$ with these distributions of $C_H(\mathbf{r})$ and $C_L(\mathbf{r})$ yields the results in Figure 3.2d, where we now see near-perfect recovery of all objects and the background.

The results in Figure 3.2d and the associated discussion illustrate the ability of the PaLEnTIR model to recover piecewise constant scenes comprised of objects with more than two contrasts using a single level-set function. This feature rests heavily on the ability to specify space-varying bounds on the contrasts at relatively coarse scales. The key issue is to do this in a way that retains the advantages of a PaLS-type model: low-order and no need for explicit regularization. In this work, this is accomplished by parameterizing $C_H(\mathbf{r})$ and $C_L(\mathbf{r})$, the spatially varying contrast limits, using bi-cubic (tri-cubic for 3D) interpolation [141, 142] (via the Matlab function "*imresize*"). Thus, in addition to α, β, γ, for each basis function, we add two new parameters, which we refer as p_H and p_L, for upper and lower contrast bounds respectively. Consequently, for a PaLEnTIR model with N basis functions there are two N-vectors of parameters, referred as $\mathbf{p}_H$ and $\mathbf{p}_L$. To see the impact of parameterizing the contrast limits, we repeat the same experiment with the same $\mathbf{w}$ to recover the multi-contrast objects in Figure 3.2a. We initialize $\mathbf{p}_H$ and $\mathbf{p}_L$ to ones and zeros respectively and rerun the `TREGS` algorithm. Figure 3.2g shows the resultant PaLEnTIR representation. The number of estimated parameters is increased from 675 to only 1125, still significantly fewer than the number of pixels in the represented image, which is 65536. As seen in Figure 3.2g, the new model with parameterized contrast limits has recovered all objects and the background accurately with near-perfect sharp boundaries, and maintains

smooth piecewise constant contrasts everywhere. The contrast limits produced with the estimated $\mathbf{p}_c = [\mathbf{p}_H; \mathbf{p}_L]$, are shown in Figures 3.2h and 3.2i. The estimated contrast limits capture the contrast values of the objects at a relatively coarse scale. Noticeably, they look very similar to the handcrafted $C_H(\mathbf{r})$ and $C_L(\mathbf{r})$ in Figures 3.2e and 3.2f, especially at or near where the objects are located.

3.2 Advancements in the shape-expressiveness

While radial basis functions (RBFs) provide flexibility in terms of shape representation for PaLS, they are limited by the fact that they posses only circular, in 2D, or spherical, in 3D, level-sets, making them inefficient for representing, e.g., highly anisotropic shapes [61]. Motivated by these observations, we replace the RBFs in the RBF PaLS [21] with a model of the form defined in (3.3) where $\mathbf{R}$ is a 2×2 matrix for 2D problems and 3×3 for 3D problems. As in [61], this model produces elliptical cross-sections; however, the parameterization of $\mathbf{R}$ is new and requires fewer parameters. We discuss the 2D and 3D cases separately. In the 2D case, $\mathbf{R}$ is defined as in (3.5). The parameters β and γ define the eccentricity and orientation of the elliptical level-sets, while μ impacts the area of the ellipses. In this work, β and γ are parameters to be determined, while μ is fixed. To elucidate the role of these parameters, we consider the $N = 1$ case for both PaLS and PaLEnTIR. We define the c-level-set representation of a curve Γ as $\Gamma = H(\phi(\mathbf{r}, \mathbf{p}) - c)$. That is, Γ is the set of points $\mathbf{r}$ such that $\phi(\mathbf{r}; \mathbf{p}) = c$ with $c > 0$:

$$\Gamma = \left\{(x, y) \in \mathbb{R}^2 | \phi(x, y; \mathbf{p}) = c\right\}. \tag{3.7}$$

We begin by examining the PaLS c-level-set of a single RBF centered at the origin:

$$\phi_{\text{rbf}}(x, y, \alpha, \beta) = \alpha e^{-\beta(x^2+y^2)}. \tag{3.8}$$

With ϕ_{rbf} in (3.7) and $c > 0$, we require $\alpha > 0$ as well to obtain a nonempty shape. Simple algebra gives

$$\Gamma_{\text{rbf}} = \left\{ (x, y) \in \mathbb{R}^2 \,\middle|\, x^2 + y^2 = \frac{1}{\beta} \ln \frac{\alpha}{c} \right\}. \tag{3.9}$$

As anticipated, (3.9) defines a circle with radius $\sqrt{\frac{1}{\beta} \ln \frac{\alpha}{c}}$. We emphasize here that, with c fixed, there are an infinite number of (α, β) pairs that will give the same circle.

Next, we consider the $N = 1$ PaLEnTIR c-level-set where $\phi(x, y, \alpha, \mathbf{R}) = \sigma_h(\alpha) \exp\left(-\|\mathbf{R}\mathbf{r}\|_2^2\right)$ with $\mathbf{r}^T = [x\ y]$. Notice that σ_h, defined in (3.4), bounds the weight coefficients of the basis functions between -1 and 1. Using ϕ in (3.7) and $c > 0$, a non-empty Γ requires $\sigma_h(\alpha) > 0$ and thus $\alpha > 0$. Dividing both sides by $\sigma_h(\alpha)$ and taking the logarithm yields

$$\Gamma_{\text{new}} = \left\{ (x, y) \in \mathbb{R}^2 \mid \left(e^{\beta} x + \gamma y\right)^2 + e^{-2\beta} y^2 = \tau^2 \right\} \tag{3.10}$$

with $\tau^2 = \frac{1}{\mu^2} \ln \frac{\sigma_h(\alpha)}{c}$. When β and γ are 0, similar to Γ_{rbf}, Γ_{new} is a circle with a radius of τ. As $0 < |\sigma_h(\alpha)| \leq 1$, and μ and c are constants, the maximum area of a single ABF is bounded, to restrict the extent to which the ABFs overlap each other. Note that in this work the c-level is set to 0.01, a value slightly larger than 0, for the reasons discussed in [21] while $\mu = 10$ which makes the maximum area of a single ABF to be around 15% of the total area when the image coordinates are normalized to a unit square or unit cube (as in all cases considered in this work).

When β and γ are nonzero, the Γ_{new} curve becomes elliptical. To understand the role of each parameter in defining the geometry of the ellipse, we start by viewing $\mathbf{R}$ as the Cholesky factor of the symmetric positive definite (SPD) matrix $\mathbf{A} = \mathbf{R}^T\mathbf{R}$, which we call the *Stretch and Slide Matrix*. This matrix includes two types of parameters which are estimated by the PaLEnTIR reconstruction: the *stretching* parameter β and the *sliding* parameter γ. In Figure 3.3a, the impact of the *stretching* parameter β is shown. With $\gamma = 0$, nonzero values of β produce ellipses with principle axes in the x and y directions. We visualize the impact of increasing β as someone holding the circle (assuming $\beta = 0$ initially) at two

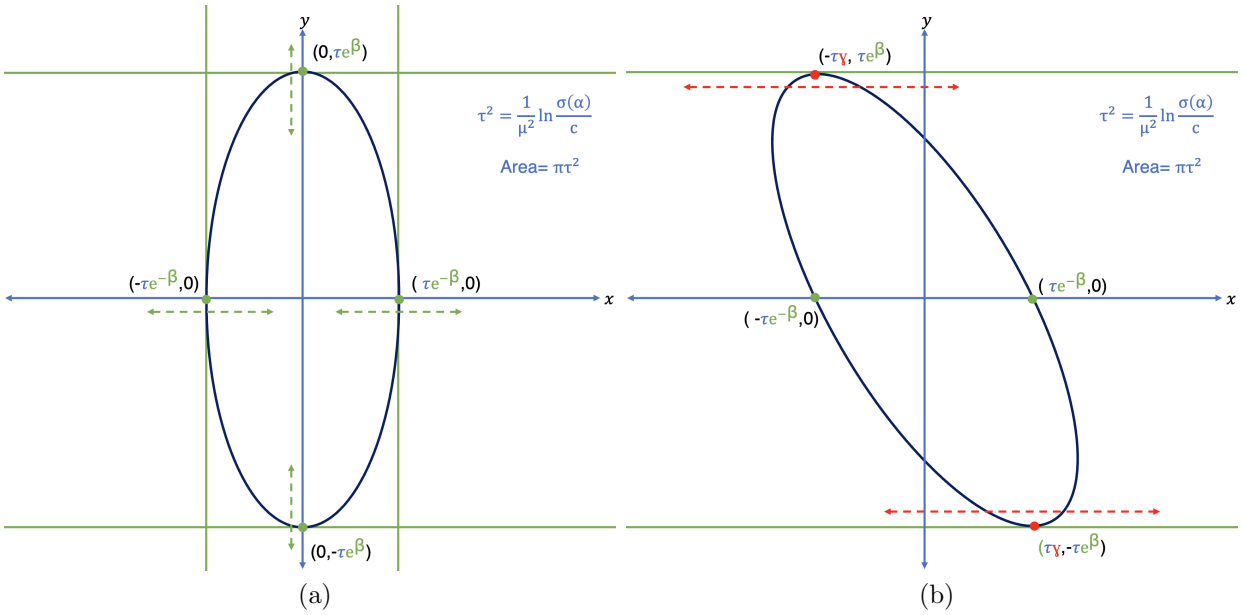


Figure 3.3: (a) Effect of the *stretching* parameter β, and (b) effect of the *sliding* parameter γ on the c-level ellipsoidal representation of the new basis function.

opposite points touching the x and y axes (points shown with green dots in Figure 3.3a), and stretching in the direction of green dashed arrows. We call this shape transformation "stretching," and hence β the *stretching* parameter. Similarly, as shown in Figure 3.3b, γ is the *sliding* parameter, because increasing γ horizontally "slides" the maximum and the minimum points of the ellipse along the vertical axes, as shown in with red dashed arrows. Since the determinant of $\mathbf{A}$ is μ^2, these two shape transformations do not change the area of the ellipse. The area depends on the constants μ, c-level, and the parameter α. In summary, with μ and c is fixed, α controls the total area, β controls the upper and lower tangent lines by stretching the ellipse, and γ slides the tangent points of the ellipse along the tangent lines.

If we compare (3.9) and (3.10), we can clearly see that the β in ϕ_{rbf}, which adjusts the scale of the circles, is replaced with μ^2. As we explained above, the parameters α and β in ϕ_{rbf} have a related (but opposite) effect on the c-level-set, and infinitely many pairs (α, β) exist that give the same Γ_{rbf}. Hence, we replaced the β in ϕ_{rbf} with a constant μ^2 which has no influence on the c-level-set, to reduce the dimension of the search space of the model by

eliminating infinite pairs of (α, β), and use only the parameter α to determine the scale of the ellipses.

For the 3D case, we construct $\mathbf{R}_{3d} \in \mathbb{R}^{3\times3}$ using three stretch and slide matrices, as follows

$$\mathbf{R}_{3d} = \mu \begin{bmatrix} e^{\beta_1} & \gamma_1 & 0 \\ 0 & e^{-\beta_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{\beta_2} & \gamma_2 \\ 0 & 0 & e^{-\beta_2} \end{bmatrix} \times \begin{bmatrix} e^{\beta_3} & 0 & \gamma_3 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-\beta_3} \end{bmatrix}.$$

Note there are six parameters in our model, with each of the three matrices in the product affecting a shear transformation in a 2D plane within $\mathbb{R}^3$. Similar to 2D case, $\mathbf{R}_{3d}$ creates anisotropy in x, y, z and, assuming all three β_i are finite, may be viewed as the Cholesky factor of the symmetric positive definite (SPD) matrix $\mathbf{A} = \mathbf{R}_{3d}^T\mathbf{R}_{3d}$. We note that in the 3D PaLS model in [61], the authors modify the CSRBF formulation to use a matrix-based dilation to define ellipsoids. Unlike what we propose here, their method uses a symmetric 3×3 matrix, parameterized by the six unique elements of such a structure. One of the advantages of our formulation is that $\mathbf{R}_{3d}^T\mathbf{R}_{3d}$ is always SPD, whereas in the 3D PaLS model in [61] regularization is required to enforce this constraint.

3.3 Numerical improvements and stability

PaLEnTIR eliminates two key sources of redundancy inherent in RBF PaLS, thus improving numerical stability and performance.

First, PaLEnTIR resolves the indeterminacy arising from floating centers in RBF PaLS. In the case of RBFs, the cost function remains invariant under permutation of the basis functions by switching the parameters, which introduces a structural redundancy in the model. This permutation invariance indicates that there is no unique representation of the shapes in terms of the basis functions. We hypothesize that these redundancies manifest as an

increased condition number in the Jacobian matrix, which can impact the convergence rate and stability of the inversion process. By fixing the centers of the anisotropic basis functions (ABFs) in PaLEnTIR, we remove this source of indeterminacy and as demonstrated shortly reduce the dimensionality of the search space, thereby improving both the conditioning of the Jacobian and the overall performance during inversion.

Second, PaLEnTIR addresses redundancy in RBF PaLS associated with what we term the "width-height trade-off." In RBF PaLS, as illustrated using a single basis function in section 3.2, the same circular shape can be obtained by varying either the α (height) or β (width) parameters, introducing further indeterminacy. This results in non-uniqueness in the parametric representation of the basis functions, which we show below degrades numerical stability. In PaLEnTIR, we mitigate this issue by (a) bounding the weight coefficients of the basis functions through the use of $\sigma_h(\alpha)$ and (b) using the stretch-and-slide matrix to define the c-level-set geometry both of which control the width-height trade-off.

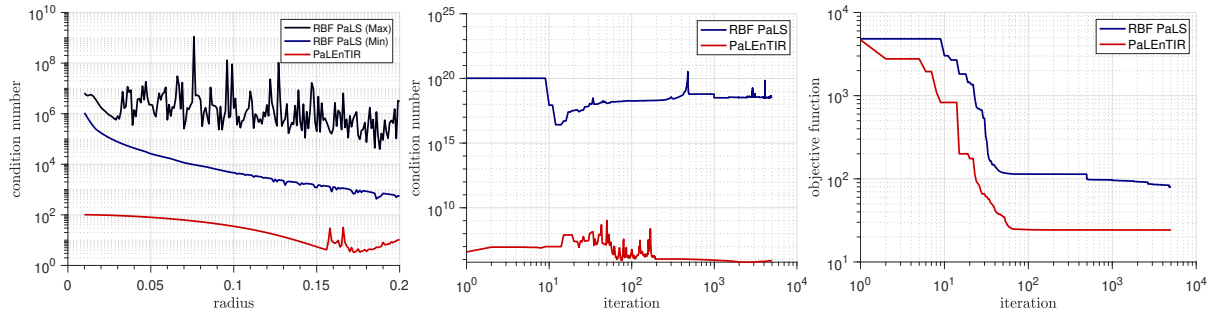
While we do not claim that PaLEnTIR entirely eliminates all possible redundancies, we argue that these three modifications—fixing the centers of ABFs, bounding the weight coefficients, and adding the stretch-and-slide matrix—substantially reduce the sources of indeterminacy in the model. These changes qualitatively improve the structure of the problem, making it more amenable to optimization. In the rest of this section, we quantitatively demonstrate these improvements through empirical results, showing that PaLEnTIR leads to better performance in terms of the conditioning of the Jacobian and the convergence properties of the TREGS algorithm.

Here, we use the condition number of the Jacobian matrix, defined as the ratio of the largest to smallest singular value, to quantify numerical performance. As discussed in [143, 144], this is a crucial metric in determining the performance of a Newton-type approach to the estimation of the model parameters. Values of the condition number near 1 indicate a well-conditioned matrix and large values indicate an ill-conditioned matrix. The expressions detailing the derivatives of the PaLEnTIR parameterization with respect to model param-

eters, which are necessary for the derivation of the Jacobian, are provided in Appendix A. However, it should be noted that the condition number of the least squares problem depends not just on the condition number of the Jacobian but also on the angle between the residual and the range of the Jacobian [145].

Similar to the discussion in section 3.2, our initial focus is on the single-basis function scenario. Given that RBF PaLS is confined to generating circular cross-sections (see section 3.2), we evaluate the numerical performance of both models in terms of their condition numbers when their respective basis functions generate identical circles in their c-level-set representations and the forward model is taken to be the identity in a noise-free case. Figure 3.4a plots the condition number of the Jacobian matrix evaluated at the true parameters against the radius of the circular cross-sections for both RBF PaLS and PaLEnTIR models. Given that the leading coefficient of PaLEnTIR limits the size of produced shapes, we focus on circular cross-sections with radii ranging from 0.01 to 0.2 units within a square region of 2 units per side. The condition number is computed using the function "*cond(.)*" in Matlab. Unlike PaLEnTIR, RBF PaLS can produce the same cross-section with many (α, β) pairs. Consequently, we produce identical cross-sections for radii ranging from 0.01 to 0.2, with increments of 0.001. We vary (α, β) pairs, increasing α by 0.01 within the range of 0.01 to 1. For each radius, we plot both the minimum and maximum condition numbers. The black and blue lines represent the maximum and minimum condition numbers for RBF PaLS, respectively, and the red line represents the condition number for PaLEnTIR. Notably, even if specifically selecting the (α, β) pairs that yield the minimum condition number for RBF PaLS, this is still much larger than the condition number obtained with PaLEnTIR. In the *worst* case scenario for RBF PaLS, where we pick the (α, β) pairs with the largest condition number for each cross section radius, the condition number is significantly larger than for PaLEnTIR.

Next, we evaluate the numerical performance of the methods using the deconvolution problem described in Subsection 3.4.1. The reference image of the object, which we aim to



(a) (b) (c)

Figure 3.4: (a) Condition number of the Jacobian matrix versus the radius of circular cross-sections (corresponding to level-sets of single basis functions) for the two models. The black line (max) and the blue line (min) represent RBF PaLS, while the red line represents PaLEnTIR. (b) Condition number of the Jacobian matrix versus `TREGS` iterations (log-log scale); and (c) Objective function values over `TREGS` iterations (log-log scale) for the deconvolution experiment discussed in Subsection 3.4.1.

reconstruct, is shown in Figure 3.5a. This analysis was performed using a 11×11 grid of basis functions per level-set for both the RBF PaLS and PaLEnTIR configurations, with the corresponding reconstructions presented in Figures 3.5d and 3.5e, respectively.

Figure 3.4b plots the condition number of the Jacobian versus TREGS iterations, where the blue line represents RBF PaLS and the red line represents PaLEnTIR. Throughout the iterations, the condition number for PaLEnTIR remains significantly lower than that of RBF PaLS, consistent with the observations in the single-basis function case. On average, the condition number of RBF PaLS is 2.7×10^{12} times higher than that of PaLEnTIR over the iterations.

Figure 3.4c plots the objective function (misfit) against the number of `TREGS` iterations. Again, the blue line represents RBF PaLS, while the red line corresponds to PaLEnTIR. According to the objective function plots, PaLEnTIR achieves a misfit score that is roughly one-thirth of the RBF PaLS misfit score, aligning with the visual observations made in section 3.4.2. Furthermore, the convergence in the objective function plot occurs much faster for PaLEnTIR, as its objective plot settles within 100 TREGS iterations.

3.4 Experimental evaluation

We explore the utility of PaLEnTIR using a variety of linear and nonlinear forward models, particularly for problems where the data are limited. All experiments require that we use a discrete form of the forward model developed in section 2.1. To keep the discussion simple, we use quadrature rules for all integral operators and finite differences if the forward operator involves a PDE. We could use other methods of discretization, but these assumptions permit a straightforward extension of the continuous to discrete notation for purposes of this work; that is, the vector of unknowns represents the values of the desired function $f(\mathbf{r};\mathbf{p})$ at a finite set of grid points. Specifically, let $\mathbf{r}_i$, for $i = 1,\ldots,N_{pts}$, denote a discrete set of 2D or 3D spatial grid points in Ω. We define the N_{pts}-vector $\mathbf{f}(\mathbf{p})$ as $[\mathbf{f}(\mathbf{p})]_i = f(\mathbf{r}_i;\mathbf{p})$. In this discrete case, $\mathcal{M}(\mathbf{f}(\mathbf{p}))$ represents the measured data for all sources given the discrete values $f(\mathbf{r}_i;\mathbf{p})$. The discrete forward model thus becomes

$$\mathbf{d} = \mathcal{M}(\mathbf{f}(\mathbf{p})) + \mathbf{w}. \tag{3.11}$$

We evaluate the performance of the PaLEnTIR model against RBF PaLS [21] and L2-Total Variation (TV) regularization methods. For the TV method, we employ `TVReg` from [146]. A notable advantage of PaLEnTIR over pixel-based approaches like TV is that it does not require a regularization parameter. In contrast, the TV method relies on selecting an appropriate regularization parameter, which we determine using the L-curve technique [147].

RBF PaLS is formulated with $\lceil \log_2 N \rceil$ level-sets for N contrast levels, meaning the number of level-sets is directly tied to the number of distinct contrasts in the image. This requires prior knowledge of contrast levels, which limits RBF PaLS's application in scenarios where the number of contrasts is uncertain. Without accurate information about contrast levels, RBF PaLS may fail to capture image variations correctly. For the purposes of this work, we assume the exact number of contrast levels is known and formulate the RBF

PaLS model accordingly. In contrast, PaLEnTIR eliminates the need for prior knowledge of contrasts by always using a single level-set, regardless of how many contrast levels exist. This feature makes PaLEnTIR more flexible, especially in cases where estimating the number of contrast levels is difficult or infeasible.

In terms of parameterization, RBF PaLS and PaLEnTIR differ in how they define and parameterize the level-sets. In RBF PaLS, each RBF requires four parameters in 2D (or five in 3D) to define its shape. For PaLEnTIR, each ABF requires three parameters in 2D (or seven in 3D). However, the key difference lies in the number of level-sets each method requires. PaLEnTIR is always formulated with a single level-set, so for a parametric level-set function with N basis functions, PaLEnTIR uses $3N$ parameters in 2D (or $7N$ in 3D). In contrast, for RBF PaLS formulated with M level-sets, where each level-set is modeled with N basis functions, RBF PaLS requires $4MN$ parameters in 2D (or $5MN$ in 3D). For PaLEnTIR, the initial values of α_j are set to 0.001, β_j and γ_j are set to 0.0001 for all j from 1 to N. For RBF PaLS, the initial values of α_j are set to 0.001, β_j are set to 10, initial RBF PaLS centers $\boldsymbol{\chi}_j \in \mathbb{R}^d$, are set to X, Y coordinates of an equally spaced $N_b \times N_b$ grid for $N_b = \sqrt{N}$ in 2D, and $N_b \times N_b \times N_b$ grid for $N_b = \sqrt[3]{N}$ in 3D.

Contrast levels are handled differently in each method. In RBF PaLS, the property values f_O and f_B are not part of the parameterization and are estimated separately from the PaLS parameters. In this work, we employed an alternating minimization approach, where the RBF PaLS parameters and property values are iteratively fixed and solved for until the algorithm converges. PaLEnTIR directly parameterizes the contrast values within the overall optimization, solving for them simultaneously with the shape parameters. The total number of unknowns for PaLEnTIR, including both shape and contrast parameters, is $5N$ in 2D (or $9N$ in 3D). For RBF PaLS, the total number of unknowns is $4MN$ shape parameters in 2D (or $5MN$ in 3D) along with 2^M property values. Compared to TV regularization, which treats each pixel as an unknown, both PaLEnTIR and RBF PaLS offer a significant reduction in the number of unknowns, independent of image resolution, making them more

computationally efficient for large-scale problems. For PaLEnTIR, the vectors p_L and p_H are initialized to the contrast limits, set to all zeros and ones in this work, respectively. In RBF PaLS, the property values are initialized as 2^M evenly spaced samples within the range defined by the contrast limits, 0 and 1.

We show results for the methods and compare (if ground-truth is available) them based on the following metrics: SSIM, MSE, SNR (dB), and PSNR (dB). The Structural Similarity Index (SSIM) was developed to replicate the human visual perception system [148]. A value closer to 1 indicates that the two images are very similar, whereas a value closer to -1 indicates the opposite. Mean Squared Error (MSE) is a commonly used metric to quantify the difference in the values of corresponding pixels between the sample and the reference images. Signal-to-Noise Ratio (SNR) measures the level of the desired signal relative to the level of noise. Peak Signal-to-Noise Ratio (PSNR) quantifies the image quality by comparing the peak level of a signal to the level of noise. The scores and the number of parameters of each method are presented in separate tables for each problem, with the better scores for each metric highlighted.

3.4.1 Deconvolution

Deconvolution is a linear inverse problem that recovers a desired signal from its convolution with a filter or a distortion function associated with an instrument or the physics of the problem. The discrete forward model is defined as in (3.11) with $\mathbf{f}(\mathbf{p})$ as in (3.1), giving the forward mapping $\mathcal{M}(\mathbf{f}(\mathbf{p})) = \mathbf{A}\mathbf{f}(\mathbf{p})$. The matrix $\mathbf{A}$ comes from discretizing an integral equation that represents convolution.

In this deconvolution experiment, we aim to reconstruct a 256×256 image, shown in Figure 3.5a, from input data that has been both blurred and corrupted by noise. The data is generated by convolving the true image with a 5×5 rotationally symmetric Gaussian low-pass filter using the MATLAB function `fspecial`, followed by contamination with 10% additive Gaussian white noise, resulting in a data SNR of 22dB (as shown in Figure 3.5b).

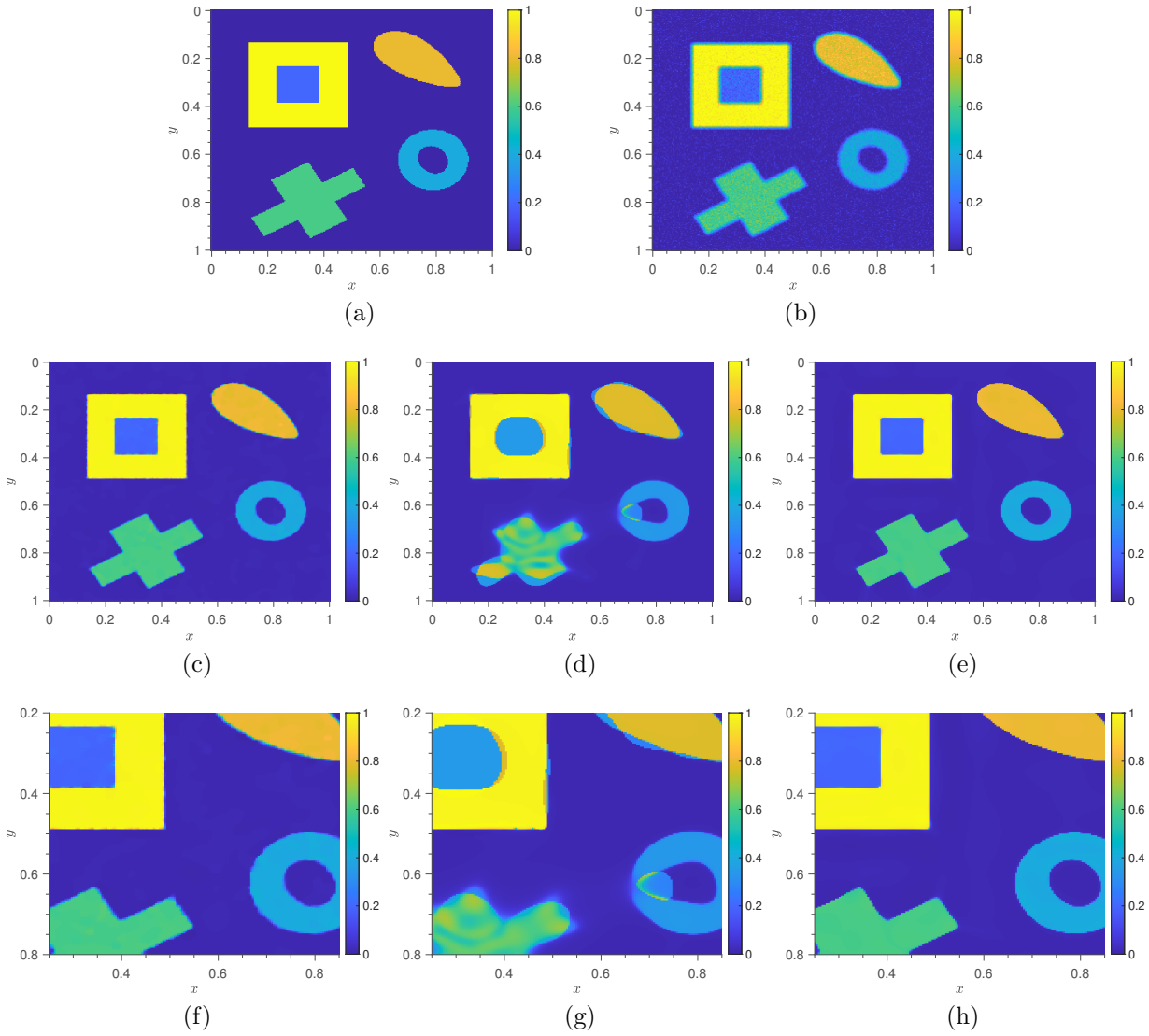


Figure 3.5: Deconvolution experiment results: (a) Reference image; (b) Blurred and noisy input image; (c) TV reconstruction; (d) RBF PaLS reconstruction; (e) PaLEnTIR reconstruction. Zoomed-in images of (f) TV reconstruction, (g) RBF PaLS reconstruction, and (h) PaLEnTIR reconstruction.

The goal is to assess the performance of PaLEnTIR, RBF PaLS, and TV regularization in recovering the original image. Figures 3.5c-3.5e display the reconstructions using TV, RBF PaLS, and PaLEnTIR, respectively. Zoomed-in versions of these reconstructions are provided in Figures 3.5f-3.5h.

PaLEnTIR uses 121 (11×11) basis functions with a total of 605 parameters, while RBF PaLS employs three level-set functions, resulting in 363 ($11 \times 11 \times 3$) basis functions and

Method	Unknowns	PSNR	SNR	SSIM	MSE
TV	65536	30.4	22.3	52.0e-02	92.2e-05
RBF PaLS	1460	22.8	14.8	47.7e-02	53.0e-04
PaLEnTIR	**605**	**31.5**	**23.5**	**53.1e-02**	**71.3e-05**

Table 3.1: Performance metrics for the deconvolution experiment: Number of unknowns, PSNR, SNR, SSIM, and MSE scores for the TV, RBF PaLS, and PaLEnTIR methods.

1452 parameters as well as 8 property values to be estimated. TV regularization, in contrast, operates on the pixel level with 65,536 unknowns. Despite the vast difference in parameter count, PaLEnTIR demonstrates exceptional performance by accurately recovering objects with smooth, piecewise constant contrast levels and sharp boundaries. PaLEnTIR's performance, as reflected in metrics like PSNR, SNR, SSIM, and MSE (Table 3.1), consistently surpasses that of both RBF PaLS and TV regularization.

TV regularization successfully recovers smooth contrast levels but suffers from slight blurring at object boundaries. This blurring is evident in the zoomed-in Figure 3.5f. While TV performs quite well in terms of PSNR and SSIM, PaLEnTIR's sharper boundaries (see Figure 3.5h) and smoother regions ultimately give it the performance edge.

RBF PaLS struggles the most as seen by the scores shown in Table 3.1 in this experiment. Despite using three level-sets, RBF PaLS fails to accurately capture the contrast levels of both the inner square and some parts of the rotated plus sign object, as shown clearly in the zoomed-in Figure 3.5g. Furthermore, RBF PaLS fails to represent geometric details of the objects. The inner blue square appears almost circular rather than a square. The rotated plus sign object exhibits overly rounded, poorly produced shapes. The RBF PaLS produces rough and inaccurate boundaries for the orange object on the top right as well. The zoomed-in Figure 3.5g shows the wavy blue surface at the boundary of the object. Also the recovery of the donut-shaped object does not reflect the true shape of the object. The inner boundary of the the object is not circular as in the reference image and there is a large artifact on the object.

Local sensitivity analysis was performed using the One-at-a-Time (OAT) technique [149]

to demonstrate that selecting the fixed parameters for PaLEnTIR is straightforward and much easier compared to the challenge of tuning the regularization parameter for TV regularization. Specifically, we assessed the robustness of PaLEnTIR with respect to the four fixed parameters: μ (a parameter controlling the maximum area of ABF ellipses), w (a parameter controlling the steepness of the transition function), c (the c-level-set), and N (the number of basis functions).

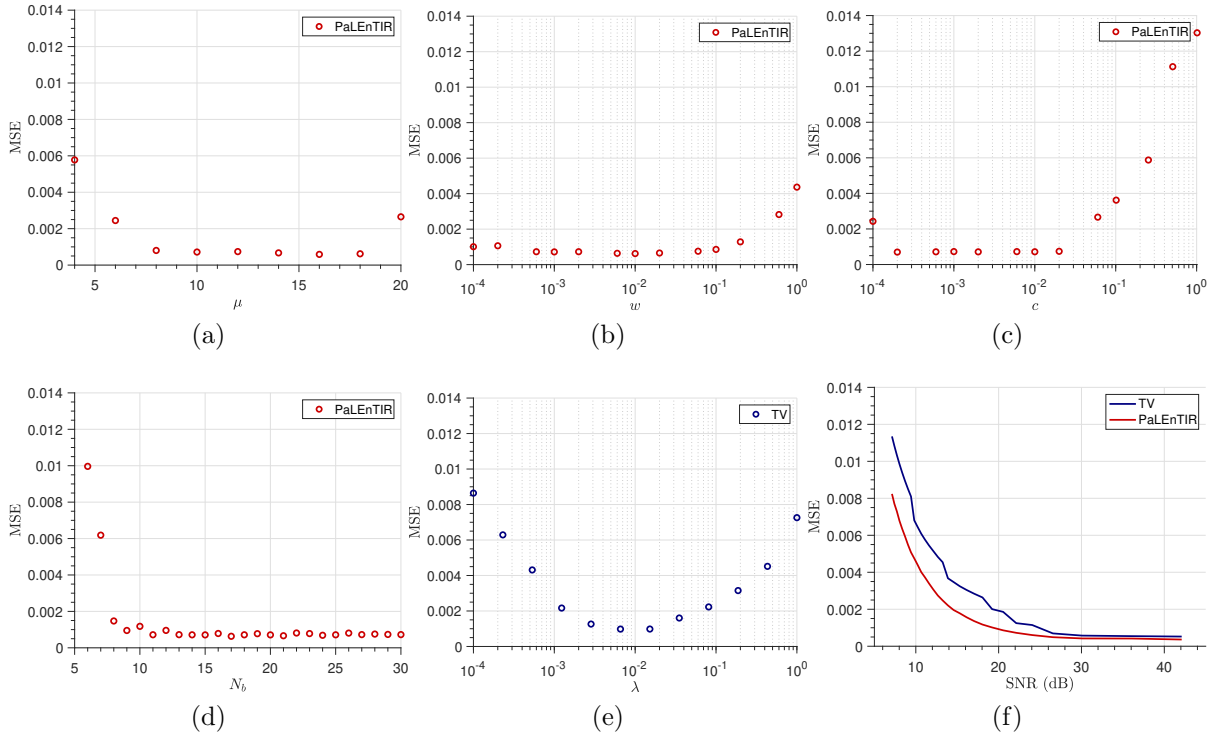


Figure 3.6: Sensitivity and stability analysis: MSE between the true image and PaLEnTIR reconstructions as a function of (a) the parameter μ, which controls the maximum area of ABF shapes; (b) the parameter w, which controls the steepness of the transition function; (c) the parameter c, which defines the c-level-set; and (d) N_b, the square root of the number of ABFs in PaLEnTIR. (e) MSE between the true image and TV reconstructions as a function of the TV regularization parameter λ. (f) MSE of TV and PaLEnTIR reconstructions across different data SNR values (dB). The blue line represents TV reconstructions, and the red line represents PaLEnTIR reconstructions.

