

Towards Decision Support and Automation for Safety Critical Ultrasonic Nondestructive Evaluation Data Analysis

by

Nicholas Anthony Torenvliet

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2025

© Nicholas Anthony Torenvliet 2025

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Jamie Baalis Coble
 Professor, Dept. Nuclear Engineering,
 University of Tennessee

Supervisor: John Zelek
 Professor, Dept. of Systems Design Engineering,
 University of Waterloo

Internal Member: Paul Fieguth
 Professor, Dept. of Systems Design Engineering,
 University of Waterloo

Internal Member: Alexander Wong
 Professor, Dept. of Systems Design Engineering,
 University of Waterloo

Internal-External Member: Mahesh Pandey
 Professor, Dept. of Civil and Environmental Engineering,
 University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

A set of machine learning techniques that provide decision support and automation to the analysis of data taken during ultrasonic non-destructive evaluation of Canada Deuterium Uranium reactor pressure tubes is proposed. Data analysis is carried out primarily to identify and characterizes the geometry of flaws or defects on the pressure tube inner diameter surface.

A baseline approach utilizing a variational auto-encoder ranks data by likelihood and performs analysis using Nominal Profiling (NPROF), a novel technique that characterize the very likely nominal component of the dataset and determines variance from it. While effective, the baseline method expresses limitations, including sensitivity to outliers, challenged explainability, and the absence of a strong fault diagnosis and error remediation mechanism.

To address these shortcomings, Diffusion Partition Consensus (DiffPaC), a novel method integrating Conditional Score-Based Diffusion with Savitzky-Golay Filters, is proposed. The approach includes a mechanism for outlier removal during training that reliably improves model performance. It also features strong explainability and, with a human in the loop, mechanisms for fault diagnosis and error correction. These features advance applicability in safety-critical contexts such as nuclear nondestructive evaluation.

Methods are integrated and scaled to provide: (a) a principled probabilistic performance model, (b) enhanced explainability through interpretable outputs, (c) fault diagnosis and error correction with a human-in-the-loop, (e) independence from dataset curation and out-of-distribution generalization (f) strong preliminary results that meet accuracy requirements on dimensional estimates as specified by the regulator in [150].

Though not directly comparable, the integrated set of methods makes many qualitative improvements upon prior work, which is largely based on discriminative methods or heuristics. And whose results rely on data annotation, pre-processing, parameter selection, and out of distribution generalization. In regard to these, the integrated set of fully learned data driven methods may be considered state of the art for applications in this niche context.

The probabilistic model, and corroborating results, imply a principled basis underlying model behaviors and provide a means to interface with regulatory bodies seeking some justification for usage of novel methods in safety critical contexts. The process is largely autonomous, but may include a human in the loop for fail-safe analysis. The integrated methods make a significant step forward in applying machine learning in this safety-critical context. And provide a state-of-the-art proof of concept, or minimum viable product, upon which a new and fully refactored process for utility owner operators may be developed.

Acknowledgements

Particular acknowledgement and thanks to Bruce Power LLP. my corporate sponsor, and John Zelek, my thesis advisor.

Dedication

To my mom, dad, friend, and daughter, kind souls, good spirits, and able educators all.

Table of Contents

Examining Committee	ii
Author’s Declaration	iii
Abstract	iv
Acknowledgements	v
Dedication	vi
List of Figures	xi
List of Tables	xiii
List of Symbols	xiv
1 Introduction	1
1.1 Motivation and Problem Statement	3
1.2 Contributions	4
1.3 Arc of research	5
1.4 Thesis organization	6

2	Requirements, design choices, and scope	8
2.1	Design choices	9
2.2	Complete analysis	11
2.3	Summary and scope	13
3	CANDU pressure tube inspection	14
3.1	CANDU pressure tubes	14
3.2	Deformation and flaws	16
3.3	Pressure tube oxidation and sediment	20
3.4	Inspections, tooling, and data acquisition	20
3.4.1	UT probe configuration	22
3.5	UT data	23
3.6	UT data analysis	26
3.7	Selected bibliography for CANDU component and inspection processes . .	28
3.8	Chapter summary	29
4	Literature review	30
4.1	CANDU reactor innovation	30
4.2	Generative methods and models	32
4.3	Disentanglement and manifolds	33
4.4	Anomaly detection	35
4.5	Learning the nominal	36
4.6	Other methods informing DiffPaC	37
5	Datasets	39
5.1	Real world datasets	39
5.2	Synthetic datasets	41
5.3	Ranking dataset complexity	41
5.4	Nomenclature, syntax, symbols, and convention	42
5.5	Chapter summary	44

6	VAE based analysis	45
6.1	A probabilistic model of data analysis	46
6.2	Method description	46
6.3	MSE based data likelihood ranking	47
6.4	Taking estimates	49
6.5	Nominal profiling	49
6.6	Results and discussion	50
6.7	Conclusion	55
7	Diffusion ablation	56
7.1	Analysis and the manifold	57
7.2	A probabilistic model for UT data analysis	58
7.3	Diffusion Probabilistic Models	60
7.4	Experiments and results	63
7.5	Practical application and future work	68
7.6	Conclusions	71
8	Diffusion Partition Consensus based analysis	72
8.1	Learning nominal criteria	75
8.2	Analysis and the manifold	75
8.3	Algorithm	79
8.4	Termination criteria and fault diagnosis	81
8.5	Partition based data biasing	82
8.6	Results	86
8.7	Discussion	90
9	Recommendations for implementation and commissioning	91
9.1	Process implementation at scale	91
9.2	Elements required for regulatory approval	92
9.3	Chapter summary	94

10 Conclusions	95
10.1 Key contributions	96
10.2 Revisiting requirements and design choices	97
10.3 Limitations	98
10.3.1 Formal evaluation of fault diagnostic	98
10.3.2 Establishing probability of detection	98
10.4 Future work	99
10.4.1 Implementation and regulatory acceptance	99
10.4.2 Integrating discriminative methods over time	99
10.4.3 Multidimensional/multivariate applications	99
10.5 Conclusions	100
References	102
APPENDICES	117
A Diffusion model analysis	118
B DiffPaC implementation	124
C Synthetic data gen	125
Glossary	129

List of Figures

3.1	CANDU reactor schematic diagram.	15
3.2	Fuel Channel Assembly schematic diagram.	17
3.3	Pressure Tube rolled joint detail.	18
3.4	Fuel Channel Assembly and configurations.	19
3.5	Typical ultrasonic inspection tooling.	21
3.6	Canonical views of ultrasonic data.	24
3.7	Flaw depth determination.	26
3.8	Simplified inspection tooling system schematic.	28
5.1	Calibration fixture ring of notches.	40
5.2	Synthetic datasets.	43
6.1	M_V^A histograms and thresholds.	48
6.2	Type i/ii/iii analysis.	51
6.3	Dimensional estimates with required accuracy bars.	53
6.4	Error, confidence, and nominal support measures.	54
7.1	No clear path to fault diagnosis.	59
7.2	CSDI vs. DDPM error curves.	64
7.3	Growth in set cardinality.	65
7.4	Error with increasing dataset size.	67
7.5	$ \hat{X} - M_\alpha X[Q] $ Error histograms.	69

7.6	Wig-wag and nominal support.	70
8.1	Converging nominal inference band.	76
8.2	Divergent response to data miss-classification.	77
8.3	DiffPaC training and error convergence.	80
8.4	DiffPaC results for synthetic dataset (16,16).	83
8.5	DiffPaC results for CF dataset.	84
8.6	DiffPaC results for HSA dataset.	85
8.7	DiffPaC HSA dataset results detail.	88
8.8	DiffPaC synthetic dataset (16,16) fault diagnosis detail.	89
9.1	Integrated system flow chart.	93

List of Tables

6.1	Dimensional estimates, specified dimensions, and required accuracies. . . .	55
8.1	DiffPaC results for synthetic and CF datasets.	86
A.1	10% Missing data ablation table.	119
A.2	20% Missing data ablation table.	120
A.3	30% Missing data ablation table.	121
A.4	40% Missing data ablation table.	122
A.5	50% Missing data ablation table.	123

List of Symbols

$a \in \mathbb{R}^q$: an *ascan* is an array of $q \in \mathbb{Z}^+$ digitized voltage response samples.

$A \in \mathbb{R}^{n \times q}$: an ordered set of $n \in \mathbb{Z}^+$ *ascans* a .

$A' \in \mathbb{R}^{n \times q}$: denoised A .

$A'' \in \mathbb{R}^{n \times q}$: type i/ii data deviation information for A .

$d_X \in \mathbb{R}^n$: convenience statistic $d_X = |X - M_\alpha(X[P])|$.

$d_{X_D} \in \mathbb{R}^n$: convenience statistic $d_{X_D} = |X - M_\alpha(X_D^m[F])|$.

D : a Conditional Score Based Diffusion model.

$F \in \{0, 1\}^n$: boolean partition vector $F = I \wedge P$.

$G \in \{0, 1\}^n$: in the implementation of CSDI, G is a boolean vector defining a ground truth mask that holds back nominally classified data for training the conditional inference mechanism.

$I \in \{0, 1\}^n$: boolean partition vector $I = \neg P \vee (P \wedge \neg G)$.

$leftShift(A, X)$, $A \in \mathbb{R}^{n \times q}$, $X \in \mathbb{R}^n$: shifts the elements of each row in A left by the value at the corresponding row in X .

$M_V^A \in \mathbb{R}^n$: the mean squared error of variational auto-encoder V trained on, and reconstructing, dataset A .

$M_V^{A,p} \in \mathbb{R}$: the p^{th} percentile value of M_V^A , for p in the range $[0, 1]$.

$M_\alpha(U[B])$: let $U \in \mathbb{R}^n$ be a univariate dataset partitioned by arbitrary boolean vector $B \in \mathbb{R}^n$. Then M_α is a Savitzky-Golay filter that uses linear interpolation to fill regions at $U[\neg B]$, and is parameterized by $\alpha = (window_size = 600, order = 5)$, such that $M_\alpha(U[B]) \in \mathbb{R}^n$.

$M_\beta(U[B])$: let $U \in \mathbb{R}^n$ be a univariate dataset partitioned by arbitrary boolean vector $B \in \mathbb{R}^n$. Then M_β is a Savitzky-Golay filter that uses linear interpolation to fill regions at $U[\neg B]$, and is parameterized by $\beta = (window_size = 100, order = 8)$, such that $M_\beta(U[B]) \in \mathbb{R}^n$.

$M_{\alpha\beta} \in \mathbb{R}^{8 \times n}$: an array of biased predictors of the nominal component in dataset X such that $M_\alpha(X[Q])$.

$$M_{\alpha\beta} := \{M_\alpha(X[P]), M_\alpha(X[F]), M_\alpha(X_D^m[I]), M_\alpha(X_D^m[F]), M_\beta(X[P]), M_\beta(X[F]), M_\beta(X_D^m[I]), M_\beta(X_D^m[F])\}$$

$M_{\alpha\beta}^s \in \mathbb{R}^n$: the row wise standard deviation $M_{\alpha\beta}^s = M_{\alpha\beta}.st(1)$.

$M_{\alpha\beta}^{sm} \in \mathbb{R}$: the statistic $M_{\alpha\beta}^{sm} = M_{\alpha\beta}^s.mn()$.

$M_{\alpha\beta}^{3s} \in \mathbb{R}$: the statistic $M_{\alpha\beta}^{3s} = M_{\alpha\beta}^{sm} + 3 \cdot M_{\alpha\beta}^s.st()$.

$\mathcal{N}$ A Gaussian distribution.

N_{mn} The mean value nominal profile $N_{mn} = A'[P'_V].mn()$. Used to generate type iii analysis.

N_{mx} The maximum value nominal profile $N_{mx} = A'[P'_V].mx()$. Used to generate type i/ii analysis.

$p(\cdot)$: given X , $p(X)$ is the probability density function of X .

$P_V \in \{0, 1\}^n$: boolean vector defining a data classifying partition of X . $P_V = M_V^A < M_\alpha(M_V^A[M_V^A < M_V^A.mn() + M_V^A.st()])$.

$P'_V \in \{0, 1\}^n$: boolean vector defining a data classifying partition of X . $P'_V = M_V^A < M_\alpha(M_V^A[M_V^A < M_V^A.mn()]) \wedge P_D$ where boolean vector P_D is either supplied by DiffPaC, or set to all *True*.

$P_D \in \{0, 1\}^n$: boolean vectors defining a data classifying partition of X as determined by the DiffPaC process.

$Q \in \{0,1\}^n$: manually constructed boolean vector defining pseudo ground truth data classifying partition of X .

$\mathbb{R}$: the set of real numbers.

$rightShift(A, X)$, $A \in \mathbb{R}^{n \times q}, X \in \mathbb{R}^n$: shifts the elements of each row in A right by the value at the corresponding row in X .

$S \in \mathbb{R}^n$: type iii data deviation information for A .

V : a variational autoencoder.

$X \in \mathbb{R}^n$: the n long array of peak amplitude response times of flight indexes generated as $X = \{a.am() | a \in A\}$.

$\hat{X} \in \mathbb{R}^n$: the n long array of estimates $\approx M_\alpha(X[Q])$.

$X_D \in \mathbb{R}^{n \times m}$: an $m \in \mathbb{Z}^+$ wide array of estimates of the learned approximate distribution of $X[P]$ sampled from D .

$X_D^m \in \mathbb{R}^n$: the row wise mean $X_D^m = X_D.mn(1)$.

$X_D^s \in \mathbb{R}^n$: the row wise standard deviation $X_D^s = X_D.st(1)$.

$X_n \in \mathbb{R}^n$: the n long array $X_n = M_\alpha(X[Q])$ of pseudo ground truth for X .

$X_{gt} \in \mathbb{R}^n$: the n long array X_{gt} of ground truth for X . Available for the synthetic dataset.

$\mathbb{Z}^+$: the set of positive integers.

$.am(dim)$:postfix dot operator $Z.am()$ returns the index of the maximum argument of array operand Z in the dimension specified by dim . If unspecified dim defaults to the first dimension 0.

$.mn(dim)$:postfix dot operator $Z.mn()$ returns the mean of array operand Z in the dimension specified by dim . If unspecified dim defaults to the first dimension 0.

$.mx(dim)$:postfix dot operator $Z.mx()$ returns the maximum value of array operand Z in the dimension specified by dim . If unspecified dim defaults to the first dimension 0.

$.st(dim)$:postfix dot operator $Z.st()$ returns the standard deviation of array operand Z in the dimension specified by dim . If unspecified dim defaults to the first dimension 0.

- $.sm(dim)$:postfix dot operator $Z.sm()$ returns the sum of array operand Z in the dimension specified by dim . If unspecified dim defaults to the first dimension 0.
- $|\cdot|$:Vertical bars acting on operand as $|Z|$ may take either the absolute value of Z or its cardinality. Intended operation should be made clear by the context.
- $\sim$:Tilde indicates distributed as. $a \sim b$ indicates a is distributed as b , where a is some random variable and b is a distribution.
- $\approx$:Double tilde indicates distributed approximately as or approximately equal to, as dictated by context. $a \approx b$ indicates a is distributed approximately as b , where a is some random variable and b is a distribution. Alternately $a \approx b$ indicate a is approximately equal to b .
- $\{a, b\}$: where a and b are any objects or elements, curly braces used for grouping or denoting a set containing a, b .
- $[a, b]$: the closed interval such that $[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$.

Chapter 1

Introduction

At the heart of the [Canada Deuterium Uranium \(CANDU\)](#) nuclear reactor is the zirconium alloy [Pressure Tube \(PT\)](#). The [PT](#) positions and supports the uranium fuel bundles and coolant flows necessary for sustaining fission and effective heat removal. In a typical reactor, there are 480 [PTs](#)—each are just under 6 *m* long, approximately 11 *cm* in diameter, and about 4 *mm* thick. During reactor operation, [PTs](#) endure high pressure, elevated temperatures, and very high neutron flux. Over time, these conditions lead to geometric deformation and cause [PT](#) pipe walls to thin and weaken. Furthermore, chemical processes may promote embrittlement—especially around developing cracks, scratches, and fissures—thereby increasing the risk of catastrophic failure [\[122\]](#). It is incumbent upon [CANDU](#) owner-operators to demonstrate to the regulator that these critical components remain fit for service. Periodic [Ultrasound \(UT\) Non Destructive Evaluation \(NDE\)](#) Inspection supplies the data and analysis necessary to make this determination. And so, during planned maintenance outages, a subset of the [PT](#) is emptied of fuel and inspected with [UT NDE](#) tooling.

Inspections locate and characterize material flaws. Flaws are sudden changes in pipe wall geometry, such as those caused by the abrasion and wear of reactor fuel bundle bearing pads and debris interacting with the [PT Inner Diameter \(ID\)](#) surface. They are of interest primarily because regions at the bounding points of their geometry are under increased mechanical stress, which enhances the risk of catastrophic failure [\[122\]](#). To monitor and mitigate the risk of such an outcome, flaw geometry is characterized and a set of criteria used to determine the associated risk to continued safe reactor operation.

[Normal Beam \(NB\) Pulse Echo \(PE\)](#) and [Shear Wave \(SW\) Pitch Catch \(PC\) UT](#) data analysis is widely employed as part of a quality assurance or inspection process and as a

means toward a fitness for service argument in safety-critical contexts. The prevalence of these [UT NDE](#) systems in harsh environments such as nuclear, is because, in many cases, they are easier to design, maintain, and operate than more complex systems such as phased array. The drawback is that data analysis is a time, material, and labor intensive process that costs nuclear operators in Ontario tens of millions of dollars annually to maintain and perform.

[UT NDE](#) data analysis involves inference concerning the geometry and condition of the material under test. Inference is based on a model of [UT](#) wave propagation and assumptions regarding the nominal geometry of the test subject. Together, these support mapping features in the data to an inferred test subject geometry.

The [UT](#) data acquired during inspection of [CANDU PT](#) supports such analysis and allows us to determine whether a [PT ID](#) surface is oxidized, covered by sediment, scratched, cracked, dented, thinned, or deformed. The analysis delivers real value estimates for identified surface flaw geometry (length, width, and depth). The geometric characterization is used to quantify exposure to the risk of postulated catastrophic failure modes.

Over the last decade, [UT](#) based [NDE](#) tooling and data acquisition systems servicing many [CANDU](#) reactors have been redesigned. In the past, data acquisition, storage, and retrieval rates proved to be significant design constraints. However, modern tool designs take full advantage of widely available commodity hardware to acquire and manage massive datasets quickly. Currently, at Bruce Power, a [CANDU](#) owner operator in Ontario Canada, a single [PT](#) inspection yields approximately $3TB$ of data.

While data acquisition has vastly improved, the process and software supporting data analysis remain unchanged. As they age, they become increasingly inefficient to execute and maintain. The process requires a team of highly skilled analysts, each receiving specialized training, to independently assess and characterize acquired data with few tools other than legacy data visualization and spreadsheets. Analysis occurs in parallel with other high-priority work, making an already complex scheduling problem more difficult. Analysis often relies on noisy data, which increases subjectivity and challenges repeatability.

The procedure-based methods, which inform manual analysis do not fully engage the dataset because they are sensitive to noise, which can induce a high false positive rate in the data requiring analysis [\[45\]\[68\]\[79\]](#). While the repetitive nature of the work can lead to human performance error [\[106\]](#). The software underlying the process relies on an obsolete code base, running on decrepit hardware, which is challenging to maintain and procure and fails to take advantage of the many improvements often taken for granted in modern information technology infrastructure. The status quo cannot be maintained

indefinitely. The recent success of machine learning methods has led to the consideration of their application in the context of [UT](#) data analysis to enhance overall efficiency and efficacy of the process.

1.1 Motivation and Problem Statement

In recent years, stakeholders have agreed to refurbish and extend the operating lives of nuclear assets in Ontario. Nuclear assets, once scheduled for decommissioning this decade, are now slated to operate well into 2060. Life extension motivates owner-operators to consider revitalizing resource-intensive legacy processes whose efficiency and efficacy may be enhanced through innovation. In particular there is a desire to understand and develop how innovative machine-learning-based methods that provide automation and decision support could improve data bound legacy processes by:

- Reducing human error,
- Improving efficiency,
- Enhancing efficacy.

Where it is assumed that decision support and automation will, by its very nature, reduce the opportunity for error induced by tedious and repetitive cognitive loads, and perform activities at rates that outstrip the capacity of humans alone [\[106\]](#)[\[2\]](#) [\[100\]](#). Thus, interest in improvement is sponsored in parts by recent advances in the theory and application of machine learning, the improved capacity of recently designed data acquisition systems, the promise of automation, and the proposed life extension of nuclear assets.

The problem at hand is to develop an unsupervised machine-learning-based approach for decision support and automation of [UT NDE](#) data analysis. The intention is to learn from and improve upon prior work in discriminative learning [\[49\]](#) and heuristic methods [\[79\]](#)[\[80\]](#)[\[141\]](#)[\[153\]](#) done with [CANDU PT NDE](#) data. In our opinion, the cited methods:

- Incur technical debt by relying on labeled dataset management and curation [\[3\]](#),
- Lack a principled basis by which to understand and communicate model performance to technical stakeholders,
- Have weak explainability, fault diagnosis, and error remediation for users in the field,

- Rely on out-of-distribution generalization.

These characteristics may render them unsuitable for use in safety-critical contexts [38][47][110][141].

We require a system that avoids such pitfalls in safety-critical contexts. Such a system must meet requirements as provided in Chapter 2. The system described in this thesis does so and is a proof of concept, or minimum viable product that forms the basis on which a fully qualified, verified, and tested production system may be built.

1.2 Contributions

In response to our perspective on the limitations of supervised discriminative and heuristic methods, this thesis proposes an unsupervised, generative, data-driven approach for decision support and automation of UT data analysis in safety-critical contexts [38][110]. The approach is based on the integration of two novel unsupervised anomaly detection algorithms: Variational Autoencoder (VAE) based Nominal Profiling (NPROF) and Diffusion-based Diffusion Partition Consensus (DiffPaC). It meets requirements as given in Chapter 2, achieves accuracies required by the regulator [150], and provides desired analysis at a rate well beyond that of human performance. In so doing, it promotes its applicability in safety-critical industrial inspection processes and establishes a business case for the revitalization of the related legacy process.

The approach is remarkable because its application recovers time-of-flight information from the dataset hidden from legacy methods to produce an improved analysis in short time frames. This is a first-of-its-kind result that dramatically improves on the efficiency and efficacy of legacy amplitude-based methods. The core contribution of this work is an efficient and efficacious safety-critical ready data analysis system that:

1. Includes a robust mechanism for fault diagnosis and error remediation, designed for clear visualization, confirmation, and verification of results,
2. Decouples system performance from reliance on out-of-distribution generalization,
3. Incurs no technical debt associated with dataset labeling, curation, and management.

These contributions advance the feasibility of generative model-based approaches for industrial NDE in general by offering a compelling alternative to supervised discriminative

and heuristic methods and support a business case for process revitalization in the nuclear context. Although not directly comparable to prior work, they establish state-of-the-art performance for decision support and automation in this space [49][79][141][153].

1.3 Arc of research

Research began with an investigation of whether machine learning processes were able to perform data-related tasks on CANDU UT NDE datasets. Initial experiments demonstrated that Generative Adversarial Network(GAN) architectures could learn UT dataset distributions and produce realistic novel synthetic data. That VAE architectures could encode and decode reasonable facsimiles of UT data. That, given a labeled dataset, a Convolutional Neural Network (CNN) could estimate and predict desired target parameters. And that a Conditional Variational Autoencoder (CVAE) could predict missing or unobserved data conditional on locally observed data. This established the feasibility of the research.

Based on the feasibility study, a research proposal describing a cascading series of CVAE [125] and various anomaly detection mechanisms was developed [6][130]. Where the architecture inferred unobserved nominal features in UT data given local observed nominal. Experimentation with conditional structures, system architectures, hyperparameters, and conditional inference mechanisms of all kinds resulted in numerous incarnations of the proposed system. Each showed promise but provided inexact estimates of target parameters on datasets with ground truth. Performance relative to a manually generated pseudo ground truth skirted about but did not solidly come within the accepted theoretical limit for measurement accuracy, determined experimentally to be $\pm 20 \mu m$ [135].

Further research developed possible root causes of inexactness, which were found to be the existence of outliers and noise in the data, and the relatively high dimension of the dataset and data-related task formulation. These increased the propensity of the dataset to exhibit spurious correlation and muddled the accuracy of data-driven learning processes whose loss functions tended to split the cost between spuriously and causally correlated data.

Experimentation with traditional data fitting and modeling methods showed that they, too, appeared to split the cost [116]. Work with iterative approaches that fit and then culled obviously anomalous data using summary statistics showed that culling criteria were difficult to determine and brittle, so these approaches either erred strongly after pruning nominal data or became stuck with no further improvement as anomalous data was consistently retained [86][133].

At this point, the use of a generative diffusion model [55][128] in conjunction with a traditional data fitting technique was conceived. The generative model induced a stochastic element in the process so that traditional fits were made on random subsets of the dataset and sampled estimates. Culling criteria became learned, and inappropriately culled data that agreed with a progressively more accurate learned approximate data distribution could be reconstituted. Research in this direction produced the DiffPaC algorithm described in Chapter 8 [133].

The success of DiffPaC rests in part on the reduction of the high dimensional UT datasets to univariate time series. The data reduction reduces the density of spurious correlates within the dataset, enables rapid learning and allows the accuracy of parameter estimation to be expressed in terms of an easily visualized and remediated binary classification problem.

But DiffPaC is a time-of-flight-based approach to data analysis. As such, it is essentially blind to small flaws easily identified using traditional amplitude based methods [135]. For these, a VAE [72]-based data partitioning approach is used to identify and characterize very likely nominal data. The very likely nominal component of the data is then characterized using NPROF [131] and all observed data compared to it. Substantial differences between the characterization and the dataset indicate regions where anomalies are likely to exist. Amazingly, the conjunction of methods recovers all time of flight information required for the desired analysis.

1.4 Thesis organization

This thesis follows the arc of research quite closely and is laid out as follows:

- **Chapter 2:** Gives a set of design requirements and rationale used to guide design choices made throughout the research,
- **Chapter 3:** Describes the physical plant and the current state of systems and processes that make up CANDU PT UT NDE data analysis,
- **Chapter 4:** Reviews literature that is either directly related to, or has affected the course of research,
- **Chapter 5:** Describes the real world and synthetic datasets used throughout the research,

- **Chapter 6:** Presents a baseline system, using VAE and NPROF, that provides good results but is sensitive to outliers and does not provide strong fault diagnosis or explainability,
- **Chapter 7:** Details the ablation study underpinning design choices for DiffPaC,
- **Chapter 8:** Proposes DiffPaC, an integration of generative and deterministic modeling techniques, that improves on the VAE based system in terms of accuracy, fault diagnosis, and explainability,
- **Chapter 9:** Integrates components to form a scalable multi-probe, cloud based system and considers the challenge of system commissioning,
- **Chapter 10:** Summarizes contributions, discusses limitations and areas for further research, and makes a set of concluding remarks.

Chapter 2

Requirements, design choices, and scope

Research is motivated by the desire to develop a proof of concept, or minimum viable product for decision support and automation of [Canada Deuterium Uranium \(CANDU\) Pressure Tube \(PT\) Ultrasound \(UT\) Non Destructive Evaluation \(NDE\)](#) data analysis ¹ using some set of machine-learning-based processes. The intention is to determine means that are systematic, objective, explainable, accurate, timely, manageable, efficient, efficacious, and usable within a safety-critical context. At a minimum, such a system must meet the following requirements:

1. Must produce results with little or no dataset pre-processing, labeling, management, or curation,
2. Must produce results with a scope and rate that outperforms human analysts,
3. Must have a principled and explainable basis of operation by which the success and failure of performance may be understood,
4. Must explicitly and robustly engage class imbalance and non-stationary datasets and decouple model performance from out-of-distribution generalization,
5. Must provide fail-safe means to fault diagnosis, fault remediation, and results verification,

¹Here-after referred to as simply UT data analysis

6. Must support the provision of dimensional analysis with some acceptable degree of accuracy.

Requirement 1: Establishes a business case for process revitalization by reducing the cost of application life-cycle maintenance with an upfront reduction in technical debt associated with dataset curation, maintenance, and processing.

Requirement 2: Establishes a business case for process revitalization by accelerating and broadening the scope of analysis while simultaneously reducing resource expenditures.

Requirement 3: A principled basis of operation comprises a theoretical foundation for understanding model performance and results. This supports rationalizing and communicating a technical justification for using novel methods to stakeholders such as engineering staff and regulatory bodies. Contrast this with explainability, which provides a simple, clear, concise, and easily communicated means to understand the success and failure of model performance and results. Explainability is a human factors design element that supports the ability to train field staff to effectively use methods whose theoretical basis they may not fully understand. A principled basis and explainability are both required [40][99].

Requirement 5: While the use of modern statistical learning methods to provide decision support in industrial NDE applications shows great promise, the risk and consequence of failures to generalize may prevent their usage in highly regulated safety-critical contexts [20][87]. Many well-accepted machine-learning techniques do not adequately address the risk posed by out-of-distribution error [5][52][53]. We require a process or method that is independent of the ability of pre-trained models to generalize over out-of-distribution data.

Requirement 5: Use in a safety critical environment implies that all available means to reduce system error have been employed. When and where it is not possible to guarantee that system error is within acceptable limits, means for fault diagnosis, error remediation, and results verification must be available.

Requirement 6: System accuracy must meet regulatory requirements [19][150].

2.1 Design choices

Only unsupervised generative models were considered during the course of research. This comes from experience implementing machine-learning models in industrial contexts where dataset management and curation implies an immediate and ongoing technical debt [35][91].

As a class, unsupervised models do not require dataset labels and formulating the data-related task correctly may ameliorate the burden of required data pre-processing (Requirement 1).

[Conditional Score Based Diffusion \(CSDI\)](#) directly models forward and reverse stochastic processes on the data, while the [Variational Autoencoder \(VAE\)](#) models the distribution of latent variables that capture or meaningfully describe complex structure in the data. Both learn using stochastic gradient descent to optimize loss functions based on score matching (CSDI) and variational inference (VAE). Each provides a reasonably principled mathematical basis for understanding model performance, but the technical nature of the basis obfuscates explainability 3.

Explainability is provided by formulating the data-related task as a binary classification problem: nominal and anominal. Estimated nominal trends are based directly on an automated data classification and are accepted or corrected on the basis of an intuitive visual verification performed by a human in the loop (Requirements 3, 5).

The question of robustness to out-of-distribution data is fraught with uncertainty. The quality of data in an industrial data inspection process is subject to the material condition of the tooling and physical plant as well as human error [62][67][68][69]. While quality measures associated with inspection processes helpfully shape potential data distributions, there are no means to quantify the nature of this help. And generative models only surely reflect the data on which they are trained [11]. So, why use a model based on what is unknown but hoped for if it is possible to engineer a process that takes full advantage of what is known? This line of thought leads to the realization of an offline process that trains models separately and independently on each incoming dataset. This relaxes reliance or blind faith in the ability of models to generalize and removes any requirement for dataset curation (Requirement 1, 4).

A process involving a generative model trained on each dataset independently, opens the door to placing a human in the loop to monitor, correct, and verify the system during model training and usage [57]. This design choice expediently supports the concept of a fail-safe watch dog mechanism on both the system and results [16][33][148]. All that is required are efficient training and sampling, and appropriate human factors to support a human in the loop [4][76]. Since, in safety-critical processes, a human is required to verify final outcomes, a machine-learning process that intentionally incorporates and leverages a human in the loop seems practical and meets Requirement 5 by design [133].

But, to include a human in the loop means training a human for the loop. To successfully manage training requirements, the system’s concepts should be intuitive and explainable. For a machine-learning-based process to be accessible, it must support un-

derstanding in some low-dimensional lingua franca. Ideally, results should, in every case, be acceptable or not, and errors should be manifest or absent, for some small bundle of readily apparent reasons not a cognitively intractable n-ball of them. This is achieved by expressing the data-related task accessibly, as an easily understood binary classification problem [151](Requirement 5).

The legacy data analysis process requires that independent analyses performed by multiple analysts agree to within 0.1 *mm*. The estimated error of UT analysis on 20 *MHz* Normal Beam (NB) probe data is thought to be $\pm 20 \mu m$ about the true value. The regulator requires that flaw dimensional analysis carry the following accuracies:

- Depth estimates within $\pm 2.00 \cdot 10^{-5} m$ of available ground truth,
- Axial length estimates within $\pm 5.00 \cdot 10^{-4} m$ of available ground truth,
- Radial width estimates within $\pm 1.00 \cdot 10^{-3} m$ of available ground truth.

Any support to analysis must consider adequately each of these and lie or conform, by some reasonable measure or determination, to these bounds on performance (Requirement 6).

2.2 Complete analysis

PT *ascans* express a series of peaks in signal response in time. Each peak indicates the time-of-flight to a sound energy reflector in the PT. If an *ascan* is nominal, the largest of the peaks represents the sound energy reflected on incidence with the pipe wall Inner Diameter (ID) surface. A second peak, of diminished amplitude, may result from sound reflection off the back wall of the pipe. Several subsequent peaks may also develop as sound energy reverberates within the pipe wall material. Typically, in nominal *ascans*, the amplitude of a series of peaks decreases with time of flight.

Deviating anominal *ascan* response exists on a spectrum determined by the quality and geometry of reflectors and ultrasonic beam characteristics. Peaks may be delayed or advanced relative to a nominal trend in adjacent *ascans*, and their amplitudes may be increased or diminished. Changes in the quality of the sound-reflecting surfaces determine the changing quality of peaks in a series of *ascans*. Our practical experience with CANDU ultrasonic data leads to the definition the following types of deviating, or anominal, signal response in *ascans*:

Type i: Caused by flaws with geometry that admit the majority, or all, of a [UT NB](#) probe beam width. Characterized by an irregular primary reflector with time of flight deviating from the surrounding nominal. Typically, but not necessarily, and especially at the edges of a flaw, where specular reflection comes into play, these express a decreased maximum amplitude relative to surrounding nominal *ascans*.

Type ii: Caused near the edges of flaws, or by flaws whose geometry admits only some fraction of the [UT NB](#) width to enter. Typically, these develop an *ascan* with multiple distinct peaks with times of flight that are different than those observed in adjacent nominal *ascans*. The portion of the beam that does not enter the flaw reflects from the nominal pipe wall [ID](#) surface to develop a primary reflector with a possibly decreased maximum amplitude and a time of flight similar to surrounding nominal *ascans*. The portion of the beam that enters the flaw reflects from its bottom or sidewalls and develops a secondary reflector response with an extended time of flight.

Type iii: High variability in the data may be caused by specular reflection and attenuation due to oxidation, sediment, and deformed geometry. These inject noise into the nominal trend and increase variance in the time of flight and amplitude of reflectors when compared to adjacent nominal *ascans*.

Manual analysis typically uses a combination of amplitude and time-of-flight-based methods to locate and characterize what we refer to as type i/ii/iii data deviations [\[135\]](#). Proposed methods accomplish the same using data-driven machine-learning processes which, at their core, identify variance from learned highly likely nominal features in [CANDU UT NDE](#) datasets.

The distinguishing feature of type ii data deviations is that, in many cases, the measured decrease in amplitude is minute. This presents a problem for legacy analysis, which uses a threshold parameter on *ascan* maximum amplitude as a means to locate flaws and then performs a time-of-flight-based analysis. Because *ascan* maximum amplitude is a relatively noisy signal, setting the parameter too low results in numerous false positives on flaw location, while setting it too high increases false negatives. The methods described in the following chapters largely avoid this dilemma by simultaneously considering amplitude and time-of-flight-based information. This keeps amplitude-based information from acting as a filter that prevents the treatment of relevant time-of-flight-based information. As a result, the proposed methods exploit more of the available and valuable information in the dataset and enhance the overall scope or efficacy of analysis.

The provision of a complete type i/ii/iii analysis of data from a collection of [NB](#) and [Pitch Catch \(PC\) UT](#) probes provides the necessary and sufficient prerequisites for all downstream analysis tasks. Tasks such as determining simple flaw geometry are straightforward and may be susceptible to automation. Determining the geometry of more complex flaws generally requires a more detailed human analysis. In this case type i/ii/iii analysis contains all the detail (that would have otherwise been manually obtained) required to support the more rigorous analysis which provides comprehensive decision support.

The remainder of this document discusses methods and presents results for a 20 *MHz* [NB](#) probe. The 20 *MHz* [NB](#) is the most accurate of the probes typically employed in [CANDU UT NDE](#) inspections, and its data supports the bulk of analysis performed. Typically, probes in an angled [PC](#) configuration are used only to increase the probability of flaw detection, confirm estimates made based on 20 *MHz* data, or to interrogate complex flaw types and geometries. As such, results for the 20 *MHz* [NB](#) probe ultimately determine the utility of proposed methods.

2.3 Summary and scope

This chapter defined the motivation, requirements, design choices, and scope for the development of a proof-of-concept system for decision support and automation in [CANDU PT UT NDE](#) data analysis. The research aims to explore machine-learning-based methods that are systematic, explainable, and robust for use in safety-critical industrial contexts.

Six primary requirements were given to guide the system development and design choices. The proposed design uses unsupervised generative models to meet these requirements and trains independently on each dataset. This avoids reliance on out-of-distribution generalization and supports a human-in-the-loop for fault diagnosis during training.

The data analysis task is framed as a binary classification of nominal and anomalous ultrasonic signal responses. This enhances explainability and sets the stage for implementing the fault diagnosis, error remediation, and results verification mechanisms necessary for safety-critical applications.

Finally, the chapter defines three types of data deviations observed in [UT](#) *ascans*, which form the basis of the system’s analysis. The focus of subsequent chapters is on demonstrating this approach using data from 20 *MHz* [NB](#) probe.

Chapter 3

CANDU pressure tube inspection

This chapter seeks to familiarize the reader with the current practice of [Ultrasound \(UT\)](#) data acquisition and analysis. Topics covered include:

- Relevant components, processes, and conditions of an operating [Canada Deuterium Uranium \(CANDU\)](#) reactor.
- Inspection tooling design and data acquisition.
- [UT](#) probe configurations.
- [UT](#) data and analysis.

The following material is provided with the depth and breadth required to understand the potential impact of proposed improvements.

3.1 CANDU pressure tubes

A typical [CANDU](#) nuclear reactor contains 480 [Fuel Channel Assembly \(FCA\)](#). As in [Figure 3.1](#) the [FCA](#) contain nuclear fuel and span the calandria, a large pressure vessel containing a mass of D_2O . D_2O , otherwise known as deuterium oxide or heavy water, is a molecule composed of two deuterium atoms, each covalently bonded with an oxygen atom. Deuterium is a stable heavy isotope of hydrogen with a nucleus containing a proton and a neutron. The chemical composition of D_2O allows it to interact with and slow, or

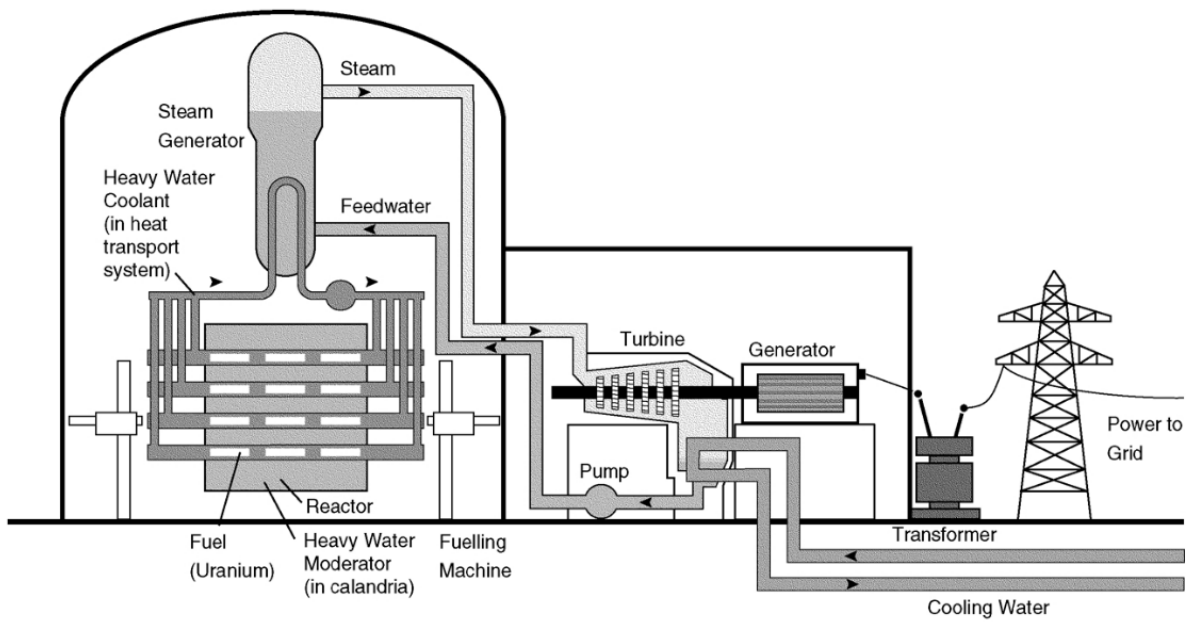


Figure 3.1: Schematic diagram of a simplified CANDU reactor design. Nuclear owner-operators typically run multiple reactors in parallel. Image used with permission [44].

thermalize, free neutrons produced by fission. Thus, the heavy water in the calandria, known as moderator fluid, functions to slow free neutrons produced by nuclear fission to speeds at which a controlled nuclear fission process may be sustained [44].

