

PREDICTING ENERGETIC MATERIAL PROPERTIES AND INVESTIGATING  
THE EFFECT OF PORE MORPHOLOGY ON SHOCK SENSITIVITY VIA  
MACHINE LEARNING

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Alex D. Casey

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2020

Purdue University

West Lafayette, Indiana

**THE PURDUE UNIVERSITY GRADUATE SCHOOL**  
**STATEMENT OF DISSERTATION APPROVAL**

Dr. Steven F. Son, co-Chair

School of Mechanical Engineering

Dr. Ilias Billionis, co-Chair

School of Mechanical Engineering

Dr. Terrence R. Meyer

School of Mechanical Engineering

Dr. Weinong Chen

School of Aeronautics and Astronautics

**Approved by:**

Dr. Nicole Key

Associate Head for Graduate Studies

School of Mechanical Engineering

To my wife, Madeline, who is sexy as hell, for her enduring support and optimism.

## ACKNOWLEDGMENTS

Many people supported me through this work. I am grateful for my academic advisors: Dr. Steven Son for his perspective and advocacy; Dr. Ilias Bilonis for his instructive class and discussions; and Dr. Terrence Meyer for his willing advice.

I would like to thank Brandon Terry for the enthusiasm he demonstrated and for the positive influence he had on me during my first year as an incoming graduate student. My time at Purdue was enhanced through friendships with my fellow classmates and researchers Zane Roberts, Vasant Vuppuluri, and Tim Manship. In particular, I would like to thank Morgan Ruesch for his friendship, thoughtful conversations, and for being a carpool buddy. Additionally, Nick Cummock merits special attention for all of our time spent together both inside and outside the lab.

I was very fortunate to receive mentorship and direction from Dr. Brian Barnes, Dr. Betsy Rice, and Dr. Barrett Hardin – all of whom were invested in my success. Specifically, I would like to thank Dr. Jesus Mares for always being willing chat and for providing much needed encouragement and guidance.

Finally, I am grateful for the examples of community members like Chris Stoker, Ben Denos, and Scott Bartholomew to name a few.

With respect to funding, I acknowledge that:

- This work was supported by the U.S. Air Force Office of Scientific Research through award no. FA9550-15-1-0102 and the project's Program Manager Dr. Martin Schmidt.
- This research was conducted with Government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a.

- This work was supported in part by a grant of computer time from the DoD High Performance Computing Modernization Program at the ARL DoD Supercomputing Resource Center.
- This work was sponsored by the High Performance Computing Modernization Program (HPCMP) and the HPC Internship Program (HIP).

DISTRIBUTION A. Approved for public release: distribution unlimited.  
(96TW-2020-0173).

## TABLE OF CONTENTS

	Page
LIST OF TABLES . . . . .	viii
LIST OF FIGURES . . . . .	ix
ABSTRACT . . . . .	xv
1. INTRODUCTION . . . . .	1
1.1 Motivation . . . . .	1
1.2 Overview and Organization . . . . .	2
2. ENERGETIC MATERIALS SAFETY . . . . .	5
2.1 Initiation Mechanisms . . . . .	5
2.2 Material Property and Sensitivity Tests . . . . .	9
3. STATISTICAL LEARNING . . . . .	15
3.1 Terminology . . . . .	17
3.2 Machine Learning Stack . . . . .	19
4. VISUALIZATION OF HOT SPOTS . . . . .	21
4.1 Introduction and Motivation . . . . .	21
4.2 Experimental details . . . . .	24
4.2.1 Sample preparation . . . . .	24
4.2.2 Setup and phosphorescence imaging . . . . .	25
4.2.3 Sylgard transmission and laser fluence . . . . .	26
4.2.4 Calibration . . . . .	29
4.3 Results and discussion . . . . .	31
4.4 Conclusion . . . . .	33
5. SENSITIVITY TESTS . . . . .	35
5.1 Neyer Sensitivity Test . . . . .	35
5.1.1 Maximum Likelihood Estimators . . . . .	37
5.1.2 Information Matrix . . . . .	39
5.1.3 Sequential Design . . . . .	43
5.2 Modernized (Generalized) Notation . . . . .	47
5.3 Proposed Methods . . . . .	49
5.3.1 Conjugate Neyer . . . . .	49
5.3.2 Maximum Likelihood Estimation Applied to Mixture of Gaussians . . . . .	49
6. MACHINE LEARNING OF THE SIMULATED COLLAPSE OF PORE GE- OMETRY IN HMX UNDER SHOCK LOADING . . . . .	63

	Page
6.1 Collaborators . . . . .	63
6.2 Introduction . . . . .	63
6.3 Extracting Void Shapes . . . . .	64
6.3.1 Simple Shape Geometries . . . . .	66
6.3.2 Removing the Effect of Size . . . . .	66
6.4 CTH Simulation . . . . .	67
6.4.1 Criticality Criterion . . . . .	69
6.4.2 Data Sets . . . . .	71
6.5 ML Models . . . . .	72
6.5.1 Pore Representations . . . . .	72
6.5.2 Model Types . . . . .	80
6.6 Results and Discussion . . . . .	82
6.6.1 Ellipse Simulations . . . . .	82
6.6.2 Distribution of Critical Thresholds . . . . .	85
6.6.3 Realistic Pore Aspect Ratio and Orientation . . . . .	87
6.6.4 Model Comparison . . . . .	89
6.6.5 Additional Viewpoints . . . . .	95
6.6.6 Web App . . . . .	101
6.7 Conclusion and Future Work . . . . .	101
7. PREDICTION OF ENERGETIC MATERIAL PROPERTIES FROM ELEC- TRONIC STRUCTURE USING 3D CONVOLUTIONAL NEURAL NETWORKS	103
7.1 Collaborators . . . . .	103
7.2 Introduction . . . . .	103
7.3 Methodology . . . . .	110
7.3.1 Data Curation . . . . .	110
7.3.2 Input Standardization . . . . .	112
7.3.3 Approximating Rotational Invariance . . . . .	114
7.3.4 Model Architecture . . . . .	115
7.3.5 Training . . . . .	117
7.4 Results and discussion . . . . .	119
7.4.1 Rotational Invariance . . . . .	119
7.4.2 Orientation Correlation . . . . .	121
7.4.3 Prediction Accuracy . . . . .	124
7.4.4 Comparison to ECFP4 and Random Forest . . . . .	125
7.4.5 $\gamma$ -Randomization . . . . .	134
7.4.6 Effect of Extent Cutoff . . . . .	136
7.5 Conclusion and Future Work . . . . .	137
8. CONCLUSION . . . . .	140
REFERENCES . . . . .	142
A. DERIVATION OF MAXIMUM LIKELIHOOD ESTIMATORS AND INFOR- MATION MATRIX FOR A MIXTURE OF TWO GAUSSIANS . . . . .	153

## LIST OF TABLES

Table	Page
2.1 Drop weight test results for HMX. Type 12 Tools. E = Explosion, N = No explosion . . . . .	14
3.1 A non-exhaustive list of terms used throughout the literature which refer to the same data object or statistical learning concept. . . . .	19
6.1 Equation of state parameters for the detonation products and the reaction rate parameters. . . . .	68
6.2 The initial range of the random forest model hyperparameters. . . . .	81
6.3 The initial range of the extreme gradient boosting machine hyperparameters. . . . .	81
6.4 $R^2$ , MSE, and MAE performance metrics for all models on pores scaled to 150 nm equivalent diameter categorized by set designation – train, validation, or test. . . . .	90
6.5 $R^2$ , MSE, and MAE performance metrics for all models on pores scaled to 300 nm equivalent diameter categorized by set designation – train, validation, or test. . . . .	91
7.1 Maximum values for the Pearson (P) and Spearman (S) correlation coefficients for each target quantity and the associated pair of orientations that produced it. . . . .	123
7.2 The random forest hyperparameters that were allowed to vary and their prior distributions. Brackets indicate a discrete distribution with support [lower bound, upper bound, step size] and curly braces indicate a choice between values. . . . .	131
7.3 $R^2$ scores for all eight prediction tasks broken down by model and set designation (train, validation, or test). . . . .	132
7.4 Mean absolute errors for all eight prediction tasks broken down by model and set designation (train, validation, or test). . . . .	133

## LIST OF FIGURES

Figure	Page
2.1 Decomposition rate of RDX. $A'$ is the amount of RDX that has not yet decomposed and $A$ is the starting amount of RDX. . . . .	7
2.2 A non-exhaustive list of material behavior/property and sensitivity tests. . . . .	12
2.3 Drop weight test results for HMX. . . . .	14
3.1 A proposed relationship between AI, machine learning, and deep learning. . . . .	18
3.2 A proposed relationship between common statistical learning terms. . . . .	18
3.3 Deep learning hardware and software stack. . . . .	20
4.1 Sample geometry. a) Phosphor layer is parallel to transducer. Laser excitation enters through the top surface and the temperature field is visualized through the same. b) Phosphor layer is orthogonal to transducer. Laser incidence and camera view can be on the same side ('in reflection') or opposing sides ('in transmission'). c) Depiction of the relative size of the HMX crystal and the phosphor layer. . . . .	24
4.2 Built-in-house stereoscope. a) lens mount, b) 50:50 beamsplitter and knife edge, c) converging lens, d) filter mount, e) high transmission mirror, f) right-angle mirror, g) converging lens, h) camera mount. Colored lines: simplified image path. . . . .	26
4.3 Experimental setup. . . . .	27
4.4 Example image of a mult-crystal sample. The left side of both images is the sample visualized through the 420+ nm filter and the right side is the image replicated (and reflected) through the $390\pm 5$ nm filter. Top: before activation of transducer. Bottom: 10 s after activation of transducer. Internal heating is observed where the right image pane becomes darkened relative to the left image pane (relative decrease in $390\pm 5$ nm filtered signal strength compared to 420+ nm filtered signal). Sample height is approximately 4 mm. . . . .	28
4.5 Effect of laser pulse energy on intensity ratio. Beam diameter was $2\text{cm}$ . . . . .	29
4.6 Top-down view of intensity ratio histograms to temperature. Breaks in curve occur when the camera systems stops recording to save data. Inset: Alternative view of histograms at select temperatures show narrower distributions at lower temperatures. . . . .	31

Figure	Page
4.7 Left axis: Temperature histograms of 100 averaged frames at select thermocouple temperatures. Right axis: (symbol: ●) Normalized standard deviation.	32
4.8 Visualized temperature field at selected frames. Frame field of view is 4mm by 4mm. . . . .	33
4.9 Plot detailing the temporal history of the hot spot. Inset: Magnified view of the average temperature and standard deviation before hot spot occurrence. . . . .	34
5.1 Expected values of estimators $\hat{\sigma}$ and $\hat{\sigma}$ for simulated data . . . . .	41
5.2 Values of the elements in the information matrix for a single specimen . . . . .	44
5.3 Flowchart of Neyer's sequential design algorithm . . . . .	53
5.4 Graph demonstrating interaction of sensitivity model random variables assuming that the threshold distribution is parameterized by two variables . . . . .	55
5.5 Left: Example of a bimodal distribution composed of two Gaussians. The left Gaussian is given by $\sim N(47, 1)$ and the right by $\sim N(54, 2.5)$ . The prefactors of the left and right Gaussian distributions are .4 and .6, respectively. Right: the cumulative density function of bimodal distribution shown in the left pane.	55
5.6 Neyer sequential design of experiments algorithm applied to a bimodal latent distribution. . . . .	56
5.7 Continued: Neyer sequential design of experiments algorithm applied to a bimodal latent distribution. . . . .	56
5.8 Continued: Neyer sequential design of experiments algorithm applied to a bimodal latent distribution. . . . .	57
5.9 Neyer sequential design of experiments algorithm applied to a disperse bimodal latent distribution. Bimodal distribution is given by $\sim N(35, 3)$ , $N(55, 2)$ , and prefactors of .5 and .5. . . . .	58
5.10 Continued: Neyer sequential design of experiments algorithm applied to a disperse bimodal latent distribution. . . . .	59
5.11 Continued: Neyer sequential design of experiments algorithm applied to a disperse bimodal latent distribution. . . . .	59
5.12 Continued: Neyer sequential design of experiments algorithm applied to a disperse bimodal latent distribution. . . . .	60
5.13 Estimated pdf and cdf using bimodal maximum likelihood estimators. . . . .	60
5.14 Values of select elements for the information matrix of a mixture of two Gaussians. The values are plotted for a single specimen assuming a mixture model defined by two Gaussians as $0.5N(-1, 1) + 0.5N(1, 1)$ . . . . .	61

Figure	Page
5.15 Terms $a_{11}$ and $a_{22}$ of Figure 5.14 are shown here with appropriately scaled axes for clarity. . . . .	62
6.1 a) An image of a sectioned and polished surface of a single pressed class 5 HMX sample acquired via FIB-SEM. b) The binarized representation of image shown in a). White indicates HMX and black is void space. . . . .	65
6.2 Example of the time evolution of the temperature field for a pore collapse simulation in CTH. The incoming flyer plate velocity was 2528 m/s. Equivalent circular diameter: 150 nm. Explicit pore id number: 106028. . . . .	70
6.3 Comparison of a) aspect ratio and b) orientation of realistic pore geometries scaled to an equivalent diameter of 150 nm using the bounding box and Fourier ellipse methods. Clusters in the upper left and lower right quadrants of b) are an artifact of phase wrapping and are not indicative of gross disagreement between the methods. . . . .	75
6.4 a) The original binary image of an example pore. b) The extracted pore contour. c) The black line is the scaled path such that the pore has an equivalent circular diameter of 300nm. The salmon colored line is the scaled pore path when carrying only the first harmonic ( $n_h=1$ , the “Fourier ellipse”). The navy colored line is the scaled pore path using the first three harmonics ( $n_h=3$ ). d) The standardized binary representation of the example pore. Dataset: 300 nm. Implicit pore id number: 400. . . . .	79
6.5 The critical velocity thresholds for various ellipses scaled to equivalent diameters of 150 nm (left) and 300 nm (right). a) The critical threshold velocity with respect to ellipse aspect ratio and orientation. b) The critical threshold velocity with respect to orientation for select aspect ratios. c) The critical threshold velocity with respect to aspect ratio for select orientations. . . . .	84
6.6 a) Distribution of critical velocity thresholds for realistic pores scaled to 150 nm equivalent diameter. b) Distribution of critical velocity thresholds for realistic pores scaled to 300 nm equivalent diameter. c) Distributions of critical velocity thresholds at 150 nm and 300 nm equivalent diameters only considering the pore shapes share between the two data sets (1452 pores). . . . .	86
6.7 The critical threshold velocities of realistic pores with respect to their measured aspect ratio and orientation. a) The data set of pores scaled to 150 nm equivalent diameter. b) The data set of pores scaled to 300 nm equivalent diameter. . . . .	88
6.8 The random forest model critical threshold velocity predictions on the test set versus the reference velocities obtained via CTH for pores scaled to 300 nm equivalent diameter when using a) the aspect ratio and orientation, b) the simple metrics, and c) the Fourier coefficients as model inputs. . . . .	94

Figure	Page
6.9 The 2D t-distributed stochastic neighbor embedding of the of 300 nm equivalent diameter pores represented in the CNN-learned feature space. Pore fill color indicates the critical threshold velocity obtained from the CTH simulations and pore border color marks the predicted critical threshold velocity from the CNN. The figure is supplied in vector format so that if viewed electronically it can be enlarged indefinitely. Not all pores shapes are displayed; pores are subsampled randomly so that pore shapes do not overlap. . . . .	97
6.10 Pore shape plotted according to their location in aspect ratio - orientation space for a) the data set of pores scaled to 150 nm equivalent diameter and b) the data set of pores scaled to 300 nm equivalent diameter. Pore fill color indicates the critical threshold velocity obtained from the CTH simulations and pore border color marks the predicted critical threshold velocity from the CNN. The figure is supplied in vector format so that if viewed electronically it can be enlarged indefinitely. Not all pores shapes are displayed; pores are subsampled randomly so that pore shapes do not overlap. . . . .	98
6.11 Enlarged versions of the regions called out in a) Figure 6.10b and b) Figure 6.9.	100
7.1 Distribution of a) molecular weight, b) oxygen balance, and c) molecular extent for the data set of 26,265 selected molecules. Molecular extent is simply defined as the length of the bounding box in each direction (x, y, z) as output by Gaussian when describing spatial point data. For c, the red dotted line represents the length of each edge in the interpolated bounding box.	110
7.2 Physics-based workflow. A sample molecule 2-nitrofurane is originally represented by a SMILES string. The string is processed by RDKit to produce atomic coordinates for a conformation near the molecule's minimum total electronic energy. Gaussian 16 is called and outputs, among other things, estimates of the molecule's total electronic energy, HOMO-LUMO gap, dipole moment, and 3D point grids of electron density and electrostatic potential. The ARL Toolkit further manipulates the electronic density and electrostatic potential point grids to estimate the molecule's crystal density and solid phase heat of formation. With these properties, the thermochemical code Cheetah v. 8.0 is used to estimate Chapman-Jouguet (C-J) detonation velocity, C-J detonation pressure, and C-J temperature. . . . .	112
7.3 Vector graphic of the electron charge density 3D spatial point grid of an example molecule, 2-nitrofurane, before and after standardization. a) The original (output by Gaussian) and standardized grid shapes. b) A 2D slice showing the location of the grid points (grey dots) before and after interpolation. Note that the colorbar applies to both subfigures. . . . .	114

Figure	Page
7.4 Selected model architecture. An example molecule, 2-nitrofurán, is represented by a standardized input consisting of two stacked 3D spatial point grids each of shape (64, 64, 64) – one for the electron charge density and one for the electrostatic potential – to form a 4D tensor of shape (64, 64, 64, 2). A convolutional kernel of shape (3, 3, 3) is applied to the input followed by ReLU activation function and a (2, 2, 2) max pooling operation. This process is repeated increasing the number of “filters” while decreasing the data shape. The result is flattened and passed to a series of fully connected layers, each followed by ReLU activation. Finally, an additional 8 sets of fully connected layers (again followed by ReLU activations) are used to reduce the output down to 8 neurons (with linear activation) – one for each target quantity. . . .	116
7.5 Examples of the two-dimensional structural frameworks of molecules present in the train, validation, and test sets after scaffold splitting. . . . .	118
7.6 Loss history of the model during training. The joint MSE is the loss function described in Eq. 7.2. During each epoch the model encountered 21,012·24=504,288 examples. . . . .	119
7.7 Activations ( <i>a</i> ) for each orientation of molecules a) 2-nitrofurán (training set), and b) <chem>CC1=COC(=N)C(=O)C(O)=NC=C(N1)[N+](O-)=O</chem> (validation set, shown in Figure 7.5). White indicates an inactive neuron; i.e. <i>a</i> = 0. . . . .	120
7.8 Prediction errors of C-J detonation pressure. a) Matrix displaying the correlation strength of prediction error between pairs of input tensor orientations. The upper triangle of the matrix specifies the correlation using the Pearson metric and the lower triangle of the matrix uses the the Spearman correlation metric. The red squares highlight the orientation pairs with the highest correlation using either the Pearson or Spearman method. b) Kernel density estimates of the distributions of C-J detonation pressure prediction error for each input tensor orientation (24) and their average. . . . .	122
7.9 Predicted value (CNN) vs. actual value (physics-based workflow estimate). For both subfigures the top plot references the training set, the middle plot consists of the validation set, and the bottom plot contains the test set. The red line indicates an exact prediction (predicted equals actual). Each subfigure corresponds to a different target output: a) total electronic energy, b) HOMO-LUMO gap. . . . .	126
7.10 Predicted value (CNN) vs. actual value (physics-based workflow estimate). For both subfigures the top plot references the training set, the middle plot consists of the validation set, and the bottom plot contains the test set. The red line indicates an exact prediction (predicted equals actual). Each subfigure corresponds to a different target output: a) dipole moment, b) crystal density. . . . .	127

Figure	Page
7.11 Predicted value (CNN) vs. actual value (physics-based workflow estimate). For both subfigures the top plot references the training set, the middle plot consists of the validation set, and the bottom plot contains the test set. The red line indicates an exact prediction (predicted equals actual). Each subfigure corresponds to a different target output: a) solid phase heat of formation, b) Chapman-Jouguet detonation temperature. . . . .	128
7.12 Predicted value (CNN) vs. actual value (physics-based workflow estimate). For both subfigures the top plot references the training set, the middle plot consists of the validation set, and the bottom plot contains the test set. The red line indicates an exact prediction (predicted equals actual). Each subfigure corresponds to a different target output: a) Chapman-Jouguet detonation pressure, b) Chapman-Jouguet detonation velocity. . . . .	129
7.13 a) Stacked histograms of total electronic energy values present in the training, validation, and test sets. b) Probability distributions of extent along the x-axis. The dashed black line represents the outer limit of the input standardization bounding box. . . . .	135
7.14 Average scaled absolute error vs. input pre-standardization x direction extent for a) training, b) validation, and c) test sets. Red dotted line indicates extent cutoff. . . . .	138

## ABSTRACT

Casey, Alex D. Ph.D., Purdue University, August 2020. Predicting Energetic Material Properties and Investigating the Effect of Pore Morphology on Shock Sensitivity via Machine Learning. Major Professors: Steven F. Son, Ilias Bilonis, School of Mechanical Engineering.

An improved understanding of energy localization (“hot spots”) is needed to improve the safety and performance of explosives. In this work I establish a variety of experimental and computational methods to aid in the investigation of hot spots. In particular, focus is centered on the implicit relationship between hot spots and energetic material sensitivity. To begin, I propose a technique to visualize and quantify the properties of a dynamic hot spot from within an energetic composite subjected to ultrasonic mechanical excitation. The composite is composed of an optically transparent binder and a countable number of HMX crystals. The evolving temperature field is measured by observing the luminescence from embedded phosphor particles and subsequent application of the intensity ratio method. The spatial temperature precision is less than 2% of the measured absolute temperature in the temperature regime of interest (23-220 °C). The temperature field is mapped from within an HMX-binder composite under periodic mechanical excitation.

Following this experimental effort I examine the statistics behind the most prevalent and widely used sensitivity test (at least within the energetic materials community) and suggest adaptations to generalize the approach to bimodal latent distributions. Bimodal latent distributions may occur when manufacturing processes are inconsistent or when competing initiation mechanisms are present.

Moving to simulation work, I investigate how the internal void structure of a solid explosive influences initiation behavior – specifically the criticality of isolated hot spots – in response to a shock insult. In the last decade, there has been a significant modeling and simulation effort to investigate the thermodynamic response of a shock induced pore

collapse process in energetic materials. However, the majority of these studies largely ignore the geometry of the pore and assume simplistic shapes, typically a sphere. In this work, the influence of pore geometry on the sensitivity of shocked HMX is explored. A collection of pore geometries are retrieved from micrographs of pressed HMX samples via scanning electron microscopy. The shock induced collapse of these geometries are simulated using CTH and the response is reduced to a binary “critical” / “sub-critical” result. The simulation results are used to assign a minimum threshold velocity required to exhibit a critical response to each pore geometry. The pore geometries are subsequently encoded to numerical representations and a functional mapping from pore shape to a threshold velocity is developed using supervised machine-learned models. The resulting models demonstrate good predictive capability and their relative performance is explored. The established models are exposed via a web application to further investigate which shape features most heavily influence sensitivity.

Finally, I develop a convolutional neural network capable of directly parsing the 3D electronic structure of a molecule described by spatial point data for charge density and electrostatic potential represented as a 4D tensor. This method effectively bypasses the need to construct complex representations, or descriptors, of a molecule. This is beneficial because the accuracy of a machine learned model depends on the input representation. Ideally, input descriptors encode the essential physics and chemistry that influence the target property. Thousands of molecular descriptors have been proposed and proper selection of features requires considerable domain expertise or exhaustive and careful statistical downselection. In contrast, deep learning networks are capable of learning rich data representations. This provides a compelling motivation to use deep learning networks to learn molecular structure-property relations from “raw” data. The convolutional neural network model is jointly trained on over 20,000 molecules that are potentially energetic materials (explosives) to predict dipole moment, total electronic energy, Chapman-Jouguet (C-J) detonation velocity, C-J pressure, C-J temperature, crystal density, HOMO-LUMO gap, and solid phase heat of formation. To my knowledge, this demonstrates the first use of the complete 3D electronic structure for machine learning of molecular properties.

## 1. INTRODUCTION

### 1.1 Motivation

It is commonly accepted that the initiation of energetic materials originates from localized regions within the material that are at elevated temperature. Even for circumstances that appear to have non-thermal beginnings – such as impacting the material – the pervading theory is that the input energy is converted to thermal energy subsequently raising the temperature of the material above that required for ignition. The localized high temperature regions are widely referred to as “hot spots”. Although the theory of hot spots has been generally embraced since the 1950s, hot spots remain an extremely active area of research.

The major difficulty in describing hot spot formation processes and their subsequent growth is that there is no single dominant process since the mechanism or mechanisms which operate depend on the energy input (stimulus type) and the physical properties of the explosive. Additionally, since many hot spots occur in extremely small spatial and temporal regimes it is difficult to observe them experimentally. Due to this complexity, over the last decade, there has been a strong movement to simulate hot spot ignition and growth with high fidelity continuum, mesoscale, and molecular dynamics models.

Throughout the literature hot spots are implicitly linked to the sensitivity of energetic materials. That is, energetic materials that are sensitive, or that respond (initiate, react) to low stimulus levels, have a greater propensity to form hot spots. There are a variety of sensitivity tests designed to provide a simple measure of an energetic material’s initiation threshold. Again, because the underlying mechanisms leading to hot spot formation are dependent on the stimulus type, these sensitivity tests generally take the form of isolating and applying a single stimulus and noting the stimulus level that results in the sustained reaction and consumption of the test specimen. While these tests are crucial in quickly screening novel energetic materials for safety concerns they have proven too unreliable

to directly investigate hot spots. Part of this unreliability stems from the fact that these tests are unaware of material microstructure.

A major goal of the recent surge in modeling and simulation efforts is to provide hot spot and sensitivity linkages over the full variety of stimulus types, bulk and molecular properties, and observed microstructure. Often, the simulations are computationally expensive and are restricted to small domains and idealized systems. Additionally, in accordance with good experimental design these simulations are narrowly focused. The confluence of these factors means that furthering the understanding of hot spots is a time intensive process.

While machine learning has existed in some form over the last century, recent advances in graphics processing units (GPUs) for computation and significant successes in long-standing artificial intelligence problems have heralded its adoption by nearly every scientific field. At a basic level, machine-learned models can approximate arbitrary functional mappings and can be formed to essentially copy physics-based simulations. Once an initial computational price has been paid to train the machine-learned models, these models can make rapid predictions on new items. As such, machine-learned models can be used to augment and increase the throughput of conventional physics-based simulations.

## 1.2 Overview and Organization

The scope of this work was broadly defined to gain insights into hot spot formation and energetic material sensitivity. To this end, the initial focus of this work centered on experimentally visualizing a hot spot *in situ*. While the existence of hot spots is backed by a myriad of experimental evidence very little work has been conducted to actually “see” a hot spot. To be clear, stunning visualizations of hot spots have been produced using simulation tools but real images of physical reacting systems are essentially non-existent. A hot spot, as defined, is a localized high temperature region. Therefore, experimental hot spot visualization is equivalent to the task of visualizing an evolving

temperature field. Direct observation of the temperature field was made using phosphor thermography on a specific, single-particle, visually-clear energetic system.

The work then shifted to explore the statistics used in parameter estimation of common sensitivity tests. Specifically, it was noted that the underlying assumption that the stimulus level latent distribution is unimodal may be incorrect for some materials.

The final efforts develop machine learning surrogate models to replicate the sensitivity of a collapsing pore under shock loading as produced by a hydrocode simulation. Furthermore, a deep learning model is produced capable of predicting energetic material detonation properties after learning a numeric representation of an energetic molecule.

The remaining chapters of the work are organized to emphasize a single topic. Chapters 2 and 3 of this text provide background information on initiation of energetic materials and their safety metrics, and statistical (or machine) learning, respectively. These chapters are not intended to be exhaustive reviews as each of the topics covered merits a panoply of textbooks, however, these chapters serve to provide the reader the requisite background information to understand the remaining chapters. That is, the basic concepts and methods used in in the latter chapters are explicitly addressed in Chapters 2 and 3.

Chapter 4 deals with hot spot formation under periodic mechanical input. In particular this chapter demonstrates a novel method for experimentally visualizing the dynamic temperature field of an evolving hot spot within a mock polymer bonded explosive.

Chapter 5 discusses the statistical tests developed since the 1930's for estimating sensitivity parameters. The Neyer algorithm, which is the dominant test in the field of energetic materials, is described in detail. Potential adaptations to the Neyer algorithm to accommodate bimodal latent distributions are also proposed.

Chapter 6 explores the effect of shape on the sensitivity (hot spot formation) of collapsing pores under shock loading. Machine learned-surrogate models are trained using sensitivity labels generated via the simulation of thousands of realistic pore structures acquired via scanning electron microscopy (SEM). This portion of work was conducted in partnership with the Air Force Research Laboratory (AFRL) at Eglin Air Force Base (AFB).

Chapter 7 details the creation of a convolutional neural network (CNN) developed to make predictions on energetic material detonation performance metrics. This work takes advantage of a deep neural networks ability to learn rich feature representations and novelly acts directly on the electronic structure of a molecule; eliminating the need to hand craft, or select, a suite of molecular descriptors. This effort was produced in collaboration with the Army Research Laboratory (ARL) located at Aberdeen, MD.

A summary of the conclusions is presented in Chapter 8. Finally, as supplemental information to Chapter 5, the derivation of the maximum likelihood estimates and information matrix for a mixture of two Gaussians is given in Appendix A.

## 2. ENERGETIC MATERIALS SAFETY

A universally accepted definition of combustion is difficult, if not impossible, to prescribe. Throughout the 18<sup>th</sup> century combustion was generally described as the “bonding” of *combustible materials* (typically hydrocarbons) with oxygen. Many, if not most, combustible systems of practicality and interest can be reduced to three broad components; fuel, oxidizer, and ignition source.

Energetic materials are generally described as the set of combustible materials which do not rely on any additional (or external) source for an oxidizer. Some may burn with air, because they are ‘fuel rich’, but the reaction is self-sustaining in an inert (non-oxidizing) environment. Energetic materials are loosely grouped into three application based categories, namely: propellant, explosives, and pyrotechnics.

The field of energetic materials incorporates a variety of sciences including chemistry, thermodynamics, fluid dynamics, aerodynamics, mechanics, electricity, electronics, and even meteorology, biology, and physiology [1]. Consequently, the field provides a rich area of research consisting of difficult problems to which the solutions often require interdisciplinary efforts to elucidate. Two major sub-areas of research within the field deal in understanding how the energetic material reactions are started (*ignition/initiation*), and how unwanted initiation can be prevented (*safety*).