Figures 3.6a–3.6d present the mean squared error (MSE) between the true image and PaLEnTIR reconstructions as a function of the parameter values. Figure 3.6a shows the MSE as a function of the μ. The MSE remains relatively unchanged for values of μ ranging

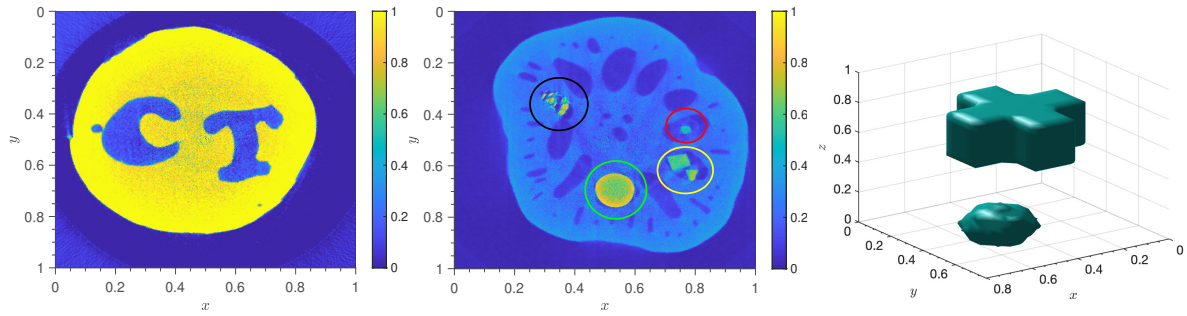
from 8 to 18. Note that μ controls the maximum area of the ABF ellipses. When μ is set to 8, a single ABF ellipse can take up to 23% of a unit square, whereas when μ is set to 18, a single ABF ellipse can take up to 4% of a unit square. Figure 3.6b shows the MSE plot as a function of w in log scale. The MSE remains relatively unchanged for values of w ranging from 10^{-4} to 10^{-1}. Similarly, Figure 3.6c shows the MSE as a function of c in log scale, with negligible variation observed and staying below 2×10^{-3} for values of c ranging from 2×10^{-4} to 2×10^{-2}. Figure 3.6d shows the MSE as a function of square root of the number of ABFs, $N_b = \sqrt{N}$. The MSE decreases until the number of ABFs reaches above 100, after which it converges, showing no further noticeable change with an increase in the number of basis functions.

For comparison, an additional sensitivity analysis was conducted for the TV method with respect to it's regularization parameter, λ. Figure 3.6e presents the MSE as a function of λ in log scale. The TV method is significantly sensitive to variations in λ, with the MSE increasing markedly as λ deviates from the optimal value near 10^{-2}.

In addition to the sensitivity analysis with respect to the fixed parameters, a stability analysis was conducted to compare PaLEnTIR and TV regularization in the presence of noise. Figure 3.6f shows the MSE versus data SNR (dB) plots for both PaLEnTIR and TV. These plots were generated by gradually increasing the amount of additive Gaussian white noise in the input data for each simulation. The regularization parameter for the TV method was carefully selected for each simulation using the L-curve method, whereas PaLEnTIR's fixed parameters remained unchanged throughout the simulations. The results demonstrate that PaLEnTIR consistently maintains lower MSE across varying data SNRs compared to TV. Furthermore, the gap between the MSE values of the two models widens as the noise in the data increases.

3.4.2 X-ray Computed Tomography

X-ray computed tomography (CT) is typically well-approximated as a linear problem and is one of the most common and well-known methods for medical imaging applications. The forward model again takes the form of a matrix vector product, i.e., $\mathcal{M}(\mathbf{f}(\mathbf{p})) = \mathbf{A}\mathbf{f}(\mathbf{p})$ where $\mathbf{A}$ denotes a (discrete) Radon transform, and the vector $\mathcal{M}(\mathbf{f}(\mathbf{p}))$ denotes the vectorized form of the sinogram data. In this work, we explore the performance of PaLEnTIR on both two and three dimensional CT problems, focusing on limited-angle angle and/or sparse view reconstructions.



(a) (b) (c)

Figure 3.7: Reference images for the X-ray CT experiments: (a) Thin slice of Edam cheese carved with "CT" letters; (b) Lotus root filled with various objects; (c) 3D objects in the shape of a plus sign and a sphere.

In the first X-ray CT experiments, we utilize an open-access dataset of tomographic X-ray data featuring a carved cheese specimen [150]. The dataset does not specify any noise-level approximations, and because a ground truth image is unavailable, quantitative performance metrics cannot be provided. Figure 3.7a shows the reference image, which is produced by applying a pixel-wise adaptive low-pass Wiener filter to the high-resolution filtered back-projection (FBP) reconstruction, computed from the provided full-angle 360-projection sinogram in the dataset. A thin slice of Edam cheese has been carved with "CT" letters. In particular, the curvy shapes of the carved letters make the target challenging for limited angle tomography applications.

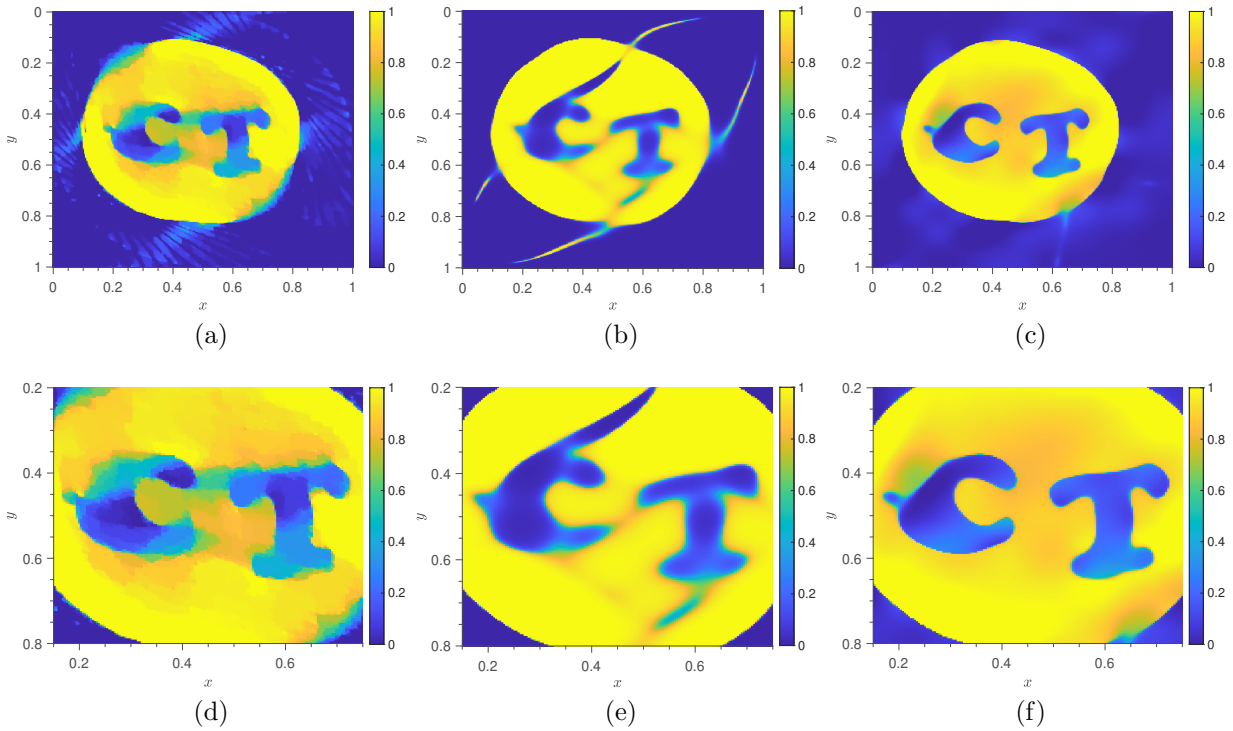


Figure 3.8: 2D X-ray CT experiment 1 results: (a) TV reconstruction; (b) RBF PaLS reconstruction; and (c) PaLEnTIR reconstruction. Zoomed-in images of (d) TV reconstruction, (e) RBF PaLS reconstruction, and (f) PaLEnTIR reconstruction.

In this experiment, we focus on a limited angle, limited view scenario, using 15 projections within a narrow angular range of 1°–90°. PaLEnTIR and RBF PaLS are both implemented using an 11×11 grid of basis functions. This is equivalent to 605 parameters for PaLEnTIR and 484 parameters plus 2 property values for RBF PaLS, less than 1% of the unknowns in pixel-based methods. The reconstructions of the methods are shown in Figures 3.8a-3.8c, with the zoomed-in Figures 3.8d-3.8f.

The TV recovery in Figure 3.8a, displays pronounced artifacts including blurred boundaries around the cheese and the carved letters. Additionally, noise artifacts appear in the background. Despite these limitations, TV is still recovers some fine details of the carved letters, such as the curvy shapes of the carved letters, seen in the zoomed-in Figure 3.8d. However, the blurring in the TV reconstruction significantly reduces the overall clarity of the recovered image.

RBF PaLS reconstruction, shown in Figure 3.8a, suffers from significant artifacts, including non-existent carvings around the letters, which make the letters difficult to distinguish. Additionally, the main body of the cheese is inaccurately recovered, with false extensions appearing. The shapes of carved letters C and T are poorly represented.

In contrast, PaLEnTIR, recovery is shown in Figure 3.8a, demonstrates superior performance with respect to other methods. The PaLEnTIR reconstruction accurately captures the intricate details of the carved letters as seen in the zoomed-in Figure 3.8f. The impact of the limited-angle of view sparse data is apparent in the rough surface of the PaLEnTIR recovery as well, yet the impact is relatively suppressed compared to the TV reconstruction.

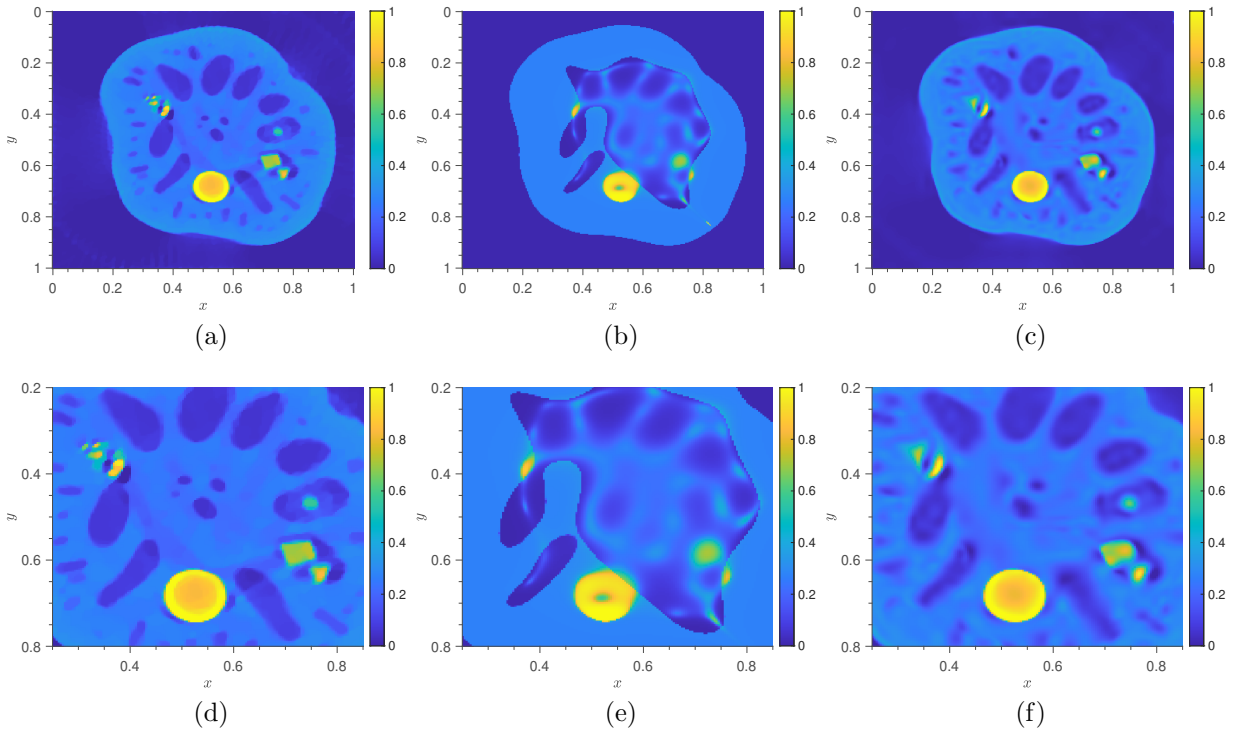


Figure 3.9: 2D X-ray CT experiment 2 results: (a) TV reconstruction; (b) RBF PaLS reconstruction; and (c) PaLEnTIR reconstruction. Zoomed-in images of (d) TV reconstruction, (e) RBF PaLS reconstruction, and (f) PaLEnTIR reconstruction.

We next reconstruct a lotus root filled with chemical objects using tomographic X-ray data from the open-access dataset [151]. The reference image in Figure 3.7b is produced by applying a pixel-wise adaptive low-pass Wiener filter to the high-resolution filtered back-

projection (FBP) reconstruction, computed from the full-angle 360-projection sinogram. The lotus root, with its starchy texture and varying hole sizes, serves as an ideal test subject for sparse-data CT applications due to its challenging structure. The filled objects are circled in the reference image, including a pencil (circled in red), chalk (circled in green), three ceramic pieces (circled in yellow), and match-heads (circled in black), contributing a range of shapes, sizes, and contrast levels.

We employ a sparse-view reconstruction setup with 120 projections over the full range of angles. The PaLEnTIR, RBF PaLS, and TV reconstructions are shown in Figures 3.9a-3.9c, and the zoomed-in Figures 3.9d-3.9f respectively. The TV method uses 65,536 pixels while PaLEnTIR uses 3,125 parameters with a 25x25 ABF grid. RBF PaLS is formulated with two level-sets and 25x25x2 RBFs, incorporating prior knowledge of four contrast levels, yielding 5,000 shape parameters plus 4 property values.

The TV reconstruction captures fine details such as small holes and the filled objects, though some blurring and mild X-ray artifacts remain. It provides the highest level of detail among the methods, with only minor imperfections. RBF PaLS, on the other hand, struggles significantly to recover the objects. In the RBF PaLS reconstruction, the lotus root's surface appears rough, with many details either missing or poorly defined, and the filled objects are indistinguishable, except for the chalk. The small holes near the root's boundary are completely missed, and larger features, such as the main holes, appear blurred and indistinct.

In contrast, PaLEnTIR produces a notably improved reconstruction compared to RBF PaLS. PaLEnTIR successfully captures key details, including the chalk, the match-heads, the ceramic pieces, and the pencil. Unlike RBF PaLS, PaLEnTIR can distinguish both the larger and smaller holes on the surface of the root. However, PaLEnTIR still struggles with some blurring and roughness on the root's surface. While it manages to capture small details that RBF PaLS misses, these details are not as sharp as in the TV reconstruction, highlighting a limitation of PaLS-like methods in resolving very fine features.

We now consider a 3D limited-view parallel beam tomography experiment using synthetic data. The true object is shown in Figure 3.7c. The input data is generated from 7 projections, each centered on the surface of one-eighth of a Lebedev sphere [152], resulting in sparse and limited-angle CT data. The data is corrupted with 1% additive Gaussian white noise. The experimental setup was constructed using `TVReg` [146].

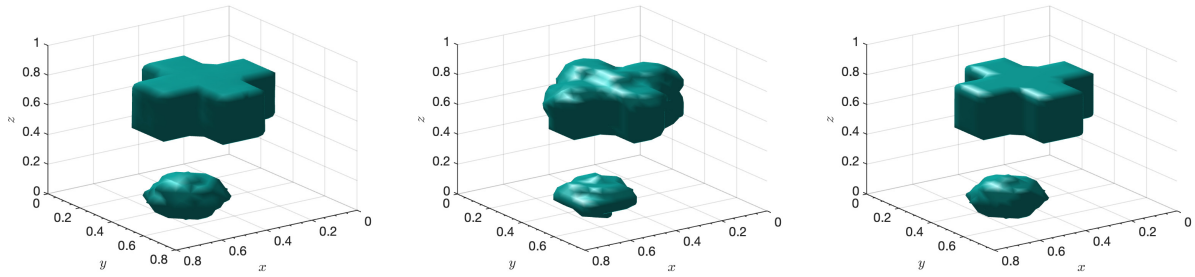
We compare the performance of PaLEnTIR, RBF PaLS, and TV regularization. PaLEnTIR and RBF PaLS used 216 basis functions centered on a $6 \times 6 \times 6$ grid, resulting in 1,944 unknowns for PaLEnTIR and 1,082 unknowns for RBF PaLS, compared to the 19,683 unknowns required by TV. Figures 3.10a–3.10c present the reconstructions produced by the three methods, while performance metrics are listed in Table 3.2.

PaLEnTIR demonstrates exceptional performance, as shown in Figure 3.10c. It not only matches the visual quality of the TV reconstruction but outperforms it in terms of key metrics such as PSNR, SNR, and MSE, with a marginally higher SSIM. The structural details of the objects are recovered accurately, with smooth surfaces and sharp edges, and the reconstruction shows no noise or artifacts.

In contrast, the RBF PaLS reconstruction, depicted in Figure 3.10b, performs significantly worse than both TV and PaLEnTIR. The edges of the object at the top are rough and overly rounded, lacking the sharpness seen in the other reconstructions. Additionally, the object on the bottom is poorly recovered compared to almost identical recoveries of TV and PaLEnTIR methods. These visual shortcomings are reflected in the substantially lower performance metrics.

Method	Unknowns	PSNR	SNR	SSIM	MSE
TV	19683	33.8	20.9	99.0e-02	41.7e-05
RBF PaLS	**1082**	18.4	5.5	83.1e-02	14.4e-02
PaLEnTIR	1944	**54.7**	**41.7**	**99.6e-02**	**34.1e-07**

Table 3.2: Performance metrics for the 3D X-ray CT experiment: Number of unknowns, PSNR, SNR, SSIM, and MSE scores for the TV, RBF PaLS, and PaLEnTIR methods.



(a) (b) (c)

Figure 3.10: 3D X-ray CT experiment results: (a) TV reconstruction; (b) RBF PaLS reconstruction; and (c) PaLEnTIR reconstruction.

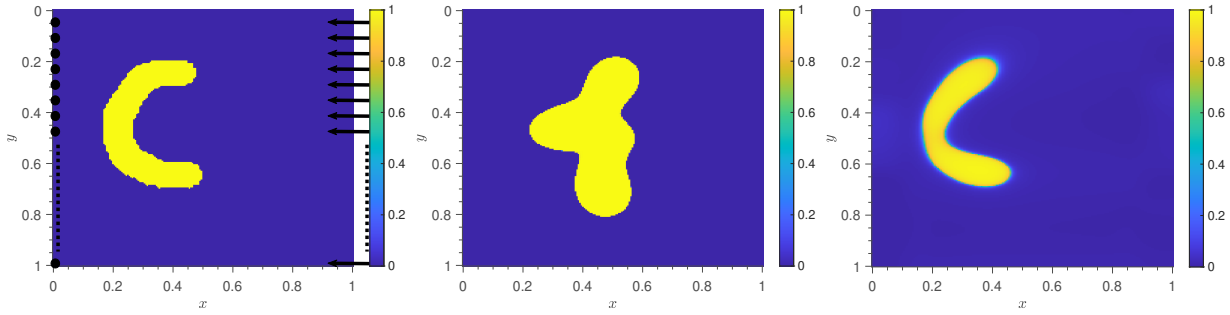
3.4.3 Diffuse Optical Tomography

Diffuse optical tomography (DOT) is a non-invasive, low-cost alternative for breast and brain imaging compared with X-ray and MRI [37]. In DOT, the tissue is illuminated with near-infrared light and the data, comprised of point measurements of diffused and partially absorbed photon fields, is collected external to the body. These measurements are used along with a mathematical model, typically a diffusion-absorption equation (posed in the frequency domain), to recover the optical absorption and (sometimes) scattering properties of the medium. Here we use such a model of the form

$$-\nabla \cdot (D(\mathbf{r})\nabla\eta(\mathbf{r})) + \kappa(\mathbf{r};\mathbf{p})\eta(\mathbf{r}) + \frac{\imath\omega}{\nu}\eta(\mathbf{r}) = g(\mathbf{r}), \tag{3.12}$$

where $D(\mathbf{r})$ represents the (here) known scalar diffusion at a point $\mathbf{r}$, $\kappa(\mathbf{r};\mathbf{p})$, the quantity for which we invert, represents the absorption as a function of space and the parameter vector $\mathbf{p}$, ω represents the modulation frequency of the light source, and ν represents the speed of light in the tissue. The sources are placed one grid point inside the medium [42], and the detectors are placed on the opposite boundary. On the boundaries where the sources and detectors are located, we have Robin boundary conditions; on the other boundaries, we use homogeneous Dirichlet boundary conditions, $\eta(\mathbf{r}) = 0$. For details, see [153, 154, 155]. The recovered absorption coefficient (and sometimes also the diffusion/scattering) can be

used to characterize the state of the tissue [37, 156]. Following, e.g., [157], we assume that



(a) (b) (c)

Figure 3.11: Diffuse optical tomography experiment results: (a) True anomaly with sources (right) and detectors (left); (b) RBF PaLS reconstruction; and (c) PaLEnTIR reconstruction.

the absorption coefficient can be modeled via (3.1). As shown in Figure 3.11a, we take the region to be imaged as a rectangle of size 1cm by 1cm, with $m_s = 32$ sources arrayed on the right side and $m_d = 32$ detectors on the left. As before, we let $\mathbf{f}(\mathbf{p})$ denote the discrete absorption image for a given parameter vector, that is, $[\mathbf{f}(\mathbf{p})]_i = f(\mathbf{r}_i, \mathbf{p})$ for any grid point $\mathbf{r}_i$. Assuming we collect data for all detectors when each source is active, we obtain a data vector $\mathbf{d}$ with $m = m_s \times m_d$ values for each modulation frequency ω.

Method	Unknowns	PSNR	SNR	SSIM	MSE
RBF PaLS	**486**	7.9	-3.1	**77.3e-02**	16.2e-02
PaLEnTIR	605	**18.3**	**7.2**	55.4e-02	**14.8e-03**

Table 3.3: Performance metrics for the diffuse optical tomography experiment: Number of unknowns, PSNR, SNR, SSIM, and MSE scores for the RBF PaLS, and PaLEnTIR methods.

The input-output map from sources to detectors [158] (also called the *transfer function*) as a function of $\mathbf{p}$ and ω, is given by

$$\mathbf{\Psi}(\mathbf{f}(\mathbf{p}); \omega) = \mathbf{C}^T \mathbf{A}(\mathbf{f}(\mathbf{p}); \omega)^{-1} \mathbf{B} \quad \in \mathbb{R}^{m_d \times m_s}, \tag{3.13}$$

where $\mathbf{B} \in \mathbb{R}^{n \times m_s}$ represents m_s sources, n is the total number of voxels or grid points, $\mathbf{A}(\mathbf{f}(\mathbf{p}); \omega) \in \mathbb{R}^{n \times n}$ represents the discretization of the diffusion-absorption equation, and $\mathbf{C}^T \in \mathbb{R}^{m_d \times n}$ simulates the measurement of outputs at m_d detectors. So, $\mathbf{A}(\mathbf{f}(\mathbf{p}); \omega)\mathbf{X} = \mathbf{B}$

represents the discretized PDE that relates photon fluence/flux at grid points to the sources [42]. The DOT inverse problem is then specified by the forward mapping $\mathcal{M}(\mathbf{f}(\mathbf{p});\boldsymbol{\omega}) = \text{vec}(\boldsymbol{\Psi}(\mathbf{f}(\mathbf{p});\boldsymbol{\omega}))$, the vectorization of the transfer function outputs for a vector of frequencies $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_{m_f}]$. Given a vector of measured data (with additive noise) $\mathbf{d}$, we solve for $\mathbf{p}$ by minimizing $\frac{1}{2}\|\text{vec}(\boldsymbol{\Psi}(\mathbf{f}(\mathbf{p});\boldsymbol{\omega})) - \mathbf{d}\|_2^2$. Note that regularization is provided implicitly by the parameterization.

We present the results of the DOT experiment in Figures 3.11a-3.11c, with the true anomaly depicted in Figure 3.11a. This experiment focuses on assessing the quality of shape reconstruction using the new PaLEnTIR parameterization and comparing it to the RBF PaLS method. The DOT problem, known for its severely ill-posed nature—due to the limited number of sources and detectors, along with the highly diffuse forward model—poses a significant challenge for accurate reconstructions. In this context, shape-based, parameterized methods like PaLEnTIR are particularly valuable for problems with well-defined boundaries.

We introduced 1% additive Gaussian white noise to the data. Both PaLEnTIR and RBF PaLS used $11^2 = 121$ basis functions. Table 3.3 summarizes the performance metrics and the number of parameters for both methods.

The PaLEnTIR reconstruction, shown in Figure 3.11c, captures both the shape and location of the anomaly. While there is some blurring and rounding of edges, the overall geometry and structure of the anomaly are well-preserved. This is notable given the highly diffusive nature of the problem and the presence of noise in the data.

In contrast, the RBF PaLS reconstruction, shown in Figure 3.11b, fails to accurately recover the anomaly's shape and location. Although RBF PaLS achieves a higher SSIM score compared to PaLEnTIR, this improvement is largely due to the background, which constitutes a significant portion of the image. A pixel-wise SSIM analysis reveals that PaLEnTIR under-performs in background where a small amount of noise is present, leading to a lower SSIM score. However, this drop is not reflective of its overall performance in the object re-

covery. In fact, PaLEnTIR significantly outperforms RBF PaLS in PSNR, SNR, and MSE, by providing an accurate reconstruction of the anomaly and highlighting its superior handling of this challenging problem.

3.5 Conclusion and Future Work

In this chapter, we redefined the parametric level-set method to improve the PaLS approach for solving inverse imaging problems involving piecewise constant contrasts. PaLEnTIR stands out, to the best of our knowledge, as the only level-set approach that employs only a single level-set function, irrespective of the number of contrast levels or their values. This enables PaLEnTIR to represent multi-contrast scenes accurately with fewer parameters than traditional level-set or pixel-based methods.

Replacing RBFs in PaLS with ABFs extends the range of shapes that can be represented by a single basis function, from circles to rotated ellipses. Compared to RBF PaLS and TV-regularization, PaLEnTIR provided superior reconstruction capability by capturing more detail, while still maintaining very few number of unknowns relative to traditional pixel-based methods. We also empirically showed that PaLEnTIR outperforms RBF PaLS in terms of numerical stability, as evidenced by improved condition numbers in both single and multiple basis function cases.

Looking ahead, while the current use of homogeneously distributed basis functions has proven effective, we anticipate further enhancements by incorporating an adaptive refinement strategy, akin to strategies proposed in [159, 160], which place additional basis functions in regions of higher geometric complexity. We acknowledge that in this work we do not share the computing times due to the fact that TREGS is currently available in MATLAB. A computing time comparison is recommended to be considered as a future work when an optimized code for TREGS is released in the future. Additionally, we recommend a focused investigation into the application of PaLEnTIR for uncertainty quantification. This could

require the development of prior models for PaLEnTIR parameters based on object priors, with a specific emphasis on leveraging these priors to quantify accuracy in tasks such as object localization and characterization.

Chapter 4

Enhanced Parametric Level-Set Methods for Tracking Evolving Objects

In this chapter, we consider monitoring spatial and temporal variations of substances or properties, such as nutrients [161], contaminants [162], in a media that is often large and inhospitable (e.g., forest soil, agricultural fields, volcanic regions [163]). Monitoring these substances generally requires multiple sensors embedded into a medium [164]. However, inserting a sensor network into such environments is not straightforward. Sensor deployments can disturb the medium under study. For example, many root quantification methods, such as ingrowth cores, soil cores, and pits are destructive [165]. Ensuring continued operation of sensors—through power supply, calibration, and data integrity—can also become cost-prohibitive on large spatial scales [166]. Moreover, embedding dense sensor networks into certain environments can be difficult or even impossible, such as volcanic regions, or regions underwater [167].

Consequently, due to the aforementioned difficulties and constraints, most of the real-world projects rely on limited sensor infrastructures to capture local measurements of sub-

stances [168]. However, a sparse sensor layout inevitably yields limited or noisy data. As discussed in Chapter 2, PaLS parameterization acts as a form of regularization. The implicit regularization of PaLS, on top of the topological advantages of a level-set method, motivated us to develop a PaLS approach for monitoring the spatial and temporal variations of substances in these type of problems.

Instead of tracking the exact representation of nutrients or substances, we primarily focus on recovering critical concentration levels of these substances. Critical concentration levels are specifically chosen thresholds that correspond to meaningful physiological or environmental states, such as the onset of toxicity, nutrient sufficiency, or deficiency thresholds [169]. Focusing on these critical thresholds simplifies interpretation, and contributes to clear decision-making. Recovering the continuous concentration profile before discretizing could provide more detailed information, but this typically requires significantly more data and computational resources, potentially reducing the method's efficiency [21]. Thus, we reduce the problem into a multi-contrast, piecewise constant formulation. As detailed in Chapter 3, the proposed model, PaLEnTIR, is specifically designed to efficiently recover such piecewise constant multi-contrast objects from limited and corrupted data. In this chapter, we adapt PaLEnTIR to account for the continuity of solutions to a diffusion process in both space and time, thereby effectively monitoring evolving property concentration levels over time.

4.1 Method

Our problem considers tracking objects evolving in time. Such systems are usually represented by partial differential equations, such as heat transfer or diffusion equations. For a well-posed problem for a partial differential equation (PDE), the solution depends continuously on the data given in the problem [170]. For the problems concerned in this chapter, we assume that the solution is smooth both in time and space. In the problem definition, the smoothness assumption in time is used to define the regularization term. We used the

smoothness assumption in space for the model definition. The rest of the section considers the problem and model definition of this work.

4.1.1 Problem definition

The problem definition for the time series reconstruction of a time-evolving object at N_T successive time points using PaLEnTIR takes the form

$$\min_{\mathbf{p}} \sum_{n=0}^{N_T} \|\mathcal{M}\left(f(\mathbf{r};\mathbf{p}_n, C_H, C_L)\right) - \mathbf{d}_n\|_2^2 + \lambda_1 \sum_{n=0}^{N_T-1} \Gamma(n, n+1) + \lambda_2 \sum_{n=1}^{N_T} \Gamma(n, n-1), \tag{4.1}$$

where

$$\Gamma(n_1, n_2) = \sum_{n=0}^{N_T} \|f(\mathbf{r};\mathbf{p}_{n_1}, C_H, C_L) - f\left(\mathbf{r};\mathbf{p}_{n_2}, C_H, C_L\right)\|_2^2, \tag{4.2}$$

and $\mathbf{p}^T := \begin{bmatrix} \boldsymbol{p}_1^T \dots \boldsymbol{p}_N^T \end{bmatrix}$. The mismatch error in (2.2) is replaced with the first term in (4.1) which is a sum of mismatch errors between model prediction and the data of time sequences. Based on the prior assumption of smoothness in time, the regularization term in (2.2) is replaced with L2 regularization between successive time points to ensure smoothness [171].

4.1.2 PaLEnTIR for tracking objects evolving in time

In this chapter, PaLEnTIR is reformulated to solve the problem defined in (4.1). In Chapter 3, PaLEnTIR estimates contrast parameters as part of the overall imaging problem. Those contrast values in PaLEnTIR represent the unknown contrast values of the objects. However, in this chapter, the contrast values represent the known concentration levels of a time-evolving substance. Therefore, this work does not require the estimation of the contrast parameters.

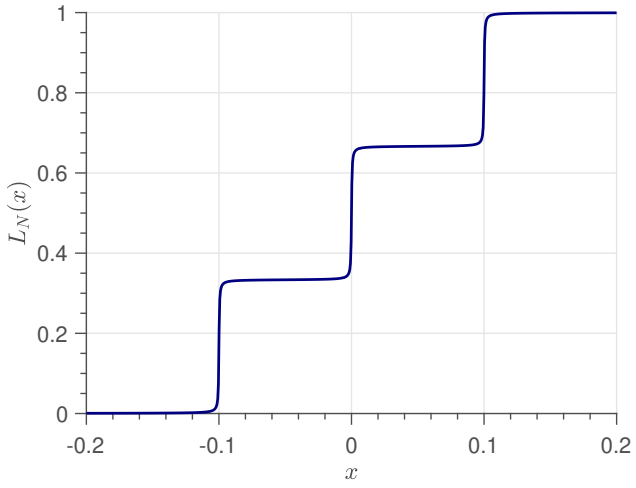
For tracking N different concentration levels of a substance, the new model is defined as

$$f(\mathbf{r};\mathbf{p}_n, C_H, C_L, n) = C_H L_N\left(\phi(\mathbf{r};\mathbf{p}_n)\right) + C_L\left(1 - L_N\left(\phi\left(\mathbf{r};\mathbf{p}_n\right)\right)\right), \tag{4.3}$$

where

$$L_N(x) = \frac{1}{N}\sum_{n=1}^{N} T(x - \delta_n) \tag{4.4}$$

is the *ladder function*, and $\mathbf{p}_n{}^T := \left[\boldsymbol{\alpha}_n{}^T \; \boldsymbol{\beta}_n{}^T \; \boldsymbol{\gamma}_n{}^T\right]$. The contrast parameters of PaLEnTIR in Chapter 3 are replaced with fixed scalars C_H and C_L, which now act as the lowest and highest concentration levels of interest. Ladder function is formed as a weighted summation of N transition functions, each located at equally spaced, fixed centers at $x = \delta_n$ for $n = 1, 2, 3 \ldots N$. Figure 4.1 illustrates $L_N(x)$ for three equally spaced steps, i.e., $N = 3$. The steps in the ladder function determine the intermediate concentration levels of interest, between C_H and C_L. The new approach can recover time-evolving multi-contrast recoveries (corresponding to different concentration levels) using only a single level-set function and does not require parameterizing the contrast coefficients as well.