As in Figure 3.2, each FCA is a conglomerate whose major components include end fittings, feeders, spacers, the Calandria Tube (CT), Pressure Tube (PT), and other associated hardware. The PT forms a pressure boundary that contains heavy water coolant in the heat transport system [44].

The PT is a zirconium alloy tube housed inside the slightly larger diameter CT. Four garter spring spacers encircle the PT to prevent PT/CT contact. PT are approximately 6 m long, almost 11 cm in diameter, and have a pipe wall just over 4 mm thick. During construction the PT are inserted into the end fittings and fixed in place by rolled joints. Rolled joint detail is shown in Figure 3.3 [59].

During operation, the PT locate and support reactor fuel bundles and provide a conduit that allows the Pressurized Heavy Water Transport (PHT) system to push a flow of D_2O heavy water coolant. As in Figure 3.1, the flow moves over the fuel and draws away the heat of fission to adjacent steam generators. There, fresh water in an isolated secondary thermal loop is heated to temperatures and pressures required to drive electricity-generating turbines. This, in turn, produces electricity to the grid [44].

3.2 Deformation and flaws

Though PT are manufactured to exacting quality and true to dimension upon installation, the stress of normal operation from pressure and temperature cycling, static loading, reactor chemistry, and neutron flux cause PT deformation [56].

Over the course of time, physical flaws develop on the PT surface material. These scratches, pits, scrapes, dents, irregularities and impressions are caused by the interaction of the PT Inner Diameter (ID) surface with fuel bundles, fueling machinery, inspection tooling, debris, foreign material and the various processes that cause PT deformation. Various types of flaws exist [134][135]:

- **Circumferential and axial scrapes:** Scrape tools routinely take small samples of PT material for inspection and chemical analysis. Circular and axial scrapes are the depressions and voids left by the tooling performing these inspection operations.
- **Fuel bundle bearing pad frets:** Fuel bundles rest on bearing pads, which are small tabs of zirconium alloy that elevate the bundle slightly from the PT ID surface.

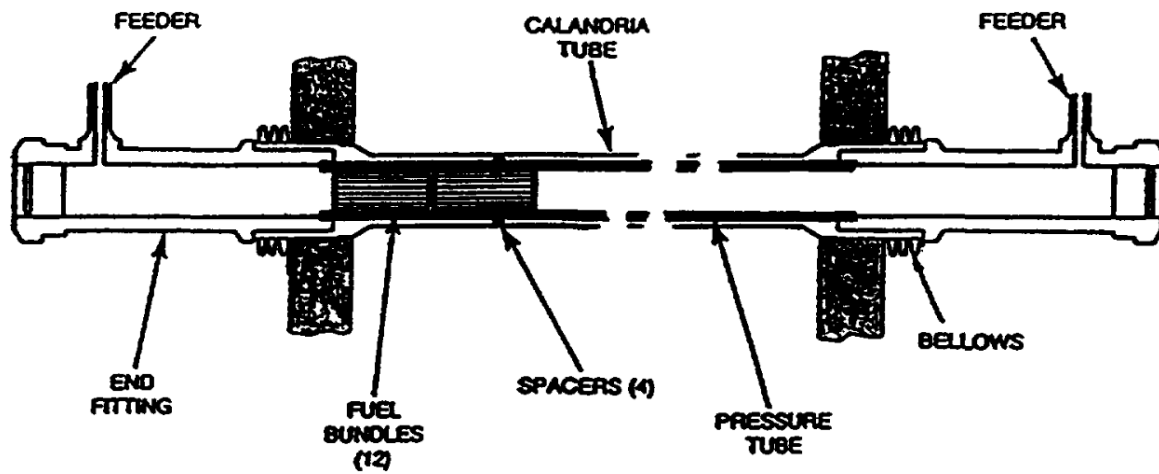


Figure 3.2: Schematic diagram of the FCA. Heat transport D_2O enters the FCA at one feeder, passes over the fuel to draw its heat of nuclear fission, exits at the other feeder, and proceeds towards the steam generator. Image used with permission [59].

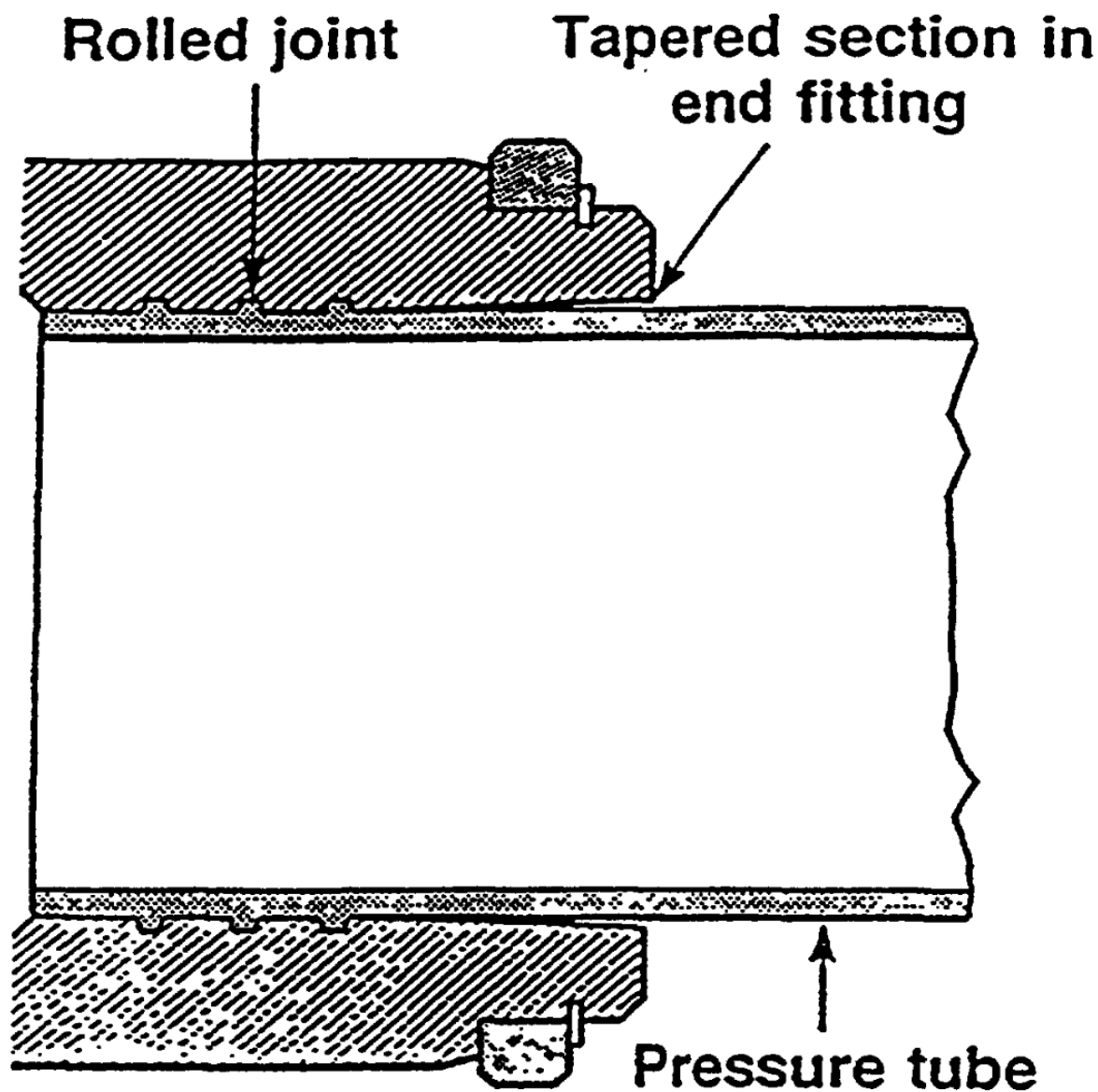


Figure 3.3: The rolled joint between the end fitting and the PT creates a seal to contain reactor coolant within a pressure boundary. Reactor coolant may migrate into the voids in the rolled joint between the pressure tube material and the end-fitting to create a pathway for deuterium ingress and accelerate hydride uptake. Image used with permission [59].

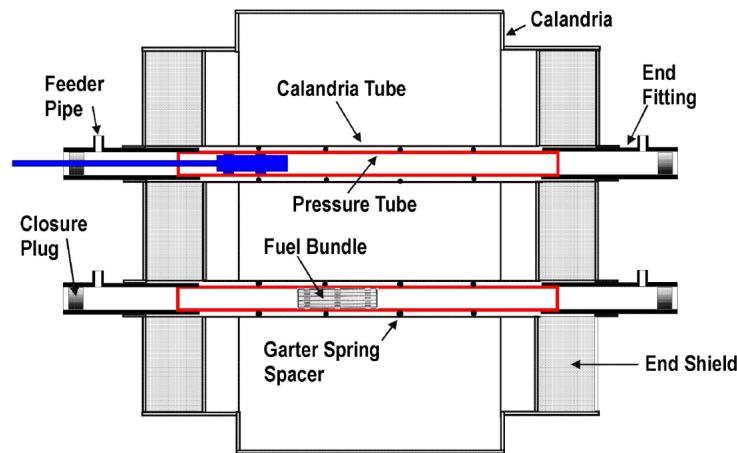


Figure 3.4: Two FCA configurations are depicted. The top FCA contains an inspection tool, in blue and in an empty PT, being progressed axially by a long blue push rod delivery system. Fuel bundles are removed from the PT, prior to inspection, to clear a path for tool traversal. Though not shown, PT deformation may at times be so extreme as to interfere with free axial tool movement; this may induce noise in acquired data. Image used with permission [44].

Constant vibration at operating pressures and temperatures causes fretting at points of contact between the PT and bearing pads.

- **Foreign material and debris:** Foreign material and/or debris in the PHT may interact with the PT ID surface and cause depressions due to wear. Debris may lodge underneath bearing pads and may cause a debris flaw to occur simultaneously with a bearing pad fret flaw.
- **Inner diameter scratches:** ID scratches may occur as debris, fuel bundles, or inspection tooling traverse the PT.
- **Crevice corrosion:** Formed by Lithium Hydroxide concentrations that build up when PHT boiling occurs underneath fuel bundle bearing pads. Crevice corrosion tends to fill flaws and obfuscate their depth when measured by UT inspection tooling.
- **Outer diameter flaws and blistering:** Are anomalies found on Outer Diameter (OD) surfaces of PT. Most OD flaws are assumed to have been formed at the time of PT manufacture or installation. Blisters are a postulated condition believed possible in the event of PT/CT contact.

3.3 Pressure tube oxidation and sediment

During operation, regions of the PT ID surface may be subject to oxidation and the collection of sediment. Both conditions are sensitive to reactor coolant chemistry, which is monitored and controlled to exacting standards [44]. Evidence of these conditions is given by variance in the position of features in the UT data. Nominal ID surfaces display stability in the time of flight of features in adjacent *ascans*. Specular reflection and impedance, induced by oxidation and sediment, variously widens, flattens, accentuates, and skews peaks in UT response. Analysts typically annotate flaw characterizations, in regions where oxidation or sediment are observed, with the caveat that results are based on noisy data and, therefore, carry greater uncertainty. This informal characterization provides the current type iii analysis [134][135].

3.4 Inspections, tooling, and data acquisition

PT Inspection is performed primarily to satisfy a regulatory requirement to monitor the progression of pre-cursors to catastrophic PT failure. Such pre-cursors include, but are not limited to, the formation and growth of hydrides in the PT zirconium lattice structure and the associated phenomena of Delayed Hydride Cracking (DHC). In particular UT inspection is performed to monitor, and assess conditions, at regions of high mechanical stress in the PT, such as those at the extremities of flaws, where hydride uptake and DHC are most likely to occur [122][150].

CANDU PT inspection tooling is typically deployed by delivery systems that access the end fitting to insert and position tools as required. A simple in-channel tooling configuration is depicted in Figure 3.4. Figure 3.5 shows a typical inspection tool. It is about 2 m in length, and just smaller in diameter than a PT. During inspection, the tool is centered in the PT with spring-loaded rolling mechanisms that damp the rotational wobble of the spinning head. As the delivery system moves the tool axially down the PT, the tool head rotates, allowing the embedded UT probes to scan the PT ID surface. Tool designs often include U-joints along their length to allow free movement against PT deformation that may potentially impinge travel. Designs are proprietary and not widely documented in publicly available literature.

Again, based on experience, and similar to that found in [30][31], and [32], Figure 3.8 provides a block diagram for a simple single probe UT Non Destructive Evaluation (NDE) data acquisition system. The control code synchronizes data collection and tool

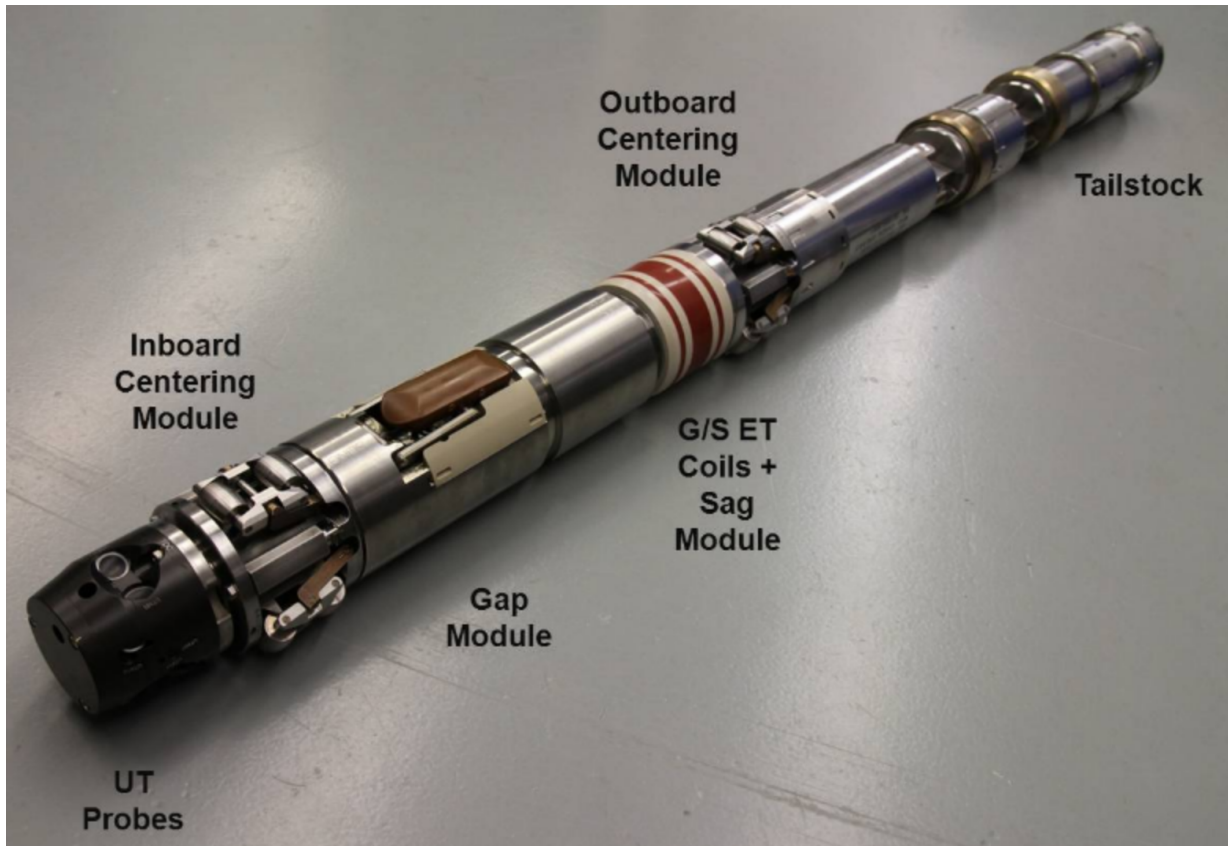


Figure 3.5: A typical UT inspection tool is about 2 *m* in length. This design includes a double u-joint and two centering modules with spring-loaded rollers. The connector module at the tailstock attaches the tool to the delivery system which positions the tool and progresses it axially down the PT. Various sensor packages perform NDE inspection with UT, eddy current, and accelerometers. The UT head rotates as the tool progresses axially down the PT to collect UT data in a helical raster over the ID surface. Image used with permission of Bruce Power LLP.

servo positioning by issuing clock based triggers. Individual **UT** *ascans* are collected when triggers from the control code stimulate the pulse generator to pass a purposefully shaped and modulated voltage signal through an amplifier and onto a piezoelectric crystal **UT** probe. The crystal responds to the voltage across its inputs and emits a pulse of ultrasonic sound energy towards the **PT** pipe wall. A portion of the sound energy is reflected back

toward the crystal and generates a measurable voltage response, which is amplified and routed for analog to digital conversion. The control code then annotates the digitized signal response with the time and location of acquisition and writes it to disk. Real-world inspection tooling designs typically drive multiple probes and signal paths simultaneously. For instance, the UT head on the tool depicted in Figure 3.5 has ten inspection probes on board [134][135].

3.4.1 UT probe configuration

UT data provides indirect information about the geometry of sound-reflecting surfaces in probe acoustic fields. The characteristics of a field and its orientation inform the interpretation of the measured response. Manufacturers characterize probes using parameters such as center frequency, beam width, and focal length to summarize their acoustic properties and aid system designers in choosing probes appropriate to specific applications.

Parameters such as beam width and center frequency determine the resolution and sensitivity of a probe. The shorter wavelength of high-frequency probes allows signal entry into narrow cavities and crevices and resolves edges with fine resolution. Beam width affects sensitivity. Wider beams smooth the variation in response due to reflecting surfaces that are small relative to overall beam width. Focal length determines the distance from the probe at which optimal resolution and signal-to-noise ratios occur. These parameters support the determination of figures for error and resolution that bound the accuracy of analysis based on associated UT data [30][31][77][134][135].

A common probe configuration is Normal Beam (NB) Pulse Echo (PE). NB probes are aligned with an interface normal to the PT ID. A single probe is set up to transmit and, again, receive sound energy pulses. This type of probe configuration provides a direct path time of flight based measurement of the distance from the probe to a reflecting surface and typically provides the most accurate measurements available to analysts [30][31][77][134][135].

Another common probe configuration is Shear Wave (SW) Pitch Catch (PC). SW probes are set up in pairs and aligned so that transmitted UT pulses generate a shear wave in the PT pipe wall. UT sound energy is transmitted by the pitch probe, enters the pipe material and reflects twice off the OD, or back-wall, and once from the ID for what is referred to as a full skip, and upon exit returns to the catch probe. This configuration provides an indirect measurement of material thickness underneath a flaw and also shows evidence of reflectors shadowing or blocking signal transmission from pitch to catch. It is sometimes used to confirm measurements derived from NB data and in cases where the

normal beam is occluded by flaw geometry, such as when the flaw width is too narrow to allow entry of the normal beam.

Most often **PC** signal is used to detect flaws, and **NB** data used for their characterization. This is because **NB** analysis is more accurate than **PC**. This makes sense when one considers that **PC** analysis makes use of a small angle approximation and a more complex signal path, which amplifies associated uncertainty in measurement [30][31][77][134][135]. Figure 3.7 depicts the use of **NB** and **PC** probe configurations for flaw depth estimation.

3.5 UT data

When an ultrasonic probe is excited by a voltage across its piezoelectric crystal, it emits a pulse of **UT** sound energy. Similarly, when the crystal is stimulated by **UT** sound energy, a voltage response is generated across it. *ascans* are voltage response curves in time that measure the intensity of **UT** sound reflected back to and stimulating the crystal in the micro-seconds following an initial voltage pulse across the crystal. Typically, *ascans* are captured sequentially as the probe moves along the surface of a test object [30][31][77][134][135].

Data acquisition occurs using various **NB** and **SW** probes. Throughout this document, experiments are performed, and results are reported for data collected using a 20 *MHz* **NB** probe, but the methods may be applied to any probe whose time-of-flight information is used to characterize flaws¹.

During data acquisition, the inspection tool moves axially along the pipe while a rotating head spins a collection of probes. This allows probes to scan the pipe wall **ID** surface. During each revolution, each probe acquires 3600 *ascans* on a helical pitch of 0.4 mm. For convenience and consistency, throughout this document, we fix axial velocity at 0.4 $\frac{mm}{s}$. In practice, axial velocity varies slightly. Each 20 *MHz* **NB** *ascan* consists of 3250 voltage response measurements taken after the initial voltage pulse to the crystal. Measurements are sampled at a rate of 250 *MHz* using commercial off-the-shelf analog to digital conversion hardware.

Using data collected at the time of acquisition, we can associate each *ascan* with a unique tuple (ϕ, δ, t) , where ϕ represents the rotational angle in degrees, δ the axial position in meters, and t the time in seconds at which the *ascan* was acquired. Sequentially acquired *ascans* may be ordered in terms of angular position, axial position, or time of acquisition.

¹During manual analysis, some probes are used only for their amplitude-based information, which is used to locate flaws. Proposed methods do not use amplitude-based information in this way.

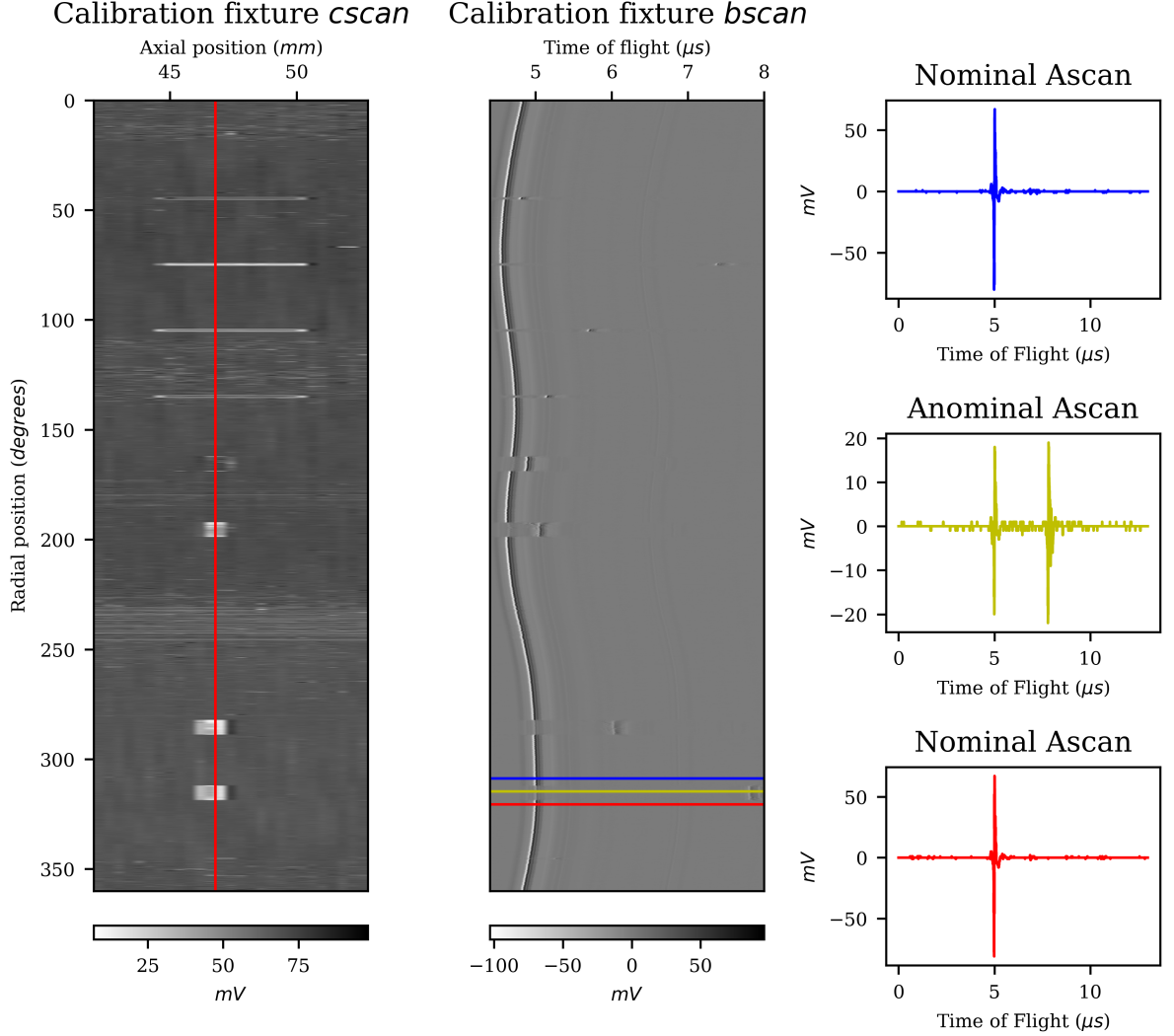


Figure 3.6: *cscan*, *bscan*, and *ascan* are canonical views of ultrasonic data used during manual data analysis. The *ascans* in the right column are taken at the color-coded locations on the *bscan* in the center column. The *bscan* is a two-dimensional plot of a series of *ascans*, stacked and viewed as an image. This *bscan* depicts the *ascans* associated with the red line on the *cscan* in the left column. The *cscan* shows the value of the peak amplitude for each *ascan* taken over its surface, as given by the plot axes.

The time of flight and amplitude of *ascan* peaks and troughs are correlated with the distance that **UT** sound energy travels through heavy water and zirconium alloy. The sound wave makes a round trip, starting and ending at the probe. In regions where there are no flaws or material impediments in the pipe wall, the nominal **ID** is determined by the time of flight of the *ascan* peak amplitude. In this case, *ascans* are classified as nominal. Nominal *ascans* are illustrated in the top and bottom plots of the third column in Figure 3.6. Alternately, in regions with flaws or impediments, *ascan* features are modulated by the anomalies. These *ascans* are classified as anominal, and an example is shown in the middle plot of the third column of the referenced figure [30][31][77][134][135].

A *bscan* is generated by plotting a series of *ascans* sequentially to form an image that provides a relief or profile of UT response across a surface. A *cscan* depicts the peak amplitude of each *ascan* over a surface area, producing an image that highlights variations in peak reflected **UT** sound intensity. Examples of *ascan*, *bscan*, and *cscan* are given in Figure 3.6.

The *bscan* in Figure 3.6 shows a predominantly nominal inner diameter characterized by a regular oscillating pattern, which is interrupted by anominal reflections from notches of known depth in the **Calibration Fixture (CF)**. The notches are used to calibrate the inspection tool prior to **PT** inspection. The oscillation of the inner diameter profile is a result of positioning noise caused by the rotational wobble of the spinning head. This wobble complicates data analysis due to its non-stationary distribution, evolving in a manner that can be informally described as chaotic [49][134][135].

Other sources of noise in the dataset may include positional error due to deformation in pipe geometry, jitter in the axial and rotary servo mechanisms, stray electronic signal, and transient ambient and tool condition [134][135].

A strong assumption made about inspection datasets is that they are overwhelmingly nominal. An assumption practically guaranteed by stringent quality control processes governing **CANDU** operation and inspection. These ensure that the vast majority of the **PT** surfaces are fit for service and that ambient and tool conditions are within some controlled range [56].

The difference between nominal and anominal *ascans* is sometimes difficult to determine. *ascan* features may be modulated by oxidation or sediment on the inner diameter surface, causing specular reflection. Specular reflection randomly alters peak and trough widths and amplitudes, thereby increasing noise in the **ID** surfave signal. *ascan* taken at the edges of flaws, or over small-diameter flaws, may contain a response from both nominal and anomalous regions interacting with the **UT** beam. Furthermore, variations in tool operation and ambient conditions can introduce anominal components in otherwise nominal

data. This ambiguity complicates meaningful data classification [30][31][77][134][135].

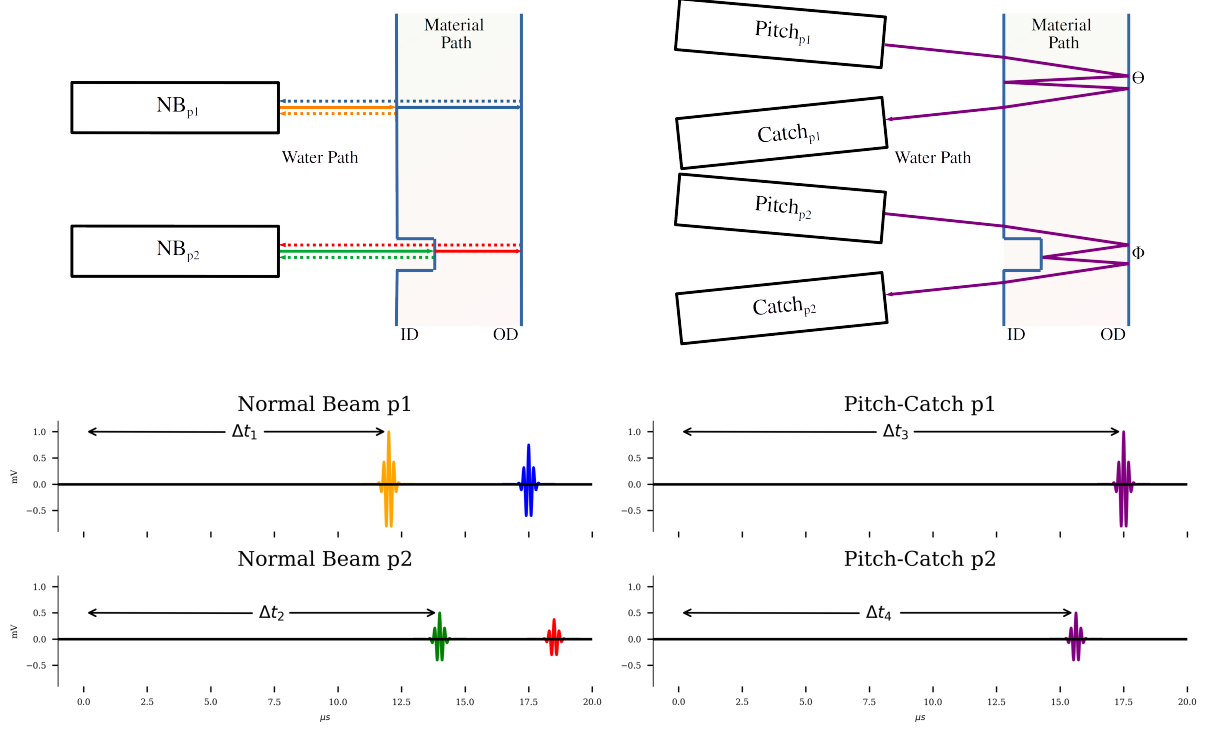


Figure 3.7: Both the *NB* and *PC* probes flag the region at *p2* for analysis on the basis of a relative decrease in *UT* response amplitude. With ID and OD the PT inner and outer diameter respectively, and with the speed of sound in the water path and test subject material v_{wp} and v_m respectively, let d_t and d_f be nominal test subject material thickness and flaw depth respectively. Let $d_t \gg d_f$ so that $\theta \approx \phi$ and let θ be a known design parameter. Then with the normal beam $d_f = v_{wp}(\Delta t_2 - \Delta t_1)/2$. And with the pitch catch pair $d_f \approx v_m(\Delta t_3 - \Delta t_4) \cos(\theta/2)/2$.

3.6 UT data analysis

The primary goal of *NDE* inspection is to locate and characterize flaws. Flaws are sudden changes in pipe wall thickness caused by abrasion, wear, and fatigue across the *PT* ID. Flaws are of interest primarily because regions at the bounding points of their geometry (flaw tips) may be susceptible to hydride uptake and *DHC* [122][114].

In addition to characterizing flaws, analysts are required to identify and locate key PT features. These features include the end of the PT, the three rolled joint rings, and the burnish mark location at the tapered section (as in Figure 3.3). The rolled joint process performed during installation of the PT is imperfect and may leave voids in the rolls between the PT and end fitting. Over time water may collect in these voids and increase the probability of hydride uptake. The presence of water may also be inferred from the data taken at the rolled joints [134][135].

To find flaws, UT data is processed using amplitude-based flagging criteria. The criteria generate a list of regions within the dataset for which analysis is required. Though time-of-flight-based methods are also sometimes used for flaw detection, amplitude-based methods are the de-facto standard in UT NDE inspection processes.² Data flags for analysis when the variance of maximum amplitude between *ascans* in a local neighborhood exceeds some predetermined threshold or criteria. Amplitude variance occurs due to the strong specular reflection of UT sound energy in and around flaws [30][31][77][134][135].

Once a list of regions with strong amplitude variance has been constructed, analysts compile the data relevant to each location and perform in-depth analysis. Flaw geometry is characterized by providing exact flaw axial and radial location, length, width, and depth. Both amplitude and time-of-flight-based techniques may be used to establish flaw geometry. Figure 3.7 demonstrates the use of time of flight to estimate flaw depth for the NB and PC probe configurations [134][135].

The industry regulator has approved an analysis process wherein UT data is analyzed by two independent analysts and then again by a third analyst who reviews and compares the independent work, checking for consistency, and resolving discrepancy. This three stream analysis is costly in terms of person hours but, given adequate training of analysts, provides reason for confidence in results [134][135]. Machine learning-based decision support and automation improves the current state of analysis only if it achieves a similar, or improved, level of confidence with similar, or fewer resources .

²Amplitude based methods are the traditional approach to managing the complexity of processing large volumes of UT data. Amplitude-based methods carry with them a penalty in terms of the probability of flaw detection. Though this is subject to further study, results suggest proposed likelihood and time of flight-based methods make improvements on this.

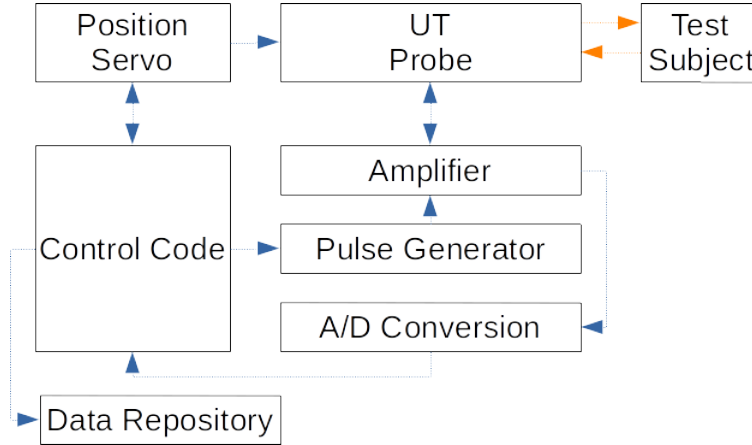


Figure 3.8: A simplified UT-based NDE system schematic. In modern inspection tools, most of the components are implemented using commercial, off-the-shelf components running application-specific control code. In the case of PT inspection, the pulse generator, amplifier, and probe configuration meet specific requirements for CANDU PT inspection.

3.7 Selected bibliography for CANDU component and inspection processes

Various sources discuss the chemical and material properties of [CANDU PT](#). [\[25\]](#) and [\[60\]](#) provide detailed descriptions of [CANDU PT](#) physical attributes, composition, and manufacturing. [\[27\]](#) and [\[109\]](#) provide an analysis of the performance of [CANDU PT](#) with guidance on criteria for inspection and failure modes. [\[114\]](#)[\[121\]](#)[\[122\]](#)[\[123\]](#) and [\[136\]](#) provide a detailed analysis of hydride uptake and embrittlement in [PT](#), underscoring a requirement to monitor and assess [PT](#) state in terms of deuterium ingress or hydride uptake. [\[28\]](#)[\[90\]](#)[\[140\]](#), and [\[147\]](#) provide analysis of [PT](#) catastrophic failure under terminal stresses and end-stage failure modes. While [\[56\]](#) discusses various modes of [PT](#) deformation. [PT](#) inspection and data analysis, are described by Trelinski in [\[134\]](#)[\[135\]](#). Karpelson has produced an extended discussion on various aspects and innovations to [UT](#) inspection and analysis, some of which may be found in [\[63\]](#)[\[64\]](#)[\[65\]](#)[\[66\]](#)[\[67\]](#)[\[68\]](#). Current inspection programs are designed to meet the requirements of an evolving set of standards that are based on empirical observation and scientific study and are endorsed by the national regulator [\[25\]](#)[\[26\]](#)[\[27\]](#)[\[103\]](#)[\[109\]](#)[\[150\]](#).

3.8 Chapter summary

This chapter describes current practice in **UT** data acquisition and analysis. Reactor components relevant to inspection, such as the **FCAs**, **PTs**, **CTs**, and rolled joints are described. **PTs** and their susceptibility to deformation and even catastrophic failure by **DHC** are described in detail. This established the basis of the requirement to perform **PT** inspection.

Various types of **PT** flaws are described, ranging from circumferential scrapes and bearing pad frets to corrosion, debris damage, and blistering. Oxidation and sediment buildup, which complicate flaw interpretation due to signal noise were also discussed.

Inspection tools and data acquisition are discussed in depth, with emphasis on **NB** and **PC** probe configurations. These acquire **UT** data as they traverse the pipe, and generate *ascans*, *bscans*, and *cscans* that analysts interpret manually. The mechanics of data acquisition, probe characteristics (such as frequency and beam geometry), and how signal reflections encode flaw information are clearly outlined.

Current analysis procedures relying on amplitude-based methods to flag anomalies and time of flight information to characterize them are described. As is the flaw characterization process, which is resource intensive but reliable and produces high confidence results.

This chapter highlights both the sophistication and limitations of current practice. The understanding it brings provides the context for evaluating whether proposed machine-learning-based improvements achieve performance that meets or exceeds the current standard.

Chapter 4

Literature review

4.1 CANDU reactor innovation

Most in-service nuclear reactors were built before the wide-spread digitization of the last several decades. As a result, nuclear systems often lack the connectivity associated with today’s internet of things. The application of often network-connected, digital, modern methods to support reactor component health and safety monitoring is considered in [29] which provides a comprehensive survey of prognostics and health management in nuclear reactors. Of late, attention has turned to the automation of inspection data analysis [49][131][132][133][141].

The application of machine-learning techniques in the context of tasks related to **Ultra-sound (UT)** inspection data has been widely reported on. All manner of model architecture, feature extraction, data pre-processing, and learning approach have been investigated. [20] surveys much of this work and makes predictions and recommendations for directions in future research based on a projected arc of progress in methods. Another survey [22] of methods in structural health monitoring cites several supervised deep learning methods and applications for **UT**.

K-Means clustering with Support Vector Machine (SVM) classification is used in [45] for artifact detection in preprocessed ultrasonic data to automate flaw detection. In [104], a discriminative system is developed that estimates the length and orientation of cracks measured in **UT** data. Results show that in some cases, the system performs with greater accuracy than manual analysis. However, researchers caution that training datasets must exhibit “all significant variation that is expected to occur in the real inspection.” This is a

fundamental fact of machine-learning, and the difficulty is that, except for trivial circumstances, there are no means to prove such coverage has been attained [7][8][93][126]. [105] proposes Gaussian Feature Approximation(GFA), a novel technique for data-dimension reduction intended to improve the interpretability and explainability of machine-learning results in UT data analysis. Several architectures and data processing techniques for feature extraction for use with architectures proposed in [127] are investigated in [85] and [102].

Numerous publications document research performed on the same Pressure Tube (PT) UT datasets treated in this thesis. [80] proposes an end-to-end expert system for automation and decision support of manual PT UT data analysis. The work integrates signal processing and ad-hoc heuristics in an expert system. The expert system codifies a manually compiled database of subject matter knowledge, which provides good explainability for results [118]. Benchmarking against the results of manual analysis on data from six PTs is performed. There is some weakness with respect to the accuracy of results which achieves a mean of 86% relative to manual analysis. The authors remark that data-driven processes might be used to improve results which appeared to suffer due to wide variability in data. [152] proposes unsupervised clustering using Density Based Spatial Clustering of Applications with Noise (DBSCAN) [42] to support *cscan* analysis. The method clusters on the basis of learned features and location, providing an approach to identify and locate PT flaws. A proof of concept supervised Convolutional Neural Network (CNN)-based architecture able to identify and locate flaws is proposed in [49]. Labels are provided by manual data analysis, and training data is down-sampled and concatenated to manage compute requirements. Results are adversely impacted by noise in the dataset.