### 2.1 Initiation Mechanisms

In combustion, the *reaction rate* is the speed at which products are turned into reactants. Considering a simple reaction, such as



a common functional form for the reaction rate is

$$r = k(T)[A]^x[B]^y \quad (2.2)$$

where  $[A]$  and  $[B]$  are the molar concentrations of species  $A$  and  $B$ , respectively, and the exponents  $x$  and  $y$  are the partial *orders of reaction* (typically determined experimentally). The prefactor  $k$  is the *rate constant*. A common expression for the rate constant is the Arrhenius equation given by

$$k(T) = A(T)e^{\frac{-E_a}{k_b T}}, \quad (2.3)$$

or

$$k(T) = A(T)e^{\frac{-E_a}{RT}} \quad (2.4)$$

where,  $A(T)$  is the pre-exponential factor,  $E_a$  is the activation energy (units depend on which form of the equation is used),  $k_b$  is the Boltzmann constant,  $R$  is the universal gas constant, and  $T$  is the absolute temperature in Kelvin.

While the equations 2.2, 2.3, and 2.4 were described herein as common, it must be noted that the origin of these equations lies in the chemical kinetics of gases and that these equations are often not representative of the reaction processes that occur in a large number of energetic materials systems. For example, the Arrhenius equation can typically be applied to homogeneous explosives but generally fails to accurately describe heterogeneous explosives. That said, the Arrhenius equation is listed here because it provides a good historical framework for the discussion of initiation mechanisms of energetic materials.

The Arrhenius equation has origins dating back to as early as 1878 as an experimental relationship developed by Hood [2]. Later, van't Hoff [3] and Arrhenius [4] demonstrated that the relationship was consistent with the principles of thermodynamics. Within this equation, the physical meaning of the activation energy and the pre-exponential factor have been the subject of much debate. In 1969, Menzinger and Wolfgang [5] discussed a more modern interpretation of these terms under collision theory but note that the original interpretation of the activation energy as proposed by Arrhenius was “the height of the energy barrier which has to be overcome by the relative translational motion of the reactants in order for reaction to occur”. This original interpretation of a “barrier to be overcome” alongside the functional form of the Arrhenius equation instill in one’s mind

the a deep notion that combustion processes have a discrete, or critical, temperature at which the reaction “turns on”.

To further illustrate this point, consider the case of the decomposition rate of RDX which can be approximated by the Arrhenius equation and has the experimentally determined values of  $3.5575 \times 10^{18} \text{ (s}^{-1}\text{)}$  and  $24027 \text{ (K)}$  for the pre-exponential factor and  $\frac{E_a}{R}$ , respectively [1]. A plot of equation 2.4 for decomposing RDX as a function of temperature is given in Figure 2.1. As seen in this figure, RDX has a distinct temperature ( $T \approx 550 \text{ K}$ ) at which the onset of significant decomposition occurs. When dealing with ‘Arrhenius-like’ rate expressions, the initiation of a self-sustaining reaction results when the combustion products reach a critical temperature (or other threshold).

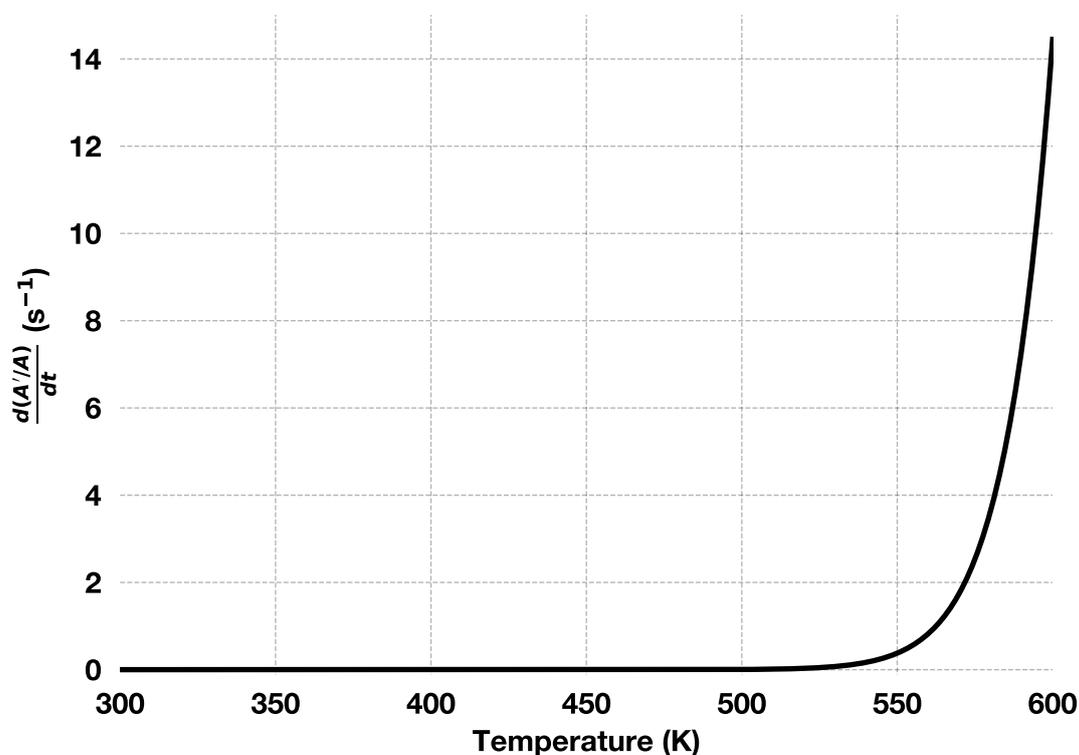


Figure 2.1. Decomposition rate of RDX.  $A'$  is the amount of RDX that has not yet decomposed and  $A$  is the starting amount of RDX.

Again, while Arrhenius type expressions do not govern all energetic materials, the paradigm of a critical temperature is deeply etched in the theories of ignition and ini-

tiation. To date, the major agreement is that underlying mechanism for ignition and growth in energetic materials is thermal in nature [6–8].

For initiation mechanisms that appear to have non-thermal origins (for example, impacting the material), the French chemist Bertholot was the first to propose an initiation mechanism; namely, that the kinetic energy of impact is converted to heat subsequently raising the temperature of the material above that required for ignition [9]. Later studies by Taylor and Weale in the 1930’s demonstrated that the mechanical work dissipated in a solid explosive as heat was much less than that required to raise the temperature of the whole sample to that required for ignition [10]. From this observation stemmed the hypothesis of energy localization which later became known as the formation of “hot spots”. Bowden and Yoffe attested the hot spot hypothesis by adding grits to explosives [6]. For example, they observed that by adding a grit with a low melting point to the explosive, the maximum local temperature is limited to the grit melting point and hot spot initiation is attenuated before ignition. This work promoted greater safety in explosives and increased efficiency in propellants. Additionally, they showed that hot spot formation affects the reaction growth and deflagration speeds.

Despite these observations, and having a widely accepted working theory for initiation, many more questions remain in regards to hot spot formation mechanisms and today this remains a highly active area of research. Modernly, Asay notes, “Once initiated, a hot spot can either fail, survive for short amount of time, or progress and grow. This is the problem of criticality, and without knowing how this process occurs, it is impossible to understand and predict the outcome of an initiation event” [11].

The major difficulty in describing hot spot formation processes and their subsequent growth is that there is “no single dominant process since the mechanism or mechanisms which operate depend on the energy input and the physical properties of the explosive” as noted in a landmark review paper by Field [8]. In the same paper Field outlined ten proposed hot spot ignition mechanisms:

- (i) adiabatic compression of trapped gas spaces

- (ii) other mechanisms involving cavity collapse such as viscous or plastic heating of the surrounding matrix material or, for very high shock collapse pressures, hydrodynamic shock focusing
- (iii) friction between sliding or impacting surfaces, or between explosive crystals and/or grit particles in an explosive
- (iv) localized adiabatic shear of the material during mechanical failure
- (v) viscous heating of material rapidly extruded between impacting surfaces
- (vi) heating at crack tips
- (vii) heating at dislocation pileups
- (viii) spark discharge
- (ix) triboluminescent discharge
- (x) decomposition, followed by Joule heating of metallic filaments.

In order to determine which mechanisms are at play in a specific material for a given type of input energy, a plethora of tests have been developed. These tests seek to characterize, qualitatively or quantitatively, the energetic material properties, and behaviors, and their *sensitivity* to various stimuli.

## 2.2 Material Property and Sensitivity Tests

In order to prevent accidental ignition, it is critical to understand the behaviors and sensitiveness of energetic materials. To emphasize, this is a difficult task because:

Whatever method we use to predict possible accidents, we have one fundamental problem. The substances can be subjected to so many different situations, each with its own pattern of different kinds of stress, that it is impossible to imagine them all. With more understanding, we can make predictions of risks in more and more situations, but a wholly complete

understanding and description of all potential risk situations will never be possible [12].

As such, a wide variety of tests have been developed in order to mimic potential situations that an energetic material may experience in the field and also to describe potentially relevant material properties. A non-exhaustive list of tests can be loosely grouped into five categories as shown in Figure 2.2.

For the sensitivity tests, most report a binary response (successful reaction or not; ‘go’ or ‘no-go’) given a simple measure of input (drop weight height, input pressure, etc.). Intuitively, these tests are designed to focus on a single type of energy input, but unfortunately many initiation mechanisms are often compounded. Additionally, with respect to explosives, Smith [13] notes

The sensitivity of an explosive is not a well-defined property of the material, expressible as a single number, say, but is, instead a complex pattern of behavior. Different sensitivity tests, even when intended to measure the same property, will frequently produce different orders of relative sensitivity for a given series of explosives. In other words, there is not even a unique qualitative scale of sensitivity. In most sensitivity tests, the response of an explosive varies in seemingly random fashion over some range of severity of the applied stimulus. That is, there is no sharp threshold above which the explosive will always explode, below which it will never explode.

As an example, consider the drop weight impact test. Simply described, a drop weight impact test consists of an apparatus which drops (gravity driven) a weighted striker from a prescribed height onto a small explosive sample. Throughout the 20<sup>th</sup> century at least six drop weight impact sensitivity tests were developed [14–19]. The reason that so many testing variations and apparatuses were created is because consistent agreement on the sensitivity of explosives has not been achieved. This is in part because the underlying initiation mechanism is not fully understood [19], but also because the testing machines often produce spurious results; usually between “identical” apparatuses but in different locations (i.e. at different laboratories). Despite the inherent drawbacks

associated with drop-weight sensitivity tests, and other sensitivity tests in general, they are still commonly used to quantify material sensitivity to impact.

Sensitivity	Material Properties	Chemical Behavior	Shock Initiation	Detonation Performance
<ul style="list-style-type: none"> <li>• Gap</li> <li>• Friction</li> <li>• Drop weight</li> <li>• Skid</li> <li>• Bullet</li> <li>• Susan</li> <li>• Shotgun</li> <li>• Spigot</li> <li>• DDT</li> <li>• ESD</li> </ul>	<ul style="list-style-type: none"> <li>• Taylor impact</li> <li>• Instron</li> <li>• MTS</li> <li>• Hopkinson bar</li> </ul>	<ul style="list-style-type: none"> <li>• Henkin</li> <li>• DSC</li> <li>• TGA</li> <li>• ARC</li> <li>• Mass spectroscopy</li> <li>• Molecular beam</li> <li>• Bomb calorimeter</li> </ul>	<ul style="list-style-type: none"> <li>• Gap</li> <li>• Wedge</li> <li>• Minimum priming charge</li> <li>• Flyer plate</li> </ul>	<ul style="list-style-type: none"> <li>• Rate stick</li> <li>• Cylinder test</li> <li>• Plate dent</li> </ul>

Figure 2.2. A non-exhaustive list of material behavior/property and sensitivity tests.

Diving further into this example, Table 2.1 lists the result, explosion or no explosion, for HMX at various drop weight heights [13]. Table 2.1 demonstrates that these tests have statistical structure. This structure is more fully revealed when the results are plotted as in Figure 2.2. As seen, the *probability* of explosion increases with drop weight height. This probability of explosion as a function of height is a very useful predictive tool. However, many times in practice, an attempt to estimate the probability function is not made and only the  $L_{50}$ , or stimulus *level* at which 50% of the samples are predicted to explode, is reported. There is danger present in comparing the sensitivity of multiple explosives using only their reported  $L_{50}$  values as it is possible for a material to have a higher  $L_{50}$  but still result in more explosions at a lower stimulus level when compared to another material.

Another interesting artifact contained in Table 2.1 is that 60 HMX specimens needed to be tested in order to accurately estimate the drop height sensitivity. Testing this many samples may prove to be costly when considering the cost of synthesis and safe handling of an energetic material; or even impossible for novel materials. Additionally, the test is only informative if the tested stimulus levels are not well into the 0% ‘go’ or 100% ‘go’ regions. That is, testing at levels that always result in an explosion or that never result in an explosion are not very useful. For a new material, discovering these bounds may require lots of trial and error and consequently waste time, money, and effort.

Since the 1930’s, a handful of statistical tests have been developed to calculate, or rather estimate, the  $L_{50}$  or probability function associated with sensitivity tests. Some of these tests also attempt to minimize the number of specimens that need to be tested by prescribing a *sequential design of experiments*. These methods are one of the main discussions of this work and further information is found in Chapter 5.

Table 2.1. Drop weight test results for HMX. Type 12 Tools. E = Explosion, N = No explosion

Height (cm)	Results
40.5	E E E E E E E E E E
36.0	E N E E E E N E E E
32.0	E E N E E E N E N E
28.5	N E N N E N N N N E N
25.5	N N N N N N E N N N
22.5	N N N N N N N N N N

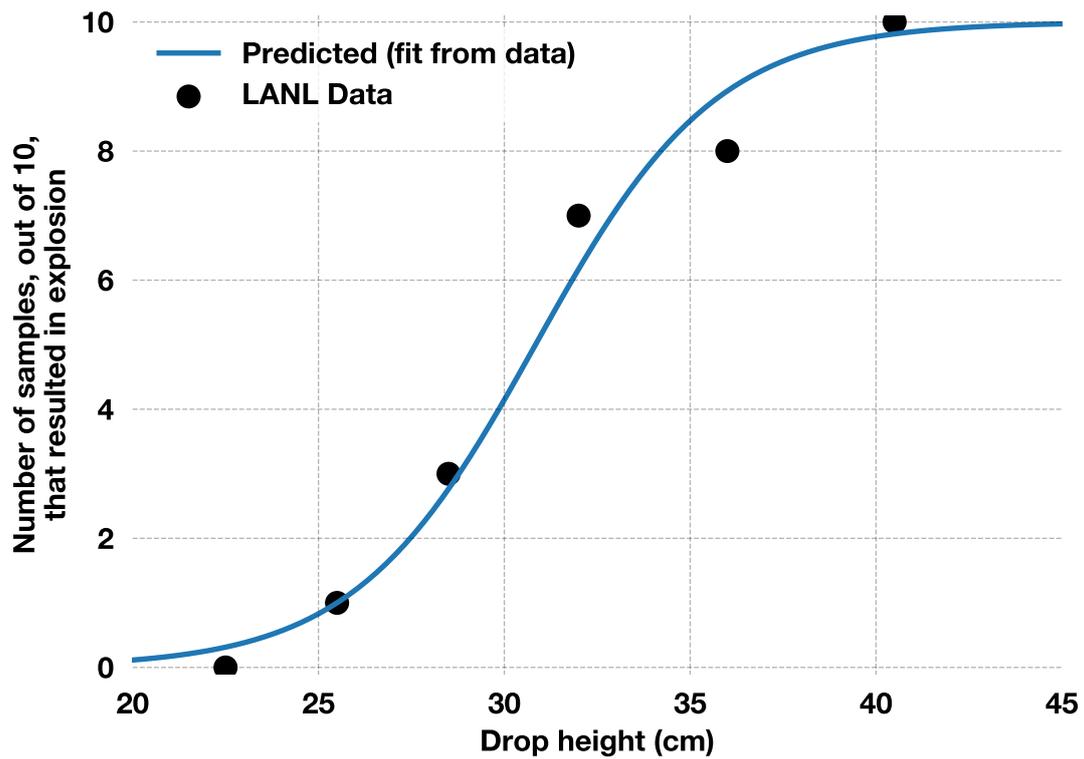


Figure 2.3. Drop weight test results for HMX.

### 3. STATISTICAL LEARNING

Statistical learning, broadly defined, is the extraction of patterns and insight from a set of data. This is commonly referred to as “learning from data”. The learning problems can be roughly categorized as either *supervised*, *unsupervised*, *self-supervised*, or *reinforcement* learning. [20–23]. In supervised learning, the goal is to predict an outcome based on a number of input measures having previously observed a set of inputs and their associated *labelled* outputs. For unsupervised learning, no outcome measure is provided, or such a measure does not exist, and the goal is to describe associations and patterns present in the inputs [22]. Self-supervised learning is like unto supervised learning but the output labels are not human annotated but rather generated from the input data. In reinforcement learning, an *agent* views its environment and chooses an action to maximize a reward [21].

The result of running a learning algorithm can be expressed as a function  $\mathbf{y}(\mathbf{x})$  which receives an input (a vector for multiple inputs) and outputs a *prediction*  $\mathbf{y}$  which may be a single value or a vector [20]. The form of the function is determined by the learning algorithm. This phase is most often referred to as the *learning, training, tuning, or fitting* phase and operates on a set of *training data*. The training set is simply a data set of inputs, and labelled outputs in the case of supervised learning, that is believed to be representative of a larger data set (*population*) or representative of inputs/outputs that will be observed in the future. Once the learning algorithm has trained a *model* (function  $\mathbf{y}$ ) it can produce predictions on a newly seen data set (inputs) known as a *test set*.

Learning algorithms require an *evaluation* metric to determine if the model is working properly (better than a baseline model) and to compare models against each other. For many algorithms this evaluation occurs by *scoring* the models based on their performance on an set of inputs for which the correct outcomes are known, but that were not used

during the training phase. This set of inputs is called the *hold-out*, or *validation* data set. The performance measure, or score, is often defined by a *prediction error*, *cost*, *loss*, or *accuracy* function.

For supervised (and self-supervised) learning problems an additional classification is based on the whether the output variable(s) is *discrete* (also referred to as *categorical* or *qualitative*) or *continuous* (also commonly described as *quantitative*). If the output is continuous the learning task is called *regression* and for discrete outputs the problem is known as *classification*. Sometimes a third output type is specified as *ordered* if the categorical values have an intuitive ordering; such as small, medium, and large [22]. Learning problems, and their associated algorithms, can often be further divided into categories such as online or offline learning, batch size (mini, batch, stochastic), and more.

For most practical applications, the input variables require *pre-processing* or *feature extraction* in preparation to be passed to the learning algorithm [20]. There are many instances where the input data may require pre-processing, such as: to speed up the computational efficiency of the algorithm, to transform the inputs to more useful representations, handling missing values, etc.

Finally, a widely accepted academic definition of statistical, or machine learning, is attributed to Professor Tom Mitchell at Carnegie Mellon University's School of Computer Science and is stated as follows [24]:

To be more precise, we say that a machine learns, with respect to a particular task T, performance metric P, and type of experience E, if the system reliably improves its performance P at task T, following experience E. Depending on how we specify T, P, and E, the learning task might also be called by names such as data mining, autonomous discovery, database updating, programming by example, etc.

### 3.1 Terminology

The phrases “artificial intelligence” (AI), “data science”, “data mining”, “statistical learning”, “machine learning”, “deep learning”, and “patter recongnition” and their definitions are often convoluted, interchanged, and used synonymously. This ambiguity arises because the modern field of “learning from data” was developed disjointedly and simultaneously by a number of classical fields like computer science, mathematics, and statistics. Also, in recent years, the concepts and near-future possibilities associated with AI, machine learning, and data science have been over-hyped in the public media [21]. The rapid dissemination of information stemming from technology-minded, but non-expert, publications has fueled the compounding of the terminology. Today, attempts to concisely define and relate these phrases have been made, but no general consensus exists as these relationships also spawn from different contexts.

François Chollet, in his book *Deep Learning with Python* [21], distinguishes between AI, machine learning, and deep learning with the diagram shown in Figure 3.1. He defines AI broadly as “the effort to automate intellectual tasks normally performed by humans”, machine learning as “a system that is trained rather than explicitly programmed”, and deep learning as a subset of machine learning which transforms the input data to outputs through more than one successive transformation.

Another relational paradigm is given by Robinson [25] and is shown in Figure 3.2. Robinson proposed a self-described oversimplified definition of the difference between the fields of data science, machine learning and artificial intelligence. He notes that AI is the oldest, largest, and hardest field to define and that, like described by Chollet, is often represented as encompassing the other fields. That said, Robinson believes that machine learning has “broken off” from AI as it has a strong focus on prediction whereas AI is modernly defined as an agent taking autonomous actions based on inputs it observes (like a self-driving car). He places deep learning as straddling the fields of machine learning and AI because deep learning models often return predictions but also these models have seen a wide variety of success when applied to agent-centric reinforcement problems like game-playing algorithms. He also notes that overlap between any of the

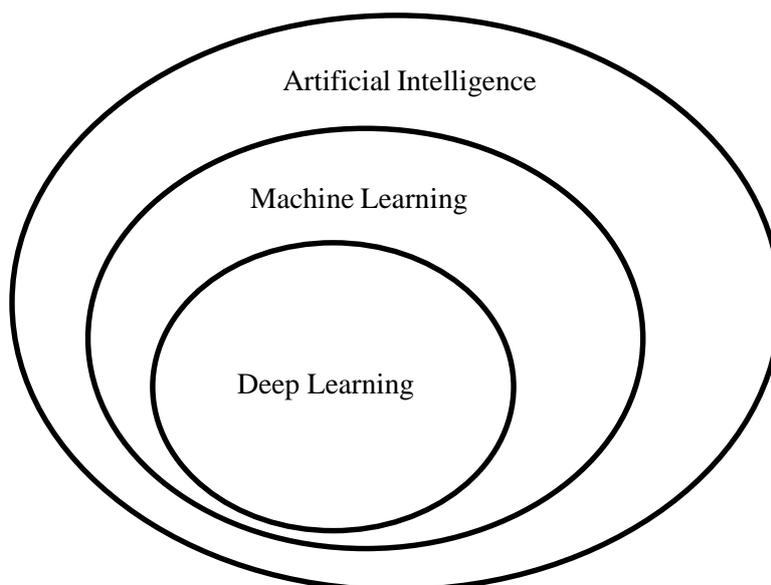


Figure 3.1. A proposed relationship between AI, machine learning, and deep learning.

fields as presently defined can occur and that some problems may involve aspects of all three fields.

	Data Science	Machine Learning	Artificial Intelligence
<b>Field:</b>			
<b>Produces:</b>	Insights	Predictions	Actions
<b>Emphasizes:</b>	<ul style="list-style-type: none"> <li>• Statistical inference</li> <li>• Data visualization</li> <li>• Experiment design</li> <li>• Domain knowledge</li> <li>• Communication</li> </ul>	<ul style="list-style-type: none"> <li>• Training data</li> <li>• Prediction accuracy over model interpretability</li> </ul>	<ul style="list-style-type: none"> <li>• Autonomy</li> </ul>
	<div style="display: flex; justify-content: center; align-items: center;"> <span style="font-size: 2em; margin-right: 10px;">←</span> <div style="background-color: #006666; color: white; border-radius: 15px; padding: 5px 15px; display: inline-block;">Deep Learning</div> <span style="font-size: 2em; margin-left: 10px;">→</span> </div>		

Figure 3.2. A proposed relationship between common statistical learning terms.

Beyond the nonchalant interchanging and synonymous usage of the field names themselves, each field has developed its own nomenclature for the basic variables and common data types. The reader may have noticed in the chapter introduction that many names for the same object exist; for example, qualitative, categorical, and ordered can all be

used to indicate that the outputs are discrete. In the literature, the choices made to describe the problem are often due to the historical origin of the problem, the objective of the work, or the use of certain classes of algorithms [26]. A small example list of synonymous terms is provided in Table 3.1.

Table 3.1. A non-exhaustive list of terms used throughout the literature which refer to the same data object or statistical learning concept.

		Field			
		Machine Learning	Statistics	Mathematics	Computer Science
Terms	inputs		predictors	independent variables	features
	outputs		targets	dependent variables	responses

While studying learning models it very likely that one will come across all forms of the terminology. Consequently, the avid student should be familiarized with the different terms and be readily able to change descriptions as required. Throughout the entirety of this text no attempt is made to use consistent nomenclature, however, notation is introduced as encountered.

### 3.2 Machine Learning Stack

Most of the computations for the machine learning problems addressed herein are performed on a HP Spectre x360 Convertible 15-bl0XX Processor ultra notebook. This notebook has a Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz, 2904 Mhz, 2 Core(s), 4 Logical Processor(s) processor and a NVIDIA GeForce 940MX graphics card.

Occasionally, high performance computing resources are used. The simulations conducted in Chapter 6 use three hundred and twenty Intel Xeon E5-2698 v3 cores. The convolutional neural network developed in Chapter 7 is trained using forty eight Intel Xeon Platinum 8168 processors alongside a NVIDIA Tesla P100 GPU.

For deep learning, Figure 3.3 shows the hardware and software stack used in this work. Keras is a well-established and user-friendly application programming interface

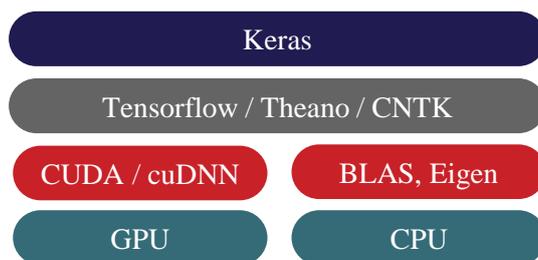


Figure 3.3. Deep learning hardware and software stack.

(API) which provides building blocks for building deep-learning models. For the backend, Keras can interface with either Tensorflow (developed by Google), Theano (developed by the MILA lab at the Université de Montréal), or the Microsoft Cognitive Toolkit (CNTK) (developed by Microsoft). For the present workflow, the Tensorflow backend is used except when the models are deployed to a web server; which use Theano on the backend. The backends can run on either a GPU or CPU. When running on the CPU the backend wraps around library for tensor operations called Eigen. Additionally the models will run faster if a Basic Linear Algebra Subprogram (BLAS) is installed. When using the GPU, NVIDIA has produced a set of drivers and a library which allow for the GPU to run a low-level parallel language for parallel computing with highly optimized routines. CUDA is the set of drivers; CUDA was originally an acronym for Compute Unified Device Architecture but NVIDIA has since stopped promoting that name and now the drivers are simply branded as CUDA. cuDNN is the library of deep learning primitives and stands for NVIDIA CUDA Deep Neural Network.

The shallow machine learning algorithms are implemented with scikit-learn [27] unless otherwise specified.

## 4. VISUALIZATION OF HOT SPOTS

### 4.1 Introduction and Motivation

Under mechanical excitation, energy is known to localize within an energetic material resulting in ‘hot spot’ formation [8]. Initiation mechanisms are closely linked to energy input type such as impact, ultrasonic excitation [28, 29], etc. Additional insight into heat dissipation mechanisms, the effect of binder/crystal interfaces, and prediction capabilities can be gained by visualizing the initiation and growth of hot spot sites and their relative intensity, spatially and temporally.

Historically, efforts to visualize hot spot formation made use of high-speed photography [8, 30, 31] or infrared arrays [32]. Recently, surface limited infrared [33, 34] and high-speed thermal microscopy [35, 36] have been employed. This work seeks to augment those methods, at potentially higher speeds and precision, with phosphor thermography.

Phosphor thermography is a well established temperature sensing technique wherein an object’s surface temperature is obtained by collecting the temperature dependent luminescence of an optically excited phosphor [37]. Typically, the phosphor is excited by a laser. If sufficient excitation energy is provided, a phosphor’s electrons are elevated to higher energy states. In order for the electrons to return to ground state the phosphor must release energy. This energy can be released via radiation, i.e., quantized photon emission, or by other non-radiative means, such as vibrational relaxation. For many phosphors, the probability of de-excitation occurring radiatively (and the nature of this radiation, e.g., luminescence lifetime and power spectrum), as opposed to other non-radiative relaxation paths, is temperature dependent. The exploitation of this fact gives these phosphors their temperature sensing, or thermographic, property.

Phosphor thermography poses several advantages over other temperature sensing techniques and devices such as pyrometry and thermocouples because it is semi-invasive, capable of producing two-dimensional measurements with high spatial resolution, and is

robust to blackbody radiation and chemiluminescence [37, 38]. There are multiple ways to relate the phosphor luminescence to temperature [39, 40]; however, two methods – the lifetime decay method, and the intensity ratio method – are most popularly employed.

The lifetime decay method temporally resolves the decay of the phosphor luminescence after excitation. The decay is typically approximated as mono-exponential [41]. The lifetime of the decay can be characterized by the time constant,  $\tau$ , in

$$I(t) = I_0 \exp\left(-\frac{t}{\tau}\right) \quad (4.1)$$

where  $I_0$  is the initial phosphor luminescence intensity. The characteristic lifetime is calibrated to temperature.

The intensity ratio method compares the relative intensity of phosphor luminescence at two spectrally separated bands as a function of temperature. The absolute signal intensity bounded by wavelengths  $\lambda_1$  and  $\lambda_2$  can be approximated by a function as [39, 41]

$$S_{\lambda_1^2}(T) \approx \int_{\lambda_1}^{\lambda_2} C_{\lambda} t_{exp} T_{\lambda} I_0 \exp\left(-\frac{\Delta E_{\lambda}}{kT}\right) d\lambda. \quad (4.2)$$

In the preceding equation,  $C_{\lambda}$  is the efficiency of the detector and  $T_{\lambda}$  is the optical filter transmission at wavelength  $\lambda$ ,  $t_{exp}$  is the exposure time,  $I_0$  is the initial pulse intensity,  $\Delta E_{\lambda}$  is the energy of the emitting state above ground state,  $k$  is the Boltzmann constant, and  $T$  is temperature. However, absolute signal intensities are difficult to measure and are plagued by non-repeatable factors such as the local thickness of phosphor coating, optical alignment, etc [41]. Many of these factors are mitigated, or eliminated, by dividing the signal obtained by one band of wavelengths by that of another band as given by

$$\begin{aligned} R(T) &= \frac{S_{\lambda_1^2}(T)}{S_{\lambda_3^4}(T)} = \frac{\int_{\lambda_1}^{\lambda_2} C_{\lambda} T_{\lambda} \exp\left(-\frac{\Delta E_{\lambda}}{kT}\right) d\lambda}{\int_{\lambda_3}^{\lambda_4} C_{\lambda} T_{\lambda} \exp\left(-\frac{\Delta E_{\lambda}}{kT}\right) d\lambda} \\ &= (\text{constant}) \exp\left(-\frac{\Delta E}{kT}\right). \end{aligned} \quad (4.3)$$

Analogous to the lifetime decay method, the ratio  $R$  defined in Eq. 4.3 can be calibrated to temperature and referenced in subsequent measurements.