(a)

Figure 4.1: The ladder function, L_N, for three equally spaced steps as $N = 3$.

We use the ladder function based on the smoothness assumption in space for the solutions of the problems in this work. Let $u(x, t_0)$ be the solution for our problem at time t_0. Suppose x_a and x_b are two spatial points where $f(x_a, t_0) > f(x_b, t_0)$. Then by the *Intermediate Value Theorem*, the function f attains all values between $f(x_a, t_0)$ and $f(x_b, t_0)$ for $x \in [x_a, x_b]$ [172]. Hence, when we add "steps" between C_H and C_L in the ladder function, based on

the *Intermediate Value Theorem*, the solution must pass through these intermediate steps in order, as long as the solution is a continuous function.

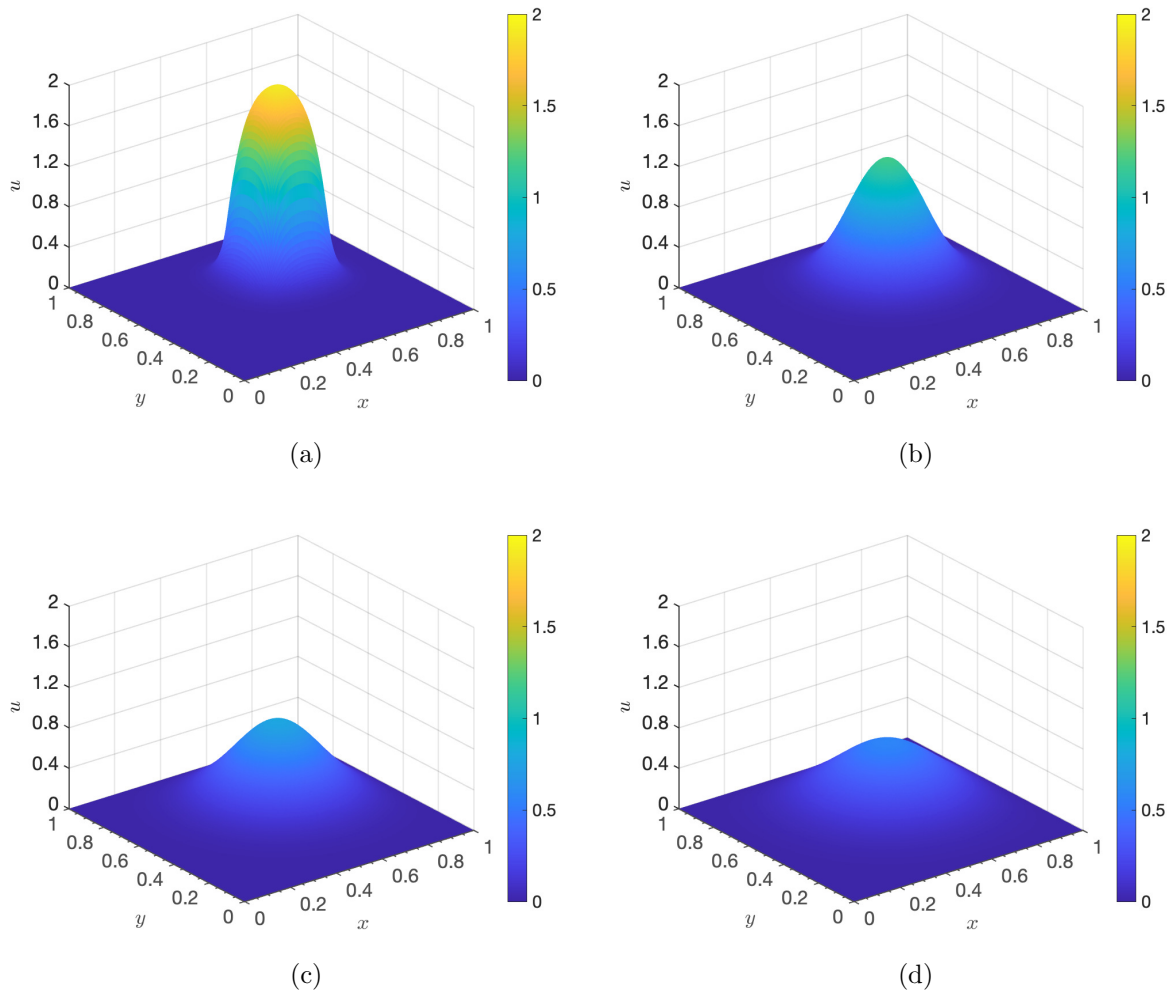


Figure 4.2: 3D representation of the property concentration $u(x, y)$ of a substance at $4th, 8th, 12th$, and $16th$ time-samples, corresponding to (a) $0.2th$,(b) $0.4th$,(c) $0.6th$,(d) $0.8th$ seconds.

4.2 Simulation setup

In this work, we evaluated the proposed method on a simulated diffusion data [173]. The diffusion coefficient was set to $0.1m^2/s$. The area of the region was set to $4m^2$. The system was initialized with the substance concentrated in a $0.25m^2$ region at the center of the

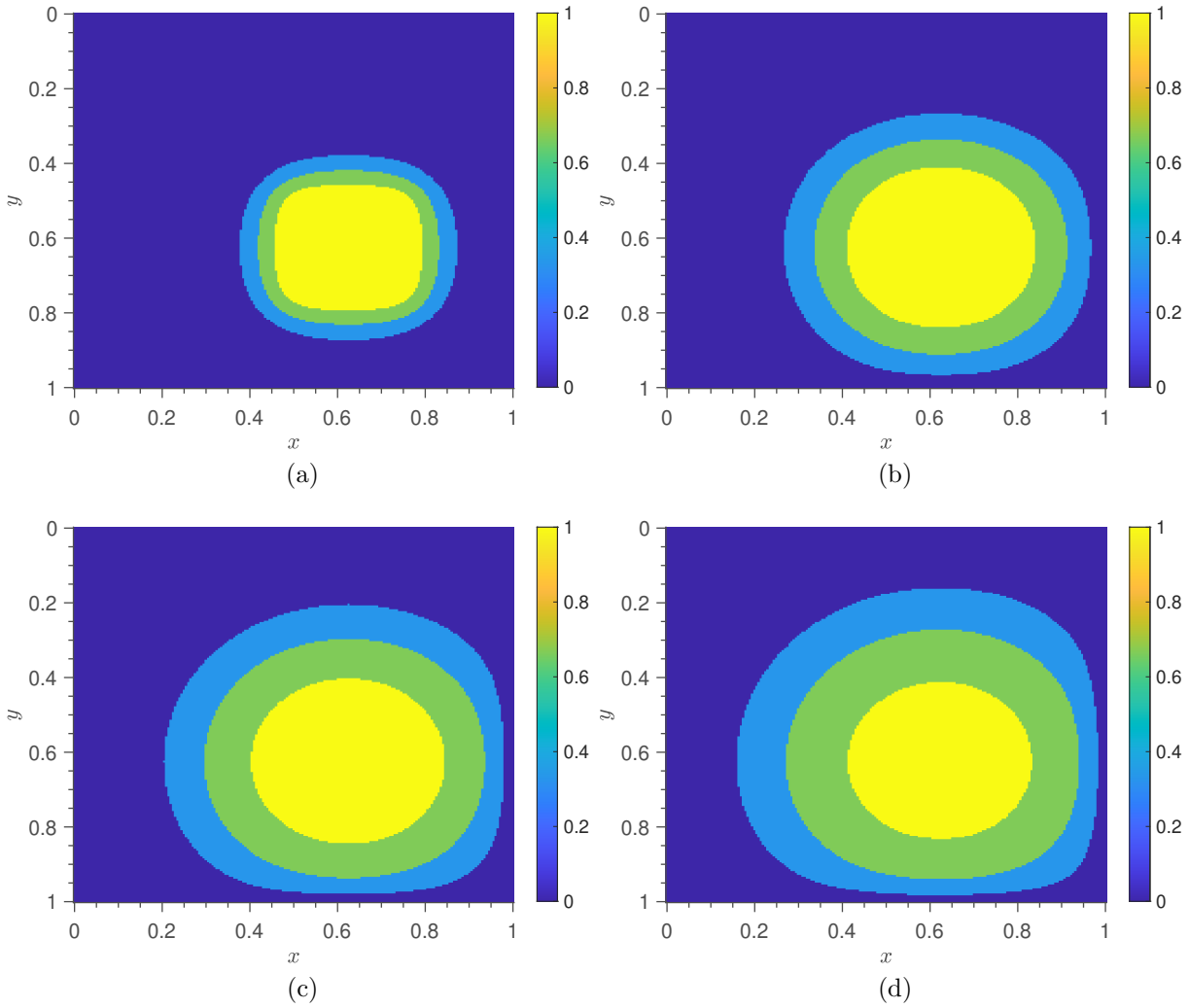


Figure 4.3: Multi-contrast piecewise constant approximation of the property concentration levels of a diffusing substance in a 2D region at $4th, 8th, 12th$, and $16th$ time-samples, corresponding to (a) $0.2th$,(b) $0.4th$,(c) $0.6th$,(d) $0.8th$ seconds.

medium with a fixed concentration level of $2kg/m^3$, and $0kg/m^3$ elsewhere. The total duration of the simulation took 1 second and sampled at $20Hz$. Figures 4.2a-4.2d show the 3D representation of the property concentration $u(x, y)$ of the substance for $4th, 8th, 12th$, and $16th$ time-samples, corresponding to $0.2th$, $0.4th$, $0.6th$, $0.8th$ seconds.

We reduce the problems into a multi-contrast piecewise constant recovery by tracking three concentration levels; 0.3, 0.1, $0.03kg/m^3$. The resultant multi-contrast piecewise constant approximations corresponding to $0.2th$, $0.4th$, $0.6th$, $0.8th$ seconds are shown in Figures

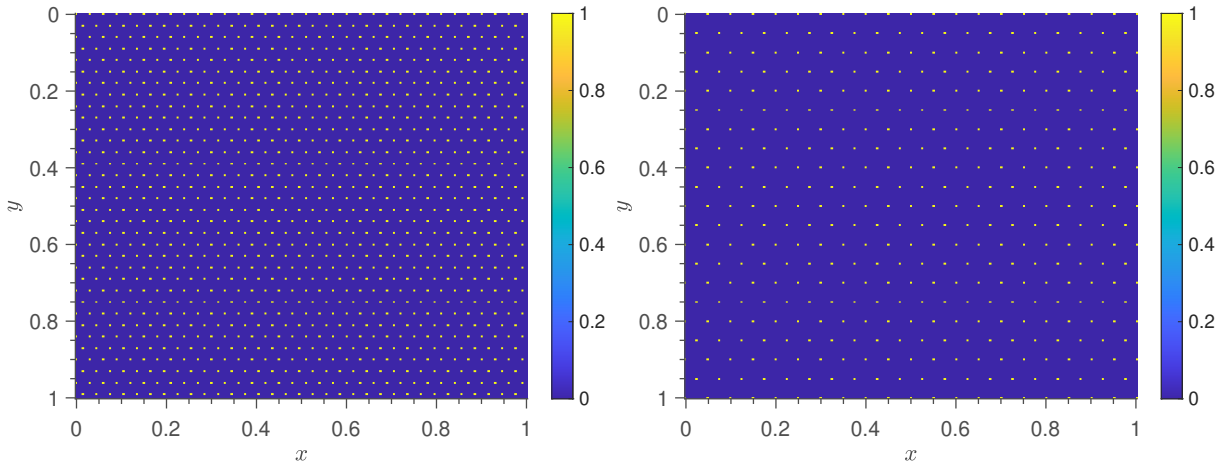


Figure 4.4: Sensor locations. (a) Locations of equally spaced 1139 sensors used in experiment 1. (b) Locations of equally spaced 431 sensors used in experiment 2.

4.3a-4.3b. For better visualization, the concentration levels are mapped into equally spaced contrast levels between 0 and 1. These images serve as the reference images in this experiments that we aim to recover from the sparse noisy data.

We conducted two separate experiments. In both experiments the sensor data is corrupted with independent, identically distributed additive Gaussian random variables, with variance such that the signal to noise ratio (SNR) of the corrupted image is 20 dB. In the first experiment, 1139 sensors are distributed homogeneously and equally spaced in the region. In the second experiment, we consider a sparser setup where input data is collected from 431 equally spaced sensors that are distributed homogeneously in the region. Sensor locations are shown in Figures 4.4a and 4.4b for the first and the second experiments respectively.

The number of steps in the ladder function is chosen as $N = 3$ since we have 3 contrast levels and the 0-level background. We consider 15 ordered time points in both experiments, corresponding to all the samples in between $0.2th$ to $1.0th$ seconds. Hence N_T in (4.1) is set to 15.

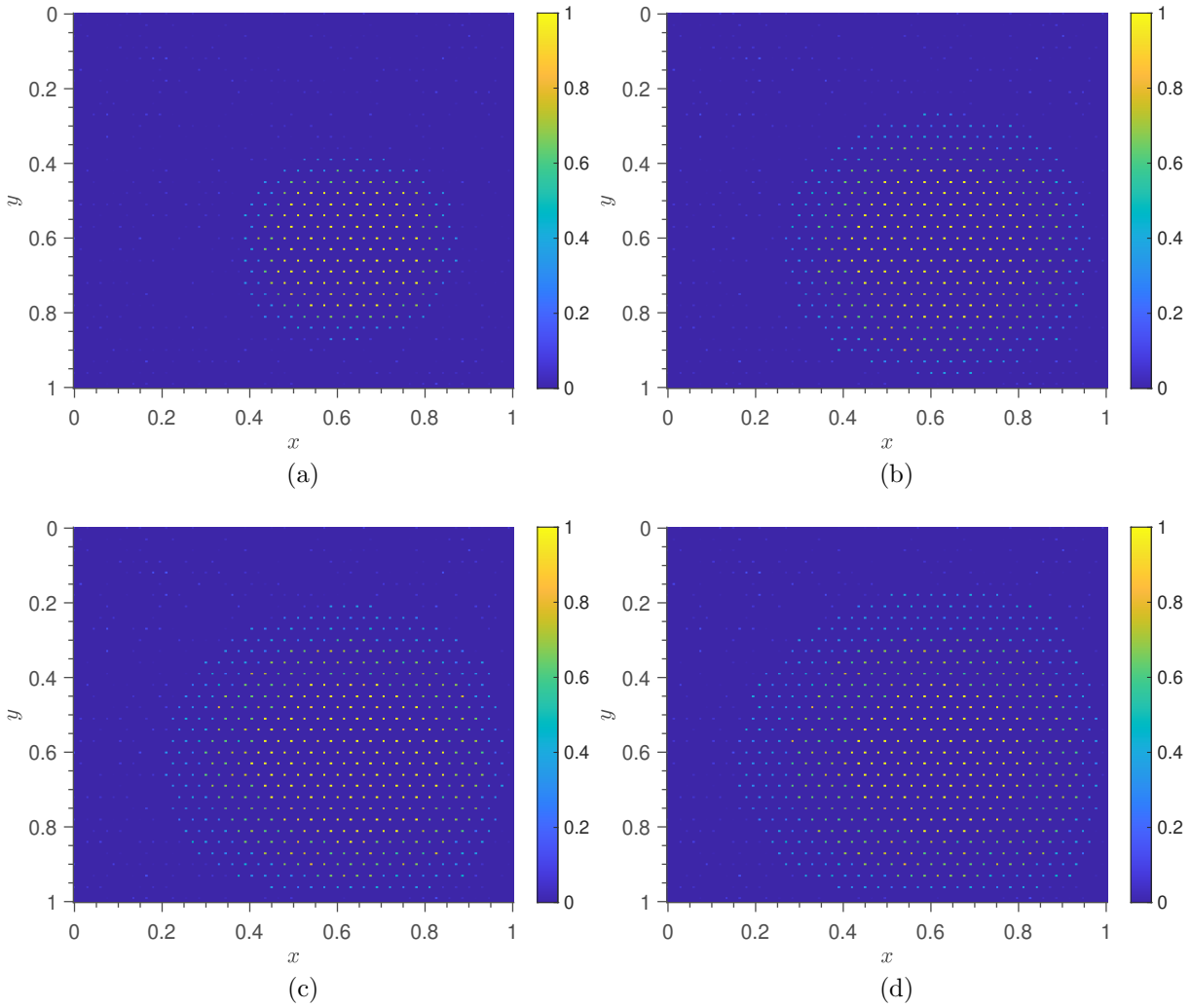


Figure 4.5: Experiment 1 sensor data of the diffusing substance in a 2D region at $4th, 8th, 12th$, and $16th$ time-samples, corresponding to (a) $0.2th$,(b) $0.4th$,(c) $0.6th$,(d) $0.8th$ seconds.

4.3 Results

4.3.1 Experiment 1

The noisy input data collected from the sensors at $0.2th$, $0.4th$, $0.6th$, $0.8th$ seconds in experiment 1, are displayed in Figures 4.5a-4.5d. Pixels located at the sensor locations show the sensor readings, and the rest of the pixels are set to zero.

The results of the first experiment are shown in Figures 4.6a-4.6d. We are displaying 4

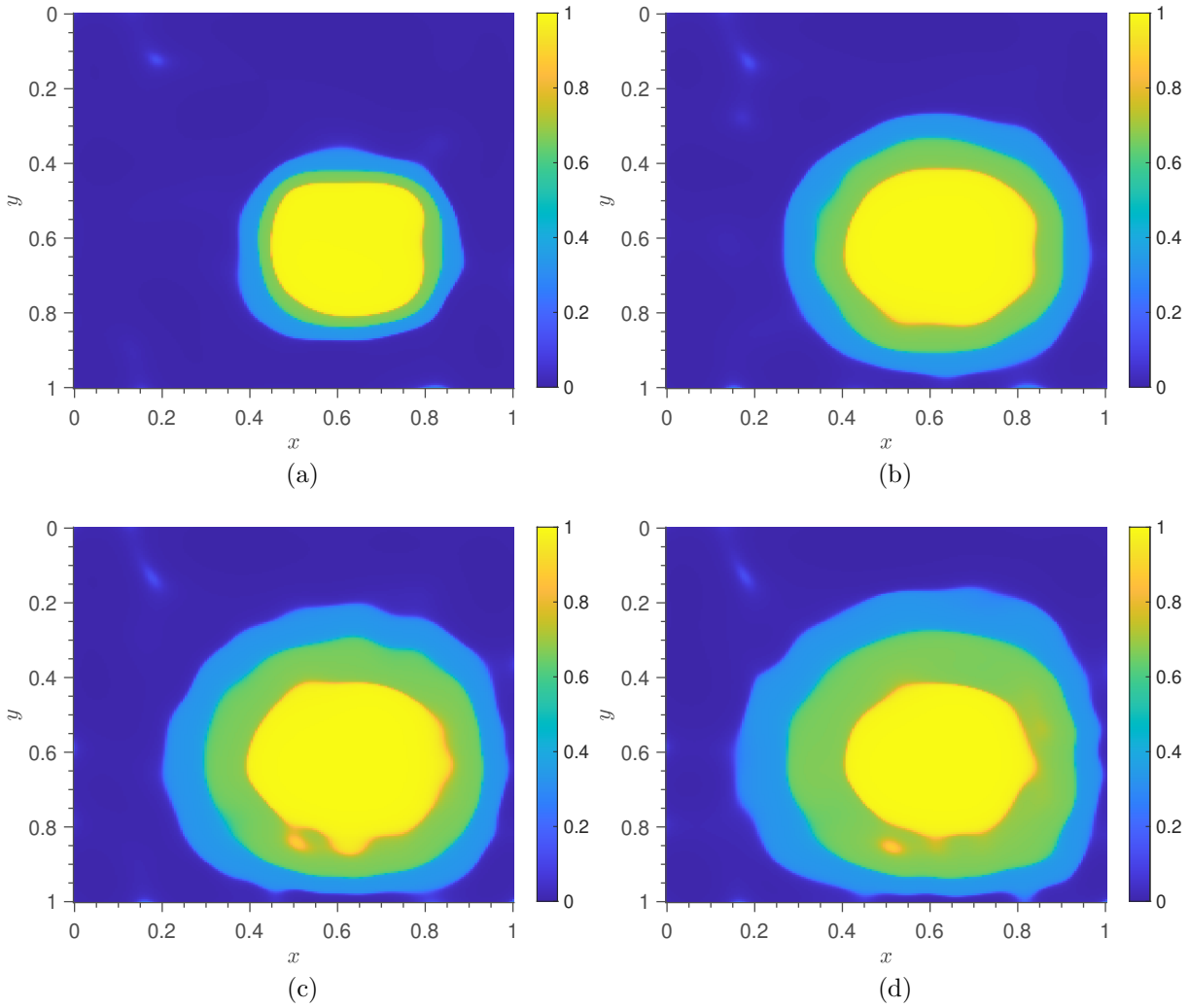


Figure 4.6: Multi-contrast piecewise constant approximation of the property concentration levels of a diffusing substance in a 2D region at $4th, 8th, 12th$, and $16th$ time-samples, corresponding to (a) $0.2th$,(b) $0.4th$,(c) $0.6th$,(d) $0.8th$ seconds.

Method	Unknowns	PSNR	SNR	SSIM	MSE
PaLEnTIR	5445	26.8	20.1	71.4e-02	21.0e-04

Table 4.1: Performance metrics for the diffusion experiment 1: Number of unknowns, PSNR, SNR, SSIM, and MSE scores for the PaLEnTIR method.

of 15 PaLEnTIR reconstructions corresponding to $0.2th$, $0.4th$, $0.6th$, $0.8th$ seconds. The performance metrics (averaged over 15 reconstructions) are shown in Table 4.1. Despite the sparsity of the problem and the presence of the noise, PaLEnTIR was able to recover all 4 of the contrasts, the order and the shape of the clustered objects accurately. We can observe

the impact of the limited and noisy data from the artifacts in the background and on the objects. Furthermore, we observe fluctuation in the object boundaries.

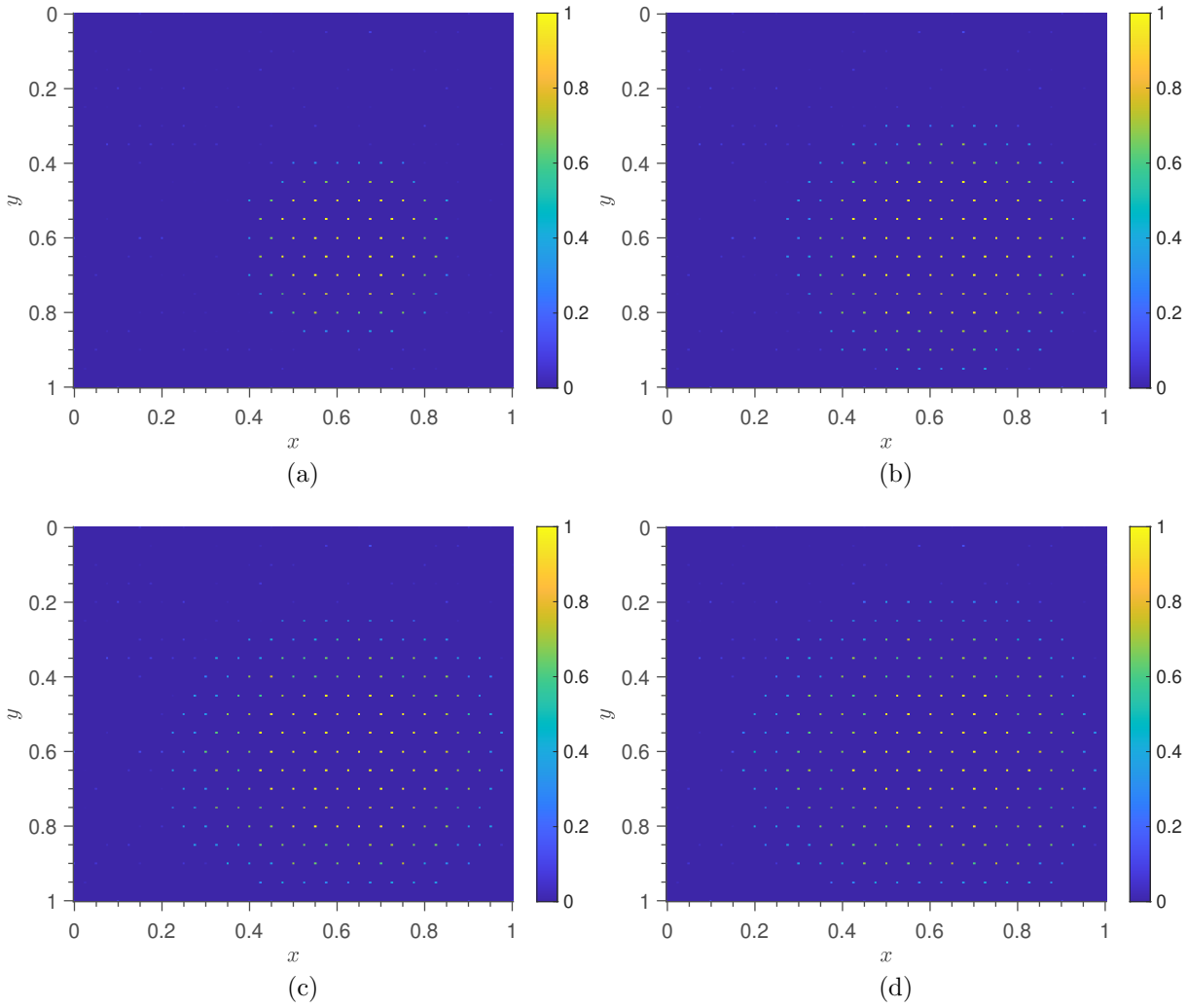


Figure 4.7: Experiment 2 sensor data of the diffusing substance in a 2D region at $4th, 8th, 12th$, and $16th$ time-samples, corresponding to (a) $0.2th$,(b) $0.4th$,(c) $0.6th$,(d) $0.8th$ seconds.

4.3.2 Experiment 2

The noisy input data measurements from the sensors at $0.2th$, $0.4th$, $0.6th$, $0.8th$ seconds in experiment 2, are displayed in Figures 4.7a-4.7d. Pixels at the sensor locations show the sensor readings, and the rest are set to zero.

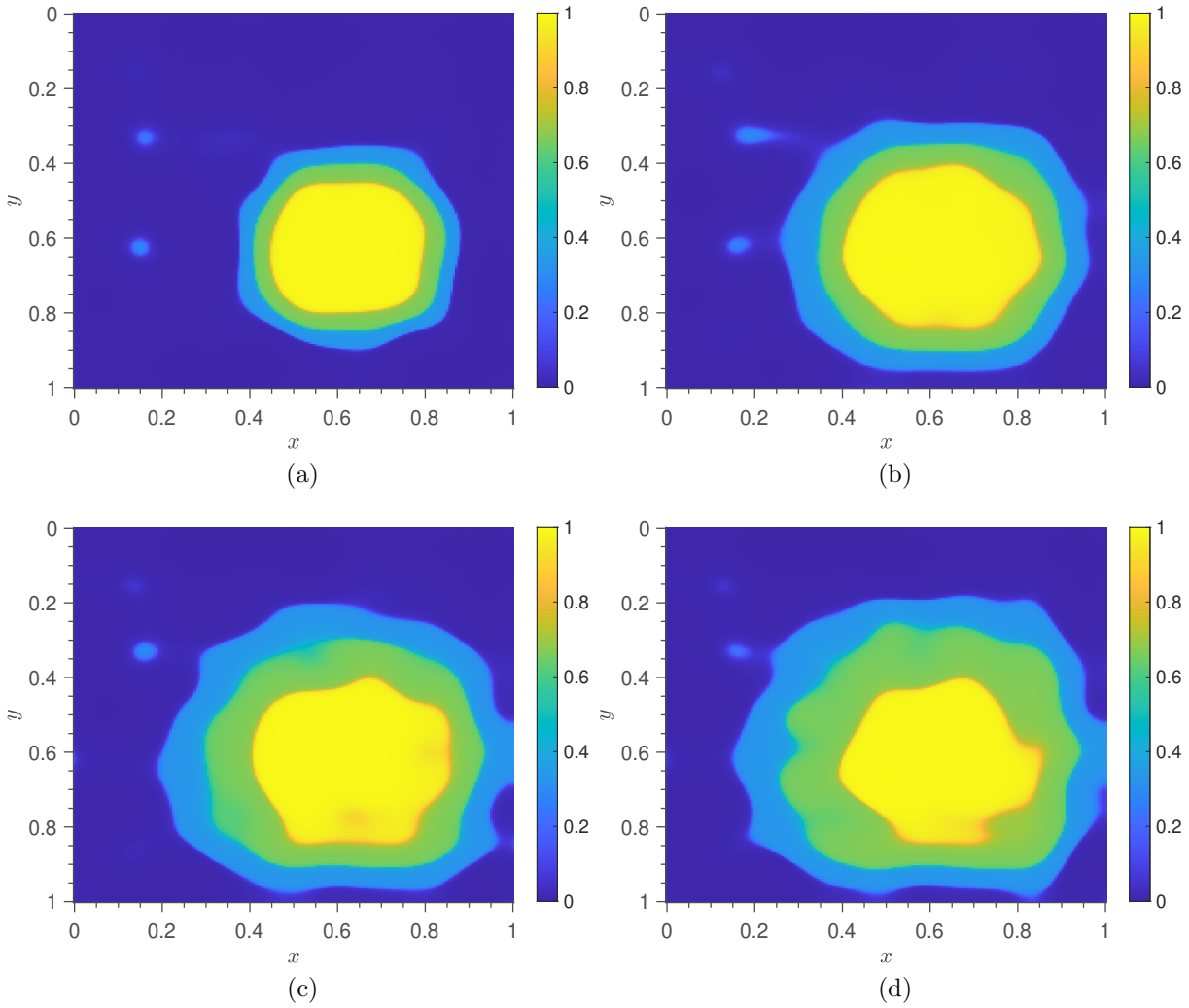


Figure 4.8: Multi-contrast piecewise constant approximation of the property concentration levels of a diffusing substance in a 2D region at $4th, 8th, 12th$, and $16th$ time-samples, corresponding to (a) $0.2th$,(b) $0.4th$,(c) $0.6th$,(d) $0.8th$ seconds.

Method	Unknowns	PSNR	SNR	SSIM	MSE
PaLEnTIR	5445	23.2	16.5	66.7e-02	21.0e-04

Table 4.2: Performance metrics for the diffusion experiment 2: Number of unknowns, PSNR, SNR, SSIM, and MSE scores for the PaLEnTIR method.

The results of the second experiment are shown in Figures 4.8a-4.8d. We are displaying 4 of 15 PaLEnTIR reconstructions corresponding to $0.2th$, $0.4th$, $0.6th$, $0.8th$ seconds. The performance metrics (averaged over 15 reconstructions) are shown in Table 4.2. Despite the increased sparsity of the problem and the presence of the noise, again, PaLEnTIR was able

to recover all 4 of the contrasts, the order and the shape of the clustered objects accurately. Similar to experiment 1, we can clearly observe the impact of the limited and noisy data in the reconstructions. This time, the impact of the noise and the limited availability of the data are much more present in the reconstructions. The noise artifacts in the background are stronger and larger compared to the first experiment results. Moreover, there is an increase in the strength of the fluctuations in the object boundaries.

4.4 Conclusion and Future Work

In this chapter, we considered an extension of the PaLEnTIR framework for tracking time-evolving objects. By leveraging the smoothness assumption in both space and time, we utilized a ladder function to efficiently represent multiple contrast levels with a single level-set function in clustered objects.

Through simulated diffusion experiments, our method demonstrated robust recovery of multiple concentration thresholds even in the presence of considerable noise and sparse sensor layouts. With as few as 431 sensors, the reconstructed piecewise constant representations closely matched the true concentration distributions.

In this work PaLEnTIR showed promising performance on time-evolving multi-contrast recovery. The results of this work motivate further investigation of the method across different scenarios. One immediate next step is to apply this new method to non-trivial forwards models and extend to 3D volumetric data (e.g., medical imaging sequences), where we expect to observe the benefits of a PaLS approach, such as the low order representation. In this work, parameters δ_n for $n = 1 \dots N$ are fixed scalars. In future work, there could be value in estimating them for certain problems. Another extension to this work could be introducing an adaptive strategy for placing the basis functions. While the current use of homogeneously distributed basis functions has proven effective, we anticipate further enhancements by incorporating an adaptive refinement strategy, akin to strategies proposed in [159, 160], which

place additional basis functions in regions of higher geometric complexity. We acknowledge that, this work does not investigate the impact of processing and computational limits of the method. A further investigation on the impact of N_T on the processing time help understanding the computational limitations of this approach. The experiments in this work were limited to simulated data. Validating our approach on real experimental data with comparing to the state of the art methods will be an essential next step in the future.

Chapter 5

AI-Driven Probabilistic Change Point Detection in High-Dimensional Time Series Data

Change point detection (CPD) is a critical research area in statistics and machine learning, and it is a vital task in time series analysis with applications across numerous fields [84]. Applications range from monitoring collective animal and crowd behaviors [88, 89] to finance [174], health monitoring [85, 86], and environmental analysis [87]. CPD aims to identify time instances at which the statistical properties of a data sequence shift abruptly. These shifts usually indicate major changes within underlying processes including rapid price movements in commodity markets [175], disease outbreak occurrences in specific regions [176], and sudden environmental condition changes [177]. Detecting and responding to these changes is essential because they create substantial effects on real-world systems.

This chapter analyzes how to detect change points in multi-agent systems including bird flocks, robotic swarms, and human social groups where collective behaviors result from environmental influences and interaction rules. Understanding and controlling collective systems depends on recognizing shifts in movement patterns and communication structures

which define transitions in collective behavior [178].

These systems have not only temporal dependencies but also dependencies among agents. Traditional CPD methods often struggle in this context, as they typically fail to scale effectively with increased dimensionality, or assume independence among dimensions to solve simpler problems with smaller dimensions [179].

The first case considers applying the CPD methods directly to the observations of multi-agent systems [88]. This is often well-suited for small multivariate time series problems of dimensions $d \times T$, where d represents the number of features and T is the observation duration. However, multi-agent systems generate higher-dimensional data structured as $N \times d \times T$, where N denotes the number of agents. This results in a huge dimensionality and complexity for traditional CPD methods.