With the exception of [152], which is based on the unsupervised but parameter sensitive DBSCAN [17][42] method, all Canada Deuterium Uranium (CANDU) specific research is dependent on labelled datasets. In the case of [79][80], and [141] the labels provided by manual dataset analysis are used to generate the set of heuristics on which the proposed system operates. In the case of [49] labeled data is required to train the neural networks used to identify and characterize flaws. In each case, consistent performance relies on out of distribution generalization, which dictates that the labeled datasets be large. But because field conditions may vary widely and unpredictably, no guarantee can be made that a dataset contains the information required for the desired level of generalization [7][8][93][126].

4.2 Generative methods and models

Given sets of data and labels X and Y with elements x and y respectively, generative models, such as [Variational Autoencoder \(VAE\)](#) [72] and [Diffusion Probabilistic Model \(DPM\)](#) [55], seek to learn underlying data distributions $p(x, y)$ and $p(x)$ (respectively), while discriminative models learn the conditional distribution $p(x|y)$. When tasked with classification, i.e., finding label y given datum x , a generative model may use Bayes' rule to calculate $p(y|x)$ from the learned joint distribution and then use the posterior to stochastically sample y . In contrast, the discriminative model arrives at the posterior $p(y|x)$ directly, via a learned deterministic mapping, without reference to the joint probability or use of a stochastic process [95].

Introduced in [72] the [VAE](#) provides a type of [Autoencoder \(AE\)](#) [111][112] that trains over a variational lower bound. By way of an involved derivation [73] develops an objective that moves target distributions from exact and intractable to approximate and feasibly calculated. Thus, given data X , [VAE](#) learn a latent distribution $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$, and posterior $\mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z})$ by optimizing objective $\mathcal{L}_{VAE}(\mathbf{X}; \theta, \phi)$ with stochastic gradient descent over parameters θ and ϕ . [96] provides a derivation of the [VAE](#) objective and [73] provides a tutorial on practical usage.

The [Beta Variational Autoencoder \(BVAE\)](#) [54] implements methods for learning disentangled latent representations. [BVAE](#) insert a β hyperparameter in the [VAE](#) loss function that modulates the contribution of error between data reconstruction and information bottleneck channel capacity [129]¹. The tunable trade-off between reconstruction error and channel capacity can be used to encourage models to learn disentangled latent representations. For even greater control modulating contribution to the loss function [13] adds a hyperparameter C to arrive at objective:

$$\mathcal{L}(\mathbf{x}, \mathbf{z}; \theta, \phi, \gamma, \kappa) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta |D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) - C| \quad (4.1)$$

The [Conditional Variational Autoencoder \(CVAE\)](#) [125], is a deep conditional generative model for structured output prediction with Gaussian latents that extends the standard [VAE](#) architecture by incorporating a conditional mechanism to perform tasks such as occluded image reconstruction and segmentation. The [CVAE](#) architecture consists of a *recognition* encoder and a *prior* encoder. The outputs from the encoders are concatenated and passed into a generator, which features skip connections to the *prior* encoder. During

¹The information bottleneck is the restriction between the encoder and decoder in a VAE neural network architecture. It may also be considered as an encoding, representation, or embedding.

training, the *recognition* encoder takes both occluded images and their corresponding intact images as input, concatenating their outputs to produce an encoding that is fed into the generator. The [CVAE](#) learns to sample from the conditional distribution $p(\mathbf{y}|\mathbf{x})$, where $\mathbf{x}$ represents occluded images and $\mathbf{y}$ the desired intact images by training over the three separate parameterized networks.

The Variational Autoencoder with Arbitrary Conditioning (VAEAC) [\[61\]](#) extends the [CVAE](#). Given a dataset $\mathbf{X}$ composed of vectors $\mathbf{x} \in \mathbb{R}^d$, VAEAC conditions on an arbitrary subset of observed features to enable sampling of the remaining, masked or possibly unobserved, features. Specifically, VAEAC samples from $p(\mathbf{x}_T|\mathbf{x}_{U \setminus T})$, where U represents the set of all features, and $U \setminus T$ is an arbitrary, potentially unobserved, subset of features in U defined by T . Learning over all possible T is computationally intractable without strong assumptions on $\mathbf{x}$, so the formulation is refined to model arbitrary distributions of T . These distributions can be fixed to apply conditioning specific to a particular application.

Generative methods have demonstrated remarkable results in audio signal processing. Self Supervised Sequential VAE (S3VAE) [\[155\]](#) uses regularization at the VAE information bottleneck to enforce learning disentangled representations of sequential audio and video data. WaveGAN [\[37\]](#) is a GAN [\[46\]](#) for unsupervised raw audio waveform synthesis. WaveGAN implements a flat convolutional GAN architecture with a phase shuffle mechanism to mitigate artifacts in generated audio. WaveGAN is capable of synthesizing short slices of realistic, coherent audio similar to the small, domain-specific audio datasets on which it is trained. DiffWave [\[75\]](#) is a [Denoising Diffusion Probabilistic Model \(DDPM\)](#) based approach to audio synthesis that avoids the instability of GAN-based adversarial training, is more versatile than a [VAE](#)-based system and trains and synthesizes audio more quickly than autoregressive approaches such as WaveNet [\[138\]](#). [Conditional Score Based Diffusion \(CSDI\)](#) is an extension of DiffWave and is a major component of [Diffusion Partition Consensus \(DiffPaC\)](#). CSDI is described in detail in Chapter 7 [\[128\]](#).

4.3 Disentanglement and manifolds

Alongside a stream of practical developments in architectures and objective functions, a principled discussion concerning the underlying nature of distributions learned by generative models has evolved. Borrowing from disciplines such as abstract algebra, topology, homology, statistics, category, and information theory, much of the conversation is focused on the concept of disentanglement. The discussion finds its basis in the assumption of the manifold hypothesis [\[11\]](#) and develops a principled basis for understanding, interpreting, and explaining the usefulness of learned representations.

Under the manifold hypothesis, the distribution of a dataset $\mathbf{X}$ is a composition of effects arising from a set of independent data-generating factors. These factors express in the data a set of independent factors of variation over dataset features. The distributions of learned latent representations are informed by these independent data-generating factors. A learned representation may be considered disentangled when independent data-generating factors of variation map onto isolated, possibly compact, regions of the learned representation space.

Various attempts have been made to define disentanglement but none has gained wide acceptance. In the opinion of the author a useful definition acknowledges the existence of independent data generating factors, and given a specific task to be performed on a dataset, groups the factors as providing information relevant to the task or as providing irrelevant nuisance information [1]. Achilles and Soatto speak of optimal representations and qualify disentanglement as follows:

To simplify the inference process, instead of working directly with the observed high dimensional data x , we want to use a representation z that captures and exposes only the information relevant for the task y . Ideally, such a representation should be (a) sufficient for the task y , i.e. $I(y; z) = I(y; x)$, so that information about y is not lost; among all sufficient representations, it should be (b) minimal, i.e. $I(z; x)$ is minimized, so that it retains as little about x as possible, simplifying the role of the classifier; finally, it should be (c) invariant to the effect of nuisances $I(z; n) = 0$, so that the final classifier will not overfit to spurious correlations present in the training dataset between nuisances n and labels y . Such a representation, if it exists, would not be unique since any bijective mapping preserves all these properties. We can use this to our advantage and further aim to make the representation (d) maximally disentangled, i.e., choose the one(s) for which $TC(z)$ is minimal. This simplifies the classifier rule, since no information will be present in the higher-order correlations between the components of z .

Where $I(y; x)$ is the mutual information between y and x , and $TC(z)$ is the total correlation of z [84].

Network architecture selection, dataset curation, and regularization as provided by objective functions, hyper-parameters, and training methodology are known to induce or enforce the learning of disentangled representations as described by Achilles and Soatto. [11] contains an early extended discussion with respect to these tools and [108] considers explicitly inductive biases to support the learning of representations.

To measure disentanglement, [70] introduces methods for exploring the Riemannian geometry in learned manifolds. The curvature in learned manifolds was found to be low, with straight lines in the latent space mapping to approximately geodesic curves on the manifold. This supports an intuitive understanding of how latent space manipulation can smoothly transform or morph samples across modes. Based in part on the work done in [70], [154] provides a dataset and model-independent disentanglement measures using persistence barcodes to characterize persistent manifold topology.

4.4 Anomaly detection

Well-cited surveys such as [23] and [24] summarize some of the many approaches to anomaly detection that have been proposed over the years. They also document an evolution of interest from classic to machine-learning-based approaches. [124] questions whether deep generative models outperform traditional peers. Well-defined benchmarking is performed across a broad swathe of datasets and models. Results show that One Class Support Vector Machine(OC-SVM) [117] outperformed other methods on tabular data and a stacked VAE [72] with OC-SVM outperformed on image datasets. [101] demonstrates CVAE [125] based anomaly detection. They use reconstruction error to score and threshold incoming signal. The method performs favorably against a small group of alternative approaches. They develop a concise appraisal of developments in semi-supervised anomaly detection and give good insight into the use of the VAE in an industrial anomaly detection context.

An entirely different class of techniques involves the use of non-parametric Bayesian methods [43]. Temporally Reweighted Chinese Restaurant Process Mixture Models [113] extends these techniques, by way of a cohesion function that ignores the identically independently distributed assumption to introduce time dependencies, for multivariate time series with non-stationary dynamics. Memoized Online Variational Bayes (MOVB) [58] implements a Bayesian non-parametric code base with a memoization technique to provide similar functionality with capped resource usage for scalability. Temporally Reweighted Dirichlet Process Mixture Anomaly Detector(TREDMAD) [130] integrates a cohesion function, as in TRCRPM, and memoization, as in MOVB, to provide an anomaly detection algorithm with a fixed amount of computational resources. TREDMAD captures the temporal dynamics of time series data and enables good online anomaly detection on industrial datasets without a requirement for domain expertise.

4.5 Learning the nominal

Similar to DiffPaC [133] are methods for unsupervised anomaly detection that seek to learn only that which is relevant to the nominal component of the data. This allows the contrast between model outputs and inputs, when fed newly observed data, to form the basis for anomaly detection. This approach is seen as a practical and necessary response to working with data that is unlabeled, strongly class imbalanced, and without a set of predictable failure modes [23].

In a straightforward example of this type of anomaly detection, [88] trains a VAE [72] on nominal UT and uses the L_1 data reconstruction error of newly observed data as an anomaly score. In a slightly more complex arrangement, [20] separately trains an AE [112] encoder and decoder to predict and filter nominal data components from UT data. This makes a possibly obscure anominal component of the data identifiable. The unsupervised method requires a curated nominal training dataset and provides real value estimates for desired target parameters.

Defects in UT data acquired by scanning carbon fiber-reinforced plastic composites are detected in [137] using a two-stage anomaly detection training pipeline. The first stage of the pipeline uses DBSCAN [42] to filter anomalous data from a training dataset that an AE is later trained to reconstruct. The trained AE is then used to find anomalies in complete, unfiltered datasets directly.

Even more complex, [62] mitigates the effects of non-stationary ambient and operating conditions on anomaly detection by fusing a VAE [72] with Singular Value Decomposition (SVD) [41] and a Forced Variational Autoencoder(f-VAE). The proposed f-VAE enforces linearity in the learned latent space. Training the composition of components on a set of curated and labelled datasets allows it to learn the effect of changing ambient and operational conditions which facilitates robust anomaly detection.

The weakness of work cited so far in this section is that each learns, or defines, nominal and anominal on the basis of a pre-trained model and a fixed, often curated, training dataset. This pre-training statically binds model performance to the distribution of the training dataset. For performance to remain stable, new data must either be stationary or the model must generalize. However, the distribution of newly arriving data will change with the ambient system and operating conditions. Even when changing conditions are anticipated, as in [62], they cannot be fully accounted for, and generalization cannot be guaranteed [7][8][93][126].

The failure to provide guarantees on generalization challenges the use of these methods in safety-critical contexts. The techniques developed in this thesis are motivated by a desire

to overcome these deficiencies and decouple performance from a requirement for stationary datasets or generalization. DiffPaC [133], in part, accomplishes this by forsaking the use of a pre-trained model and learning each dataset distribution independently.

4.6 Other methods informing DiffPaC

The CSDI [128] model, on which DiffPaC [133] is based, is a current state-of-the-art model in multivariate time series data imputation. But blindly training a CSDI model on each new incoming dataset does not by itself produce results usable within a safety critical context; and without access to ground truth, means for fault diagnosis and remediation must be developed. An ablation study of DDPM and CSDI models [132](DDPM) demonstrates that, while the inclusion of conditional inference and a modified loss function improves CSDI results over DDPM, accuracy still suffers from even very low outlier rates [55][128]. DiffPaC improves the accuracy of vanilla CSDI by accessing means to remove outliers from the training dataset during training. This Fine-tuning or data forgetting is inspired by Large Language Model methods that service concerns for data privacy, the right to be forgotten, and fairness and use similar techniques to bias the information learned by models [78][81][142].

Numerous methods that actively manipulate training signal, or data, while training exist. Unsupervised time series anomaly detection by way of comparison to a learned nominal distribution, with active suppression of anomalous training signal, is implemented in [39] by placing a filter between a GAN decoder and discriminator [46]. The filter is informed by data reconstruction error and effectively removes the impact of diagnosed anomalies from discriminator training.

[149] uses the confidence of model prediction to boost the effect of high confidence data and reduce the impact of low-confidence data during training. Low-confidence data is further isolated and perturbed to establish a boundary or envelope around the nominal distribution.

AE [112] reconstruction error is used in [71] to decompose reconstructed time series into nominal and anomalous components. The learned anomalous component is then subtracted from time-lagged input to an auto-encoder to enhance nominal reconstruction and anomaly detection.

DiffPaC provides a rationale for output that is readily visualized and characterized in terms of consensus between estimates and an underlying binary data classification. This accommodates means for error correction with a human in the loop to act as an oracle

correcting any errors that may occur. This is similar to Active Learning [15] and Data Triage [98]. Both techniques are designed to enable the usefulness of machine-learning techniques by explicitly accounting for edge cases in the data where methods invariably fail. The use of active learning in DiffPaC provides a means towards fail-safe analysis as required in a safety-critical context.

Chapter summary

This chapter provided a review of the literature relevant to the provision of automation and decision support to UT data analysis. It emphasized the use of machine-learning techniques in this context. A wide range of machine-learning approaches has been explored in prior work, including discriminative, heuristic, and generative models. However, much of the research relies heavily on labeled datasets and is dependent on out-of-distribution generalization. Such generalization cannot be guaranteed, and this limitation leads one to consider new approaches that are robust to data variability.

Several generative models, such as the VAE, DDPM, and CSDI were mentioned. These models provide a principled framework for learning data distributions and support tasks like anomaly detection, signal reconstruction, and representation learning.

The notion of disentangled latent representations, grounded in the manifold hypothesis and information theory, was also presented as a key theoretical concept underlying the explainability and efficacy of generative models.

We also reviewed literature on anomaly detection, particularly methods relevant to UT data. While numerous methods have demonstrated potential, many rely on curated training datasets and lack robustness to changing ambient conditions.

Approaches that modified training signal by some treatment of outliers such as, algorithmic active data selection, adaptive training signal filtering, and human-in-the-loop were also discussed. As were model unlearning and forgetting, which present results important to proposed methods.

Taken as a whole, this literature review discusses both the opportunity of possible approaches and methods to attain an efficient, principled, explainable, and fail-safe approach to automation and decision support for UT data analysis. It motivates the methods presented in subsequent chapters and highlights, by exposing gaps and opportunities for synthesis, the specific contributions this thesis makes within the broader field of machine-learning for UT data analysis.

Chapter 5

Datasets

This chapter describes the [Calibration Fixture \(CF\)](#), [Pressure Tube \(PT\)](#) [High Stress Area \(HSA\)](#), and synthetic datasets used during the course of research to evaluate the performance of proposed methods. The datasets exercise proposed methods over a range of task complexity. The [CF](#) and [PT](#) dataset present what might be considered the least and most difficult contexts for analysis in observed real-world data. The synthetic datasets simulate extreme complexity that is not likely to be observed in a real-world dataset. This range in dataset complexity is chosen to demonstrate the capacity of proposed methods.

5.1 Real world datasets

Evaluation of methods was performed based on work done with two real-world datasets derived from scans of a [CF](#) and [PT](#). As shown in Figure 5.1, the calibration fixture features a relatively clean, smooth, and deformation-free [Inner Diameter \(ID\)](#) surface with notches inscribed for calibrating [Ultrasound \(UT\)](#) inspection tool response. [CF](#) geometry matches that of the [PT](#) at time of manufacture. Calibration scans are performed prior to each [PT](#) inspection to ensure the inspection tool is operating within intended design parameters. This quality assurance component helps to bound variation between datasets and supports the quality of both manual and machine-learning-based downstream analysis [50][97][134][135].

The [PT](#) data for which results are reported is obtained from the [HSA](#). This region is characterized by significant geometric regime change, deformation, oxidation, and sediment accumulation. The turbulent coolant flows in and around the [HSA](#) vibrate the reactor fuel

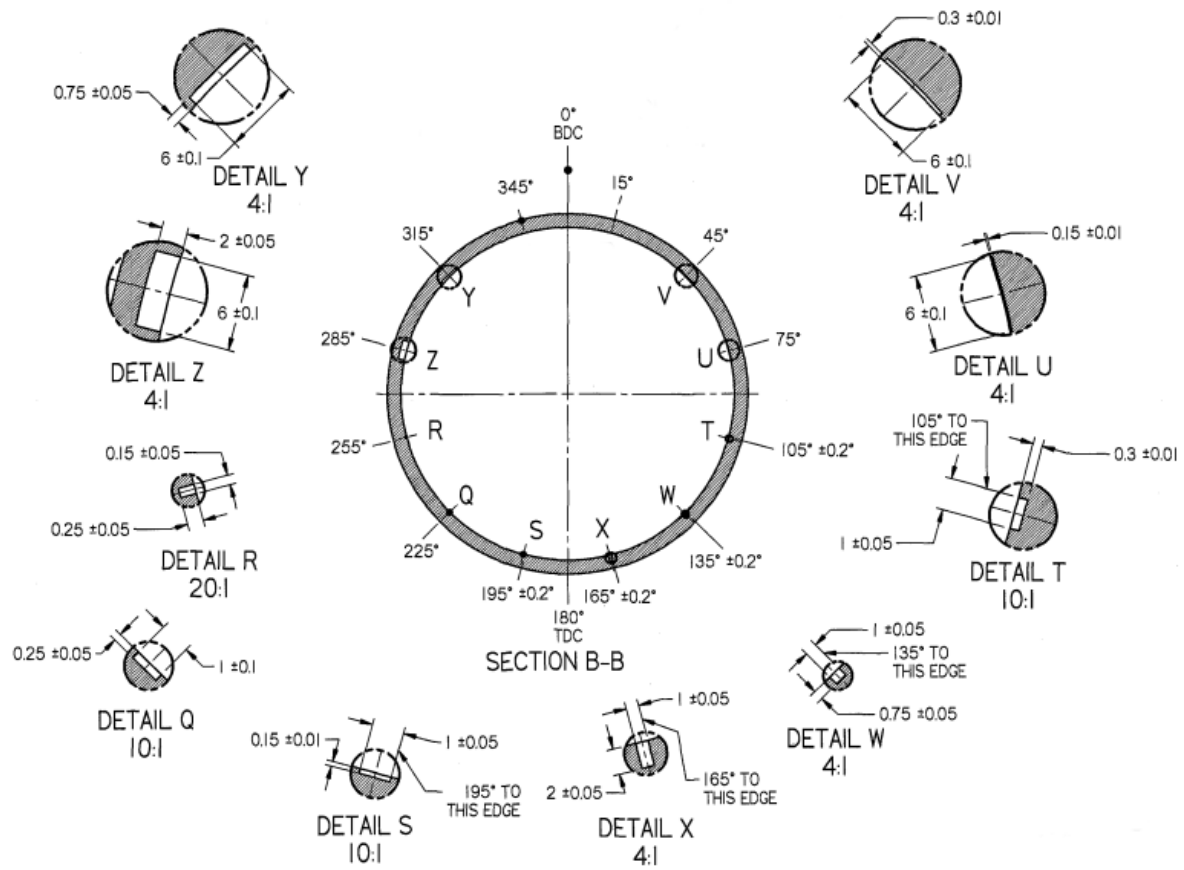


Figure 5.1: The calibration fixture includes a ring of notches of known depth, width, and length. These are used to calibrate the response of UT inspection tools. All distances are in millimeters (*mm*). Figure used with permission Bruce Power LLP.

bundles and lead to considerable fuel bundle bearing pad fretting on the ID surface of the PT. Oxidation and sediment accumulation create a noisy, specular-reflected UT response that obscures nominal trend. These factors result in high-variance data that increases the subjectivity and difficulty of manual analysis [134][135]. Previous work with PT UT datasets in [80] and [141] work with main body of tube data and avoids the difficulty associated with analyzing this region of the PT.

Both real-world datasets were acquired using a 20 MHz Normal Beam (NB) probe sampled at 250 MHz to obtain *ascans* containing 3250 samples each. The full calibration dataset contains over $8 \cdot 10^5$ *ascans*, and the full PT dataset, of which the HSA is a small subset, contains more than $5.8 \cdot 10^7$ *ascans*. Training data subset sizes are indicated throughout this document as required.

A pseudo ground truth partition Q for the CF dataset has been formed by manually classifying the data. X_n is generated as $M_\alpha(X[Q])$ with X the CF dataset.

5.2 Synthetic datasets

The proposed methods do not rely on out-of-distribution generalization. Instead, they are designed to learn the approximate on-hand data distribution [133]. To validate this approach, Diffusion Partition Consensus (DiffPaC) was tested on a series of 25 synthetic univariate datasets. These datasets simulate real-world data by introducing noise, random geometric regime change, and random simulated anomalies of varying size to a sinusoidal base curve. Each synthetic dataset is built on a base curve containing r randomly generated geometric regime changes, where $r \in 0, 2, 4, 8, 16$. Random anominal data (i.e. flaws) are incorporated into the base curve at a rate of approximately $f \in [0\%, 2\%, 4\%, 8\%, 16\%]$. This allows each synthetic dataset to be identified by a unique (r, f) tuple. Figure 5.2 provides a plot of each synthetic dataset, and the code used to generate them is listed in Appendix C.

5.3 Ranking dataset complexity

Dataset standard deviation is used as a naive metric to rank the CF, PT HSA, and synthetic datasets in terms of their complexity [84]. Synthetic datasets have standard deviations ranging from $9.41 \cdot 10^{-5}$ to $8.40 \cdot 10^{-4}$. In comparison, the CF and PT datasets have standard deviations of $1.41 \cdot 10^{-4}$ and $1.92 \cdot 10^{-4}$, respectively. By this measure, the CF

and [PT](#) datasets rank in the very low end of the complexity scale, falling between the synthetic datasets (0, 0) and (0, 2).

5.4 Nomenclature, syntax, symbols, and convention

A full listing of the nomenclature, symbols, and syntax is found in the List of Symbols at the front matter of this document. A few additional conventions and details are noted here.

Let $A \in \mathbb{R}^{n \times q}$ be an ordered set of $n \in \mathbb{Z}^+$ *ascans*, with $a \in \mathbb{R}^q$. Though q varies depending on probe type, for the majority of what follows, work is done on *ascans* taken by 20 *MHz* [NB](#) probe for which $q = 3250$. Each *ascan* has a unique tuple (ϕ, δ, t) associated with its location and time of acquisition. Radial coordinate ϕ is in degrees, axial coordinate δ in meters(*m*), and time of acquisition t is in second(*s*). The *ascans* may be ordered by their evenly spaced locations, or times, of acquisition.

Throughout the various descriptions that follow. Array indexing is zero-based, and array indexing or slicing occurs as in the *python* programming language. Also, as in *python*, arrays of any type may be indexed by boolean arrays, or partition vectors as they are referred to. Arrays indexed by partition vectors carry the data ordering and spacing information of the arrays from which they were derived. [Savitzky-Golay Filter \(SGF\)](#) models [\[116\]](#) applied to arrays of length less than n (where n is the size of the complete dataset) use this persistent information with linear interpolation to return fits with dimension n .

To relieve possible confusion going forward, the array index of a feature in an *ascan*, the associated time of flight of the feature, and the associated probe to pipe wall [ID](#) distance are bijective mappings. Being in this sense synonymous, the terms are used interchangeably as context dictates. This makes sense because human error and inspection reporting requirements are given in terms of times of flight and distance, while algorithms are expressed in terms of indexes to arrays.

A time of flight t is generated from an index i as $t = \frac{i}{f}$ where f is the *ascan* sampling frequency of 250 *MHz*. The time of flight t of peak amplitude response in each *ascan* relates to the distance d between the probe and the pipe wall [ID](#) surface as $d = \frac{1}{2}v \cdot t$, where the speed of sound v in deuterium oxide is $\approx 1410 \frac{m}{s}$ at the pressures and temperatures observed during inspection, and division by two accounts for the fact that times of flight are round trip.

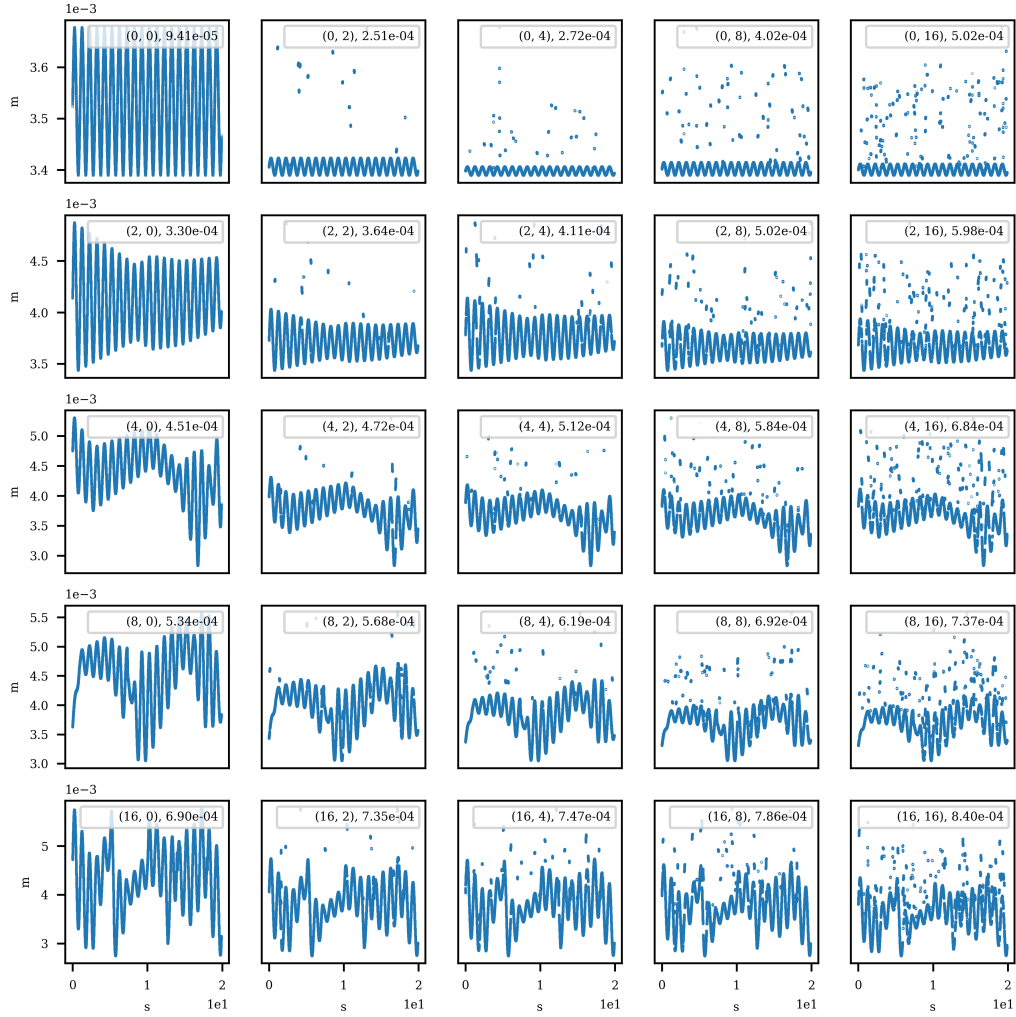


Figure 5.2: The tested synthetic datasets are shown with geometric regimes and flaw rates as given by their associated tuples. The standard deviation of each dataset is listed beside its identifying tuple. The scale of synthetic data approximately matches real data.

In some places, the vertical bar operator $|z|$ takes the absolute value of its operand z . In other places, vertical bars may denote cardinality of a set operand. $c \sim d$ is read c is distributed as d , and $c \approx d$ is read c is approximately distributed as d . $p(X)$ is the probability function of its operand, in this case X . We use probability statements and associated notation in a way that, if not abusive of notation, is at least mildly passive-aggressive; apologies for any confusion, annoyance, or mild heartburn this lack of formalism may elicit.

5.5 Chapter summary

This chapter introduced the datasets used to evaluate the performance of proposed methods for **UT** analysis. Three dataset types were described: real-world data from the **CF** and **PT HSA**, and a suite of synthetic datasets designed to stress-test method performance under simulated variations in data complexity.

The **CF** dataset represents an idealized case with smooth geometry and low noise, while the **PT HSA** dataset presents a more difficult context due to high geometric variability, surface fretting, oxidation, and sediment. Synthetic datasets simulate increasing levels of regime complexity and flaw density, allowing to benchmark methods beyond observed real-world conditions.

Standard deviation was used as a naive complexity metric to position real-world and synthetic datasets on a comparable scale. The real-world datasets were shown to be among the least complex by this measure, falling below all but one of the synthetic cases. The chapter concluded by describing nomenclature and conventions used throughout the document.

Chapter 6

VAE based analysis

Most of the material in this chapter was originally published in [131] and has been updated to align with the nomenclature that developed over the course of two subsequent publications [132] [133]. This chapter documents a first phase of research where the **Ultrasound (UT)** data analysis process is modeled using simple probabilistic statements and implemented by a **Variational Autoencoder (VAE)** that learns the approximate distributions of high-dimensional *ascan* datasets. The work was done to secure confidence in the capacity of machine-learning methods to perform tasks with **UT** datasets, provide a baseline for performance, and demonstrate a relation between results and assumptions about the dataset in a way communicable to technically oriented process stakeholders.

Proposed methods use the **Mean Squared Error (MSE)** of data reconstruction by a **VAE** to rank the likelihood of the data, and form binary classification partitions [12][72]. The **MSE** based ordering allows for the identification and removal of outliers, enhancing the efficacy of traditional model-fitting techniques. The reduction of outlier bias facilitates downstream deterministic modelling and enables **Nominal Profiling (NPROF)**, a method for characterizing nominal response, and deviation from it, in **UT** *ascans*, to produce type i/ii/iii analysis.

In this work, **VAE** models are used as tools, like slide rules or colanders, to measure and sift through each dataset independently. This approach minimizes reliance on out-of-distribution generalization and reduces model development, training, and maintenance efforts, particularly in managing the technical debt tied to dataset curation and maintenance [3][48].

Taken as a whole, the integrated techniques establish a stepwise process rooted in information theory and supported by a principled technical foundation [72]. The strength

of the integration lies in its straightforward and principled nature. It does not rely on out-of-distribution generalization, data preprocessing, curation, labelling, or require class balance. Its limitations include the lack of a strong fault diagnosis mechanism, the absence of robust error correction, explainability that requires interaction with information theory, and a dependence on a manually selected [MSE](#) threshold for data classification. Despite these weaknesses, the approach provides the desired analysis of type i/ii/iii data deviations and serves as a baseline for performance that may potentially be usable in safety-critical contexts.

6.1 A probabilistic model of data analysis

Fundamental to manual [UT Non Destructive Evaluation \(NDE\)](#) data analysis are estimates of unobserved nominal data. We model these estimates as $\sim p(X_n)$, where X_n is the available pseudo ground truth for dataset X . In the manual process, the first step in generating estimates is the classification of observed data X as nominal or anominal [\[134\]\[135\]](#). Here, classification is implemented by the construction of a Boolean partition vector P_V so that $X[P_V]$ selects nominal data, and $X[\neg P_V]$ selects anominal data. Given P_V accurately classifies X , we can assume $X[P_V] \approx p(X_n)$ and proceed to use some type of model fitting to fill the gaps in $X[P_V]$ at $\neg P_V$. It is reasonable to assert that this method is effective only if there is strong nominal majority in the dataset, i.e., when $|X[P_V]| \gg |X[\neg P_V]|$. Since anomalies are the exception in safety-critical contexts, this assumption may be justified.

Given $X[P_V] \approx p(X_n)$, the process of estimating X_n may be modelled as drawing:

$$\hat{X} \sim p(\hat{X}|X[P_V]) \tag{6.1}$$

Where $\hat{X}$ is an estimate of pseudo ground truth X_n . Sampling $\sim p(\hat{X}|X[P_V])$ is approximated using [Savitzky-Golay Filter \(SGF\)](#) $M_\alpha(X[P_V])$ which fits a curve on the basis of evidence $X[P_V]$. What remains is a means to construct a relevant P_V .

6.2 Method description

Let dataset A be a set of *ascans* taken from a [Canada Deuterium Uranium \(CANDU\) Calibration Fixture \(CF\)](#) or [Pressure Tube \(PT\)](#). Thresholds on M_V^A , the [MSE](#) of training data reconstruction of [VAE](#) V trained on A , are used to construct a partition vector P_V

and, slightly more restrictive P'_V . P_V completes the implementation of Expression 6.1 and obtains estimates $\hat{X} \sim p(\hat{X}|X[P_V])$.

Estimates $\hat{X}$ are used to eliminate the noise due to rotational wobble in A and construct denoised A' . Maximum nominal response profile N_{mx} , characterizing the nominal component in A , is constructed as $A'[P'_V].mx()$. N_{mx} is then compared to each element of A' to obtain A'' , where A'' recovers the times of flight of features in A' that exceed the nominal profile N_{mx} . This provides type i/ii analysis. Similarly, the mean nominal response profile N_{mn} is constructed as $A'[P'_V].mn()$ and provides type iii analysis by way of comparison with each denoised *ascan* in A' . This provides full analysis of type i/ii/iii data deviations in A . These steps are developed as pseudocode in Algorithm 1.

6.3 MSE based data likelihood ranking

VAE are parameterized neural networks that consist of an encoder and decoder. The encoder compresses input data to a low-dimensional learned representation, and the decoder works in the opposite direction, reconstructing learned representations to their original data dimension. The objective is to learn network weights for the encoder and decoder that generate reconstructions as similar to the input data as possible [72].

VAE train by gradient descent with a loss function, based on the Kullback Leibler Divergence(D_{KL}) [84], a measure of the distance between the distributions implied by the neural network and the true data distribution. But computation dependent on the unknown true data distribution is intractable. Instead, a bound on the true data distribution, given by the Evidence Lower Bound (ELBO) and dependent only on the observed data distribution, is optimized. And so the VAE loss function computes the ELBO, and the gradient descent method moves the encoder and decoder weights in a direction that maximizes its value.

Unless the input data is pathologically uninformative VAE compression is guaranteed to cause information loss [72]. In the case of 20 MHz Normal Beam (NB) data, *ascans* are compressed from $q = 3250$ sampled values down to a representation of 32 values from which they are reconstructed. VAE optimization seeks to mitigate the impact of this information loss on reconstruction error. In each training pass the optimization schedule updates the network weights so that the information encoded is that to which the aggregate loss function is most sensitive. The aggregate behavior of optimization and the assumption regarding a nominal majority in the dataset imply that loss of data reconstruction M_V^A provides an ordering that may be used to generate relevant partitions of UT datasets [6][72]. So data is ordered and partitioned by training V on A to obtain M_V^A .

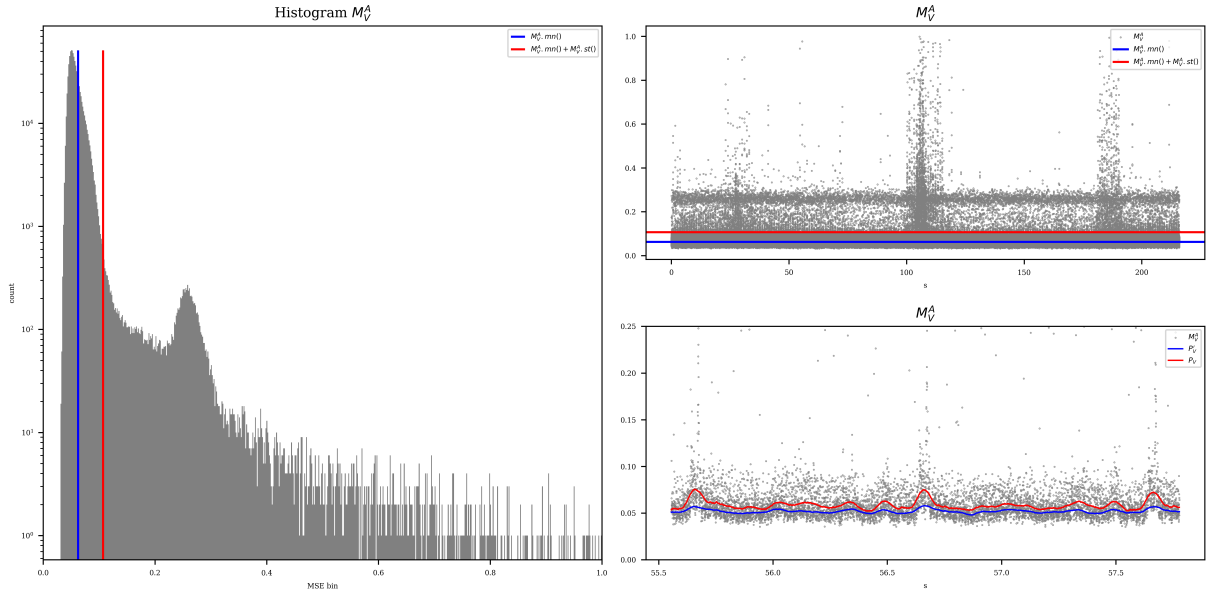


Figure 6.1: With CF dataset $|A| \sim 8e5$, the leftmost plot is a histogram of M_V^A on $10e3$ bins. In the column on the right, the time axes order M_V^A in terms of the time of acquisition of *ascans* in the CF dataset. The top right plot shows M_V^A for the entire dataset. The bottom-right plot displays M_V^A over two rotations of the tool head with Savitzky-Golay based dynamic threshold determining P_V and P'_V .

6.4 Taking estimates

We desire P_V such that $X[P_V] \approx p(X_n)$ with cardinality $|X[P_V]|$ large enough to ensure $M_\alpha(X[P_V]) \approx p(X_n)$. The bottom right panel of Figure 6.1 features a plot of M_V^A over two revolutions of the UT probe. The plot reveals a clear correlation between M_V^A and periodic tool wobble every second, or 3600 *ascans*. Thus, dynamic SGF-based thresholds are selected to provide dense nominal support throughout the dataset. To generate P_V , first set $P_V = M_V^A < M_V^A.mn() + M_V^A.st()$, and then generate P_V with dynamic threshold as $P_V = M_V^A < M_\alpha(M_V^A[P_V])$. Estimates $\hat{X}$ are then obtained as $\hat{X} = M_\alpha(X[P_V])$.

Algorithm 1 NPROF: Given A , $\hat{X}$, M_V^A , and P_D returns A'' and S . A'' contains the time of flight of possible type i and ii data deviations relative to $\hat{X}$. S provides a type iii characterization based on summary statistics. P_D is an optional argument included to accommodate integration with DiffPaC. P_D defaults to all *True* if not supplied.

Input: $A, \hat{X}, M_V^A, P_D$
 $A', A'' \leftarrow A, A$
 $P'_V \leftarrow (M_V^A < M_\alpha(M_V^A[M_V^A < M_V^A.mn()]))) \wedge P_D$
for $i \in [0, \dots, n]$ **do**
 $A'[i, :] \leftarrow leftShift(A'[i, :], \hat{X}[i])$
end for
 $N_{mx} \leftarrow A'[P'_V].mx()$
 $N_{mn} \leftarrow A'[P'_V].mn()$
for $i \in [0, \dots, n]$ **do**
 $S \leftarrow |A'[i, :] - N_{mn}|.mn()$
 $A''[i, :] \leftarrow rightShift(A'[i, :] - N_{mx}, \hat{X}[i])$
end for
 $A''[A'' \leq 0] \leftarrow 0$
return A'', S

6.5 Nominal profiling

As outlined in Algorithm 1, given A , $\hat{X}$, and M_V^A (with P_D being optional), NPROF performs type i/ii/iii analysis by first isolating and characterizing the highly likely nominal subset of A identified by P'_V . And then reporting variance in the comparisons of each denoised *ascan* in A' and nominal profiles N_{mx} and N_{mn} generated on the basis of P'_V .