Overall, the lifetime decay method is considered more precise (by several orders of magnitude) and more robust to external changes (variation in laser intensity, optical

alignment, etc.) than the intensity ratio method [42]. Due to the difference in precision and overall robustness, the intensity ratio method is best considered when measuring fast moving objects [40], or objects with fast temperature transients, where the lifetime decay method may prove inadequate.

In the present work, the intensity ratio method is demonstrated for measurement of the dynamic temperature field occurring during internal heating within an energetic composite. An octahydro 1,3,5,7-tetranitro-1,3,5,7-tetrazocine (HMX) crystal is embedded within Sylgard 184, an optically transparent binder, to form a simplified composite polymer bonded explosive (PBX). Gallium doped zinc-oxide (ZnO:Ga) phosphors are embedded in a plane in the vicinity of the HMX crystal within the Sylgard binder. Zinc oxide phosphors (ZnO:Ga, ZnO:Zn, ZnO) [43, 44] are well characterized and are highly sensitive to temperature [39]. The intensity ratio of two detection bands is employed to map the temperature field while the sample undergoes ultrasonic mechanical excitation. Although, in the presented experiment, the hot spot heating rate does not exceed the limitations of the lifetime decay approach and the object of interest is stationary, the intensity ratio method is developed herein for later use with more general testing conditions where the lifetime decay method would be inapt due to short the short time scales of hot spot formation (Bowden and Yoffe determined hot spot durations to be on the order of  $10^{-5}$ – $10^{-3}$ s [6]); e.g. drop weight impact, shock, etc. This study aims to establish the feasibility of this approach for potential measurements of internal temperature fields during highly dynamic events, such as high speed impact loading, for which the intensity ratio method is well suited. Confounding factors such as the transmissivity of Sylgard and the affect of laser fluence are considered. The objective of this work is to demonstrate a high precision, real-time, dynamic temperature measurement within a simplified PBX and to characterize its performance for the first time.

## 4.2 Experimental details

### 4.2.1 Sample preparation

The simplified PBX sample tested consisted of a countable number of HMX crystals placed on a plane of ZnO:Ga, all suspended within a Sylgard binder. To create this sample, first, a base layer of Sylgard was poured (10:1 base to curative) and cured under vacuum to remove trapped air bubbles. Next, the phosphor was mixed with Sylgard (1:15, by weight) and the mixture was spin coated (MTI Corp. VTC-50A) onto the base layer. Once the phosphor-Sylgard layer was cured, the HMX crystal(s) was placed on top of the sample and a ‘capping’ layer of Sylgard was poured and cured. The sample was then cut to size and applied to an ultrasonic transducer (Steiner & Martins, Inc. SMD10T2R111) with ultrasonic couplant (Sonotech Soundsafe) according to the desired orientation. Sample dimensions and tested orientations are shown in Fig. 4.1. Microscopic imaging of the sample cross-section (Hirox KH-8400) revealed that the applied phosphor-Sylgard layer was approximately  $30\ \mu\text{m}$  thick.

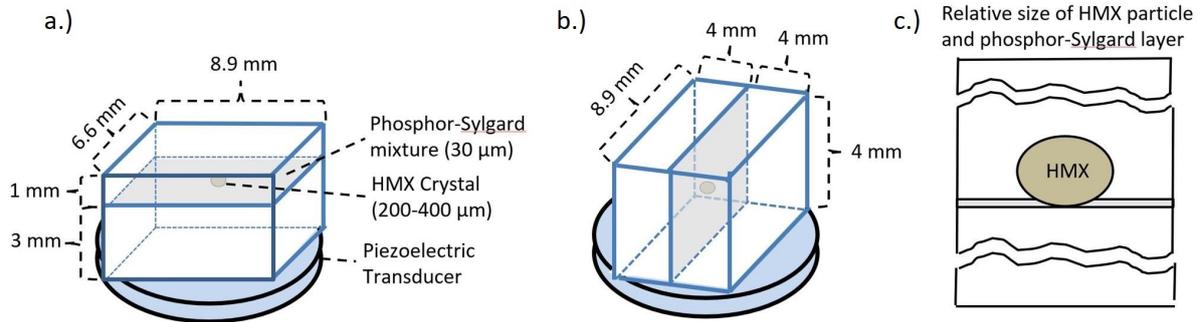


Figure 4.1. Sample geometry. a) Phosphor layer is parallel to transducer. Laser excitation enters through the top surface and the temperature field is visualized through the same. b) Phosphor layer is orthogonal to transducer. Laser incidence and camera view can be on the same side (‘in reflection’) or opposing sides (‘in transmission’). c) Depiction of the relative size of the HMX crystal and the phosphor layer.

### 4.2.2 Setup and phosphorescence imaging

The phosphor was illuminated by a 10 Hz Nd:YAG laser operating at its third harmonic (355 nm, 8 ns pulse width). A waveplate and polarizer were placed in the beam path and allowed for control of the applied laser fluence. A diffuser was placed in front of the sample to homogenize the beam profile before incidence. Depending on the orientation of the sample on the transducer, the laser beam can either contact the sample ‘in reflection’ or ‘in transmission’. The former is described as the beam contacting the sample from the same side as viewed by the camera. ‘In transmission’ refers to the beam contacting the sample from the side opposite the viewpoint of the camera; that is, the sample is placed directly between the beam and the camera system. The ‘in reflection’ mode provides greater signal, whereas, the ‘in transmission’ mode allows for perfect orthogonality between the beam, phosphor plane, and camera system – mitigating angular dependence of the beam profile and detection system.

The ultrasonic transducer received an amplified sinusoidal signal (Agilent N9310A, MiniCircuits LZY-220, Keysight E3634A) near a resonance frequency of 210.5 kHz [28]. The amplifier was fixed at +43 dB and the function generator was set to  $-3$  dBm supplying 10 W to the transducer. The signal and its duration were recorded on an oscilloscope (Tektronix DPO 4034). The phosphor luminescence was collected and spectrally separated into two images by a custom stereoscope and a single attached lens that allowed imaging of the temperature distribution occurring within sum-mm hot spots. The selected filters were  $390\pm 5$  nm (CW, FWHM) and 420+ nm (cut-on wavelength). The two images were placed onto a single camera detector by the same stereoscope. Details on the stereoscope are provided in Fig. 4.2. The images were recorded by a CMOS camera (Vision Research Phantom V7.3) synchronized with the leading edge of the laser source with a  $100\mu s$  exposure time. The entire experimental setup is seen in Fig. 4.3. After the images were acquired, a background image was subtracted and the images were spatially registered via a MATLAB program. The images were divided to form an intensity ratio image and flat fielded by division with a reference intensity ratio

image at room temperature. Example camera images of a multi-crystal sample before and after observed heating are shown in Fig. 4.4.

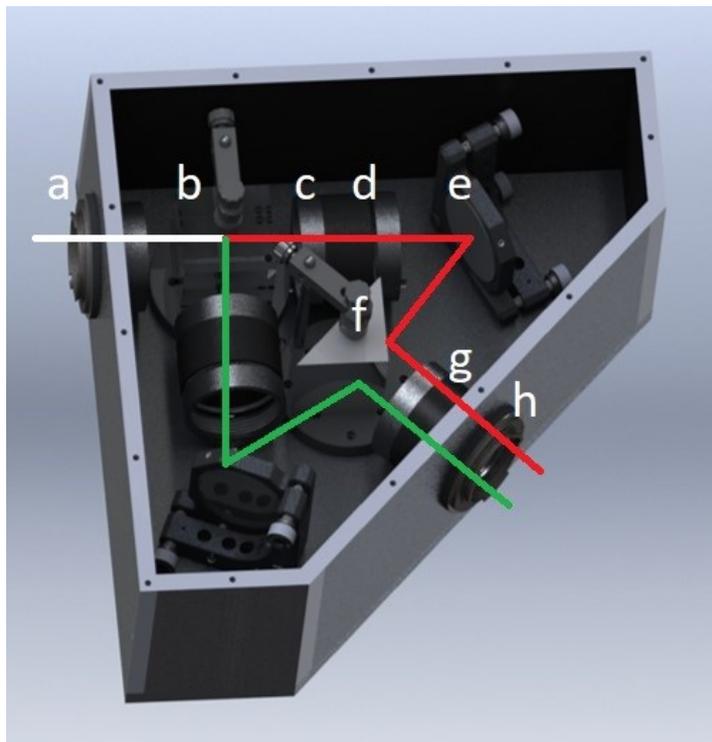


Figure 4.2. Built-in-house stereoscope. a) lens mount, b) 50:50 beam-splitter and knife edge, c) converging lens, d) filter mount, e) high transmission mirror, f) right-angle mirror, g) converging lens, h) camera mount. Colored lines: simplified image path.

### 4.2.3 Sylgard transmission and laser fluence

The transmissivity of Sylgard was explored as a potential confounding factor in temperature evaluation. Absorption of the incident UV laser beam through the volume of the sample could result in laser induced heating. To observe laser induced heating a test sample was placed in the beam path and the transducer was not engaged. During this time the sample was observed by an infrared camera (FLIR A320) operating at 30 Hz with an assumed emissivity of 1. The surface temperature was noted to rise  $\approx 25^{\circ}\text{C}$

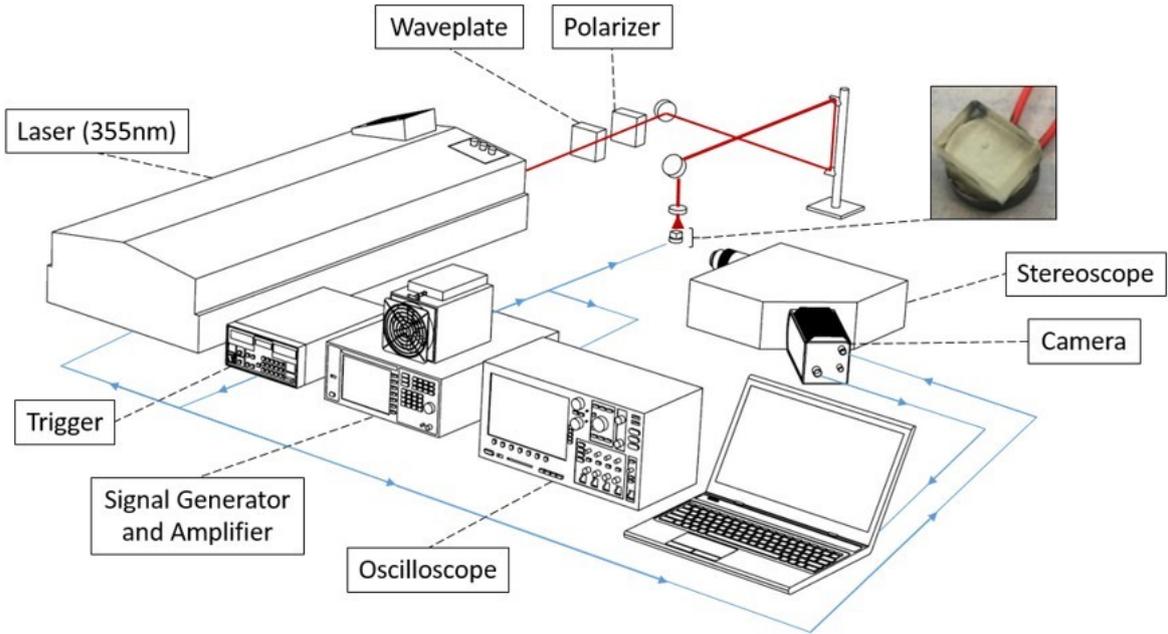


Figure 4.3. Experimental setup.

over a time period of  $\approx 90s$  before reaching steady state. This rate of heating could be significant for some cases but does not affect the conclusion for short-time dynamics.

Additionally, the Sylgard could absorb the phosphor luminescence. Preferential transmission of one spectral band over another could adversely affect sensitivity to temperature for a given filter selection. To explore this effect, a clear block of Sylgard (without phosphor) was placed in a spectrophotometer (Agilent Cary 6000i UV-Vis-NIR) and a transmission efficiency profile was generated. The spectrophotometer chamber temperature was varied between  $0^{\circ}C$  and  $100^{\circ}C$ . The transmission profile was relatively flat over the near-UV and visible wavelengths and did not change with temperature over the prescribed range. Consequently, we concluded that Sylgard is an acceptable candidate for making phosphor thermography measurements from within the sample volume.

Laser fluence has been identified as a confounding factor in temperature measurements for many phosphors. Abram et al. [44] investigated the effect of laser fluence on ZnO phosphor and its effect was briefly considered here. The ZnO:Ga intensity ratio

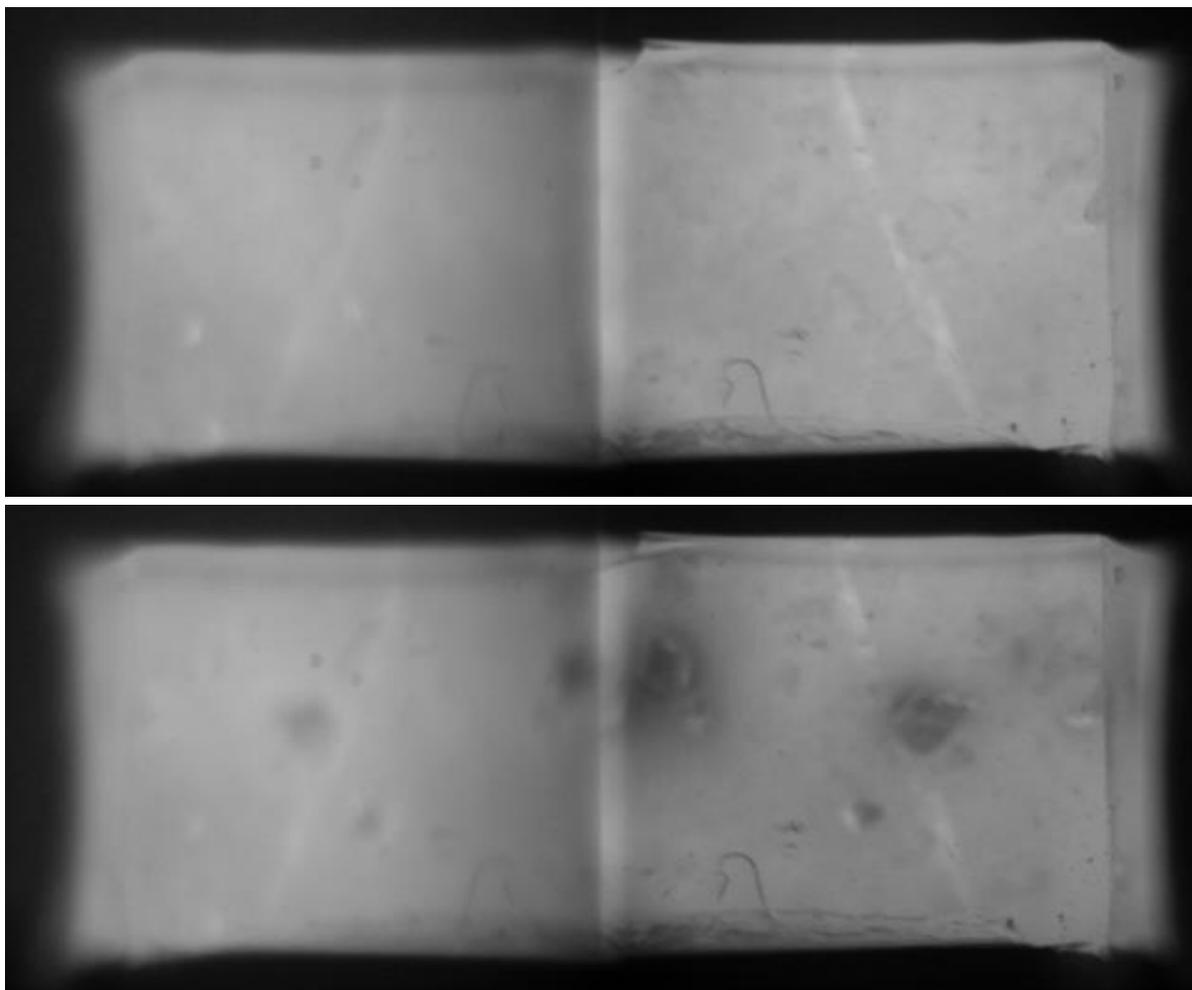


Figure 4.4. Example image of a mult-crystal sample. The left side of both images is the sample visualized through the  $420+$  nm filter and the right side is the image replicated (and reflected) through the  $390\pm 5$  nm filter. Top: before activation of transducer. Bottom: 10 s after activation of transducer. Internal heating heating is observed where the right image pane becomes darkened relative to the left image pane (relative decrease in  $390\pm 5$  nm filtered signal strength compared to  $420+$  nm filtered signal). Sample height is approximately 4 mm.

was recorded at various temperatures and laser fluences by both the experimental setup previously described – with a stereoscope and camera – and with a spectrometer (Ocean Optics USB2000+UV-VIS). When determining the intensity ratio with the spectrometer, the power spectrum was numerically integrated over the filter bands assuming a perfect band profile and the results are given in Fig. 4.5. The intensity ratio determined by the camera system agreed with the spectrometer results after multiplication by a constant scalar. The discrepancy between spectrometer and camera intensity ratios is attributed to non-perfect filters and poor transmission efficiency of the  $390\text{nm}$  light in the stereoscope. Conforming with Abram et al. we observed that the intensity ratio is dependent on laser fluence, and moreover point out that the phosphor is less sensitive to temperature at lower fluences. For the presented tests, the laser fluence was kept at  $49 \frac{\text{mJ}}{\text{cm}^2}$ .

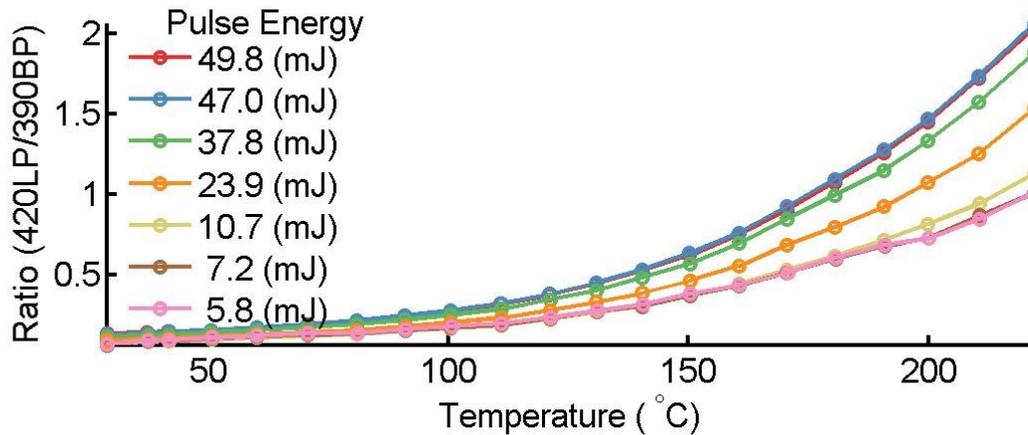


Figure 4.5. Effect of laser pulse energy on intensity ratio. Beam diameter was  $2\text{cm}$ .

#### 4.2.4 Calibration

Developing a calibration curve relating observed intensity ratio to temperature requires accurate knowledge of temperature. To this end, a custom, optically accessible furnace was procured (Mellen Company) allowing precise temperature control with few

spatial variations. A calibration sample was surrounded by a copper block containing three type-K thermocouples embedded in close proximity to the sample. The copper block served to dampen any spatial or temporal fluctuations in temperature. The temperature, indicated by the thermocouples, was recorded at each shot while the furnace was slowly heated. Use of the copper block and automated calibration routine was inspired by Nada et al. [45]. The thermocouple readings were never observed to differ by more than  $1^{\circ}\text{C}$ . For each image, a histogram of intensity ratios was generated and the average thermocouple temperature was recorded. A top-down view of these histograms evolving with temperature is given in Fig. 4.6. Select histograms are also called out in the figure inset. A calibration curve was created by fitting the mean intensity ratio of each image against temperature. With the calibration curve, the intensity ratio of 100 averaged images at various temperatures (determined by the thermocouples) were converted to temperature and subsequent temperature histograms. That is, each histogram is composed from the pixel ratio values after averaging of 100 images acquired at a single temperature. A selection of these histograms are seen in Fig. 4.7 alongside the thermocouple temperature.

The normalized standard deviation of the phosphor-determined temperature is also given in Fig. 4.7. Although there is greater variation in intensity ratios at elevated temperatures, the phosphor also has greater sensitivity in this range. Consequently, the normalized variation in temperature determined by the phosphor system is nearly constant over the given temperature range. The authors note that because the 100 images were averaged, the presented normalized standard deviation only serves as an uncertainty estimate spatially. Also, this estimate is optimistic because it and the calibration curve were generated by the same data. In the future, the authors prescribe compiling (as opposed to averaging) the intensity ratios over a range of shots at selected temperatures on a new calibration sample. This method would provide an accurate estimate of the temperature uncertainty accounting for both spatial and temporal variations. Finally, an optical rail system was used so that the calibration sample could be slid out of the furnace and the sample for testing could be slid into the same spot as the calibration

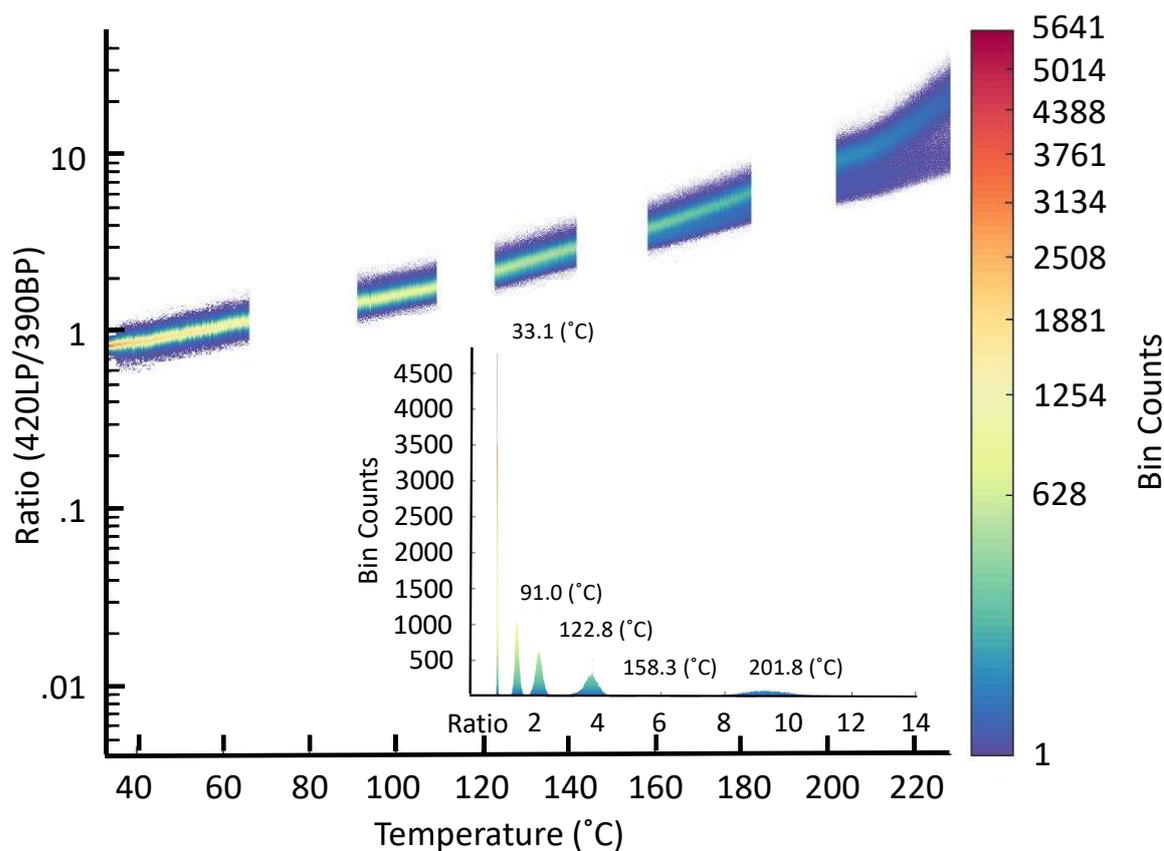


Figure 4.6. Top-down view of intensity ratio histograms to temperature. Breaks in curve occur when the camera systems stops recording to save data. Inset: Alternative view of histograms at select temperatures show narrower distributions at lower temperatures.

sample (with the furnace off and cool) so that no readjustment or refocusing was needed between calibration and testing.

### 4.3 Results and discussion

An example image of a hot spot evolving near an HMX crystal under ultrasonic mechanical excitation is shown in Fig. 4.8. In this case, the laser beam is illuminating the phosphors ‘in transmission’ and the phosphor plane is oriented vertically – orthogonal

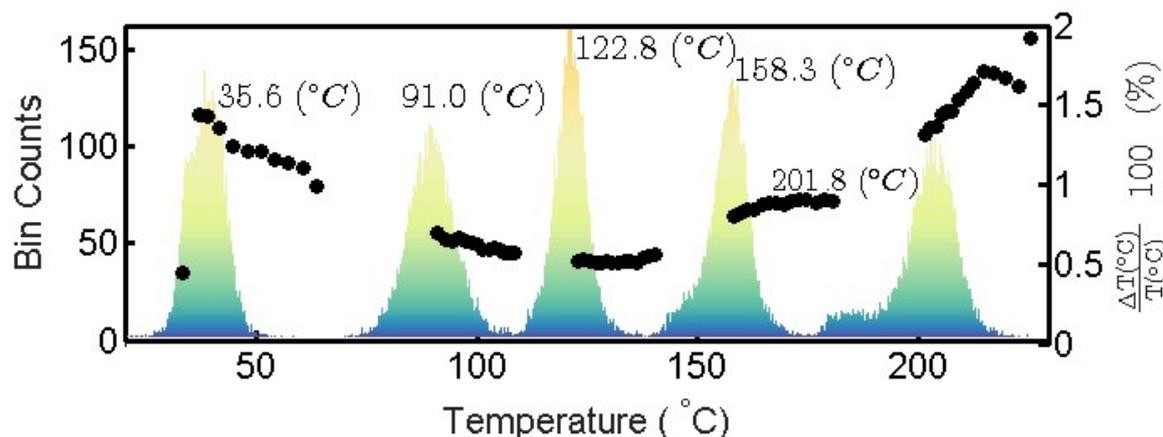


Figure 4.7. Left axis: Temperature histograms of 100 averaged frames at select thermocouple temperatures. Right axis: (symbol: ●) Normalized standard deviation.

to the transducer. The temperature is seen to rise in the vicinity of the HMX crystal, and this event occurs at elevated temperatures and in a short time span indicating that heat conduction from the transducer surface is not yet significant.

For the same example, Fig. 4.9 demonstrates how the hot spot evolves in time. In this figure, before the hot spot is initiated, the average sample temperature and the spatial standard deviation are plotted. Once hot spot initiation has occurred, the maximum temperature over the sample is shown. Before the hot spot initiates, the mean sample temperature steadily increases. The rate of the bulk temperature increase is on the same order as that observed by an infrared camera while testing for laser induced heating, and is subsequently attributed to slight laser heating. It is also noted that while the spatial standard deviation of the recorded temperature (width of the errorbars) remains fairly constant during this period, the temperature varies by 1-2°C from frame-to-frame. This is likely due to shot-to-shot fluctuations in laser energy, which was not accounted for during testing. This could be corrected using procedures outlined by Abram et al. [44]; however, even with this temporal deviation, the error in temperature is small

compared to the hot spot temperature and the relaxed precision still marks a significant improvement over other measurement attempts.

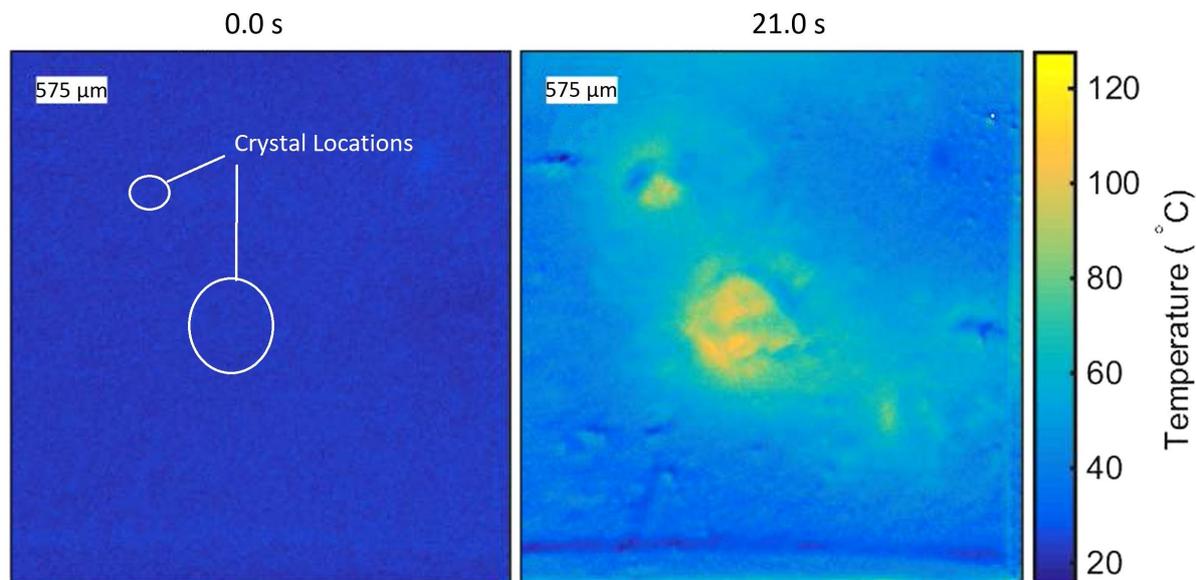


Figure 4.8. Visualized temperature field at selected frames. Frame field of view is  $4\text{mm}$  by  $4\text{mm}$ .