For the second case, the first option is to make assumptions about the nature of time series data. Many CPD methods assume that the input data is independent and identically distributed (iid), which most of the time does not reflect the reality [180]. CPD methods with iid assumption do not capture the temporal dependencies, which is critical for effective CPD in time series data.

The second option is to make assumptions regarding the nature of agent interactions. In this case, multi-agent problems are divided into N sub-problems for every individual agent. However, changes in multi-agent systems often emerge from collective interactions rather than individual trajectories [88].

Finally, statistical and machine learning algorithms need datasets that maintain consistent entity associations (agent identifiers) through different time points. Entities entering or exiting the system or missing data create misalignment which disrupts the continuity of observational datasets [181].

In this work, we develop a robust CPD framework for high-dimensional, multi-agent time series that can handle (i) non-iid time series data, (ii) inter-agent dependencies, (iii) potentially large dimensionality of the problem, and (iii) missing or variable-sized inputs.

To address the non-iid concerns, we turn the time series data from all the agents at each time into a point cloud and model this cloud over time as a stochastic process. Directly modeling the data as a stochastic process results in a huge dimensionality especially as we discretize over time. Therefore, we apply cascading dimensionality reduction techniques to significantly reduce the dimensionality and then build a nonparametric model in the low dimensional space as the basis for detecting changes.

5.1 AI-Driven Probabilistic Change Point Detection

In this work, we propose a AI-driven probabilistic model for change point detection problems in high dimensional multi-agent time series problems. The architecture of the proposed CPD model is shown in Figure 5.1. The approach comprises four stages: data processing, modeling, dimensionality reduction, density estimation and CPD. In the remainder of this section, we describe the components of these three stages and their importance regarding this problem.

5.1.1 Data processing

In the proposed framework, time-series data are captured from N agents interacting within an environment. Each agent's data comprise d-dimensional measurements. The first stage of the proposed approach involves data processing. In statistical machine learning and deep learning pipelines data processing becomes essential because the quality and structure of data directly affect how well trained models perform [182]. Data processing works to clean and transform raw data into a structured format enabling machine learning algorithms to learn patterns and produce accurate predictions [183].

Z-score normalization stands as a widely utilized method for data processing in machine learning tasks [184]. While z-score normalization (standardization) is a widely used pre-processing technique that scales data to have zero mean and unit variance, it assumes that

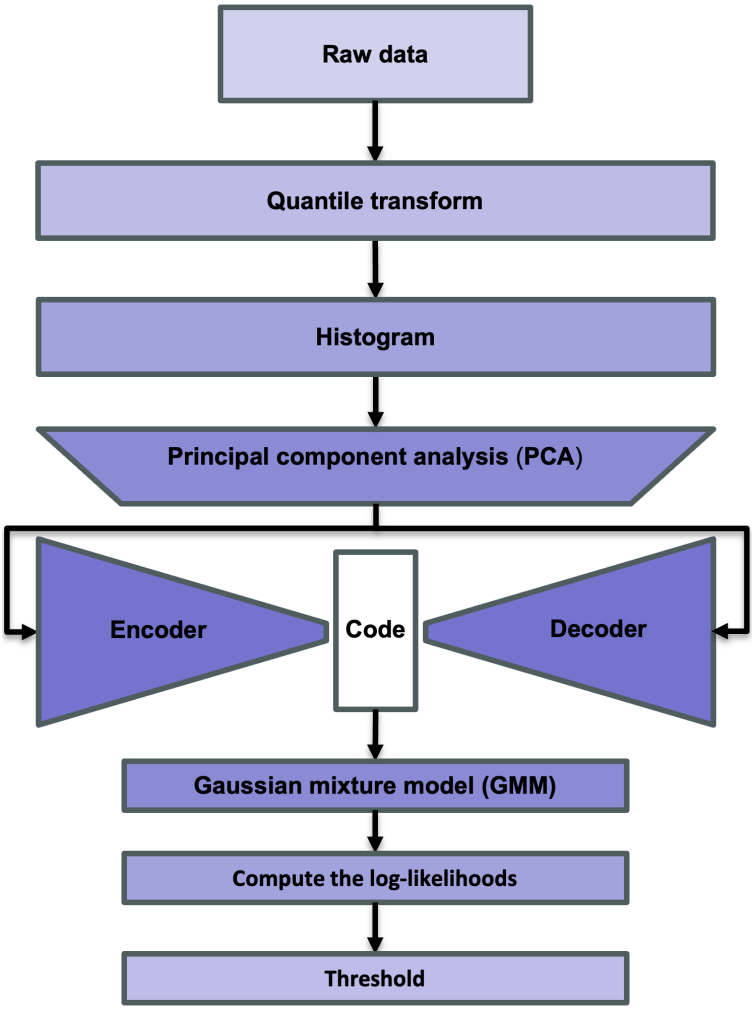


Figure 5.1: Architecture of the proposed CPD model.

the original data distribution is approximately Gaussian [185]. Real-world time-series measurements frequently display skewed or heavy-tailed distributions along with multimodal characteristics which contradict this assumption. Z-score normalization falls short in reducing outliers' influence when dealing with skewed time-series data and fails to generate features that fit a well-behaved distribution for machine learning purposes [186].

Quantile transformation represents a non-parametric method which avoids assumptions about data distribution while demonstrating greater resistance to outlier effects compared with scaling methods [187]. The process of mapping the empirical CDF of the data into a Gaussian distribution makes sure that every transformed feature matches the intended

statistical properties [188]. The use of this method proves advantageous in change point detection tasks because extreme values and non-Gaussian structures typically reduce model performance. This makes it a preferable choice for the proposed CPD framework. Consequently, in this work, quantile transformation is employed to standardize and transform the original data distribution into a Gaussian distribution.

Suppose we have a set of n observed data points, $\{x_1, \ldots, x_n\} \subset \mathbb{R}$, from a probability distribution F. We want to map our data onto an output distribution G. Then, the quantile transformation is defined as

$$y_i = Q\left(F(x_i)\right), \text{ for } i \in \{1, \ldots, n\}, \tag{5.1}$$

where F is the CDF of the input distribution, and Q is the quantile function of the desired output distribution. In practice, the distribution of the observed data points is usually unknown. Therefore, F is often replaced with an empirical CDF $\hat{F}_n$, a discrete approximation of the true distribution F based on the observed data. Quantile function Q is defined as

$$Q(p) = G^{-1}(p), \tag{5.2}$$

where G denotes the cumulative distribution function (CDF) of the target output distribution. Typically, G is selected to be either a uniform distribution or a standard normal distribution with zero mean and unit variance.

According to [189], the measurement units used in features can affect the data analysis. To help avoid dependence on the choice of measurement units, the data should be normalized or standardized. This involves transforming the data to fall within a common range. Normalizing the data attempts to give all attributes an equal weight. In addition, when training neural networks via backpropagation, normalizing or standardizing each measured attribute in the training data can speed up the learning process [189]. In this work, the quantile transformation is applied to each feature dimension independently, and the data

distribution is transformed to a standard normal, yielding standardized features with zero mean and unit variance. This ensures that all dimensions contribute comparably to data analysis and model training.

5.1.2 Data Modeling

Traditional change-point detection (CPD) methods often assume that data points are independent and identically distributed (iid) within each stationary regime [190]. In classical formulations, one typically models a time series as a sequence of iid observations drawn from one distribution before the change and from another distribution after the change [191]. While this assumption simplifies analysis, it rarely holds in high-dimensional, temporal data where observations are correlated in time (and across dimensions). Lifting the iid assumption in time series data introduces significant complexity since the data can no longer be treated as simple random samples from a static distribution, but rather as observations from a non-stationary stochastic process. This means that the underlying distribution itself may evolve over time, violating the convenient independence structure and requiring complex system modeling.

The second stage of the proposed CPD approach considers modeling the behavior of multi-agent systems as a stochastic process. Therefore, a change-point corresponds to a change in the generative mechanism of this stochastic process [178]. In probabilistic terms, there exists a time τ (the change-point), where $\tau \in [1, T]$, such that the joint distribution of $(Y(1), \ldots, \ldots, Y(T))$ cannot be described by a single consistent model across the entire timeline. Instead, there is a regime change at τ. One formalization is to say there are two (or more) different stochastic regimes: e.g., there exists τ such that

$$Y(1), Y(2), \ldots, Y(\tau - 1) \sim \mathcal{P}_0, \qquad Y(\tau), Y(\tau + 1), \ldots, Y(T) \sim \mathcal{P}_1,$$

where $\mathcal{P}_0$ and $\mathcal{P}_1$ are probability laws (process measures) governing the time series before

and after the change. Under $\mathcal{P}_0$, the marginal at time t is $f_t(x)$ (nominal density), whereas under $\mathcal{P}_1$, the marginal at time t might be a different $g_t(x)$, reflecting the new dynamics or behavior after the change. In the simplest case, $\mathcal{P}_0$ might correspond to an i.i.d. process with distribution $f(x)$ and $\mathcal{P}_1$ an i.i.d. process with distribution $g(x)$. Our more general formulation allows $\mathcal{P}_0$ and $\mathcal{P}_1$ to be non-i.i.d. processes, for example Markov chains with different transition probabilities or temporal dependencies [192]. The cost of this generality is that one must explicitly model (or at least acknowledge) the time correlations in the data, which we have done implicitly via the windowing approach.

Sliding window analysis is a key strategy for lifting the iid assumption. Instead of assuming independence at the level of individual time steps, we process the data in local windows of time. In a sliding-window CPD approach, one moves a window through the time series and treats each potential split between two windows as a candidate change-point. By aggregating data over a window of length W, short-range temporal dependencies are captured within the window, and the local statistics can be assumed (approximately) stationary over that short duration. In effect, each window provides a batch of observations that can be treated as coming from a single distribution (the distribution that governed that segment of time). This local stationarity assumption is much weaker than the global iid assumption, making the model applicable to many real-world time series problems.

For our application, the time series represents the evolving state of a crowd of N agents in a high-dimensional feature space. At each time step t, we collect data from all agents $X_1(t), X_2(t), \ldots, X_N(t)$, where each $X_i(t) \in \mathbb{R}^d$ is a d-dimensional feature vector (e.g. velocity, acceleration, etc. of agent i at time t). The collection of all agents' states at time t can be thought of as a point cloud in the d-dimensional state space. This point cloud encodes the crowd's configuration or behavior at that time. Rather than analyzing each agent's time series separately (which would be impractical for large N and high d), we focus on the aggregate distribution of points at each time. In other words, we treat the set $X_1(t), \ldots, X_N(t)$ as N samples drawn from an underlying population distribution of the crowd's state at time

t. This distribution captures the nominal behavior of the crowd at that time.

To construct a probabilistic model of the crowd's nominal behavior, we construct histograms derived from the point cloud data. Specifically, we partition the state space into a finite set of bins (cells), denoted by $B_1, B_2, \ldots, B_K$. In the experiments, K was determined empirically. For each time step t, we calculate the empirical frequency of the crowd's states falling into each bin, defined as:

$$H_t(k) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{B_k}(X_i(t)), \quad k = 1, 2, \ldots, K,$$

where $\mathbf{1}_{B_k}(\cdot)$ represents the indicator function defined by:

$$\mathbf{1}_{B_k}(x) = \begin{cases} 1, & \text{if } x \in B_k \\ 0, & \text{otherwise} \end{cases}.$$

The vector thus serves as a histogram representing the empirical probability distribution of the crowd's state at time t. We process these histograms over time, and therefore we can characterize the evolving structure of the point cloud. The use of histograms (or more generally, density estimates) is powerful here because it aims to capture the global motion patterns of the crowd rather than tracking individual movements.

To incorporate these histogram-based representations within the sliding window framework, we organize sequences of histograms into temporal windows of fixed length W. At each time step t, our model considers a window containing $H_{t-W+1}, H_{t-W+2}, \ldots, H_t$. By structuring these histograms into a tensor of shape $K \times W \times d$, we effectively capture both the spatial distributions of crowd states (encoded through histograms) and their temporal evolution.

5.1.3 Dimensionality Reduction

Dimensionality reduction is the third stage of the proposed framework, where the aim is to condense the potentially large input of size $K \times W \times d$ into a significantly more tractable latent representation. High-dimensional data in multi-agent systems can substantially raise both computational and sample complexity requirements. Traditional methods often struggle to scale effectively, leading to reduced accuracy or outright failure when dimensionality grows. Theoretically, methods such as Gaussian Mixture Models (GMMs) can estimate any density function if we have large enough number of Gaussian mixture components. Yet in practice, increasing number of components would lead to large numbers of parameters to be estimated by the maximum likelihood algorithm [193]. In this work, we adopt a cascading strategy that combines a linear projection step via Principal Component Analysis (PCA) and a subsequent nonlinear compression step using autoencoders.

In our approach, PCA is first applied to each vectorized windowed histograms, where each tensor of size $K \times W \times d$ is flattened into a vector of length $K \cdot W \cdot d$. We choose k such that the total variance loss remains below 10%. This threshold is empirically determined to keep a balance between computational feasibility and representational fidelity. The resulting low-dimensional projections serve as an efficient input for further nonlinear dimensionality reduction.

After applying PCA, we employ a custom autoencoder to capture nonlinear structures that cannot be captured with linear methods alone. In the autoencoder architecture, each layer of the encoder and decoder is a fully connected feedforward network; this architecture is chosen for its simplicity, ease of training, and capacity to model complex, nonlinear relationships.

The encoder is composed of four hidden layers that gradually reduce the dimensionality by factors of 2, 2, 2, and 3, respectively—leading to a final reduction factor of 24 from the encoder's input to its bottleneck layer. Symmetrically, the decoder mirrors this structure

in reverse. Notably, the exact input and output dimensions of the autoencoder depend on both $K \times W \times d$ and the number of principal components retained by PCA, making the architecture dynamically sized to be able to address problems with different dimensionalities.

We impose a lower bound of 2 on the reduced dimensionality to ensure that sufficient structural and statistical information is preserved for density estimation and change point detection. We empirically determined that retaining at least two dimensions maintains the expressive power of the covariance structure in GMMs, and avoids overcompression of the signal relevant to binary change point detection.

By cascading these two methods—PCA followed by a feedforward autoencoder—we achieve a dramatic reduction in dimensionality up to a factor of 3750. The use of a PCA enables to use of a smaller autoencoder architecture. A smaller autoencoder architecture not only reduces the training time and cost but also reduces the chance of overfitting training data. Overall, this layered approach mitigates the curse of dimensionality, and yields representations well-suited for subsequent density estimation and change point detection.

5.1.4 Density Estimation and Change Point Detection

The final stage of the proposed hybrid AI framework comprises density estimation followed by change point detection (CPD), leveraging the condensed latent representations obtained from the preceding dimensionality reduction steps.

To model the distributional characteristics of these latent embeddings effectively, we employ Gaussian Mixture Models (GMMs), a flexible and powerful probabilistic modeling technique. GMMs are widely used in high-dimensional statistics since, in addition to being universal approximators, they often result in mathematically tractable problems [194].

In this work, we applied the Bayesian Inversion Criterion (BIC) on the data to determine the number of Gaussian components as 20 [195]. We used the Expectation-maximization (EM) algorithm to determine the GMM parameters by fitting to reduced representation (the latent representations) for the training data. After training the GMM on latent representa-

tions derived from stable system states (i.e., data without change points), the trained model serves to evaluate the log-likelihood scores of new incoming data:

$$\ell(\mathbf{x}_t) = \log p(\mathbf{x}_t|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{5.3}$$

This work adopts a threshold-based strategy for binary change point detection. The detection threshold is determined as the q'th quantile of the log-likelihood distribution obtained from the training set, where $q \in (0, 1)$ is an empirically determined hyperparameter that controls the sensitivity of the detection algorithm and needs to be selected based on the problem requirements and priorities. A lower q results in a more conservative threshold (increasing false negatives), whereas a higher q makes the method more sensitive (increasing false positives). Typically, thresholds are chosen to satisfy a false alarm rate of a problem [196]. By setting the threshold as q'th quantile of the log-likelihood distribution of the training samples, we are approximating a $q\%$ false alarm rate with the assumption that the test data is statistically similar to the training data. This approximation is valid under the assumption that the training data only represents the distribution of scores under the null hypothesis, i.e., no change points or anomaly within the time series of the training set [197]. Formally, given the log-likelihood scores for the reduced-order representations (latent representations) of the training data samples $\{\ell(\mathbf{x}_t)\}_{t=1}^{T}$, the detection threshold ϱ is computed as:

$$\varrho = Q_\ell(q), \tag{5.4}$$

where Q_ℓ is the quantile function (inverse cumulative distribution function) of the random variable $\ell(\mathbf{x}_t)$, defined as:

$$F_\ell(a) = P(\ell(\mathbf{x}_t) \leq a), \quad a \in \mathbb{R}, \tag{5.5}$$

$$Q_\ell(p) = F_\ell^{-1}(p) = \inf\{a : F_\ell(a) \geq p\}, \quad p \in [0, 1]. \tag{5.6}$$

Thus, ϱ marks the threshold below which only q'th quantile of training log-likelihood scores

fall. In this work, the threshold is set to the 1% quantile of the log-likelihood distribution of the normal training samples. This choice helps detect data points that have a low probability (less than 1% in this work) of belonging to the stable state (state of the training data), and hence have an increased chance of belonging to an unstable state.

When monitoring new data sequences, each time step's latent representation is evaluated through the trained GMM. If the computed log-likelihood $\ell(\mathbf{x}_t)$ falls below the threshold ϱ, the corresponding time t is flagged as a potential change point:

$$\hat{\tau} = t : \ell(\mathbf{x}_t) < \varrho. \tag{5.7}$$

5.2 Classical Vicsek Model

Vicsek model is a seminal framework in the study of collective behavior in systems of self-propelled particles which was first proposed in 1995 [198]. Initially proposed as a possible explanation for bird flocking, Vicsek model simulates the dynamics of a group of agents that align their velocities with their neighbors, leading to emergent collective motion [199]. The Vicsek model is characterized by its simplicity and the ability to reproduce complex phenomena observed in biological systems, such as flocking in birds and schooling in fish [200]. In this section, we empirically investigate the phase transitions of the Vicsek model through extensive simulations, and then evaluate our CPD model for detecting change points in collective behavior of particles in Vicsek simulations.

5.2.1 Model Definition

The movement of each particle is governed by two factors: noise and local interactions with neighbors. Each particle averages the orientations over all neighboring particles within a specified radius R.

Consider a two-dimensional system of N_a self-propelling agents. The velocity of i-th

agent, $v_i(t) \in \mathbb{R}^2$ is defined as:

$$v_i(t) = v_{\text{abs}} \begin{bmatrix} \cos\theta_i(t) \\ \sin\theta_i(t) \end{bmatrix}, \tag{5.8}$$

where v_{abs} absolute speed of particles. The orientation of particle i evolves as:

$$\theta_i(t) = \langle \theta_j(t) \rangle_{\|x_i(t) - x_j(t)\| < R} + \eta(t). \tag{5.9}$$

where $\eta(t) \sim \text{Uniform}(-\sigma\pi, \sigma\pi)$. Then the position of agent i is updated by

$$x_i(t + \Delta t) = x_i(t) + v_i(t)\Delta t. \tag{5.10}$$

Throughout the simulation in this work, the particles are constrained to an $L \times L$ periodic box. Particles are initialized with uniformly random orientation and position within the $L \times L$ box. The change points occur based on the change in the two parameters of the model; σ controlling the noise, and R controlling the interaction radius of the particles. For the simulations we use the code provided in [200].

5.2.2 Order Parameter in the Vicsek Model

In [198], Vicsek et al. investigate the following formula as the order parameter of the Vicsek model:

$$v_{\text{op}}(t) := \frac{1}{v_{\text{abs}} n_{\text{ag}}} \| \sum_{i=1}^{n_{\text{ag}}} v_i(t) \|_F. \tag{5.11}$$

According to Vicsek et al., (5.11), characterizes the phase transition within the Vicsek model.

5.2.3 Phase Transition

In this work, we use the order parameter defined in (5.11) to explore the phase transition of the model. To investigate the phase transitions in the Vicsek model, we designed a sim-

ulation study by discretizing the two-dimensional parameter space defined by noise (σ) and interaction radius (R). We adopted a logarithmic scale for both parameters, to explore behaviors from ordered to disordered states. For the noise parameter σ, we set the range from 10^{-5} to 10^{1}. Similarly, the interaction radius R was varied between 10^{-4} and 10^{0}. A smaller radius limits interactions to very close neighbors, potentially leading to fragmented groups, isolated, or almost no collective behavior, while a larger radius promotes more global, cohesive movement patterns across the entire agent population. Constants $v_{\text{abs}}, n_{\text{ag}}, L$ are chosen as $0.01, 100$ and 1. Both parameters, σ and R, are discretized using 100 logarithmically

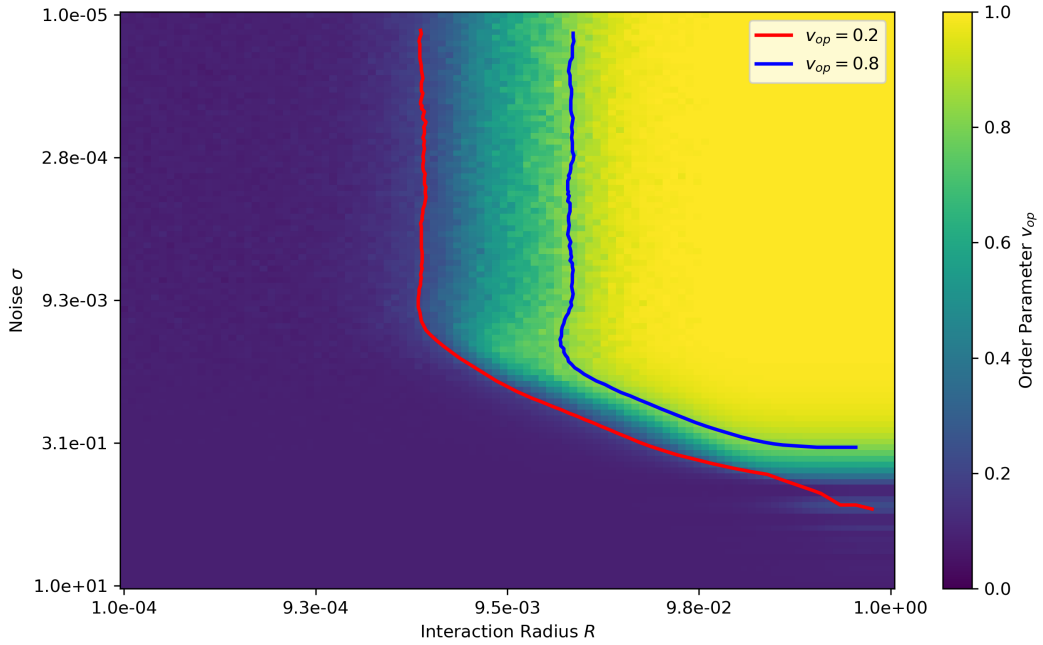


Figure 5.2: Phase transition of Vicsek model.

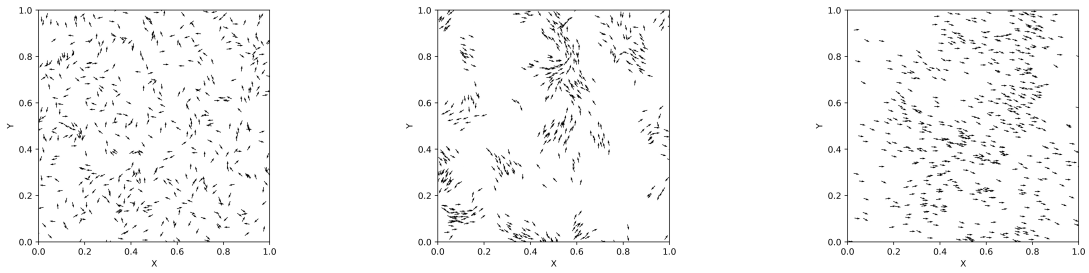
spaced points. This discretization results in a grid of $10,000$ unique (σ, R) combinations, each representing a distinct simulation condition. In our simulations, each parameter combination is initialized and run independently to reach a steady state, total of 1000 discrete steps, before measuring the order parameter $v_{\text{op}}(t)$. For each (σ, R) pair, we ran 100 independent simulations and computed the statistical mean for the order parameter v_{op}. Overall, this phase transition simulation consisted of $500,000$ individual model runs and a total of

$500,000,000$ time frames. The resultant phase transition heat map is shown in Figure 5.2. This phase transition map illustrates how the Vicsek model's behavior changes as a function of two parameters: the interaction radius R and noise level σ both presented on logarithmic scales. Horizontal axis represents the values for the interaction radius parameter R ranging from 10^{-4} and 10^{0}, and vertical axis represents the values for the noise parameter σ ranging from 10^{-5} to 10^{1}. The color bar represents the order parameter v_{op}, which measures the collective alignment or coherence of the agents' directions. The figure also highlights the contour lines corresponding to $v_{op} = 0.2$ (red line) and $v_{op} = 0.8$ (blue line). These two lines divide the map into three regions where the same or similar color intensities are grouped together; dark purple, yellow, and greenish regions.

For $v_{op} > 0.8$, we observe the yellow region. In this region system is in an ordered phase, where agents strongly align their directions, resulting in coherent collective motion. The system enters this phase under conditions of low noise (σ) and a relatively large interaction radius (R). At these parameters, agtents interact strongly and maintain alignment due to low noise, forming an ordered, and flocking behavior. Figure 5.3c displays a frame from a Vicsek simulation in an ordered phase.

For $v_{op} < 0.2$, we observe the dark purple region. In this region system is in a disordered phase, where agents exhibit random directions, showing no coherent group motion. The system enters this phase under conditions of High noise levels or relatively small interaction radius. Figure 5.3a displays a frame from a Vicsek simulation in a disordered phase.

For $0.2 < v_{op} < 0.8$, we observe intermediate, greenish color intensity, where the system is in a transition phase between ordered and disordered states. In the transition phase, we can observe partial alignment, transient or unstable clustering of agents. Intermediate noise and interaction radius could result in a system in a transition phase. Figure 5.3b displays a frame from a Vicsek simulation in a transition phase.



(a) (b) (c)

Figure 5.3: Phases of Vicsek model: (a) Disordered phase, (b) Transition between the disordered and ordered phases (c) Ordered phase.

5.3 Change Point Detection in Vicsek Model

In this section, we explore the efficacy of the proposed CPD method by evaluating through simulations of the Vicsek model across the discretized parameter space in Figure 5.2. The experiments in this section has the following same steps;

1) A parameter pair (σ, R) from the discretized space (Figure 5.2) is selected for training.

2) Independent Vicsek simulations are generated using this chosen parameter pair.

3) A test region in the parameter space is identified for model evaluation.

4) Testing data is produced by running Vicsek simulations initially at the training parameters, then switching (mid-simulation) to parameter values from the test region. This process is repeated 100 times for each point in the test region.

5) The proposed CPD model is trained on the data from Step 2.

6) The trained CPD model is evaluated on the testing data; we present the resulting heat maps of true positives and detection delays.

We selected 9 parameter pairs in the discretized parameter space that are equally spaced in log-scale. Figure 5.4 shows the nine distinct training points we selected (black "×") in the

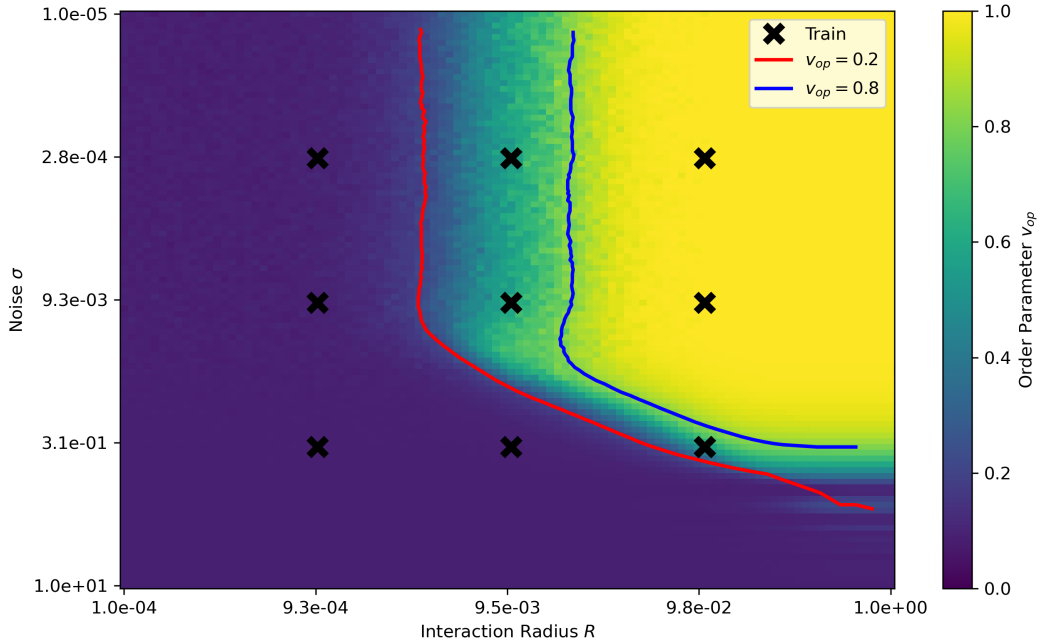


Figure 5.4: Parameter points used for generating the training dataset of the CPD model are shown with blue '×' on the phase transition map.

(σ, R) parameter space. These points represent the parameter pairs which we use to generate Vicsek simulations for the training dataset. We investigate the model for 9 different datasets in 9 separate experiments, where the datasets were generated by using these 9 parameter pairs shown in the figure. These points are equally spaced in the logarithmic parameter space. For convenience, we label the parameter choices as "small," "medium," and "large":

$$\text{Small } \sigma = 3.3 \times 10^{-4}, \quad \text{Medium } \sigma = 1.1 \times 10^{-2}, \quad \text{Large } \sigma = 3.5 \times 10^{-1},$$

$$\text{Small } R = 1.0 \times 10^{-3}, \quad \text{Medium } R = 1.0 \times 10^{-2}, \quad \text{Large } R = 1.1 \times 10^{-1}.$$

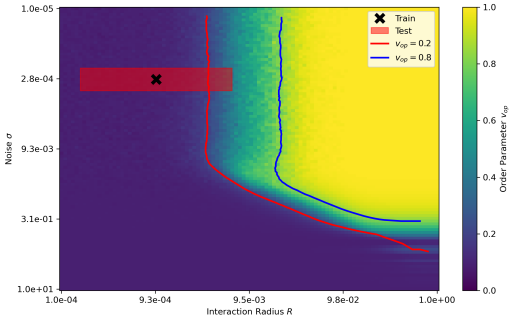
We thus conduct nine experiments, each corresponding to one of these (σ, R) pairs. Although each experiment is described in a dedicated subsection, we draw comparisons and contrasts across the experiments to highlight how the CPD model's performance changes with respect to σ and R. Subsections are named based on the selected parameter pairs fo

the training. For instance, the first subsection is named as "Small σ, Small R", indicating that in the first subsection we will investigate the results for the model that was trained with the Vicsek simulations generated with $\sigma = 3.3e-04$, and $R = 1.0e-03$.

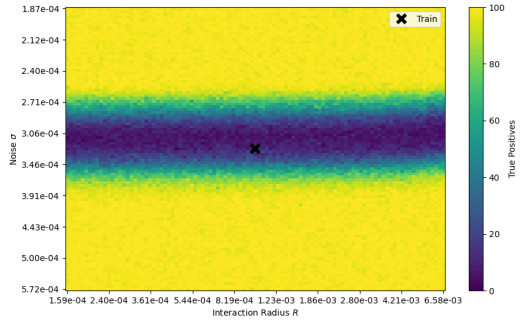
In each experiment, we perform a *local analysis* (a narrow test region around the training point) and a *global analysis* (covering a broader parameter range). We evaluate both *true positives* (detection accuracy) and *detection delays* (the time required to detect parameter changes). True positives are calculated based on how many change points are detected in 100 different test simulations. Detection delays are calculated based on how long it takes for the method to detect a change point after it occurs. Each test simulations lasts for 100 seconds, where the change points occur at $50th$ second. Hence maximum possible detection delay is 50 seconds. If method fails to detect a change point, then the detection time is recorded as 50 seconds. Below, we provide an integrated discussion of the results from all nine experiments, emphasizing recurring patterns and differences across varying σ and R.

5.3.1 Small σ, Small R

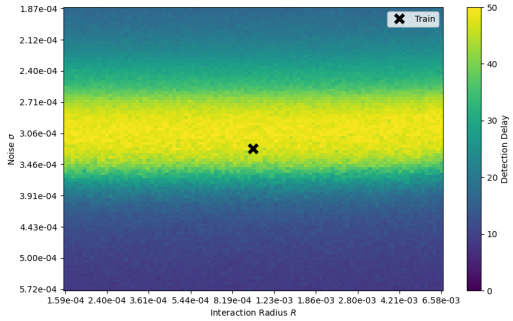
We first consider the case of small σ and small R, specifically $\sigma = 3.3 \times 10^{-4}$ and $R = 1.0 \times 10^{-3}$. Figure 5.5a highlights the training parameter point (marked by black '$\times$') and the specific testing area (highlighted in red) on the phase transition map of the Vicsek model. The measured order parameter at the training parameter point is approximately 0.1, suggesting a disordered phase for the training data simulations. The window length of the CPD model and the number of histogram bins are chosen as 50. Therefore, dimensionality reduction is applied to tensors of shape $50 \times 50 \times 2$, total of 5000 dimensions. 838 principal components were selected from the PCA with a 10% loss in the variance. The latent representation of the trained autoencoder was 34. Overall, dimensionality of the problem was dropped by factor of 147.



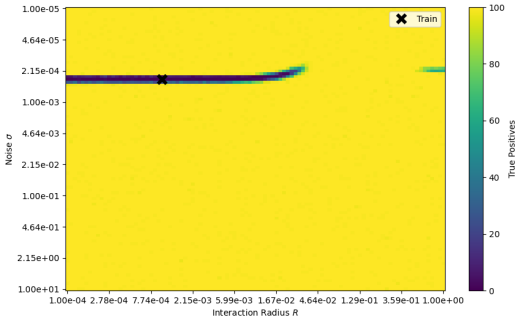
(a)



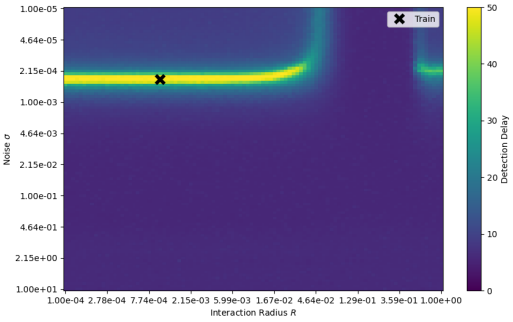
(b)



(c)



(d)



(e)

Figure 5.5: Vicsek analysis for small σ and small R: (a) The training point (black '$\times$') and the local test area (red area) overlaid on the phase transition map of the Vicsek model. (b) True positives heat map; (c) Detection delay heat map (seconds) for the local analysis. (d) True positives heat map; (e) Detection delay heat map (seconds) for the global analysis.