The comparison is enabled by a denoising process that uses $\hat{X}$ to eliminate noise caused by the rotational wobble of the tool during inspection. This wobble, clearly depicted in the second-from-left plot of Figure 6.2, exhibits as non-stationary, periodic, oscillating, positioning error in X . $\hat{X}$ estimates this noise and is used to transform A to denoised dataset A' . $A'[P'_V]$ is then used to generate nominal profiles $N_{mx} = A'[P'_V].mx()$ and $N_{mn} = A'[P'_V].mn()$. Type i/ii/iii data deviations are then identified by comparing the denoised *ascans* in A' to the nominal profiles.

The sensitivity of this analysis is dependent on cardinality $|A[P'_V]|$. To understand this, consider that as $|A[P'_V]|$ approaches $|A|$ anomalous *ascan* information is admitted to $A'[P'_V]$; this decreases the sensitivity of comparative analysis using profiles based on P'_V . Using threshold $M_\alpha(M_V^A[M_V^A < M_V^A.mn()])$ ensures that $A'[P'_V]$ contains information about only commonly occurring and easily reconstructed *ascan* features.

Thus, having P'_V , A , and $\hat{X}$ in hand enables the application of Algorithm 1 to arrive at A'' containing the times of flight, if they exist, of the exceedances of each *ascan* relative to the bounds provided by the maximum nominal profile N_{mx} , and S the mean difference of each *ascan* relative to the mean nominal profile N_{mn} .

6.6 Results and discussion

The proposed techniques are applied to the CF dataset, and estimates for flaw dimensional analysis are obtained. First, a VAE is trained to reconstruct the CF dataset. Then, dynamic thresholds on M_V^A used to determine P_V and generate $\hat{X}$. Finally NPROF, as given in Algorithm 1, is applied to obtain A'' and S which provide the desired type i/ii/iii analysis.

The region plotted in Figure 6.2 contains the greatest density of flaws in the CF dataset. By inspection, all type i/ii data deviations corresponding to notches $Q, R, S, T, U, V, W, X, Y$, and Z (as in Figure 5.1) are located. The leftmost panel plots A'' as an image and provides a depth map of the calibration facility inner diameter surface. Though A'' is based on a restrictive P'_V , it still contains false positives; thus Density Based Spatial Clustering of Applications with Noise (DBSCAN) is used to remove randomly occurring singleton exceedances in A' [42].

The second panel in Figure 6.2 shows the estimated $\hat{X}$ tracking tool wobble well. The accuracy of this estimate is critical to the denoising process that generates nominal profiles for the dataset. Further evidence of this goodness of fit is given in the top row plots of

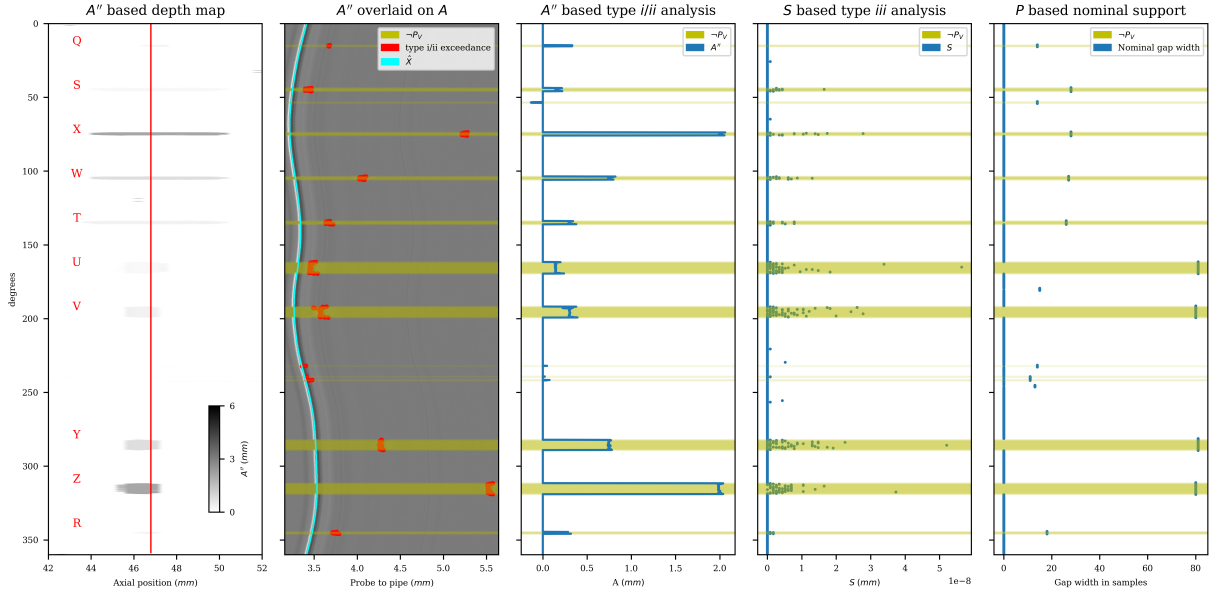


Figure 6.2: The plots visualize the various components of NPROF generated type i/ii/iii analysis. The leftmost plot gives a denoised depth map provided by the time of flight of maximum values in each row of A'' ; this constitutes type i/ii analysis. The second plot gives the *bscan* along the red line of the leftmost plot. The olive shaded bands highlight *ascans* selected as anomalous by $\neg P'_V$. The time of flight of the maximum exceedances in A'' relative to N_{mx} are plotted in red, and the cyan trace gives $\hat{X}$. The third panel plots information in A'' . The fourth plot shows type iii analysis as given by S . And the last panel provides the width of gaps in nominal support, where nominal support gaps of width ten samples or less are filtered from view.

Figure 6.4, which show absolute error $|\hat{X} - X_n| \leq 9.23 \cdot 10^{-6} m$ and absolute nominal error $|\hat{X}[P] - X[P]| \leq 3.33 \cdot 10^{-5} m$.

Flaw axial, radial, and depth dimensional estimates may be taken directly from A'' . The third panel in Figure 6.2 shows A'' over the same region of the dataset as in the second panel. Radial and depth dimension estimates may be taken directly from the plot as shown, while the axial dimension may be estimated by considering the number of rotations over which A'' exceedances are identified.

Measures of the quality of data on which dimensional estimates are based are given in panels four and five of Figure 6.2. Panel four provides S , which gives the mean difference of each *ascan* with N_{mn} over the range depicted. Consistent with Figure 3.6, panel four shows a region with noisy *ascans* between 225 and 250 degrees. And as expected, it identifies significant variance with nominal in the *ascans* associated with regions where A'' exceedances occur. The density of nominal support for estimates $\hat{X}$ is an important confidence measure as high density ensures $M_\alpha(X[P_V]) \approx p(X_n)$. The rightmost plot in Figure 6.2 shows that a significant majority of the data in $X[P_V]$ are within one degree (ten samples) of at least one other nominally classified datum in $X[P_V]$. Nominal support over the range is given in panel five, where it can be seen that gaps in support occur mainly at regions with identified A'' exceedances. Both S and nominal support may be used in support of arguments to demonstrate inspection processes have met data quality requirements as given in [150].

The information in A'' is used to automatically generate the dimensional estimates given in Table 6.1, which also provides the as manufactured specified dimension with error tolerances, and figures for required accuracy of analysis in each dimension [150]. Kernel density estimation is used to process the information in A'' and determine the most likely values for dimensional estimates [119]. Figure 6.3 plots the results given in Table 6.1.

Figure 6.4 characterizes error and data quality. The first row of plots gives the difference between nominal estimates and pseudo ground truth $|\hat{X} - X_n|$. The maximum difference is $9.23 \cdot 10^{-6} m$. Although well within the desired range of error ($\pm 2.0 \cdot 10^{-5} m$) in practice, without reference X_n other means are required to verify model performance. The second row of plots characterizes $|\hat{X}[P] - X[P]|$, the difference between estimates of nominally classified data and the nominally classified data. The maximum difference is $3.33 \cdot 10^{-5} m$. If using this metric in a safety-critical process, differences above $2.00 \cdot 10^{-5} m$ would likely require some sort of manual disposition [135][150].

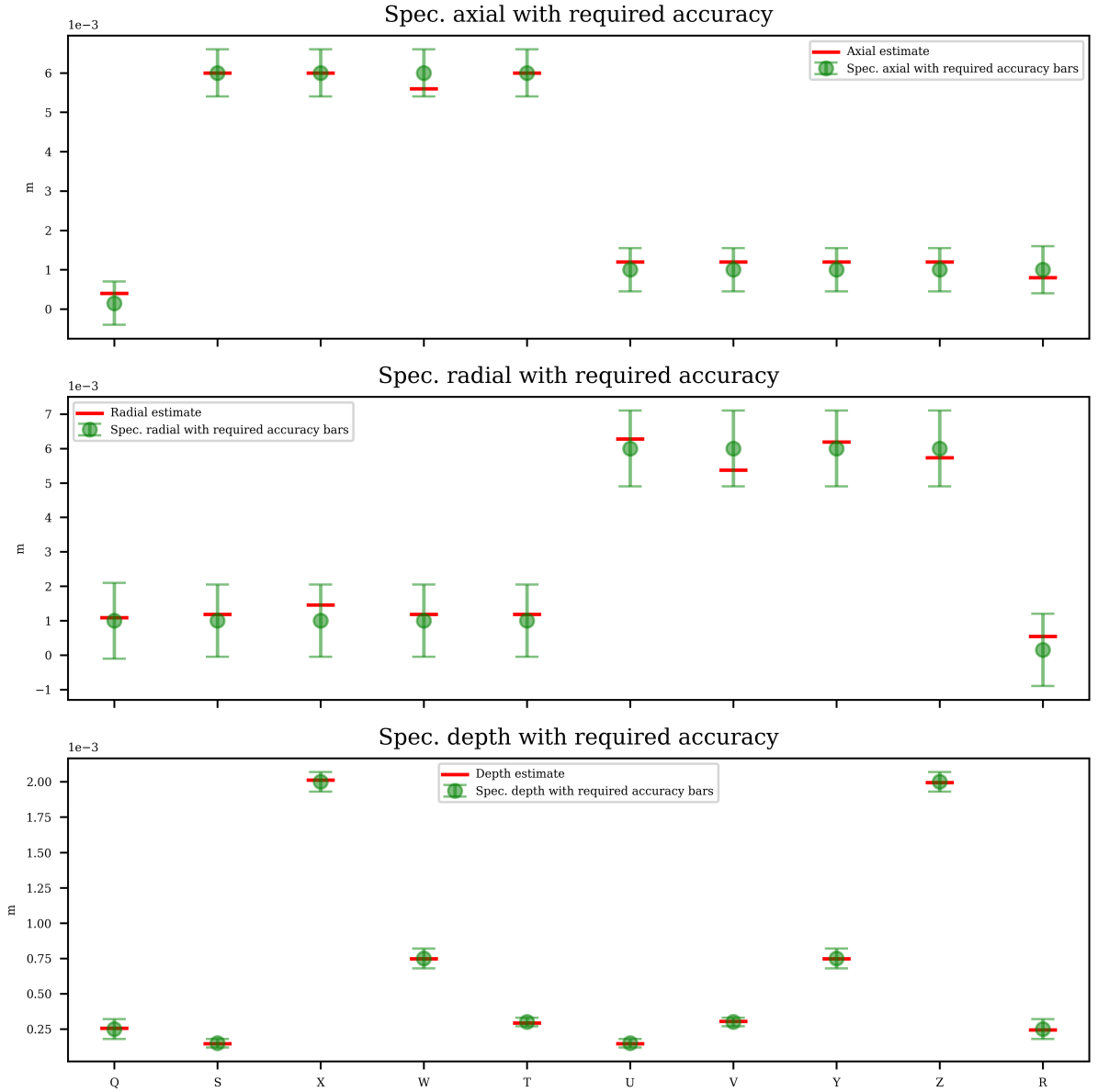


Figure 6.3: Results from Table 6.1, which provide dimensional analysis of CF notches $Q, R, S, T, U, V, W, X, Y$, and Z , are plotted. Green bars reflect the sum of error tolerance and the required accuracy of analysis in the relevant dimension.

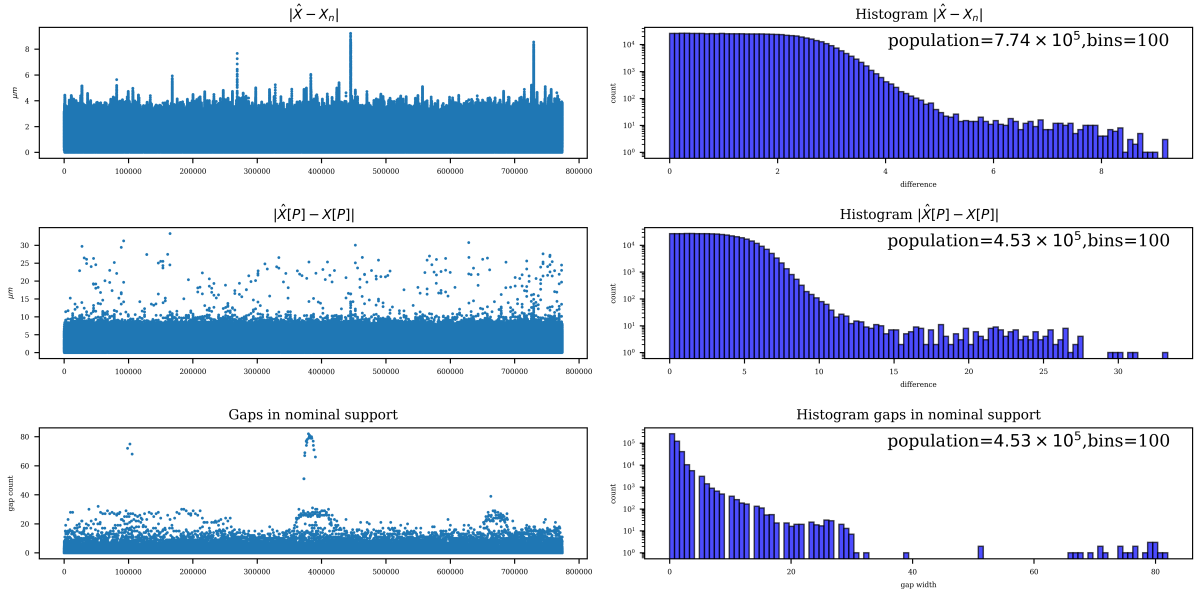


Figure 6.4: Error, confidence, and nominal support measures. The top plots give the absolute error of estimates $\hat{X}$ vs. pseudo ground truth X_n . The middle plots show the error between estimates and nominally classified data, which is an error metric available in the absence of ground truth. The bottom row quantifies nominal support for estimates.

Notch	Dimensional estimate			Dimensional specification			Required accuracy		
	Axial	Radial	Depth	Axial	Radial	Depth	Axial	Radial	Depth
Q	$4.00 \cdot 10^{-4}$	$1.09 \cdot 10^{-3}$	$2.57 \cdot 10^{-4}$	$1.50 \cdot 10^{-4} \pm \epsilon_1$	$1.00 \cdot 10^{-3} \pm \epsilon_2$	$2.50 \cdot 10^{-4} \pm \epsilon_1$	$5.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-5}$
S	$6.00 \cdot 10^{-3}$	$1.18 \cdot 10^{-3}$	$1.47 \cdot 10^{-4}$	$6.00 \cdot 10^{-3} \pm \epsilon_2$	$1.00 \cdot 10^{-3} \pm \epsilon_1$	$1.50 \cdot 10^{-4} \pm \epsilon_3$	$5.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-5}$
X	$6.00 \cdot 10^{-3}$	$1.46 \cdot 10^{-3}$	$2.01 \cdot 10^{-3}$	$6.00 \cdot 10^{-3} \pm \epsilon_2$	$1.00 \cdot 10^{-3} \pm \epsilon_1$	$2.00 \cdot 10^{-3} \pm \epsilon_1$	$5.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-5}$
W	$5.60 \cdot 10^{-3}$	$1.18 \cdot 10^{-3}$	$7.47 \cdot 10^{-4}$	$6.00 \cdot 10^{-3} \pm \epsilon_2$	$1.00 \cdot 10^{-3} \pm \epsilon_1$	$7.50 \cdot 10^{-4} \pm \epsilon_1$	$5.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-5}$
T	$6.00 \cdot 10^{-3}$	$1.18 \cdot 10^{-3}$	$2.93 \cdot 10^{-4}$	$6.00 \cdot 10^{-3} \pm \epsilon_2$	$1.00 \cdot 10^{-3} \pm \epsilon_1$	$3.00 \cdot 10^{-4} \pm \epsilon_3$	$5.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-5}$
U	$1.20 \cdot 10^{-3}$	$6.28 \cdot 10^{-3}$	$1.47 \cdot 10^{-4}$	$1.00 \cdot 10^{-3} \pm \epsilon_1$	$6.00 \cdot 10^{-3} \pm \epsilon_2$	$1.50 \cdot 10^{-4} \pm \epsilon_3$	$5.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-5}$
V	$1.20 \cdot 10^{-3}$	$5.37 \cdot 10^{-3}$	$3.05 \cdot 10^{-4}$	$1.00 \cdot 10^{-3} \pm \epsilon_1$	$6.00 \cdot 10^{-3} \pm \epsilon_2$	$3.00 \cdot 10^{-4} \pm \epsilon_3$	$5.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-5}$
Y	$1.20 \cdot 10^{-3}$	$6.19 \cdot 10^{-3}$	$7.47 \cdot 10^{-4}$	$1.00 \cdot 10^{-3} \pm \epsilon_1$	$6.00 \cdot 10^{-3} \pm \epsilon_2$	$7.50 \cdot 10^{-4} \pm \epsilon_1$	$5.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-5}$
Z	$1.20 \cdot 10^{-3}$	$5.73 \cdot 10^{-3}$	$1.99 \cdot 10^{-3}$	$1.00 \cdot 10^{-3} \pm \epsilon_1$	$6.00 \cdot 10^{-3} \pm \epsilon_2$	$2.00 \cdot 10^{-3} \pm \epsilon_1$	$5.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-5}$
R	$8.00 \cdot 10^{-4}$	$5.46 \cdot 10^{-4}$	$2.45 \cdot 10^{-4}$	$1.00 \cdot 10^{-3} \pm \epsilon_2$	$1.50 \cdot 10^{-4} \pm \epsilon_1$	$2.50 \cdot 10^{-4} \pm \epsilon_1$	$5.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-5}$

Table 6.1: Dimensional estimates, specified dimensions (with error tolerances), and required accuracies, in meters. Let $\epsilon_1 = 5.0 \cdot 10^{-5} m$, $\epsilon_2 = 1.0 \cdot 10^{-4} m$, $\epsilon_3 = 1.0 \cdot 10^{-5} m$.

6.7 Conclusion

We have proposed a [VAE](#)-based method that uses [NPROF](#) to provide decision support and automation to [CANDU NDE UT](#) data analysis. This method addresses many of the drawbacks of prior similar work based on supervised training or heuristics. In particular, it provides type i/ii/iii analysis that does not depend on dataset curation, labeling, or management of any kind and does not depend on out of distribution generalization.

Experimental results show the method performs well and meets requirements for accuracy with respect to specified dimensions of the [CF](#). However, there are no strong means for fault diagnosis in the absence of ground truth. $|\hat{X} - X_n|$ is available but blind to all regions with nominal support gaps. Thus, the method does not include means for robust fault diagnosis.

In terms of a principled basis and explainability, each step in the process has a clear and transparent methodology. Model behavior may be understood in terms of probabilistic statements and some basic information theory. However, the interaction with information theory required to fully understand the method, challenges explainability.

The choice of a threshold values to partition [MSE](#) is somewhat arbitrary. While the ordering provided to M_V^A implies the general properties of the partitions that may be derived from it, there are no means by which to determine optimal thresholds.

As demonstrated, outputs are sufficient to support downstream data related tasks such as flaw dimensional analysis. A host of unsupervised clustering, object detection, edge detection, and segmentation techniques exist that may support this task more ably than the extremely simple kernel density estimation technique applied.

Chapter 7

Diffusion ablation

In Chapter 6 we demonstrated the capacity of a [Variational Autoencoder \(VAE\)](#) and [Nominal Profiling \(NPROF\)](#) to provide type i/ii/iii for the analysis of the calibration dataset. Results met requirements for accuracy established by regulatory bodies but fell short on explainability and fault diagnosis.

Having achieved those results, the research investigated means to improve confidence in estimates by performing the same analysis using a second independent approach. This is in keeping with safety-critical contexts that often rely on the consensus of multiple independent voices to establish confidence in results and use disagreement to flag error. But to mitigate the risk of common mode faults, a second redundant system must be sufficiently independent [\[115\]](#).

To achieve the required independence of methods, a univariate approach based on the nominal peak amplitude times of flight X rather than A was devised. This change in data dimension necessarily reduced the number of spurious correlations in the dataset and improved resource loading. These positively impacted performance relative to the requirements listed in Chapter 2.

Research began by considering various machine learning models for time series analysis. Detailed evaluation involved several models regarded as state-of-the-art. Comparing outcomes, it became apparent that the ability to smooth the presence of outliers in a largely unimodal distribution varied. It was also apparent that all models were susceptible to overfitting and often appeared to memorize data.

The behavior relative to over-fitting seemed important. If an industrial process includes elements that monitor or guard against over-fitting, then the explanation of the process

will likely involve some interaction with advanced concepts in machine learning; this challenges explainability. For this reason, the proposed method in Chapter 8 is designed to be insensitive to overfitting.

Among the models evaluated, diffusion processes were found to be able to learn approximate data distributions. The generative aspect of these models facilitated working with datasets with missing data and supported sampling for both observed and missing values. In the final analysis, two models, [Conditional Score Based Diffusion \(CSDI\)](#) and [Denoising Diffusion Probabilistic Model \(DDPM\)](#), were leading candidates for in-depth evaluation.

[CSDI](#) is an extension of a [DDPM](#) model for time series that includes a score-based loss function and a conditional inference mechanism. [CSDI](#) differentiates itself by exhibiting state-of-the-art results on numerous bench-marks and time series performance metrics [128].

This chapter compares the performance of [CSDI](#) and [DDPM](#) architectures. Models are tested on the [Calibration Fixture \(CF\)](#) dataset with data classified by a series of candidate binary partitions generated using M_V^A . Data classified as anomalous is presented to the models as unobserved. The results demonstrate that [CSDI](#) achieves superior results relative to [DDPM](#), given datasets with outliers and possibly missing data,

7.1 Analysis and the manifold

[Ultrasound \(UT\)](#) data analysis often involves the identification and characterization of variance from nominal trend in some set of quantities assumed to be continuous. In most industrial contexts anomalies have a relatively low rate of occurrence. This puts discriminative learning methods which rely on large datasets with balanced classes at a disadvantage [24][51]. The premise on which this research rests is that generative models like [VAE](#) and [Diffusion Probabilistic Model \(DPM\)](#) are accommodating in this regard and can be made to access means to learn a prevalent nominal trend and predict it when unobserved. Additionally, the probabilistic expressions used to describe these models lend well to the expression of [UT](#) data analysis as sampling from their learned approximate data distributions. This facilitates a principled basis on which a technical justification for the use of generative methods may be based.

An extension of the manifold hypothesis [10] is that there exists a set of distinct data-generating factors, each contributing to the distribution of an observed data set [54]. Given this is true, it could be argued that the process of manual [UT](#) data analysis is simply to

classify the data caused by these factors as nominal or anominal and then use the classification to identify and infer a possibly obscured nominal trend. Thus, the existence of two groups of factors in [Non Destructive Evaluation \(NDE\)](#) datasets is assumed. A nominal group consists of all factors supporting the smooth operation of the data acquisition system over a continuous surface of the material under test. And an anominal set of data generating factors causing spurious or noisy operation in data acquisition and unanticipated discontinuity in otherwise continuous features of the material under test. On the basis of this understanding, the task of analysis is reframed as the determination of a binary data classifying partition, over nominal and anominal data, and subsequent estimation of a prevalent nominal trend.

Traditional approaches establishing a useful partition may involve some form of data classification and curve fitting. For instance, some threshold may be used to partition observed data, and then some curve fitting technique applied to identify nominal trend. However, accurate identification of the nominal trend is sensitive to the quality of the partition. Given a correct partition, the relative ease of identifying nominal trend follows. But as in [Figure 7.1](#), heuristic approaches are brittle and fail when the partition mixes support from nominal and anomalous data generating factors. This detracts from the usefulness of such methods and leads to the consideration of a probabilistic approach.

7.2 A probabilistic model for UT data analysis

Given an ordered set of *ascans* A taken as a [UT](#) probe passes over the surface of a test subject. Let dataset X be the peak amplitude response time of flight of the *ascans* in A . Let P be a boolean partition vector over X such that:

$$X = X[P] \cup X[\neg P] \quad (7.1)$$

According to the given understanding of data-generating factors, X is distributed as joint probability:

$$X \sim p(X[P], X[\neg P]) \quad (7.2)$$

Let $X[P]$ select observed data that is classified nominal, and $X[\neg P]$ select observed data classified anominal. A method is sought to generate estimates $\hat{X} \approx p(X_n)$ where X_n is pseudo ground truth for X .

Analysts make desired estimates $\hat{X}$ by first classifying observed data as either nominal or anomalous and then estimating unobserved nominal data on the basis of the observed

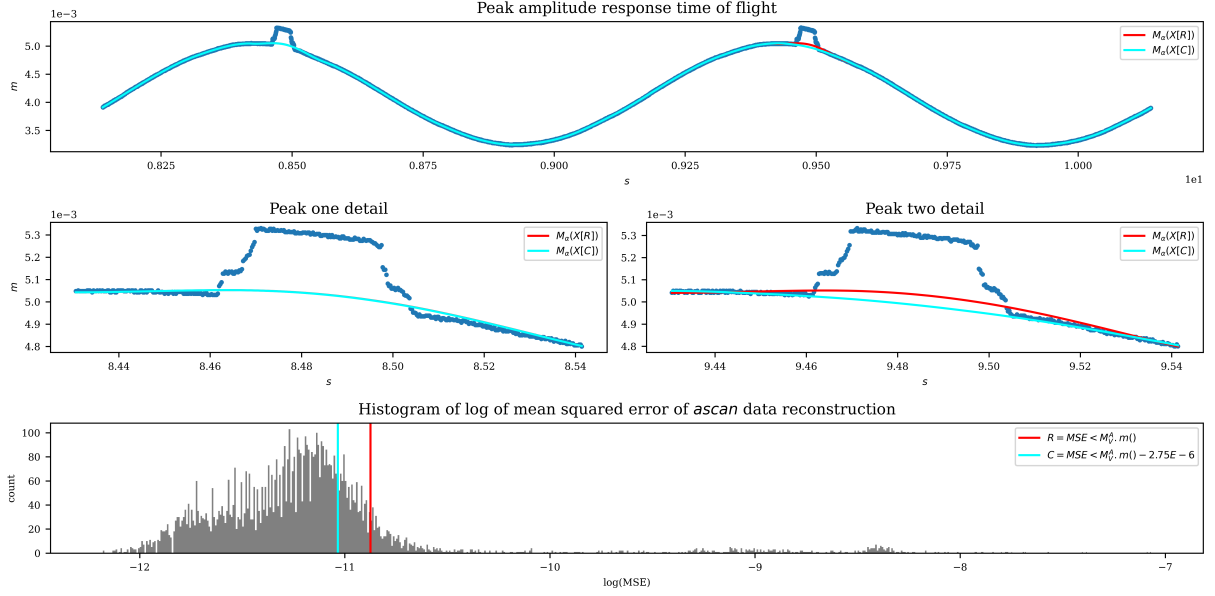


Figure 7.1: The upper plot shows the peak amplitude response time of flight X for a subset of *ascans* in A taken in a PT over two rotations of the probe. The depicted notch in the PT is a void in material taken as part of a scrape inspection. Detail of the two passes over the notch is given in the second row of plots. The lower plot gives a histogram of VAE error of data reconstruction M_V^A . The red and cyan lines on the histogram are at $M_V^A.m()$ and $M_V^A.m() - 2.75E - 6$ respectively. These thresholds are used to form partitions $R = \text{MSE} < M_V^A.m()$ and $C = \text{MSE} < M_V^A.m() - 2.75E - 6$, and generate curves $M_\alpha(X[R])$ and $M_\alpha(X[C])$. Across the first peak neither curve tracks the nominal trend well but are nearly identical. On the second peak the curves diverge so that the cyan trace appears to represent the nominal trend.

nominal. This estimation process is modeled as sampling estimates $\hat{X}$ with a conditional dependence on observed nominal:

$$\hat{X} \sim p(\hat{X}|X[P]) \quad (7.3)$$

Because [Pressure Tubes \(PTs\)](#) are well maintained, and data is acquired with very strict quality controls, it is expected that cardinality $|X[P]| \gg |X[\neg P]|$. This implies that in general $p(X[P]|X) \gg p(X[\neg P]|X)$ and allows to consider the naive approximation:

$$\hat{X} \sim p(\hat{X}|X) \quad (7.4)$$

Equations [7.3](#) and [7.4](#) are referred to as the Nominal Data Model (NDM) and Observed Data Model (ODM), respectively.

7.3 Diffusion Probabilistic Models

Diffusion Probabilistic Models(DPM) are a class of latent variable models of the form $p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})d\mathbf{x}_{1:T}$ where:the data vector $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ is of the same dimension as latent vectors $\mathbf{x}_1, \dots, \mathbf{x}_T$, p is parameterized by θ , and T is some positive integer [\[55\]](#). The model chains the input variable vector $\mathbf{x}_0$ with the T latent variable vectors and defines a forward and reverse process over the chain. The model architecture presumes a learned Gaussian transition from each vector in the chain to the next, with each transition parameterized by a neural network of input and output dimension equal to the input variable vector.^{[1](#)}

The approximate posterior $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$, or forward process, is defined as a Markov chain that transforms input data vectors to a target distribution by gradually adding noise at each of T steps according to a learnable variance schedule $\beta_1, \dots, \beta_T$:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := N(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad (7.5)$$

¹We assume that the reader is familiar with the semantics and notation that is the lingua franca of variational inference and diffusion processes so that this high-level overview may serve as a refresher. For a full treatment and development of variational inference and [VAE](#) see [\[73\]](#), for Denoising Diffusion Probabilistic Models see [\[55\]](#), and for Conditional Score-Based Diffusion see [\[128\]](#).

The reverse process is a joint probability $p_\theta(\mathbf{x}_{0:T})$ defined as a Markov chain starting at $p(\mathbf{x}_T) = N(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$ that transforms its input across T learned Gaussian transitions:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := N(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)\mathbf{I}) \quad (7.6)$$

Diffusion models admit closed-form expressions for sampling $\mathbf{x}_t$ at arbitrary step t . Being differentiable, they support training by stochastic gradient descent to optimize the variational lower bound L on negative log likelihood:

$$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] = \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] := L \quad (7.7)$$

Where $\mathbb{E}$ is the expectation, and $\mathbb{E}_q$ is the expectation taken with respect to q . Sampling of $\mathbf{x}_t$ at arbitrary time step t is achieved as:

$$q(\mathbf{x}_t|\mathbf{x}_0) = N(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I}) \quad (7.8)$$

Where $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$.

Denoising Diffusion Probabilistic Models (DDPM) reformulate the variational bound in Equation 7.7 [55] as:

$$L := \mathbb{E}_q \left[D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T)) + \sum_{t \geq 1} D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right] \quad (7.9)$$

Where the forward process variance schedule β_t , which may be learned, is fixed to constants. This removes all learnable parameters from the first term of Equation 7.9; removing it from consideration during training. The reverse process covariance matrix $\Sigma_\theta(\mathbf{x}_t, t) = \sigma^2 \mathbf{I}$ is also set to step-dependent unlearned constants, which removes it from consideration during training as well. Parameterizing the sampling procedure in Equation 7.8 with $\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I})$, leads to the parameterization:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\alpha_t} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \alpha_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right), \quad \sigma_\theta(\mathbf{x}_t, t) = \tilde{\beta}_t^{1/2}, \quad (7.10)$$

$$\text{where } \tilde{\beta}_t = \begin{cases} \frac{1 - \alpha_t - 1}{1 - \alpha_t} \beta_t & t > 1 \\ \beta_1 & t = 1 \end{cases} \quad (7.11)$$

ϵ_θ is a learnable denoising function that reverses the forward process. With this parameterization, the [DDPM](#) loss function reduces to:

$$L_{DDPM}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \epsilon} [\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)\|^2] \quad (7.12)$$

The [DDPM](#) model includes a decoder in the final step of the reverse process to improve sampling. The decoder relies on the linear scaling of data on the range $[-1, 1]$ and allows the direct sampling of $\mu_\theta(\mathbf{x}_1, 1)$ without any addition of noise.

[CSDI](#) [128] models introduce a conditional into the reverse process of the [DDPM](#) as:

$$p_\theta(\mathbf{x}_{0:T}^{ta} | \mathbf{x}_0^{co}) := p(\mathbf{x}_T^{ta}) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}^{ta} | \mathbf{x}_t^{ta}, \mathbf{x}_0^{co}), \quad \mathbf{x}_T^{ta} \sim N(\mathbf{0}, \mathbf{I}), \quad (7.13)$$

$$p_\theta(\mathbf{x}_{t-1}^{ta} | \mathbf{x}_t^{ta}, \mathbf{x}_0^{co}) := N(\mathbf{x}_{t-1}^{ta}; \mu_\theta(\mathbf{x}_t^{ta}, t | \mathbf{x}_0^{co}), \sigma_\theta(\mathbf{x}_t^{ta}, t | \mathbf{x}_0^{co}) \mathbf{I}) \quad (7.14)$$

Where imputation targets $\mathbf{x}_0^{ta} \in X^{ta}$ are unobserved data, and $\mathbf{x}_0^{co} \in X^{co}$ are the observed data on which estimates of the unobserved data are made conditional. Introducing a conditional into the expression for ϵ_θ in the [DDPM](#) parameterization in Equation 7.10, we arrive at the parameterization for [CSDI](#):

$$\mu_\theta(\mathbf{x}_t^{ta}, t | \mathbf{x}_0^{co}) = \frac{1}{\alpha_t} \left(\mathbf{x}_t^{ta} - \frac{\beta_t}{\sqrt{1 - \alpha_t}} \epsilon_\theta(\mathbf{x}_t^{ta}, t | \mathbf{x}_0^{co}) \right) \quad (7.15)$$

$$\sigma_\theta(\mathbf{x}_t^{ta}, t | \mathbf{x}_0^{co}) = \tilde{\beta}_t^{1/2}, \quad \text{where } \tilde{\beta}_t = \begin{cases} \frac{1 - \alpha_t - 1}{1 - \alpha_t} \beta_t & t > 1 \\ \beta_1 & t = 1 \end{cases} \quad (7.16)$$

Which implies the corresponding [CSDI](#) loss function:

$$\min_{\theta} L(\theta) := \min_{\theta} \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim N(\mathbf{0}, \mathbf{I}), t} \|\epsilon - \epsilon_\theta(\mathbf{x}_t^{ta}, t | \mathbf{x}_0^{co})\|_2^2 \quad (7.17)$$

[CSDI](#) implements a self-supervised learning method inspired by training in masked language models, which holds back observed data, in an amount determined by the “missing ratio” hyperparameter, as simulated unobserved targets with ground truth. The [CSDI](#) architecture uses an attention mechanism with multi-head and fully connected layers to identify temporal and feature dependencies. The feature dependencies are used to generate “side information” that is passed to the gated activation unit in each residual diffusion layer. The diffusion layers are each composed as a U-Net with skip connections, all passing information to the final convolutional output layers.

7.4 Experiments and results

Experiments are performed in order to uncover detail concerning model performance and behaviour. Results are interpreted in terms of their agreement with the probabilistic models developed in Section 7.2. Alignment between the models and their implementations will be taken to imply a principled basis to which an understanding of performance may be fixed. The hope is that this may, in the future, provide the basis for a technical justification of novel methods.

Relevant observations are made using a series of training datasets constructed using a series of partitions on X . Manual classification is first used to construct a pseudo ground truth partition vector Q , and pseudo ground truth X_n is constructed as $X_n = M_\alpha(X[Q])$. Then, similar to Chapter 6, a series of partition vectors based on percentile values M_V^A is generated. So that given percentile value p , partition vector P_p is constructed as $P_p = M_V^A < M_V^{A,p}$. This allows the construction of datasets $X[P_p]$ for a range of p .

DDPM and **CSDI** diffusion models are trained on each of the datasets for five epochs. Implementations of the models are based on the code provided with [128]. The **CSDI** conditional mechanism requires a missing ratio hyperparameter that determines the percentage of observed data held back as simulated ground truth during training. This percentage regulates the degree to which the conditional mechanism of the model is trained on observed data. **DDPM** models do not exercise a conditional mechanism when training, but their performance on datasets with missing data is also tested. The models are tested on the range [10%, 20%, 30%, 40%, 50%] of missing data rates.

CSDI and **DDPM** models are trained on each training dataset for five epochs. Then, five estimates of each data-point in the dataset are sampled from each trained model to obtain X_D . The mean and standard deviation of each five wide set is taken to obtain X_D^m and X_D^s . Using mask $X_D^s < 1$ curve $M_\alpha(X[X_D^s < 1])$ is generated. Then, standard error measures for each curve are taken with respect to X_n .

Figure 7.2 shows the dynamics related to model training. In general, the **CSDI** models outperform their **DDPM** counterparts and, with few exceptions, achieve the best performance in each error measure. The hatched constant vertical lines in the plots of Figure 7.2 show the performance of the models when trained on the ground truth dataset $X[Q]$. Tables A.1-A.5 present the complete results.

Although there are some minor exceptions, generally, models trained on ground truth out-perform their counterparts trained on the series of $M_V^{A,p}$ datasets. As advertised, this suggests that models trained on the ground truth datasets are learning to sample



Figure 7.2: The plots summarize error terms, calculated with respect to $M_\alpha(X_D^m)$ and $M_\alpha(X[Q])$ for the CSDI and DDPM models. The models were trained against the datasets indicated on the x axis for five epochs. Shown are the results for data missing rates of 10% and 50%. The hatched cyan and orange constant lines mark the baseline performance of the CSDI and DDPM models trained on pseudo ground truth $X[Q]$.

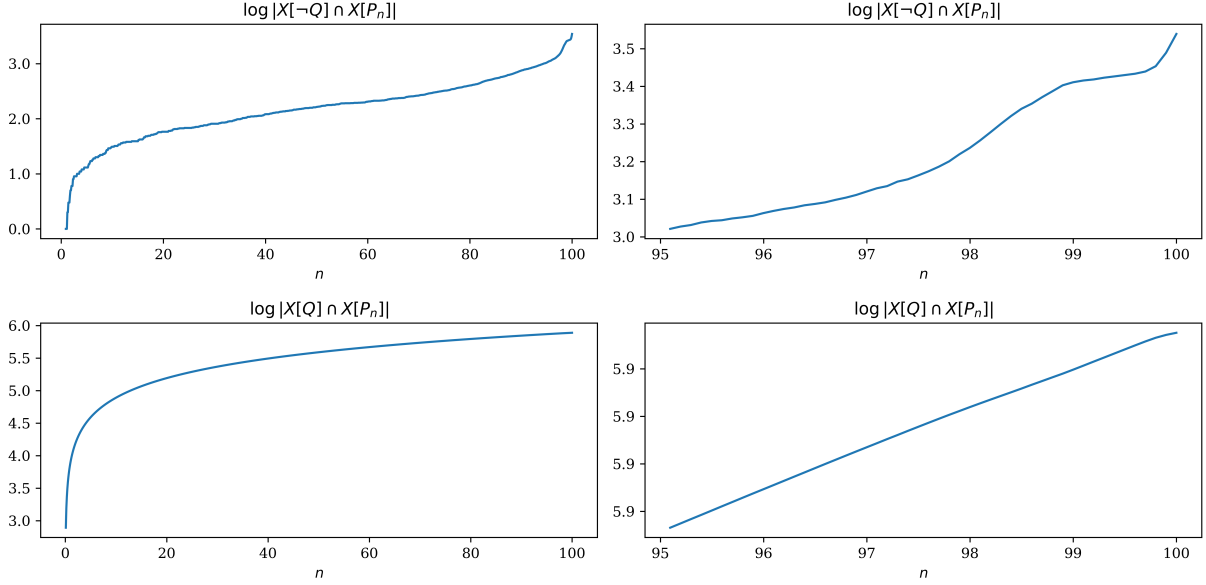


Figure 7.3: The plots on the first row show the growth in the cardinality $|X[\neg Q] \cap X[P_p]|$ generated on the range of percentile values $[0, 1]$ and high range $[0.95, 1]$. The second row shows the same plots for the growth in the cardinality of intersection $|X[Q] \cap X[P_p]|$ generated on like ranges.

$\sim p(X[Q])$. It also demonstrates that even low levels of noise and outlier density in the data materially affect model performance.

Tables A.1-A.5 and Figure 7.2 show that standalone CSDI performance improves as the missing data rate increases. The gap between CSDI and DDPM performance also increases, from negligible to substantial, with missing data rate. This suggests that the conditional mechanism in the CSDI model improves as it is exercised and that it meaningfully contributes to model performance.