#### 4.4 Conclusion

The intensity ratio method of phosphor thermography was performed in order to remotely measure temperature from within a simplified PBX undergoing mechanical excitation. Sylgard 184 was used as the binder in the PBX due to its optical clarity and was not shown to have adverse affects on the temperature measurement or sensitivity. ZnO:Ga phosphor sensitivity and calibration are shown to be dependent on incident fluence and should be considered in future designs. Spatially, the pixel-to-pixel standard deviation was shown to be less than 2% for the prescribed temperatures  $\leq 220\text{ }^{\circ}\text{C}$ . The temporal, frame-to-frame, standard deviation was not computed, but observed measurement fluctuations are likely attributed to shot-to-shot fluctuations in laser energy and can be corrected in future tests. The spatially and temporally resolved temperature field

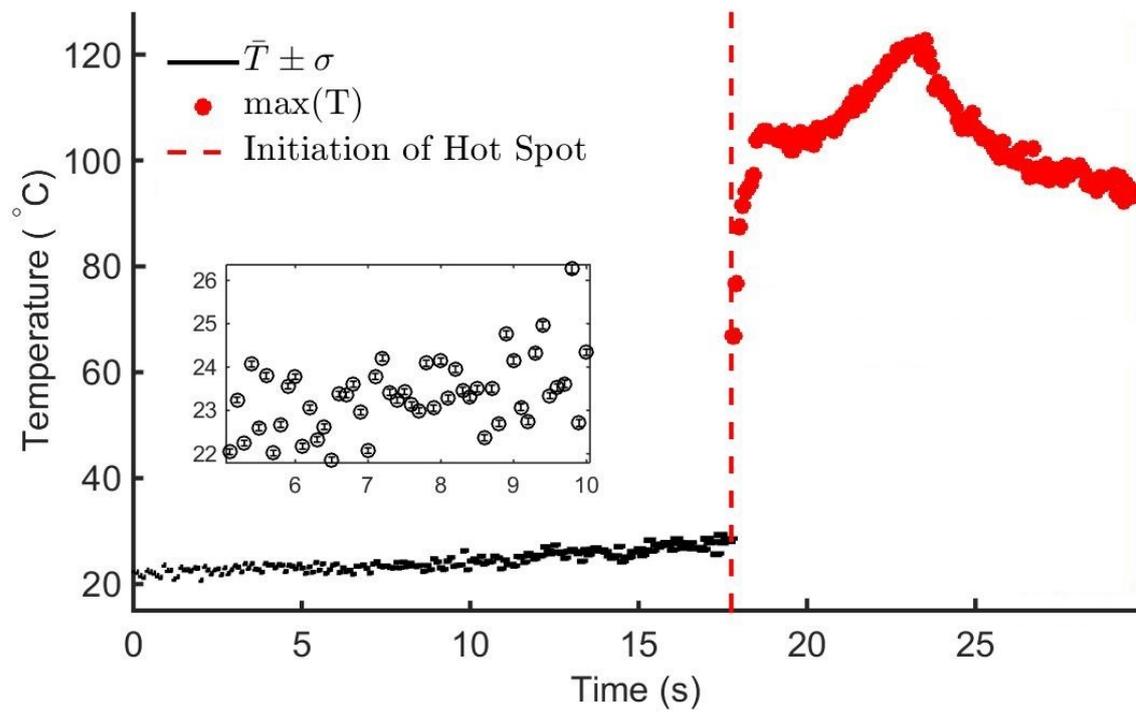


Figure 4.9. Plot detailing the temporal history of the hot spot. Inset: Magnified view of the average temperature and standard deviation before hot spot occurrence.

of an evolving hot spot can be used to shed light on the underlying physical phenomena governing hot spot initiation and growth.

## 5. SENSITIVITY TESTS

### 5.1 Neyer Sensitivity Test

Sensitivity tests are commonly used to estimate hidden (or latent) distribution parameters. For example, upon producing a batch of explosives it is assumed that each explosive specimen has a critical threshold value that, upon surpassing, will result in a positive response (detonation, a ‘go’). The critical threshold values are assumed to be a continuous variable with an unknown probability density function. The difficulty of this problem arises because the critical threshold of each specimen cannot be directly measured. Consider the case of determining the yield or fracture threshold of a given material. These thresholds can be directly measured by slowly increasing the stress (or rather force) applied to the specimen with a tensile testing machine and noting the stress at which yielding or fracture occurs. An analogous test does not exist when trying to determine the critical threshold values of explosives.

When testing the sensitivity of explosives to impact or shock, one can impose a non-varying stress level (input pressure, drop-weight height, etc.) on the test article (explosive specimen) and note whether a response (detonation, ‘go’ or ‘no-go’) occurs. Unlike the fracture case, the input level cannot be slowly augmented until detonation is achieved. Additionally, if a positive response is not achieved the specimen cannot simply be re-tested at a higher input level because the specimen may have been damaged by the original test. Damage (void creation, phase change, etc.) is theorized to alter the sensitivity, or critical threshold, of the specimen – which is obviously unwanted since this is exactly the value one wishes to measure.

**Thus, sensitivity tests are herein defined as methods to estimate the parameters which prescribe the probability density function of critical threshold values; where these critical thresholds cannot be directly measured (hidden, latent).**

In 1989, Barry T. Neyer published a paper outlining a ‘new sensitivity’ test and detailed its advantages over previous methods such as Probit, Bruce-Ton, Robbins-Monro, and Langlie [46]. Later, in 1994, Neyer published an updated version of his sensitivity test; altering the sequential design algorithm (described later) [47]. The test assumes that the distribution of critical thresholds is modelled by a normal or logistic probability density function (pdf), or can be easily transformed to make the distribution normal. Normal and logistic distributions are parameterized by two variables,  $\mu$  and  $\sigma$ . Neyer estimates these parameters using maximum likelihood estimators. As a point of notation,  $\hat{\mu}$  and  $\hat{\sigma}$  are the *estimates* or *estimators* of the true parameters  $\mu$  and  $\sigma$ . An in-depth description of the maximum likelihood estimation to sensitivity tests is attributed to Cornfield and Mantel [48] and Golub and Grubbs [49] - with the method originating in earlier publications by Dixon and Mood [50] and by Finney [51], Bliss [52], and Fisher [53].

Once the maximum likelihood estimates (MLEs) have been calculated, Neyer proposed a sequential design in which the next stimulus level to be tested is that which maximizes the determinate of the Fisher information matrix [47]. This essentially attempts to simultaneously minimize the variances of the estimated parameters,  $\hat{\mu}$  and  $\hat{\sigma}$ , by decreasing the Cramér-Rao lower bound (CRLB) of said parameters. Although, as will be described later, the MLE of  $\sigma$  is not unbiased - and therefore is not bounded by the CRLB - minimizing the CRLB will generally decrease the variance of  $\hat{\sigma}$  because they are computed from the same likelihood function.

‘Unique’ maximum likelihood estimates will not be obtained unless the stimulus level of the smallest positive response is less than the stimulus level of the greatest non-response (no “overlap” in responses) [54]. That is, the MLE will return a value of zero for  $\hat{\sigma}$  and any value between the greatest non-response level and smallest positive response level for  $\hat{\mu}$ . Neyer’s sequential design algorithm overcomes this deficiency by using an expandable binary-search algorithm to test points until overlap is achieved.

While much of the information outlined in Neyer’s papers - in terms of the statistics - was previously published, Neyer greatly impacted the field by bringing the test to the energetic materials community, comparing the test to all other known methods,

creating a robust sequential design algorithm, and producing a commercial software implementation of the test [55].

Specific descriptions of the maximum likelihood estimators, the information matrix, and the sequential design algorithm are given below.

### 5.1.1 Maximum Likelihood Estimators

Here, we will use the notation used by Neyyer and the previous authors. In section 5.2, the notation will be modernized.

Let  $L^{(i)}$  represent the stimulus level to be tested on specimen  $i$ . The standardized level is then given by  $z^{(i)} = \frac{L^{(i)} - \mu}{\sigma}$ , where  $\mu$  and  $\sigma$  are the parameters that describe the assumed normal distribution of critical threshold values. Then the probability of observing a positive response in a randomly selected specimen is

$$p(z^{(i)}) = \int_{-\infty}^{z^{(i)}} f(t) dt \quad (5.1)$$

where,

$$f(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \quad (5.2)$$

Let  $y^{(i)}$  represent the observed response at stimulus level  $L^{(i)}$ ; and let  $y^{(i)} = 1$  and  $y^{(i)} = 0$  for a positive and negative response ('go', 'no go'), respectively. Because the response is described by a binary variable, the likelihood function follows the form of a binomial model

$$L(\mu, \sigma) = \prod_{i=1}^n p(z^{(i)})^{y^{(i)}} (1 - p(z^{(i)}))^{1-y^{(i)}} \quad (5.3)$$

where  $n$  is the number of specimens tested. If the same stimulus level is tested more than once then let  $N^{(i)}$  and  $M^{(i)}$  represent the number of positive and negative responses observed at  $L^{(i)}$ , respectively. Then, the total number of test performed at level  $L^{(i)}$  is equal to  $(N^{(i)} + M^{(i)})$ . Using this notation, equation 5.3 can be written as

$$L(\mu, \sigma) = \prod_{i=1}^n \binom{N^{(i)} + M^{(i)}}{N^{(i)}} p(z^{(i)})^{N^{(i)}} (1 - p(z^{(i)}))^{M^{(i)}} \quad (5.4)$$

Additionally,  $(1 - p(z^{(i)}))$  is equivalent to  $p(-z^{(i)})$  and can be referred to as  $q^{(i)}$  so that the likelihood function is

$$L(\mu, \sigma) = \prod_{i=1}^n \binom{N^{(i)} + M^{(i)}}{N^{(i)}} p(z^{(i)})^{N^{(i)}} q(z^{(i)})^{M^{(i)}} \quad (5.5)$$

The values of  $\mu$  and  $\sigma$  that maximize the likelihood function for the data set  $\{(L^{(1)}, N^{(1)}, M^{(1)}), (L^{(2)}, N^{(2)}, M^{(2)}), \dots, (L^{(n)}, N^{(n)}, M^{(n)})\}$  are the MLEs  $\hat{\mu}$  and  $\hat{\sigma}$ . Note that in the notation of equation 5.5,  $n$  now represents the number of unique stimulus levels tested and not the total number of tests performed.

The parameters that maximize the likelihood function are the same as those that maximize the natural logarithm of the likelihood function because the natural logarithm is a monotonically increasing function. The log-likelihood function is often easier to work with analytically, and additionally, it is the log-likelihood function that is used in the definition of the Fisher information matrix used later. Thus, the log-likelihood function is

$$l = \log(L) = \sum_{i=1}^n \left[ \log \binom{N^{(i)} + M^{(i)}}{N^{(i)}} + N^{(i)} \log(p(z^{(i)})) + M^{(i)} \log(q(z^{(i)})) \right]. \quad (5.6)$$

The first order partial derivatives of equation 5.6 are useful to compute the MLE, as all of the first order partial derivatives must equal zero when  $l$  is maximized. They are derived as follows:

$$\frac{\partial l}{\partial \mu} = \sum_{i=1}^n \left[ N^{(i)} \frac{\partial (\log(p(z^{(i)}))}{\partial p(z^{(i)})} \frac{\partial p(z^{(i)})}{\partial z^{(i)}} \frac{\partial z^{(i)}}{\partial \mu} + M^{(i)} \frac{\partial (\log(q(z^{(i)}))}{\partial q(z^{(i)})} \frac{\partial q(z^{(i)})}{\partial z^{(i)}} \frac{\partial z^{(i)}}{\partial \mu} \right] \quad (5.7)$$

$$\frac{\partial l}{\partial \sigma} = \sum_{i=1}^n \left[ N^{(i)} \frac{\partial (\log(p(z^{(i)}))}{\partial p(z^{(i)})} \frac{\partial p(z^{(i)})}{\partial z^{(i)}} \frac{\partial z^{(i)}}{\partial \sigma} + M^{(i)} \frac{\partial (\log(q(z^{(i)}))}{\partial q(z^{(i)})} \frac{\partial q(z^{(i)})}{\partial z^{(i)}} \frac{\partial z^{(i)}}{\partial \sigma} \right] \quad (5.8)$$

As seen, equations 5.7 and 5.8 are derived by the continuous, albeit tedious, application of the ‘chain rule’ in calculus. The partial derivatives found in these equations are formulated as

$$\frac{\partial (\log(p(z^{(i)}))}{\partial p(z^{(i)})} = \frac{1}{p(z^{(i)})} \quad (5.9)$$

$$\frac{\partial (\log(q(z^{(i)}))}{\partial q(z^{(i)})} = \frac{1}{q(z^{(i)})} \quad (5.10)$$

$$\frac{\partial p(z^{(i)})}{\partial z^{(i)}} = f(z^{(i)}) \quad (5.11)$$

$$\frac{\partial(z^{(i)})}{\partial z^{(i)}} = -f(z^{(i)}) \quad (5.12)$$

$$\frac{\partial z^{(i)}}{\partial \mu} = \frac{-1}{\sigma} \quad (5.13)$$

$$\frac{\partial z^{(i)}}{\partial \sigma} = \frac{-(L^{(i)} - \mu)}{\sigma^2} \quad (5.14)$$

Substituting equations 5.9 – 5.14 into equations 5.7 and 5.8 yields

$$\frac{\partial l}{\partial \mu} = \sum_{i=1}^n \left[ N^{(i)} \frac{1}{p(z^{(i)})} f(z^{(i)}) \frac{-1}{\sigma} + M^{(i)} \frac{1}{q(z^{(i)})} f(z^{(i)}) \frac{1}{\sigma} \right] \quad (5.15)$$

$$\frac{\partial l}{\partial \sigma} = \sum_{i=1}^n \left[ N^{(i)} \frac{1}{p(z^{(i)})} f(z^{(i)}) \frac{-(L^{(i)} - \mu)}{\sigma^2} + M^{(i)} \frac{1}{q(z^{(i)})} f(z^{(i)}) \frac{L^{(i)} - \mu}{\sigma^2} \right] \quad (5.16)$$

These derivatives are useful as they will almost certainly be used in any optimization routine or algorithm in order to calculate the parameter values which maximize  $l$ .

### 5.1.2 Information Matrix

The Fisher information matrix is defined as

$$INF = -E [H[\log(L)]] \quad (5.17)$$

where  $E[\cdot]$  is the expectation operator,  $H[\cdot]$  is the Hessian, and  $L$  is the likelihood function. The log-likelihood function was previously defined as  $l$  and the Hessian is a matrix of second partial derivatives. The partial derivatives are taken with respect to the parameters being estimated; in this case,  $\mu$  and  $\sigma$ . Therefore,

$$H [\log(L)] = H [l] = \begin{bmatrix} \frac{\partial^2 l}{\partial \mu^2} & \frac{\partial^2 l}{\partial \sigma \partial \mu} \\ \frac{\partial^2 l}{\partial \mu \partial \sigma} & \frac{\partial^2 l}{\partial \sigma^2} \end{bmatrix} \quad (5.18)$$

Let the elements of the information matrix be described as

$$INF = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (5.19)$$

and the inverse of the information matrix be

$$[INF]^{-1} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \quad (5.20)$$

Under this description  $b_{11}$  and  $b_{22}$  are the Cramér-Rao lower bounds (CRLBs) for  $\hat{\mu}$  and  $\hat{\sigma}$ , respectively. However, the CRLB only applies to estimators that are unbiased. If  $\hat{\mu}$  is the estimator for  $\mu$  and  $\hat{\sigma}$  for  $\sigma$ , then  $\hat{\mu}$  and  $\hat{\sigma}$  are unbiased estimators of  $\mu$  and  $\sigma$  if  $E[\hat{\mu}] = \mu$  and  $E[\hat{\sigma}] = \sigma$ . Rather than attempt to solve for  $\hat{\mu}$  and  $\hat{\sigma}$  when equations 5.15 and 5.16 are equal to zero and then take the expectation of those expressions, a simple simulation is run with synthetic data.

Consider the latent random variable, which represents the critical threshold values, to be given by a normal distribution  $\sim N(\mu, \sigma)$ . For the simulation, let  $\mu = 40$  and  $\sigma = 3$ .  $n$  stimulus levels are tested; stimulus levels are spaced equidistant from each other between the range of [41, 49]. The outcome of each test is decided randomly via a Monte-Carlo simulation. For every value of  $n$  the MLE of  $\hat{\mu}$  and  $\hat{\sigma}$  is calculated and this simulation is repeated 2000 times (holding  $n$  constant); the values of  $\hat{\mu}$  and  $\hat{\sigma}$  are averaged together in order to estimate  $E[\hat{\mu}]$  and  $E[\hat{\sigma}]$ . The results for  $n = 3$  to 39 are shown in Figure 5.1.2. This figure demonstrates that  $\hat{\mu}$  is an unbiased estimator but  $\hat{\sigma}$  is a biased estimator; although the bias decreases with increasing sample size  $n$ . Neyyer estimated the bias of  $\hat{\sigma}$  over a wide variety of designs [47](permutations of good and poor initial guesses of the parameters) and approximated the relative bias as

$$relative\ bias = \frac{-3.5}{n}. \quad (5.21)$$

However, despite the biasedness of  $\hat{\sigma}$ , minimizing the CRLB will decrease the variance of  $\hat{\sigma}$  since both the CRLB and  $\hat{\sigma}$  are derived from the same likelihood function.

Moving forward, minimizing the variance of  $\hat{\mu}$  and  $\hat{\sigma}$ , or decreasing the values of  $b_{11}$  and  $b_{22}$ , is related to maximizing the values of  $a_{11}$  and  $a_{22}$  in the information matrix (*see last paragraph of this section*). Referring back to equations 5.17 and 5.18, it can be seen that these terms are

$$a_{11} = -E \left[ \frac{\partial^2 l}{\partial \mu^2} \right] \quad (5.22)$$

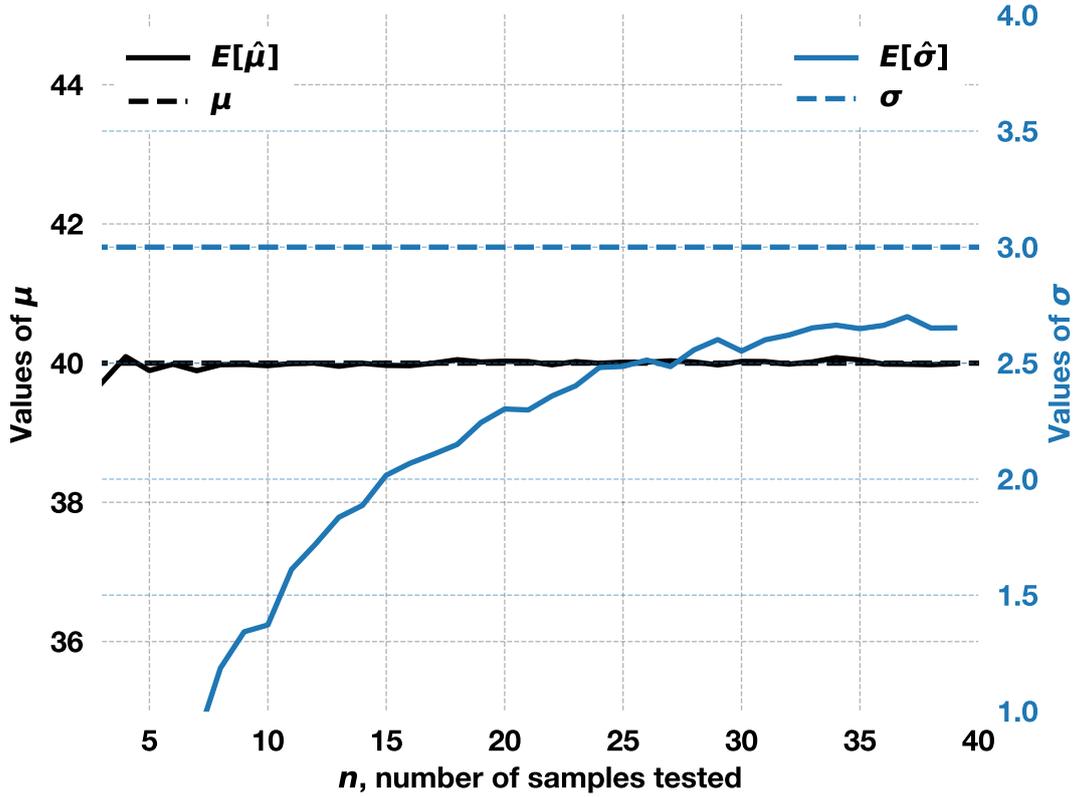


Figure 5.1. Expected values of estimators  $\hat{\sigma}$  and  $\hat{\mu}$  for simulated data

$$a_{22} = -E \left[ \frac{\partial^2 l}{\partial \sigma^2} \right] \quad (5.23)$$

recalling that  $l$  is the log-likelihood function given by equation 5.6. A derivation of the  $a_{11}$  term is provided here starting from equation 5.15. The end result of the  $a_{12}$  and  $a_{22}$  terms are given but an explicit derivation is not provided as these follow a similar form to the derivation of  $a_{11}$ .

$$\frac{\partial^2 l}{\partial \mu^2} = \sum_{i=1}^n \left[ N^{(i)} \frac{-1}{\sigma} \frac{\partial \left( \frac{f(z^{(i)})}{p(z^{(i)})} \right)}{\partial \mu} + M^{(i)} \frac{-1}{\sigma} \frac{\partial \left( \frac{-f(z^{(i)})}{q(z^{(i)})} \right)}{\partial \mu} \right] \quad (5.24)$$

$$\frac{\partial \left( \frac{f(z^{(i)})}{p(z^{(i)})} \right)}{\partial \mu} = \frac{1}{p(z^{(i)})} \frac{\partial (f(z^{(i)}))}{\partial \mu} + f(z^{(i)}) \frac{-1}{(p(z^{(i)}))^2} \frac{\partial (p(z^{(i)}))}{\partial \mu} \quad (5.25)$$

$$\frac{\partial \left( \frac{-f(z^{(i)})}{q(z^{(i)})} \right)}{\partial \mu} = \frac{-1}{q(z^{(i)})} \frac{\partial (f(z^{(i)}))}{\partial \mu} + f(z^{(i)}) \frac{1}{(q(z^{(i)}))^2} \frac{\partial (q(z^{(i)}))}{\partial \mu} \quad (5.26)$$

$$\frac{\partial (f(z^{(i)}))}{\partial \mu} = \frac{\partial f(z^{(i)})}{\partial z^{(i)}} \frac{\partial z^{(i)}}{\partial \mu} = z^{(i)} f(z^{(i)}) \frac{1}{\sigma} \quad (5.27)$$

$$\frac{\partial (p(z^{(i)}))}{\partial \mu} = \frac{\partial (p(z^{(i)}))}{\partial z^{(i)}} \frac{\partial z^{(i)}}{\partial \mu} = f(z^{(i)}) \frac{-1}{\sigma} \quad (5.28)$$

$$\frac{\partial (q(z^{(i)}))}{\partial \mu} = \frac{\partial (q(z^{(i)}))}{\partial z^{(i)}} \frac{\partial z^{(i)}}{\partial \mu} = f(z^{(i)}) \frac{1}{\sigma} \quad (5.29)$$

Making the substitutions results in

$$\begin{aligned} \frac{\partial^2 l}{\partial \mu^2} = \sum_{i=1}^n \left\{ N^{(i)} \frac{-1}{\sigma} \left[ \frac{1}{p(z^{(i)})} z^{(i)} f(z^{(i)}) \frac{1}{\sigma} + f(z^{(i)}) \frac{1}{(p(z^{(i)}))^2} f(z^{(i)}) \frac{1}{\sigma} \right] + \right. \\ \left. M^{(i)} \frac{-1}{\sigma} \left[ \frac{-1}{q(z^{(i)})} z^{(i)} f(z^{(i)}) \frac{1}{\sigma} + f(z^{(i)}) \frac{1}{(q(z^{(i)}))^2} f(z^{(i)}) \frac{1}{\sigma} \right] \right\}. \end{aligned} \quad (5.30)$$

And now, after some simplification the expression becomes

$$\begin{aligned} \frac{\partial^2 l}{\partial \mu^2} = \sum_{i=1}^n \left\{ N^{(i)} \frac{1}{\sigma^2} \frac{1}{p(z^{(i)})} f(z^{(i)}) \left[ -z^{(i)} - \frac{f(z^{(i)})}{p(z^{(i)})} \right] + \right. \\ \left. M^{(i)} \frac{1}{\sigma^2} \frac{1}{q(z^{(i)})} f(z^{(i)}) \left[ +z^{(i)} - \frac{f(z^{(i)})}{q(z^{(i)})} \right] \right\}. \end{aligned} \quad (5.31)$$

Applying the expectation operator and negating

$$\begin{aligned} -E \left[ \frac{\partial^2 l}{\partial \mu^2} \right] = \sum_{i=1}^n \left\{ E[N^{(i)}] \frac{1}{\sigma^2} \frac{1}{p(z^{(i)})} f(z^{(i)}) \left[ +z^{(i)} + \frac{f(z^{(i)})}{p(z^{(i)})} \right] + \right. \\ \left. E[M^{(i)}] \frac{1}{\sigma^2} \frac{1}{q(z^{(i)})} f(z^{(i)}) \left[ -z^{(i)} + \frac{f(z^{(i)})}{q(z^{(i)})} \right] \right\}. \end{aligned} \quad (5.32)$$

and noting that

$$E[N^{(i)}] = p(z^{(i)})(N^{(i)} + M^{(i)}) \quad (5.33)$$

$$E[M^{(i)}] = q(z^{(i)})(N^{(i)} + M^{(i)}) \quad (5.34)$$

after substitution the expression becomes

$$-E \left[ \frac{\partial^2 l}{\partial \mu^2} \right] = \sum_{i=1}^n \left\{ (N^{(i)} + M^{(i)}) \frac{1}{\sigma^2} f(z^{(i)}) \left[ z^{(i)} + \frac{f(z^{(i)})}{p(z^{(i)})} - z^{(i)} + \frac{f(z^{(i)})}{q(z^{(i)})} \right] \right\}. \quad (5.35)$$

This expression is further simplified to its final form

$$a_{11} = -E \left[ \frac{\partial^2 l}{\partial \mu^2} \right] = \sum_{i=1}^n \left\{ (N^{(i)} + M^{(i)}) \frac{1}{\sigma^2} f(z^{(i)})^2 \left[ \frac{1}{p(z^{(i)})} + \frac{1}{q(z^{(i)})} \right] \right\}. \quad (5.36)$$

The  $a_{12}$  (same as  $a_{21}$ ) and  $a_{22}$  terms are given as

$$a_{12} = -E \left[ \frac{\partial^2 l}{\partial \mu^2} \right] = \sum_{i=1}^n \left\{ (N^{(i)} + M^{(i)}) \frac{z^{(i)}}{\sigma^2} f(z^{(i)})^2 \left[ \frac{1}{p(z^{(i)})} + \frac{1}{q(z^{(i)})} \right] \right\}. \quad (5.37)$$

$$a_{22} = -E \left[ \frac{\partial^2 l}{\partial \mu^2} \right] = \sum_{i=1}^n \left\{ (N^{(i)} + M^{(i)}) \frac{(z^{(i)})^2}{\sigma^2} f(z^{(i)})^2 \left[ \frac{1}{p(z^{(i)})} + \frac{1}{q(z^{(i)})} \right] \right\}. \quad (5.38)$$

Next, these ‘a terms’ ( $a_{11}$ ,  $a_{12}$ ,  $a_{22}$ ) are plotted in Figure 5.1.2 for a single specimen. This plot demonstrates that  $a_{11}$  and  $a_{22}$  cannot be maximized simultaneously. However, a good amount of information about  $\mu$  and  $\sigma$  is achieved if the test is conducted at  $\mu \pm \sigma$ . Quoting directly from Neyr [47],

A D-optimal result will be obtained when the determinant of the information matrix is maximized.

...

Since the off-diagonal terms [ $a_{12}$  and  $a_{21}$  here] of the matrix are typically small compared to the diagonal terms, a D-optimal test will also approximately minimize the the product of the asymptotic variances of both  $\mu$  and  $\sigma$ .

Thus, when choosing the next stimulus level to test in a sequential design of experiments, Neyr’s algorithm elects the level which maximizes the determinate of the information matrix because this relates to minimizing the variances of  $\hat{\mu}$  and  $\hat{\sigma}$ .

### 5.1.3 Sequential Design

As noted in section 5.1.2, Neyr’s sensitivity test chooses the stimulus level which maximizes the information matrix. However, in order to compute the information matrix, at least one point needs to be tested and estimates of  $\mu$  and  $\sigma$  are required.

When beginning an experiment, it is possible that the experimenter has no prior knowledge about  $\mu$  or  $\sigma$ ; this scenario becomes more likely when testing a new material.

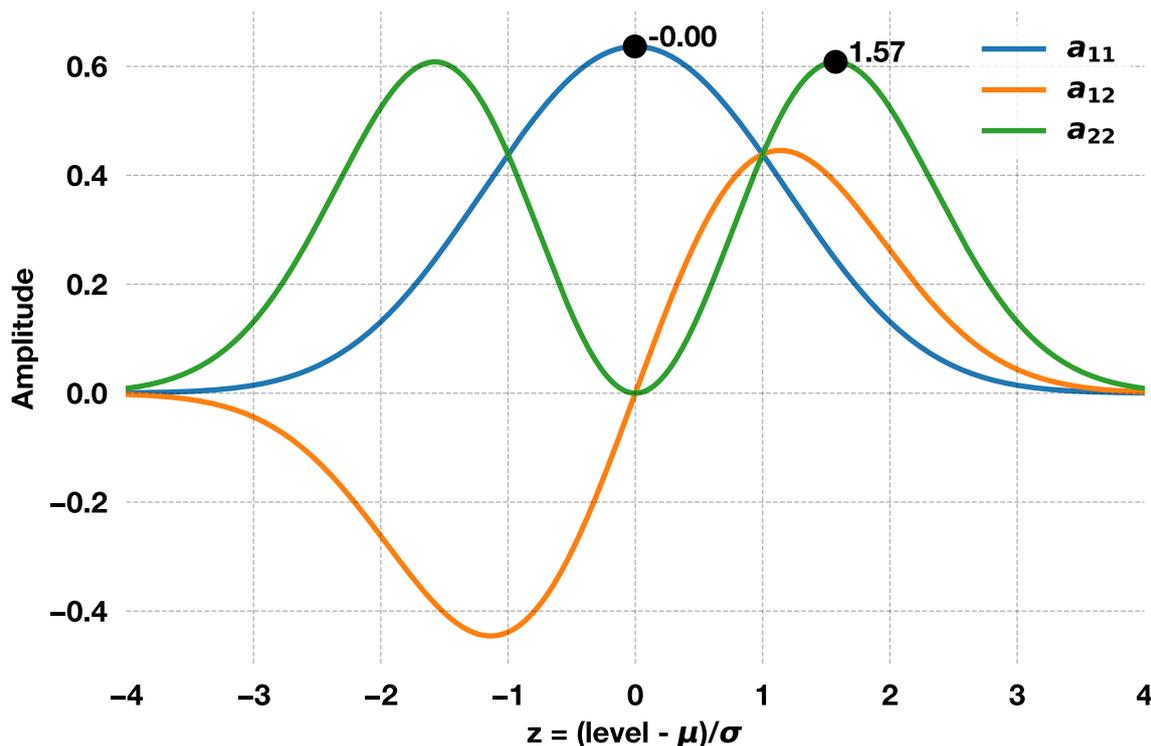


Figure 5.2. Values of the elements in the information matrix for a single specimen

Additionally, as mentioned, unique maximum likelihood estimators will only occur if there is overlap in the test outcomes. Neyer's algorithm is robust and overcomes these deficiencies and a brief summary of its is provided here.