Local Analysis

Figure 5.5b shows a heat map of true positives across the local testing area. The training parameter pair is marked by a black '$\times$'. For small σ and small R, the color gradient ranges

from dark purple (low true positives) to bright yellow (high true positives). In general, the proposed model performs exceptionally well (high true positives, near 100%) in regions that significantly deviate from the trained noise parameter. We observe bright areas both above and below the trained noise parameter, suggesting that the method can detect significant shifts toward either higher or lower noise values.

Noticeable horizontal dark-purple band around the trained noise parameter indicates a substantial drop in true positives when the parameter change is primarily in interaction radius R, with minimal noise shift. This horizontal band highlights the model's limited sensitivity to changes in R alone when noise remain roughly constant at the trained value. The horizontal dark-purple band is not perfectly centered at the trained noise parameter, rather slightly shifted towards lower noise values. This shift indicates that the method is much more responsive to shifts in the parameters towards a higher noise value.

Figure 5.5c illustrates the detection delay (seconds) heat map across the local testing area. The training parameter pair is shown by a black '$\times$'. We observe a color distribution consistent with the observations in Figure 5.5b. Similar to the true positives heat map, we observe dark-to-bright horizontal color gradient, suggesting model is potentially blind to subtle changes in the interaction radius when noise remains largely unchanged. There is a clear increase in detection delay around the trained noise parameter, forming a bright-yellow horizontal band. This horizontal band is shifted towards lower noise values, supporting the previous observations regarding the superior performance of the method in detecting the shifts towards higher noise levels.

Global Analysis

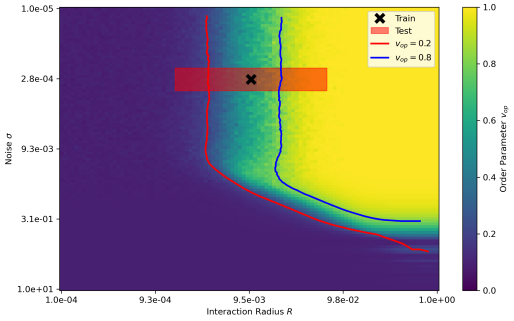
Figure 5.5d shows a heat map displaying the number of true positives across the discretized parameter space. The heat map shows extremely high true positive rates (yellow) across most of the parameter space. There is a narrow dark horizontal band (low true positives) centered around the trained noise parameter. The horizontal band starts to curve (before

vanishing) towards lower noise values, when the shift in the interaction radius increases enough. There is another separate dark horizontal band at the rightmost part of the map, occurring at or near the same noise levels as the main horizontal band.

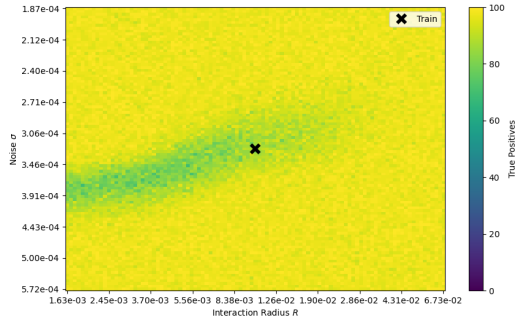
Figure 5.5e shows a heat map displaying the detection delays across the discretized parameter space. Dark colors (quick detection) persist across most of the parameter map. A bright horizontal band (high delay) aligns with the trained noise parameter, again starts to curve towards low noise levels after reaching high enough interaction radius values. We observe that this curved bright band, divides the space into two regions. The smaller region on the upper left corner of this band is brighter than the other dark purple region. These two regions have extremely high (almost 100%) true positive rates in Figure 5.5d, yet the difference between the detection times can be distinguished in Figure 5.5e. Similar to the true positive map, there is another bright horizontal band at the rightmost part of the map, at the same noise level with the main band.

5.3.2 Small σ, Medium R

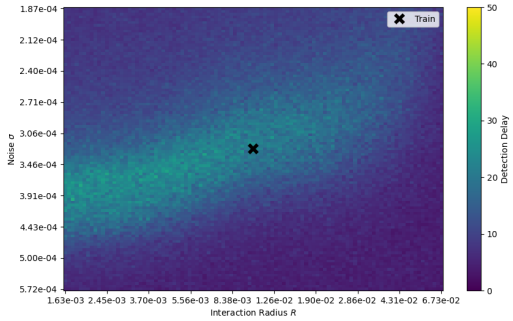
Next, we analyze the case of small noise (σ) and medium interaction radius (R), specifically with parameters $\sigma = 3.3 \times 10^{-4}$ and $R = 1.0 \times 10^{-2}$. Figure 5.6a shows the selected training parameter point (marked by a black '$\times$') and the local testing region (highlighted in red) overlaid on the Vicsek model phase transition map. The order parameter at the training location is approximately 0.6, indicating that the training simulations fall near the transition zone between ordered and disordered phases. The window length of the CPD model and the number of histogram bins are chosen as 50. Therefore, dimensionality reduction is applied to tensors of shape $50 \times 50 \times 2$, total of 5000 dimensions. 742 principal components were selected from the PCA with a 10% loss in the variance. The latent representation of the trained autoencoder was 30. Overall, dimensionality of the problem was dropped by factor of 167.



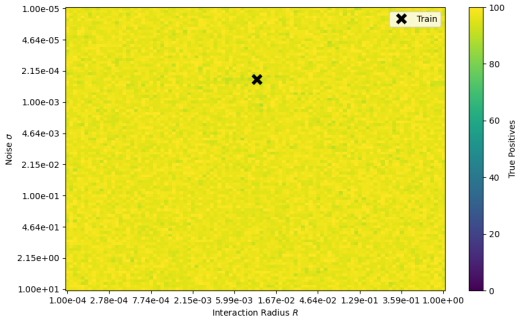
(a)



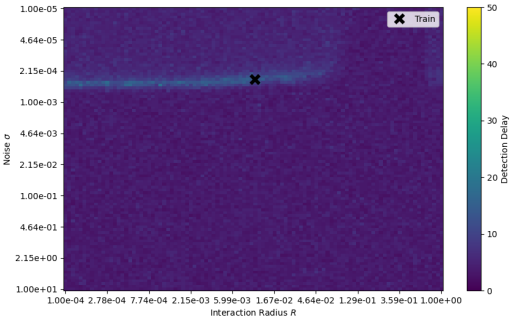
(b)



(c)



(d)



(e)

Figure 5.6: Vicsek analysis for small σ and medium R: (a) The training point (black '×') and the local test area (red area) overlaid on the phase transition map of the Vicsek model. (b) True positives heat map; (c) Detection delay heat map (seconds) for the local analysis. (d) True positives heat map; (e) Detection delay heat map (seconds) for the global analysis.

Local Analysis

The true positives heat map presented in Figure 5.6b illustrates the detection performance around the local testing area. The training parameter point is denoted by a black '×'. The

results show consistently high true positive rates (yellow regions) throughout most of the testing domain, indicating robust model sensitivity to changes in both noise and interaction radius. However, a distinct region of lower true positive rates (blue-green band) emerges, starting from the left side (small R) at around $\sigma \approx 3.9 \times 10^{-4}$, extending horizontally initially and continuing diagonally from lower-left to upper-right of the training parameter point, eventually vanishing near $R \approx 2 \times 10^{-2}$. This mixed horizontal-diagonal structure contrasts with the purely horizontal band observed for small σ and small R in two significant ways: 1) The band around the training parameter pair is diagonal rather than horizontal, reflecting the model's responsiveness to shifts in interaction radius. 2) Unlike the previous dark-purple band (close to 0 true positives), this band exhibits blue-green coloration, successfully detecting approximately $50 - 60\%$ of change points even for minimal parameter shifts.

Figure 5.6c provides insights into detection delays across the local parameter domain. An increase in detection delays aligns with the previously mentioned diagonal band of lower true positives. Detection times remain mostly below 30 seconds across the local testing area. Darker colors around the bottom-right corner indicate greater model responsiveness to simultaneous increases in σ and R.

Global Analysis

In the global analysis, Figure 5.6d presents a true positives heat map spanning the discretized parameter space. Remarkably, the global performance is exceptionally strong, characterized by universal high true positive rates (yellow coloration), indicating robust detection capability throughout the broad parameter range. Unlike the small σ, small R case, no clearly defined horizontal or diagonal low true-positive bands appear, signifying enhanced global sensitivity at these specific training parameters.

Figure 5.6e shows the global detection delay heat map. Rapid detection (dark colors) dominates the entire parameter map, reinforcing the model's efficacy across various parameter shifts. A subtle horizontal band of slightly elevated delays at the trained noise parameter,

that curves towards higher noise values at $R \approx 4.6 \times 10^{-2}$, persists; however, unlike the results in the previous experiment, the magnitude of these delays remains relatively small.

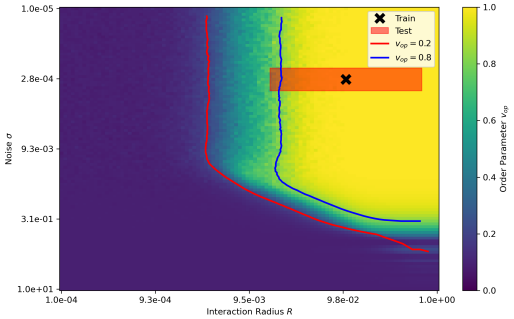
5.3.3 Small σ, Large R

In this experiment, we explore the scenario characterized by small noise (σ) and large interaction radius (R), specifically using parameters $\sigma = 3.3 \times 10^{-4}$ and $R = 1.1 \times 10^{-1}$. Figure 5.7a illustrates the selected training parameter point (marked by a black '×') along with the corresponding local testing area (highlighted in red), positioned clearly within the ordered phase region of the Vicsek model, as indicated by an order parameter value near 1.0. The window length of the CPD model and the number of histogram bins are chosen as 50. Therefore, dimensionality reduction is applied to tensors of shape $50 \times 50 \times 2$, total of 5000 dimensions. 48 principal components were selected from the PCA with a 4% loss in the variance. The latent representation of the trained autoencoder was 2. Overall, dimensionality of the problem was dropped by factor of 2500.

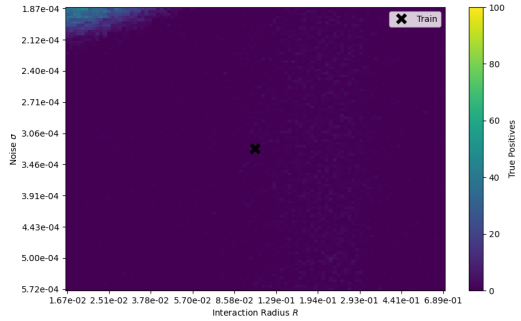
Local Analysis

Figure 5.7b shows the heat map of true positives around the local testing area. The training parameter point is denoted by a black '×'. We observe predominantly low true positive rates (dark purple regions) throughout the entire local testing domain, highlighting a pronounced difficulty in detecting subtle parameter shifts around these specific conditions. Notably, only a very small region located in the upper-left corner of the map (representing lower R and higher σ) shows slightly elevated true positive rates, yet these remain significantly below those observed in the previous experiments. This finding indicates that the model's sensitivity to parameter changes substantially diminishes at higher interaction radius values when the noise is held constant.

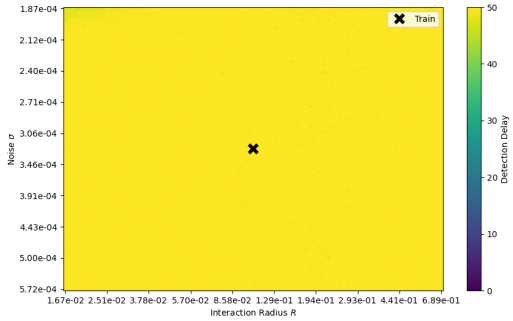
The corresponding detection delay heat map, presented in Figure 5.7c, further reinforces these findings. Detection delays are uniformly high (yellow) across the entire local parame-



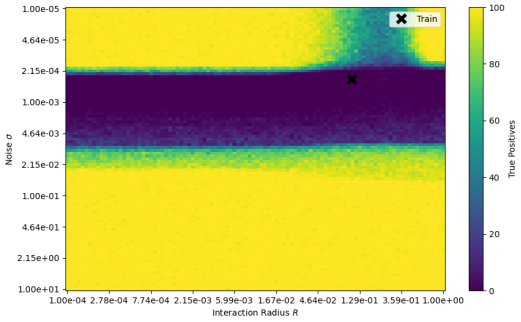
(a)



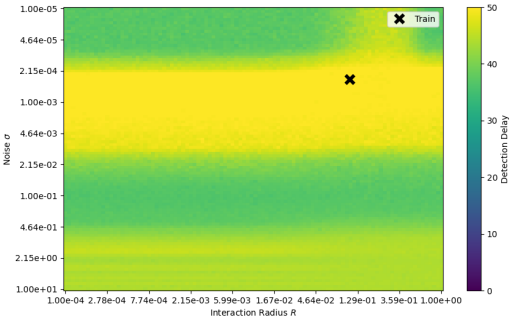
(b)



(c)



(d)



(e)

Figure 5.7: Vicsek analysis for small σ and large R: (a) Training point (black '×') and local test area (red area) on the Vicsek model's phase transition map. (b) True positives heat map; (c) Detection delay heat map (seconds) for local analysis. (d) True positives heat map; (e) Detection delay heat map (seconds) for global analysis.

ter region, reflecting prolonged model response times that approach the upper limit of the measured delays (approximately 50 seconds).

Global Analysis

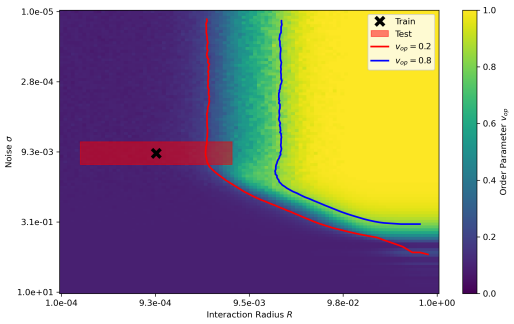
In the global context, Figure 5.7d shows the true positives heat map spanning the complete discretized parameter space. The global analysis reveals a prominent dark horizontal band centered just below the trained noise parameter, $\sigma \approx 1 \times 10^{-3}$, indicating markedly reduced sensitivity to variations in the interaction radius alone within this band. Conversely, the sensitivity improves notably with significant deviations in the noise parameter (both increases and decreases), where the model achieves near-perfect detection (yellow). This horizontal band indicates the model's limited capability to detect parameter changes if σ shifts to a value approximately between $2 \times 10^{-4} - 5 \times 10^{-3}$. Additionally, there is a noticeable vertical extension of this low-detection region into lower noise values around $\sigma \approx 2 \times 10^{-1}$, indicating a weak performance for the model around this region located in the ordered phase.

Figure 5.7e presents the global detection delay heat map, which aligns well with the observations in the global true positive map. A distinct bright-yellow horizontal band, indicative of substantial detection delays, centered below the trained noise parameter. Although true positive rates remain generally high outside this specific band, detection delays are notably elevated compared to previous scenarios. This suggests that the model requires additional time to detect significant changes when operating in a strongly ordered phase.

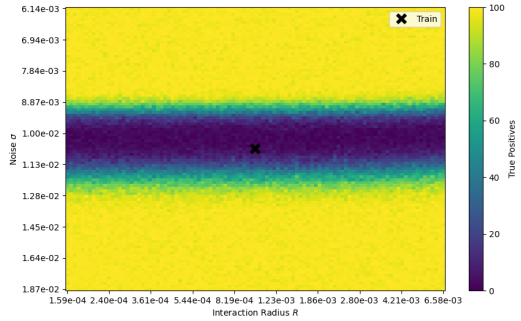
5.3.4 Medium σ, Small R

We now investigate the scenario characterized by medium noise (σ) and small interaction radius (R), specifically focusing on parameters $\sigma = 1.1 \times 10^{-2}$ and $R = 1.0 \times 10^{-3}$. Figure 5.8a illustrates the selected training parameter point (marked by a black '$\times$') and the associated local testing region (highlighted in red). The order parameter at this training point is approximately 0.1, indicating a disordered phase of collective motion. The window length of the CPD model and the number of histogram bins are chosen as 50. Therefore, dimensionality reduction is applied to tensors of shape $50 \times 50 \times 2$, total of 5000 dimensions.

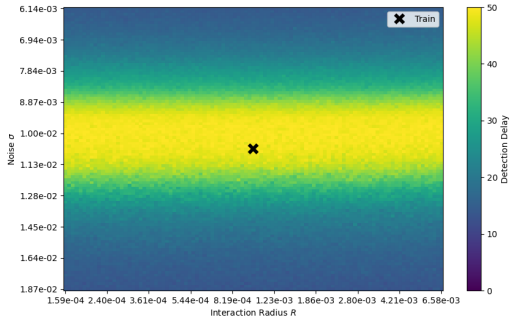
856 principal components were selected from the PCA with a 10% loss in the variance. The latent representation of the trained autoencoder was 35. Overall, dimensionality of the problem was dropped by factor of 143.



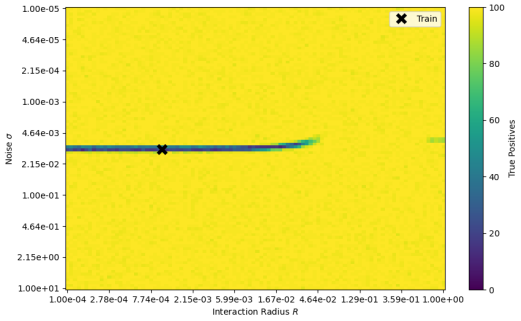
(a)



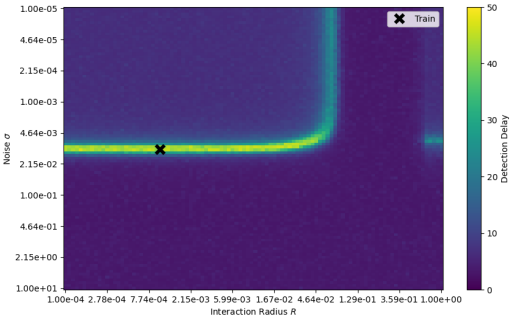
(b)



(c)



(d)



(e)

Figure 5.8: Vicsek analysis for medium σ and small R: (a) The training point (black '×') and the local test area (red area) overlaid on the phase transition map of the Vicsek model. (b) True positives heat map; (c) Detection delay heat map (seconds) for the local analysis. (d) True positives heat map; (e) Detection delay heat map (seconds) for the global analysis.

Local Analysis

Figure 5.8b depicts the true positives heat map within the local testing area. The training parameter location is again marked with a black '×'. Similar to earlier small σ, small R experiments, the map features a clearly defined horizontal dark-purple band of significantly lower true positives centered around the training noise value. This indicates reduced sensitivity primarily to shifts in the interaction radius alone in the range of noise values within this dark-purple band. Conversely, true positive rates quickly reach near-perfect detection (yellow) for shifts to both lower and higher noise values away from the trained noise parameter, demonstrating strong responsiveness of the method to noise changes.

The detection delay heat map in Figure 5.8c complements these observations. A distinct bright-yellow horizontal band emerges around the trained noise parameter, indicating substantially increased detection times (up to about 50 seconds) for parameter shifts predominantly involving the interaction radius, without significant noise deviations. Rapid detection (dark colors) is achieved quickly when shifts involve substantial changes in noise, emphasizing again the model's enhanced sensitivity to noise parameter variations over radius changes.

Global Analysis

In the global analysis, Figure 5.8d shows a true positives heat map across the discretized parameter space. The global true positives map reveals a distinct horizontal dark-purple band at the trained noise level, indicative of the model's consistent insensitivity to variations in interaction radius at or near this noise level. This insensitivity is notably narrower compared to previously discussed small σ, small R case, signifying slightly improved performance in detecting more subtle shifts. The horizontal band starts to curve (before vanishing) towards lower noise values, when the shift in the interaction radius increases enough. There is another separate dark horizontal band at the rightmost part of the map, occurring at or near the same noise levels as the main horizontal band.

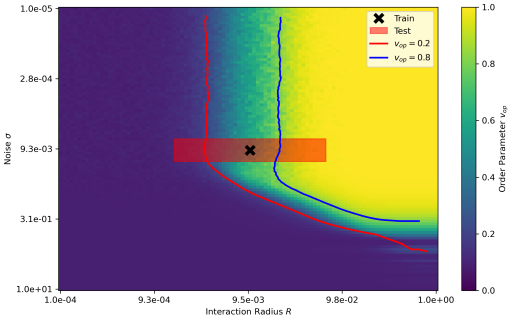
Figure 5.8e provides the global detection delay heat map. A bright horizontal band (high delay) aligns with the trained noise parameter, again starts to curve towards low noise levels after reaching high enough interaction radius values. We observe that this curved bright band, divides the space into two regions. The smaller region on the upper left corner of this band is brighter than the other dark purple region. These two regions have extremely high (almost 100%) true positive rates in Figure 5.8d, yet the difference between the detection times can be distinguished in Figure 5.8e.

5.3.5 Medium σ, Medium R

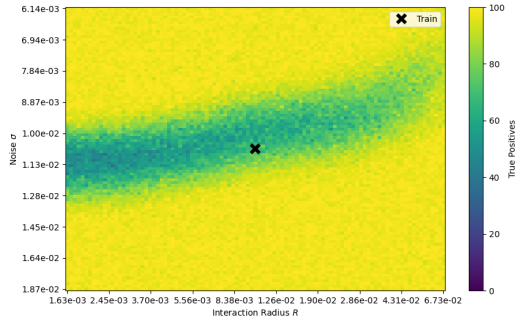
In this experiment, we focus on a medium noise, medium radius setting, selecting parameters $\sigma = 1.1 \times 10^{-2}$ and $R = 1.0 \times 10^{-2}$. Figure 5.9a highlights the training parameter point (black '$\times$') and the corresponding local test region (red area) overlaid on the Vicsek phase diagram. At this parameter configuration, the order parameter is roughly 0.6, suggesting that the system is at transition phase. The window length of the CPD model and the number of histogram bins are chosen as 50. Therefore, dimensionality reduction is applied to tensors of shape $50 \times 50 \times 2$, total of 5000 dimensions. 718 principal components were selected from the PCA with a 10% loss in the variance. The latent representation of the trained autoencoder was 29. Overall, dimensionality of the problem was dropped by factor of 172.

Local Analysis

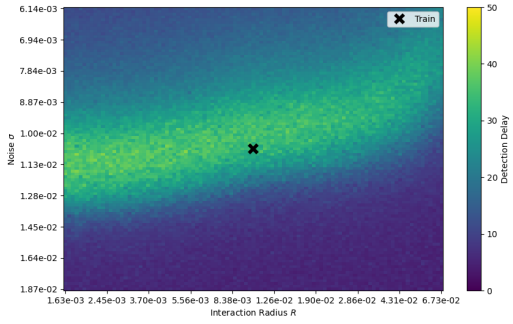
Figure 5.9b shows the true positives heat map in the vicinity of the training parameter point. Despite occupying a region near the boundary between ordered and disordered phases, the model achieves generally high true positive rates (green to yellow) for a wide range of noise and radius deviations, indicating robust detection performance around the training parameters. Notably, a moderate diagonal band of lower true positives (blue-green region) emerges around the training point, reflecting reduced sensitivity when σ and R shift proportionally in that particular trajectory. However, the overall dip in performance is less severe than the



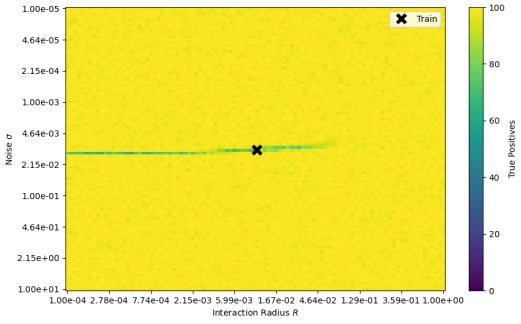
(a)



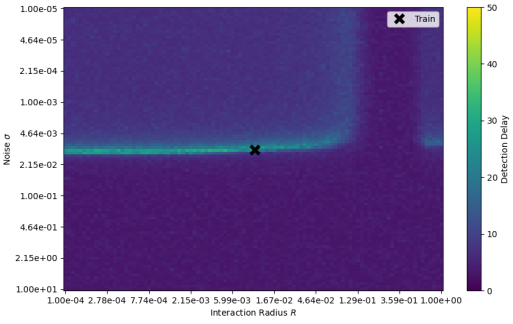
(b)



(c)



(d)



(e)

Figure 5.9: Vicsek analysis for medium σ and medium R: (a) The training point (black '×') and the local test area (red area) overlaid on the phase transition map of the Vicsek model. (b) True positives heat map; (c) Detection delay heat map (seconds) for the local analysis. (d) True positives heat map; (e) Detection delay heat map (seconds) for the global analysis.

sharper bands observed in experiments with small R values. Notably, this diagonal band of lower true positives more distinct and wider than the one observed in the small σ, medium R case.

The detection delay heat map in Figure 5.9c corroborates these results. The detection delays remain under 40 seconds across the across the local test area. A wide diagonal zone with increased detection delay (transitioning from dark blue/purple to green) correlates with the band of lower true positives, underscoring that the model requires slightly more time to detect changes that preserve a certain ratio between noise and radius. Beyond this limited region, however, the model retains prompt responsiveness, especially for the bottom right corner, where the parameters shift towards large σ, large R.

Global Analysis

Turning to the global parameter space, Figure 5.9d displays a true positives map across the broad parameter space. The dominant yellow expanse highlights the model's strong detection capabilities across broad noise and interaction radius shifts. A faint horizontal region of lower detection rates persists near the trained noise parameter, though it is considerably narrower compared to scenarios with small R, confirming that benefits from medium R conditions.

Figure 5.9e shows the accompanying detection delay map. The majority of the parameter space retains short detection times (dark purple). A subtle horizontal band of slightly elevated delays at the trained noise parameter, that curves towards higher noise values at $R \approx 5 \times 10^{-2}$, persists; however, unlike the results in the small R cases, the magnitude of these delays remains relatively mild.

Collectively, the global analysis reveals that although the model exhibits slightly reduced sensitivity and a modest slowdown in detection along particular σ-R trajectories, it performs reliably well across the vast majority of parameter shifts at this medium-noise, medium-radius setting.

5.3.6 Medium σ, Large R

In this sixth experiment, we examine a medium noise, large radius configuration, specifically choosing $\sigma = 1.1 \times 10^{-2}$ and $R = 1.1 \times 10^{-1}$. Figure 5.10a displays the training parameter

point (black '×') along with the local testing region (red band), situated well within the higher-order zone of the Vicsek phase diagram. At this chosen parameter pair, the order parameter is approximately 1.0, indicating a strongly ordered state. The window length of the CPD model and the number of histogram bins are chosen as 50. Therefore, dimensionality reduction is applied to tensors of shape $50 \times 50 \times 2$, total of 5000 dimensions. 48 principal components were selected from the PCA with a 4% loss in the variance. The latent representation of the trained autoencoder was 2. Overall, dimensionality of the problem was dropped by factor of 2500.

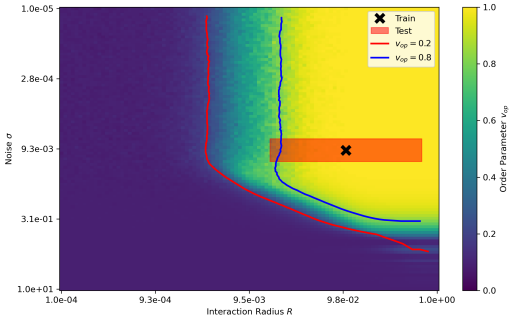
Local Analysis

Figure 5.10b depicts the true positives heat map around the local domain. The predominantly dark-purple region indicates generally low true positive rates, suggesting the model struggles to detect subtle shifts in parameter space when the interaction radius is large, and noise remains at a medium level. A slight increase in true positives (blue-green shading) appears at the periphery (bottom right corner).

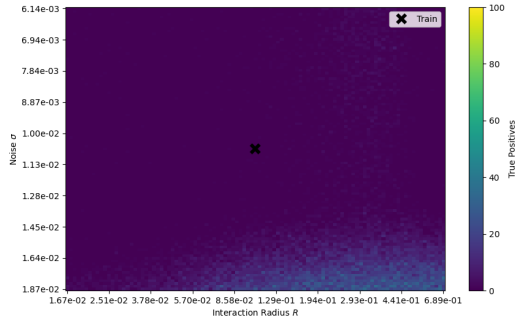
The detection delay heat map, shown in Figure 5.10c, reinforces these observations. Almost all of the local domain exhibits long detection times (yellow), consistent with the true positive results. For both small and medium σ scenarios, the model's sensitivity to local parameter changes substantially diminishes at high interaction radius values.

Global Analysis

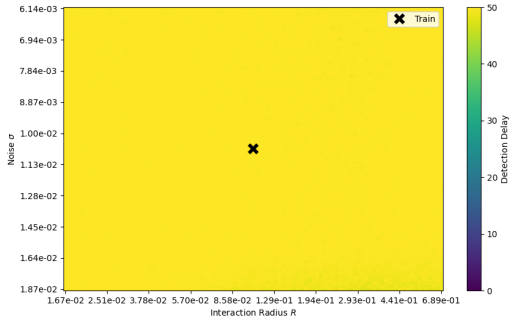
Turning to the global parameter space, Figure 5.10d presents the true positives heat map. A broad dark horizontal band centered at $\sigma \approx 1.0 \times 10^{-2}$ spans all of the interaction radius axis, indicating the model's diminished sensitivity to variations in R around this noise level. However, away from this horizontal zone, near-perfect true positive rates (yellow) are observed for large deviations in noise. Notably, the horizontal band of low true positives is significantly thinner than the small σ, large R case, whereas thicker than the rest of the



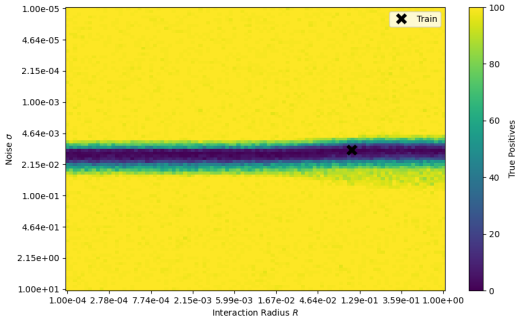
(a)



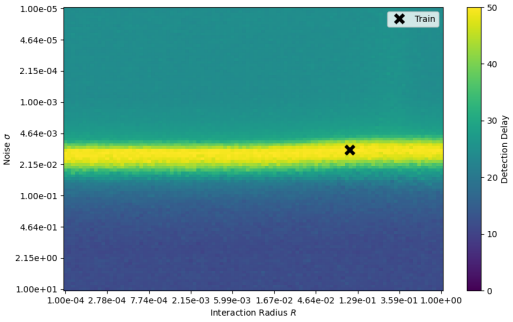
(b)



(c)



(d)



(e)

Figure 5.10: Vicsek analysis for medium σ and large R: (a) The training point (black '×') and the local test area (red area) overlaid on the phase transition map of the Vicsek model. (b) True positives heat map; (c) Detection delay heat map (seconds) for the local analysis. (d) True positives heat map; (e) Detection delay heat map (seconds) for the global analysis.

previous cases.

Finally, Figure 5.10e shows the global detection delay heat map. Consistent with the true positives map, the region around the trained noise level exhibits a bright horizontal band,

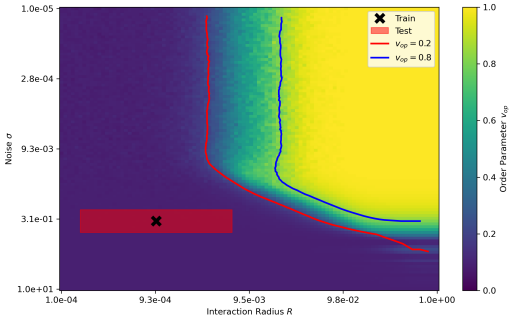
signaling longer detection times when changes in R are small but σ remains fixed at the training value. As the noise parameter shifts up or down significantly, the delays shorten. The model detects parameter shifts towards higher noise levels faster than the shifts towards lower noise levels, as seen from the brighter colors above the trained noise parameter.

5.3.7 Large σ, Small R

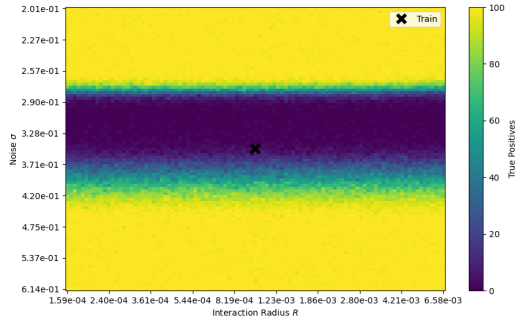
We now investigate the scenario of large noise (σ) and small interaction radius (R). In particular, we choose $\sigma = 3.5e-1$ and $R = 1.0e-3$. Figure 5.11a displays the training parameter point (black '$\times$') and highlights the local testing region (red band). At this configuration, the order parameter is approximately 0.1, indicating a disordered phase. The window length of the CPD model and the number of histogram bins are chosen as 50. Therefore, dimensionality reduction is applied to tensors of shape $50 \times 50 \times 2$, total of 5000 dimensions. 856 principal components were selected from the PCA with a 10% loss in the variance. The latent representation of the trained autoencoder was 35. Overall, dimensionality of the problem was dropped by factor of 143.