A consistent feature of the plots in Figure 7.2 is a spike in error rates somewhere above the 96th Mean Squared Error (MSE) percentile. It is at this point that strongly anominal information enters the datasets. The spike may represent the impact of the naive approximation in Equation 7.4 taking hold as the conditions required for Equation 7.3 to hold are challenged.

Figure 7.3 provides insight into the quality of the data being added to the nominal population at the high percentile M_V^A range. As in the top row plots, it is at just above the

96th MSE percentile the qualitative rate of change in $|X[\neg Q] \cap X[P_n]|$ increases strongly. The response in model performance speaks directly to the type and quantity of information in the data being added to the intersection set. As per information theory, the more unique a datum, the more information it carries relative to the data. So as new intersecting set members become increasingly unique, the model error in Figure 7.2 increases dramatically [84].

Figure 7.2 shows that at thresholds less than the 96th percentile, and as missing data rates increase, CSDI models generally outperform DDPM models. But, Figure 7.2 also illustrates that CSDI results deteriorate more rapidly than those of DDPM as anomalous data is introduced into the training dataset. This suggests that the conditional mechanism lends the CSDI model a greater sensitivity to outliers. Ideally, the CSDI conditional loss would somehow rate anomalous information content as largely irrelevant. After all, an outlying time of flight has very little information to relate about the largely continuous nominal data in and around which it occurs. Moreover, with their high information content, outliers may generate some number of spurious correlations. But the conditional CSDI model struggles to make sense of the anomalous information. Conversely, when on average there is less information per element in the nominal partition, the CSDI model appears enabled to learn the possibly continuous, possibly well-defined structure of the data; this is implied by CSDI performance at high missing data rates below the 96th percentile. This suggests that in the case of a highly structured dataset, each datum bears a strong, identifiable, possibly causal conditional relation to neighboring data. CSDI leverages possible structure and, given a largely nominal signal in the dataset, readily learns the nominal distribution.

Figure 7.4 shows the results of a second set of experiments with the CSDI model. Here, the test dataset is enlarged while its anomalous content is fixed. A CSDI model is trained on each of the three datasets and, to ensure like-for-like comparison, error for each model is calculated only over the data in the smallest dataset. The experiments is designed to confirm that $\lim_{\frac{|X[P]|}{|X|} \rightarrow 1} P(\hat{X}|X) = P(\hat{X}|X[P])$ is true of the model, which is again to provide a confirmation that, depending on the quality of the dataset, the CSDI implements the Nominal Data Model (NDM) and Observed Data Model (ODM) of Equations 7.3 and 7.4 respectively.

The curves in Figure 7.4 show that, as dataset size increases, Mean Average Error (MAE) error improves across the entire series of partitioned datasets. The range of error also increases so that over and above the 96th percentile error ramps up more extremely as dataset size enlarges. This suggests that as training becomes saturated by nominal signal, the loss function responds by encoding information in the neural network weights that secure the greatest numerical benefit. This causes the error statistic to become increasingly

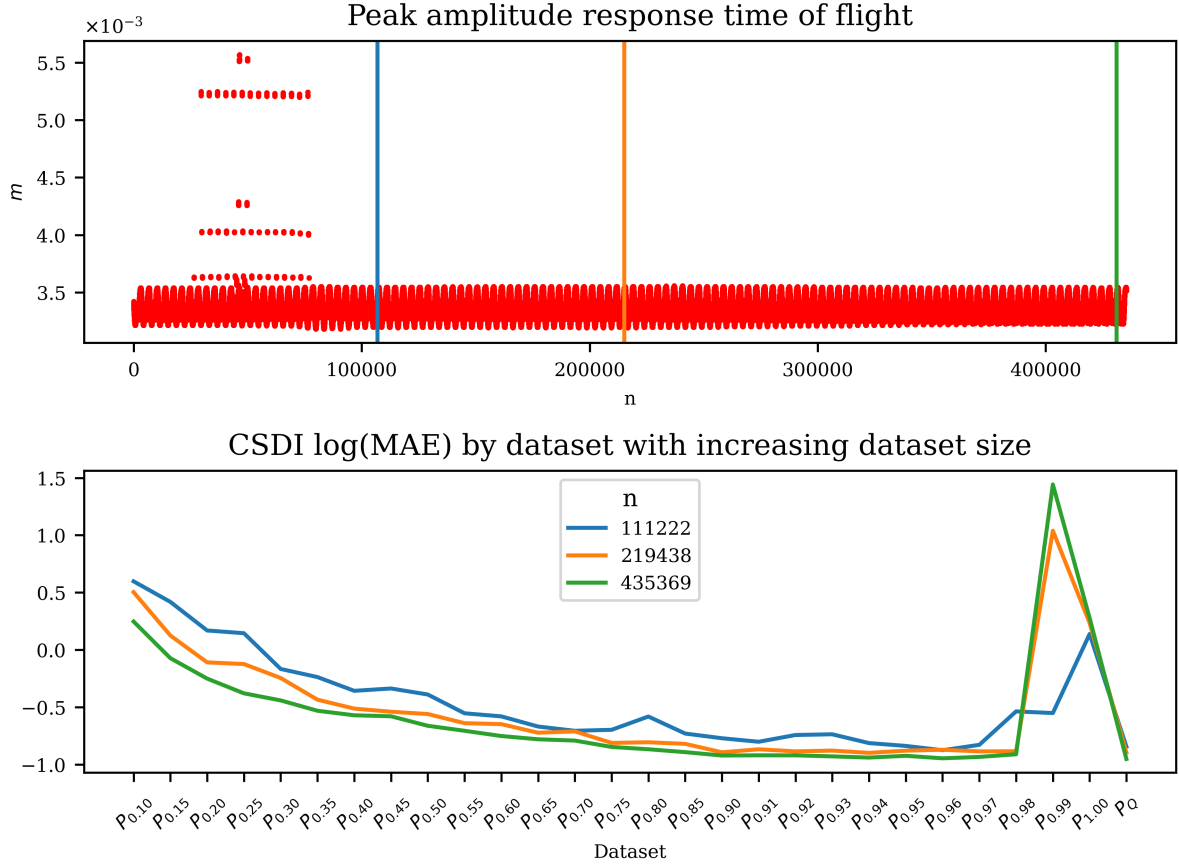


Figure 7.4: The top plot shows peak amplitude time of flight data, and three color-coded vertical bars mark three increasingly large datasets, with size n of 111222, 219438, and 435369 elements. The density of ground truth anominal data decreases as the dataset grows. This is because the vast majority of anominal information is in the leftmost region of the data. The second-row plot shows the color-coded MAE error of a CSDI model trained with 50% missing data ratio on the three datasets for five epochs. When the density of anominal information is high, CSDI appears to “split the difference”, so that while overall performance is adversely impacted, worst case performance is improved.

sensitive to anominal data.

7.5 Practical application and future work

Figure 7.5 gives the error against ground truth, for estimates of peak amplitude time of flight as provided by CSDI models trained on $X[P_{0.95}]$ and $X[P_{0.70}]$ partitioned datasets. Here error on $X[P_{0.95}]$ and $X[P_{0.70}]$ relative to pseudo ground truth is $\pm 6.6 \mu m$ and $\pm 8.15 \mu m$ respectively. This suggests that CSDI based estimates for nominal time of flight $\hat{X}$ could meet regulatory requirements for accuracy[150].

In field usage of DPM based estimates would require some means for fault diagnosis without reference to ground truth. There are numerous engineering solutions that could leverage agreement between multiple independent estimation techniques. Diffusion Partition Consensus (DiffPaC), developed in the next chapter, is one such solution. DiffPaC uses an array of estimates for nominal inner diameter time of flight, generated on the basis of multiple dynamically updated partitions, to generate self-supervised signal training signal.

A number of opportunities exist for improving results. Figure 7.6 shows $M_\alpha(X_D^m)$ and support from nominal classified data for CSDI trained on $X[P_{0.95}]$. What is clear is that the Savitzky-Golay Filter (SGF) fit on estimates overshoots the weight of the data in and around regions where the derivative of X with respect to its ordering variable is close to zero. This wig wag could, in part, be due to the Gaussian prior distribution used by CSDI models and could also, in part, be due to the nature of SGF curve fitting. A post-processing step that introduced weight to the SGF from observed data in and around the neighborhood of inflection points might serve to ameliorate the wig wag. A slightly more involved improvement might involve the use of alternative priors with more degrees of freedom than the Gaussian in the diffusion process [92].

One area of concern is the degree of support from nominal data on which estimates are based. Typically, inspection processes have some data quality argument attached to them that specifies the minimum level of local support from observed data on which an estimate must be based [150]. Figure 7.6 includes a shaded gray region where set membership in $X[P_{0.95}]$ is sparse. There, we see that the partition selected by the 95th MSE percentile discriminates against likely nominal observed data. Although the estimates are reasonable, it is desirable, from a process quality point of view, to enlarge the nominal set. An iterative training process with a human in the loop could include tools that identify regions of low support, allow for suitable data reclassification, and then retraining.

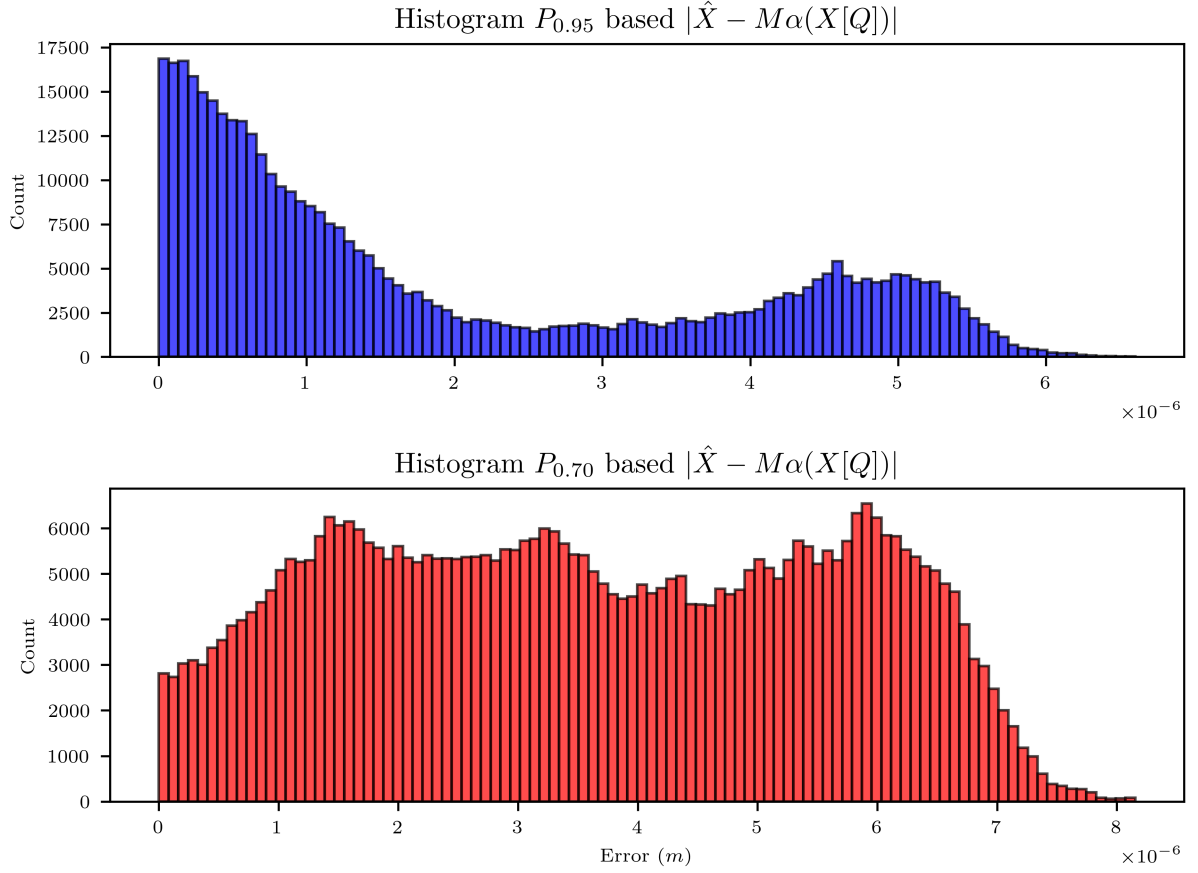


Figure 7.5: The practical distance-based error of a CSDI model trained on the $X[P_{0.95}]$ and $X[P_{0.70}]$ partitioned datasets with $n = 435369$, as shown in Figure 7.4, for 5 epochs. The $X[P_{0.95}]$ and $X[P_{0.70}]$ trained models have MXAE, relative to ground truth, of $6.60 \mu m$ and $8.15 \mu m$ respectively.

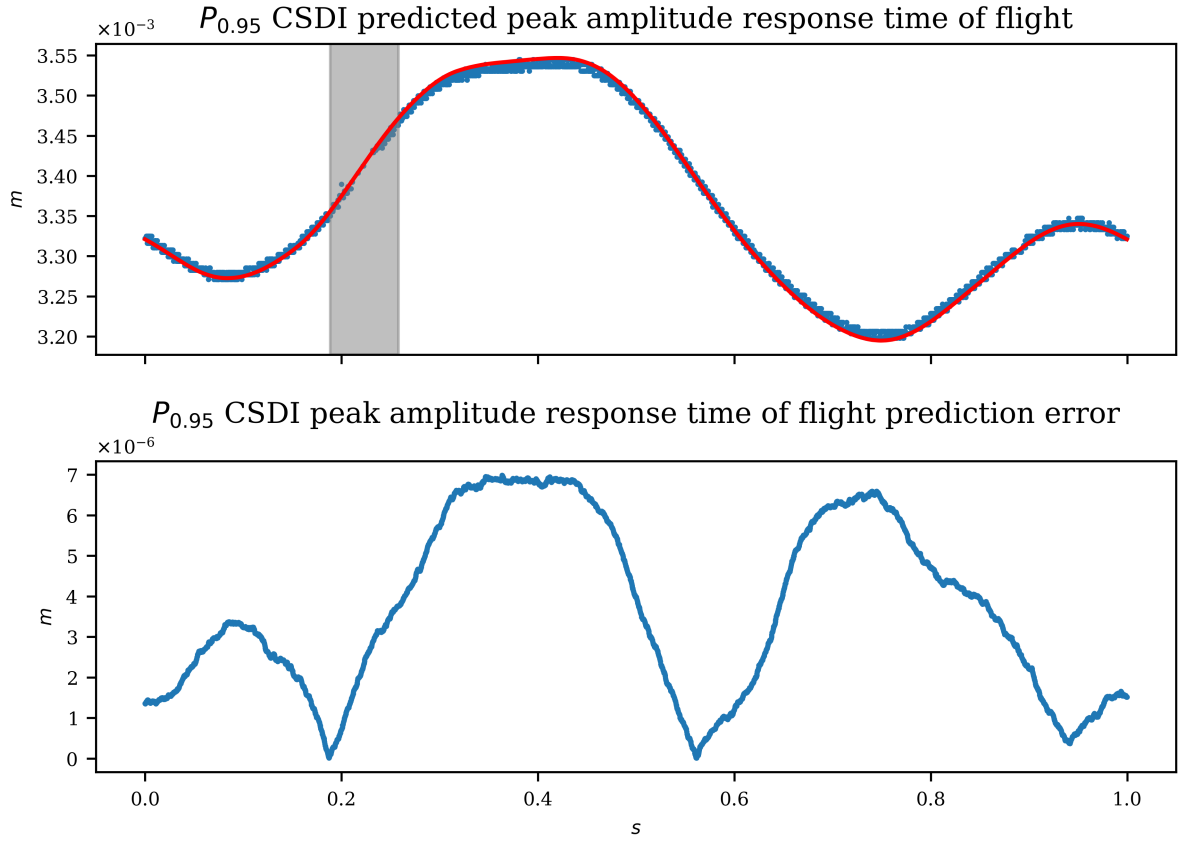


Figure 7.6: The plot shows detail of the CSDI prediction fit and nominal data support from $X_{0.95}^n$. Estimates $M_\alpha(X_D^m)$ overshoot at predictable locations where error increases.

7.6 Conclusions

The **UT** data analysis task is modelled probabilistically, and the potential to provide decision support and automation using **CSDI** and **DDPM** based implementations of the model is evaluated. The veracity of the **NDM** and **ODM** models is indirectly demonstrated and confirmed by way of their ability to explain **CSDI** and **DDPM** model performance across a variety of uniquely partitioned **UT NDE** datasets. The **NDM** and **ODM** provide a principled basis for understanding the behavior of **CSDI** and **DDPM** models. And may inform a technical justification supporting the use of such models in safety-critical contexts.

Trained **CSDI** models are sampled to obtain estimates of nominal peak amplitude response time of flight. Sample estimates were found to be accurate to within bands of $\pm 6.60 \mu m$ and $\pm 8.15 \mu m$, which supports a requirement for accuracy of depth estimation [150]. The unsupervised training procedure does not rely on dataset labelling or annotation, and the accuracy of sampled estimates depends only on learning the distribution of a single dataset. In this way, the approach may be used offline on a per-inspection basis, with no data annotation and without recourse to out-of-distribution generalization. Various means to improve results and confidence therein are considered, and the inclusion of a human in the loop of an iterative training and verification process is also contemplated, in lieu of the availability of ground truth, as a means to support fault detection and remediation.

The method, though not directly comparable, qualitatively improves upon prior work based on discriminative methods where results rely on data annotation, pre-processing, brittle heuristics, and out of distribution generalization. The probabilistic model and corroborating results, imply a principled basis underlying model behaviors and provide a means to interface with regulatory bodies seeking some justification for usage of novel methods in safety-critical contexts.

Chapter 8

Diffusion Partition Consensus based analysis

In Chapter 7, an evaluation of [Conditional Score Based Diffusion \(CSDI\)](#) and [Denoising Diffusion Probabilistic Model \(DDPM\)](#) model performance illustrated the deleterious effect of anominal information in a training dataset. Having identified that outliers negatively impacted model performance, attention is paid to developing methods to ameliorate their impact.

[Diffusion Partition Consensus \(DiffPaC\)](#) is a method that ably identifies and removes outliers from training datasets during training. The novel technique relies on an iterative approach that uses training to push forward towards a learned approximate data distribution, while a greedy technique operating on biased partitions of the dataset, prunes and reconstitutes elements of the training data. Active selection in the training dataset is applied based on criteria derived from data-driven summary statistics. This shapes the distribution learned by a diffusion model and induces a self-supervised training signal. This chapter describing [DiffPaC](#) closely follows material first published in [\[133\]](#).

[DiffPaC](#) is a novel generative technique for time series anomaly detection and data imputation in the presence of outliers. To illustrate the method, an implementation with design choices tailored for well-structured time series typical of [Canada Deuterium Uranium \(CANDU\) Ultrasound \(UT\) Non Destructive Evaluation \(NDE\)](#) datasets is proposed. The technique relies on cross-talk between a [CSDI](#) model and two well-chosen [Savitzky-Golay Filters \(SGFs\)](#). Testing and evaluation are performed on a series of progressively information-rich synthetic datasets, and two real-world [UT NDE](#) datasets taken from a [CANDU](#) nuclear reactor [Pressure Tube \(PT\)](#) and [Calibration Fixture \(CF\)](#). The iterative

technique is a blend of stochastic and deterministic methods that develops confidence and consensus in target parameter estimates using several dynamically updated binary partitions over the dataset. This, in turn, allows a new set of estimates and confidence measures to be established. Data classification induces a progressive bias in the training datasets and allows a diffusion model to identify the prevalent distribution. Methods for fault diagnosis support the efficacious inclusion of a human in the loop, making the technique suitable for safety-critical applications. Its main advantages are that it is an unsupervised technique with principled performance and strong explainability. It does not require dataset curation or significant data pre-processing, does not rely on out-of-distribution generalization, provides means for fault diagnosis without recourse to ground truth, converges with stability, provides results that are insensitive to over-fitting, and naturally includes a human in the loop for fail-safe verification of results. The quality of results, the checks and balances provided by the fault diagnosis mechanism, and the opportunity to include a human in the loop support the case for usage in safety-critical contexts such as [UT NDE](#) data analysis.

By and large, [UT](#) data analysis depends on the ability of analysts to accurately estimate nominal trend in acquired data [\[79\]\[80\]\[132\]](#). As is true for many materials under test in [NDE](#) contexts, [PT Inner Diameter \(ID\)](#) surfaces are generally smooth and well-defined. Because strong departures from nominal are the exception rather than the rule, datasets with strongly unbalanced multimodal data distributions are typical. [DiffPaC](#) exploits unbalanced multimodal distributions using the consensus of various estimates from a parameterized data-smoothing technique and generative samples to expose data that contradicts an assumption of unimodality. Methods are integrated to develop [DiffPaC](#), a novel unsupervised process that classifies [UT](#) data as nominal or anominal, and acquires nominal peak amplitude times of flight, across the dataset.

[DiffPaC](#) integrates a [CSDI](#) [\[128\]](#) model and a pair of [SGFs](#) [\[116\]](#) to generate meaningful semi-supervised training signal. The signal allows the dynamic selection of a data classifying binary partition over the dataset. The partition is updated on a per-epoch basis, using evidence from the models, to learn a progressively more accurate data classification, which necessarily improves ensuing estimates and samples of nominal trend. Progressive learning is facilitated by the susceptibility of [CSDI](#) to fine-tuning, which allows it to learn and unlearn information as training data is manipulated [\[81\]](#). Epoch-wise progression of the algorithm is supported by the conditional inference mechanism embedded in [CSDI](#) and stochastic data partitioning.

[DiffPaC](#) is suitable for use in safety-critical contexts because it trains on each dataset independently and so does not rely on out of distribution generalization. This makes it robust to non-stationary ambient and physical plant condition. [DiffPaC](#) uses the summary statistics of [SGF](#) fits of biased data partitions as means for fault diagnosis. Which, in

conjunction with the epoch-wise training regime, facilitates the insertion of a human in the loop for remediation and verification of results. This provides a fail-safe back-stop against model error and strengthens the case for use in safety-critical industrial contexts, contexts that might benefit from the efficiency of machine learning but cannot bear the risk or consequence of automation error. Thus, given a dataset A of 20 *MHz Normal Beam (NB) ascans* taken over the *ID* surface of a *CANDU PT* with flaws *DiffPaC*:

- Finds a principled basis of operation in the mathematical formulation of *CSDI* [128],
- Supports explainability by reducing the estimation of nominal target parameters to an easily understood binary classification problem,
- Provides real value continuous estimates for nominal peak amplitude times of flight in A and classifies observed times of flight as nominal or anomalous. Estimates are accurate on the tested calibration facility dataset to within $\pm 6.67 \mu m$. Well within the accepted band of accuracy ($\pm 20 \mu m$) for this measurement process [134][135],
- Learns the approximate distribution of each dataset, maximizing model robustness to ambient system and operational conditions to relax dependency on out-of-distribution generalization,
- Has no requirement to label, maintain, or curate a reference training dataset of any kind,
- Provides fault diagnosis and supports the inclusion of a human in the loop, for active learning, to remediate edge cases and provide fail-safe analysis in safety-critical applications.¹

To our knowledge, *DiffPaC* is a first of a kind, and novel, usage of a diffusion model in a *UT NDE* or Structural Health Monitoring (SHM) context and establishes state of the art. *DiffPaC* is also the first unsupervised machine learning method to simultaneously provide anomaly detection, anomaly localization, and real value target parameter estimates for *UT NDE* data. *DiffPaC* is readily customized for other applications requiring custom *ascan* peak or feature following in the presence of outliers.

The ability to estimate nominal peak amplitude time of flight is significant as it constitutes the necessary initial heavy lifting of *UT NDE* data analysis. Downstream tasks, typical of many inspection processes, such as flaw sizing (estimating depth, length, and

¹The results reported in this research were achieved without the use of a human in the loop or active learning.

width) are well-supported by DiffPaC outputs, which may be routed to existing foundation models, or other methods for object detection and segmentation [74][89][107].

8.1 Learning nominal criteria

DiffPaC dynamically learns the definition of nominal in a dataset. Manual analysis is based on the amplitudes and times of flight of features in a set of *ascans*. The first step in manual analysis is to locate *ascans* whose peak amplitudes deviate from nominal. Excursions and deviations in peak amplitude are evidence of atypical reflectors and often indicate that an *ascan* contains information about a flaw or anomaly. Analysis consists of locating atypical peaks within these *ascans* and assessing their times of flight. Finally, reference nominal peak amplitude times of flight are established. The reference nominal values are those that would have occurred if the surface under test had been nominal. In nominal *ascans* the peak amplitude response is associated with reflection from the PT inner diameter pipe wall [134][135].

DiffPaC estimates nominal peak response time of flight by learning performed on dimensionally reduced datasets that have shed amplitude-related information. In this context, the meaning of nominal and anominal data classifications differs from the broader analysis task. The DiffPaC specific task is to learn the nominal and assumed prevalent distribution within the dataset, where the definition of nominal and anominal is data-driven. The process begins by assuming the entire dataset is nominal and then culling data that most strongly challenge the assumption. The SGF based culling criteria rely on stochastic behaviors, and the summary statistics of fits and residuals on biased partitions of nominally classified and sampled data. Summary statistics evolve as samples and data classifications improve, and the definition of nominal and anominal is learned progressively as this occurs. Figure 8.1 illustrates the progression from initialization to convergence and, Figure 8.2 provides a visualization of SGF estimators M_α and M_β acting on data biasing partitions P_D , I , and F .

8.2 Analysis and the manifold

Given the strength of the manifold hypothesis [10][82] we assume the existence of two sets of disentangled, or identifiable, data generating factors underlying CANDU PT UT NDE data distributions. Let $\mathcal{F}_n$ be the distribution of nominal data generating factors and $\mathcal{F}_a$ the distribution of anominal data-generating factors. Given dataset X and a Boolean

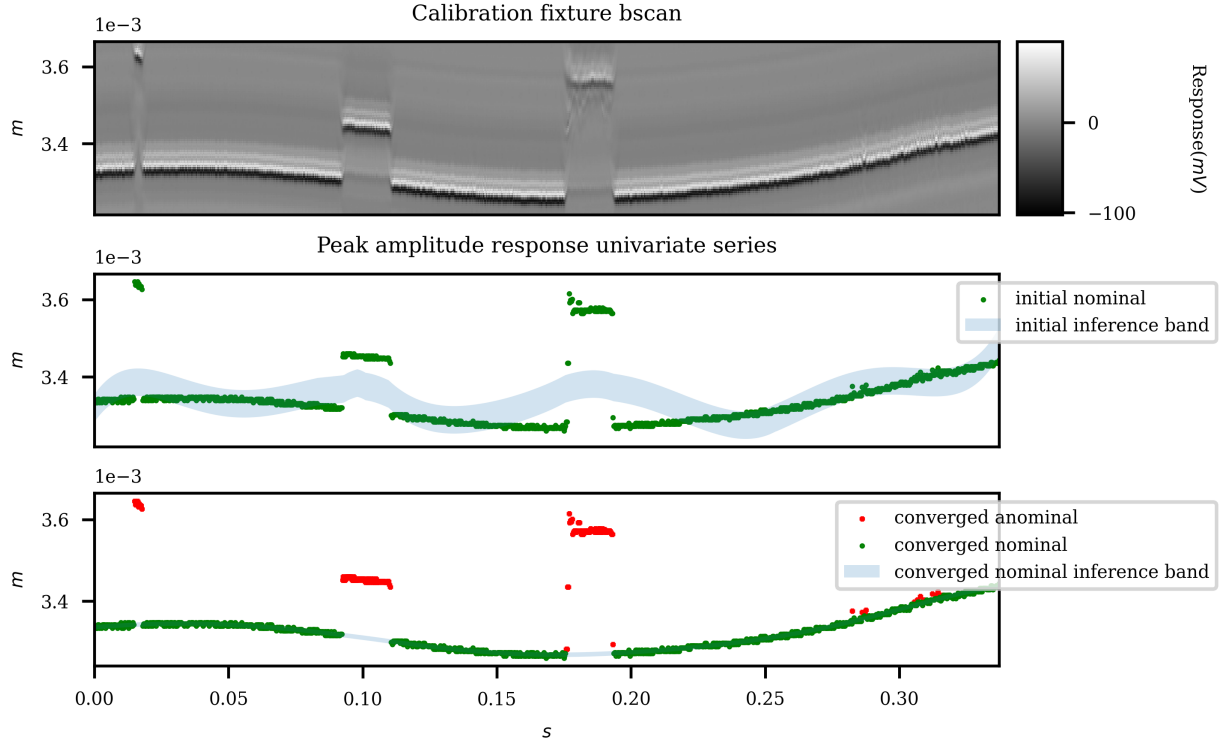


Figure 8.1: Depicted is a *bscan*(top) of CF features T, U , and V (see Figure 5.1) with derived univariate time series (middle, bottom). The middle plot shows DiffPaC initialization with all data assumed nominal and an initial nominal inference band biased by outliers. Bottom plot show convergence with data classification. An improved nominal inference band develops based on learned summary statistics over fits and residuals of the nominal classified data.

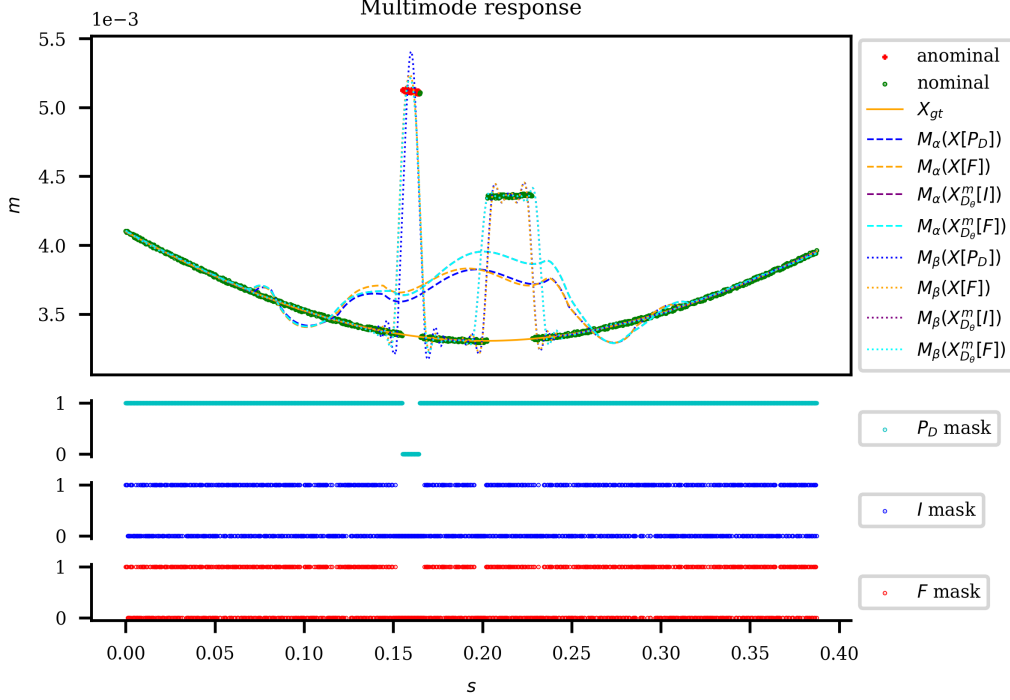


Figure 8.2: Fits of models M_α and M_β to P_D , I , and F partitioned subsets of synthetic dataset X , and samples X_D^m . The three lower plots give the contents of partitions P_D , I , and F .

partition vector P_D , the stage is set for a probabilistic binary classification problem: find P_D such that $X[P_D] \sim \mathcal{F}_n$ and $X[\neg P_D] \sim \mathcal{F}_a$.

Given the existence of a pseudo ground truth binary partition Q of X we can generate $X_n = M_\alpha(X[Q])$, the pseudo ground truth of X . We have access to true ground truth X_{gt} for the synthetic datasets, and pseudo ground truth X_n for the calibration facility dataset; these are used in Section 8.6 to gauge the performance of DiffPaC and its fault diagnosis mechanism.

DiffPaC provides a method to distill the data generating factors of X and learn partition vector P_D such that: $X[P_D] \approx p(X[Q])$, $X = X[P_D] \cup X[\neg P_D]$, $X[P_D] = \{x|x \in X \wedge x \approx \mathcal{F}_n\}$ and $X[\neg P_D] = \{x|x \in X \wedge x \approx \mathcal{F}_a\}$. Additionally, DiffPaC imputes missing data $\approx \mathcal{F}_n$ at locations where observed data is $\sim \mathcal{F}_a$.

Algorithm 2 DiffPaC: Terminates on criteria function $\tau(\delta)$ dependent on basis for criteria δ . Yields $P_D \approx p(Q)$ and $\hat{X} \approx M_\alpha(X[Q])$.

Input: M_α, M_β, D, X

$P_D \leftarrow \{True | \forall x \in X\}$

while $\tau(\delta)$ **do**

 # Learn approximate data distribution

$D, X_D, P_D, I, F \leftarrow \text{trainOneEpoch}(D, X, P_D)$

 # D stochastic nominal cull

$S_c \leftarrow X_D^s > X_D^s[I].m()$

$P_D[S_c], I[S_c], F[S_c] \leftarrow False, False, False$

 # $M_{\alpha\beta}$ geometric nominal cull

$M_c \leftarrow M_{\alpha\beta}^s > M_{\alpha\beta}^{3s}$

$P_D[M_c], I[M_c], F[M_c] \leftarrow False, False, False$

 # d_{X_D}, d_X geometric nominal restoration

$R_{d_{X_D}} \leftarrow d_{X_D} < d_{X_D}.m() + 3 \cdot d_{X_D}.s()$

$R_{d_X} \leftarrow d_X < d_X.m() + 3 \cdot d_X.s()$

$P_D[R_{d_{X_D}} \vee R_{d_X}] \leftarrow True$

$P_D \leftarrow \text{manualRemediation}(M_{\alpha\beta}, X, P_D)$

▷ optional

end while

$\hat{X} \leftarrow M_\alpha(X_D^m[I])$ **return** $P_D, \hat{X}$

8.3 Algorithm

Q is unknown, and an unsupervised system requires some data derived signal to facilitate learning. We generate signal by constructing an iterative process on a Bayesian prior posterior chain used to drive a binary classification:

$$p(X[\hat{P}_D]) = p(X[\hat{P}_D]|X[P_D])p(X[P_D]) \quad (8.1)$$

Direct implementation of Equation 8.1 is problematic as the right-hand side terms are intractable. Instead, [DiffPaC](#) implements the approximation given in Algorithm 2.

On each iteration of the *while* loop in Algorithm 2 D trains on $X[P_D]$ to learn an approximate data distribution; this stands as proxy for prior $p(X[P_D])$ in Equation 8.1. Likelihood $p(X[\hat{P}_D]|X[P_D])$ and its interaction with the prior is approximated by a three-phase process that includes a stochastic cull based on the confidence D expresses in samples X_D , a cull using $M_{\alpha\beta}$ to test the assumption that $P_D = Q$, and reconstruction based on the assumption that the two culling operations have indeed moved $p(X[P_D])$ towards $p(X[Q])$. This produces an aesthetically pleasing, fully learned algorithmic flow wherein decision criteria for the classification of $x \in X$ are:

1. Dependent on underlying stochastic behaviors,
2. Understood as identifying data that most strongly challenges or supports the assumption that $P_D = Q$,
3. Based on standard deviations of fits and fit residuals and may be expressed in explainable terms and units natural to the dataset,
4. Learned from the data.

Learned updates to P_D consistently bias the training dataset $X[P_D]$ towards $\sim \mathcal{F}_n$. This engages fine-tuning to bias the distribution learned by D on subsequent epochs towards $\sim \mathcal{F}_n$. Updates to P_D are generated based on discordant [SGF](#) sensitivity to multimodal data distributions. Discordant response is stimulated by intentionally biased partitions of X and X_D^m (as described in Section 8.5). This provides a series of datasets with initially varying unbalanced multimodal distributions that eventually converge to a single mode.

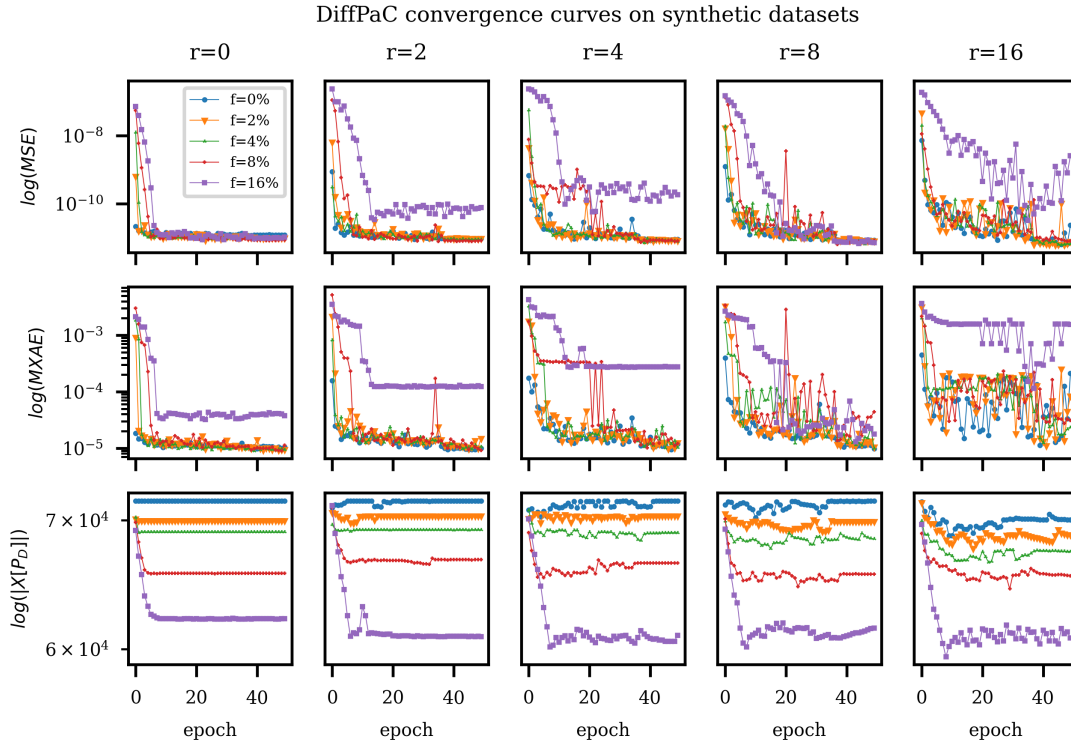


Figure 8.3: DiffPaC convergence over 50 epochs of training on the synthetic datasets described in Chapter 5. These curves report the training runs whose results are provided in Table 8.1.

8.4 Termination criteria and fault diagnosis

Termination criteria $\tau(\delta)$ may be time based, as in the number of epochs trained or based on the convergence of some metric. Generally speaking, as **DiffPaC** training progresses epoch by epoch the following traits exhibit:

1. Cardinality $|X[P_D]|$ converges,
2. The distribution of $M_{\alpha\beta}^s$ tightens about its mean.

Trait 1 is visualized in Figure 8.3. There we see that as error measures initially decrease and stabilize $|X[P_D]|$ tends to stabilize and drift towards or around some value. We explain these results as **DiffPaC** oscillating about a point or region on the manifold from which it cannot escape. The point may be a location where $X[P_D] \approx p(X[Q])$ and results are good, or it may be that some data miss-classification error has occurred which requires manual remediation. In some sense **DiffPaC** is a greedy algorithm and some data configurations may cause it to paint itself into a corner it cannot step out of.

Trait 2 is used for fault diagnosis as exceptions to convergence about the mean indicate likely misclassified data. The lower plots of Figures 8.4, 8.5, and 8.6 show this behaviour. The bottom plot in Figure 8.5 shows an $M_{\alpha\beta}^s$ converged tightly around its mean with no strong exception to the group behavior. On the other hand, Figure 8.4 shows a plot with strong exceptions, or errors, relative to tight convergence; these align with L_1 error against ground truth. As shown in the plots $M_{\alpha\beta}^{sm}$ and $M_{\alpha\beta}^{3s}$ may be used as a means to characterize confidence in results. For high-confidence results, the bulk of the diagnostic should reside evenly below $M_{\alpha\beta}^{3s}$.

As in all machine learning, faults invariably arise due to edge cases and pathologies in the data. With **DiffPaC**, faults are directly attributable to an underlying data misclassification. $M_{\alpha\beta}^s$ behavior provides a means for fault diagnosis that allows a human in the loop to quickly identify and remediate data misclassifications. This is easily accomplished using some graphical user interface that allows manual interaction with the data, and is a step most would regard as necessary in a safety-critical verification and acceptance process.