To initialize the algorithm, the user needs only to provide a guess as to the lower and upper bounds of the expected  $\mu$  and a guess for  $\sigma$ . The algorithm begins with a binary search between the given bounds. The bounds are updated so that lower bound becomes the maximum stimulus level that produced a negative response and the upper bound is the minimum stimulus level that provides a positive response. This procedure is repeated until difference in the bounds (upper bound minus lower bound) becomes less than the guess provided for  $\sigma$ . It should be noted that if the algorithm only detects a single type of response, the bounds are automatically adapted to increase the search length until mixed responses are observed.

Next, the algorithm uses the midpoint between highest stimulus level returned a negative response and the lowest stimulus level that returned a positive response as an estimate for  $\mu$ . This  $\mu$  along with the original guessed  $\sigma$ ,  $\sigma_{guess}$ , and the data taken thus far (set of stimulus levels and their respective responses) are used to maximize the determinate of the information matrix. The stimulus level that maximizes the determinate of the information matrix is the next level tested. This procedure is repeated until overlap is achieved, and on each repetition  $\sigma_{guess}$  is updated by  $\sigma_{guess} = 0.8 * \sigma_{guess}$ . It should be noted that since  $\sigma_{guess}$  is decreasing, it is possible that the difference ([lowest stimulus that resulted in a positive response] - [highest stimulus that resulted in a negative response]) becomes greater than  $\sigma_{guess}$ . In this case, the algorithm resumes a binary search until this difference becomes less than the updated  $\sigma_{guess}$  or until overlap is achieved.

Once overlap is achieved, the guessed values for  $\mu$  and  $\sigma$  are no longer used, but the maximum likelihood estimates are calculated. The algorithm performs a brief check to ensure that these estimates are not ‘wild’ and then uses these to determine the information matrix. Again, the stimulus level which maximizes the determinate of the information matrix is suggested as the next tested level. From this point, the algorithm continues on indefinitely until the experimenter as tested as sufficient number of specimens or until the set of specimens have been depleted.

As a visual aid, Figure 5.3 gives a slightly adapted flowchart of Neyer’s sequential design of experiments algorithm. In the original version, the maximum likelihood estimates of  $\mu$  and  $\sigma$  are computed after the binary search has terminated regardless of whether overlap has been achieved. Without overlap, non-unique estimates are returned and  $\hat{\sigma}$  should be zero.  $\hat{\sigma} = 0$  is used as a decision flag: if true, then the current guesses of  $\mu$  and  $\sigma$  are provided to the information matrix; and if false, the unique maximum likelihood estimates are passed to the information matrix (after ‘wild’ check, of course). However, I did not wish to rely on the MLE of  $\sigma$  to be zero as numerical precision or the choice of optimizer may preclude  $\hat{\sigma}$  from being precisely zero. In my adapted algorithm, which is represented by the flowchart, the MLEs of  $\mu$  and  $\sigma$  are only calculated if overlap is achieved. The occurrence of overlap is determined by simply looking at

the difference between the highest negative level and lowest positive level; a quantity which is already calculated and stored. If this difference is negative, then overlap in the data is present. Without presenting proof, I believe this alteration will provide a more stable decision block between various coding languages, machine types, and choice of optimization routine.

## 5.2 Modernized (Generalized) Notation

In the preceding works, the distribution of the hidden variable is quickly assumed to be normal or logistic. Once this assumption is made, the subsequent derivations mostly focus on the maximum likelihood estimators, their properties, and numerical procedures for their calculation. However, a more general picture of the problem – estimating the latent distribution in a sensitivity test – emerges when no assumption about the distribution is made.

Using ‘plate’ notation, the current problem framework is shown graphically in Figure 5.4. In this figure  $i$  is the specimen number and there are  $n$  total specimens to be tested.  $x_i$  is the stimulus level at which specimen  $i$  is tested and  $y_i$  is the specimen response (binary response).  $h_i$  is the actual threshold of specimen  $i$ , and all of the specimens are assumed to have threshold values that originate from a distribution uniquely determined by two parameters,  $\mu$  and  $\sigma$ .

Under this model the probability  $y_i$  is easily described as

$$p(y_i = 1|x_i, h_i) = \begin{cases} 1, & \text{if } x_i \geq h_i \\ 0, & \text{otherwise} \end{cases} \quad (5.39)$$

or

$$p(y_i = 0|x_i, h_i) = \begin{cases} 1, & \text{if } x_i < h_i \\ 0, & \text{otherwise} . \end{cases} \quad (5.40)$$

Now, maintaining generality, it’s assumed that the distribution  $h_i$  is given by a set of parameters  $\vec{\theta}$ , which is written as  $p(h_i|\vec{\theta})$ . Following the rules of probability, the joint distribution of  $y_i$  and  $h_i$  is

$$p(y_i, h_i|x_i, \vec{\theta}) = p(y_i|x_i, h_i)p(h_i|\vec{\theta}). \quad (5.41)$$

$h_i$  can be marginalized out to produce a probability measure for  $y_i$  that is given given  $x_i$  and  $\vec{\theta}$  as

$$p(y_i|x_i, \vec{\theta}) = \int_{-\infty}^{\infty} p(y_i|x_i, h_i)p(h_i|\vec{\theta})dh_i. \quad (5.42)$$

Now, a ‘nested’ description of  $p(y_i|x_i, h_i)$  could be produced, but it is easier to work with the case of  $p(y_i = 1|x_i, h_i)$  which was intuitively formed and shown in equation 5.39. Making this substitution into equation 5.42 yields

$$p(y_i = 1|x_i, \vec{\theta}) = \int_{-\infty}^{\infty} p(y_i = 1|x_i, h_i)p(h_i|\vec{\theta})dh_i. \quad (5.43)$$

According to equation 5.39,  $p(y_i = 1)$  is equal to one when  $x_i \geq h_i$  and is zero otherwise. Thus, in the integral of equation 5.43 the integrand is zero for all values  $h_i > x_i$  so that  $x_i$  can replace  $\infty$  as the integral upper limit.

$$p(y_i = 1|x_i, \vec{\theta}) = \int_{-\infty}^{x_i} (1)p(h_i|\vec{\theta})dh_i. \quad (5.44)$$

Once  $p(y_i = 1)$  is known,  $p(y_i = 0)$  is simply given by  $1 - p(y_i = 1)$  and the likelihood function can be formed. As mentioned, the previously listed works assume that  $h_i$  is described by either a normal or logistic distribution parameterized by  $\mu$  and  $\sigma$ . If the the normal distribution assumption is made then  $p(h_i) \sim N(\mu, \sigma)$  and equation 5.44 becomes

$$p(y_i = 1|x_i, \vec{\theta}) = \int_{-\infty}^{x_i} N(\mu, \sigma)dh_i = \Phi(x_i|\mu, \sigma) \quad (5.45)$$

where  $\Phi(x_i|\mu, \sigma)$  is the cumulative distribution function (cdf) of a normal distribution with given parameters  $\mu$  and  $\sigma$ . Likewise, if a logistic distribution assumption is made then equation 5.44 reduces to

$$p(y_i = 1|x_i, \vec{\theta}) = \int_{-\infty}^{x_i} Logistic(\mu, \sigma)dh_i = S(x_i|\mu, \sigma) \quad (5.46)$$

where  $S(x_i|\mu, \sigma)$  is the cdf of the logistic distribution;

$$S(x_i|\mu, \sigma) = \frac{1}{1 + \exp\left(\frac{-(x_i - \mu)}{\sigma}\right)}. \quad (5.47)$$

Using this generalized model, and starting from either equation 5.42 or 5.44 opens up the application of additional statistical methods such as expectation maximization, Bayesian analysis, and Gaussian processes; as demonstrated in subsequent sections.

### 5.3 Proposed Methods

In an effort to update – or at least provide more testing options to – the currently used sensitivity tests, additional tests, alongside their derivation and performance, are presented here.

#### 5.3.1 Conjugate Neyer

Neyer’s sequential design of experiments algorithm, as described in section 5.1.3, elects the next stimulus level for testing by maximizing the determinate of the information matrix. The process of maximizing the determinate maximizes information about both parameters  $\mu$  and  $\sigma$  simultaneously. However, the stimulus level that simultaneously maximizes both parameters is most likely not the same stimulus level that maximizes the information (or minimizes the CRLB) of either parameter  $\mu$  or  $\sigma$  when considered individually. This test is a simple adaption of Neyer’s sequential design of experiments wherein the next stimulus level is chosen as best suited with respect to a single parameter for a given test. For each new test, the parameter  $\mu$  or  $\sigma$  is alternated so that no single estimator completely dominates the experiment design.

#### 5.3.2 Maximum Likelihood Estimation Applied to Mixture of Gaussians

One major drawback to most of the aforementioned sensitivity tests is that the underlying threshold is assumed to be either normally or logistically distributed. While this assumption is valid for many real systems and materials, when it is not true, discovery of the actual latent variable distribution is not possible. Thus, a more generalized approach is desirable – which can return a normal distribution when applicable but also return an arbitrary distribution that is more representative of the probability structure.

One potential case of interest occurs when the threshold distribution is bimodal, like that produced by a mixture of two Gaussian distributions. This scenario arises in a variety of situations, including: when two physical mechanisms are competing to govern the material sensitivity (e.g. two initiation mechanisms), or when specimen preparation

yields two batches of samples where each batch has a unique mean threshold value. For the latter case discovery of the bimodal distribution would serve as a useful anomaly detection or quality control tool.

It should be noted that when a the latent threshold values are distributed bimodally the application of a normal (or logistic) assumption will result in an estimate for  $\mu$  that is between the two modes and an estimate for  $\sigma$  that is too large. In other words, this assumption will fit a broad normal (or logistic) distribution that encompasses the entirety of the bimodal distribution. Also, when using Neyer’s sequential design of experiments algorithm,  $\hat{\sigma}$  will require more runs to stabilize. To illustrate these points, Figure 5.5 is an example bimodal threshold distribution alongside its accompanying cumulative density function (cdf).

Figures 5.6 to 5.8 show the evolution of the estimates  $\hat{\mu}$  and  $\hat{\sigma}$  when the Neyer sequential design of experiments algorithm is applied to the latent distribution of Figure 5.5. The response, ‘go’ or ‘no-go’, of each tested stimulus level was determined via a simple Monte Carlo simulation. In these figures the black lines indicate the actual pdf and cdf, the red lines indicate the current estimated distributions, the vertical ‘carrot’ symbols indicate the location of the next requested stimulus level, the x-axis is the stimulus level (a.u.), and the tested points and their results are shown in blue and green points.

As expected, figures 5.6 to 5.8 demonstrate that a normal assumption results in a normal distribution that is broad and cannot capture the fine details of the actual distribution and its cdf. This affect is exacerbated when greater separation between the Gaussians is present as seen in figures 5.9 to 5.12.

As a first step toward generalizing sensitivity testing to a richer set of distributions, the maximum likelihood estimators and the information matrix criteria for stimulus level selection – or the method described by Neyer and predecessors – is expanded to the bimodal (mixture of two Gaussians) case. It will be demonstrated shortly that maximum likelihood estimation is a poor method of choice for a mixture of Gaussians.

The derivation of the MLEs and information matrix for a mixture of two Gaussians is lengthy. As such, it is not reproduced here but can be found in appendix A. Assuming the underlying distribution given in Figure 5.5, and testing at 30 equidistant stimulus

levels from 45 to 59, inclusive, the result of the maximum likelihood estimators are shown in Figure 5.13.

Intuition into this result is gained by considering the estimators assuming a normal hidden distribution when no overlap is present in the data. This is the ‘non-unique’ case as described earlier which returns estimates  $\hat{\sigma} = 0$  and  $\hat{\mu} =$  (midpoint between highest negative response and lowest positive response). If plotted, these estimates would appear as a single spike in the pdf centered at  $\hat{\mu}$  and a step from 0 to 1 in the cdf, again centered at  $\hat{\mu}$ . Figure 5.13 has a similar profile but with two ‘steps’ centered over  $\hat{\mu}_1$  and  $\hat{\mu}_2$  respectively. So, the bimodal MLEs are simply ‘fits’ to the two regions of non-overlapping data; which will always be present. Upon recognizing the failure mode of the currently described MLE framework it is not hard to find resources which explain other methods to approximate the MLEs of a mixture of Gaussians. For example, Rossi [56] states that the E-M algorithm is the only reliable way to compute maximum likelihood estimates for the mixture of normals model.

## Sequential Design

Despite the drawbacks of estimating the parameters of a mixture of normals with the maximum likelihood method, the information matrix may prove to be useful in determining which stimulus levels to test. In this model, the information matrix is composed of 25 terms; 15 of which are unique. The derivation of these terms is lengthy and is provided in appendix A. For now, it suffices to plot the information matrix terms like in Figure 5.1.2 for the unimodal case. Plotting all 15 terms at once is crowded so only the 5 terms on information matrix diagonal are shown in Figure 5.14. In this figure, the terms  $a_{11}$ ,  $a_{22}$ ,  $a_{33}$ ,  $a_{44}$ ,  $a_{55}$  are *related to* the variances of the estimators  $\hat{\mu}_1$ ,  $\hat{\sigma}_1$ ,  $\hat{\mu}_2$ ,  $\hat{\sigma}_2$ , and  $\hat{\pi}_1$ , respectively. Figure 5.15 plots only terms  $a_{11}$  and  $a_{22}$  and is provided for clarity. As seen in Figure 5.14, simultaneous maximization of information about the estimators is impossible, like unto the unimodal case, and moreover, a suitable compromise point does not exist.

If no single stimulus level can serve as an adequate compromise to maximize the information about all the estimators then a ‘conjugate’, or alternating, design should be used. This design would elect the stimulus level which maximizes information about  $\hat{\mu}_1$  ( $a_{11}$ ), then  $\hat{\sigma}_1$  ( $a_{22}$ ), and so on. It should be noted that maximizing the information about a term, like  $a_{11}$ , is related to, but not equivalent to, minimizing the variance of  $\hat{\mu}_1$ . Consequently, the design can be further altered to alternate between stimulus levels which minimize  $V[\hat{\mu}_1]$ ,  $V[\hat{\sigma}_1]$ , and so forth.

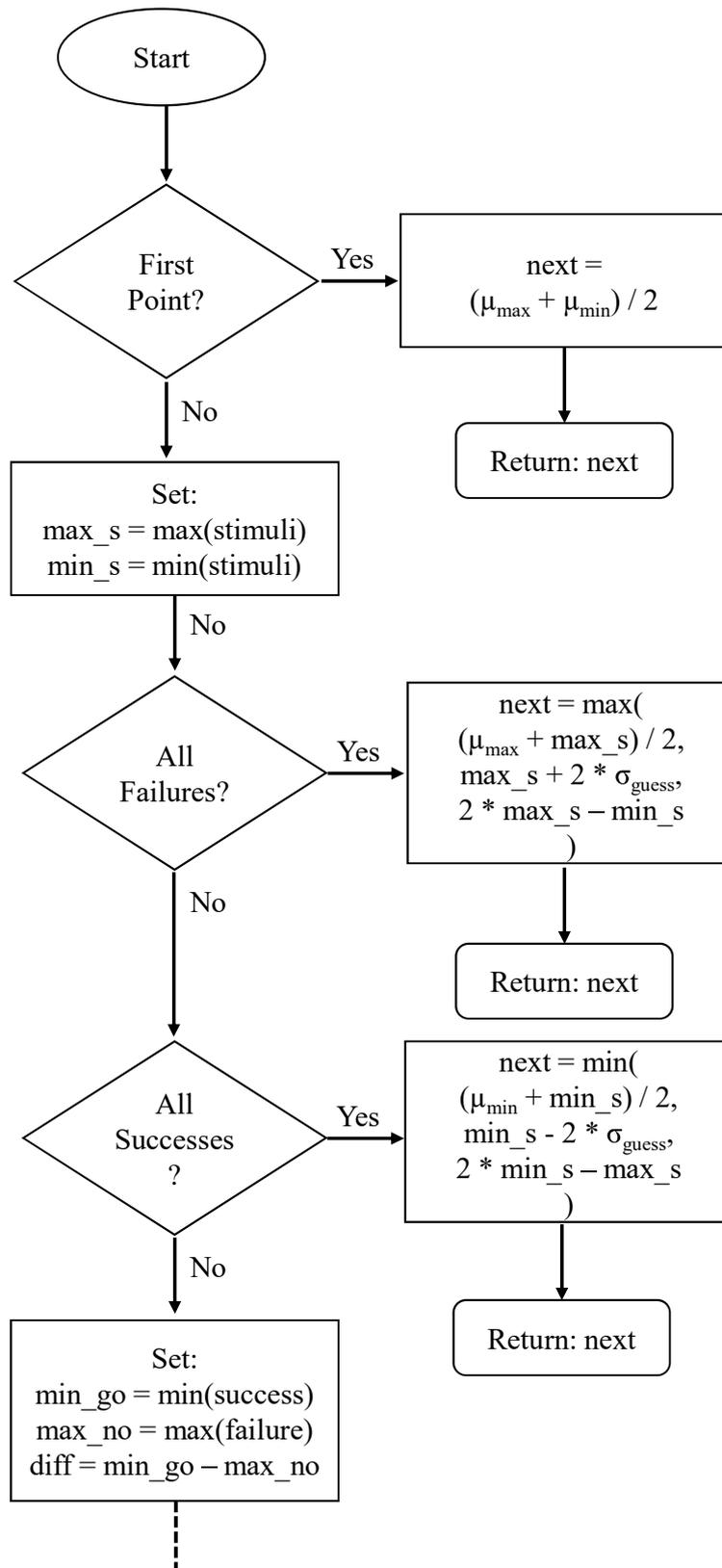
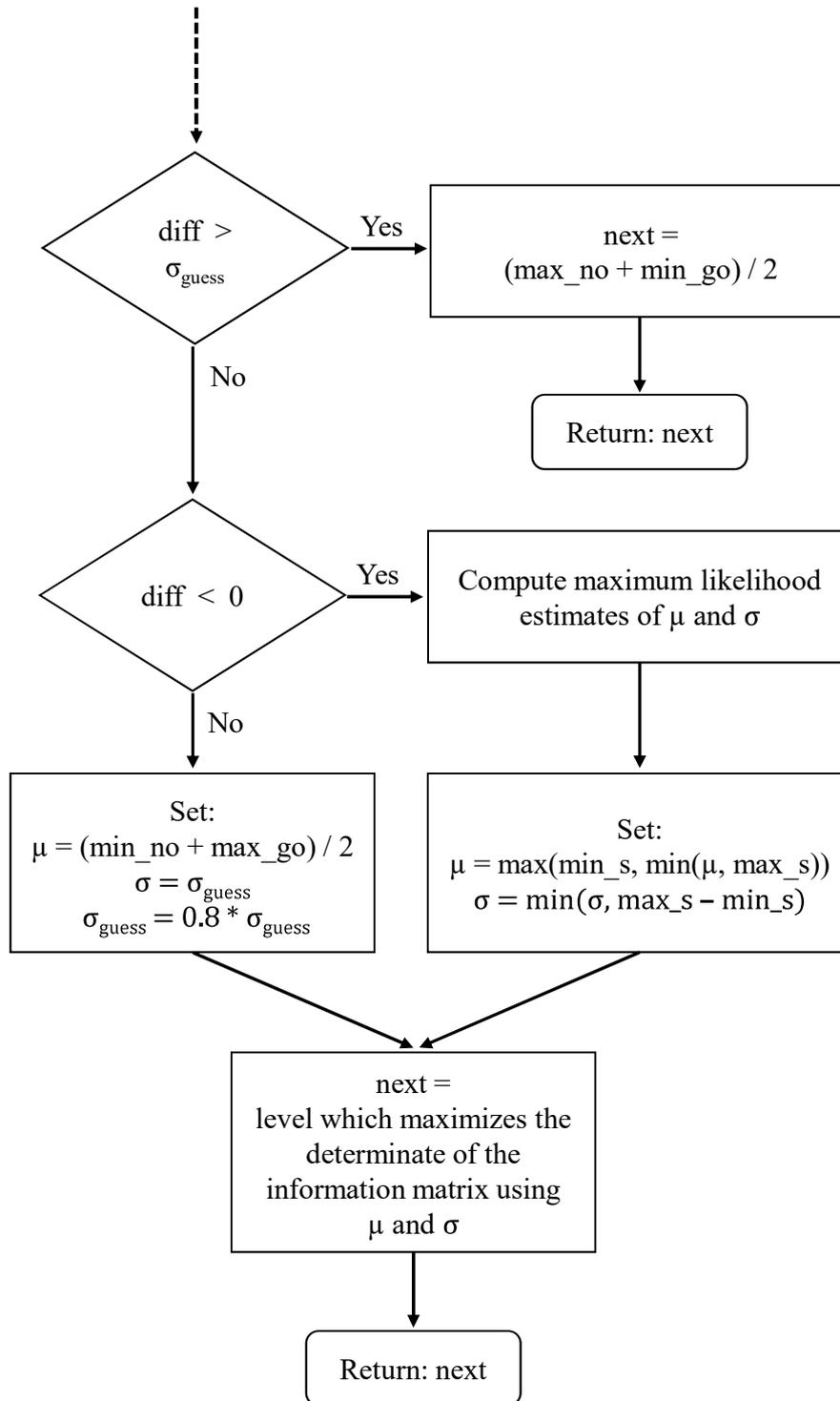


Figure 5.3. Flowchart of Neyer's sequential design algorithm

Figure 5.3. continued



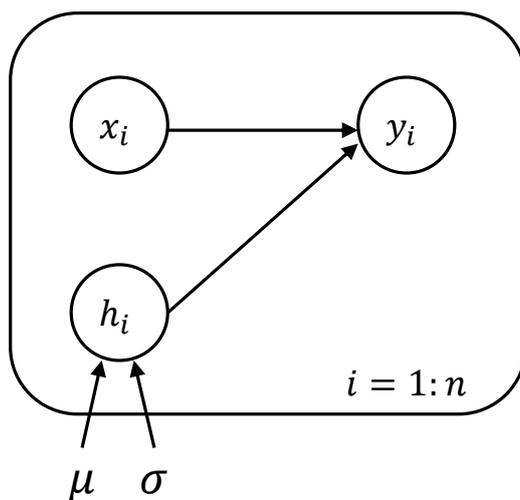


Figure 5.4. Graph demonstrating interaction of sensitivity model random variables assuming that the threshold distribution is parameterized by two variables

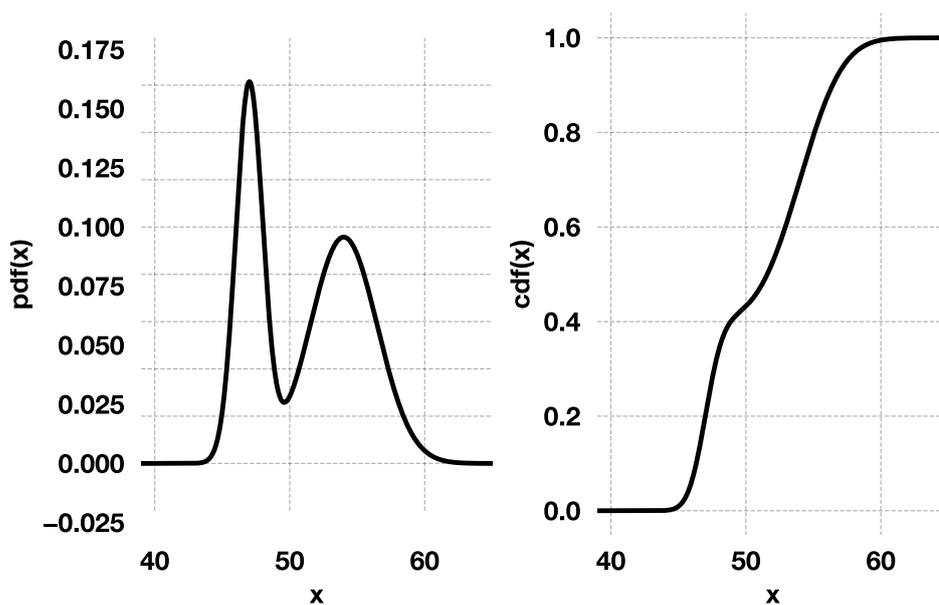


Figure 5.5. Left: Example of a bimodal distribution composed of two Gaussians. The left Gaussian is given by  $\sim N(47, 1)$  and the right by  $\sim N(54, 2.5)$ . The prefactors of the left and right Gaussian distributions are .4 and .6, respectively. Right: the cumulative density function of bimodal distribution shown in the left pane.

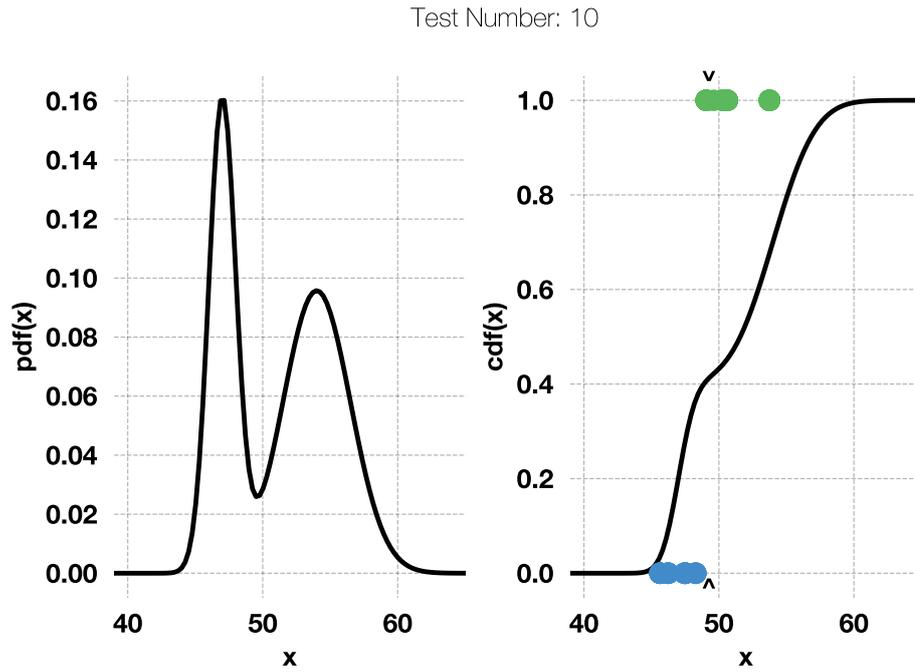


Figure 5.6. Neyer sequential design of experiments algorithm applied to a bimodal latent distribution.

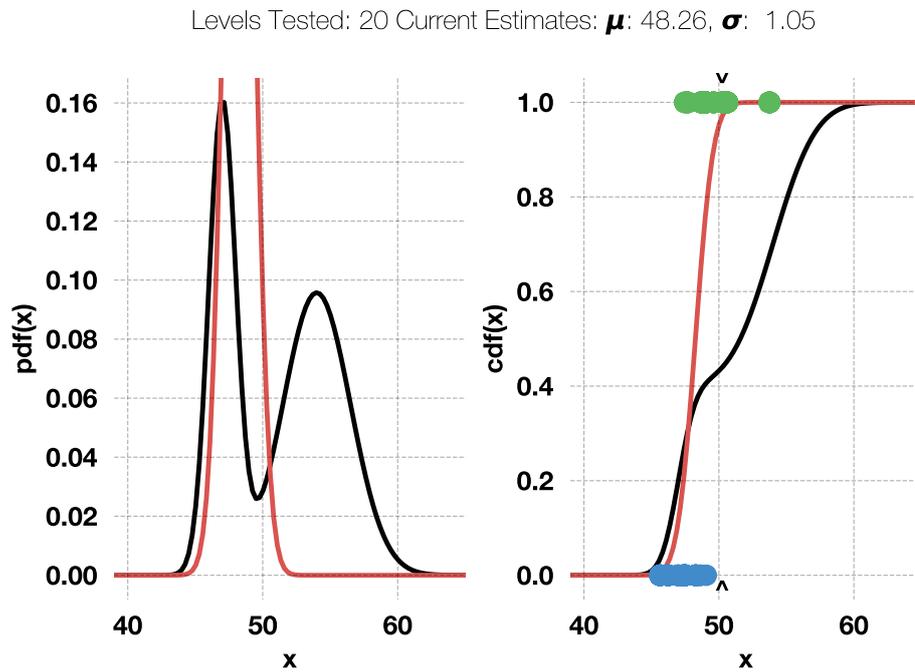


Figure 5.7. Continued: Neyer sequential design of experiments algorithm applied to a bimodal latent distribution.

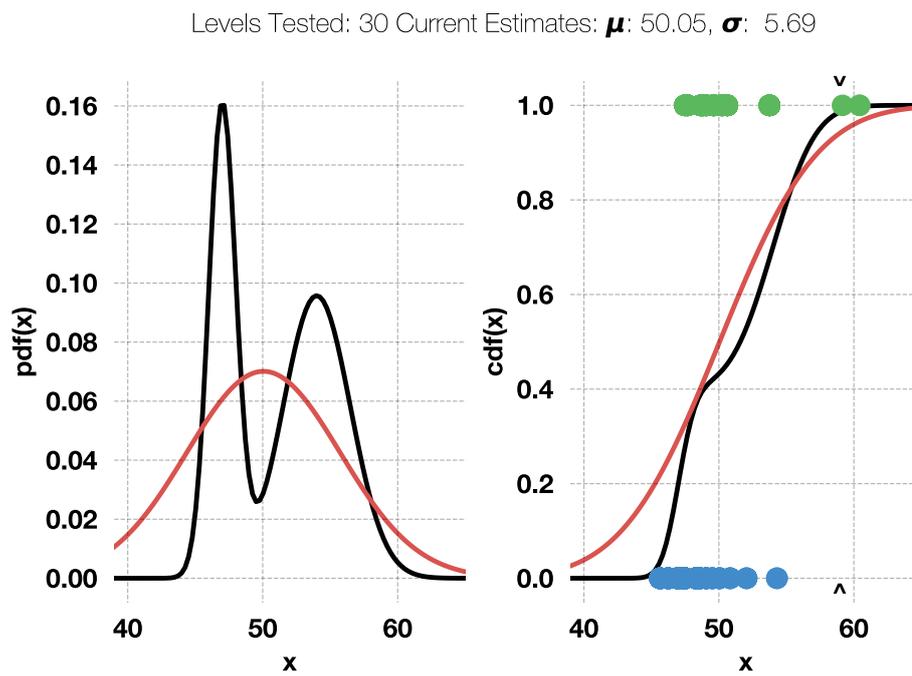


Figure 5.8. Continued: Neyer sequential design of experiments algorithm applied to a bimodal latent distribution.