Local Analysis

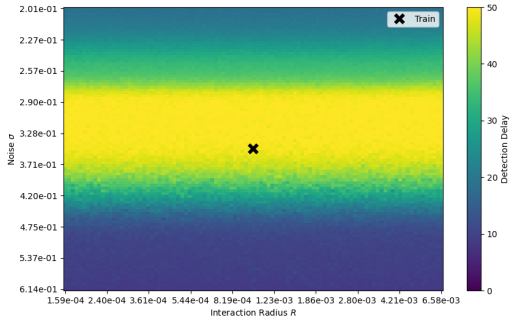
Figure 5.11b shows the true positives heat map across the local testing area. The training parameter location is again marked with a black '$\times$'. Similar to earlier small σ, small R and medium σ, small R experiments, the map features a clearly defined horizontal dark-purple band of significantly lower true positives centered around the training noise value. This indicates reduced sensitivity primarily to shifts in the interaction radius alone in the range of noise values within this dark-purple band. Below the trained noise parameter, we observe that this dark-purple region smoothly transitions towards bright-yellow where we observe near-perfect true positive rates. Above the trained noise parameter, we observe a sharper transition to bright-yellow, after consistent dark-purple area for a period of noise values. Significant noise shifts to both lower and higher values away from the trained noise



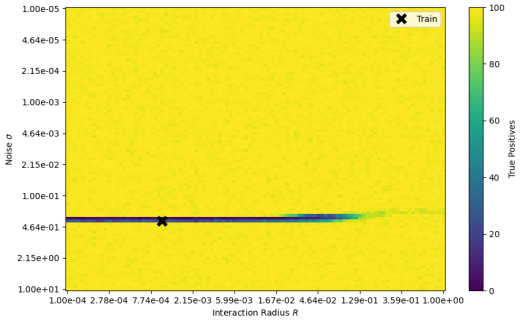
(a)



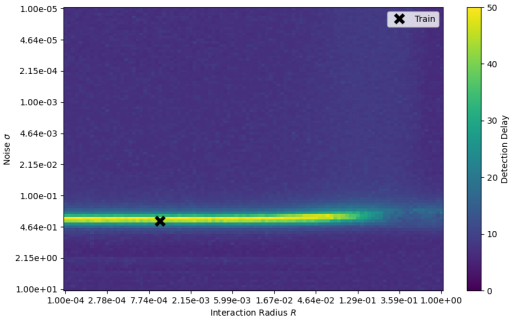
(b)



(c)



(d)



(e)

Figure 5.11: Vicsek analysis for large σ and small R: (a) The training point (black '$\times$') and the local test area (red area) overlaid on the phase transition map of the Vicsek model. (b) True positives heat map; (c) Detection delay heat map (seconds) for the local analysis. (d) True positives heat map; (e) Detection delay heat map (seconds) for the global analysis.

parameter yields strong true positive rates.

The detection delay map, presented in Figure 5.11c, complements the true positives analysis. A large portion of the parameter space is characterized by prolonged detection

times (bright-yellow horizontal band), reflecting the difficulty of discerning subtle changes at high noise levels and small radii. Again, these high detection times starts to drop as the noise parameter shifts towards higher or lower values away from the trained noise parameter.

Global Analysis

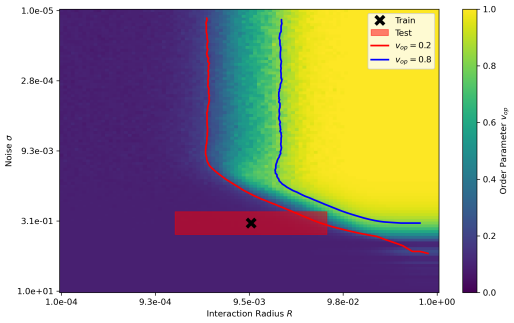
Turning to the global parameter space, Figure 5.11d provides a heat map of the true positives. A narrow horizontal band of reduced detection appears at the trained noise level, extending across a wide range of R until R becomes large enough where the band starts to vanish. This feature underscores the model's pronounced insensitivity to variations in the interaction radius (except for very large variations) alone under conditions of elevated noise. In contrast to previous two experiments with small R case, the length of this band extends to farther right of the map.

Lastly, Figure 5.11e reveals the detection delay across the entire parameter space. A narrow, long horizontal zone of higher delays aligns with the trained noise value, mirroring the low true-positive band. Model remains comparatively unresponsive to changes in R around the trained noise level. Outside this zone, the model recognizes shifts in noise away from the large σ baseline relatively quickly.

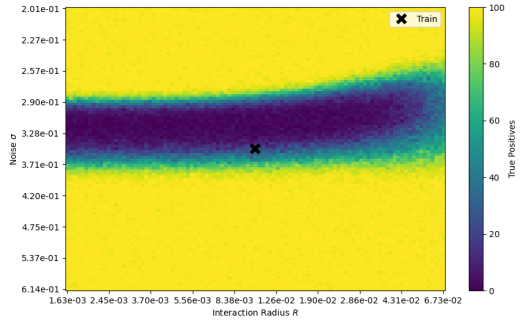
5.3.8 Large σ, Medium R

In this eighth experiment, we explore a large noise, medium radius parameter setting, specifically choosing $\sigma = 3.5 \times 10^{-1}$ and $R = 1.0 \times 10^{-2}$. Figure 5.12a highlights the training parameter point (marked by a black '$\times$') and the associated local testing region (red band). The order parameter measured at this point is approximately 0.1, signifying a disordered state. The window length of the CPD model and the number of histogram bins are chosen as 50. Therefore, dimensionality reduction is applied to tensors of shape $50 \times 50 \times 2$, total of 5000 dimensions. 842 principal components were selected from the PCA with a 10% loss in the variance. The latent representation of the trained autoencoder was 35. Overall, di-

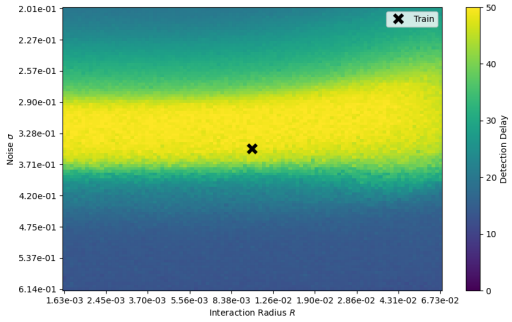
mensionality of the problem was dropped by factor of 143.



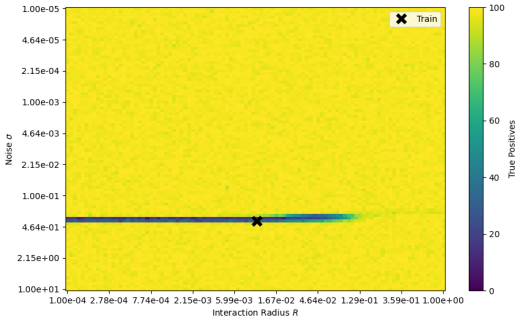
(a)



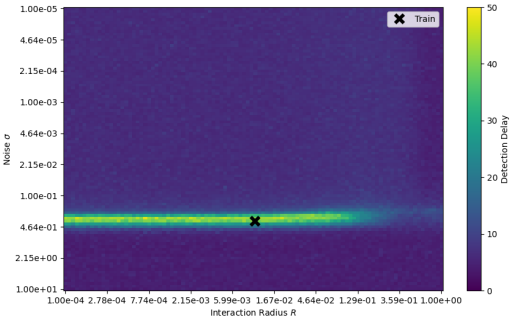
(b)



(c)



(d)



(e)

Figure 5.12: Vicsek analysis for large σ and medium R: (a) The training point (black '×') and the local test area (red area) overlaid on the phase transition map of the Vicsek model. (b) True positives heat map; (c) Detection delay heat map (seconds) for the local analysis. (d) True positives heat map; (e) Detection delay heat map (seconds) for the global analysis.

Local Analysis

Figure 5.12b shows the distribution of true positives within the local testing area. Similar to small R experiments, the map features a clearly defined horizontal dark-purple band of significantly lower true positives centered around the training noise value. This indicates reduced sensitivity primarily to shifts in the interaction radius alone in the range of noise values within this dark-purple band. Below the trained noise parameter, we observe that this dark-purple region smoothly transitions towards bright-yellow where we observe near-perfect true positive rates. Above the trained noise parameter, we observe a sharper transition to bright-yellow, after consistent dark-purple area for a period of noise values. Significant noise shifts to both lower and higher values away from the trained noise parameter yields strong true positive rates.

Figure 5.12c shows the detection delay heat map. A big portion of the parameter space is characterized by prolonged detection times (bright-yellow horizontal band). Regions where detection is faster (green/blue shades) align with areas of higher true positives, emphasizing that the model responds more promptly when the parameter shifts deviate noticeably from the training point.

In both heat maps, we observe that in the ride side of the maps, we observe that the the horizontal low true positive (high detection delay) zone starts to curve towards lower noise values, indicating some sensitivity to changes in R. In Figure 5.12a, we can see that this behavioral change happens in parallel with the significant increase in the order parameter, changing the state from a disordered phase to a transition phase that we distinguish by the contour line $v_{op=0.2}$.

Global Analysis

Moving to the global parameter space, Figure 5.12d provides an overview of the true positives across the entire grid. A dark narrow horizontal band centered closely to the trained noise parameter extends through a wide range of R values (similar to Figure 5.10d), indicating

the model's diminished responsiveness in this region when only R changes but noise remains fixed at its large trained noise value. In contrast, far from this horizontal band, the map transitions to bright yellow, indicating near-perfect true positive rates.

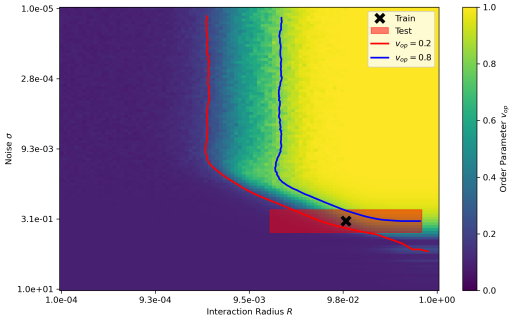
Finally, Figure 5.12e displays the global detection delay. It similarly features a narrow, bright horizontal band around the trained noise level, where detection is slowest. More rapid detection (darker hues) occurs when σ shifts away from its trained value, signifying the model's heightened sensitivity to noise variations relative to interaction radius changes.

5.3.9 Large σ, Large R

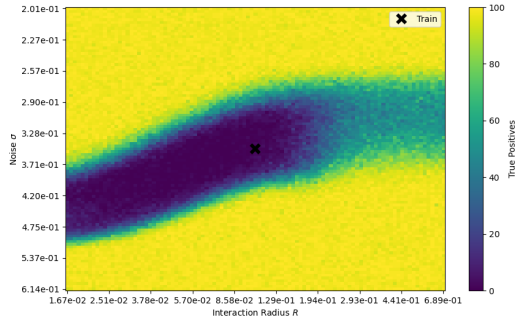
In this ninth experiment, we consider a large noise, large radius setting. Specifically, we set $\sigma = 3.5e-1$ and $R = 1.1e-1$. Figure 5.13a highlights the training parameter point (marked by a black '$\times$') and the associated local test region (red band). At this parameter configuration, the order parameter is approximately 0.5, indicating that the system resides in a transition phase. The window length of the CPD model and the number of histogram bins are chosen as 50. Therefore, dimensionality reduction is applied to tensors of shape $50 \times 50 \times 2$, total of 5000 dimensions. 480 principal components were selected from the PCA with a 10% loss in the variance. The latent representation of the trained autoencoder was 20. Overall, dimensionality of the problem was dropped by factor of 250.

Local Analysis

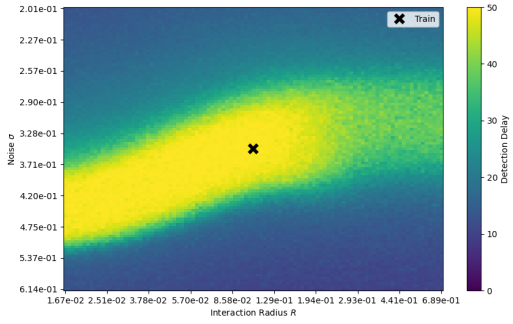
Figure 5.13b shows the true positives heat map in the vicinity of the training parameter point. The map features a wide dark-purple diagonal band of significantly lower true positives starting from bottom left and continuing towards the center of the local area. The dark-purple band then smoothly transitions into blue-green after passing the trained parameter pair and continues horizontally. This change happens in parallel with the significant increase in the order parameter, changing the state from a transition phase to an ordered phase that we distinguish by the contour line $v_{op=0.8}$.



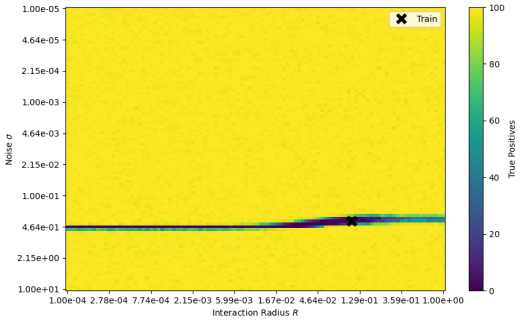
(a)



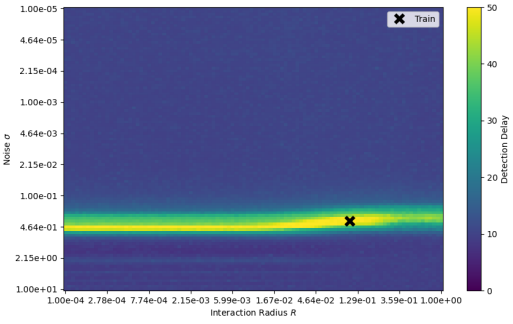
(b)



(c)



(d)



(e)

Figure 5.13: Vicsek analysis for large σ and large R: (a) The training point (black '$\times$') and the local test area (red area) overlaid on the phase transition map of the Vicsek model. (b) True positives heat map; (c) Detection delay heat map (seconds) for the local analysis. (d) True positives heat map; (e) Detection delay heat map (seconds) for the global analysis.

The detection delay heat map in Figure 5.13c mirrors these results. We observe a bright yellow diagonal band starting from bottom left and reaching towards the center of the local test area, which after passing the center, transitions into a horizontal green band for higher

R values. Outside of this bright zone, detection times improve, dropping below 30 seconds.

Global Analysis

Turning to the global parameter space, Figure 5.13d displays a true positives heat map across the broad parameter space. The dominant yellow color highlights the model's strong detection capabilities across broad parameter shifts. A dark-purple horizontal region of lower detection rates persists near the trained noise parameter, though it is considerably narrower compared to other scenarios with large R.

Figure 5.13e shows the accompanying detection delay map. The majority of the parameter space retains detection times below 25 seconds (blue). A narrow horizontal band of highly elevated delays (bright yellow) at the trained noise parameter persists; however, it is considerably narrower compared to other scenarios with large R.

5.3.10 Discussion

Sensitivity to Noise vs. Interaction Radius

A recurring theme across all nine experiments is that the model is generally more sensitive to changes in noise σ than to changes in interaction radius R. Horizontal low true positive (TP) bands consistently form at or near the trained σ, underscoring the CPD method's relative insensitivity to R-only shifts. Although this phenomenon appears in every experiment, the exact shape and extent of these low-TP regions vary with σ and R.

The impact of changes in R is especially evident in the global detection delay heat maps. In experiments with small-to-medium σ and R, the bright (yellow or green) high-delay zones often curve within ordered-phase regions. This curvature indicates that the proposed method remains responsive to R-induced phase transitions (from disordered or transitional states into ordered states). Furthermore, when R increases sufficiently, the formerly horizontal high-delay bands bend upward and begin to take on a more vertical shape, suggesting that R can

overshadow σ as the primary determinant of the model's responsiveness in these portions of the parameter space.

Ordered vs. Disordered Phases

Model performance is highly sensitive with respect to whether the system is in ordered or disordered phase. For the experiments where the model is trained in a disordered phase, we experience high true positive rates and fast detection unless the change solely occurs in the interaction radius. Whereas model performs poorly and slowly across the entire parameter space in the experiments where the model is trained in an ordered phase. When model is trained at a transition phase, it shows it's peak performance, with highest detection rates, and it is responsive to R-only shifts unlike the rest.

5.4 Unity Crowd Simulation

In the previous section, we evaluated our approach on Vicsek simulations, which models the collective behavior of the active particles in 2D. In this section, we consider a 3D human-crowd simulation.

5.4.1 Dataset

This study[1] uses Unity[2], a cross-platform game engine developed by Unity Technologies, to simulate crowd behaviors in a synthetic train station scenario, leveraging Unity's real-time 3D engine for realistic agent navigation and crowd dynamics.

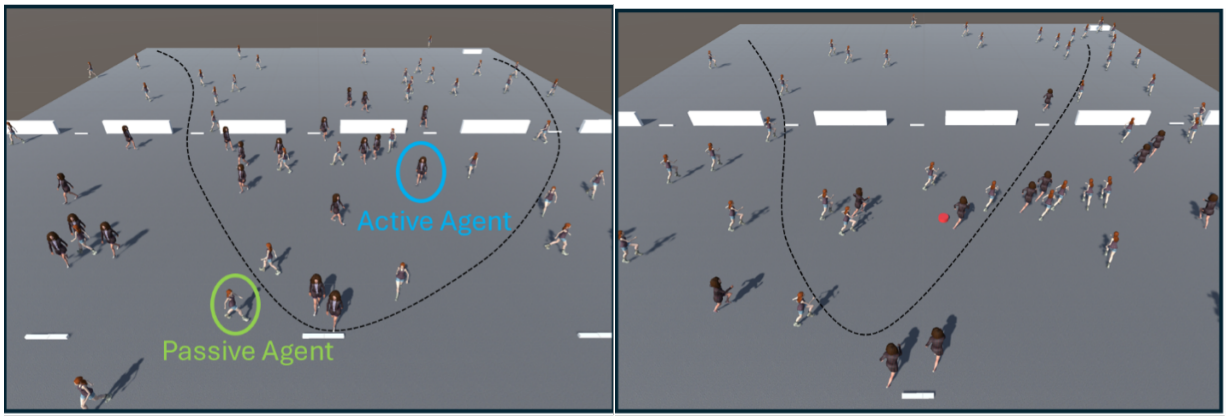
The simulation includes both normal and abnormal scenarios where we can vary parameters such as the number of agents, time of arrival, and rate of appearance. In the normal scenario, the simulated train station features a structured environment where agents follow

[1]The simulation and experimental results presented in this work were conducted in private collaboration with Bahar Kor, a PhD student in our research team, and are included here with explicit permission.

[2]Unity, the Unity logo, and all related names, logos, product and service names, designs, and slogans are registered trademarks of Unity Technologies or its subsidiaries, https://unity.com/legal/trademarks.

a predictable sequence of activities. Upon arrival, agents first spawn at the entrance gate, then proceed to move toward the train door, and ultimately continue to the exit gate. Figure 5.14a displays a normal scenario of agents at a train station.

For abnormal scenarios, an object is introduced into the scene, simulating an emergency event. This triggers agents to abandon their standard behaviors, rushing toward the exit gate to simulate a rapid evacuation. These emergency dynamics mimic real-world reactions to unexpected events, adding depth to the simulation. Figure 5.14b displays an abnormal scenario of agents at a train station.



(a) (b)

Figure 5.14: Scenes from the Unity crowd simulation of agents at a train station (a) in a normal scenario, (b) in an abnormal scenario.

For realistic simulations of the crowd behavior, there are two types of agents in the system: active and passive agents. Active agents provide their accelerometer information, while passive agents, with random movement patterns, disturb the movement of the active agents. The presence of these agents further randomizes the "normal" behavior of the crowd and simulates diverse real-world crowd scenarios.

During the simulation, detailed positional data is recorded for each agent, specifically tracking the x, y, and z coordinates of their upper left leg at every time step. These raw positional time series undergo post-processing, including smoothing with a Savitzky-Golay filter to mitigate artifacts introduced by imperfections in the Unity simulation. Subsequently,

a second-order numerical differentiator generates accelerometer data to emulate readings from a device placed in the agent's pocket. The smoothing is critical to ensure the data quality by reducing noise and other inconsistencies, which mirrors the challenges of working with real-world data.

5.4.2 Experiment Setup

In this experiment, we ran 1000 independent crowd simulations under normal scenario to generate the training data of the model. For evaluation, we conducted 200 additional simulations under abnormal conditions, each containing a scripted emergency event at the 24.5^{th} second.

A successful detection is recorded if the model flags a change point at or after the 24.5^{th} second. If it flags a time step before 24.5^{th} second, we label this as a false positive detection. If no change point is detected after 24.5 seconds, the result is considered a false negative. The detection delay is measured from 24.5 seconds onward.

Similar to the Vicsek experiments, both the window length of the CPD model and the number of histogram bins are chosen as 50. Therefore, dimensionality reduction is applied to tensors of shape $50 \times 50 \times 3$, total of 7500 dimensions. 48 principal components were selected from the PCA with a 7% loss in the variance. The latent representation of the trained autoencoder was 2. Overall, dimensionality of the problem was dropped by factor of 3750.

5.4.3 Benchmark Models

UnSupervised Anomaly Detection (USAD)

USAD is an unsupervised model that is composed of three elements; an encoder and two decoders which are connected into an architecture composed of two autoencoders sharing the same encoder in an adversarial training framework [201]. The architecture is trained in

two phases. The first phase involves training both autoencoders to learn to reconstruct the input data. The second phase involves an adversarial training, where first autoencoder seeks to fool the second, while the second autoencoder tries to learn if the data is real or reconstructed by the first autoencoder [201]. This design combines the strengths of autoencoders (learning compact representations of normal time-series patterns) and adversarial learning (focusing deviations), achieving fast, stable convergence without the mode-collapse issues seen in GAN-based methods [202]. A key advantage of USAD is its training efficiency. The method does not use resource-hungry recurrent models, but only attention-based network architectures to improve training speed [203]. However, because it processes fixed-length windows independently and forgoes explicit long-term sequence dependence, USAD can struggle with long-term anomalies that only become evident over extended temporal contexts [203].

TranAD

TranAD is a deep transformer network for multivariate time-series anomaly detection that uses attention-driven sequence encoders to model both local and long-range temporal patterns [203]. The network uses focus score-based self-conditioning to enable robust multimodal feature extraction and adversarial training to gain stability. Additionally, TranAD incorporates a meta-learning technique (MAML) to enable model training in the limited data scenarios. These architectural choices enable TranAD to achieve robust detection and diagnosis performance with data and time-efficient training [203].

OmniAnomaly

OmniAnomaly is a probabilistic anomaly detection model for multivariate time series [204]. The model uses a stochastic recurrent neural network and a planar normalizing flow to generate reconstruction probabilities [201]. It also proposes an adjusted Peak Over Threshold (POT) method for automated anomaly threshold selection that outperforms the previously used a non-parametric dynamic error thresholding (NDT) approach [205, 203]. The model

achieves significantly improved performance, yet comes with the expense of increased energy consumption and training times [201].

5.4.4 Results

We compare the efficacy of our model to the benchmark methods described above. We compare the models based on their Area Under the Curve and F1 scores. F1 score is a harmonic mean of the precision and recall, where the score reaches its best value at 1. The ROC curve is a graphical representation of a classifier's performance across different threshold values. The Area Under the Curve (AUC) is a single scalar value summarizing the overall performance of the model. ROC curve is a plot of the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. AUC measures the entire two-dimensional area under the ROC curve, giving a single number to evaluate the classifier's performance. For a classifier, if ROC AUC takes 0.5, then we can interpret the classifier as a random classifier such as a coin toss. If ROC AUC takes 1.0, the classifier is ideal.

Table 5.1 shows the area under the receiver operating characteristic curve (ROC AUC), and F1 scores.

The proposed model achieves the highest ROC AUC of 0.996, outperforming all three benchmark methods. Although the margins are small, the improvement indicates that the proposed model provides slightly superior discriminative capability in distinguishing between normal and abnormal scenarios.

The proposed model exhibits a higher F1-Score (0.965) compared to the benchmark methods' already high scores. This improvement shows the proposed model's better performance at keeping the balance between precision and recall, by its robust detection performance and reduced false positive rates.

In terms of detection delay, OmniAnomaly achieves the fastest response at 0.8s, while USAD and TranAD share a slightly slower delay at 0.9s. The proposed model, although

demonstrating superior accuracy metrics, has a slightly longer detection delay of 1.0s. 1 or 2 ms increase in detection time could be ignorable in many applications. However, in certain applications where even a millisecond is critical, a trade-off between false alarm rate and detection time can be performed by choosing a less aggressive threshold (e.g., higher percent quantile).

USAD			TranAD			OmniAnomaly			Proposed Model		
AUC	F1	DD	AUC	F1	DD	AUC	F1	DD	AUC	F1	DD
0.990	0.948	0.9s	0.989	0.944	0.9s	0.990	0.950	**0.8s**	**0.996**	**0.965**	1.0s

Table 5.1: Performance comparison of the methods on the Unity Crowd Dataset based on ROC AUC, F1-Score, and detection delays (DD)

5.5 Conclusion

In this chapter, we have introduced a AI-driven probabilistic framework for change point detection in high-dimensional time series data, with a particular emphasis on multi-agent systems characterized by potentially large and time-varying numbers of agents. Our approach employs a carefully designed pipeline consisting of cascading stages; data processing, modeling, dimensionality reduction, density estimation and change point detection. The main contribution of this work is a CPD approach that models a time varying probability density function capturing the nominal behavior of the entities, and use it as the basis for detecting changes. The proposed approach takes temporal dependencies and dependencies arising from the agent interactions into consideration and models the time series for the multiple interacting entities as a stochastic process.

We validated our method on two different systems of collective behavior. First, through extensive Vicsek model simulations, we studied how the proposed CPD algorithm behaves across different regions of parameter space, including disordered, transition, and ordered phases. The results highlighted that our framework exhibits high sensitivity to changes in the noise parameter σ while being comparably less sensitive to changes in the interaction

radius R, especially when the training state is highly ordered or highly disordered. When trained in intermediate or transition regimes, the model showed robust detection capabilities across a wide range of perturbations.

Second, we applied our method to a realistic 3D Unity simulation of human crowd dynamics. Our approach demonstrated high performance in detecting abrupt changes—specifically, emergencies or evacuations—outperforming several state-of-the-art anomaly detection baselines (USAD, TranAD, and OmniAnomaly) in terms of both ROC AUC and F1 scores.

Overall, these results suggest that the proposed hybrid AI model is well-suited for practical change point detection tasks in complex, high-dimensional, and time-varying multi-agent systems. Future research may focus on finding a more sophisticated autoencoder architecture that captures much more temporal and multi-agent dependencies. The sensitivity of the model to the number of agents can be analyzed further to understand the limits of the model. Finally, evaluation of the method on benchmark datasets can help further validate the results in this work.

Chapter 6

Conclusions and Future Work

This thesis presented reduced order methods for imaging inverse problems and time series analysis to enhance image reconstruction accuracy and change point detection in problems associated with data and computational limitations. We developed an enhanced parametric level-set framework for image reconstruction and extended its application to spatiotemporal sensor data for tracking evolving objects while creating an AI-based probabilistic CPD method to identify change points in high-dimensional time series data arising from multi-agent systems. These advancements highlighted the importance of robust reduced order methods in addressing large-scale problems to solve critical real-world problems.

The central theme was to retain the essential structure of the problem (be it an image with distinct regions or a time series with complex dependencies) while drastically reducing the number of parameters or features required to describe it. Through this paradigm, we developed new methods that handle high dimensionality and ill-posedness more effectively than classical approaches. In this chapter, we summarize the main findings of each contribution (Chapters 3–5) and discuss promising directions for future research building upon these results.

6.1 Conclusions

6.1.1 Chapter 3: Parametric Level-sets Enhanced To Improve Reconstruction (PaLEnTIR)

Chapter 3 introduced Parametric Level-sets Enhanced to Improve Reconstruction (PaLEnTIR), a significantly enhanced parametric level-set method for reconstructing piecewise-constant images. This work addressed three key challenges faced by traditional level-set and PaLS approaches: representing multiple contrast levels, expressing geometrically complex shapes, and improving numerical stability.

PaLEnTIR uses a single level-set function to represent multiple object regions with different contrast values, thanks to a new smooth transition function that replaces the binary Heaviside function. This is noteworthy as previous methods required multiple level-set functions for multiple regions and contrasts. By maintaining a single level-set regardless of how many distinct contrasts are present, the model avoids the growth in complexity while improving the overall performance implied in visual perspective and across various score metrics throughout the experiments.

Another major contribution of PaLEnTIR is its enhanced geometric flexibility. The method replaces conventional radial basis functions with anisotropic basis functions (ABFs), enabling representation of a wider class of shapes (e.g., stretched and rotated ellipses) with each basis element. This helps to represent finer image details with fewer basis functions, directly tackling limitations of RBFs, which can only produce circular cross-sections in a single basis function case. In practice, this means PaLEnTIR can reconstruct complex object boundaries and extended structures more accurately than RBF-based approaches using the same or even fewer parameters.

Moreover, the refined parameterization in PaLEnTIR eliminates key sources of redundancy and indeterminacy that are associated with RBF PaLS model. By resolving such

redundancies, PaLEnTIR improves the inversion and optimization process as implied by the improved condition numbers in the Jacobians.

Empirical results presented in Chapter 3 showed that PaLEnTIR produced more detailed and accurate reconstructions than both RBF PaLS and TV methods, all while using far fewer number of parameters relative to number of pixels in the reconstruction. It also exhibited improved numerical stability, as evidenced by the sensitivity results relative to TV regularization parameter, and significantly lower condition numbers in the Jacobian matrices compared to the RBF PaLS formulation. These advancements show that PaLEnTIR is a powerful PaLS approach and a reduced order technique for imaging by combining the topological advantages of level sets with the efficiency of parametric modeling.

6.1.2 Chapter 4: Enhanced Parametric Level-Set Methods for Tracking Evolving Objects

Chapter 4 extended the PaLEnTIR methodology to address time-evolving imaging problems, specifically focusing on the monitoring of diffusive processes with sparse and noisy sensor data. In many real-world scenarios; such as environmental or biomedical monitoring, it is often needed to monitor how substance concentrations in regions of interest change over time from limited observations. We addressed this by reformulating PaLEnTIR for spatiotemporal reconstruction of evolving objects. The core innovation is the introduction of a ladder function for the level-set formulation. Instead of a single binary Heaviside function defining binary structure for the region, the ladder function is constructed as a sum of multiple transition functions at different levels, which allows multiple concentration thresholds to be represented together using a single level-set formulation. This approach leverages the mathematical property that solutions to diffusion equations (and many physical processes) vary continuously in time; any intermediate value between an initial and final state must occur at some time by the Intermediate Value Theorem. The ladder function capitalizes on this by ensuring the level-set field passes through those intermediate levels in a smooth, ordered

manner.

In addition to the ladder function formulation, we integrated a temporal smoothness regularization into the objective of the problem, based on the assumption that the object's evolution is smooth and continuous. The temporal smoothness regularization becomes especially crucial in noisy and limited data scenarios to maintain smoothness between successive PaLEnTIR reconstructions in time. Together, the multi-concentration single level-set representation and temporal regularization make the framework well-suited for tracking applications where one seeks a low-order representation of objects diffusing into a region over time.

The effectiveness of this spatiotemporal PaLS approach was demonstrated through simulation studies of diffusion processes. Even with sparse sensor layouts and noisy measurements, the method reliably represented the key features of evolving concentration levels. Notably, as discussed in Chapter 4, accurate recovery of multiple concentration thresholds was achieved with as few as a couple hundred sensors over a large domain and relative to more than $40,000$ pixels in the displayed images. The reconstructions closely matched the contrasts and shapes of the true concentration profiles at each time, indicating that the proposed reduced-order method successfully encoded the essential spatial and temporal information. Overall, Chapter 4 demonstrated a low-order parametric approach for tracking evolving objects under real-world data limitations, with the combination of advantages of a PaLS approach, ladder function, and temporal regularization.

6.1.3 Chapter 5: AI-Driven Probabilistic Change Point Detection in High-Dimensional Time Series Data

Chapter 5 focused on the development of an AI-driven probabilistic change point detection (CPD) method built for high dimensional, multi-agent time series data. This work was motivated by scenarios in which numerous agents or sensors generate time-dependent data where one needs to detect abrupt changes in the system behavior. Traditional CPD algorithms face difficulties in such settings due to four main challenges; the first challenges is to deal with

temporal dependencies where the data points are not independent in time, second challenge arises from the inter-agent dependencies where the trajectories or signals of different agents are correlated, third challenge arises from the huge dimensionality of multivariate time series data coming from multiple agents, and the forth challenge appears from variable data dimensionality due to the fact that the number of active agents can change over time, leading to missing data or varying input size. In Chapter 5, it has been shown that the proposed CPD model overcomes these challenges by combining principled probabilistic modeling with modern statistical and machine learning tools.

At its core, the proposed approach treats the multi-agent system as a non-stationary stochastic process observed through time. To achieve this, the proposed model processes the data in local windows of time and therefore short-range temporal dependencies are captured within the window. This sliding window technique makes the model applicable to various real-world time series problems that does not hold independence assumption at the level of individual time steps.

Rather than analyzing each agent separately or ignoring critical dependencies among dimensions, we model the joint configuration of all agents at a given time as a random sample from an evolving probability distribution. A change point is then reflected as a significant shift in this underlying distribution. To implement this, we introduced a representation of the multi-agent state at each time step as a fixed-dimensional feature, constructed by aggregating agent attributes into a histogram over a discretized state space. This histogram-based encoding serves two crucial purposes: it captures inter-agent relationships by considering the collective state rather them processing each individual agent separately, and it yields a consistent size representation even if the number of agents changes since agents entering or leaving simply alter the histogram counts and keep the number of bins fixed. This representation addresses the issue of variable size in the data, and therefore allows the use of standard machine learning models that rely on consistent and fixed-size feature space.

Probabilistic modeling of the data as a stochastic process results in a problem with huge

dimensionality to deal with. Therefore, the CPD model incorporates a cascaded dimensionality reduction pipeline to drastically compress the data before performing change point detection. We first apply linear dimensionality reduction,PCA , to capture the principle component that matter the most, and then a nonlinear reduction via autoencoders with ANNs to further compress the data while preserving complex patterns. This two-stage reduction was shown to reduce the effective dimensionality by up to a factor of 3750 in the experiments. With the reduced latent representation of the data, we modeled its temporal evolution using a nonparametric density estimation technique to estimate the distribution of a "normal" system behavior. After learning the probability distribution of the latent features during an initial training phase when the system is assumed to be in a nominal state, the model starts monitoring for shifts or deviations from this distribution to serve as an evidence to a change point. Consequently, this approach accounts for both time correlations through the sliding window of histograms, and inter-agent relations through the histogram representation of agents, and therefore avoids the unrealistic independence assumptions that limit many CPD methods.