The $M_{\alpha\beta}^s$ diagnostic is weak in the sense that it does not rank or order associated errors in terms of their size. This is demonstrated in Figure 8.8, where the ordering of L_1 error magnitude vs. ground truth does not match the ordering of diagnostic magnitudes. This is a weakness of parameterized models which overfit or ring unanticipatedly; **DiffPaC** exploits ringing but does not solve this weakness. The reliance on parameterization also leads to modest errors in alignment between L_1 error and $M_{\alpha\beta}$ peaks so that ground truth errors and their indications are slightly askew on the time axis.

8.5 Partition based data biasing

[83] proves that the unsupervised learning of disentangled factors cannot occur without bias. The directed manipulation of dataset content to provide bias to training is critical to DiffPaC. SGF models are fit on partitions of data and samples that expose and stimulate divergent sensitivity to multimodal data distributions. This allows D to train on a progressively improved $X[P_D]$. Data of various types are useful in this regard: (i) all observed data classified as nominal, (ii) observed data classified as nominal that have associated high confidence diffusion samples, (iii) high confidence diffusion samples of observed nominal classified data, (iv) high confidence diffusion samples of observed anominal classified data. DiffPaC partitions data according to these subtypes to generate intentionally biased datasets, each with a unique multimodal distribution.

The CSDI model tracks unobserved, or missing, data in a dataset using a mask provided to it at initialization. DiffPaC uses this mask as the binary data classifying partition P_D , and dynamically classifies data by updating this mask between epochs during training. To this end, the CSDI observed data mask becomes the DiffPaC binary classifying partition P_D and $X[P_D]$ indexes nominally classified data made available to train on.

CSDI implements a ground truth mask G to hold back a randomly selected fraction of the observed data as labels for training its conditional inference mechanism. CSDI training proceeds such that samples of the marked ground truth data exhibit high variance or low confidence. DiffPaC discriminates against these and uses high-confidence samples exclusively as it performs partition updates. $I = \neg P_D \vee (P_D \wedge \neg G)$ is constructed so that $X_D^m[I]$ yields the means of high-confidence samples of both nominal and anominal classified data. Because CSDI randomly selects G from observed data P_D each epoch, each G dependent partition is to some degree stochastic.

High confidence CSDI samples for anominal data are relatively inaccurate when compared to samples for nominal data. This divergence is exposed by using $F = I \wedge P_D$ so that $X_D^m[F]$ selects high confidence samples of nominally classified data only and $X[F]$ selects nominal classified data that has associated high confidence sample means.

As $\sim p(P_D)$ approaches $\sim p(Q)$, high confidence diffusion samples become progressively $\sim p(X[P_D])$. At the same time, each partitioned dataset listed above also tends to $\sim p(X[Q])$. But, when $\sim p(P_D)$ is dissimilar to $\sim p(Q)$ the datasets carry variously multimodel mixtures. The divergent sensitivity of the SGF models to the various data mixtures supplies training signal to update P_D .

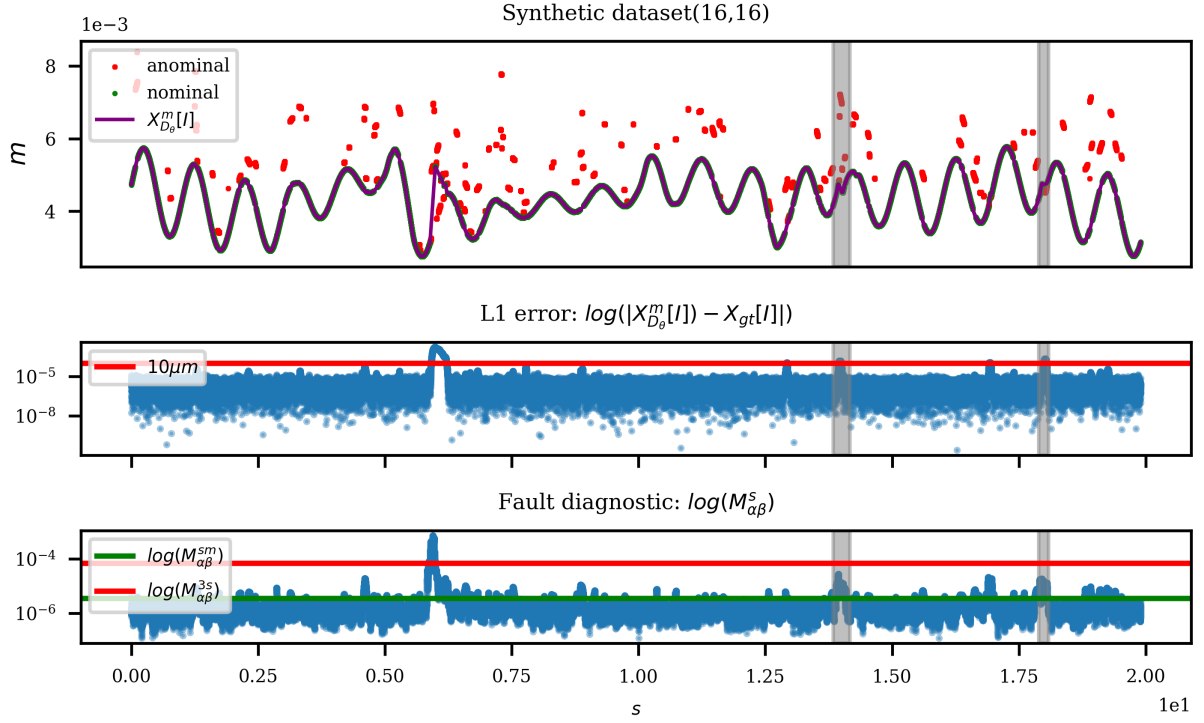


Figure 8.4: Synthetic dataset (16, 16) is the most challenging of those tested. Of note: $M_{\alpha\beta}^s$ excursions match the L_1 ground truth error and $M_{\alpha\beta}^s$ does not lie evenly between $M_{\alpha\beta}^m$ and $M_{\alpha\beta}^{3s}$; indicating the presence of data classification errors. Detail of the first and second grey-filled regions is found in Figure 8.8.

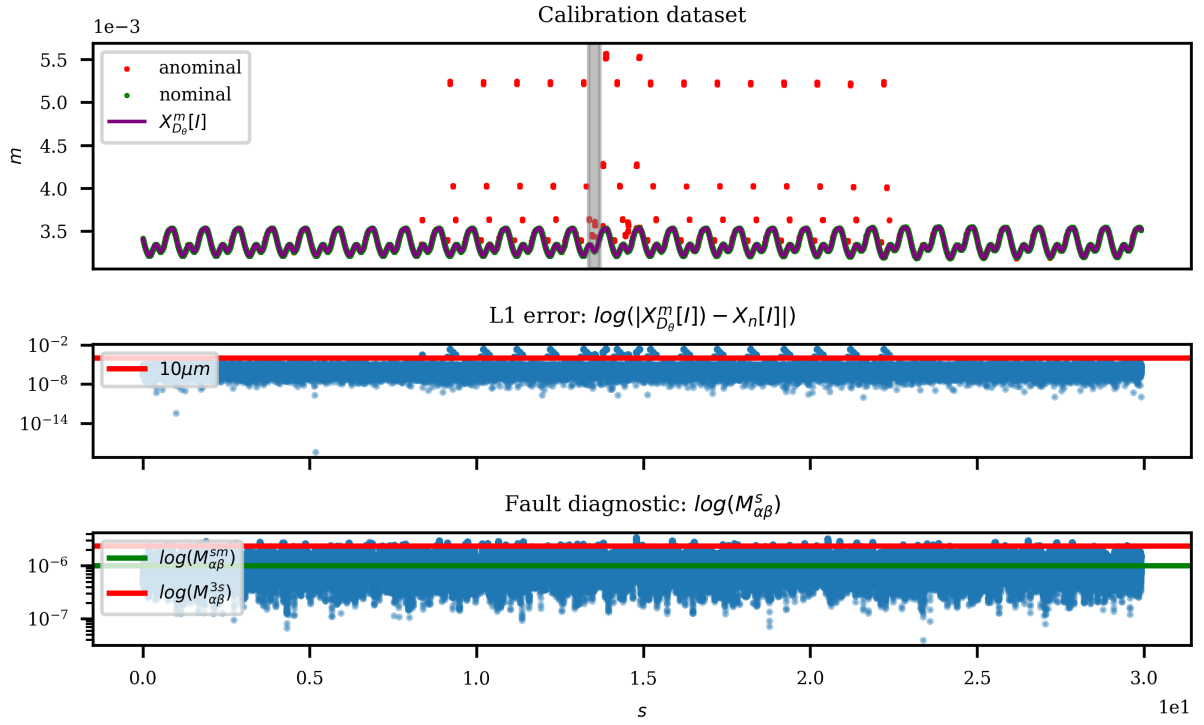


Figure 8.5: $M_{\alpha\beta}^s$ diagnostic for the calibration dataset has its firm majority lying evenly distributed beneath $M_{\alpha\beta}^{3s}$ which strongly increases confidence in results. Detail of the greyed region is found in Figure 8.1.

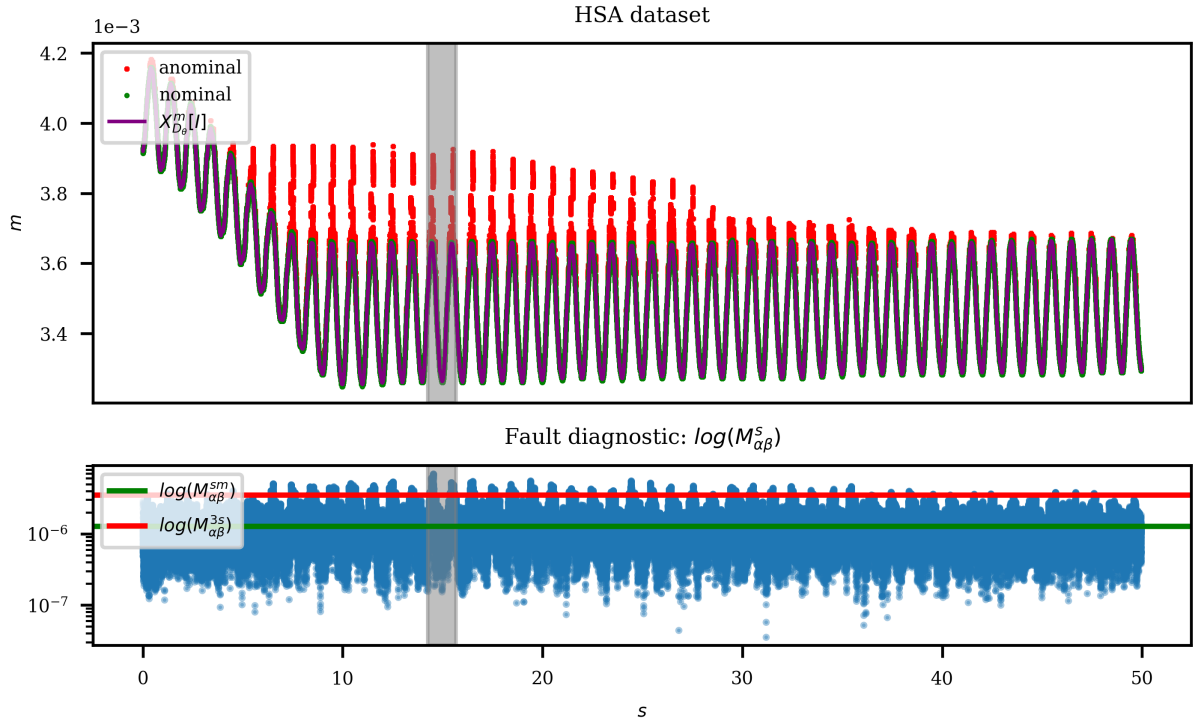


Figure 8.6: Sediment, fretting, and strong geometric regime change increase the complexity of the HSA dataset. $M_{\alpha\beta}^s$ lies somewhat unevenly in and around $M_{\alpha\beta}^{3s}$. This decreases confidence in results. However, as in Figure 8.7, which provides detail of the gray shaded region, estimates $X_D^m[I]$ are within a band of acceptable error that encompasses any reasonable set of estimates manual analysis might attribute to this data.

Table 8.1: Standard error measures, and confusion matrix components, for DiffPaC performance trained on the synthetic and CF datasets. DiffPaC was trained for 50 epochs on each dataset with no active learning. Each synthetic dataset is identified by its (r, f) tuple and plotted in Figure 5.2. Shaded gray rows highlight results with MXAE less than $\pm 20 \mu m$. The synthetic and CF datasets have sizes 71643 and 107715, respectively.

Dataset	Std. dev.(m)	True Pos.	False Pos.	False Neg.	True Neg.	MSE(m)	MXAE(m)
(0,0)	$9.41E-05$	71643	0	0	0	$1.20E-11$	$1.05E-05$
(0,2)	$2.51E-04$	69971	0	0	1672	$9.80E-12$	$9.43E-06$
(0,4)	$2.72E-04$	69037	0	0	2606	$9.55E-12$	$9.41E-06$
(0,8)	$4.02E-04$	65684	0	0	5959	$8.53E-12$	$1.12E-05$
(0,16)	$5.02E-04$	61946	289	49	9359	$1.04E-11$	$3.75E-05$
(2,0)	$3.30E-04$	71643	0	0	0	$9.52E-12$	$9.94E-06$
(2,2)	$3.64E-04$	70352	24	0	1267	$9.73E-12$	$1.49E-05$
(2,4)	$4.11E-04$	69214	0	0	2429	$9.03E-12$	$1.07E-05$
(2,8)	$5.02E-04$	66772	0	6	4865	$8.34E-12$	$9.19E-06$
(2,16)	$5.98E-04$	60540	374	396	10333	$7.67E-11$	$1.24E-04$
(4,0)	$4.51E-04$	71643	0	0	0	$9.18E-12$	$1.19E-05$
(4,2)	$4.72E-04$	70350	0	0	1293	$8.45E-12$	$1.25E-05$
(4,4)	$5.12E-04$	68984	0	264	2395	$7.97E-12$	$1.03E-05$
(4,8)	$5.84E-04$	66505	8	0	5130	$8.02E-12$	$1.20E-05$
(4,16)	$6.84E-04$	60728	292	295	10328	$1.85E-10$	$2.75E-04$
(8,0)	$5.34E-04$	71613	0	30	0	$8.52E-12$	$9.82E-06$
(8,2)	$5.68E-04$	69896	0	17	1730	$8.16E-12$	$1.04E-05$
(8,4)	$6.19E-04$	68464	0	331	2848	$7.87E-12$	$9.98E-06$
(8,8)	$6.92E-04$	65639	0	35	5969	$8.32E-12$	$4.39E-05$
(8,16)	$7.37E-04$	61514	24	77	10028	$7.24E-12$	$1.76E-05$
(16,0)	$6.90E-04$	70050	0	1593	0	$2.39E-11$	$2.11E-04$
(16,2)	$7.35E-04$	68788	0	1407	1448	$7.80E-12$	$8.39E-05$
(16,4)	$7.47E-04$	67443	0	1252	2948	$6.63E-12$	$2.36E-05$
(16,8)	$7.86E-04$	65177	321	1394	4751	$8.07E-12$	$3.08E-05$
(16,16)	$8.40E-04$	60395	427	816	10005	$2.58E-09$	$1.56E-03$
CF	$1.41E-04$	106312	0	209	1194	$2.32E-12$	$6.67E-06$
HSA	$1.92E-04$	N/A	N/A	N/A	N/A	N/A	N/A

8.6 Results

Figures 8.4, 8.5, and 8.6 show the results of running DiffPaC on synthetic dataset (16, 16), the CF dataset, and the High Stress Area (HSA) dataset respectively. Each figure shows the data, its classification, estimates $X_D^m[I]$ and diagnostic $M_{\alpha\beta}^s$. When ground truth X_{gt} is available L_1 error $|X_D^m[I] - X_{gt}[I]|$ is also provided. Table 8.1 tabulates DiffPaC performance on the various datasets. Error measures such as Mean Squared Error (MSE) and Maximum Absolute Error (MXAE) relative to ground truth and various partition cardinalities are reported, along with relevant dataset characteristics. The cardinalities may be used to generate confusion matrices for tested datasets with ground truth.

Of particular interest are the [MXAE](#) figures, which are, in some cases relatively large so that even with seemingly excellent [MSE](#) results, large estimate error occurs. This is shown in Figure 8.4, where, though performing well overall, a handful of data classification errors lead to estimates with errors over $10\text{ }\mu\text{m}$ relative to ground truth.

[MXAE](#) performance was especially poor on synthetic dataset (16, 16). With a standard deviation of $8.40e - 4$, 16 regime changes, and an approximately 16% anomalous data rate, this dataset challenges the initial assumption of a strongly imbalanced multimodal distribution with only irregular variance from a prevalent nominal distribution. Contrast with the real world calibration dataset having a standard deviation of $1.41e - 4$, no discernible regime change, and a flaw rate of 0.98%, and the [HSA](#) dataset with standard deviation of $1.92e - 4$ and only one observed regime change - and the exceptional nature of the synthetic data should be clear. Even given the challenge of this data, the $M_{\alpha\beta}^s$ based diagnostic provides the appropriate information to facilitate a human in the loop towards manual error correction that would quickly restore results.

The calibration dataset results tell a good story about performance in real-world conditions. With a maximum absolute error of $6.67\text{ }\mu\text{m}$, the calibration results are within the accepted accuracy band of $\pm 20\text{ }\mu\text{m}$ and are arguably usable for any practical intent and purpose. This figure compares favorably with results in Section 6.6 of $9.23\text{ }\mu\text{m}$ using a [Variational Autoencoder \(VAE\)](#), and results in Section 7.4 of $6.6\text{ }\mu\text{m}$ using [CSDI](#).

In terms of identifying nominal data. The calibration facility confusion matrix entries in Table 8.1 show that [DiffPaC](#) produced no false positives and 209 false negatives. This demonstrates a conservative bias in data classification that can be expected as the learned nominal criteria consistently weed out the tails of the classified nominal data distribution. This conservative bias is also expressed in the synthetic data results where false positives tend to occur in the datasets with a high flaw rate, but false negatives generally occur at a much higher rate.

The [HSA](#) dataset is from a region at the end of the [PT](#) where the tube is pressed in a rolled joint to form a pressure boundary with interfacing components at the reactor face. The high variance in both the data and estimates comes as a result of densely occurring fuel-bundle bearing pad frets, oxidation, and sediment. This alignment of circumstance challenges manual analysis, which requires some degree of subjective decision-making to assess the data fully. As shown in Figure 8.7, [DiffPaC](#) finds the data's relevant statistical weight and structure with systematic and repeatable objectivity. And estimates provided are, by inspection, within $\pm 20\text{ }\mu\text{m}$ of any reasonable interpretation of the data.

The convergence curves in the bottom row of Figure 8.3 plot $\log(|X[P_V]|)$ over the course of 50 epochs of training. The curves show relative stability in the cardinality $|X[P_V]|$, which

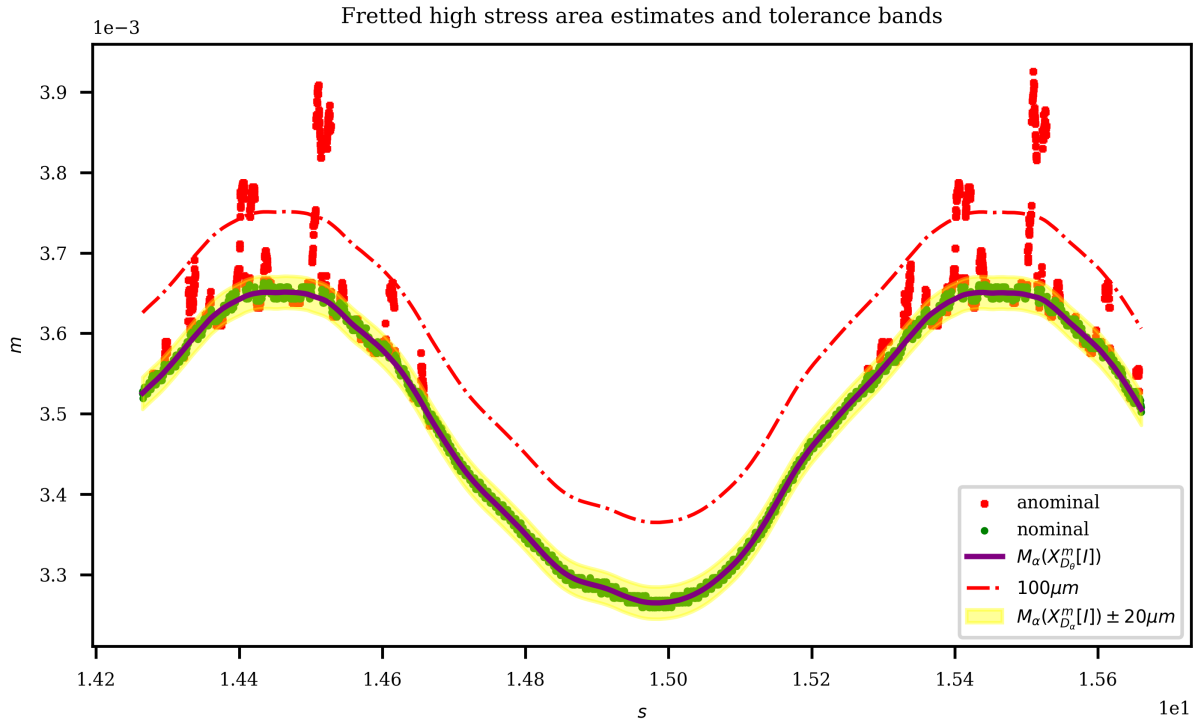


Figure 8.7: Detail is provided of the shaded gray region in Figure 8.6. The dash-dotted red trace gives the $100 \mu m$ threshold above which flaws are considered significant, and the yellow fill shows the $\pm 20 \mu m$ band. Note that any reasonable manual fit of the nominal data would, with margin, fall easily within $\pm 20 \mu m$ of $M_\alpha(X_D^m[I])$.

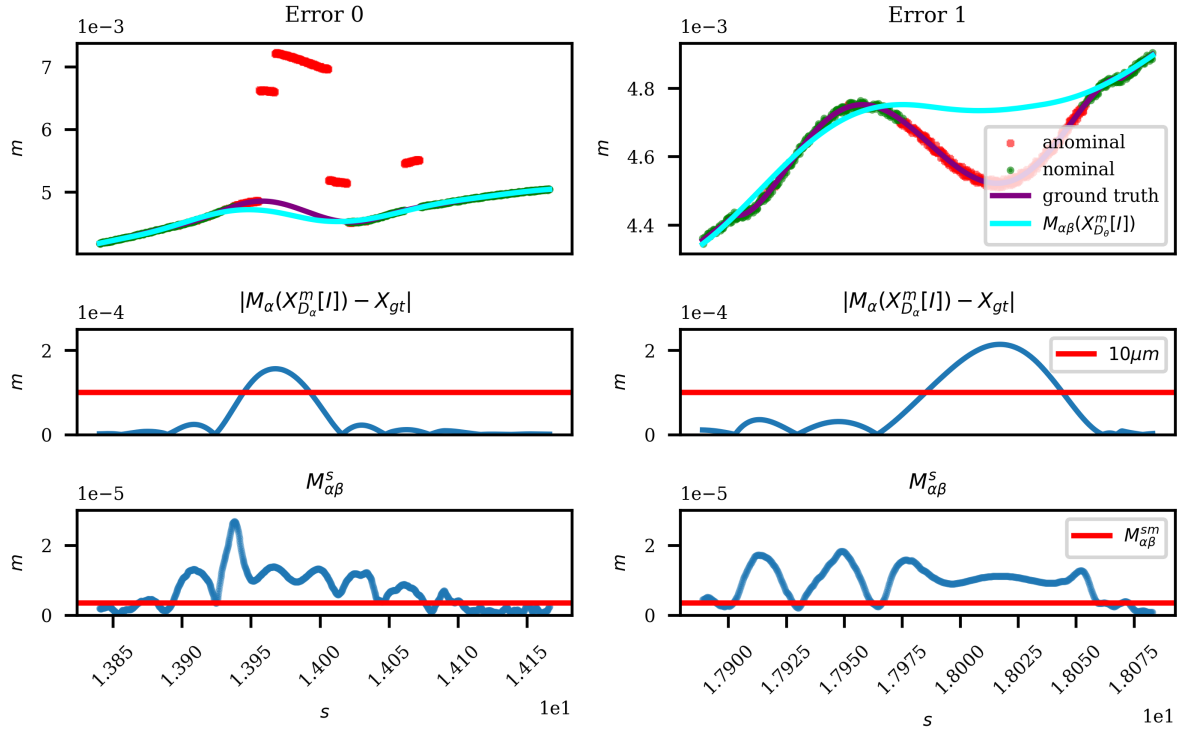


Figure 8.8: Detail is provided of errors in the grey shaded regions of Figure 8.4. As a result of miss-classified data, the accuracy of estimates suffers. While the $M_{\alpha\beta}^s$ diagnostic flags errors, it does not order their magnitude or precise location. Despite these shortcomings, fault diagnosis allows a human in the loop to visually locate errors and manually update P_D .

appear to converge and oscillate around some fixed point. Our testing showed this behavior remained consistent as training runs were extended up to 200 epochs. The stability in $|X[P_V]|$ implies that the solution is insensitive to over-fitting and improves the method’s usefulness in general; training time beyond initial convergence is one less parameter that requires treatment by the practitioner.

8.7 Discussion

[DiffPaC](#) sets itself apart from alternative approaches. It trains independently on each dataset, which removes any requirement or blind faith in generalization. The ability to place a human in the loop reduces the risk associated with errors of automation. This template for generative model usage is different from the discriminative approaches reviewed in [Chapter 4](#) where results ultimately depend on generalization achieved by training over very large datasets and seek possibly in vain, and certainly with no guarantees, to deal with edge cases by curating an increasingly large dataset. De-emphasizing generalization and the forthright disposition of edge cases by facilitating a human in the loop successfully leverages generative modelling to potentially provide automation and decision support in a wide range of real world, possibly safety critical, contexts involving largely nominal data.

Given an appropriate choice of parameters for the [SGF](#) models and [CSDI](#) architecture², [DiffPaC](#) is largely parameter-free, with updates relying only on the learned summary statistics of fits and fit residuals and the stochastic behaviors of the [CSDI](#) diffusion model. There are however many opportunities to tune the update method for dealing with specific datasets. For instance, it is trivial to implement bias that increases the reliance on stochastic behaviors or causes a preference during updates for data with desired characteristics. This supports customization of the method for use cases in various contexts.

²See [Appendix B](#) for details on these.

Chapter 9

Recommendations for implementation and commissioning

In this chapter proposed methods are integrated to take advantage of their respective strengths, the integration is scaled for use on multi-probe inspection datasets, and the challenges related to commissioning such a system are considered.

9.1 Process implementation at scale

Figure 9.1 outlines a process workflow that generates the analysis of time of flight information for an arbitrary **Ultrasound (UT)** probe. The process integrates elements from Chapters 6 and 8, using **Diffusion Partition Consensus (DiffPaC)** generated P_D and $\hat{X}$ with **Variational Autoencoder (VAE)** generated M_V^A as inputs to **Nominal Profiling (NPROF)**.

During a **Pressure Tube (PT)** inspection, each of ten **UT** probes acquires in and around $5.8 \cdot 10^7$ *ascans*. Manual **UT** data analysis may access any of this data but, in general, analysis for a given flaw requires data from only a subset of the available probes [134][135].

Our test system was configured with an AMD 64 core 2.7 *GHz* chip, with 1 *TB* RAM, SSD data storage, and an NVIDIA RTX A6000 GPU. The nearest equivalent system on a well-known Cloud Platform is an A2 series with 48 cores, 1.36 *TB* RAM, and an NVIDIA A100 GPU. This system is available in the cloud for just under \$3.75/*hr*.

Using the test system described above, **DiffPaC** training on **UT** datasets $1 \cdot 10^6$ *ascans* in size typically converged in and around 10 epochs, and each epoch took less than 20

minutes. VAE training was performed for 10 epochs at about two minutes training time per epoch. This gives a total compute requirement of under 215 GPU hours per probe.

Moving to a system in the cloud with a larger GPU would improve both time to convergence and the associated economics. But, the worst case as based on test system compute times, would cost just under \$800 per inspected probe, or approximately \$4K to inspect all the data acquired on the five probes most often considered relevant to analysis [134] [135].

Multiple GPUs may be used to decrease training time. The entire five-probe dataset may be processed in 4 hours using 64 GPUs. This level of parallelization is trivial to implement and allows analysis to occur during data acquisition. However, cloud compute services may not be able to provision such requirements on demand unless special arrangements are made with service providers.

The given figures represent the worst case. They are based on processing the complete five-probe dataset using a desktop system. Though beyond the scope of the present treatment, means to reduce compute requirements significantly exist. This is true because most analysis is straightforward and relies on data from only one probe [134][135]. Taking advantage of this fact could reduce total training costs by as much as 80%.

Sketching this out in greater detail, one approach to reducing required training time is to perform analysis on only the most accurate 20 MHz Normal Beam (NB) probe (used to demonstrate and report results throughout this research). Additional analysis using data from the remaining probes would be performed only at regions where the 20 MHz data is incomplete or ambiguous. This matches the intent of the legacy process which, by procedure, utilizes data from probes only on an as-needed basis [135].

9.2 Elements required for regulatory approval

Adopting an agile software development process would allow a small team of software engineers, human factors engineers, and UT data analysts to quickly integrate a scaled and working system. Initial development and testable usage of a decision support and automation system could proceed alongside using the legacy system. This would involve developing a system offline and testing it in parallel with legacy analysis during inspection campaigns. Parallel development and testing would support a strong informal verification and validation of system results. It would also allow a new inspection process to inherit elements of the legacy process that meaningfully contributed to its success by osmosis. And there would be opportunity to address, in use, the many human factors challenges known

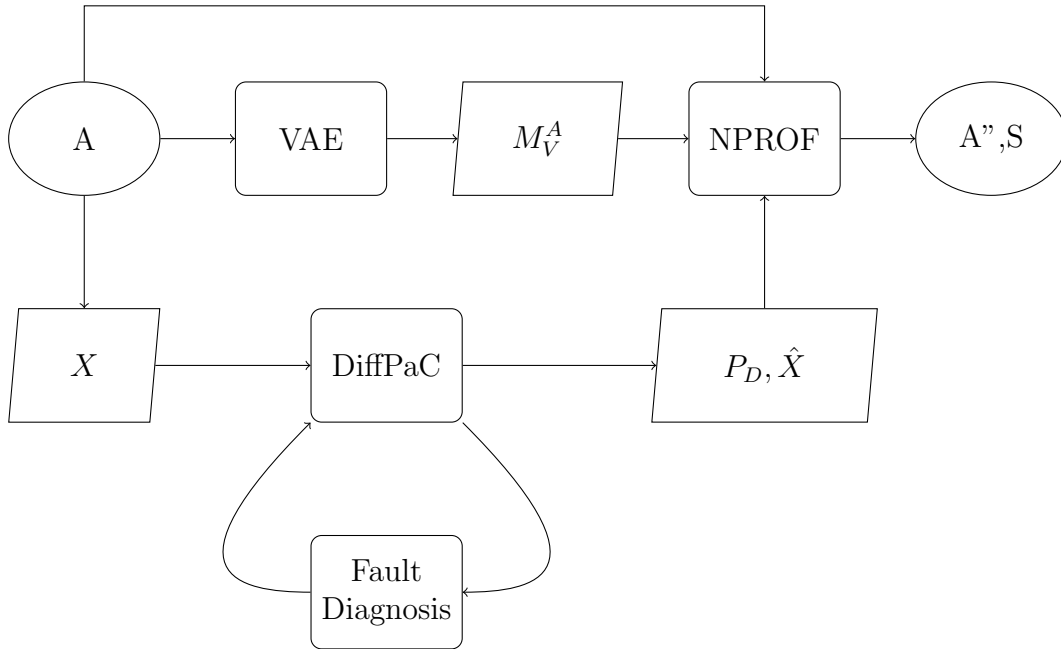


Figure 9.1: Automation and decision support for UT data analysis is achieved by integrating VAE training output M_V^A with DiffPaC training outputs P_D and $\hat{X}$ via NPROF to attain type i/ii/iii analysis elements A'' and S .

to be associated with process automation such as trust, skills fade, and organizational or cultural resistance [9][14][34][36][120][146].

Scientific computing aids, tools, and code in the nuclear industry are subject to stringent software quality assurance programs. An agile development process would be required to meet the needs of such a program through the provision of reasonable quality management systems and documentation such as requirements and plans for development, test, validation, and verification [94][143][144]. Further to this, the regulator requires that inspection processes be qualified by the [CANDU Inspection Qualification Bureau \(CIQB\)](#). To this end, a rigorous technical justification of methods mapping system results to regulatory requirements for inspection [150], and a thorough process for training and testing analysts using the system software would be required. Ultimately, the provision of a well-engineered system with demonstrated quality assurance and [CIQB](#) qualification is necessary to gain regulatory approval [18][19].

9.3 Chapter summary

This chapter contemplated the implementation and commissioning of a scalable integrated multi-probe [UT](#) data analysis system based on proposed methods. Analysis costs and compute requirements, based on training with a desktop PC were evaluated as a worst-case scenario. To improve feasibility, strategies for reducing training time and resource usage were explored. These included selective probe analysis, parallel processing, and the adoption of high-performance GPUs. These approaches suggest that significant cost and time savings of up to 80%, over and against the worst case, are achievable.

Aspects of system implementation, commissioning, and regulatory approval were considered. The adoption of an agile development process and parallel validation with the legacy system supports both system integration and the evolution of appropriate human factors. Meeting nuclear industry software quality standards and achieving [CIQB](#) qualification are essential steps toward commissioning and regulatory acceptance.

The technical aspects required to develop and scale a legitimate process moving forward are fairly well understood. The path to commissioning and regulator acceptance for the use of novel methods is similar in that it is demarcated by a series of well-defined elements. The greatest unknown at this point is the human factors engineering required to transform a set of technologically advanced methods into a process amenable for use by the human analysts in the loop. We suspect human factors engineering may be the single most important element contributing to the over-all success of system deployment [21][145].

Chapter 10

Conclusions

This thesis has addressed the challenge of providing decision support and automation to [Ultrasound \(UT\)](#) data analysis. [Pressure Tube \(PT\) Non Destructive Evaluation \(NDE\)](#) ensures fitness for service and is critical to the safe operation of [Canada Deuterium Uranium \(CANDU\)](#) nuclear reactors. The legacy process relies on manual data analysis, which is resource-intensive, subjective, and repetitive. Motivated by the opportunity for possible improvement through the application of modern methods, this research develops a machine-learning-based approach for decision support and automation that improves process efficiency and efficacy while maintaining safety-critical reliability.

Research initially focused on the use of a [Variational Autoencoder \(VAE\)](#) to form likelihood-based partitions of [UT](#) datasets. [Nominal Profiling \(NPROF\)](#), a novel data profiling technique, was then used to generate the desired analysis by comparing data to its most likely features. This approach was effective but limited by sensitivity to outliers, a lack of robust fault diagnosis, and poor explainability.

Research then turned to the investigation of means to limit the impact of outliers and provide fault diagnosis by integrating generative and deterministic modelling techniques. [Diffusion Partition Consensus \(DiffPaC\)](#) conjoins a [Conditional Score Based Diffusion \(CSDI\)](#) model with a pair of [Savitzky-Golay Filter \(SGF\)](#) models to provide analysis using explainable mechanisms for anomaly detection and fault diagnosis. [DiffPaC](#) achieves desired objectives using an iterative partitioning process that may include a human in the loop to provide required safety critical fail-safe behaviours when faced with edge cases.

Testing was performed on real world [CANDU Calibration Fixture \(CF\)](#), [PT](#), and synthetic datasets. Testing exercised model performance at the extremes of possible, or expected, data distributions. Experiments demonstrated the facility of techniques when

faced with real-world, highly variable, non-stationary data challenges. The integrated system provided estimates and achieved accuracy requirements as specified in [150]. Design requirements, as laid out in Chapter 2, were also fulfilled. The method is a novel approach and establishes state of the art in this niche application context.

A pipeline integrating techniques was developed and scaled for use in cloud-based multi-probe systems. Various elements required for eventual regulator acceptance were considered. The challenges related to human factors were highlighted.

10.1 Key contributions

This thesis has introduced a novel, unsupervised, generative approach for decision support and automation in safety-critical UT data analysis. The proposed system integrates two unsupervised anomaly detection algorithms: VAE-based NPROF and Diffusion-based DiffPaC. It meets stringent regulatory accuracy requirements, provides explainability and fault diagnosis and significantly improves the throughput and consistency of human analysis.

A key innovation of this work lies in its ability to recover hidden time-of-flight information from the data, enabling markedly improved analysis over legacy amplitude-based methods. This represents a first-of-its-kind result with substantial implications for efficiency and diagnostic capability in the UT data analysis workflows.

The core contribution is an efficient and efficacious, safety critical appropriate decision support and automation system that features:

- A robust, principled, and explainable mechanism for fault diagnosis, error remediation and results verification.
- A design that avoids reliance on out-of-distribution generalization,
- An approach that eliminates the technical debt of data labeling and curation inherent to supervised methods.

Together, these contributions demonstrate the viability of generative, unsupervised models for industrial non-destructive evaluation NDE and support the modernization of legacy UT data analysis by providing a technically and economically compelling alternative to existing methodologies. Though not directly comparable, and only in our opinion, the integrated methods make significant qualitative improvements over prior work in this area, and establish state of the art for this application.

10.2 Revisiting requirements and design choices

The development and evaluation of the proposed methods were guided by six requirements given in Chapter 2. The requirements predicated a set of deliberate design choices, all aimed at enabling robust, explainable, and efficient UT data analysis.

The requirement for minimal dataset management (1) was addressed through the exclusive use of unsupervised generative models, trained independently on each incoming dataset. This approach eliminated the need for labelled data or extensive dataset curation, thereby reducing technical debt, and supports the long-term maintenance lifecycle of any intended application.

An ability to outperform the efficiency and efficacy of manual data analysis (2) was demonstrated by a scalable implementation and automation of type i/ii/iii signal deviation detection. The integration of methods provides a system that automates the recovery of all time of flight information relevant to data analysis much more efficiently than legacy amplitude-based methods.

The human performance of analysts engaging in a process that has implemented proposed methods will necessarily improve. The estimation of type i/ii/iii data deviation constitutes the bulk of the work performed during manual analysis. The tedious repetitive nature of producing these estimates makes human performance error more likely. Proposed methods do this work automatically and can be made to provide dimensional analysis of many flaws with straight forward geometry. This removes a significant opportunity for human performance error from analyst workflows.

The design emphasized both a principled mathematical foundation and explainability (3). The use of variational inference and score-based learning provided a rigorous theoretical basis for model behavior, while the reduction of the primary task to a binary classification problem lent visual interpretability of results and supported explainability.

Robustness to out-of-distribution data and class imbalance (4) was achieved by avoiding model generalization altogether. Training separate models for each dataset mitigated risks related to non-stationary data and provided a segue for human-in-the-loop oversight during training.

The system was explicitly designed to operate within the constraints of safety-critical environments (5). Incorporating a human-in-the-loop and a fault diagnosis mechanism provided an intentional fail-safe mechanism.

Ultimately, the system provided dimensional analysis to the accuracy required by the regulator (6). While more complex flaw geometries may still require manual interpretation,

the automated identification of relevant data deviations ensures that analysts are equipped with comprehensive decision support sufficient to enable all subsequent analysis.

As a whole, design choices were validated through practical implementation and evaluation of proposed methods. They supported the development of a system that satisfies the given requirements and offers a meaningful path forward for automation and decision support in [UT](#) data analysis.

10.3 Limitations

10.3.1 Formal evaluation of fault diagnostic

Strong confidence in model outputs must be established for use in safety-critical contexts. It has been demonstrated that there is reason to be confident in using $M_{\alpha\beta}$ for fault diagnosis. And though it is reasonable to assume that accuracy of $\hat{X}$ cannot be better than some measure related to $M_{\alpha\beta}^s$, given the fault diagnosis mechanism must provide fail-safe behaviour, the exact nature of this measure and the strengths and weaknesses of the relation require some formal examination. Integration with some aspects of legacy methods to guarantee proposed methods are “as good or better than legacy” may be required. This may be accomplished by testing on a large corpus of data and characterizing $M_{\alpha\beta}$ response.

10.3.2 Establishing probability of detection

The justification of the legacy process involved a rigorous determination of the probability of detection of flaws that could be expected from a combination of amplitude and time-of-flight-based flaw identification. The argument considered probe configurations, the efficacy of methods, and best practice as defined by industry standard. The net result was an accepted probability of flaw detection. [DiffPaC](#) is a time-of-flight and likelihood-based method for flaw detection. And though the [VAE](#)-based likelihood analysis underlying [NPROF](#) shares many characteristics with amplitude-based flaw detection, it is obviously distinct from it. Some formal work must yet be done to determine how the probability of flaw detection using proposed methods compares to that of legacy processes [\[150\]](#) [\[68\]](#)[\[135\]](#). From experience, and in our opinion, we believe the proposed methods improve the probability of flaw detection; however, this result must be formally obtained.