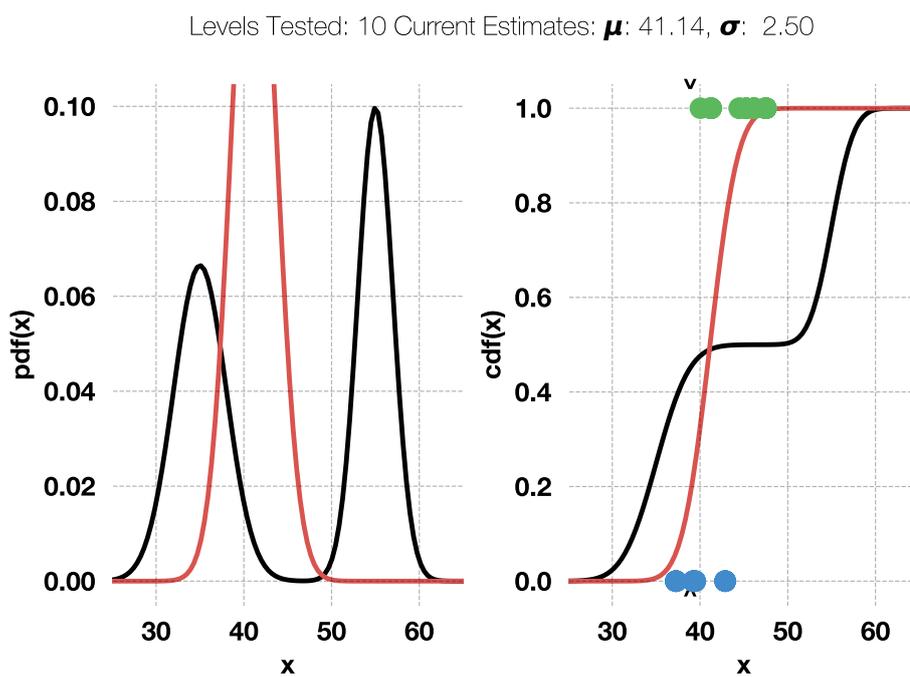


Figure 5.9. Neyer sequential design of experiments algorithm applied to a disperse bimodal latent distribution. Bimodal distribution is given by  $\sim N(35, 3)$ ,  $N(55, 2)$ , and prefactors of .5 and .5.

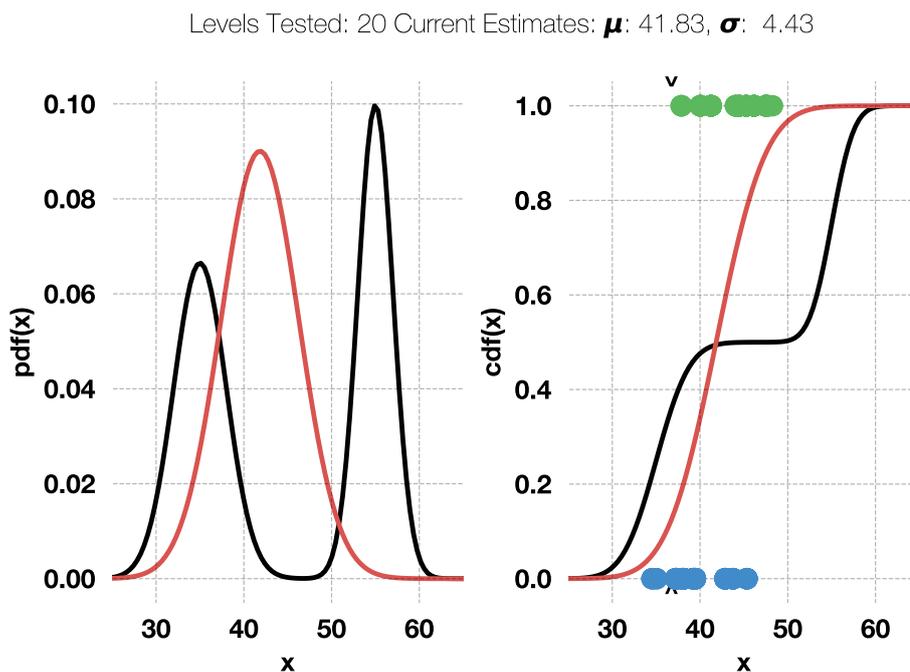


Figure 5.10. Continued: Neyer sequential design of experiments algorithm applied to a disperse bimodal latent distribution.

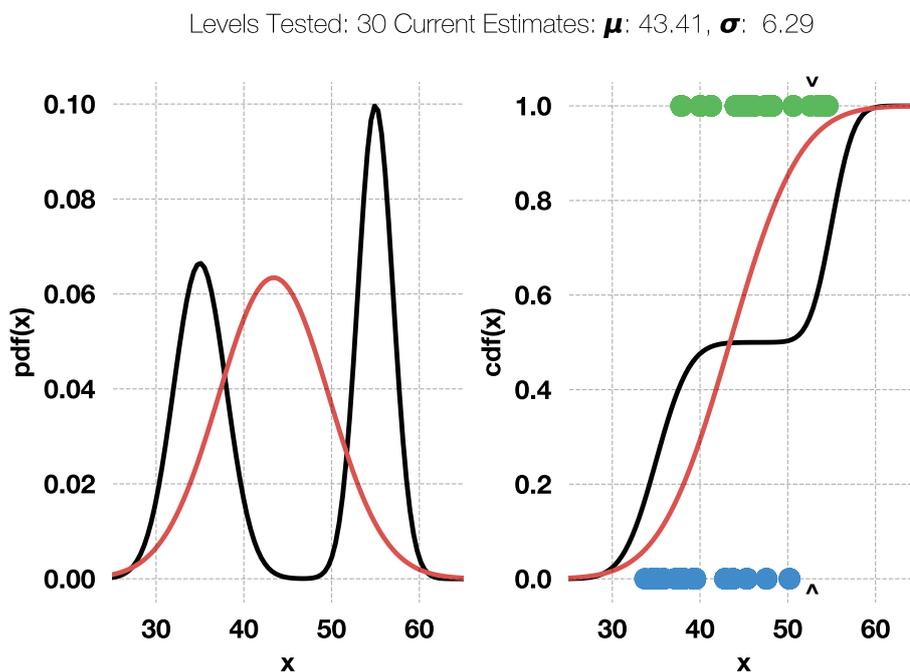


Figure 5.11. Continued: Neyer sequential design of experiments algorithm applied to a disperse bimodal latent distribution.

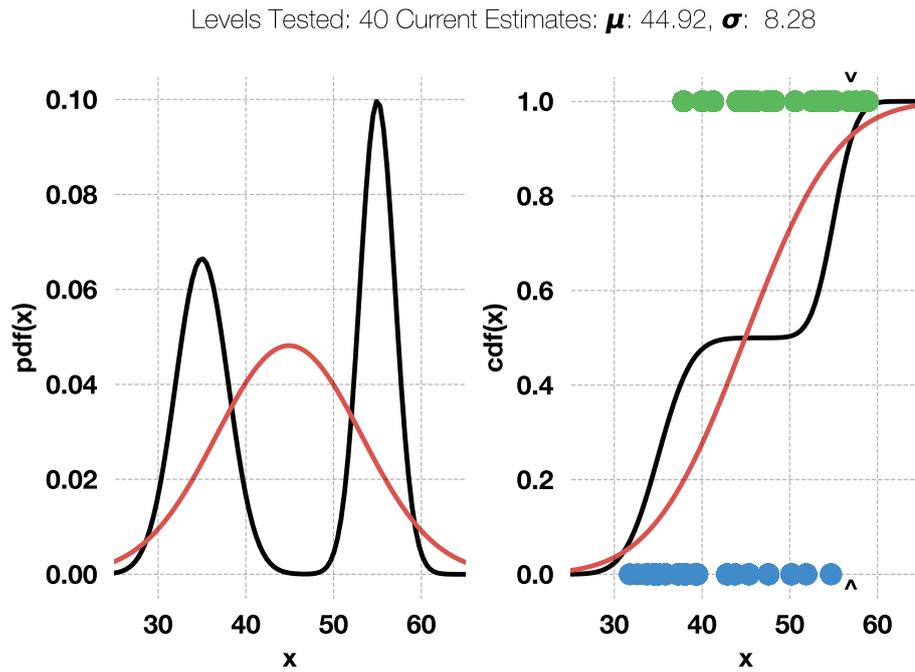


Figure 5.12. Continued: Neyer sequential design of experiments algorithm applied to a disperse bimodal latent distribution.

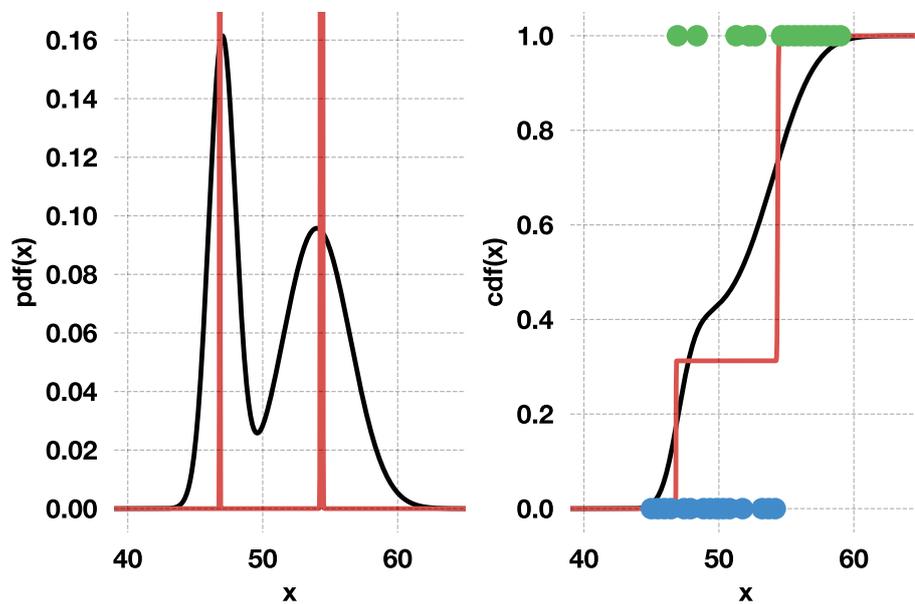


Figure 5.13. Estimated pdf and cdf using bimodal maximum likelihood estimators.

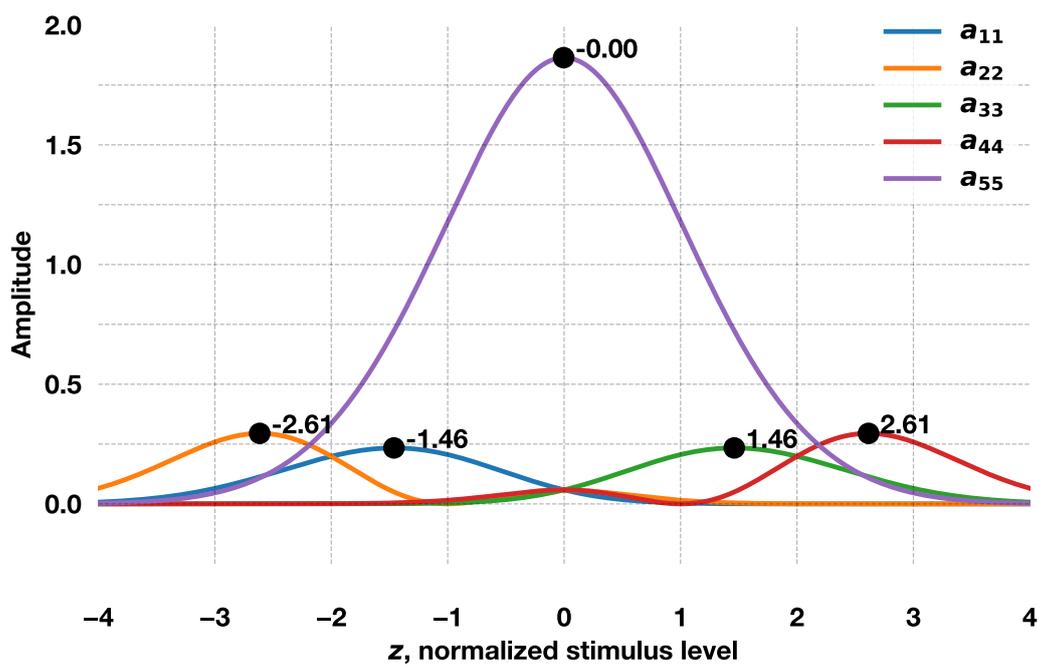


Figure 5.14. Values of select elements for the information matrix of a mixture of two Gaussians. The values are plotted for a single specimen assuming a mixture model defined by two Gaussians as  $0.5N(-1, 1) + 0.5N(1, 1)$ .

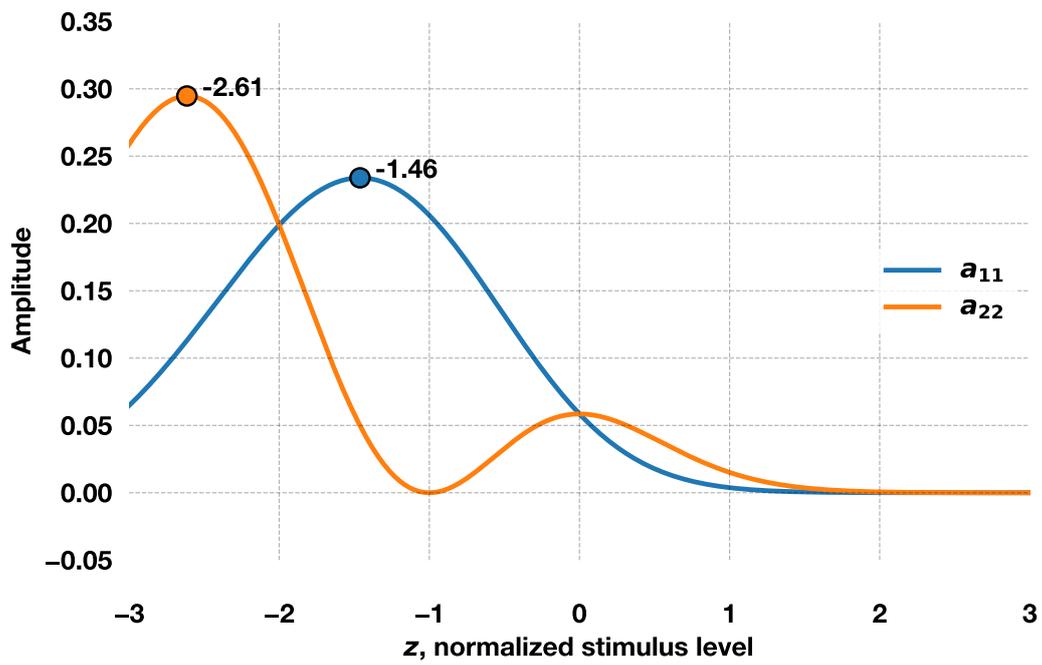


Figure 5.15. Terms  $a_{11}$  and  $a_{22}$  of Figure 5.14 are shown here with appropriately scaled axes for clarity.

## 6. MACHINE LEARNING OF THE SIMULATED COLLAPSE OF PORE GEOMETRY IN HMX UNDER SHOCK LOADING

### 6.1 Collaborators

The work performed in this chapter was done in collaboration with the Air Force Research Laboratory (AFRL) at Eglin Air Force Base. Specifically, the coauthors are Barrett Hardin (AFRL), Jesus Mares (AFRL), Ilias Billionis (Purdue), and Steven Son (Purdue).

### 6.2 Introduction

It is widely accepted that the initiation of energetic materials occurs through energy localization near material heterogeneities. The energy localization causes regions of high temperature referred to as hot spots [57]. While many mechanisms of hot spot formation have been proposed – such as friction [58], shear, adiabatic heating [59, 60], areas of dislocation pile-up [61] – it is believed that under shock loading hot spots formed in the vicinity of collapsing pore spaces are crucial to the initiation process [62].

Over the last decade, investigation of pore collapse has been performed on a wide variety of energetic materials [63–66], material models [67], input pressures [68], 3D effects [69, 70], initiation modes [71, 72], and pore morphologies [73–75]. In studies concerning the effect of pore morphology, a number of simple pore geometries have been examined; however, no rigorous study has yet been performed on realistic pore shapes occurring in real energetic material microstructures. Furthermore, no efforts have been made to compare the relative differences in the shock loading response of realistic pore shapes and the idealized geometries considered within these studies.

In this work, we first simulate the shock loading response of simple elliptical pore structures within HMX (cyclo-tetramethylene-tetranitramine,  $C_4H_8N_8O_8$ ). This is done

to establish an in-house baseline of idealized pore geometries to which we can compare the results of realistic pores. We repeat the simulation process using realistic pore shapes retrieved from the microstructure of a pressed HMX sample obtained via scanning electron microscopy (SEM). The simulation results are reduced to a single scalar value by noting the minimum incoming flyer velocity required to obtain a “critical” outcome.

All pore shapes are normalized to the same area – to either 150 nm or 300 nm equivalent circular diameters – to remove the effect of size. We then develop numerical representations of the pore geometries. Using these pore representations and the associated critical velocities, random forest (RF), extreme gradient boosted machine (XGB), and convolutional neural network (CNN) machine-learned (ML) models are trained and act as surrogates to the simulation process. The performance of these models is strongly related to the input representation indicating the amount of useful information each representation contains – with respect to the learning task at hand. Finally, we provide an interactive interface to these models to make the exploration of the effect of pore shape to shock sensitivity more accessible.

### 6.3 Extracting Void Shapes

In order to examine realistic pore geometries, high resolution images of a single pressed pellet of Class 5 HMX were obtained via scanning electron microscopy (SEM). The pellet was pressed to a nominal density of 94% theoretical maximum density (TMD) and sectioned with a Hitachi IM 4000Plus argon ion-milling system. The pellet surface was further polished and the images collected with as ZEISS Crossbeam 340 focused ion beam scanning electron microscope (FIB-SEM). The resulting image resolution was 9.77 nm per pixel.

Images were analyzed in a Python environment. First, the void space and material were distinguished within the image using a modified Otsu’s threshold technique [76] based on the grayscale intensities to produce a binary image. An example SEM image and its binary representation are provided in Figure 6.1. From the binarized image,

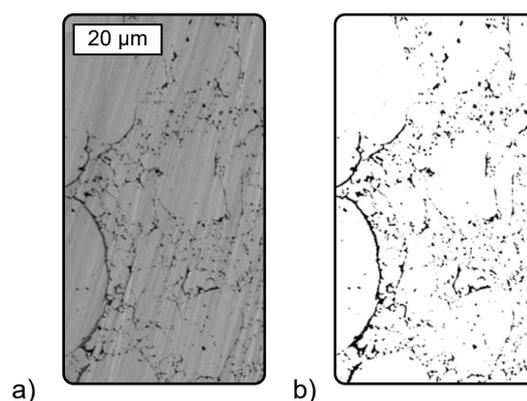


Figure 6.1. a) An image of a sectioned and polished surface of a single pressed class 5 HMX sample acquired via FIB-SEM. b) The binarized representation of image shown in a). White indicates HMX and black is void space.

individual void regions were identified using the scikit-image [77] package and the void contours were extracted using the OpenCv [78] library.

The geometric properties of aspect ratio, solidity, and circularity were calculated for each void. The aspect ratio is determined by identifying the minimum area rectangle (or bounding box) which fully encompasses the void space and dividing the length of the larger side of said rectangle by the length of the smaller side. Circularity is defined as the ratio of the perimeter of the void space to the perimeter of a circle with equivalent area. Solidity is defined as the ratio of the area of the void to the area of the bounding box. Using these properties the voids were separated into two categories; namely, "pore shapes" and "crack networks". A void was classified as a pore shape if its aspect ratio was less than 7.0, its solidity was greater than 0.3, and its circularity was greater than or equal to 0.65; otherwise, the void was categorized as a crack. For this work, we only analyzed voids deemed pore shapes and did not consider crack networks. Using the aforementioned metrics, 55 FIB-SEM acquired images were parsed resulting in 179,207 pore shapes.

### 6.3.1 Simple Shape Geometries

Although the primary goal of this work is to understand how realistic pore geometries influence shock sensitivity, it is useful to examine how simple geometries affect the simulation outcome. To this end, we created a set of pores structured after ellipses with varying aspect ratios and orientations. For ellipses, the aspect ratio is defined as the ratio of the major axis to the minor axis. All ellipses are centered to the underlying coordinate system and the ellipse orientation is defined as the angle the major axis with respect to the positive x-axis measured counter-clockwise. Ellipses with aspect ratios of 1 to 4 in .15 increments and orientations of 0 to 90 degrees in 5 degree increments were produced. In total, 381 ellipses are contained within the set because an ellipse with an aspect ratio of 1 is a circle and does not change with orientation.

### 6.3.2 Removing the Effect of Size

It is well known that the pore size is heavily related to shock sensitivity. To remove the effect of size – and focus on pore shape – the pore contours and ellipse structures are scaled to same internal area.

Here, we choose to reference a specified area by noting the diameter of a circle with equal area. For example, rather than state that the pore shapes were “normalized to an area of 70,685.83 nm<sup>2</sup>” ( $\pi \left(\frac{300}{2}\right)^2$ ), we simply say that the pore shapes were “normalized to an equivalent circular diameter of 300 nm” (or just “equivalent diameter”). Herein, the extracted realistic pore shapes and the simple pore ellipses are scaled to equivalent diameters of 150 nm and 300 nm constituting two different data sets aptly named the “150 nm data set” and the “300 nm data set”, respectively.

To maintain consistency, each pore has been shifted such that its centroid is coincident with the underlying coordinate system. Furthermore, like unto the ellipse, we reference the pore orientation as the angle of the longer side of the bounding box with respect to the positive x-axis measured counter-clockwise.

## 6.4 CTH Simulation

Armed with the contours of thousands of realistic pore geometries and simple ellipses, we shift to determining if a pore shape will contribute and sustain an incoming shock wave upon collapse. We simulate the pore collapse using the hydrocode CTH (version 12.0) developed by Sandia National Laboratory.

Each simulation consists of a rectangular slab of HMX containing a single, empty pore structure. The slab domain has a width and height of 10  $\mu\text{m}$  and 5  $\mu\text{m}$ , respectively, with a mesh resolution of 5 nm. After proper scaling, the pore structure is inserted into the HMX material. The centroid of the pore is placed 2.5  $\mu\text{m}$  into the HMX slab measured from the bottom and left edges.

The HMX material is modeled using distinct equations of state (EOS) for the unreacted material and the reaction products. The unreacted material uses a tabular EOS corresponding to published experimental data and also accounts for the temperature dependence of the heat capacity and the pressure dependence of the melt temperature. The reaction product EOS is defined with the Jones-Wilkins-Lee (JWL) formula which has the form

$$P = A \left( 1 - \frac{\omega V_0}{V R_1} \right) e^{\frac{-R_1 V}{V_0}} + B \left( 1 - \frac{\omega V_0}{V R_2} \right) e^{\frac{-R_2 V}{V_0}} + \frac{\omega e}{V}, \quad (6.1)$$

where  $P$ ,  $e$ , and  $V$  correspond to pressure, specific energy, and volume, respectively,  $V_0$  is the initial volume,  $\omega$  is the Grüneisen coefficient, and  $A$ ,  $B$ ,  $R_1$ ,  $R_2$  are parameters. The parameter values are summarized in Table 6.1.

The reaction progress follows an Arrhenius reactive burn model

$$\frac{d\lambda}{dt} = (1 - \lambda) F e^{\frac{-\Theta}{T}}, \quad \Theta = \Theta_0 (1 + A_P P) \quad (6.2)$$

where  $\lambda$  is the reaction progress,  $t$  is time,  $T$  is temperature, and  $P$  is pressure. The remaining parameters are material specific:  $\Theta_0$  is the activation temperature,  $F$  is the frequency factor,  $A_P$  is the pressure derivative of  $\ln(\Theta)$ . The actual values used are provided in Table 6.1.

Table 6.1. Equation of state parameters for the detonation products and the reaction rate parameters.

JWL EOS		Arrhenius Reactive Burn	
Parameter	Value	Parameter	Value
$R_1$	4.6240	$\Theta_0$ (K)	17871
$R_2$	1.4705	$F$ ( $s^{-1}$ )	5.6e12
A	9.639914e12	$A_P$	0.0
B	2.14317e11		
$\omega$	0.4319		

In the simulation the HMX material is considered perfectly plastic with a yield strength of 0.37 GPa and Poisson’s ratio of 0.20.

The shock loading is applied by specifying the velocity of an incoming flyer plate composed of inert HMX. The flyer plate has a width and height of 5  $\mu\text{m}$  and impacts the HMX slab from the left. The simulation is stopped after a fixed time period of 2 ns after the flyer plate contacts the HMX slab.

In coupled thermal and mechanical hydrocode simulations it has been observed that the thermal conductivity calculation can lead to vanishingly small Courant time steps near the pore structure prior to complete collapse. To implement a thermal diffusion model while avoiding numerical instabilities, conduction is not considered until 0.91 ns into the simulation to ensure the pore has fully collapsed. A constant thermal conductivity of  $3.62975 \frac{\text{W}}{\text{m}\cdot\text{K}}$  is applied to HMX up to a temperature of 1160 K, after which it is assumed to go to zero.

During the simulation the spatial field variables of temperature, reacted mass, and pressure are tracked. The evolution of the temperature field surrounding a pore is displayed in Figure 6.2. In particular, the reacted mass field is used to determine whether the pore reached “criticality”, as explained in the following section.

Unfortunately, due to clerical errors, the yield strength, Poisson ratio, and thermal conductivity employed in the CTH simulation are not in agreement with reported values [79–81]. The aggregate effect of these errors was investigated and the difference between critical threshold values attained when using the correct HMX properties was found to be less than 10 m/s than those reported here. This difference is trivial and does not affect any conclusions drawn from this work.

#### 6.4.1 Criticality Criterion

At the end of the simulation period the reacted mass field variable is integrated over the domain to calculate the total reacted mass. If the total reacted mass exceeds  $1.343\text{e-}11$  grams then we state that the hot spot will progress into a thermal run-away scenario (critical condition), otherwise the reaction is quenched due to the heat transfer away

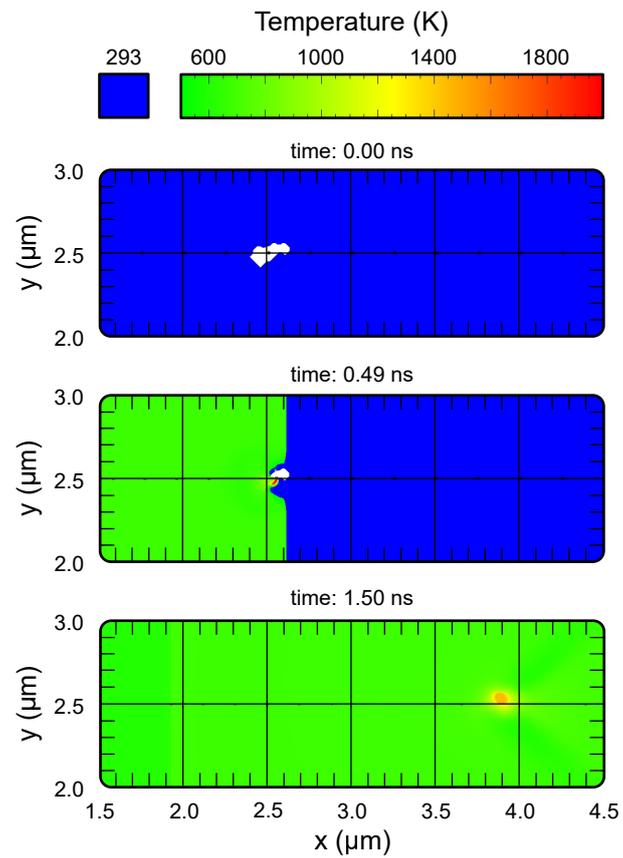


Figure 6.2. Example of the time evolution of the temperature field for a pore collapse simulation in CTH. The incoming flyer plate velocity was 2528 m/s. Equivalent circular diameter: 150 nm. Explicit pore id number: 106028.

from the heated material overcoming the heat generation due to the chemical reaction (sub-critical). The reacted mass threshold was chosen to correspond to the mass of a critical hot spot with an equivalent diameter of 30 nm using the methodology of Tarver et. al. [80] for the determination of a critical hot spot.

The initial flyer plate velocity is adjusted using an adaptive binary search algorithm until criticality is achieved within 1 m/s of a sub-critical result. Thus, the outcome of the entire simulation process is reduced to a single threshold velocity which corresponds to the minimum required velocity of the incoming flyer necessary for the pore to reach a critical state.

#### 6.4.2 Data Sets

On average, a single pore collapse simulation takes 6.35 minutes to run using 320 Intel Xeon E5-2698 v3 processors. The binary search algorithm typically required 10 simulations to determine the critical threshold velocity with 1 m/s uncertainty. Therefore, approximately 338.6 compute-hours are consumed to assign a critical threshold velocity for a single pore.

Although we extracted 179,207 pore shapes, it is computationally prohibitive to simulate all of them in CTH; nor is it likely necessary to simulate all pores to adequately fit machine-learned surrogate models. As mentioned, the pore collapse simulations were performed at two equivalent circular diameters creating two sets of data.

The first data set consists of 6,258 pore shapes scaled to an equivalent diameter of 150 nm. The pores were selected at random so long as they were resolved by at least 25 pixels. The second set is comprised of pore shapes scaled to 300 nm equivalent diameter. Any pore resolved by at least 100 pixels in the 150 nm data set (1,452 pores) is also contained in the 300 nm data set. The remaining pores in the 300 nm data set were randomly selected as long as they were resolved by 100 pixels. There are a total of 6,152 pores in the 300 nm data set. Again, we note that the set of ellipses were also scaled to equivalent diameters of 150 nm and 300 nm for direct comparison with the realistic pores data sets.

## 6.5 ML Models

Constructing an effective ML model first requires developing a suitable machine-readable representation of the input object [82,83]. These representations are commonly referred to in the literature as “descriptors”, “features”, “fingerprints”, or “profiles”. The input representation to the ML models produced here take one of four forms: 1) an array of just the aspect ratio and orientation of the pore as determined by the Fourier ellipse method; 2) an array of aspect ratio and orientation augmented with the remaining simple metrics of perimeter, mean diameter, circularity, and solidity; 3) the Fourier coefficients; 4) the 2D standardized array. Descriptions of these representations are given in the following subsections.

### 6.5.1 Pore Representations

#### Aspect Ratio and Orientation

After normalization, an ellipse is fully described by specifying its aspect ratio and orientation. Likewise, the bounding box of a realistic pore shape can be used to assign an aspect ratio and orientation to each pore. An array of just the aspect ratio and orientation is the first numeric representation of the shape geometries. We note that these features minimally encode the "bulk" form of the shape effectively reducing the pore shape to an equivalent ellipse. ML models that use this input are given the appendage “ARO” for aspect ratio and orientation.

However, a caveat to this representation that will be shown later, is that the orientations of bounding boxes are biased toward discrete values. To overcome this deficiency the orientation and aspect ratio of realistic pores are instead determined by their Fourier ellipse described in Section 6.5.1.