Experimental results in Chapter 5 demonstrated the efficacy of the proposed CPD framework. In simulations based on the Vicsek model, the sensitivity of the model is evaluated in different phases of the model. The results showed that the proposed approach is able to detect most of the variations in the noise parameter of the agents, and in the interaction radius parameter when the change triggers a phase transition. Furthermore, in a realistic 3D human crowd simulation at a train station, the proposed framework successfully identified sudden changes arising from emergency evacuations. It outperformed several state-of-the-art anomaly detection models by achieving higher ROC-AUC and F1 scores for change point detection with a minimal trade-off in the detection time, which can be improved with a trade-off in false alarm rate. These experiments validated that the combination of probabilistic modeling and AI-driven dimensionality reduction can be effectively used in very high-dimensional time series problems, for detecting change points in multi-agent systems

with complex dependencies among dimensions.

6.2 Future Work

6.2.1 Chapter 3: Parametric Level-sets Enhanced To Improve Reconstruction (PaLEnTIR)

Building on top of the enhancements in PaLEnTIR, several road-maps can be pursued to further improve its capabilities. Potential future work includes:

1. **3D and Complex Applications:** One of the roadmaps is to consider extending the PaLEnTIR to 3D imaging problems and more complex applications. Applying the method to 3D volumetric reconstructions (e.g. in 3D medical imaging) would test its ability to handle increased dimensionality and new physics.

2. **Adaptive Basis Function Placement:** Another option for the next steps is to consider incorporating adaptive refinement strategies for the placement of basis functions. Rather than using a fixed, homogeneous grid of basis functions, an adaptive approach could allocate additional ABFs in regions with high curvature or detail, and could possibly improve reconstruction accuracy without a proportional increase in the number of overall parameters. Such strategies, i.e. to adaptive mesh refinement, could allow PaLEnTIR to represent fine features more efficiently and address inhomogeneously distributed fine details in region of interests.

3. **Computational Efficiency and Optimization:** While PaLEnTIR significantly reduces the number of parameters, the computational cost of solving the inverse problem can still be notable for large-scale problems. Future efforts should include optimizing the implementation and conducting a thorough analysis of runtime and scalability. Comparing PaLEnTIR's computational performance against traditional pixel-wise

methods in practical settings will help quantify the gains from reduced-order modeling and identify any limitations to address.

4. **Experimental Validation and Benchmarking:** This method has been validated on simulated and real data in various 2D inverse problems and compared to TV and RBF PaLS methods. A critical next step is to apply the enhanced PaLS framework to more real-world datasets and compare its performance with state-of-the-art methods. Such studies would show the advantages and disadvantages of the proposed approach over the benchmark models, to demonstrate advantages of a PaLS approach in real applications, and to highlight any limitations to address in future research.

5. **Uncertainty Quantification:** Finally, we recommend an investigation into the application of PaLEnTIR for uncertainty quantification. This could involve introducing Bayesian formulations or prior distributions for the PaLEnTIR parameters and performing inference (e.g., via MCMC) to estimate confidence intervals for the reconstructed shapes and contrasts. By developing prior models for PaLEnTIR parameters based on object priors, one could evaluate the reliability of reconstructions by quantifying accuracy in tasks such as object localization and characterization.

PaLEnTIR can be further refined into a more adaptable and trustworthy imaging tool, extend its advantages to a wider range of inverse problems by taking the recommended next steps.

6.2.2 Chapter 4: Enhanced Parametric Level-Set Methods for Tracking Evolving Objects

The promising outcomes of the spatiotemporal PaLEnTIR framework encourage several paths for further research:

1. **Application to Complex Systems and 3D Data:** Similar to Chapter 3, one immediate extension is to apply the method to more complex forward models and

three-dimensional time-evolving data. For instance, the approach could be tested on dynamic medical imaging modalities, or large-scale environmental systems beyond diffusion. Adapting the ladder-function PaLS model to these scenarios would validate its advantages and highlight any modifications needed to handle additional physics or higher dimensional reconstructions.

2. **Estimating Unknown Level Parameters:** In our current formulation, the specific contrast or concentration levels (the "steps" of the ladder function) were treated as known, predefined thresholds. In future work, it would be valuable to jointly estimate these levels alongside the shapes. Making the steps or levels of the ladder function adaptive could allow the model to automatically discover critical thresholds in the data, and therefore broaden its applicability to cases where these levels are not predetermined or not certain. This could provide more flexibility in representing multi-contrast, time-evolving objects, and can expand the possible applications of the proposed approach.

3. **Adaptive Spatiotemporal Basis Refinement:** Similar to Chapter 3, an adaptive strategy for placing basis functions in both space and time could improve reconstruction fidelity. By dynamically adding more basis functions in regions or time intervals with complex and rapid changes, the model can concentrate representational power where it's most needed. This could also possibly allow handling of, for example, objects that deform or split/merge over time by locally increasing resolution instead of strictly clustered objects assumption, all while keeping the overall parameter count modest. Developing criteria for when and where to introduce additional basis functions (perhaps based on residual errors or estimated uncertainty) would be an important aspect of this extension.

4. **Computational Scalability and Real-Time Performance:** As the temporal dimension grows, even a reduced-order model will face increased computational load. It is important to study the scalability of the method with respect to the number of

time frames (N_T) and ensure that the approach can be made computationally efficient. Techniques such as parallel processing, or incremental updates (processing data sequentially rather than solving a giant joint optimization) could be explored to enable near-real-time tracking for applications.

5. **Experimental Validation and Benchmarking:** So far, the method has been validated on simulated data. A critical next step is to apply the spatiotemporal PaLS framework to real-world datasets and compare its performance with state-of-the-art methods. Such studies would show the advantages and disadvantages of the proposed approach over the benchmark models, possibly illustrate the advantages of a PaLS approach in spatiotemporal problems, and may highlight any limitations to address in future research.

By pursuing these recommended future steps, the proposed extension to PaLEnTIR for tracking evolving objects could be further improved and possibly used in various time-dependent inverse problems.

6.2.3 Chapter 5: AI-Driven Probabilistic Change Point Detection in High-Dimensional Time Series Data

The proposed AI-driven CPD model showed promising results throughout the experiments. There is still room left for further improvements and investigation. The future directions for investigation and improvements include:

1. **Advanced Deep Learning Models:** One of the directions is to explore more sophisticated architectures for the autoencoder used in the dimensionality reduction stage. Improving how the model captures temporal and inter-agent dependencies in its latent representation could further boost change point detection performance. Currently, the architecture is built with fully connected neural networks. These neural networks are simple yet powerful AI networks. However, there are more complex and sophisticated

architectures, such as LSTMs and attention mechanisms, that could further improve the dimension reduction and learning the temporal dependencies.

2. **Scalability and Sensitivity Analysis:** Another important direction is to analyze the model's behavior as the scale of the problem grows. While the current method handles variable numbers of agents, a detailed study on how detection performance change with the number of agents would be valuable. This includes testing the limits of the approach in extremely large swarms. Sensitivity analysis with respect to number of histogram bins and the window length are also critical next steps for future work. These analysis could help us learn the trade-offs and optimal values for desired outcomes depending on different task needs, such as faster detections or fewer false positives. For histogram binning, we recommend an adaptive binning strategy where the binning is determined based on the data. There are various strategies that determine binning according to some optimization procedure or based on bias-variance trade-off, as well as techniques that allow the bin widths to vary and determine the bin edges based on the structure of the distribution of the data, such as the Bayesian Blocks algorithm [206].

3. **Benchmarking with Real Data:** To validate the performance of the proposed CPD method further, extensive evaluations on real world datasets are crucial. Future work should evaluate the framework to benchmark multi-variate time series datasets. Comparing the performance against established CPD algorithms on these benchmarks will highlight strengths and any weaknesses of this approach.

These next steps will further improve and evaluate the performance of the proposed CPD framework.

Appendix A

Derivatives of the 2D PaLEnTIR parameterization with respect to the model parameters

In this section, we present the derivations essential for calculating the Jacobian of the 2D PaLEnTIR model, denoted as $f(\mathbf{r};\mathbf{p})$. Since $\mathbf{p}$ encompasses both anisotropic basis function (ABF) parameters and contrast parameters, we represent these parameters with two separate vectors: $\hat{\mathbf{p}}$ for the ABF parameters and $\mathbf{p}_c$ for the contrast parameters. The PaLEnTIR model, now expressed as $f(\mathbf{r};\hat{\mathbf{p}},\mathbf{p}_c)$, is defined as:

$$f(\mathbf{r};\hat{\mathbf{p}};\mathbf{p}_c) = C_\Delta(\mathbf{r};\mathbf{p}_c)T\left(\phi(\mathbf{r};\hat{\mathbf{p}})\right) + C_L(\mathbf{r};\mathbf{p}_c). \tag{A.1}$$

Here $C_\Delta(\mathbf{r};\mathbf{p}_c) = C_H(\mathbf{r};\mathbf{p}_c) - C_L(\mathbf{r};\mathbf{p}_c)$. We begin by examining the derivatives with respect to $\hat{\mathbf{p}}$. The derivative of $f(\mathbf{r};\hat{\mathbf{p}};\mathbf{p}_c)$ with respect to a parameter $\hat{p}_i$, that is the i^{th} parameter in $3N$-vector $\hat{\mathbf{p}}$, is expressed using the chain rule as:

$$\frac{\partial f(\mathbf{r};\hat{\mathbf{p}};\mathbf{p}_c)}{\partial \hat{p}_i} = C_\Delta(\mathbf{r};\mathbf{p}_c)T'(\phi(\mathbf{r};\hat{\mathbf{p}}))\frac{\partial \phi(\mathbf{r};\hat{\mathbf{p}})}{\partial p_i}. \tag{A.2}$$

The first factor on the right side in (A.2) is the difference between the contrast coefficients. The second term is the derivative of the transition function. The transition function for a c-level-set, $T(x)$, is defined as:

$$T\left(x\right)=\frac{1}{2}\left[1+\frac{2}{\pi}\tan^{-1}\left(\frac{\pi(x-c)}{w}\right)\right], \tag{A.3}$$

and the derivative of (A.3) is given by:

$$T'\left(x\right)=\frac{1}{w}\left[\frac{1}{1+(\frac{\pi(x-c)}{w})^2}\right]. \tag{A.4}$$

The third factor in (A.2) is the derivative of the parametric level-set function, ϕ, with respect to model parameters. ϕ is defined in (3.2). The expressions for the derivatives of (3.2) with respect to each parameter are provided below, where elements $[\alpha_j;\beta_j;\gamma_j]$ correspond to the parameters of the j^{th} ABF for $j=1,2\ldots N$:

$$\frac{\partial\phi(\mathbf{r};\hat{\mathbf{p}})}{\partial\alpha_j}=\frac{\partial\sigma_h(\alpha_j)}{\partial\alpha_j}\psi\left(\mathbf{R}_j(\mathbf{r}-\boldsymbol{\chi}_j)\right), \tag{A.5}$$

$$\frac{\partial\phi(\mathbf{r};\hat{\mathbf{p}})}{\partial\beta_j}=\sigma_h\left(\alpha_j\right)\frac{\partial\psi\left(\mathbf{R}_j(\mathbf{r}-\boldsymbol{\chi}_j)\right)}{\partial\beta_j}, \tag{A.6}$$

$$\frac{\partial\phi(\mathbf{r};\hat{\mathbf{p}})}{\partial\gamma_j}=\sigma_h\left(\alpha_j\right)\frac{\partial\psi\left(\mathbf{R}_j(\mathbf{r}-\boldsymbol{\chi}_j)\right)}{\partial\gamma_j}. \tag{A.7}$$

The derivatives with respect to parameters in the equations above are defined as:

$$\frac{\partial\sigma_h(\alpha_j)}{\partial\alpha_j}=\frac{2e^{-\alpha_j}}{\left(1+e^{-\alpha_j}\right)^2}, \tag{A.8}$$

$$\frac{\partial\psi\left(\mathbf{R}_j\mathbf{r}_\Delta\right)}{\partial\beta_j}=\psi\left(\mathbf{R}_j\mathbf{r}_\Delta\right)\mathbf{r}_\Delta^T\begin{bmatrix}e^{2\beta} & 0\\ \gamma e^{\beta} & -e^{-\beta}\end{bmatrix}\mathbf{r}_\Delta, \tag{A.9}$$

$$\frac{\partial\psi\left(\mathbf{R}_j\mathbf{r}_\Delta\right)}{\partial\gamma_j}=\psi\left(\mathbf{R}_j\mathbf{r}_\Delta\right)\mathbf{r}_\Delta^T\begin{bmatrix}0 & e^{\beta}\\ e^{\beta} & 2\gamma\end{bmatrix}\mathbf{r}_\Delta. \tag{A.10}$$

where $\mathbf{r}_\Delta = (\mathbf{r} - \boldsymbol{\chi}_j)$. Next, the derivative of $f(\mathbf{r}; \hat{\mathbf{p}}; \mathbf{p}_c)$ with respect to $\mathbf{p}_c$ is defined as

$$\frac{\partial f(\mathbf{r}; \hat{\mathbf{p}}; \mathbf{p}_c)}{\partial pc_k} = C_\Delta(\mathbf{r}; \boldsymbol{e_k}) T\left(\phi(\mathbf{r}; \hat{\mathbf{p}})\right) + C_L(\mathbf{r}; \boldsymbol{e_k}), \tag{A.11}$$

where $\boldsymbol{e_k} \in \mathbb{R}^{2N}$ is a $2N-$vector with it's k^{th} element equal to 1 and all other elements equal to 0, for $k = 1, 2 \ldots 2N$.

Appendix B

Objective function properties

The objective function is given in (2.7), where $\mathbf{p^T} = [\hat{\mathbf{p}}^\mathbf{T}\ \mathbf{p_c^T}]$. Furthermore, we define the nonlinear residual $\boldsymbol{\rho}(\mathbf{p}) = \mathcal{M}(\,\mathbf{f}(\mathbf{p})\,) - \mathbf{d}$, so that

$$\mathrm{F}(\mathbf{p}) = \frac{1}{2}\sum_{j=1}^{\widehat{m}} \rho_j^2(\mathbf{p}), \tag{B.1}$$

where $\widehat{m} = m_s \cdot m_d \cdot m_f$. It follows directly from (2.7) and (B.1) that F is bounded from below. The gradient of $\mathrm{F}(\mathbf{p})$ is given by

$$\frac{\partial \mathrm{F}}{\partial p_k} = \sum_{j=1}^{\widehat{m}} \rho_j(\mathbf{p}) \frac{\partial \rho_j}{\partial p_k}, \tag{B.2}$$

for $k = 1, \ldots, m_p$ (the number of parameters), which gives

$$\nabla \mathrm{F} = \mathbf{J^T}\boldsymbol{\rho}, \tag{B.3}$$

where $\mathbf{J}$ is the Jacobian of the residual, $\mathbf{J}_{jk} = \frac{\partial \rho_j}{\partial p_k}$. The second derivatives of the objective function are then given by

$$\frac{\partial^2 \mathrm{F}}{\partial p_k \partial p_\ell} = \sum_{j=1}^{\widehat{m}} \frac{\partial \rho_j}{\partial p_\ell}\frac{\partial \rho_j}{\partial p_k} + \sum_{j=1}^{\widehat{m}} \rho_j \frac{\partial^2 \rho_j}{\partial p_\ell \partial p_k}, \tag{B.4}$$

which gives for the Hessian of F,

$$\mathbf{H}_F = \mathbf{J}^T\mathbf{J} + \sum_{j=1}^{\widehat{m}} \rho_j \mathbf{H}_j, \tag{B.5}$$

where $\mathbf{H}_j$ is the Hessian of ρ_j.

We first consider the forward model that is linear in $\mathbf{f}(\mathbf{p})$, $\mathcal{M}(\mathbf{f}(\mathbf{p})) = \mathbf{A}\mathbf{f}(\mathbf{p})$. We limit ourselves to the 2D case, as the 3D case is analogous.

In this case, we have $\mathbf{J} = \mathbf{A}\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$, where $\partial \mathbf{f}/\partial \mathbf{p}$ is the Jacobian of $\mathbf{f}$. The matrix $\mathbf{A}$ typically comes from some discretization and has finite norm. The derivatives of $\mathbf{f}(\mathbf{p})$ are given in A. It is clear from (3.1)–(3.5) and the fact that $\mathbf{A}$ has finite norm that $\boldsymbol{\rho}(\mathbf{p})$ remains bounded for all parameters vectors $\mathbf{p}$ (including the coefficients of both $\hat{\mathbf{p}}$ and $\mathbf{p}_c$), independent of $\mathbf{r}_\Delta$. Considering the coefficients in the Jacobian $\mathbf{J}$ of $\boldsymbol{\rho}$, it follows from the equations in A that all (first) derivatives remain bounded for all parameter values, independent of $\mathbf{r}_\Delta$, and that all second derivatives exist. Notice that in (A.9) the first factor on the right-hand side ($\psi(.)$) contains the negative exponential of the exponential terms $e^{\pm 2\beta}$ and γe^{β}, whereas the second term only contains polynomial terms in the (same) exponentials. Hence $\partial\psi(\mathbf{R}_j\mathbf{r}_\Delta)/\partial\beta_j$ remains bounded for all β_j, independent of $\mathbf{r}$. The same holds for $\partial\psi(\mathbf{R}_j\mathbf{r}_\Delta)/\partial\gamma_j$ in (A.10). Hence $\|\nabla F\|_2 = \|\mathbf{J}^T\boldsymbol{\rho}\|_2$ remains bounded for any parameter values, independent of $\mathbf{r}_\Delta$.

For the second derivatives, $\|\mathbf{J}^T\mathbf{J}\|_2$ remains bounded, since $\|\mathbf{J}\|_2$ is bounded. For the second derivatives of the functions ρ_j essentially the same discussion holds as for the first derivatives, and so the Hessian of $\|F(\mathbf{p})\|_2$ remains bounded. In conclusion, the objective function (B.1) for the forward model $\mathcal{M}(\mathbf{f}(\mathbf{p})) = \mathbf{A}\mathbf{f}(\mathbf{p})$ satisfies the conditions on the objective function for the convergence proof: bounded from below, twice differentiable, and the Hessian is bounded.

Next, we consider the PDE-based forward model in (3.12). For simplicity, we consider only a finite difference discretization. For the detailed derivation of the structure and properties of $\mathbf{A}(\mathbf{f}(\mathbf{p});\omega)$ we refer to [153, 207]. From [153] and (3.12), we have that $\mathbf{A}(\mathbf{f}(\mathbf{p});\omega) =$

$\mathbf{A}_0 + \mathbf{A}_1(\mathbf{p}) + \frac{\iota\omega}{\nu}\widetilde{\mathbf{I}}$. Here, $\mathbf{A}_0$ is the finite difference discretization of $-\nabla \cdot (D(\mathbf{r})\nabla\eta)$ for a known scalar diffusion field $D(\mathbf{r}) \geq D_0 > 0$ scaled by h^2, so that the diagonal coefficients of $\mathbf{A}_0$ are $O(1)$. The matrix $\mathbf{A}_1(\mathbf{p}) = \text{diag}(\mathbf{f}(\mathbf{p})) = \text{diag}(f(\mathbf{r}_s; \mathbf{p}))$, with $\mathbf{r}_s$ the location of grid point s, represents the strictly positive absorption term $\kappa(\mathbf{r}; \mathbf{p})$ scaled by h^2 ($\kappa = 0.015$ inside the anomaly and $\kappa = 0.005$ outside the anomaly). The matrix $\widetilde{\mathbf{I}}$ represents the identity scaled by h^2 and with the diagonal coefficients corresponding to the right and left boundaries set to zero (due to the boundary conditions). So, the matrix $\mathbf{A}(\mathbf{f}(\mathbf{p}); \omega)$ is invertible for any parameter vector $\mathbf{p}$ and the norm of its inverse is bounded by a constant independent of $\mathbf{p}$.

In this case, we define the (vectorized) residual as $\boldsymbol{\rho}(\mathbf{p}) = \text{vec}\left(\boldsymbol{\Psi}(\mathbf{f}(\mathbf{p}); \omega)\right) - \mathbf{d}$, where $\mathbf{d}$ is the noisy measured data, and again define $\mathbf{F}$ as in (B.1). We assume for ease of discussion that we have only data for a single frequency ω (there are no real differences for multiple frequencies). Let

$$\boldsymbol{\Psi}_{i,j}(\mathbf{f}(\mathbf{p}); \omega) \quad = \quad \psi_{i,j}(\mathbf{f}(\mathbf{p}); \omega) = \mathbf{c}_i^T \mathbf{A}(\mathbf{f}(\mathbf{p}); \omega)^{-1} \mathbf{b}_j. \tag{B.6}$$

Then, writing $\mathbf{A}$ for $\mathbf{A}(\mathbf{f}(\mathbf{p}); \omega)$ and $\psi_{i,j}$ for $\psi_{i,j}(\mathbf{f}(\mathbf{p}); \omega)$, we have

$$\frac{\partial \psi_{i,j}}{\partial p_k} = -\mathbf{c}_i^T \mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial p_k} \mathbf{A}^{-1} \mathbf{b}_j,$$

and

$$\frac{\partial^2 \psi_{i,j}}{\partial p_l \partial p_k} = \mathbf{c}_i^T \mathbf{A}^{-1} \left(\frac{\partial \mathbf{A}}{\partial p_l} \mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial p_k} + \frac{\partial \mathbf{A}}{\partial p_k} \mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial p_l} - \frac{\partial^2 \mathbf{A}}{\partial p_l \partial p_k} \right) \mathbf{A}^{-1} \mathbf{b}_j. \tag{B.7}$$

Since $\partial \mathbf{A}/\partial p_k = \partial \mathbf{A}_1/\partial p_k = \text{diag}(\partial f(\mathbf{r}_s; \mathbf{p})/\partial p_k)$, and $\partial^2 \mathbf{A}/(\partial_l \partial p_k) = \partial^2 \mathbf{A}_1/(\partial p_l \partial p_k) = \text{diag}\left(\partial^2 f(\mathbf{r}_s; \mathbf{p})/(\partial p_l \partial p_k)\right)$, with the same partial derivatives as above for the forward model that is linear in $\mathbf{f}(\mathbf{p})$, the norms $\|\mathbf{A}(\mathbf{f}(\mathbf{p}); \omega)^{-1}\|_2$, $\|\partial \mathbf{A}_1/\partial p_k\|_2$, and $\|\partial^2 \mathbf{A}_1/(\partial p_l \partial p_k)\|_2$ are all bounded from above independent of p. With $\boldsymbol{\rho}$ as defined above, we get for its Jacobian,

$$\mathbf{J} = \left[\frac{\partial \boldsymbol{\rho}(\mathbf{p})}{\partial p_1} \quad \frac{\partial \boldsymbol{\rho}(\mathbf{p})}{\partial p_2} \quad \cdots \right].$$

Hence, the gradient, defined by (B.3), remains bounded independent of $\mathbf{p}$. The Hessian is again defined by (B.5). Since $\mathbf{J}$ is bounded in norm, so is $\mathbf{J}^T\mathbf{J}$. The coefficients of the Hessian of ρ_k are given by (B.7) for the (matrix) indices i and j corresponding to the (vector) index k. Since each coefficient of the Hessian of ρ_k is bounded (and so is ρ_k) the Hessian of $\mathrm{F}(\mathbf{p})$ is bounded as well (independent of $\mathbf{p}$).

So, in this case, too, the objective function for the PDE-based forward model satisfies the conditions on the objective function for the convergence proof: bounded from below, twice differentiable, and the Hessian is bounded.

Bibliography

[1] Dinggang Shen, Guorong Wu, and Heung-Il Suk. "Deep learning in medical image analysis". In: *Annual review of biomedical engineering* 19.1 (2017), pp. 221–248.

[2] Luís ML Oliveira and Joel JPC Rodrigues. "Wireless Sensor Networks: A Survey on Environmental Monitoring". In: *J. Commun.* 6.2 (2011), pp. 143–151.

[3] Ruey S Tsay. *Multivariate time series analysis: with R and financial applications*. John Wiley & Sons, 2013.

[4] K Kirk Shung, Michael Smith, and Benjamin MW Tsui. *Principles of medical imaging*. Academic Press, 2012.

[5] Suranga Seneviratne, Yining Hu, Tham Nguyen, Guohao Lan, Sara Khalifa, Kanchana Thilakarathna, Mahbub Hassan, and Aruna Seneviratne. "A survey of wearable devices and challenges". In: *IEEE Communications Surveys & Tutorials* 19.4 (2017), pp. 2573–2620.

[6] David L Donoho et al. "High-dimensional data analysis: The curses and blessings of dimensionality". In: *AMS math challenges lecture* 1.2000 (2000), p. 32.

[7] Michalis Frangos, Youssef Marzouk, Karen Willcox, and Bart van Bloemen Waanders. "Surrogate and reduced-order modeling: a comparison of approaches for large-scale statistical inverse problems". In: *Large-Scale Inverse Problems and Quantification of Uncertainty* (2010), pp. 123–149.

[8] David R. Cox, Andrew M.W. Newton, and Mads Huuse. "Chapter 22 - An introduction to seismic reflection data: acquisition, processing and interpretation". In: *Regional Geology and Tectonics (Second Edition)*. Ed. by Nicola Scarselli, Jürgen Adam, Domenico Chiarella, David G. Roberts, and Albert W. Bally. Second Edition. Elsevier, 2020, pp. 571–603. ISBN: 978-0-444-64134-2. DOI: https://doi.org/10.1016/B978-0-444-64134-2.00020-1. URL: https://www.sciencedirect.com/science/article/pii/B9780444641342000201.

[9] Paul Suetens. *Fundamentals of medical imaging*. Cambridge university press, 2017.

[10] Richard Bellman and Robert Kalaba. "A mathematical theory of adaptive control processes". In: *Proceedings of the National Academy of Sciences* 45.8 (1959), pp. 1288–1290.

[11] Juan L Fernández-Martínez and Zulima Fernández-Muñiz. "The curse of dimensionality in inverse problems". In: *Journal of Computational and Applied Mathematics* 369 (2020), p. 112571.

[12] Jiaxin Zhang. "Modern Monte Carlo methods for efficient uncertainty quantification and propagation: A survey". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 13.5 (2021), e1539.

[13] Mohammad Sabokrou, Mahmood Fathy, Mojtaba Hoseini, and Reinhard Klette. "Real-time anomaly detection and localization in crowded scenes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2015, pp. 56–62.

[14] John E Dennis Jr and Robert B Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996.

[15] David Galbally, Krzysztof Fidkowski, Karen Willcox, and Omar Ghattas. "Non-linear model reduction for uncertainty quantification in large-scale inverse problems". In:

International journal for numerical methods in engineering 81.12 (2010), pp. 1581–1608.

[16] Tadeusz Burczyński. "Sensitivity analysis, optimization and inverse problems". In: *Boundary Element Advances in Solid Mechanics*. Springer, 2003, pp. 245–307.

[17] Reza Alizadeh, Janet K Allen, and Farrokh Mistree. "Managing computational complexity using surrogate models: a critical review". In: *Research in Engineering Design* 31.3 (2020), pp. 275–298.

[18] Athanasios C Antoulas, Danny C Sorensen, and Serkan Gugercin. "A survey of model reduction methods for large-scale systems". In: *Contemporary mathematics* 280 (2001), pp. 193–220.

[19] Humberto Peredo Fuentes and Manfred Zehn. "Application of the Craig-Bampton model order reduction method to a composite structure: MACco, COMAC, COMAC-S and eCOMAC". In: *Open Engineering* 6.1 (2016).

[20] Joao P Hespanha. *Linear systems theory*. Princeton university press, 2018.

[21] Alireza Aghasi, Misha Kilmer, and Eric L. Miller. "Parametric Level Set Methods for Inverse Problems". In: *SIAM Journal on Imaging Sciences* 4.2 (2011), pp. 618–650. DOI: 10.1137/100800208. eprint: https://doi.org/10.1137/100800208. URL: https://doi.org/10.1137/100800208.

[22] Benjamin Peherstorfer and Karen Willcox. "Dynamic data-driven reduced-order models". In: *Computer Methods in Applied Mechanics and Engineering* 291 (2015), pp. 21–41.

[23] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[24] Christopher JC Burges. "Geometric methods for feature extraction and dimensional reduction-a guided tour". In: *Data mining and knowledge discovery handbook* (2010), pp. 53–82.

[25] Lawrence K Saul, Kilian Q Weinberger, Fei Sha, Jihun Ham, and Daniel D Lee. "Spectral methods for dimensionality reduction". In: (2006).

[26] Thomas Simpson, Nikolaos Dervilis, and Eleni Chatzi. "Machine learning approach to model order reduction of nonlinear systems via autoencoder and LSTM networks". In: *Journal of Engineering Mechanics* 147.10 (2021), p. 04021061.

[27] Tak-chung Fu. "A review on time series data mining". In: *Engineering Applications of Artificial Intelligence* 24.1 (2011), pp. 164–181.

[28] Scott L Zeger, Rafael Irizarry, and Roger D Peng. "On time series analysis of public health and biomedical data". In: *Annu. Rev. Public Health* 27.1 (2006), pp. 57–79.

[29] Aitor Corchero Rodriguez and Mario Reyes de los Mozos. "Improving network security through traffic log anomaly detection using time series analysis". In: *Computational Intelligence in Security for Information Systems 2010: Proceedings of the 3rd International Conference on Computational Intelligence in Security for Information Systems (CISIS'10)*. Springer. 2010, pp. 125–133.

[30] Manfred Mudelsee. "Trend analysis of climate time series: A review of methods". In: *Earth-science reviews* 190 (2019), pp. 310–322.

[31] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. "Anomaly detection in crowded scenes". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010, pp. 1975–1981. DOI: 10.1109/CVPR.2010.5539872.

[32] Alan A Berryman. "The orgins and evolution of predator-prey theory". In: *Ecology* 73.5 (1992), pp. 1530–1535.

[33] Andrew J. King, Alan M. Wilson, Simon D. Wilshin, John Lowe, Hamed Haddadi, Stephen Hailes, and A. Jennifer Morton. "Selfish-herd behaviour of sheep under threat". In: *Current Biology* 22.14 (2012), R561–R562. ISSN: 0960-9822. DOI: https:

//doi.org/10.1016/j.cub.2012.05.008. URL: https://www.sciencedirect.com/science/article/pii/S0960982212005295.

[34] Mario Bertero, Patrizia Boccacci, and Christine De Mol. *Introduction to inverse problems in imaging*. CRC press, 2021.

[35] Sergey I Kabanikhin. *Inverse and ill-posed problems: theory and applications*. de Gruyter, 2011.

[36] Andy Adler and David Holder. *Electrical impedance tomography: methods, history and applications*. CRC Press, 2021.

[37] David A Boas, Dana H Brooks, Eric L Miller, Charles A DiMarzio, Misha Kilmer, Richard J Gaudette, and Quan Zhang. "Imaging the body with diffuse optical tomography". In: *IEEE signal processing magazine* 18.6 (2001), pp. 57–75.

[38] Oliver Dorn and Dominique Lesselier. "Level set methods for inverse scattering". In: *Inverse Problems* 22 (June 2006), R67. DOI: 10.1088/0266-5611/22/4/R01.

[39] Jin Cheng and Bernd Hofmann. "Regularization Methods for Ill-Posed Problems". In: *Handbook of Mathematical Methods in Imaging* 1 (2015), p. 2.

[40] Gene H Golub, Per Christian Hansen, and Dianne P O'Leary. "Tikhonov regularization and total least squares". In: *SIAM journal on matrix analysis and applications* 21.1 (1999), pp. 185–194.

[41] David Strong and Tony Chan. "Edge-preserving and scale-dependent properties of total variation regularization". In: *Inverse problems* 19.6 (2003), S165.

[42] Simon R Arridge. "Optical tomography in medical imaging". In: *Inverse problems* 15.2 (1999), R41.

[43] Sarah Fakhreddine, Jonghyun Lee, Peter Kitanidis, Scott Fendorf, and Massimo Rolle. "Imaging geochemical heterogeneities using inverse reactive transport modeling: An example relevant for characterizing arsenic mobilization and distribution". In: *Ad-*

vances in Water Resources 88 (Dec. 2015). DOI: 10.1016/j.advwatres.2015.12.005.

[44] Gad El-Qady, Mahfooz Hafez, M.A. Abdalla, and K. Ushijima. "Imaging subsurface cavities using geoelectric tomography and ground-penetrating radar". In: *Journal of Cave and Karst Studies* 67 (Dec. 2005), pp. 174–181.

[45] I.T. Rekanos, T.V. Yioultsis, and T.D. Tsiboukis. "Inverse scattering using the finite-element method and a nonlinear optimization technique". In: *IEEE Transactions on Microwave Theory and Techniques* 47.3 (1999), pp. 336–344. DOI: 10.1109/22.750236.

[46] Rainer Kress and William Rundell. "A quasi-Newton method in inverse obstacle scattering". In: *Inverse Problems* 10.5 (1994), p. 1145.

[47] Andreas Kirsch. "The domain derivative and two applications in inverse scattering theory". In: *Inverse problems* 9.1 (1993), p. 81.

[48] Fadil Santosa. "A level-set approach for inverse problems involving obstacles fadil santosa". In: *ESAIM: Control, Optimisation and Calculus of Variations* 1 (1996), pp. 17–33.

[49] Stanley Osher and James A Sethian. "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations". In: *Journal of Computational Physics* 79.1 (1988), pp. 12–49. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(88)90002-2. URL: https://www.sciencedirect.com/science/article/pii/0021999188900022.

[50] Tony F Chan and Luminita A Vese. "Active contours without edges". In: *IEEE Transactions on image processing* 10.2 (2001), pp. 266–277.

[51] Kees van den Doel and Uri M Ascher. "On level set regularization for highly ill-posed distributed parameter estimation problems". In: *Journal of Computational Physics* 216.2 (2006), pp. 707–723.

[52] Stanley J Osher and Fadil Santosa. "Level set methods for optimization problems involving geometry and constraints: I. Frequencies of a two-density inhomogeneous drum". In: *Journal of Computational Physics* 171.1 (2001), pp. 272–288.

[53] Michael McMillan, Christoph Schwarzbach, Eldad Haber, and Douglas Oldenburg. "Multiple body parametric inversion of frequency-and time-domain airborne electromagnetics". In: *SEG Technical Program Expanded Abstracts 2016*. Society of Exploration Geophysicists, 2016, pp. 846–851.

[54] Ajinkya Kadu, Tristan van Leeuwen, and Wim A. Mulder. "Salt Reconstruction in Full-Waveform Inversion With a Parametric Level-Set Method". In: *IEEE Transactions on Computational Imaging* 3.2 (2017), pp. 305–315. DOI: 10.1109/TCI.2016.2640761.

[55] GM Hoversten, C Schwarzbach, P Belliveau, Eldad Haber, and R Shekhtman. "Borehole to surface electromagnetic monitoring of hydraulic fractures". In: *79th EAGE Conference and Exhibition 2017*. Vol. 2017. European Association of Geoscientists & Engineers. 2017, pp. 1–5.