10.4 Future work

10.4.1 Implementation and regulatory acceptance

As described in the previous section, several elements required for a complete technical justification of methods remain outstanding. These elements are a subset of the elements required for commissioning and regulatory acceptance outlined in Chapter 9.

10.4.2 Integrating discriminative methods over time

A significant amount of work has been done in using discriminative methods for flaw sizing and categorization. The problem with such methods, fully addressed in this research, is the requirement for out-of-distribution generalization, dataset curation, and labeling. However, opportunities exist for taking advantage of discriminative methods. Chapter 4 cites the relevant research using discriminative and heuristic methods for flaw sizing and categorization [49][141]. It may be possible to bootstrap the cited methods using those presently proposed. The process in Figure 9.1 projects X onto A'' . Where A'' is in a space closer to an unobserved noiseless actual system state or common distribution. The analysis process itself labels this data. Thus, over time, the process accrues a well-projected, or denoised, labelled dataset. A'' is also susceptible to straightforward data augmentation. This may support integrating discriminative and heuristic methods applied in a denoised space.

10.4.3 Multidimensional/multivariate applications

Though CSDI supports multivariate data, DiffPaC, as proposed, is a univariate approach. UT data is high dimensional, so there is ample opportunity to be creative in terms of expressing and solving a multivariate analysis problem. A univariate approach has been selected for the following reasons:

- Explainability: DiffPaC approximates sampling an approximate learned data distribution using a pair of SGFs. SGF is available as a two-dimensional filter, but the one-dimensional use case works well and has the upside that it is explainable. Each cull and reconstruction of data operates on the basis of a learned summary statistic and what is essentially a very simple geometric argument that relates assumptions and contradictions. This translates into readily understood and visualized reasoning

about the dataset by the machine learning process. Increasing the dimension would muddy this explainability considerably.

- **Accuracy:** in unsupervised generative conditional modeling, increasing the dimension of the problem space necessarily increases the density of spurious correlation in the dataset. The knock-on effect is a decreased fidelity in learned approximate conditional data distributions. This finding influenced the arc of research towards framing the problem in terms of continuous univariate time series rather than a two-dimensional image.
- **CSDI window size:** the **CSDI** model field of vision is based on a window size. **DiffPaC** sensitivity is governed by this window size, which must be larger than the flaw for which analysis is desired. During experimentation with multivariate datasets and **CSDI**, when the dataset dimension was increased, window size had to be decreased in order to maintain reasonable model memory footprint and training time. This is problematic because, in the axial dimension, reactor fueling operations cause very long flaws orders of magnitude larger than those in the radial.

Pulling the multidimensional thread, given some reasonably well-behaved dataset $D \in \mathcal{R}^{n \times m}$ ¹ with distribution $p(D)$. And some sensible $n \times m$ dimensional ground truth binary classification over the dataset. A necessary condition for a multidimensional approach would be the existence of a closed form approximation of conditional $p(\hat{D}|D)$ that could be bootstrapped or somehow “convolved”, by way of some ad-hoc or formal method, with a current estimate of $p(D)$ to provide $p(\hat{D}|D)p(D)$. For a complete generalization, one might provide some conditions under which closed-form approximations always exist given an appropriately well-behaved multidimensional dataset. The success of **DiffPaC** could imply the existence of this interesting general class of problems and solutions. How such a thing would improve the near-term state for **UT** inspection data analysis is uncertain. And so, this possibility is referenced only as a recommendation for further research.

10.5 Conclusions

This research marks a significant advance in the application of machine-learning to industrial inspection processes. It demonstrates how unsupervised generative methods can meet stringent safety and reliability standards while enabling practical decision support and

¹Formally, there would be some continuity and regularity conditions a dataset must observe.

automation. However, given the risks associated with possible error, and without mathematical guarantees on performance, it is unlikely that full automation of a safety critical process, such as [UT](#) data analysis will ever be adopted. As such, the quality of results, decoupled from generalization, and the inclusion of a human in the loop for active learning through fine-tuning, position [DiffPaC](#) as a possible means to address safety-critical requirements and realize the efficiency modern machine-learning methods offer a legacy industrial process [\[20\]](#).

With accuracy, objectivity, and repeatability that meets or exceeds manual analysis by some degree, and the efficiency of a learning process that can do in minutes what might have taken a team of analysts hours or days, [DiffPaC](#) gives ample reason for cautious optimism with regard to its potential. To develop a refactored [DiffPaC](#)-based analysis process in earnest, the next steps may be to implement a prototype that can run in parallel with manual analysis. This would allow an opportunity to flush out the details of a production system by exposing edge cases, vagaries and catch points in the datasets and processes currently in use. Such parallel operation would provide grounds to establish the strengths and weaknesses of proposed methods and also provide segue to understand the human factors that will shape an eventual process.

References

- [1] Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(50):1–34, 2018.
- [2] International Atomic Energy Agency. Managing human performance to improve nuclear facility operation. Technical report, IAEA, 2016.
- [3] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 291–300. IEEE, 2019.
- [4] Saleema Amershi, Maya Cakmak, W. Bradley Knox, and Tessa Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014.
- [5] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.
- [6] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, pages 1–18, 2015.
- [7] Martin Arjovsky. Out of distribution generalization in machine learning. *arXiv preprint arXiv:2103.02667*, 2021.
- [8] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.
- [9] Lisanne Bainbridge. Ironies of automation. *Automatica*, 19(6):775–779, 1983.

- [10] Yoshua Bengio. Deep learning of representations: Looking forward. In *Proceedings of the International Conference on Statistical Language and Speech Processing*, pages 1–37. Springer, 2013.
- [11] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.
- [12] Christopher M. Bishop and Nasser M. Nasrabadi. *Pattern recognition and machine learning*. Springer, 2006.
- [13] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β -vae. *arXiv preprint arXiv:2006.15680*, 2018.
- [14] Lisa A. Burke and Holly M. Hutchins. Skill decay: A review of the literature and implications for training and skill retention in the nuclear industry. *International Journal of Training and Development*, 11(1):33–49, 2007.
- [15] Davide Cacciarelli and Murat Kulahci. Active learning for data streams: a survey. *Machine Learning*, 113(1):185–239, 2024.
- [16] Weiwei Cai, Ying Zhang, and Jun Zhou. Maximizing expected model change for active learning in regression. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 51–60, 2017.
- [17] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining*, volume 7819, pages 160–172. Springer, 2013.
- [18] Canadian Nuclear Safety Commission. Regulatory standard s-99: Reporting requirements for operating nuclear power plants. Technical report, Canadian Nuclear Safety Commission, Ottawa, Canada, 2003.
- [19] CANDU Owners Group. Ciqb process for inspection qualification. Technical Report COG-06-9015, CANDU Owners Group, Toronto, Ontario, 2006. Restricted access – internal publication.
- [20] Sergio Cantero-Chinchilla, Paul D Wilcox, and Anthony J Croxford. Deep learning in automated ultrasonic nde–developments, axioms and opportunities. *Ndt & E International*, 131:102703, 2022.

- [21] John M. Carroll. Human-computer interaction: Psychology as a science of design. *Annual Review of Psychology*, 48(1):61–83, 1997.
- [22] Young-Jin Cha, Rahmat Ali, John Lewis, and Oral Buyukozturk. Deep learning-based structural health monitoring. *Automation in Construction*, 161:105328, 2024.
- [23] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.
- [24] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009.
- [25] BA Cheadle, CE Coleman, and H Licht. Candu-phw pressure tubes: their manufacture, inspection, and properties. *Nuclear Technology*, 57:413–425, 1982.
- [26] B.A. Cheadle, C.E. Coleman, D.K. Rodgers, P.H. Davies, C.K. Chow, and M. Griffiths. Examination of core components removed from candu reactors. Technical Report AECL–9710, Atomic Energy of Canada Limited (AECL), 1988.
- [27] BA Cheadle and EG Price. Operating performance of candu pressure tubes. Technical report, Atomic Energy of Canada Ltd., 1989.
- [28] C. Chow, C. Coleman, R. Hosbons, P. Davies, M. Griffiths, and R. Choubey. Fracture toughness of irradiated zr-2.5nb pressure tubes from candu reactors. Technical Report STP 25511S, ASTM International, 1991.
- [29] Jamie Coble, Pradeep Ramuhalli, Leonard J Bond, J Hines, and Belle Ipadhyaya. A review of prognostics and health management applications in nuclear power plants. *International Journal of prognostics and health management*, 6:016, 2015.
- [30] Olympus Corporation. *Ultrasonic Transducers Technical Notes*. Olympus Corporation, 2019.
- [31] Olympus Corporation. *Ultrasonic Flaw Detection Tutorial*. Olympus Corporation, 2021.
- [32] Olympus Corporation. *Ultrasonic Thickness Gauge Tutorial*. Olympus Corporation, 2021.
- [33] Kate Crawford and Ryan Calo. There is a blind spot in ai research. *Nature*, 538(7625):311–313, 2016.

- [34] M. L. Cummings. Automation and accountability in decision support system interface design. In *Proceedings of the AIAA 1st Intelligent Systems Technical Conference*, Arlington, VA, 2006. AIAA.
- [35] Roxana Daneshjou, Kailas Vodrahalli, Roberto A. Novoa, Melissa Jenkins, Weixin Liang, Veronica Rotemberg, Justin Ko, Susan M. Swetter, Elizabeth E. Bailey, Olivier Gevaert, Pritam Mukherjee, Michelle Phung, Kiana Yekrang, Bradley Fong, Rachna Sahasrabudhe, Johan A. C. Allerup, Utako Okata-Karigane, James Zou, and Albert S. Chiou. Disparities in dermatology ai performance on a diverse, curated clinical image set. *Science Advances*, 8(32):6147, 2022.
- [36] Frédéric Dehais, Raphaël N. Roy, Samuel Scannella, and Aurélien Lafont. The paradox of automation: The more we automate, the more we need human expertise. *Current Directions in Psychological Science*, 29(4):386–391, 2020.
- [37] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2019.
- [38] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.
- [39] Bowen Du, Xuanxuan Sun, Junchen Ye, Ke Cheng, Jingyuan Wang, and Leilei Sun. Gan-based anomaly detection for multivariate time series using polluted training set. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12208–12219, 2021.
- [40] Michael T. Dzindolet, Shane A. Peterson, Rachel A. Pomranky, Lawrence G. Pierce, and Heather P. Beck. The role of trust in automation reliance. *International Journal of Human-Computer Studies*, 58(6):697–718, 2003.
- [41] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.
- [42] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.
- [43] Thomas S. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

- [44] Wm. J. Garland, editor. *The Essential CANDU: A Textbook on the CANDU Nuclear Power Plant Technology*. University Network of Excellence in Nuclear Engineering (UNENE), 2024.
- [45] Rui Gongzhang and Anthony Gachagan. Advanced defect detection algorithm using clustering in ultrasonic nde. *AIP Conference Proceedings*, 1706(1):180009, 2016.
- [46] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- [47] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2015.
- [48] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.
- [49] Issam Hammad, Ryan Simpson, Hippolyte Djonon Tsague, and Sarah Hall. Using deep learning to automate the detection of flaws in nuclear fuel channel ut scans. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 69(1):323–329, Jan 2022.
- [50] Sahar Hassani and Ulrike Dackermann. A systematic review of advanced sensor technologies for non-destructive testing and structural health monitoring. *Sensors*, 23(4):2204, 2023.
- [51] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.
- [52] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.
- [53] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.
- [54] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.
- [55] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

- [56] RA Holt. In-reactor deformation of cold-worked zr-2.5 nb pressure tubes. *Journal of Nuclear Materials*, 372(2-3):182–214, 2008.
- [57] Andreas Holzinger. Interactive machine learning for health informatics: When do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, 2016.
- [58] Michael C Hughes and Erik Sudderth. Memoized online variational inference for dirichlet process mixture models. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.
- [59] International Atomic Energy Agency (IAEA). Assessment and management of ageing of major nuclear power plant components important to safety: Candu pressure tubes. Technical Report IAEA-TECDOC-1037, International Atomic Energy Agency, August 1998.
- [60] E. F. Ibrahim and B. A. Cheadle. Development of zirconium alloys for pressure tubes in candu reactors. *Canadian Metallurgical Quarterly*, 24(3):273–281, 1985.
- [61] Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary conditioning. *arXiv preprint arXiv:1806.02382*, 2019.
- [62] Rafael Junges, Luca Lomazzi, Lorenzo Miele, Marco Giglio, and Francesco Cadini. Mitigating the impact of temperature variations on ultrasonic guided wave-based structural health monitoring through variational autoencoders. *Sensors*, 24(5):1494, 2024.
- [63] A. Karpelson. Operation of highly focussed immersion ultrasonic transducers at elevated temperatures. *e-Journal of Nondestructive Testing*, 2004.
- [64] A. Karpelson. Ultrasonic characterization of flaws. *e-Journal of Nondestructive Testing*, 2008.
- [65] A. Karpelson. Ultrasonic measurement of oxide layer thickness and oxide influence on flaw detection and sizing in pressure tubes. *e-Journal of Nondestructive Testing*, 2021.
- [66] A. Karpelson. Single ultrasonic probe able to reproduce flaw shape. *e-Journal of Nondestructive Testing*, 2025.
- [67] Alex Karpelson. Impact of temperature variations on flaw detection and sizing. *e-Journal of Nondestructive Testing*, 2013.

- [68] Alex Karpelson. Probability of detection and uncertainty of measurement during ultrasonic inspection of candu reactor pressure tubes. *e-Journal of Nondestructive Testing*, 2015.
- [69] Sylvia Keßler, Vamsi Krishna Rentala, Fabian Dethof, and Daniel Kanzler. Application of the german standardization approach for the reliability assessment of ndt-honeycomb detection in concrete using ultrasonic testing. *Journal of Advanced Concrete Technology*, 22(10):664–673, 2024.
- [70] Valentin Khruikov and Ivan Oseledets. Geometry score: A method for comparing generative adversarial networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2621–2629. PMLR, 10–15 Jul 2018.
- [71] Tung Kieu, Bin Yang, Chenjuan Guo, Christian S. Jensen, Yan Zhao, Feiteng Huang, and Kai Zheng. Robust and explainable autoencoders for unsupervised time series outlier detection. In *Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 3038–3050. IEEE, 2022.
- [72] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2014.
- [73] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.
- [74] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.
- [75] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.
- [76] Thanasis Kotsiopoulos, Gerasimos Papakostas, Thanasis Vafeiadis, Vasileios Dimitriadis, Alexandros Nizamis, Andrea Bolzoni, Davide Bellinati, Dimosthenis Ioannidis, Konstantinos Votis, Dimitrios Tzovaras, et al. Revolutionizing defect recognition in hard metal industry through ai explainability, human-in-the-loop approaches and cognitive mechanisms. *Expert Systems with Applications*, 255:124839, 2024.

- [77] Josef Krautkramer and Herbert Krautkramer. *Ultrasonic Testing of Materials*. Springer, 3 edition, 1983.
- [78] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in Neural Information Processing Systems*, 36, 2024.
- [79] T. Lardner, G. M. West, G. Dobie, and A. Gachagan. An expert-systems approach to automatically determining flaw depth within candu pressure tubes. Technical report, OSTI.GOV, United States, 6 2017.
- [80] Timothy Lardner, Graeme West, Gordon Dobie, and Anthony Gachagan. Automated sizing and classification of defects in candu pressure tubes. *Nuclear Engineering and Design*, 325:25–32, 2017.
- [81] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Xiaojun Xu, Yuguang Yao, Hang Li, Kush R. Varshney, et al. Rethinking machine unlearning for large language models. *arXiv preprint arXiv:2402.08787*, 2024.
- [82] Gabriel Loaiza-Ganem, Brendan Leigh Ross, Rasa Hosseinzadeh, Anthony L Caterini, and Jesse C Cresswell. Deep generative models through the lens of the manifold hypothesis: A survey and new connections. *arXiv preprint arXiv:2404.02954*, 2024.
- [83] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2019.
- [84] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.
- [85] Duje Medak, Luka Posilovic, Marko Subasic, Marko Budimir, and Sven Loncaric. Automated defect detection from ultrasonic images using deep learning. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 68(10):3126–3134, 2021.
- [86] Sreeram V Menon and Chandra Sekhar Seelamantula. Robust savitzky-golay filters. In *2014 19th International Conference on Digital Signal Processing*, pages 688–693. IEEE, 2014.

- [87] Norbert Meyendorf, Nathan Ida, Ripudaman Singh, and Johannes Vrana. Nde 4.0: Progress, promise, and its role to industry 4.0. *NDT & E International*, 140:102957, 2023.
- [88] Fran Milković, Branimir Filipović, Marko Subašić, Tomislav Petković, Sven Lončarić, and Marko Budimir. Ultrasound anomaly detection based on variational autoencoders. In *2021 12th International symposium on image and signal processing and analysis (ISPA)*, pages 225–229. IEEE, 2021.
- [89] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3523–3542, 2021.
- [90] GD Moan, CE Coleman, EG Price, DK Rodgers, and S Sagat. Leak-before-break in the pressure tubes of candu reactors. *International Journal of Pressure Vessels and Piping*, 43(1-3):1–21, 1990.
- [91] Aiswarya Raj Munappy, Jan Bosch, Helena Holmström Olsson, Anders Arpteg, and Björn Brinne. Data management for production quality deep learning models: Challenges and solutions. *Journal of Systems and Software*, 191:111359, 2022.
- [92] Eliya Nachmani, Robin San Roman, and Lior Wolf. Denoising diffusion gamma models. *OpenReview*, 2022.
- [93] Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*, 2020.
- [94] Josh Newell, Linna Pang, David Tremaine, Alan Wassyng, and Mark Lawford. Translation of iec 61131-3 function block diagrams to pvs for formal verification with real-time nuclear application. *Journal of Automated Reasoning*, 60(1):63–84, 2018.
- [95] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS’01, page 841–848. MIT Press, 2001.
- [96] Stephen Odaibo. Tutorial: Deriving the standard variational autoencoder (vae) loss function. *arXiv preprint arXiv:1907.08956*, 2019.

- [97] Mihajlo Ogrizović, Dražen Drašković, and Dragan Bojić. Quality assurance strategies for machine learning applications in big data analytics: an overview. *Journal of Big Data*, 11(1):156, 2024.
- [98] Nastaran Okati, Abir De, and Manuel Rodriguez. Differentiable learning under triage. *Advances in Neural Information Processing Systems*, 34:9140–9151, 2021.
- [99] Raja Parasuraman and Victor Riley. Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39(2):230–253, 1997.
- [100] Raja Parasuraman, Thomas B Sheridan, and Christopher D Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30(3):286–297, 2000.
- [101] Adrian Alan Pol, Victor Berger, Gianluca Cerminara, Cecile Germain, and Maurizio Pierini. Anomaly detection with conditional variational autoencoders. *arXiv preprint arXiv:2010.05531*, 2020.
- [102] Luka Posilović, Duje Medak, Marko Subašić, Marko Budimir, and Sven Lončarić. Generating ultrasonic images indistinguishable from real images using generative adversarial networks. *Ultrasonics*, 119:106610, 2022.
- [103] EG Price and BA Cheadle. Fast fracture of a zirconium alloy pressure tube: Cause and implications. Technical report, Atomic Energy of Canada Ltd., 1985.
- [104] Richard J Pyle, Rhodri LT Bevan, Robert R Hughes, Rosen K Rachev, Amine Ait Si Ali, and Paul D Wilcox. Deep learning for ultrasonic crack characterization in nde. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 68(5):1854–1865, 2020.
- [105] Richard J. Pyle, Robert R. Hughes, and Paul D. Wilcox. Interpretable and explainable machine learning for ultrasonic defect sizing. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 70(4):277–290, 2023.
- [106] James Reason. *Human Error*. Cambridge University Press, Cambridge, UK, 1990.
- [107] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- [108] Karl Ridgeway. A survey of inductive biases for factorial representation-learning. *arXiv preprint arXiv:1612.05299*, 2016.

- [109] D.K. Rodgers, C.E. Coleman, M. Griffiths, G.A. Bickel, J.R. Theaker, I. Muir, A.A. Bahurmuz, S. St. Lawrence, and M. Resta Levi. In-reactor performance of pressure tubes in candu reactors. *Journal of Nuclear Materials*, 383(1):22–27, 2008.
- [110] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.
- [111] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. Technical report, California University, San Diego, La Jolla, Institute for Cognitive Science, 1985.
- [112] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
- [113] Feras Saad and Vikash Mansinghka. Temporally-reweighted chinese restaurant process mixtures for clustering, imputing, and forecasting multivariate time series. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 755–764. PMLR, 09–11 Apr 2018.
- [114] Anup K. Sahoo and Mahesh D. Pandey. A risk-informed approach to assessment of dhc initiation in pressure tubes of candu reactors. *Nuclear Engineering and Design*, 240(3):630–638, 2010.
- [115] Rick Salay, Rodolfo Queiroz, and Krzysztof Czarnecki. An analysis of ISO 26262: Using machine learning safely in automotive software. In *IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 636–642, 2017.
- [116] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.
- [117] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.
- [118] Guus Schreiber, Bob Wielinga, Robert Hoog, Hans Akkermans, and Walter Velde. Commonkads: A comprehensive methodology for kbs development. *IEEE Expert*, 9:28–37, 12 1994.

- [119] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.
- [120] Thomas B. Sheridan. *Human-Centered Automation: Where Are We and Where Do We Go From Here?* Lawrence Erlbaum Associates, Mahwah, NJ, 2002.
- [121] RN Singh, R Kishore, SS Singh, TK Sinha, and BP Kashyap. Stress-reorientation of hydrides and hydride embrittlement of zr-2.5 wt% nb pressure tube alloy. *Journal of nuclear materials*, 325(1):26–33, 2004.
- [122] RN Singh, Niraj Kumar, R Kishore, S Roychaudhury, TK Sinha, and BP Kashyap. Delayed hydride cracking in zr-2.5 nb pressure tube material. *Journal of Nuclear Materials*, 304(2-3):189–203, 2002.
- [123] RN Singh, S Mukherjee, Anuja Gupta, and S Banerjee. Terminal solid solubility of hydrogen in zr-alloy pressure tube materials. *Journal of Alloys and Compounds*, 389(1-2):102–112, 2005.
- [124] Vít Škvára, Jan Francá, Matěj Zorek, Tomáš Pevný, and Václav Šmídl. Comparison of anomaly detectors: Context matters. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2494–2507, 2021.
- [125] Kihyuk Sohn, Xinchun Yan, and Honglak Lee. Learning structured output representation using deep conditional generative models. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’15, page 3483–3491. MIT Press, 2015.
- [126] Andrzej Słowik. *Out-of-Distribution Generalisation in Machine Learning*. PhD thesis, University of Cambridge, 2022.
- [127] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.
- [128] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csd: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.
- [129] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

- [130] JunYong Tong and Nick Torenvliet. Temporally-reweighted dirichlet process mixture anomaly detector. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 267–274, 2020.
- [131] Nick Torenvliet, Yizhe Liu, and John Zelek. Automating safety critical ultrasonic data analysis with a variational auto-encoder. In *2023 IEEE Sensors Applications Symposium (SAS)*, pages 1–6, 2023.
- [132] Nick Torenvliet and John Zelek. Evaluating diffusion models for the automation of ultrasonic nondestructive evaluation data analysis. *Algorithms*, 17(4), 2024.
- [133] Nick Torenvliet and John S. Zelek. Diffusion partition consensus: Diffusion-aided time-of-flight estimates, anomaly detection, and localization for ultrasonic nondestructive evaluation data. *IEEE Open Journal of Instrumentation and Measurement*, 3:1–11, 2024.
- [134] Michael Trelinski. Pressure tube flaws and artifacts observed in candu fuel channels; detection, sizing and characterization issues. In *IV Conferencia Panamericana de END, Buenos Aires, Argentina, Oct*, pages 22–26, 2007.
- [135] Michael Trelinski. Inspection of candu reactor pressure tubes using ultrasonics. In *17th World Conference on Nondestructive Testing*, pages 25–28, 2008.
- [136] E Tulk, M Kerr, and MR Daymond. Study on the effects of matrix yield strength on hydride phase stability in zircaloy-2 and zr 2.5 wt% nb. *Journal of Nuclear Materials*, 425(1-3):93–104, 2012.
- [137] Vedran Tunukovic, Shaun McKnight, Richard Pyle, Zhiming Wang, Ehsan Mohseni, S Gareth Pierce, Randika KW Vithanage, Gordon Dobie, Charles N MacLeod, Sandy Cochran, et al. Unsupervised machine learning for flaw detection in automated ultrasonic testing of carbon fibre reinforced plastic composites. *Ultrasonics*, 140:107313, 2024.
- [138] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12, 2016.
- [139] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson,

- C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [140] J.R. Walker. A probabilistic approach to leak-before-break in candu pressure tubes. *International Journal of Pressure Vessels and Piping*, 43(1):229–239, 1990.
- [141] C. Wallace, G. West, P. Zacharis, G. Dobie, and A. Gachagan. Experience, testing and future development of an ultrasonic inspection analysis defect decision support tool for candu reactors. In *11th Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies (NPIC&HMIT) 2019*, February 2019.
- [142] Lingzhi Wang, Xingshan Zeng, Jinsong Guo, Kam-Fai Wong, and Georg Gottlob. Selective forgetting: Advancing machine unlearning techniques and evaluation in language models. *arXiv preprint arXiv:2402.05813*, 2024.
- [143] Alan Wassyng and Mark Lawford. Lessons learned from a successful implementation of formal methods in an industrial project. In *FME 2003: Formal Methods: International Symposium of Formal Methods Europe, Pisa, Italy, September 8-14, 2003. Proceedings*, pages 133–153. Springer, 2003.
- [144] Alan Wassyng and Mark Lawford. Software tools for safety-critical software development. *International Journal on Software Tools for Technology Transfer*, 8:337–354, 2006.
- [145] S. J. Westerman and G. R. J. Hockey. Human factors in safety-critical systems: An underestimated contribution? In *Engineering Psychology and Cognitive Ergonomics*, pages 217–224. Springer, 1996.
- [146] Christopher D. Wickens, John D. Lee, Yili Liu, and Sallie E. Gordon-Becker. *Engineering Psychology and Human Performance*. Pearson Education, Upper Saddle River, NJ, 3rd edition, 2004.
- [147] H.W. Wong, V.K. Bajaj, G.D. Moan, M. Huterer, and C.O. Poidevin. The role of leak-before-break in assessments of flaws detected in candu pressure tubes. *International Journal of Pressure Vessels and Piping*, 43(1):23–37, 1990.
- [148] Xuefei Wu and Yangyong Zhu. Human-in-the-loop learning for text classification. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

- [149] Hongzuo Xu, Yijie Wang, Songlei Jian, Qing Liao, Yongjun Wang, and Guansong Pang. Calibrated one-class classification for unsupervised time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- [150] S. Xu, N.P. Singh, L. Gutkin, M. Elmoselhi, I. Balbaa, and A. Karpelson. Inspection specification for candu fuel channels. Internal Report COG-JP-4027-V10-R01, CANDU Owners Group Inc., September 2008.
- [151] Yiwei Yang, Eser Kandogan, Yunyao Li, Prithviraj Sen, and Walter S Lasecki. A study on interaction in human-in-the-loop machine learning for text analytics. In *ACM IUI Workshops*, 2019.
- [152] Panagiotis Zacharis, Graeme West, Gordon Dobie, Timothy Lardner, and Anthony Gachagan. Data-driven analysis of ultrasonic inspection data of pressure tubes. *Nuclear Technology*, 202(2-3):153–160, 2018.
- [153] Huan Zhao, Anthony Gachagan, Gordon Dobie, Christopher Wallace, and Graeme West. Synthetic aperture focusing technique for correction of poorly-focused ultrasonic pressure tube inspection data. In *9th European Workshop on Structural Health Monitoring Series (EWSHM)*, 2018.
- [154] Sharon Zhou, Eric Zelikman, Fred Lu, Andrew Y. Ng, Gunnar E. Carlsson, and Stefano Ermon. Evaluating the disentanglement of deep generative models through manifold topology. In *International Conference on Learning Representations*, 2021.
- [155] Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf. S3vae: Self-supervised sequential vae for representation disentanglement and data generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6538–6547, 2020.

APPENDICES

Appendix A

Diffusion model analysis

The following tables provide the full set of results for experimentation performed in Chapter [7](#).

Table A.1: Training with missing ratio of 10%.

Dataset	Model	MAE	MAPE	MSE	MXAE
$P_{0.70}$	csdi	$5.04 \cdot 10^3$	$1.94 \cdot 10^{-2}$	$4.71 \cdot 10^{-5}$	$3.06 \cdot 10^{-2}$
$P_{0.70}$	ddpm	$1.38 \cdot 10^2$	$1.47 \cdot 10^{-1}$	$1.67 \cdot 10^{-3}$	$4.68 \cdot 10^{-1}$
$P_{0.75}$	csdi	$5.78 \cdot 10^3$	$3.55 \cdot 10^{-2}$	$6.00 \cdot 10^{-5}$	$3.52 \cdot 10^{-2}$
$P_{0.75}$	ddpm	$8.90 \cdot 10^3$	$8.67 \cdot 10^{-2}$	$4.99 \cdot 10^{-4}$	$2.68 \cdot 10^{-1}$
$P_{0.80}$	csdi	$3.96 \cdot 10^3$	$2.12 \cdot 10^{-2}$	$2.83 \cdot 10^{-5}$	$3.11 \cdot 10^{-2}$
$P_{0.80}$	ddpm	$5.76 \cdot 10^3$	$4.54 \cdot 10^{-2}$	$1.08 \cdot 10^{-4}$	$9.51 \cdot 10^{-2}$
$P_{0.85}$	csdi	$4.06 \cdot 10^3$	$1.51 \cdot 10^{-2}$	$2.82 \cdot 10^{-5}$	$2.55 \cdot 10^{-2}$
$P_{0.85}$	ddpm	$5.02 \cdot 10^3$	$4.05 \cdot 10^{-2}$	$6.09 \cdot 10^{-5}$	$5.75 \cdot 10^{-2}$
$P_{0.90}$	csdi	$3.89 \cdot 10^3$	$2.02 \cdot 10^{-2}$	$2.61 \cdot 10^{-5}$	$3.46 \cdot 10^{-2}$
$P_{0.90}$	ddpm	$4.06 \cdot 10^3$	$1.82 \cdot 10^{-2}$	$3.14 \cdot 10^{-5}$	$6.28 \cdot 10^{-2}$
$P_{0.91}$	csdi	$4.17 \cdot 10^3$	$1.60 \cdot 10^{-2}$	$3.22 \cdot 10^{-5}$	$3.22 \cdot 10^{-2}$
$P_{0.91}$	ddpm	$3.92 \cdot 10^3$	$3.48 \cdot 10^{-2}$	$3.02 \cdot 10^{-5}$	$4.55 \cdot 10^{-2}$
$P_{0.92}$	csdi	$4.69 \cdot 10^3$	$1.96 \cdot 10^{-2}$	$3.73 \cdot 10^{-5}$	$3.17 \cdot 10^{-2}$
$P_{0.92}$	ddpm	$3.81 \cdot 10^3$	$1.61 \cdot 10^{-2}$	$2.56 \cdot 10^{-5}$	$3.79 \cdot 10^{-2}$
$P_{0.93}$	csdi	$3.91 \cdot 10^3$	$1.29 \cdot 10^{-2}$	$2.65 \cdot 10^{-5}$	$3.07 \cdot 10^{-2}$
$P_{0.93}$	ddpm	$3.66 \cdot 10^3$	$2.05 \cdot 10^{-2}$	$2.20 \cdot 10^{-5}$	$2.78 \cdot 10^{-2}$
$P_{0.94}$	csdi	$4.56 \cdot 10^3$	$1.70 \cdot 10^{-2}$	$3.53 \cdot 10^{-5}$	$3.20 \cdot 10^{-2}$
$P_{0.94}$	ddpm	$3.80 \cdot 10^3$	$1.31 \cdot 10^{-2}$	$2.52 \cdot 10^{-5}$	$2.96 \cdot 10^{-2}$
$P_{0.95}$	csdi	$4.39 \cdot 10^3$	$1.53 \cdot 10^{-2}$	$3.10 \cdot 10^{-5}$	$2.61 \cdot 10^{-2}$
$P_{0.95}$	ddpm	$3.79 \cdot 10^3$	$1.73 \cdot 10^{-2}$	$2.86 \cdot 10^{-5}$	$5.43 \cdot 10^{-2}$
$P_{0.96}$	csdi	$4.62 \cdot 10^3$	$1.85 \cdot 10^{-2}$	$3.51 \cdot 10^{-5}$	$2.51 \cdot 10^{-2}$
$P_{0.96}$	ddpm	$4.00 \cdot 10^3$	$2.17 \cdot 10^{-2}$	$2.80 \cdot 10^{-5}$	$3.95 \cdot 10^{-2}$
$P_{0.97}$	csdi	$4.26 \cdot 10^3$	$1.90 \cdot 10^{-2}$	$3.16 \cdot 10^{-5}$	$2.65 \cdot 10^{-2}$
$P_{0.97}$	ddpm	$3.97 \cdot 10^3$	$1.38 \cdot 10^{-2}$	$2.79 \cdot 10^{-5}$	$3.52 \cdot 10^{-2}$
$P_{0.98}$	csdi	$8.30 \cdot 10^1$	$9.13 \cdot 10^{-2}$	$1.11 \cdot 10^{+2}$	$1.94 \cdot 10^{+2}$
$P_{0.98}$	ddpm	$5.07 \cdot 10^3$	$2.62 \cdot 10^{-2}$	$2.75 \cdot 10^{-4}$	$3.70 \cdot 10^{-1}$
$P_{0.99}$	csdi	$3.92 \cdot 10^2$	$1.53 \cdot 10^{-1}$	$1.65 \cdot 10^{-1}$	$7.33 \cdot 10^{+0}$
$P_{0.99}$	ddpm	$4.98 \cdot 10^3$	$1.83 \cdot 10^{-2}$	$2.31 \cdot 10^{-4}$	$3.05 \cdot 10^{-1}$
$P_{1.00}$	csdi	$4.29 \cdot 10^2$	$6.27 \cdot 10^{-1}$	$3.01 \cdot 10^{-2}$	$2.87 \cdot 10^{+0}$
$P_{1.00}$	ddpm	$3.60 \cdot 10^2$	$1.14 \cdot 10^{+0}$	$3.71 \cdot 10^{-2}$	$4.51 \cdot 10^{+0}$
P_{GT}	csdi	$4.31 \cdot 10^3$	$1.58 \cdot 10^{-2}$	$3.11 \cdot 10^{-5}$	$3.01 \cdot 10^{-2}$
P_{GT}	ddpm	$4.07 \cdot 10^3$	$1.72 \cdot 10^{-2}$	$2.62 \cdot 10^{-5}$	$2.19 \cdot 10^{-2}$

Table A.2: Training with missing ratio of 20%.

Dataset	Model	MAE	MAPE	MSE	MXAE
$P_{0.70}$	csdi	$4.60 \cdot 10^{-3}$	$2.67 \cdot 10^{-2}$	$3.64 \cdot 10^{-5}$	$2.70 \cdot 10^{-2}$
$P_{0.70}$	ddpm	$1.57 \cdot 10^{-2}$	$3.43 \cdot 10^{-1}$	$1.89 \cdot 10^{-3}$	$5.21 \cdot 10^{-1}$
$P_{0.75}$	csdi	$4.71 \cdot 10^{-3}$	$3.70 \cdot 10^{-2}$	$3.74 \cdot 10^{-5}$	$3.12 \cdot 10^{-2}$
$P_{0.75}$	ddpm	$1.20 \cdot 10^{-2}$	$2.15 \cdot 10^{-1}$	$8.40 \cdot 10^{-4}$	$2.65 \cdot 10^{-1}$
$P_{0.80}$	csdi	$4.89 \cdot 10^{-3}$	$1.96 \cdot 10^{-2}$	$4.02 \cdot 10^{-5}$	$2.62 \cdot 10^{-2}$
$P_{0.80}$	ddpm	$7.18 \cdot 10^{-3}$	$5.17 \cdot 10^{-2}$	$1.91 \cdot 10^{-4}$	$1.35 \cdot 10^{-1}$
$P_{0.85}$	csdi	$4.65 \cdot 10^{-3}$	$1.76 \cdot 10^{-2}$	$3.60 \cdot 10^{-5}$	$3.50 \cdot 10^{-2}$
$P_{0.85}$	ddpm	$5.41 \cdot 10^{-3}$	$3.48 \cdot 10^{-2}$	$6.95 \cdot 10^{-5}$	$6.77 \cdot 10^{-2}$
$P_{0.90}$	csdi	$4.34 \cdot 10^{-3}$	$1.37 \cdot 10^{-2}$	$3.23 \cdot 10^{-5}$	$2.69 \cdot 10^{-2}$
$P_{0.90}$	ddpm	$4.12 \cdot 10^{-3}$	$3.93 \cdot 10^{-2}$	$3.41 \cdot 10^{-5}$	$5.77 \cdot 10^{-2}$
$P_{0.91}$	csdi	$4.13 \cdot 10^{-3}$	$1.66 \cdot 10^{-2}$	$2.95 \cdot 10^{-5}$	$3.49 \cdot 10^{-2}$
$P_{0.91}$	ddpm	$4.21 \cdot 10^{-3}$	$2.58 \cdot 10^{-2}$	$3.62 \cdot 10^{-5}$	$5.67 \cdot 10^{-2}$
$P_{0.92}$	csdi	$4.12 \cdot 10^{-3}$	$3.65 \cdot 10^{-2}$	$2.83 \cdot 10^{-5}$	$2.08 \cdot 10^{-2}$
$P_{0.92}$	ddpm	$3.91 \cdot 10^{-3}$	$4.89 \cdot 10^{-2}$	$2.86 \cdot 10^{-5}$	$5.47 \cdot 10^{-2}$
$P_{0.93}$	csdi	$3.70 \cdot 10^{-3}$	$1.25 \cdot 10^{-2}$	$2.22 \cdot 10^{-5}$	$2.66 \cdot 10^{-2}$
$P_{0.93}$	ddpm	$3.99 \cdot 10^{-3}$	$1.64 \cdot 10^{-2}$	$3.11 \cdot 10^{-5}$	$5.40 \cdot 10^{-2}$
$P_{0.94}$	csdi	$4.32 \cdot 10^{-3}$	$1.31 \cdot 10^{-2}$	$3.10 \cdot 10^{-5}$	$2.90 \cdot 10^{-2}$
$P_{0.94}$	ddpm	$3.87 \cdot 10^{-3}$	$1.55 \cdot 10^{-2}$	$2.63 \cdot 10^{-5}$	$4.38 \cdot 10^{-2}$
$P_{0.95}$	csdi	$3.88 \cdot 10^{-3}$	$1.30 \cdot 10^{-2}$	$2.78 \cdot 10^{-5}$	$3.02 \cdot 10^{-2}$
$P_{0.95}$	ddpm	$4.16 \cdot 10^{-3}$	$1.05 \cdot 10^{-1}$	$4.28 \cdot 10^{-5}$	$8.78 \cdot 10^{-2}$
$P_{0.96}$	csdi	$3.39 \cdot 10^{-3}$	$1.35 \cdot 10^{-2}$	$2.07 \cdot 10^{-5}$	$2.64 \cdot 10^{-2}$
$P_{0.96}$	ddpm	$3.73 \cdot 10^{-3}$	$1.34 \cdot 10^{-2}$	$2.87 \cdot 10^{-5}$	$6.32 \cdot 10^{-2}$
$P_{0.97}$	csdi	$3.40 \cdot 10^{-3}$	$1.27 \cdot 10^{-2}$	$2.07 \cdot 10^{-5}$	$3.73 \cdot 10^{-2}$
$P_{0.97}$	ddpm	$3.99 \cdot 10^{-3}$	$1.58 \cdot 10^{-2}$	$3.15 \cdot 10^{-5}$	$6.13 \cdot 10^{-2}$
$P_{0.98}$	csdi	$4.67 \cdot 10^{-2}$	$1.67 \cdot 10^{-1}$	$3.08 \cdot 10^{-1}$	$1.22 \cdot 10^{+1}$
$P_{0.98}$	ddpm	$4.64 \cdot 10^{-3}$	$2.02 \cdot 10^{-2}$	$1.71 \cdot 10^{-4}$	$2.85 \cdot 10^{-1}$
$P_{0.99}$	csdi	$1.66 \cdot 10^{-2}$	$1.79 \cdot 10^{-1}$	$2.43 \cdot 10^{-2}$	$3.93 \cdot 10^{+0}$
$P_{0.99}$	ddpm	$5.56 \cdot 10^{-3}$	$2.88 \cdot 10^{-2}$	$4.32 \cdot 10^{-4}$	$3.94 \cdot 10^{-1}$
$P_{1.00}$	csdi	$4.17 \cdot 10^{-2}$	$3.28 \cdot 10^{-1}$	$3.63 \cdot 10^{-2}$	$3.77 \cdot 10^{+0}$
$P_{1.00}$	ddpm	$3.32 \cdot 10^{-2}$	$1.31 \cdot 10^{-1}$	$2.60 \cdot 10^{-2}$	$3.66 \cdot 10^{+0}$
P_{GT}	csdi	$4.32 \cdot 10^{-3}$	$1.40 \cdot 10^{-2}$	$3.13 \cdot 10^{-5}$	$2.32 \cdot 10^{-2}$
P_{GT}	ddpm	$3.69 \cdot 10^{-3}$	$1.68 \cdot 10^{-2}$	$2.19 \cdot 10^{-5}$	$2.18 \cdot 10^{-2}$

Table A.3: Training with missing ratio of 30%.