## Simple Metrics

Another way to describe the encountered pore shapes is with the descriptive metrics of perimeter, circularity, solidity, and mean diameter. The mean diameter is the multiple of 2 and the average distance between the pore centroid and its contour. These metrics are appended to the array of orientation and aspect ratio to create a new numeric input representation referred to as the simple metrics. While relatively cheap to compute, these simple metrics fail to uniquely define realistic pore shapes (two different shapes can have the same metrics) and cannot be inverted to reproduce the original shape. ML models that use this input will be associated with the word “Simple.”

## Fourier Coefficients

Another manner in which each pore shape is represented is through complex Fourier series coefficients [84–87]. For this method, the x-y coordinates of the shape contour are sampled at equidistant distances with respect to contour length. It should be noted that this sampling procedure ensures that the x versus distance and y versus distance signals are periodic. Recalling that the coordinate system of the shape contours is centered on the shape centroid, we sampled the shape contours starting at the intersection of the contour and the positive x-axis, proceeding in the counter-clockwise direction, using 1,023 points. If the more than one contour and x-axis intersection existed, then sampling began at the intersection with the largest x-value.

The x- and y-distance information can be considered simultaneously as a single complex signal where the real and imaginary components are the x and y coordinates, respectively. Under this formulation, the periodic, complex-valued function can be represented as a Fourier series wherein the Fourier coefficients,  $C_i$ , encode all the information required to reproduce the original shape contour. The Fourier coefficients are the frequency spectrum of the shape, and because the shape was sampled with 1,023 points, discrete Fourier transform routines return Fourier coefficients ranging from  $C_{-511}$  to  $C_{511}$ . The Fourier coefficients are complex-valued and note that because the original

x- and y-distance signal is complex-valued the Fourier coefficients need not be complex conjugates as is the case when dealing with strictly real signals. Most ML models cannot readily accept complex values. As such, the Fourier coefficients are further processed by splitting them into real and imaginary components before ingestion by ML models. If all the Fourier coefficients are used,  $C_{-511}$  to  $C_{511}$  then this representation will be an array of length 2048.

The Fourier coefficients demonstrate a number of useful properties. We note that simple transformations can be applied to the Fourier coefficients to translate, rotate, and scale the original contour. Additionally, the lower order harmonics describe the “bulk” form of the shape, whereas the higher order harmonics contain information about the finer spatial details of the shape. Consequently, the higher order harmonics can be filtered out to produce a smoothed version of the shape. In particular, when only the first harmonic is used (number of harmonics,  $n_h$ , equals 1) the shape contour is reduced to an ellipse, herein described as the “Fourier ellipse”. This suggests an alternate method to identify the shape aspect ratio and orientation by referencing the Fourier ellipse of the shape instead of the bounding box. A comparison of the aspect ratios and orientations returned by the bounding box and Fourier ellipse methods is provided in Figures 6.3a and 6.3b, respectively, for a set of realistic pore geometries scaled to an equivalent diameter of 150 nm. As seen, the values for aspect ratio are in good agreement, but the orientations returned by the bounding box method are biased toward values of 0, 45, 90, and 135 degrees whereas the orientations determined by inspecting the Fourier ellipse are continuous. The tendency of the bounding box to assume discrete orientations is associated with the task of placing a rectangular box into image (pixelated) space. As such, for the remainder of this work the Fourier ellipse values for aspect ratio and orientation are used when referencing the simple descriptive metrics.

Great care should be used when using the Fourier coefficients as they also encode information about the x and y signal mean (which is not necessarily the shape centroid) and the starting location of the parameterized path (here, the positive x-axis).

At a high level, machine learning data sets are categorized as either "structured" (tabular) or "unstructured" (image or signal-like). In structured data, column identifiers

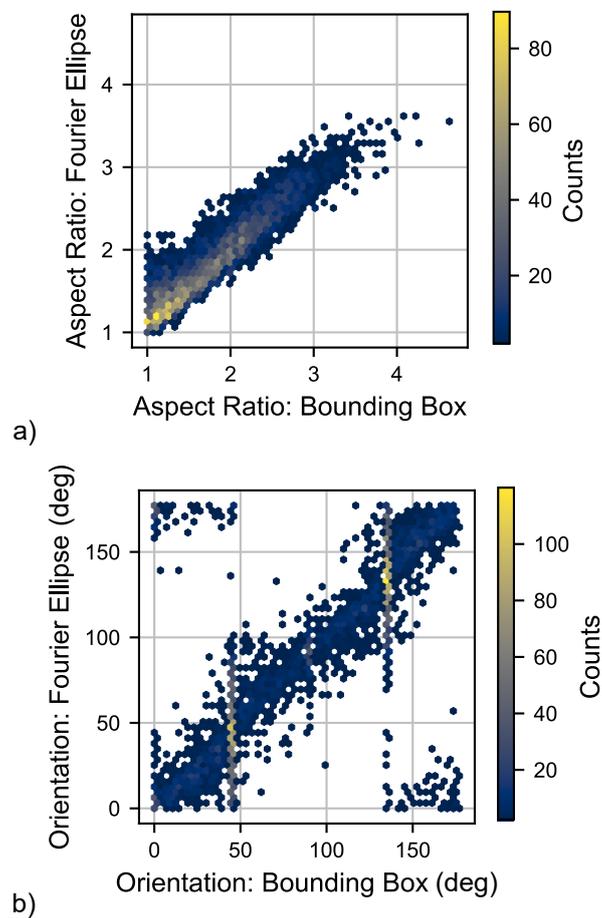


Figure 6.3. Comparison of a) aspect ratio and b) orientation of realistic pore geometries scaled to an equivalent diameter of 150 nm using the bounding box and Fourier ellipse methods. Clusters in the upper left and lower right quadrants of b) are an artifact of phase wrapping and are not indicative of gross disagreement between the methods.

are constant for all training instances. In unstructured data, for example, an image, every pixel need not represent the same thing. That is, in an image there is no guarantee that some arbitrary pixel – say row 2 column 40 – corresponds to the same object (like a tree, or a car) in every image.

The pore images and paths are unstructured. Thus, the Fourier series and resultant coefficients can be consider as a way of converting the pore path to a structured representation. This is useful because many machine learning models require structured data.

The utility of the Fourier coefficient representation of shapes in pattern recognition and machine learning tasks was recognized as early as the 1970s to classify hand-written characters and digits [88, 89]. Interestingly, Zahn and Roskies [89] found that even just the amplitudes of two Fourier coefficients were sufficient to separate some digits into differentiable clusters. This suggest that for some tasks, the Fourier descriptors may provide all the feature engineering required to adequately perform classification and regression tasks. As a more recent example, Rice et. al. [90] classified colloidal nanocrystals into basic shapes – such as rod, pentagon, triange, etc. – using Fourier coefficients.

While the Fourier coefficients uniquely encode the complete shape contour (that is, the shape can be reproduced to its original form from the Fourier coefficients) and the frequency representation of the pores may effectively featurize the shape, they are difficult to interpret. Although each Fourier coefficient can be independently determined, when they coalesce to construct a shape small changes in a coefficient can drastically alter the final contour. As such, it is difficult to prescribe a meaning to relative changes in Fourier coefficients for all but basic shapes (square, triangle, rod, etc.).

Finally, we note that a strong visual interpretation exists for the Fourier coefficients: because the coefficients are complex valued each can be considered a vector on the complex plane. To retrieve the original shape contour, each vector will rotate and “trace” out a circle according to its periodicity. For example, the coefficient  $C_1$  is the first positive harmonic and will rotate in the counter-clockwise direction, completing one rotation every period.  $C_{-3}$  is the third negative harmonic and completes 3 rotations per

period in the clockwise direction. Note that the direction of rotation (positive coefficients rotate counter-clockwise) is a direct consequence of how the shape contour was sampled. The summation of the rotating vectors will trace out the shape contour over the course of one period.

### Standardized Array

Deep learning (DL) is distinguished as a set of specialized algorithms constructed of layers of nonlinear processing units. The interconnected layers form a network capable of learning rich data representations [91, 92]. Thus, deep neural networks (DNNs) can learn from “raw” data without having to explicitly construct a representation of the input object. A main benefit to this approach is that the model can remain relatively free from bias introduced by the choice of input representation.

Convolutional neural networks (CNNs), a type of deep learning network, are particularly apt at image processing tasks such as object detection and segmentation. Attempting to hand-craft a robust set of feature detectors designed to detect, segment, or classify an object is a difficult task. For example, accommodating slight changes in object orientation, shading, and location necessitate the creation of additional detection rules. CNNs overcome these issues by learning local patterns in the data and building on these patterns in a hierarchically fashion to learn increasingly complex and abstract concepts [93]. Recently, CNNs have been applied to microstructure images to successfully investigate microstructure-property linkages [94–96].

The pore shapes are originally extracted as images. These images can be minimally processed and used as inputs to a convolutional neural network. The original binary images of the pore shapes are, by definition, not normalized to size and have different widths and heights in terms of pixels. While there are algorithmic adaptations such that a CNN can make use of dissimilar input sizes the simplest approach is to standardize the input array.

To this end, we elected a standard grid of 64 by 64 square elements. The extent (physical size) of the grid is chosen so that the largest pore path (scaled contour) – while

only considering the training set – fits within the inner 62 by 62 grid elements. Once the grid extent is determined, pore paths are effectively “stamped” into the grid. For a single pore the result is an array (pixelated image) composed of 64 by 64 elements representing the scaled pore shape. It is emphasized that under the standardized array approach, relative differences in pore height and width are preserved even after pores are normalized to the same equivalent circular diameter.

### **Pore Representation Summary**

For an example pore, the binary array and contour extracted from the SEM image are shown in Figures 6.4a and b, respectively. The pore path, after scaling, along with the pore shape when all but the first  $n_h$  harmonics are filtered out is given in Figure 6.4c. As seen, the lower harmonics capture the bulk of the shape and when only the first harmonic is retained, the shape is reduced to its Fourier ellipse. The Fourier ellipse is used to assign an aspect ratio and orientation to the pore when using said values to describe the pore or when using the full set of simple metrics. Finally, Figure 6.4d shows the example pore as a standardized array to be used as an input for the CNN.

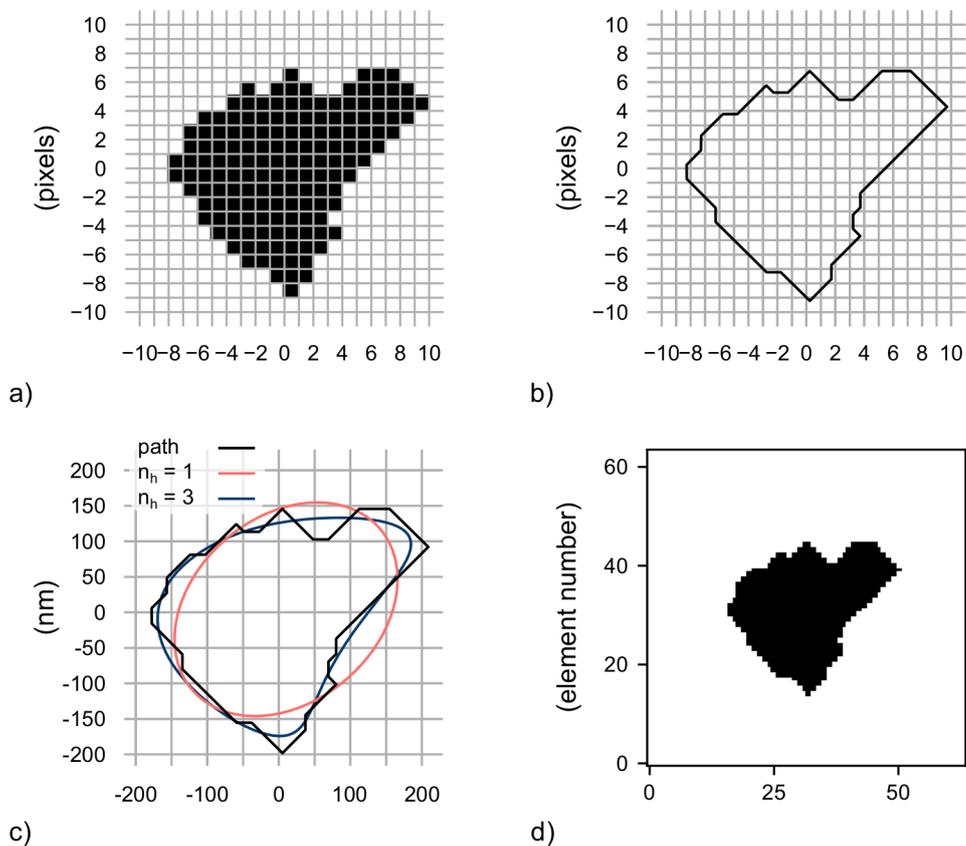


Figure 6.4. a) The original binary image of an example pore. b) The extracted pore contour. c) The black line is the scaled path such that the pore has an equivalent circular diameter of 300nm. The salmon colored line is the scaled pore path when carrying only the first harmonic ( $n_h=1$ , the “Fourier ellipse”). The navy colored line is the scaled pore path using the first three harmonics ( $n_h=3$ ). d) The standardized binary representation of the example pore. Dataset: 300 nm. Implicit pore id number: 400.

### 6.5.2 Model Types

While the resultant critical threshold velocities of pore geometries acquired via CTH simulations are insightful, determining the threshold velocity of a new pore shape requires another simulation. Additionally, we hypothesize that certain small perturbations in the pore contour can considerably shift the critical threshold velocity. To quickly assess the critical threshold velocity of a new pore, or an altered version of a previously simulated pore, we developed machine-learned surrogate models.

We used three types of machine learning models; a random forest (RF) implemented with scikit-learn [77], an extreme gradient boosting machine (XGB) through the XGBoost [97] package, and a convolutional neural network (CNN) modeled using Keras [98]. All of these algorithms are for supervised learning tasks. The critical threshold velocities served as the data labels (or annotations) during model fitting.

For training and evaluation, each data set (150 nm or 300 nm) was considered separately and randomly split into training, validation, and test sets. The training, validation, and test sets contained 80%, 10%, and 10% of the original data, respectively.

For each data set, random forest models were trained using aspect ratio and orientation (ARO), full set of simple metrics (Simple), and Fourier coefficients as model inputs. The hyperparameters were optimized using Hyperopt [99] to explore and exploit the hyperparameter space while evaluating the mean squared error of the validation set on the fitted model over 500 trials. To kick off the optimization process a prior distribution is applied to each hyperparameter that is allowed to vary. The hyperparameter prior distributions for the RF models are summarized in Table 6.2. In this table distribution bounds are listed in square brackets as [lower bound, upper bound, step size]; the step size is only applicable to discrete distributions. Curly braces indicate a choice between items in a set and all items are initially equally weighted. For the model using the the Fourier coefficients as inputs, the number of harmonics retained is treated as an additional hyperparameter. For example,  $n_h=3$  signifies that the Fourier coefficients from  $C_{-3}$  to  $C_3$  are supplied to the model.

Table 6.2. The initial range of the random forest model hyperparameters.

<b>Hyperparameter</b>	<b>Initial Distribution</b>
max depth	uniform [1, 100, 1]
number of trees	uniform [50, 300, 5]
minimum samples per leaf	uniform [1, 10, 1]
max features	choice {'all', 'sqrt', 'log2'}
number of harmonics, $n_h$	uniform [1, 511, 1]

The XGB model was applied to both data sets but using only the Fourier coefficient representation of the pore shapes. Similar to the RF models, the XGB hyperparameters were tuned using Hyperopt with respect to the validation set. The initial parameter specifications are provided in Table 6.3.

The CNN was the last model implemented. This network was composed of four sequential 3x3 convolutional kernels and 2x2 max pooling layers followed by a densely connected neural network with 3 layers. The neural network layers had 64, 32, and 1 units, respectively, with the last unit providing the network's predicted critical threshold velocity. Both the convolutional kernels and fully connected layers were followed by

Table 6.3. The initial range of the extreme gradient boosting machine hyperparameters.

<b>Hyperparameter</b>	<b>Initial Distribution</b>
max depth	uniform [1, 20, 1]
minimum loss reduction, $\gamma$	log uniform [ $\log(0.0001)$ , $\log(10)$ ]
learning rate	log uniform [ $\log(0.05)$ , $\log(0.1)$ ]
$L_1$ regularization, $\alpha$	uniform [0.0, 3.0]
$L_2$ regularization, $\lambda$	uniform [0.0, 3.0]
column sample by tree	uniform [0.6, 1.0]
number of harmonics, $n_h$	uniform [1, 511, 1]

the exponential linear unit (ELU) activation; with the exception of the final neural network unit which was followed with a linear activation. Despite the large number choices available when designing a deep network architecture – such as: number of layers, type of activation functions used, size of convolutional kernels, type and magnitude of regularization – the initial architecture yielded performance on par with the tuned RF and XGB models. For this reason, and because deep neural networks require longer training times, the CNN hyperparameter space was not further explored.

## 6.6 Results and Discussion

### 6.6.1 Ellipse Simulations

The critical threshold velocities for the elliptical pore structures scaled to equivalent diameters of 150 nm and 300 nm at various orientations and aspect ratios are given in Figure 6.5. Recall that the shock enters the simulation domain from the left and that pore orientation is measured counter-clockwise from the positive x-axis to the major axis of the ellipse. Also, due to the symmetry of true ellipses, orientations from 90 to 180 degrees would mirror ellipses with orientations from 90 to 0 degrees so simulations are not needed in this range.

We describe pores with low critical threshold velocities as “sensitive” because they require a lower flyer plate velocity – and thus a lower shock pressure – to trigger the critical criterion. Immediately, a trend is observed wherein sensitivity increases (lower critical threshold velocities) with high aspect ratio pores with their major axis parallel to the incoming shock (0 degree orientation). Furthermore, high aspect ratio elliptical pores oriented at 90 degrees (major axis is perpendicular to the incoming shock) are the least sensitive.

When comparing across pore sizes (150 nm or 300 nm equivalent diameter), as expected, the smaller pores require higher incoming flyer plate velocities to achieve criticality. Figures 6.5b and c are slices (or views) into Figure 6.5a. These figures reveal some interesting trends: pore sensitivity generally decreases with orientation with the 300 nm

equivalent diameter pores exhibiting a sharp transition around 50 degrees; and aspect ratio can either have a sensitizing or desensitizing effect depending on the orientation of the pore.

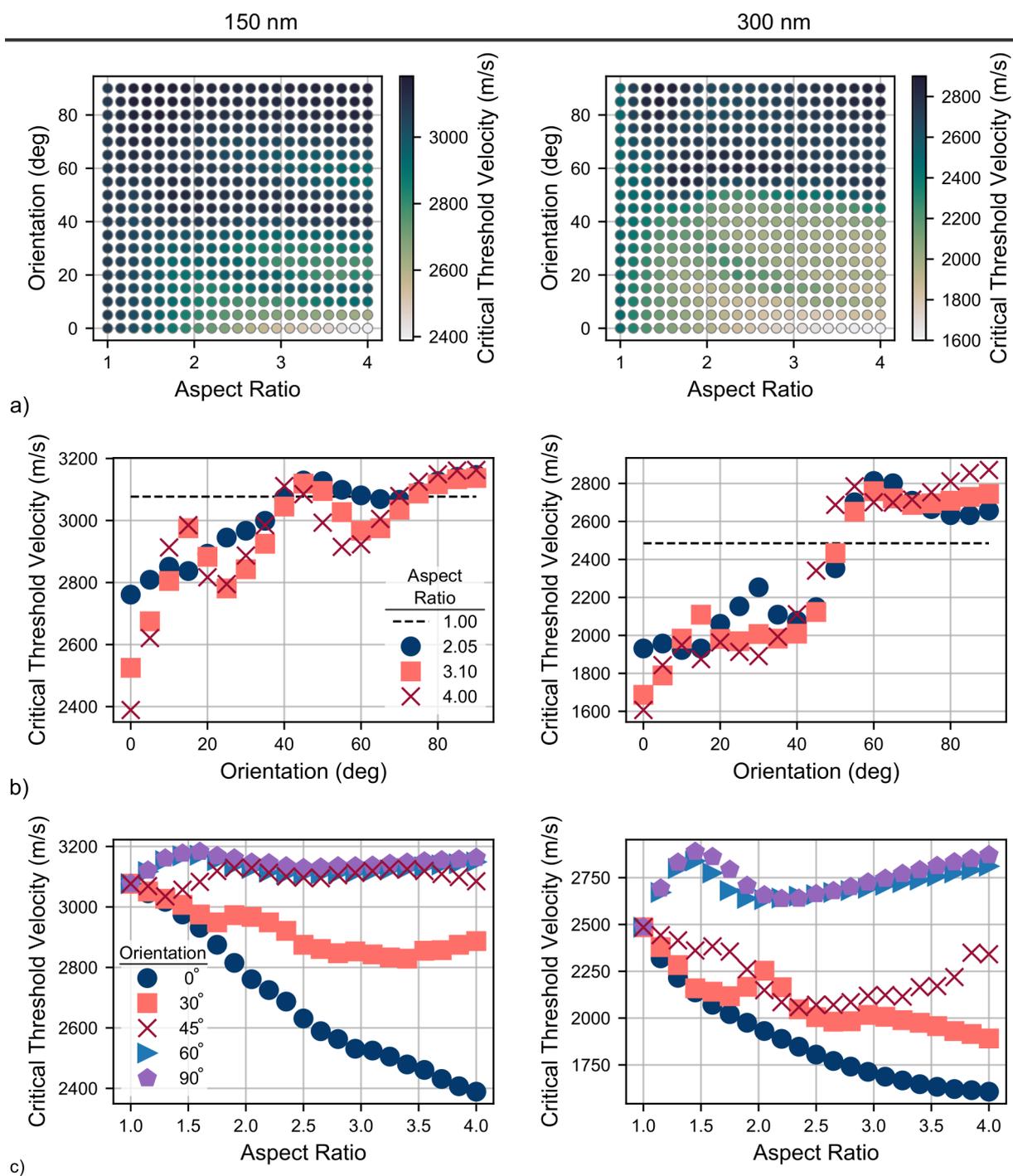


Figure 6.5. The critical velocity thresholds for various ellipses scaled to equivalent diameters of 150 nm (left) and 300 nm (right). a) The critical threshold velocity with respect to ellipse aspect ratio and orientation. b) The critical threshold velocity with respect to orientation for select aspect ratios. c) The critical threshold velocity with respect to aspect ratio for select orientations.

### 6.6.2 Distribution of Critical Thresholds

Moving to the simulation outcomes of the realistic pore geometries, the critical threshold velocities for all pores are plotted as a histogram in Figures 6.6a and b for pore sizes of 150 nm and 300 nm, respectively. The critical velocities of pores scaled to 150 nm equivalent diameter appear to have a limit on the maximum threshold achievable near 3200 m/s. This indicates that no pore, regardless of its shape, can remain sub-critical when presented with incoming flyer plate velocities in excess of 3200 m/s at this pore size. We expect that this upper bound will increase as the equivalent pore diameter decreases approaching the “homogeneous” limit, defined as the flyer plate velocity that would result in a critical condition (initiation of material and sustainment of a shock wave) even in the absence of a pore heterogeneity.

The critical threshold velocity of a similarly scaled circular shape is displayed on all parts of Figure 6.6 as a dashed line. The circle critical threshold velocity is at the 76.6<sup>th</sup> percentile and 87.5<sup>th</sup> in the distribution for the 150 nm and 300 nm data sets, respectively. This suggests that simulations which use a circle as an idealized pore geometry significantly underestimate the sensitivity when compared to actual pore shapes.

Figure 6.6c compares the probability density functions for the two pore equivalent diameters. This plot only considers the common pore shapes between the data sets. Further inspection reveals that when moving from 150 nm to 300 nm equivalent diameter all pores experience a reduction in reported critical velocity threshold with the average reduction being 860 m/s; again illustrating the strong influence of pore size.

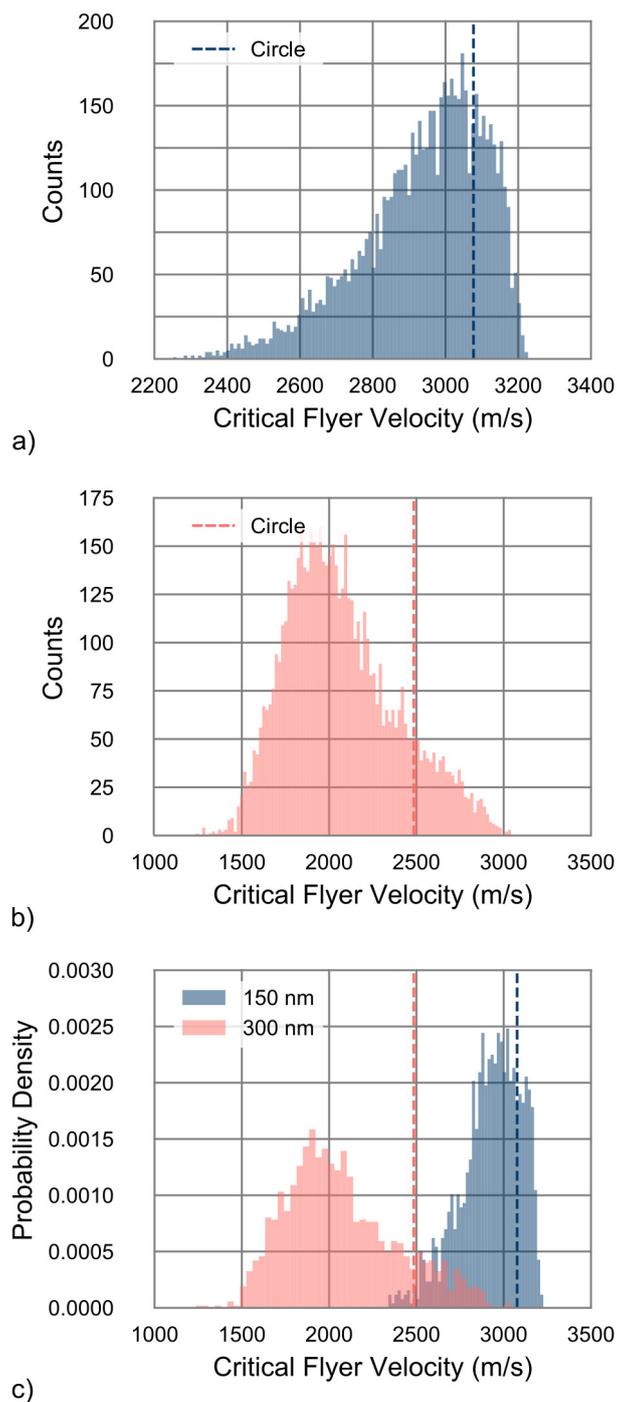


Figure 6.6. a) Distribution of critical velocity thresholds for realistic pores scaled to 150 nm equivalent diameter. b) Distribution of critical velocity thresholds for realistic pores scaled to 300 nm equivalent diameter. c) Distributions of critical velocity thresholds at 150 nm and 300 nm equivalent diameters only considering the pore shapes share between the two data sets (1452 pores).

### 6.6.3 Realistic Pore Aspect Ratio and Orientation

The results of the simulations conducted on elliptical pores demonstrate that the aspect ratio and orientation are strong drivers of pore sensitivity. To compare this result to realistic pore geometries, the realistic pore critical threshold velocities are plotted according to their aspect ratio and orientation as shown in Figure 6.7. The same general trend found in the elliptical pores exists here in that high aspect ratio pores with their major axis aligned with the incoming shock (orientations of either 0 or 180 degrees) have the lowest critical velocity thresholds. Likewise, vertical (90 degree orientation) pores have the highest critical velocity thresholds with an increase in aspect ratio actually serving to decrease sensitivity.

However, unlike the elliptical pores, aspect ratio and orientation alone are not enough to specify the critical threshold velocity. This is evident because multiple pores with different critical threshold velocities occupy the same region in aspect ratio - orientation space. Intuitively, the difference in critical threshold velocities experienced by pores with the same aspect ratio and orientation is attributed to the “fine” or “higher order” shape features; that is, the deviations from the idealized elliptical shape and contour roughness.

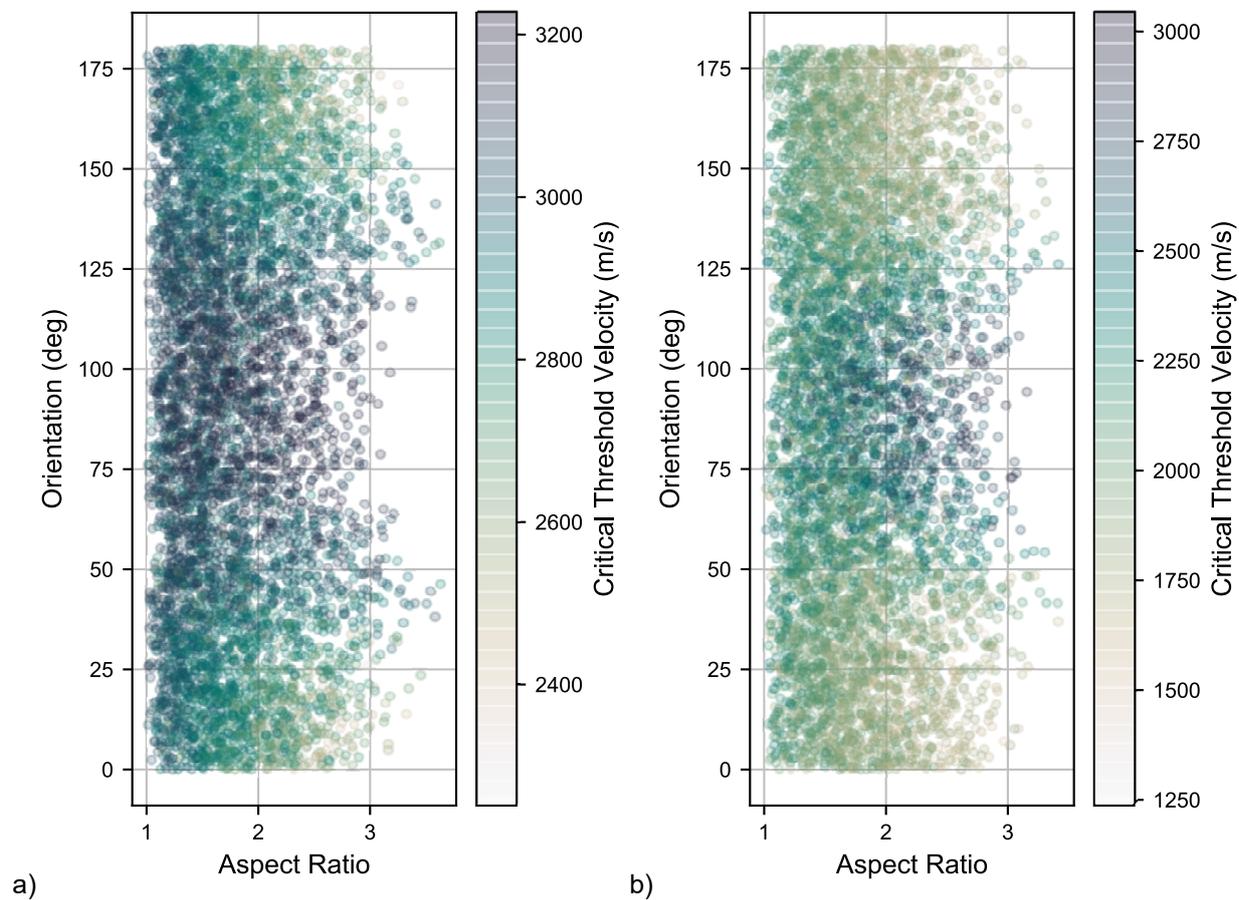


Figure 6.7. The critical threshold velocities of realistic pores with respect to their measured aspect ratio and orientation. a) The data set of pores scaled to 150 nm equivalent diameter. b) The data set of pores scaled to 300 nm equivalent diameter.