[56] Fitsum Mesadi, Mujdat Cetin, and Tolga Tasdizen. "Disjunctive normal level set: An efficient parametric implicit method". In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2016, pp. 4299–4303. DOI: 10.1109/ICIP.2016.7533171.

[57] Naren Naik, Rick Beatson, and Jerry Eriksson. "Radial-basis-function level-set-based regularized Gauss–Newton-filter reconstruction scheme for dynamic shape tomography". In: *Applied Optics* 53 (Oct. 2014). DOI: 10.1364/AO.53.006872.

[58] Zhi-Tian Niu, Hong Qi, Ze-Yu Zhu, Ke-Fu Li, Ya-Tao Ren, and Ming-Jian He. "A novel parametric level set method coupled with Tikhonov regularization for tomographic laser absorption reconstruction". In: *Applied Thermal Engineering* 201 (2022), p. 117819. ISSN: 1359-4311. DOI: https://doi.org/10.1016/j.applthermaleng.

2021.117819. URL: https://www.sciencedirect.com/science/article/pii/S1359431121012436.

[59] Haytham A. Ali and Hiroyuki Kudo. "Level-Set Method for Limited-Data Reconstruction in CT using Dictionary-Based Compressed Sensing". In: *2023 15th International Conference on Computer and Automation Engineering (ICCAE)*. 2023, pp. 264–268. DOI: 10.1109/ICCAE56788.2023.10111292.

[60] Gabriele Incorvaia and Oliver Dorn. "Stochastic Optimization Methods for Parametric Level Set Reconstructions in 2D through-the-Wall Radar Imaging". In: *Electronics* 9.12 (2020). ISSN: 2079-9292. DOI: 10.3390/electronics9122055. URL: https://www.mdpi.com/2079-9292/9/12/2055.

[61] Moshe Eliasof, Andrei Sharf, and Eran Treister. "Multimodal 3D Shape Reconstruction under Calibration Uncertainty Using Parametric Level Set Methods". In: *SIAM Journal on Imaging Sciences* 13.1 (2020), pp. 265–290. DOI: 10.1137/19M1257895. eprint: https://doi.org/10.1137/19M1257895. URL: https://doi.org/10.1137/19M1257895.

[62] Alex Hiles and Oliver Dorn. "Colour level set regularization for the electromagnetic imaging of highly discontinuous parameters in 3D*". In: *Inverse Problems in Science and Engineering* 29 (July 2020), pp. 1–36. DOI: 10.1080/17415977.2020.1797003.

[63] Hong-Kai Zhao, T. Chan, B. Merriman, and S. Osher. "A Variational Level Set Approach to Multiphase Motion". In: *Journal of Computational Physics* 127.1 (1996), pp. 179–195. ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1996.0167. URL: https://www.sciencedirect.com/science/article/pii/S0021999196901679.

[64] J. Lie, M. Lysaker, and Xue-Cheng Tai. "A binary level set model and some applications to Mumford-Shah image segmentation". In: *IEEE Transactions on Image Processing* 15.5 (2006), pp. 1171–1181. DOI: 10.1109/TIP.2005.863956.

[65] Dong Liu, Danny Smyl, and Jiangfeng Du. "Nonstationary Shape Estimation in Electrical Impedance Tomography Using a Parametric Level Set-Based Extended Kalman Filter Approach". In: *IEEE Transactions on Instrumentation and Measurement* 69.5 (2020), pp. 1894–1907. DOI: 10.1109/TIM.2019.2921441.

[66] Oguz Semerci and Eric Miller. "A Parametric Level Set Approach to Simultaneous Object Identification and Background Reconstruction for Dual Energy Computed Tomography". In: *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 21 (May 2012), pp. 2719–34. DOI: 10.1109/TIP.2012.2186308.

[67] Maokun Li, Aria Abubakar, and Tarek M. Habashy. "A Three-Dimensional Model-Based Inversion Algorithm Using Radial Basis Functions for Microwave Data". In: *IEEE Transactions on Antennas and Propagation* 60.7 (2012), pp. 3361–3372. DOI: 10.1109/TAP.2012.2196931.

[68] James D Hamilton. *Time series analysis*. Princeton university press, 2020.

[69] Zhongyang Han, Jun Zhao, Henry Leung, King Fai Ma, and Wei Wang. "A review of deep learning models for time series prediction". In: *IEEE Sensors Journal* 21.6 (2019), pp. 7833–7848.

[70] Greg Tkacz. "Neural network forecasting of Canadian GDP growth". In: *International Journal of Forecasting* 17.1 (2001), pp. 57–69.

[71] Prapanna Mondal, Labani Shit, and Saptarsi Goswami. "Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices". In: *International Journal of Computer Science, Engineering and Applications* 4.2 (2014), p. 13.

[72] Sushmitha Kothapalli and SG Totad. "A real-time weather forecasting and analysis". In: *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*. IEEE. 2017, pp. 1567–1570.

[73] Aixia Guo, Sakima Smith, Yosef M Khan, James R Langabeer II, and Randi E Foraker. "Application of a time-series deep learning model to predict cardiac dysrhythmias in electronic health records". In: *PloS one* 16.9 (2021), e0239007.

[74] Mohsen Maleki, Mohammad Reza Mahmoudi, Mohammad Hossein Heydari, and Kim-Hung Pho. "Modeling and forecasting the spread and death rate of coronavirus (COVID-19) in the world using time series models". In: *Chaos, Solitons & Fractals* 140 (2020), p. 110151.

[75] Hamid Mirshekali, Athila Q Santos, and Hamid Reza Shaker. "A survey of time-series prediction for digitally enabled maintenance of electrical grids". In: *Energies* 16.17 (2023), p. 6332.

[76] Philippe Esling and Carlos Agon. "Time-Series Data Mining". In: *ACM Computing Surveys (CSUR)* 45 (Nov. 2012), p. 12. DOI: 10.1145/2379776.2379788.

[77] Changqing Cheng, Akkarapol Sa-Ngasoongsong, Omer Beyca, Trung Le, Hui Yang, Zhenyu Kong, and Satish TS Bukkapatnam. "Time series forecasting for nonlinear and non-stationary processes: a review and comparative study". In: *Iie Transactions* 47.10 (2015), pp. 1053–1071.

[78] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. "Anomaly detection and localization in crowded scenes". In: *IEEE transactions on pattern analysis and machine intelligence* 36.1 (2013), pp. 18–32.

[79] Irfan Pratama, Adhistya Erna Permanasari, Igi Ardiyanto, and Rini Indrayani. "A review of missing values handling methods on time-series data". In: *2016 international conference on information technology systems and innovation (ICITSI)*. IEEE. 2016, pp. 1–6.

[80] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. "A review of recurrent neural networks: LSTM cells and network architectures". In: *Neural computation* 31.7 (2019), pp. 1235–1270.

[81] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. "Transformers in time series: A survey". In: *arXiv preprint arXiv:2202.07125* (2022).

[82] Giancarlo Valente, Agustin Lage Castellanos, Gianluca Vanacore, and Elia Formisano. "Multivariate linear regression of high-dimensional fMRI data with multiple target variables". In: *Human brain mapping* 35.5 (2014), pp. 2163–2177.

[83] Laurens Van Der Maaten, Eric O Postma, H Jaap Van Den Herik, et al. "Dimensionality reduction: A comparative review". In: *Journal of machine learning research* 10.66-71 (2009), p. 13.

[84] Samaneh Aminikhanghahi and Diane J Cook. "A survey of methods for time series change point detection". In: *Knowledge and information systems* 51.2 (2017), pp. 339–367.

[85] Anna Louise Schröder and Hernando Ombao. "Fresped: Frequency-specific change-point detection in epileptic seizure multi-channel eeg data". In: *Journal of the American Statistical Association* 114.525 (2019), pp. 115–128.

[86] Guangyuan Chen, Guoliang Lu, Wei Shang, and Zhaohong Xie. "Automated change-point detection of EEG signals based on structural time-series analysis". In: *IEEE Access* 7 (2019), pp. 180168–180180.

[87] Jaxk Reeves, Jien Chen, Xiaolan L Wang, Robert Lund, and Qi Qi Lu. "A review and comparison of changepoint detection techniques for climate data". In: *Journal of applied meteorology and climatology* 46.6 (2007), pp. 900–915.

[88] Kongjing Gu, Liang Yan, Xiang Li, Xiaojun Duan, and Jingjie Liang. "Change point detection in multi-agent systems based on higher-order features". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 32.11 (2022).

[89] Maksims Fiosins, Jelena Fiosina, and Jörg P Müller. "Change point analysis for intelligent agents in city traffic". In: *Agents and Data Mining Interaction: 7th International Workshop on Agents and Data Mining Interation, ADMI 2011, Taipei, Taiwan, May 2-6, 2011, Revised Selected Papers 7*. Springer. 2012, pp. 195–210.

[90] Karl L Hallgren, Nicholas A Heard, and Niall M Adams. "Changepoint detection in non-exchangeable data". In: *Statistics and Computing* 32.6 (2022), p. 110.

[91] Kolbjørn Tunstrøm, Yael Katz, Christos Ioannou, Cristian Huepe, Matthew Lutz, and Iain Couzin. "Collective States, Multistability and Transitional Behavior in Schooling Fish". In: *PLoS computational biology* 9 (Feb. 2013), e1002915. DOI: 10.1371/journal.pcbi.1002915.

[92] Shaochuan Lu. "Bayesian multiple changepoint detection with missing data and its application to the magnitude-frequency distributions". In: *Environmetrics* 34.4 (2023), e2775. DOI: https://doi.org/10.1002/env.2775. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/env.2775. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/env.2775.

[93] Ege Ozsar, Misha Kilmer, Eric de Sturler, Arvind K Saibaba, and Eric Miller. "Parametric level-sets enhanced to improve reconstruction (PaLEnTIR)". In: *Inverse Problems* 41.2 (Jan. 2025), p. 025004. DOI: 10.1088/1361-6420/ada798. URL: https://dx.doi.org/10.1088/1361-6420/ada798.

[94] Ralph A. Willoughby. "Solutions of Ill-Posed Problems (A. N. Tikhonov and V. Y. Arsenin)". In: *SIAM Review* 21.2 (1979), pp. 266–267. DOI: 10.1137/1021044. eprint: https://doi.org/10.1137/1021044. URL: https://doi.org/10.1137/1021044.

[95] M Ben, Mohamed Khames Ben Hadj Miled, and Eric Miller. "A projection-based level-set approach to enhance conductivity anomaly reconstruction in electrical resistance tomography". In: *Inverse Problems* 23 (Dec. 2007), pp. 2375–2400. DOI: 10.1088/0266-5611/23/6/007.

[96] Rui Guo, Zekui Jia, Xiaoqian Song, Maokun Li, Fan Yang, Shenheng Xu, and Aria Abubakar. "Pixel- and Model-Based Microwave Inversion With Supervised Descent Method for Dielectric Targets". In: *IEEE Transactions on Antennas and Propagation* 68.12 (2020), pp. 8114–8126. DOI: 10.1109/TAP.2020.2999741.

[97] Rui Li, Yongfu Zhang, Lihui Peng, and Xinghe Liao. "An Image Reconstruction For Electrical Capacitance Tomography Using Parametric Level Set". In: *2020 5th International Conference on Computer and Communication Systems (ICCCS)*. 2020, pp. 384–390. DOI: 10.1109/ICCCS49078.2020.9118589.

[98] Dong Liu, Anil Kumar Khambampati, and Jiangfeng Du. "A Parametric Level Set Method for Electrical Impedance Tomography". In: *IEEE Transactions on Medical Imaging* 37.2 (2018), pp. 451–460. DOI: 10.1109/TMI.2017.2756078.

[99] Eric de Sturler and Misha Kilmer. "A Regularized Gauss–Newton Trust Region Approach to Imaging in Diffuse Optical Tomography". In: *SIAM J. Scientific Computing* 33 (Jan. 2011), pp. 3057–3086. DOI: 10.1137/100798181.

[100] Oliver Dorn and Dominique Lesselier. "Level Set Techniques For Structural Inversion In Medical Imaging". In: *Deformable Models: Biomedical and Clinical Applications*. New York, NY: Springer New York, 2007, pp. 61–90. ISBN: 978-0-387-68413-0. DOI: 10.1007/978-0-387-68413-0_3. URL: https://doi.org/10.1007/978-0-387-68413-0_3.

[101] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 2000. DOI: 10.1137/1.9780898719857. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9780898719857. URL: https://epubs.siam.org/doi/abs/10.1137/1.9780898719857.

[102] Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. Society for Industrial and Applied Mathematics, 1998. DOI: 10.1137/1.9780898719697. eprint:

https://epubs.siam.org/doi/pdf/10.1137/1.9780898719697. URL: https://epubs.siam.org/doi/abs/10.1137/1.9780898719697.

[103] Diederik P Kingma. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[104] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." In: *Journal of machine learning research* 12.7 (2011).

[105] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[106] Bingbing Li. "A Principal Component Analysis Approach to Noise Removal for Speech Denoising". In: *2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*. 2018, pp. 429–432. DOI: 10.1109/ICVRIS.2018.00111.

[107] Daniel Engel, Lars Hüttenberger, and Bernd Hamann. "A survey of dimension reduction methods for high-dimensional data analysis and visualization". In: *Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering-Proceedings of IRTG 1131 Workshop 2011*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik. 2012.

[108] Jinbo Guo, Feng Li, Zhonghua Yu, and Linsheng Chen. "A novel dimensionality reduction method based on flow model". In: *Neurocomputing* 599 (2024), p. 128066. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2024.128066. URL: https://www.sciencedirect.com/science/article/pii/S0925231224008373.

[109] John A Lee, Michel Verleysen, et al. *Nonlinear dimensionality reduction*. Vol. 1. Springer, 2007.

[110] Michael Greenacre, Patrick JF Groenen, Trevor Hastie, Alfonso Iodice d'Enza, Angelos Markos, and Elena Tuzhilina. "Principal component analysis". In: *Nature Reviews Methods Primers* 2.1 (2022), p. 100.

[111] Ian T Jolliffe and Jorge Cadima. "Principal component analysis: a review and recent developments". In: *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202.

[112] Chuang Wang, Tianyi Zhang, Jing Jia, Jin Wang, and Shan Ren. "Complex Network-Based Resilience Assessment of the Integrated Circuit Industry Chain". In: *Sustainability* 16 (June 2024), p. 5163. DOI: 10.3390/su16125163.

[113] Yasi Wang, Hongxun Yao, and Sicheng Zhao. "Auto-encoder based dimensionality reduction". In: *Neurocomputing* 184 (2016), pp. 232–242.

[114] Pierre Baldi. "Autoencoders, unsupervised learning, and deep architectures". In: *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings. 2012, pp. 37–49.

[115] Wei Wang, Yan Huang, Yizhou Wang, and Liang Wang. "Generalized autoencoder: A neural network framework for dimensionality reduction". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014, pp. 490–497.

[116] Pengzhi Li, Yan Pei, and Jianqiang Li. "A comprehensive survey on design and application of autoencoder in deep learning". In: *Applied Soft Computing* 138 (2023), p. 110176.

[117] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. "A comprehensive survey of loss functions in machine learning". In: *Annals of Data Science* (2020), pp. 1–26.

[118] Mohamad H Hassoun. *Fundamentals of artificial neural networks*. MIT press, 1995.

[119] Jinming Zou, Yi Han, and Sung-Sau So. "Overview of artificial neural networks". In: *Artificial neural networks: methods and applications* (2009), pp. 14–22.

[120] Yann LeCun, Y. Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521 (May 2015), pp. 436–44. DOI: 10.1038/nature14539.

[121] Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. "Multilayer perceptron and neural networks". In: *WSEAS Transactions on Circuits and Systems* 8.7 (2009), pp. 579–588.

[122] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. "State-of-the-art in artificial neural network applications: A survey". In: *Heliyon* 4.11 (2018).

[123] Larry R Medsker, Lakhmi Jain, et al. "Recurrent neural networks". In: *Design and Applications* 5.64-67 (2001), p. 2.

[124] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. "Convolutional, long short-term memory, fully connected deep neural networks". In: *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. Ieee. 2015, pp. 4580–4584.

[125] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. "A survey of convolutional neural networks: analysis, applications, and prospects". In: *IEEE transactions on neural networks and learning systems* 33.12 (2021), pp. 6999–7019.

[126] Michael J Evans and Jeffrey S Rosenthal. *Probability and statistics: The science of uncertainty*. Macmillan, 2004.

[127] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[128] Xinhua Zhuang, Yan Huang, Kannappan Palaniappan, and Yunxin Zhao. "Gaussian mixture density modeling, decomposition, and applications". In: *IEEE Transactions on Image Processing* 5.9 (1996), pp. 1293–1302.

[129] Alan Julian Izenman. "Review papers: Recent developments in nonparametric density estimation". In: *Journal of the american statistical association* 86.413 (1991), pp. 205–224.

[130] David Freedman and Persi Diaconis. “On the histogram as a density estimator: L 2 theory”. In: *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 57.4 (1981), pp. 453–476.

[131] Anja Heim, Eduard Gröller, and Christoph Heinzl. “AccuStripes: Adaptive Binning for the Visual Comparison of Univariate Data Distributions”. In: *arXiv preprint arXiv:2207.13663* (2022).

[132] Kostantinos N Plataniotis and Dimitris Hatzinakos. “Gaussian mixtures and their applications to signal processing”. In: *Advanced signal processing handbook* (2017), pp. 89–124.

[133] Arthur P Dempster, Nan M Laird, and Donald B Rubin. “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the royal statistical society: series B (methodological)* 39.1 (1977), pp. 1–22.

[134] Francesco Ginelli, Fernando Peruani, Marie-Helène Pillot, Hugues Chaté, Guy Theraulaz, and Richard Bon. “Intermittent collective dynamics emerge from conflicting imperatives in sheep herds”. In: *Proceedings of the National Academy of Sciences* 112.41 (2015), pp. 12729–12734.

[135] Thomas E Angelini, Edouard Hannezo, Xavier Trepat, Manuel Marquez, Jeffrey J Fredberg, and David A Weitz. “Glass-like dynamics of collective cell migration”. In: *Proceedings of the National Academy of Sciences* 108.12 (2011), pp. 4714–4719.

[136] Dirk Helbing, Illés Farkas, and Tamas Vicsek. “Simulating dynamical features of escape panic”. In: *Nature* 407.6803 (2000), pp. 487–490.

[137] Yann-Edwin Keta, Étienne Fodor, Frédéric van Wijland, Michael E Cates, and Robert L Jack. “Collective motion in large deviations of active particles”. In: *Physical Review E* 103.2 (2021), p. 022603.

[138] Yanyan Shi, Xiaolong Kong, Meng Wang, Yuehui Wu, and Lan Yang. "A Non-Convex L1-Norm Penalty-Based Total Generalized Variation Model for Reconstruction of Conductivity Distribution". In: *IEEE Sensors Journal* 20.14 (2020), pp. 8137–8146. DOI: 10.1109/JSEN.2020.2981873.

[139] Michael Yu Wang and Xiaoming Wang. ""Color" level sets: a multi-phase method for structural topology optimization with multiple materials". In: *Computer Methods in Applied Mechanics and Engineering* 193.6 (2004), pp. 469–496. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2003.10.008. URL: https://www.sciencedirect.com/science/article/pii/S0045782503005644.

[140] G. Allaire, C. Dapogny, G. Delgado, and G. Michailidis. "Multi-phase structural optimization via a level set method". In: *ESAIM: Control, Optimisation and Calculus of Variations* 20.2 (2014), pp. 576–611. DOI: 10.1051/cocv/2013076.

[141] R. Keys. "Cubic convolution interpolation for digital image processing". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 29.6 (1981), pp. 1153–1160. DOI: 10.1109/TASSP.1981.1163711.

[142] Francois Lekien and J Marsden. "Tricubic interpolation in three dimensions". In: *International Journal for Numerical Methods in Engineering* 63.3 (2005), pp. 455–471.

[143] Stephen J. Wright Jorge Nocedal. *Numerical Optimization*. Second. Springer New York, NY, 2006. DOI: 10.1007/978-0-387-40065-5.

[144] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Second. Society for Industrial and Applied Mathematics, 2002. DOI: 10.1137/1.9780898718027. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9780898718027. URL: https://epubs.siam.org/doi/abs/10.1137/1.9780898718027.

[145] James W. Demmel. *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997, pp. xii+419. ISBN: 0-89871-389-

7. DOI: 10.1137/1.9781611971446. URL: https://doi.org/10.1137/1.9781611971446.

[146] Tobias Lindstrøm Jensen, Jakob Heide Jørgensen, Per Christian Hansen, and Søren Holdt Jensen. *TVReg*. English. 2010.

[147] Per Christian Hansen. "The L-Curve and its Use in the Numerical Treatment of Inverse Problems". In: *Computational Inverse Problems in Electrocardiology*. WIT Press, 2001, pp. 119–142.

[148] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.

[149] A Saltelli. *Global Sensitivity Analysis: the Primer*. John Wiley & Sons, 2008.

[150] Tatiana A. Bubba, Markus Juvonen, Jonatan Lehtonen, Maximilian März, Alexander Meaney, Zenith Purisha, and Samuli Siltanen. "Tomographic X-ray data of carved cheese". In: *arXiv* abs/1705.05732 (2017).

[151] Tatiana A. Bubba, Andreas Hauptmann, Simo Huotari, Juho Rimpeläinen, and Samuli Siltanen. "Tomographic X-ray data of a lotus root filled with attenuating objects". In: *ArXiv* abs/1609.07299 (2016).

[152] Vyacheslav Ivanovich Lebedev. "Quadratures on a sphere". In: *USSR Computational Mathematics and Mathematical Physics* 16.2 (1976), pp. 10–24.

[153] Eric de Sturler, Serkan Gugercin, Misha E. Kilmer, Saifon Chaturantabut, Christopher Beattie, and Meghan O'Connell. "Nonlinear Parametric Inversion Using Interpolatory Model Reduction". In: *SIAM J. Sci. Comput* 37(3) (2015), B495–B517.

[154] Selin S. Aslan, Eric de Sturler, and Misha E. Kilmer. "Randomized approach to nonlinear inversion combining random and optimized simultaneous sources and detectors". In: *SIAM J. Sci. Comput* 41 (2019), B229–B249.

[155] Athanasios Constantinos Antoulas, Christopher Andrew Beattie, and Serkan Güğercin. *Interpolatory methods for model reduction*. SIAM, 2020.

[156] Qianqian Fang et al. "Combined optical imaging and mammography of the healthy breast: optical contrast derived from breast structure and compression". In: *Medical Imaging, IEEE Transactions on* 28.1 (2009), pp. 30–42.

[157] Fridrik Larusson, Sergio Fantini, and Eric L Miller. "Parametric level set reconstruction methods for hyperspectral diffuse optical tomography". In: *Biomedical optics express* 3.5 (2012), pp. 1006–1024.

[158] Arvind K. Saibaba, Pranjal Prasad, Eric de Sturler, Eric Miller, and Misha E. Kilmer. "Randomized approaches to accelerate MCMC algorithms for Bayesian inverse problems". In: *Journal of Computational Physics* 440 (2021), p. 110391. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2021.110391. URL: https://www.sciencedirect.com/science/article/pii/S0021999121002862.

[159] Alexandre Baussard, Eric L Miller, and Denis Prémel. "Adaptive B-spline scheme for solving an inverse scattering problem". In: *Inverse Problems* 20.2 (Jan. 2004), p. 347. DOI: 10.1088/0266-5611/20/2/003. URL: https://dx.doi.org/10.1088/0266-5611/20/2/003.

[160] Alexandre Baussard, Eric L Miller, and Dominique Lesselier. "Adaptive multiscale reconstruction of buried objects". In: *Inverse Problems* 20.6 (Nov. 2004), S1. DOI: 10.1088/0266-5611/20/6/S01. URL: https://dx.doi.org/10.1088/0266-5611/20/6/S01.

[161] Bhuwan Kashyap and Ratnesh Kumar. "Sensing methodologies in agriculture for soil moisture and nutrient monitoring". In: *IEEE Access* 9 (2021), pp. 14095–14121.

[162] Syahidah Nurani Zulkifli, Herlina Abdul Rahim, and Woei-Jye Lau. "Detection of contaminants in water supply: A review on state-of-the-art monitoring technologies

and their applications". In: *Sensors and Actuators B: Chemical* 255 (2018), pp. 2657–2689.

[163] M Lorenz, G Becher, V Mues, and E Ulrich. "Monitoring forest condition in Europe: concentrations of nitrogen and sulphur in bulk deposition and defoliation of main tree species". In: *International Journal of Environmental Studies* 65.3 (2008), pp. 299–309.

[164] Philip W Rundel, Eric A Graham, Michael F Allen, Jason C Fisher, and Thomas C Harmon. "Environmental sensor networks in ecological research". In: *New Phytologist* 182.3 (2009), pp. 589–607.

[165] Daniel B. Stover, Frank P. Day, John R. Butnor, and Bert G. Drake. "EFFECT OF ELEVATED CO2ON COARSE-ROOT BIOMASS IN FLORIDA SCRUB DETECTED BY GROUND-PENETRATING RADAR". In: *Ecology* 88.5 (May 2007), pp. 1328–1334. ISSN: 0012-9658. DOI: 10.1890/06-0989. URL: http://dx.doi.org/10.1890/06-0989.

[166] Narayanan Sadagopan and Bhaskar Krishnamachari. "Maximizing data extraction in energy-limited sensor networks". In: *International Journal of Distributed Sensor Networks* 1.1 (2005), pp. 123–147.

[167] Bruce M Howe et al. "SMART cables for observing the global ocean: science and implementation". In: *Frontiers in Marine Science* 6 (2019), p. 424.

[168] Numa J Bertola, Alberto Costa, and Ian FC Smith. "Strategy to validate sensor-placement methodologies in the context of sparse measurement in complex urban systems". In: *IEEE Sensors Journal* 20.10 (2020), pp. 5501–5509.

[169] Stephen Lofts, David J Spurgeon, Claus Svendsen, and Edward Tipping. "Deriving soil critical limits for Cu, Zn, Cd, and Pb: a method based on free ion concentrations". In: *Environmental Science & Technology* 38.13 (2004), pp. 3623–3631.

[170] Lawrence C Evans. *Partial differential equations*. Vol. 19. American Mathematical Society, 2022.

[171] Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. "Tensor decompositions for temporal knowledge base completion". In: *arXiv preprint arXiv:2004.04926* (2020).

[172] W. Rudin. *Principles of Mathematical Analysis*. International series in pure and applied mathematics. McGraw-Hill, 1976. ISBN: 9780070856134. URL: https://books.google.com/books?id=kwqzPAAACAAJ.

[173] Suraj Shankar. *Diffusion in 1D and 2D*. MATLAB Central File Exchange. 2025. URL: https://www.mathworks.com/matlabcentral/fileexchange/38088-diffusion-in-1d-and-2d.

[174] Reza Habibi. "Bayesian online change point detection in finance". In: *Financial Internet Quarterly* 17.4 (2021), pp. 27–33.

[175] Xiangyun Gao, Wei Fang, Feng An, and Yue Wang. "Detecting method for crude oil price fluctuation mechanism under different periodic time series". In: *Applied energy* 192 (2017), pp. 201–212.

[176] Feiyu Jiang, Zifeng Zhao, and Xiaofeng Shao. "Time series analysis of COVID-19 infection curve: A change-point perspective". In: *Journal of econometrics* 232.1 (2023), pp. 1–17.

[177] Kyle R Anderson and Michael P Poland. "Bayesian estimation of magma supply, storage, and eruption rates using a multiphysical volcano model: Kīlauea Volcano, 2000–2012". In: *Earth and Planetary Science Letters* 447 (2016), pp. 161–171.

[178] Eliezer Gurarie, Russel D Andrews, and Kristin L Laidre. "A novel method for identifying behavioural changes in animal movement data". In: *Ecology letters* 12.5 (2009), pp. 395–408.

[179] Tengyao Wang and Richard J Samworth. "High dimensional change point estimation via sparse projection". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 80.1 (2018), pp. 57–83.

[180] Michele Basseville, Igor V Nikiforov, et al. *Detection of abrupt changes: theory and application*. Vol. 104. Prentice hall Englewood Cliffs, 1993.

[181] Yao Xie, Jiaji Huang, and Rebecca Willett. "Change-Point Detection for High-Dimensional Time Series With Missing Data". In: *IEEE Journal of Selected Topics in Signal Processing* 7.1 (2013), pp. 12–27. DOI: 10.1109/JSTSP.2012.2234082.

[182] Muktesh Gupta, Rajesh Wadhvani, and Akhtar Rasool. "Real-time Change-Point Detection: A deep neural network-based adaptive approach for detecting changes in multivariate time series data". In: *Expert Systems with Applications* 209 (2022), p. 118260. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.118260. URL: https://www.sciencedirect.com/science/article/pii/S0957417422014026.

[183] Amal Tawakuli, Bastian Havers, Vincenzo Gulisano, Daniel Kaiser, and Thomas Engel. "Survey:Time-series data preprocessing: A survey and an empirical analysis". In: *Journal of Engineering Research* (2024). ISSN: 2307-1877. DOI: https://doi.org/10.1016/j.jer.2024.02.018. URL: https://www.sciencedirect.com/science/article/pii/S2307187724000452.

[184] Walter Scheirer, Anderson Rocha, Ross Micheals, and Terrance Boult. "Robust Fusion: Extreme Value Theory for Recognition Score Normalization". In: *Computer Vision – ECCV 2010*. Ed. by Kostas Daniilidis, Petros Maragos, and Nikos Paragios. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 481–495. ISBN: 978-3-642-15558-1.

[185] Herve Abdi, Lynne J Williams, et al. "Normalizing data". In: *Encyclopedia of research design* 1 (2010), pp. 935–938.

[186] Anil Jain, Karthik Nandakumar, and Arun Ross. "Score normalization in multimodal biometric systems". In: *Pattern Recognition* 38.12 (2005), pp. 2270–2285. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2005.01.012. URL: https://www.sciencedirect.com/science/article/pii/S0031320305000592.

[187] B.M. Bolstad, R.A Irizarry, M. Åstrand, and T.P. Speed. "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias". In: *Bioinformatics* 19.2 (Jan. 2003), pp. 185–193. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/19.2.185. eprint: https://academic.oup.com/bioinformatics/article-pdf/19/2/185/48903970/bioinformatics_19_2_185.pdf. URL: https://doi.org/10.1093/bioinformatics/19.2.185.

[188] Dhammika Amaratunga and Javier Cabrera and. "Analysis of Data From Viral DNA Microchips". In: *Journal of the American Statistical Association* 96.456 (2001), pp. 1161–1170. DOI: 10.1198/016214501753381814. eprint: https://doi.org/10.1198/016214501753381814. URL: https://doi.org/10.1198/016214501753381814.

[189] Jiawei Han. *Data Mining Concepts and Techniques*. eng. 4th edition. Cambridge, MA, United States: Elsevier, 2023. ISBN: 0128117613.

[190] Ryan Prescott Adams and David JC MacKay. "Bayesian online changepoint detection". In: *arXiv preprint arXiv:0710.3742* (2007).

[191] Denis Volkhonskiy, Evgeny Burnaev, Ilia Nouretdinov, Alexander Gammerman, and Vladimir Vovk. "Inductive conformal martingales for change-point detection". In: *Conformal and Probabilistic Prediction and Applications*. PMLR. 2017, pp. 132–153.

[192] James R Norris. *Markov chains*. 2. Cambridge university press, 1998.

[193] Zhipeng Wang and David W. Scott. "Nonparametric density estimation for high-dimensional data—Algorithms and applications". In: *WIREs Computational Statistics* 11.4 (2019), e1461. DOI: https://doi.org/10.1002/wics.1461. eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.1461. URL: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.1461.

[194] Yatin Dandi, Ludovic Stephan, Florent Krzakala, Bruno Loureiro, and Lenka Zdeborová. "Universality laws for gaussian mixtures in generalized linear models". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 54754–54768.

[195] Andrew A Neath and Joseph E Cavanaugh. "The Bayesian information criterion: background, derivation, and applications". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 4.2 (2012), pp. 199–203.

[196] Zahra Zohrevand and Uwe Glässer. "Should i raise the red flag? A comprehensive survey of anomaly scoring methods toward mitigating false alarms". In: *arXiv preprint arXiv:1904.06646* (2019).

[197] Qihang Zhou, Shibo He, Haoyu Liu, Jiming Chen, and Wenchao Meng. "Label-free multivariate time series anomaly detection". In: *IEEE Transactions on Knowledge and Data Engineering* 36.7 (2024), pp. 3166–3179.

[198] Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. "Novel type of phase transition in a system of self-driven particles". In: *Physical review letters* 75.6 (1995), p. 1226.

[199] Hideyuki Miyahara, Hyu Yoneki, and Vwani Roychowdhury. "Vicsek model meets DBSCAN: Cluster phases in the Vicsek model". In: *arXiv preprint arXiv:2307.12538* (2023).

[200] Pei Tan and Christopher E Miles. "Intrinsic statistical separation of subpopulations in heterogeneous collective motion via dimensionality reduction". In: *Physical Review E* 109.1 (2024), p. 014403.

[201] Julien Audibert, Frédéric Guyard, Sébastien Marti, and Maria Zuluaga. "USAD: UnSupervised Anomaly Detection on Multivariate Time Series". In: Aug. 2020, pp. 3395–3404. DOI: 10.1145/3394486.3403392.

[202] Zheng Xu, Yumeng Yang, Xinwen Gao, and Min Hu. "DCFF-MTAD: A Multivariate Time-Series Anomaly Detection Model Based on Dual-Channel Feature Fusion". In: *Sensors* 23.8 (2023). ISSN: 1424-8220. DOI: 10.3390/s23083910. URL: https://www.mdpi.com/1424-8220/23/8/3910.

[203] Shreshth Tuli, Giuliano Casale, and Nicholas Jennings. "TranAD: deep transformer networks for anomaly detection in multivariate time series data". In: *Proceedings of the VLDB Endowment* 15 (Feb. 2022), pp. 1201–1214. DOI: 10.14778/3514061.3514067.

[204] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. "Robust anomaly detection for multivariate time series through stochastic recurrent neural network". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2828–2837.

[205] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. "Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding". In: July 2018, pp. 387–395. DOI: 10.1145/3219819.3219845.

[206] Brian Pollack, Saptaparna Bhattacharya, and Michael Schmitt. "Bayesian block histogramming for high energy physics". In: *arXiv preprint arXiv:1708.00810* (2017).

[207] Meghan O'Connell, Misha E. Kilmer, Eric de Sturler, and Serkan Gugercin. "Computing Reduced order Models via Inner-Outer Krylov Recycling in Diffuse Optical Tomography". In: *SIAM J. Sci. Comput* 39.2 (2017), pp. 272–297.