Dataset	Model	MAE	MAPE	MSE	MXAE
$P_{0.70}$	csdi	$4.85 \cdot 10^{-3}$	$3.84 \cdot 10^{-2}$	$4.70 \cdot 10^{-5}$	$4.77 \cdot 10^{-2}$
$P_{0.70}$	ddpm	$2.01 \cdot 10^{-2}$	$2.04 \cdot 10^{-1}$	$2.22 \cdot 10^{-3}$	$4.06 \cdot 10^{-1}$
$P_{0.75}$	csdi	$3.98 \cdot 10^{-3}$	$4.05 \cdot 10^{-2}$	$2.70 \cdot 10^{-5}$	$2.52 \cdot 10^{-2}$
$P_{0.75}$	ddpm	$1.23 \cdot 10^{-2}$	$9.85 \cdot 10^{-2}$	$6.79 \cdot 10^{-4}$	$2.17 \cdot 10^{-1}$
$P_{0.80}$	csdi	$4.11 \cdot 10^{-3}$	$2.01 \cdot 10^{-2}$	$3.13 \cdot 10^{-5}$	$3.37 \cdot 10^{-2}$
$P_{0.80}$	ddpm	$1.10 \cdot 10^{-2}$	$1.02 \cdot 10^{-1}$	$6.89 \cdot 10^{-4}$	$3.86 \cdot 10^{-1}$
$P_{0.85}$	csdi	$3.73 \cdot 10^{-3}$	$2.18 \cdot 10^{-2}$	$2.34 \cdot 10^{-5}$	$2.36 \cdot 10^{-2}$
$P_{0.85}$	ddpm	$6.81 \cdot 10^{-3}$	$4.51 \cdot 10^{-2}$	$1.40 \cdot 10^{-4}$	$1.64 \cdot 10^{-1}$
$P_{0.90}$	csdi	$4.51 \cdot 10^{-3}$	$1.94 \cdot 10^{-2}$	$3.39 \cdot 10^{-5}$	$2.55 \cdot 10^{-2}$
$P_{0.90}$	ddpm	$5.44 \cdot 10^{-3}$	$3.70 \cdot 10^{-2}$	$7.38 \cdot 10^{-5}$	$1.19 \cdot 10^{-1}$
$P_{0.91}$	csdi	$4.14 \cdot 10^{-3}$	$1.49 \cdot 10^{-2}$	$3.11 \cdot 10^{-5}$	$2.40 \cdot 10^{-2}$
$P_{0.91}$	ddpm	$5.44 \cdot 10^{-3}$	$2.40 \cdot 10^{-2}$	$8.15 \cdot 10^{-5}$	$1.10 \cdot 10^{-1}$
$P_{0.92}$	csdi	$3.62 \cdot 10^{-3}$	$1.20 \cdot 10^{-2}$	$2.28 \cdot 10^{-5}$	$2.82 \cdot 10^{-2}$
$P_{0.92}$	ddpm	$5.04 \cdot 10^{-3}$	$3.41 \cdot 10^{-2}$	$6.21 \cdot 10^{-5}$	$9.73 \cdot 10^{-2}$
$P_{0.93}$	csdi	$3.92 \cdot 10^{-3}$	$1.25 \cdot 10^{-2}$	$2.70 \cdot 10^{-5}$	$2.75 \cdot 10^{-2}$
$P_{0.93}$	ddpm	$4.73 \cdot 10^{-3}$	$1.83 \cdot 10^{-2}$	$5.37 \cdot 10^{-5}$	$1.08 \cdot 10^{-1}$
$P_{0.94}$	csdi	$3.79 \cdot 10^{-3}$	$1.53 \cdot 10^{-2}$	$2.49 \cdot 10^{-5}$	$2.97 \cdot 10^{-2}$
$P_{0.94}$	ddpm	$4.81 \cdot 10^{-3}$	$2.94 \cdot 10^{-2}$	$6.37 \cdot 10^{-5}$	$1.06 \cdot 10^{-1}$
$P_{0.95}$	csdi	$3.43 \cdot 10^{-3}$	$1.07 \cdot 10^{-2}$	$2.03 \cdot 10^{-5}$	$3.10 \cdot 10^{-2}$
$P_{0.95}$	ddpm	$4.74 \cdot 10^{-3}$	$2.51 \cdot 10^{-2}$	$4.23 \cdot 10^{-5}$	$5.65 \cdot 10^{-2}$
$P_{0.96}$	csdi	$3.36 \cdot 10^{-3}$	$1.19 \cdot 10^{-2}$	$1.91 \cdot 10^{-5}$	$2.67 \cdot 10^{-2}$
$P_{0.96}$	ddpm	$4.20 \cdot 10^{-3}$	$1.72 \cdot 10^{-2}$	$3.66 \cdot 10^{-5}$	$7.12 \cdot 10^{-2}$
$P_{0.97}$	csdi	$3.71 \cdot 10^{-3}$	$1.37 \cdot 10^{-2}$	$2.50 \cdot 10^{-5}$	$3.52 \cdot 10^{-2}$
$P_{0.97}$	ddpm	$4.21 \cdot 10^{-3}$	$1.95 \cdot 10^{-2}$	$3.81 \cdot 10^{-5}$	$7.50 \cdot 10^{-2}$
$P_{0.98}$	csdi	$5.24 \cdot 10^{-2}$	$3.40 \cdot 10^{-1}$	$3.51 \cdot 10^{-1}$	$1.51 \cdot 10^{+1}$
$P_{0.98}$	ddpm	$3.01 \cdot 10^{-1}$	$1.01 \cdot 10^{-1}$	$9.83 \cdot 10^{+0}$	$5.54 \cdot 10^{+1}$
$P_{0.99}$	csdi	$7.87 \cdot 10^{-3}$	$1.05 \cdot 10^{-1}$	$1.57 \cdot 10^{-3}$	$7.01 \cdot 10^{-1}$
$P_{0.99}$	ddpm	$4.51 \cdot 10^{-2}$	$1.00 \cdot 10^{-1}$	$1.68 \cdot 10^{-1}$	$9.09 \cdot 10^{+0}$
$P_{1.00}$	csdi	$4.32 \cdot 10^{-2}$	$4.72 \cdot 10^{-1}$	$5.42 \cdot 10^{-2}$	$5.40 \cdot 10^{+0}$
$P_{1.00}$	ddpm	$3.21 \cdot 10^{-2}$	$2.03 \cdot 10^{-1}$	$2.47 \cdot 10^{-2}$	$3.55 \cdot 10^{+0}$
P_{GT}	csdi	$3.48 \cdot 10^{-3}$	$3.08 \cdot 10^{-2}$	$2.02 \cdot 10^{-5}$	$2.44 \cdot 10^{-2}$
P_{GT}	ddpm	$4.22 \cdot 10^{-3}$	$6.94 \cdot 10^{-2}$	$3.03 \cdot 10^{-5}$	$3.76 \cdot 10^{-2}$

Table A.4: Training with missing ratio of 40%.

Dataset	Model	MAE	MAPE	MSE	MXAE
$P_{0.70}$	csdi	$4.78 \cdot 10^{-3}$	$9.55 \cdot 10^{-2}$	$4.45 \cdot 10^{-5}$	$3.28 \cdot 10^{-2}$
$P_{0.70}$	ddpm	$2.81 \cdot 10^{-2}$	$2.40 \cdot 10^{-1}$	$3.31 \cdot 10^{-3}$	$4.80 \cdot 10^{-1}$
$P_{0.75}$	csdi	$4.77 \cdot 10^{-3}$	$2.93 \cdot 10^{-2}$	$4.19 \cdot 10^{-5}$	$3.18 \cdot 10^{-2}$
$P_{0.75}$	ddpm	$2.02 \cdot 10^{-2}$	$4.94 \cdot 10^{-1}$	$1.64 \cdot 10^{-3}$	$3.66 \cdot 10^{-1}$
$P_{0.80}$	csdi	$4.33 \cdot 10^{-3}$	$2.01 \cdot 10^{-2}$	$3.49 \cdot 10^{-5}$	$3.07 \cdot 10^{-2}$
$P_{0.80}$	ddpm	$1.43 \cdot 10^{-2}$	$9.67 \cdot 10^{-2}$	$7.76 \cdot 10^{-4}$	$3.56 \cdot 10^{-1}$
$P_{0.85}$	csdi	$4.28 \cdot 10^{-3}$	$1.70 \cdot 10^{-2}$	$3.07 \cdot 10^{-5}$	$2.41 \cdot 10^{-2}$
$P_{0.85}$	ddpm	$9.11 \cdot 10^{-3}$	$6.59 \cdot 10^{-2}$	$2.01 \cdot 10^{-4}$	$1.41 \cdot 10^{-1}$
$P_{0.90}$	csdi	$3.61 \cdot 10^{-3}$	$2.98 \cdot 10^{-2}$	$2.30 \cdot 10^{-5}$	$2.37 \cdot 10^{-2}$
$P_{0.90}$	ddpm	$8.51 \cdot 10^{-3}$	$4.84 \cdot 10^{-2}$	$1.97 \cdot 10^{-4}$	$1.78 \cdot 10^{-1}$
$P_{0.91}$	csdi	$3.73 \cdot 10^{-3}$	$1.40 \cdot 10^{-2}$	$2.47 \cdot 10^{-5}$	$2.62 \cdot 10^{-2}$
$P_{0.91}$	ddpm	$7.44 \cdot 10^{-3}$	$4.25 \cdot 10^{-2}$	$1.94 \cdot 10^{-4}$	$2.13 \cdot 10^{-1}$
$P_{0.92}$	csdi	$3.43 \cdot 10^{-3}$	$1.48 \cdot 10^{-2}$	$1.98 \cdot 10^{-5}$	$2.17 \cdot 10^{-2}$
$P_{0.92}$	ddpm	$6.74 \cdot 10^{-3}$	$4.30 \cdot 10^{-2}$	$1.51 \cdot 10^{-4}$	$2.24 \cdot 10^{-1}$
$P_{0.93}$	csdi	$3.87 \cdot 10^{-3}$	$1.46 \cdot 10^{-2}$	$2.55 \cdot 10^{-5}$	$2.31 \cdot 10^{-2}$
$P_{0.93}$	ddpm	$6.62 \cdot 10^{-3}$	$3.27 \cdot 10^{-2}$	$1.03 \cdot 10^{-4}$	$1.10 \cdot 10^{-1}$
$P_{0.94}$	csdi	$3.55 \cdot 10^{-3}$	$1.13 \cdot 10^{-2}$	$2.20 \cdot 10^{-5}$	$2.66 \cdot 10^{-2}$
$P_{0.94}$	ddpm	$6.23 \cdot 10^{-3}$	$2.20 \cdot 10^{-2}$	$1.02 \cdot 10^{-4}$	$1.15 \cdot 10^{-1}$
$P_{0.95}$	csdi	$3.80 \cdot 10^{-3}$	$1.77 \cdot 10^{-2}$	$2.41 \cdot 10^{-5}$	$2.83 \cdot 10^{-2}$
$P_{0.95}$	ddpm	$6.70 \cdot 10^{-3}$	$2.56 \cdot 10^{-2}$	$1.22 \cdot 10^{-4}$	$1.39 \cdot 10^{-1}$
$P_{0.96}$	csdi	$3.43 \cdot 10^{-3}$	$1.18 \cdot 10^{-2}$	$2.02 \cdot 10^{-5}$	$2.57 \cdot 10^{-2}$
$P_{0.96}$	ddpm	$5.75 \cdot 10^{-3}$	$1.85 \cdot 10^{-2}$	$6.82 \cdot 10^{-5}$	$7.44 \cdot 10^{-2}$
$P_{0.97}$	csdi	$3.46 \cdot 10^{-3}$	$1.40 \cdot 10^{-2}$	$2.09 \cdot 10^{-5}$	$3.24 \cdot 10^{-2}$
$P_{0.97}$	ddpm	$5.74 \cdot 10^{-3}$	$2.26 \cdot 10^{-2}$	$7.12 \cdot 10^{-5}$	$9.69 \cdot 10^{-2}$
$P_{0.98}$	csdi	$9.62 \cdot 10^{-3}$	$1.21 \cdot 10^{-1}$	$3.89 \cdot 10^{-3}$	$1.33 \cdot 10^{+0}$
$P_{0.98}$	ddpm	$6.42 \cdot 10^{-3}$	$1.95 \cdot 10^{-2}$	$1.45 \cdot 10^{-4}$	$1.69 \cdot 10^{-1}$
$P_{0.99}$	csdi	$6.97 \cdot 10^{-2}$	$9.52 \cdot 10^{-2}$	$6.84 \cdot 10^{-1}$	$1.73 \cdot 10^{+1}$
$P_{0.99}$	ddpm	$6.75 \cdot 10^{-3}$	$2.36 \cdot 10^{-2}$	$2.75 \cdot 10^{-4}$	$3.32 \cdot 10^{-1}$
$P_{1.00}$	csdi	$3.80 \cdot 10^{-2}$	$1.65 \cdot 10^{-1}$	$3.46 \cdot 10^{-2}$	$4.06 \cdot 10^{+0}$
$P_{1.00}$	ddpm	$2.40 \cdot 10^{-2}$	$8.85 \cdot 10^{-2}$	$1.13 \cdot 10^{-2}$	$2.05 \cdot 10^{+0}$
P_{GT}	csdi	$3.18 \cdot 10^{-3}$	$1.67 \cdot 10^{-2}$	$1.67 \cdot 10^{-5}$	$2.22 \cdot 10^{-2}$
P_{GT}	ddpm	$6.58 \cdot 10^{-3}$	$4.23 \cdot 10^{-2}$	$7.30 \cdot 10^{-5}$	$5.85 \cdot 10^{-2}$

Table A.5: Training with missing ratio of 50%.

Dataset	Model	MAE	MAPE	MSE	MXAE
$P_{0.70}$	csdi	$5.45 \cdot 10^{-3}$	$2.91 \cdot 10^{-2}$	$5.50 \cdot 10^{-5}$	$4.39 \cdot 10^{-2}$
$P_{0.70}$	ddpm	$6.76 \cdot 10^{-2}$	$1.29 \cdot 10^{+0}$	$1.69 \cdot 10^{-2}$	$8.06 \cdot 10^{-1}$
$P_{0.75}$	csdi	$5.15 \cdot 10^{-3}$	$3.92 \cdot 10^{-2}$	$4.83 \cdot 10^{-5}$	$3.21 \cdot 10^{-2}$
$P_{0.75}$	ddpm	$4.79 \cdot 10^{-2}$	$6.44 \cdot 10^{-1}$	$8.94 \cdot 10^{-3}$	$6.32 \cdot 10^{-1}$
$P_{0.80}$	csdi	$4.20 \cdot 10^{-3}$	$1.87 \cdot 10^{-2}$	$3.16 \cdot 10^{-5}$	$3.39 \cdot 10^{-2}$
$P_{0.80}$	ddpm	$3.21 \cdot 10^{-2}$	$3.10 \cdot 10^{+0}$	$3.93 \cdot 10^{-3}$	$4.82 \cdot 10^{-1}$
$P_{0.85}$	csdi	$4.34 \cdot 10^{-3}$	$2.11 \cdot 10^{-2}$	$3.28 \cdot 10^{-5}$	$2.66 \cdot 10^{-2}$
$P_{0.85}$	ddpm	$1.94 \cdot 10^{-2}$	$1.59 \cdot 10^{-1}$	$1.18 \cdot 10^{-3}$	$4.16 \cdot 10^{-1}$
$P_{0.90}$	csdi	$3.66 \cdot 10^{-3}$	$1.32 \cdot 10^{-2}$	$2.24 \cdot 10^{-5}$	$2.73 \cdot 10^{-2}$
$P_{0.90}$	ddpm	$1.49 \cdot 10^{-2}$	$1.55 \cdot 10^{-1}$	$5.74 \cdot 10^{-4}$	$1.76 \cdot 10^{-1}$
$P_{0.91}$	csdi	$4.44 \cdot 10^{-3}$	$2.90 \cdot 10^{-2}$	$3.39 \cdot 10^{-5}$	$3.41 \cdot 10^{-2}$
$P_{0.91}$	ddpm	$1.22 \cdot 10^{-2}$	$4.88 \cdot 10^{-2}$	$3.48 \cdot 10^{-4}$	$2.10 \cdot 10^{-1}$
$P_{0.92}$	csdi	$3.40 \cdot 10^{-3}$	$1.46 \cdot 10^{-2}$	$1.92 \cdot 10^{-5}$	$3.22 \cdot 10^{-2}$
$P_{0.92}$	ddpm	$1.12 \cdot 10^{-2}$	$4.24 \cdot 10^{-2}$	$3.70 \cdot 10^{-4}$	$2.70 \cdot 10^{-1}$
$P_{0.93}$	csdi	$3.82 \cdot 10^{-3}$	$1.16 \cdot 10^{-2}$	$2.49 \cdot 10^{-5}$	$2.55 \cdot 10^{-2}$
$P_{0.93}$	ddpm	$1.19 \cdot 10^{-2}$	$4.52 \cdot 10^{-2}$	$3.88 \cdot 10^{-4}$	$2.74 \cdot 10^{-1}$
$P_{0.94}$	csdi	$4.04 \cdot 10^{-3}$	$1.54 \cdot 10^{-2}$	$2.83 \cdot 10^{-5}$	$2.68 \cdot 10^{-2}$
$P_{0.94}$	ddpm	$1.09 \cdot 10^{-2}$	$3.92 \cdot 10^{-2}$	$2.51 \cdot 10^{-4}$	$1.33 \cdot 10^{-1}$
$P_{0.95}$	csdi	$3.48 \cdot 10^{-3}$	$1.21 \cdot 10^{-2}$	$2.15 \cdot 10^{-5}$	$3.75 \cdot 10^{-2}$
$P_{0.95}$	ddpm	$1.08 \cdot 10^{-2}$	$7.26 \cdot 10^{-2}$	$3.28 \cdot 10^{-4}$	$2.67 \cdot 10^{-1}$
$P_{0.96}$	csdi	$3.51 \cdot 10^{-3}$	$1.76 \cdot 10^{-2}$	$2.11 \cdot 10^{-5}$	$2.23 \cdot 10^{-2}$
$P_{0.96}$	ddpm	$9.84 \cdot 10^{-3}$	$3.66 \cdot 10^{-2}$	$2.55 \cdot 10^{-4}$	$1.89 \cdot 10^{-1}$
$P_{0.97}$	csdi	$4.17 \cdot 10^{-3}$	$1.79 \cdot 10^{-2}$	$5.38 \cdot 10^{-5}$	$1.35 \cdot 10^{-1}$
$P_{0.97}$	ddpm	$2.08 \cdot 10^{-1}$	$8.47 \cdot 10^{-2}$	$5.98 \cdot 10^{+0}$	$5.54 \cdot 10^{+1}$
$P_{0.98}$	csdi	$1.90 \cdot 10^{-2}$	$1.38 \cdot 10^{-1}$	$3.37 \cdot 10^{-2}$	$4.45 \cdot 10^{+0}$
$P_{0.98}$	ddpm	$9.61 \cdot 10^{-3}$	$3.12 \cdot 10^{-2}$	$2.97 \cdot 10^{-4}$	$2.77 \cdot 10^{-1}$
$P_{0.99}$	csdi	$5.13 \cdot 10^{-1}$	$1.10 \cdot 10^{-1}$	$4.54 \cdot 10^{+1}$	$1.25 \cdot 10^{+2}$
$P_{0.99}$	ddpm	$9.58 \cdot 10^{-3}$	$5.11 \cdot 10^{-2}$	$4.75 \cdot 10^{-4}$	$4.05 \cdot 10^{-1}$
$P_{1.00}$	csdi	$3.99 \cdot 10^{-2}$	$2.70 \cdot 10^{-1}$	$2.62 \cdot 10^{-2}$	$2.98 \cdot 10^{+0}$
$P_{1.00}$	ddpm	$2.45 \cdot 10^{-2}$	$8.68 \cdot 10^{-2}$	$1.07 \cdot 10^{-2}$	$2.24 \cdot 10^{+0}$
P_{GT}	csdi	$3.40 \cdot 10^{-3}$	$1.42 \cdot 10^{-2}$	$1.94 \cdot 10^{-5}$	$2.38 \cdot 10^{-2}$
P_{GT}	ddpm	$8.34 \cdot 10^{-3}$	$3.96 \cdot 10^{-2}$	$1.13 \cdot 10^{-4}$	$5.17 \cdot 10^{-2}$

Appendix B

DiffPaC implementation

The [Conditional Score Based Diffusion \(CSDI\)](#) model used to implement Algorithm 2 is virtually unchanged from that provided by its authors at <https://github.com/ermongroup/CSDI>. [Diffusion Partition Consensus \(DiffPaC\)](#) implements an extension that operates after each epoch of training to generate, maintain, and modify partition vectors P, I , and F (as described in Section 8.5), sample X_D , and perform updates as per Algorithm 2. All [CSDI](#) hyperparameters are set to the defaults given by the original authors, except *eval_length* which is set to 501 for our dataset and percentage ground truth, which is set to 50%.

[Savitzky-Golay Filter \(SGF\)](#) M_α and M_β are implemented using *savgol_filter* and *interp1d* from *SciPy*[139]. The M_α parameters are window size 600 and order 5, The M_β parameters are window size 100 and order 8. Practically speaking the larger the window size, the lower the order, the more pronounced the smoothing effect of the filter. [SGF](#) model nominal UT data reasonably well and, as in Figure 8.2, parameter selection strongly influences sensitivity to the presence of anomalous data. Thus SGF models M_α and M_β are parameterized such that they converge when fitting only nominal data $\sim p(\mathcal{F}_n)$ and diverge discordantly when fitting data with outliers. *interp1d* with default parameters is used to fill in missing partitioned values as required by *savgol_filter*.

Appendix C

Synthetic data gen

The following code randomly generates a set of synthetic curves which may be used to test and verify [Diffusion Partition Consensus \(DiffPaC\)](#) performance.

```
1  import numpy as np
2  import torch
3  from scipy.signal import savgol_filter
4  from scipy.interpolate import interp1d
5
6  # ===== Utility Functions =====
7
8  def get_observed_curve(length, rate, inf=0, lo=1, hi=2, o_ride=False)
9  :
10     """
11     Generates an observed curve based on frequency modulation.
12
13     Parameters:
14         length (int): Length of the curve.
15         rate (int): Sampling rate.
16         inf (int): Number of inflection points.
17         lo (float): Minimum frequency.
18         hi (float): Maximum frequency.
19         o_ride (bool): Override interpolation type.
20
21     Returns:
22         tuple: Index array, signal array, and frequency curve.
23     """
24     freq_curve = 2 * np.pi * get_inflection_curve(length, rate, inf=
inf, lo=lo, hi=hi, o_ride=o_ride)
    x = np.arange(0, length, 1 / rate)
```

```

25     y = np.sin((freq_curve * (1 / rate)).cumsum()) # Compute
cumulative sum once
26     return np.arange(len(x)), y, freq_curve
27
28
29 def get_inflection_curve(length, rate, inf=0, lo=10, hi=20, o_ride=
False):
30     """
31     Creates an inflection curve with optional customization.
32
33     Parameters:
34         length (int): Length of the curve.
35         rate (int): Sampling rate.
36         inf (int): Number of inflection points.
37         lo (float): Minimum value.
38         hi (float): Maximum value.
39         o_ride (bool): Override interpolation type.
40
41     Returns:
42         np.ndarray: Inflection curve.
43     """
44     x = np.linspace(0, length, int(length * rate))
45     l = len(x)
46
47     if inf > 0:
48         inflection_points = np.sort(np.random.randint(0, l, inf + 2))
49         inflection_values = np.random.uniform(lo, hi, inf + 2)
50         kind = "quadratic" if not o_ride else "linear"
51     else:
52         inflection_points = np.array([0, l])
53         inflection_values = np.array([lo, hi])
54         kind = "linear"
55
56     interpolation = interp1d(inflection_points, inflection_values,
kind=kind, fill_value="extrapolate")
57     return interpolation(np.arange(len(x)))
58
59
60 def generate_flaws(ground_truth, flaw_rate, flaw_height, flaw_width):
61     """
62     Introduces random flaws into the ground truth signal.
63
64     Parameters:
65         ground_truth (np.ndarray): Original signal.
66         flaw_rate (float): Rate of flaws per unit length.

```

```

67         flaw_height (int): Maximum height of flaws.
68         flaw_width (int): Maximum width of flaws.
69
70     Returns:
71         np.ndarray: Signal with flaws.
72     """
73     flaw_count = int(len(ground_truth) * flaw_rate)
74     flaw_idx = np.random.randint(0, len(ground_truth), flaw_count)
75     heights = np.random.randint(0, flaw_height, flaw_count)
76     widths = np.random.randint(0, flaw_width, flaw_count)
77
78     observed = ground_truth.copy()
79     for idx, height, width in zip(flaw_idx, heights, widths):
80         observed[idx:idx + width] += height
81
82     return observed
83
84     # ===== Simulation Parameters
85     =====
86
87     inflections = [0, 2, 4, 8, 16]
88     frates = [0, 2, 4, 8, 16]
89     snrs = [1000]
90
91     length = 20
92     sample_rate = 3600
93
94     # Frequency, amplitude, and DC parameters
95     param_ranges = {
96         "freq": (0, 1, 5),
97         "amp": (0, 47, 403),
98         "dc": (0, 1253, 1817)
99     }
100
101     # ===== Simulation Loop =====
102     for inflect in inflections:
103         f_inf, f_lo, f_hi = param_ranges["freq"]
104         a_inf, a_lo, a_hi = inflect, *param_ranges["amp"][1:]
105         dc_inf, dc_lo, dc_hi = inflect, *param_ranges["dc"][1:]
106
107         # Generate curves
108         x, y, freq_curve = get_observed_curve(length, sample_rate, inf=
f_inf, lo=f_lo, hi=f_hi, o_ride=True)
109         amp_curve = get_inflection_curve(length, sample_rate, inf=a_inf,
lo=a_lo, hi=a_hi, o_ride=True)

```

```

109     dc_curve = get_inflection_curve(length, sample_rate, inf=dc_inf,
110                                     lo=dc_lo, hi=dc_hi, o_ride=True)
111
112     # Generate ground truth signal
113     ground_truth = savgol_filter(y * amp_curve + dc_curve, 600, 5)
114
115     for snr in snrs:
116         for frate in frates:
117             # Generate observed data with flaws
118             f_rate = frate * 0.01 / (200 / 2) # Simplified flaw rate
119             calculation
120             observed = generate_flaws(ground_truth, f_rate, 800, 200)
121
122             # Add noise based on SNR
123             noise = np.random.normal(0, np.sqrt(np.mean(ground_truth
124                                                         **2)) / np.sqrt(snr), observed.shape) if snr != 0 else np.zeros_like(
125                 observed)
126             noisy = observed + noise
127
128             # Convert to tensors for further processing
129             observed_data, gt_data, index = map(torch.tensor, [noisy,
130                                                         ground_truth, x])
131
132             # Placeholder for additional processing
133             pass

```

Listing C.1: Code used to generate the synthetic datasets.

Glossary

Autoencoder (AE): An autoencoder is a neural network that learns to encode and decode data to and from a learned lower dimensional representation. It is an unsupervised learning model used for tasks such as dimension reduction, feature extraction, embedding, denoising, and data compression. Variants include Denoising Autoencoders, Sparse Autoencoders, and Variational Autoencoders (VAEs). [32](#), [36](#), [37](#)

Beta Variational Autoencoder (BVAE): A Beta-VAE is a type of generative model that extends the standard Variational Autoencoder (VAE) by introducing a hyperparameter β in the loss function to scale the Kullback–Leibler (KL) divergence term. This encourages the model to learn disentangled and interpretable latent representations. The ability to modulate loss term contribution provides latent regularization that can help reveal underlying factors of variation in the data. [32](#)

Calandria Tube (CT): The CT is a zirconium alloy tube that spans the CANDU reactor main pressure or Calandria vessel. CT are slightly wider in diameter than, and contain, CANDU PT. Four garter springs lie between the CT and PT to prevent tube to tube contact. [16](#), [19](#), [29](#)

Calibration Fixture (CF): CF are facilities containing small sections of zirconium alloy PT material with inscribed notches and slots. These are used to calibrate the response of UT NDE inspection tool probes. [25](#), [39](#), [41](#), [44](#), [46](#), [50](#), [55](#), [57](#), [72](#), [86](#), [95](#)

Canada Deuterium Uranium (CANDU): The CANDU Reactor is a Canadian nuclear reactor design that uses natural uranium as fuel and heavy water (deuterium oxide) as a coolant and a neutron moderator. By avoiding the use of enriched uranium fuel, this reactor design reduces fuel processing costs and decreases the risk of nuclear weapons proliferation. [1–3](#), [5](#), [6](#), [8](#), [11–14](#), [20](#), [25](#), [28](#), [31](#), [46](#), [55](#), [72](#), [74](#), [75](#), [95](#)

CANDU Inspection Qualification Bureau (CIQB): The CIQB was established by the CANDU Owners Group (COG) and administers the inspection qualification process for CANDU nuclear reactors. It ensures that inspection procedures, techniques, methods, and personnel meet stringent standards, and supports the safe operation of these reactors. [94](#)

Conditional Score Based Diffusion (CSDI): CSDI is a generative modeling technique based on a conditional, score matching, diffusion process. It learns to reverse a noising process conditional on observed data to produce estimates of observed and unobserved or, missing data. [10, 33, 37, 38, 57, 62, 63, 65, 66, 68, 71–74, 82, 87, 90, 95, 99, 100, 124](#)

Conditional Variational Autoencoder (CVAE): A CVAE is a type of VAE that learns a structured latent space by incorporating conditioning information during both encoding and decoding. This allows for controlled data generation, where outputs are influenced by the conditioning attributes. CVAEs ensure the learned latent space is more interpretable by aligning it with meaningful conditioning information, and have found application in guided image synthesis, anomaly detection, and missing data imputation. [5, 32, 33, 35](#)

Convolutional Neural Network (CNN): A CNN is a type of deep learning model designed to process structured data, such as images, using neural network layers to detect spatial hierarchies of features such as edges, textures, and objects. CNN consist of multiple layers, including convolutional, pooling, normalizing, and fully connected layers, which allow to learn and extract relevant patterns for tasks like image classification, object detection, and segmentation. [5, 31](#)

Delayed Hydride Cracking (DHC): DHC is a time-dependent fracture mechanism that occurs in metals, particularly zirconium alloys, due to the diffusion and precipitation of brittle hydrogen hydrides. When subjected to stress, these hydrides form and grow at crack tips, increasing the risk of progressive crack propagation over time. DHC is a major concern in CANDU nuclear reactor components, where hydrogen absorption from coolant interactions can compromise material integrity and lead to catastrophic failure. [20, 26, 29](#)

Denoising Diffusion Probabilistic Model (DDPM): DDPMs are generative models that learn to create data by gradually reversing a noise corruption process through a series of denoising steps. During training, data is progressively perturbed with Gaussian noise, and the model learns to reconstruct the original data by predicting

and removing this noise. DDPMs have gained popularity for generating high-quality images, surpassing GANs in fidelity and diversity while being more stable to train. [33, 37, 38, 57, 62, 63, 65, 66, 71, 72](#)

Density Based Spatial Clustering of Applications with Noise (DBSCAN): DBSCAN is an unsupervised clustering algorithm that groups data points based on density, identifying clusters of varying shapes and sizes while marking outliers as noise. It relies on parameters ϵ , the neighborhood radius, and minPts, the minimum number of points required to form a dense region. DBSCAN does not require specifying the number of clusters in advance and is effective at discovering arbitrarily shaped clusters in noisy data. [31, 36, 50](#)

Diffusion Partition Consensus (DiffPaC): DiffPaC integrates CSDI and SGF to iteratively learn the nominal trend in a UT dataset and provide a binary classification over the data. [4, 6, 7, 33, 37, 38, 41, 68, 72–75, 77, 79, 81, 82, 86, 87, 90, 91, 95, 96, 98–101, 124, 125](#)

Diffusion Probabilistic Model (DPM): DPM refers to the broad class of Diffusion based probabilistic models. Generally these models define a forward noising and reverse denoising diffusion process. During training, the model learns to reverse incremental noising process, to bring sampled noise a target distribution to a learned approximate data distribution. [32, 57, 68](#)

Evidence Lower Bound (ELBO): The ELBO is an objective function used in Variational Inference to approximate intractable posterior distributions in Bayesian models. It is derived by decomposing the log marginal likelihood as the ELBO and the Kullback-Leibler (KL) divergence between the true posterior and the variational approximation. Maximizing the ELBO minimizes the divergence and ensures the variational distribution closely approximates the true posterior while providing a tractable optimization target. [47](#)

Fuel Channel Assembly (FCA): A CANDU fuel channel assembly is a key component of CANDU nuclear reactors, designed to hold and facilitate the flow of heavy water coolant around natural uranium fuel bundles. Each fuel channel assembly consists of a PT, which contains the fuel bundles, and is surrounded by a CT, allowing moderator circulation within the reactor core. This design enables on-power refueling and supports an efficient neutron economy required for ongoing reactor operation. [14, 16, 29](#)

High Stress Area (HSA): The high-stress areas in CANDU PT are located at their inlet and outlet ends in and around the rolled joint between the PT and end fitting, where the tube is mechanically constrained by spacers and bearings. These regions experience high thermal and mechanical stresses due to temperature and pressure gradients, from the heavy water coolant, and irradiation-induced material degradation. Over time, these stresses contribute to creep, hydrogen absorption, and potential deformation, and require careful monitoring to ensure the integrity and safe operation of the reactor. [39, 41, 44, 86, 87](#)

Inner Diameter (ID): In this context, the ID surface interface of a pipe such as a PT or CF. [1, 2, 11, 12, 16, 19, 20, 22, 23, 25, 26, 39, 41, 42, 73, 74](#)

Maximum Absolute Error (MXAE): MXAE is a performance metric that measures the largest absolute difference between predicted and actual values in a dataset. It is computed by taking the absolute differences for each prediction and then selecting the maximum value, thereby indicating the worst-case error. This metric is useful in applications where the largest deviation from the true value is critical for assessing model performance. [86, 87](#)

Mean Average Error (MAE): MAE is a metric used to measure the average absolute difference between predicted and actual values in a dataset. It is calculated by taking the sum of absolute errors and dividing it by the total number of observations. Unlike MSE, MAE is linear, and does not disproportionately penalize larger errors, causing it to be more robust to outliers in certain applications. [66](#)

Mean Squared Error (MSE): Mean Squared Error (MSE) is a loss function that measures the average of the squared differences between predicted and actual values. It is calculated by summing the squared errors for each prediction, then dividing by the number of observations, making it sensitive to large errors due to the squaring operation. Because of this property, MSE may be used when penalizing larger errors more heavily is desirable, such as in regression problems where extreme deviations must be minimized. [45, 46, 55, 65, 66, 68, 86, 87](#)

Nominal Data Model (NDM): The NDM models nominal data on the basis of that which is considered, or classified, as nominal:

$$\hat{X} \sim p(\hat{X}|X[P]) \tag{C.1}$$

[66, 71](#)

Nominal Profiling (NPROF): NPROF is a technique that characterizes the highly likely nominal component of a dataset and identifies where data exceeds the nominal characteristic. 4, 6, 7, 45, 49, 50, 55, 56, 91, 95, 96, 98

Non-Destructive Evaluation (NDE): NDE is a set of techniques used to inspect and assess materials, components, or structures for defects without causing damage. It includes methods such as ultrasonic testing, radiographic testing, and magnetic particle inspection, which help detect cracks, voids, or material degradation. NDE is widely used in industries like aerospace, nuclear, and manufacturing to ensure safety, reliability, and compliance with quality standards. 1–6, 8, 9, 12, 13, 20, 26, 27, 46, 55, 58, 71–75, 95, 96

Normal Beam (NB): In UT NDE, a NB refers to a testing technique where sound waves propagate perpendicularly (normal) to the surface of a material to detect internal flaws or measure thickness. A transducer generates high-frequency sound waves, which travel through the material and reflect back from interfaces or defects, to support the NDE function. 1, 11–13, 22, 23, 27, 29, 41, 42, 47, 74, 92

Observed Data Model (ODM): The ODM model data on the basis of all observed data:

$$\hat{X} \sim p(\hat{X}|X) \quad (\text{C.2})$$

66, 71

Outer Diameter (OD): In this context, the OD surface interface of a pipe such as a PT or CF. 19, 22

Pitch Catch (PC): UT PC is a NDE technique where one transducer (the "pitch") sends an UT wave through a material, and another transducer (the "catch") receives the transmitted or reflected wave. The time-of-flight and amplitude of the received signal provide information about the material's integrity, detecting defects such as cracks, voids, and delaminations. 1, 13, 22, 23, 27, 29

Pressure Tube (PT): CANDU PT are key components of the CANDU reactor core. They house the nuclear fuel bundles and allows heavy water coolant to flow through it under high pressure. Made of a zirconium alloy, they are designed to withstand intense radiation, high temperatures, and mechanical stresses while minimizing neutron absorption. The PT are located inside the CTs in the FCA and are about 4 m in length, 10 cm in diameter, and 5 mm thick. 1–3, 6, 8, 11, 13, 16, 19–22, 25–29, 31, 39, 41, 42, 44, 46, 60, 72–75, 87, 91, 95

Pressurized Heavy Water Transport (PHT): The PHT system in a CANDU nuclear reactor, circulates heavy water D_2O under high pressure to remove the heat of nuclear fission from the fuel channels. The system maintains heavy water in a liquid state at high temperatures and pressure, and transfers the absorbed heat from the reactor core to steam generators, where it produces steam to drive turbines for electricity generation. By using heavy water as both a coolant and a neutron moderator, this system enhances neutron economy, allowing the reactor to operate efficiently with natural uranium fuel. [16](#), [19](#)

Pulse Echo (PE): PE is an UT NDE technique where a single transducer emits high-frequency sound waves into a material and then detects the echoes reflected from internal defects or the material's boundaries. The time delay and amplitude of these echoes provide information about the location, size, and nature of flaws such as cracks, voids, or inclusions. [1](#), [22](#)

Savitzky-Golay Filter (SGF): An SFG is a digital smoothing filter that fits successive local windows of data with a polynomial using least squares regression, preserving trends in the data while reducing noise. Unlike simple moving averages, it maintains the shape and features of the signal, making it useful for applications requiring signal differentiation or retention of peak details. It is widely used in signal processing, and time-series analysis where data smoothing without significant distortion is desired. [42](#), [46](#), [49](#), [68](#), [72](#), [73](#), [75](#), [79](#), [82](#), [90](#), [95](#), [99](#), [124](#)

Shear Wave (SW): SW are generated in test subject material when UT signal interfaces with the material at an off normal alignment to the material surface. Shear induces particle motion perpendicular to the direction of wave propagation, making it sensitive to defects oriented perpendicularly to the wave path. Shear waves can detect flaws that are not easily identified with NB methods. [1](#), [22](#), [23](#)

Ultrasound (UT): UT is sound waves with frequencies higher than the human hearing range, typically above 20 kHz, used in various applications such as medical imaging and NDE. These high-frequency waves travel through materials and reflect off interfaces or discontinuities, providing information about the material internal structure. [1–6](#), [8](#), [11–14](#), [19–23](#), [25](#), [27–31](#), [36](#), [38](#), [39](#), [41](#), [44–47](#), [49](#), [55](#), [57](#), [58](#), [71–75](#), [91](#), [92](#), [94–101](#)

Variational Autoencoder (VAE): VAE are generative models that learn to encode input data into a lower-dimensional latent space while enforcing probabilistic structure

using variational inference. They consist of an encoder that maps inputs to a distribution over latent variables and a decoder that reconstructs data by sampling from this distribution. VAE are widely used in applications such as anomaly detection, image generation, and data imputation due to their ability to model complex data distributions effectively. Though similar, VAE are distinct from AE which directly map inputs to a fixed latent space representation; VAE instead learn a distribution over the latent space. [4–7](#), [10](#), [32](#), [33](#), [35](#), [36](#), [38](#), [45–47](#), [50](#), [55–57](#), [60](#), [87](#), [91](#), [92](#), [95](#), [96](#), [98](#)