#### 6.6.4 Model Comparison

The performance of the machine-learned models is measured using the  $R^2$  score, mean squared error (MSE), and the mean absolute error (MAE). These evaluation measures are reported on the training, validation, and test sets. While common procedure, it bears repeating that the models weights were fit while accessing the training set, the model hyperparameters were tuned to maximize performance while evaluating the validation set, and the test set was only evaluated once after final model selection. Therefore, the test set scores serve as the best approximation of the models ability to generalize to unseen instances (new data). The performance measures for the 150 nm and 300 nm equivalent diameter data sets are tabulated in Tables 6.4 and 6.5, respectively.

Table 6.4.  $R^2$ , MSE, and MAE performance metrics for all models on pores scaled to 150 nm equivalent diameter categorized by set designation – train, validation, or test.

	$R^2$			Mean Squared Error			Mean Absolute Error		
	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
<b>RF ARO</b>	0.6969	0.5951	0.6256	9061.72	11357.40	11244.87	70.25	80.94	79.49
<b>RF Simple</b>	0.7695	0.6272	0.6721	6892.02	10458.29	9847.84	60.25	76.50	72.59
<b>RF</b>	0.9748	0.8168	0.8236	752.07	5139.79	5298.39	19.32	51.83	53.37
<b>XGB</b>	0.9999	0.8192	0.8274	2.10	5072.21	5185.57	1.02	49.38	50.85
<b>CNN</b>	0.8919	0.8306	0.8487	3230.68	4752.13	4545.63	42.38	50.10	49.92

Table 6.5.  $R^2$ , MSE, and MAE performance metrics for all models on pores scaled to 300 nm equivalent diameter categorized by set designation – train, validation, or test.

	$R^2$			Mean Squared Error			Mean Absolute Error		
	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
<b>RF ARO</b>	0.5832	0.5297	0.4747	42376.85	51712.36	50969.87	157.88	176.75	173.03
<b>RF Simple</b>	0.6706	0.5835	0.5685	33489.94	45792.87	41865.06	138.89	164.66	156.02
<b>RF</b>	0.9642	0.8271	0.7834	3643.97	19011.74	21016.85	44.23	106.59	108.36
<b>XGB</b>	0.9870	0.8484	0.8061	1319.61	16669.59	18811.32	25.77	99.55	102.81
<b>CNN</b>	0.9400	0.8693	0.8564	6094.67	14365.54	13932.92	60.67	89.92	88.03

The relative ranking of the models is similar for both data sets. The random forest model trained using only the pore aspect ratio and orientation has the lowest performance with  $R^2$  scores of 0.63 and 0.48 on the test sets – depending on which data set is referenced. This is not unexpected as Figure 6.7 already demonstrated that critical threshold velocities are not distinguishable using solely these descriptors. When the remaining simple metrics of solidity, circularity, perimeter, and mean diameter are added to the random forest model the model performance improves. The test set  $R^2$  increases by roughly 7 percentage points which equates to a reduction of the mean absolute error by 7 m/s on the 150 nm equivalent diameter pores and by 17 m/s on the 300 nm equivalent diameter pores.

The RF and XGB models using the Fourier coefficient representation of the pore as inputs showed significant performance gains on both pore sizes. Interestingly, for all four models the optimal number of harmonics,  $n_h$ , as determined by the hyperparameter optimization process was three. This is not entirely surprising, as already noted, the bulk of the pore information is contained in the lower order harmonics. The lower order harmonics prescribe the basic outline of the shape. Higher order harmonics recreate the actual pore contour by adding small deviations from the base outline. Generally, sequentially higher ordered harmonics add successively smaller deviations. As more higher order harmonics are retained the pore contour will approach the original contour extracted from the SEM-obtained images, however, at some point, the hydrocode outcome will no longer change with the increased resolution of the correct pore shape. That is, the hydrocode is not able to capture the differences made by small deviations in the pore contour – if physically present at all. This effect is slightly exacerbated by the fact that the CTH outcomes are reduced to a single critical threshold velocity for each pore and because this velocity is only resolved to within 1 m/s. As a brief aside, the choice to resolve the critical threshold velocity to 1 m/s was made: 1) to reduce the number of simulations required, 2) because 1 m/s is small relative to the range of critical thresholds observed in preliminary tests, and 3) because the numerical limit of the hydrocode, in this regard, has not yet been approximated. Returning to the discussion at hand, we emphasize that retaining only the first three harmonics severely

smooths the pore contour (see Figure 6.4c). Furthermore, the performance gain seen by using Fourier coefficient input representation reveals that even just the bulk pore shape carried by the first three harmonics possesses more discriminative information than the full set of simple metrics. The RF and XGB models also exhibit high  $R^2$  scores on the training, in both the absolute sense and relative the validation and test sets. This suggests that, despite the hyperparameter optimization to the validation set, these models are overfitting the training data. None of the final model hyperparameters are at, or near, the bounds set by the prior distribution so it cannot be said that the models were unable to reduce model complexity (by adding regularization or decreasing tree depth).

The convolutional neural network, which acts on the standardized array representation of the pores (Figure 6.4d), manifests performance on par with the Fourier descriptor RF and XGB models on the 150 nm data set and surpasses these models on the 300 nm data set. Specifically, on the 300 nm data set the CNN reduces the test set mean absolute error by nearly 15 m/s relative to the XGB model. Recall that the CNN did not undergo hyperparameter tuning. As such, this model has potential for further improvement.

Overall, larger prediction errors exist on pores scaled to 300 nm equivalent diameter compared to those scaled to 150 nm equivalent diameter. This is likely because the range of critical threshold values observed on the 300 nm data set is greater than that on the 150 nm data set; approximately 1800 m/s to 1000 m/s. However, additional investigation is required to state this with certainty.

Finally, a partial depiction of model results is provided in Figure 6.8 by plotting the predicted critical threshold velocities to those obtained from the CTH simulations for the 300 nm data set random forest models using the aspect ratio-orientation, simple metrics, and Fourier coefficient input representations; subfigures a, b, and c, respectively. This figure only contains the pore shapes listed in the test set. As acknowledged, the RF model prediction error decreases when using the simple metrics and more so with the Fourier coefficients.

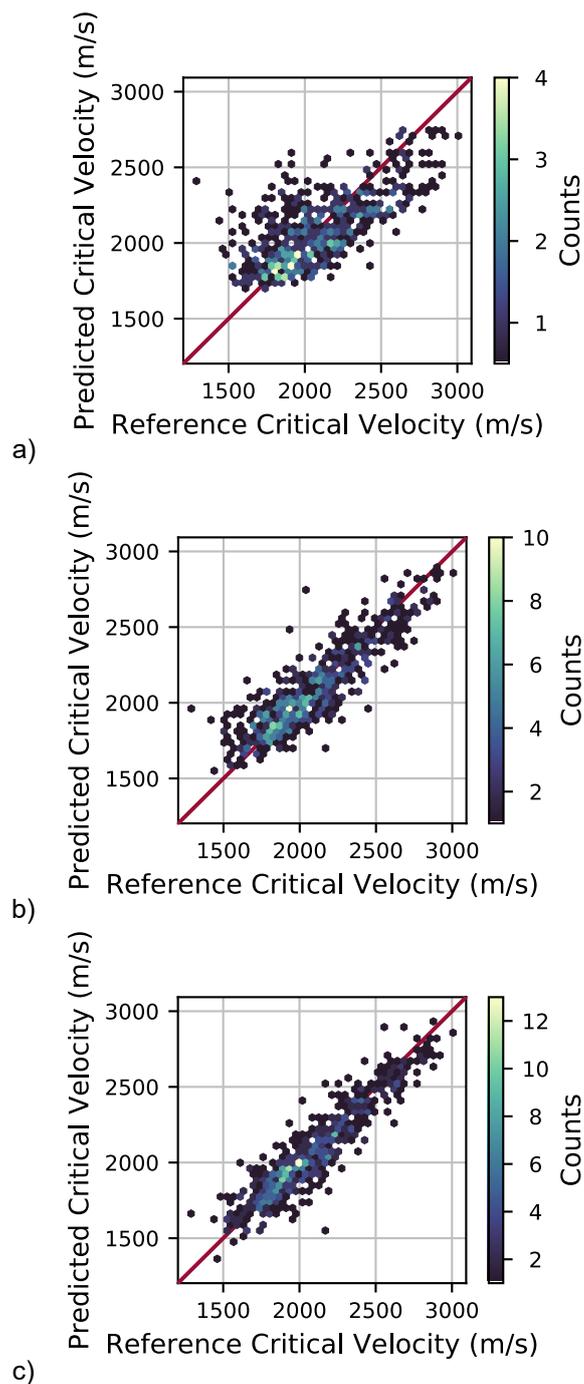


Figure 6.8. The random forest model critical threshold velocity predictions on the test set versus the reference velocities obtained via CTH for pores scaled to 300 nm equivalent diameter when using a) the aspect ratio and orientation, b) the simple metrics, and c) the Fourier coefficients as model inputs.

### 6.6.5 Additional Viewpoints

A primary goal of this work is to identify which shape features increase or decrease pore sensitivity to shock loading. The influence of aspect ratio and orientation immediately fall out when considering pores idealized as ellipses and these trends generally hold for realistic pores. Moreover, we observe that models with access to more pore shape information demonstrate better predictive performance. However, determining exactly which aspects of a realistic pore contribute to sensitivity is a difficult task.

For example, CNNs typically have greater interpretability than their deep learning counterparts. Because the CNN acts on image data (at least herein), the learned convolutional kernels can be visualized. Additionally, the activations can be tracked as a pore shape propagates through the network and theoretical pore shapes which maximize filter activations can be generated. These techniques provided assurance that the CNN was learning increasingly complex patterns built from simple learned line and edge detectors, however, no single shape feature stood out from the pack. The CNN can also be decomposed into two parts, the convolutional base and the model “head”. The convolutional base consists of convolution kernels (thus its namesake) with associated ELU activation functions and max pooling operations. The convolutional base output units are “flattened” and fed to the simple forward feed neural network which acts as the CNN head. The convolutional base acts as a feature extractor and transforms the input into a representation more suitable for machine learning. Rather than pass this learned representation onto the neural network head we intercept it for interrogation. That is, all of the pores are passed through the CNN base and collected as a new pore representation. For our CNN architecture, the learned pore representation, after flattening, is an array of length 1,056.

To explore the 1,056<sup>th</sup> dimensional feature space we reduced the space to two dimensions via a t-distributed stochastic neighbor embedding (tSNE) implemented with scikit-learn (version 0.23.1 with default settings). For the 300 nm data set the pore contours are plotted according to their spatial location in the 2D embedding as seen in Figure 6.9. The 2D tSNE can be thought of as a complex manifold through the high-

dimensional CNN-extracted feature space. Due to the complexity of the manifold, and because the meaning of the dimensions in the CNN feature space is unknown, the axes in Figure 6.9 have no physical meaning. What should be emphasized is that t-distributed stochastic neighbor embedding preserves distance. Therefore, pore-to-pore distance in the figure directly corresponds to pore similarity in the CNN-extracted feature representation. Qualitatively, adjacent pores have striking similarities indicating that the CNN has learned useful filters.

A color scheme is also applied to the pores in Figure 6.9. The internal, or “fill”, color of the pore denotes the CTH simulation critical threshold velocity and the outline, or “border”, color marks the predicted critical threshold velocity produced by the CNN. The border and fill colors are, generally, in good agreement again illustrating the predictive power of the CNN. Note that the tSNE algorithm has no knowledge of the critical threshold velocities. This is important because it aligns with our intuition that similar pores should have similar critical threshold velocities. Here, pore adjacency (ie; similarity as viewed from the CNN feature space) varies smoothly with observed critical threshold velocities. This method of plotting actual pore contours according to their location in a 2D space is not unique to the embedding of the CNN-extracted representation. In fact, the aspect ratio and orientation representation of the pore is already two-dimensional and does not require an embedding. Like unto Figure 6.9, the 150 nm and 300 nm equivalent diameter pore shapes are plotted according to their aspect ratio and orientation in Figure 6.10. This figure is essentially a recreation of Figure 6.7 except now the pore shapes are subsampled so that they do not overlap. Again, the fill color indicates the simulation determined critical threshold velocity and the border color displays the predicted critical threshold velocity using the RF ARO model. Examination of the pore shape border colors in Figures 6.7a and b show that the RF ARO model output varies smoothly and continuously over the aspect ratio - orientation domain. Here, adjacent pores are, by definition, similar in aspect ratio and orientation. While pores with nearly identical aspect ratios and orientations definitely look similar, the authors argue that the adjacent pores (similar pores) in Figure 6.9 have common features beyond aspect ratio and orientation. This is called out in greater detail in Figure 6.11.

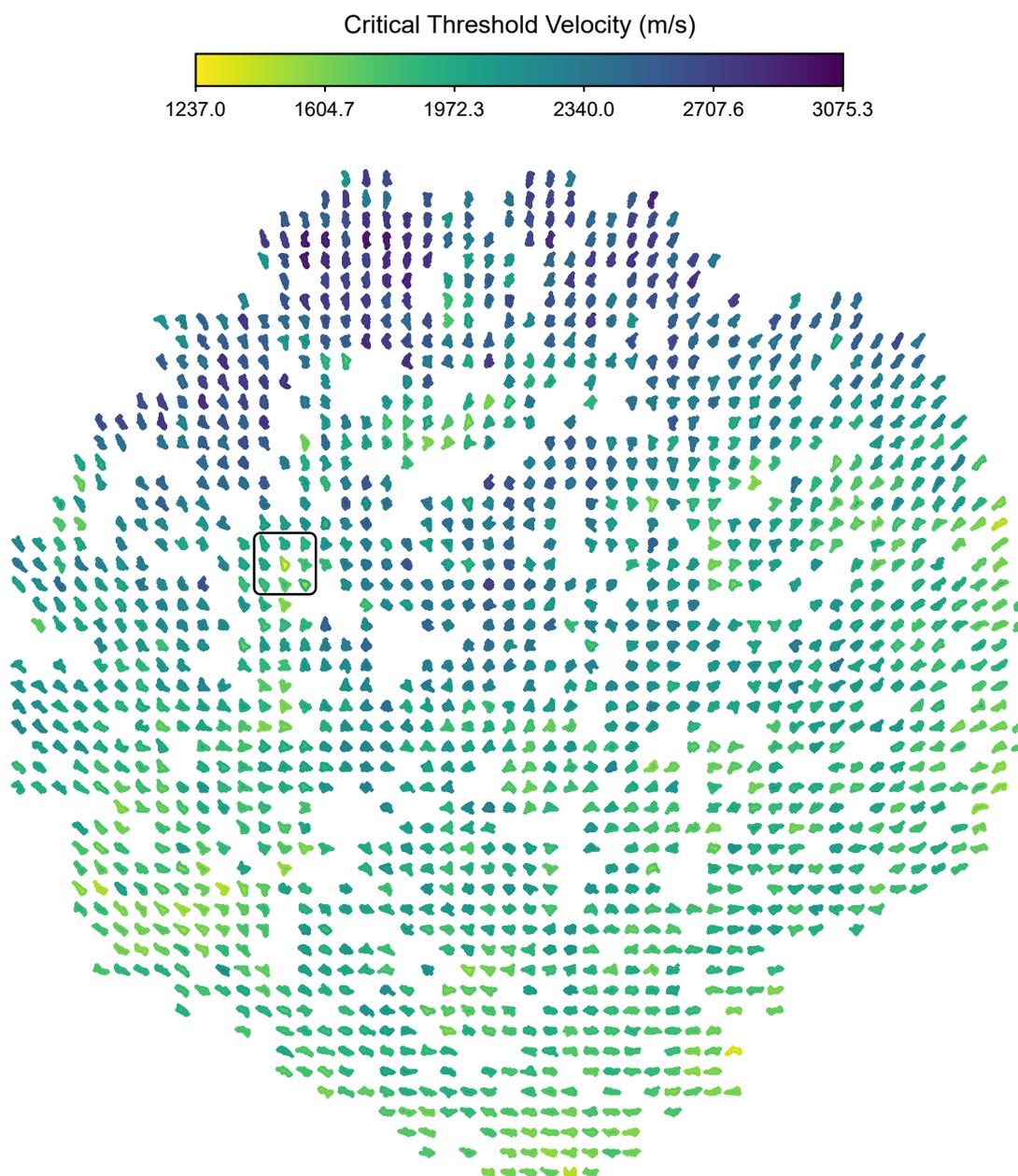


Figure 6.9. The 2D t-distributed stochastic neighbor embedding of the of 300 nm equivalent diameter pores represented in the CNN-learned feature space. Pore fill color indicates the critical threshold velocity obtained from the CTH simulations and pore border color marks the predicted critical threshold velocity from the CNN. The figure is supplied in vector format so that if viewed electronically it can be enlarged indefinitely. Not all pores shapes are displayed; pores are subsampled randomly so that pore shapes do not overlap.

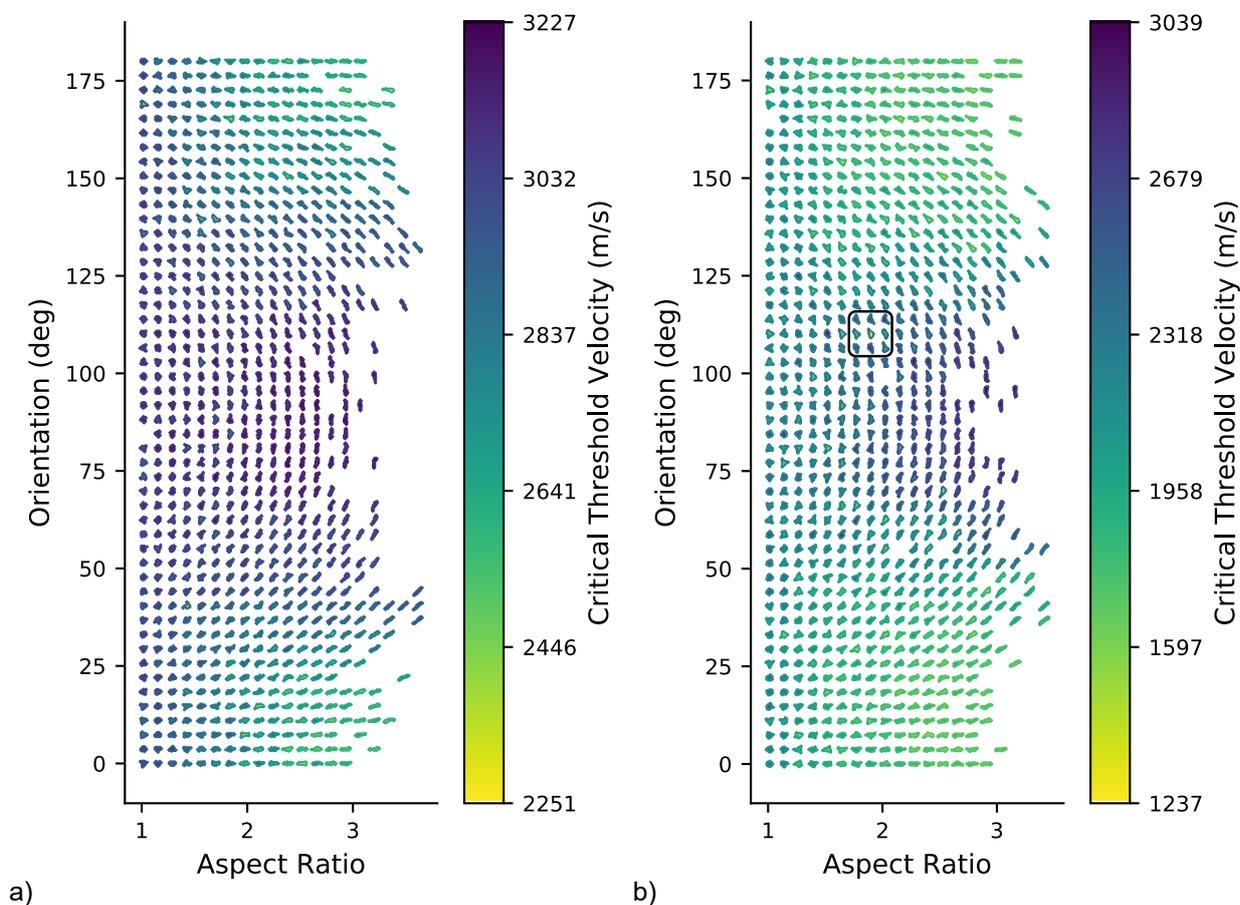


Figure 6.10. Pore shape plotted according to their location in aspect ratio - orientation space for a) the data set of pores scaled to 150 nm equivalent diameter and b) the data set of pores scaled to 300 nm equivalent diameter. Pore fill color indicates the critical threshold velocity obtained from the CTH simulations and pore border color marks the predicted critical threshold velocity from the CNN. The figure is supplied in vector format so that if viewed electronically it can be enlarged indefinitely. Not all pores shapes are displayed; pores are subsampled randomly so that pore shapes do not overlap.

Both Figures 6.9 and 6.10b have a region encompassed by a square box. Figure 6.11a shows the pores taken from Figure 6.10b (RF ARO model) and are given labels 1 through 9. Likewise, Figure 6.11b contains the pores from the boxed region of Figure 6.9 (tSNE and CNN model) and are also labelled 1 through 9. The pores in these regions are from the 300 nm data set. These regions are called out, specifically, because in each set the pore labelled as number 5 is the same pore. This pore is interesting because both the RF ARO and CNN model under predict the critical threshold velocity with large error.

These regions, while not otherwise unique, also serve to highlight some points already discussed. The border colors in Figure 6.11a do not vary much from pore to pore within this region. This is because, as discussed, the RF ARO model predictions vary smoothly over the aspect ratio - orientation domain such that adjacent pores are assigned the same predicted value. However, pores with similar aspect ratios and orientations need not have the same critical threshold velocities as demonstrated by the variance in fill colors seen in these pores. This discrepancy in fill colors is not unexpected because beyond orientation and aspect ratio these pores do not possess many common features. It could be argued that in Figure 6.11a, only pores 4 and 9 closely resemble pore 5. The disparity between border color and fill color (prediction and reference critical threshold velocity) among these pores represents that the RF ARO model predictions are in error on every pore shape in this region.

Shifting to Figure 6.11b, it is immediately seen that the border color and fill color are in close alignment in accordance with the accuracy of the CNN predictions. Additionally, it could be argued that pores 1, 7, 8, and even pores 4 and 2 are similar to pore 5 in features beyond aspect ratio and orientation. Again, providing evidence that the CNN has extracted pore shape characteristics and can group them appropriately. Finally, the pore border colors vary significantly from pore to pore indicative of the increased model flexibility held by the CNN.

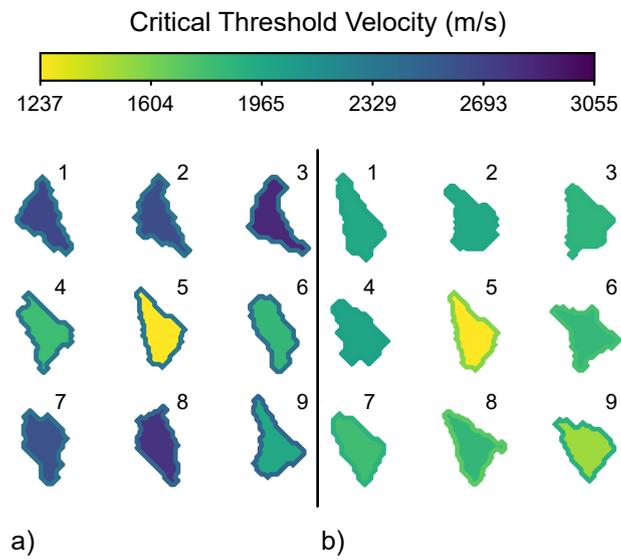


Figure 6.11. Enlarged versions of the regions called out in a) Figure 6.10b and b) Figure 6.9.

### 6.6.6 Web App

It is of the authors' opinion that much more information about the effect of pore shape on shock sensitivity can be gleaned from this work. In an effort to make the research accessible, the complete data sets of pore representations and simulated critical threshold velocities will be published on a web site. In addition to the data, the web site will expose the machine-learned models described in this work via an interactive graphical user interface (GUI). The GUI will allow the users to fetch all pore shapes from the data sets, perturb, alter and create pore contours at will, and get live critical threshold velocities from the models.

## 6.7 Conclusion and Future Work

We have shown that pore geometry significantly influences the sensitivity of shocked HMX. Pore orientation and aspect ratio are strong drivers of shock sensitivity. Generally speaking, elliptical pores with their major axis aligned with the incoming shock are the most sensitive while those elliptical pores with the major axis perpendicular to the incoming shock are the least sensitive. Despite its usefulness, the metrics of aspect ratio and orientation alone do not adequately capture the deviations in shock sensitivity of realistic pore geometries. Machine-learned models trained on the simulated collapse of realistic pore shapes can predict sensitivity with good accuracy but interrogation of these models to understand the exact effect of shape features is difficult. Of the models tested, the convolutional neural network (CNN) acting on a standardized binary image representation of the pore shapes demonstrated the best performance. Furthermore, the shape features extracted by the CNN align with human intuition in that pore shapes of similar appearance are adjacent in feature space and have comparable critical threshold velocities. The data generated by this work and the produced surrogate models will be made publicly available through an interactive web application.

Future work may address opportunities for model improvement; namely, work to quantify the uncertainty of the model predictions will greatly enhance the models' utility.

Additional simulations can be run to augment the existing data set with an increased number of pores sizes so that surrogate models can directly compare and investigate the relative importance of pore size and pore shape. Finally, the collected distribution of critical threshold velocities (relating to pore sensitivity) could be used to inform full-scale stochastic continuum models.

## 7. PREDICTION OF ENERGETIC MATERIAL PROPERTIES FROM ELECTRONIC STRUCTURE USING 3D CONVOLUTIONAL NEURAL NETWORKS

### 7.1 Collaborators

The work in this chapter was conducted in coordination with the Army Research Laboratory (ARL) at Aberdeen, MD. Specifically, the coauthors are Brian Barnes (ARL), Ilias Bilonis (Purdue), and Steven Son (Purdue).

### 7.2 Introduction

In recent years, machine learning (ML) has been increasingly applied to a wide variety of problems in computational chemistry, physics, and materials science [92,100]. Greater accessibility to improved computational resources coupled with large-scale data have contributed to the success of ML in applications such as synthesis planning [101–104], reaction optimization [105,106], drug design and discovery [91,107], materials discovery [108,109], property (or activity) predictions [110], quantum mechanical calculations [111–115], and more [116].

Quantitative structure–activity relationships (QSAR) and quantitative structure–property relationships are quantitative methods that deal with finding a model to relate chemical structural features to their biological *activity* or *physical/chemical properties*, respectively [117]. Both methods are considered subfields of *chemometrics* or *chemoinformatics* and are associated with a large number of publications. Commonly modelled activities/properties are bioactivity (absorption, distribution, metabolism, excretion, and toxicity), aqueous solubility, and melting point [118]. Properties that do not lend themselves well to machine learning are dipole moments, polarizabilities, and vibrational frequencies.

Other, related terms also present in the literature are: quantitative structure—reactivity relationships (QSRR), quantitative structure—chromatography relationships (QSCR) and, quantitative structure—toxicity relationships (QSTR), quantitative structure—electrochemistry relationships (QSER), and quantitative structure—biodegradability relationships (QSBR) [117].

Machine learning, broadly defined, is the extraction of patterns and insight from data [20, 93, 119, 120]. This is commonly referred to as “learning from data.” For supervised ML algorithms the data are composed of “input” and “output” pairs, wherein the goal of the algorithm is to learn the input-output mapping. The result of such a learning algorithm can be expressed as a function  $f(\mathbf{x})$  which receives an input, often a numeric vector which encodes information about the input object, and outputs a prediction  $y$  which may be a single value or a vector [20].

Constructing an effective ML model first requires developing a suitable machine-readable representation of the input object [82, 83, 114]. These representations are commonly referred to in the literature as “descriptors”, “features”, “fingerprints”, or “profiles”. In short, molecular descriptors are formal mathematical representations of a molecule. In other words, the molecular descriptor is the final result of a procedure which transforms chemical information encoded within a symbolic representation of a molecule into a useful number [121].

In their book *Understanding the basics of QSAR for applications in pharmaceutical sciences and risk assessment*, Roy, Kar, and Das dedicate a chapter to molecular descriptors [122]. Some key points that they note are:

A single molecule can be described in many ways and it is possible to compute thousands of numerical descriptors. The selection of relevant descriptors is a well-known problem. . . . A set of descriptors may efficiently encode the chemical information perfectly for the small molecules, but the same set of descriptors may not be able to encode the required features for polymers, protein structures, and inorganic molecules. Thus, not only the calculation but also the selection of suitable descriptors requires a lot of knowledge and

experience. The descriptors may be physicochemical (hydrophobic, steric, or electronic), structural (based on frequency of occurrence of a substructure), topological, electronic (based on molecular orbital calculations), geometric (based on a molecular surface area calculation), or simple indicator parameters (dummy variables). . . . Counts of types of atoms or bonds can be considered as constitutional descriptors that only consider atom and bond labels of the compound. Topological descriptors take into account connectivity and labeled graph theory.

Many molecular structure representations have been used: Coulomb matrix [113,123], fragment descriptors [124], Waber–Cromer pseudopotential radii, Pauling electronegativity, metallic radius, valence electron number (VEN), Clementi’s atomic radii, Pettifor chemical scale [108], eigenvalues of the molecule’s connectivity matrix [125], Simplified Molecular Input Line Entry System (SMILES), Voronoi tessellation [126], to name only a few. As an example, an old fingerprint type is the MACCS keys which consists of 166 bits which can either take on a value of 0 or 1. Each bit corresponds to the presence of a molecular feature, such as the presence of a carbonyl group (key 154) [127]. Another type of fingerprint is the Morgan fingerprint. This fingerprint is *circular* in nature as each atom’s environment and connectivity is analyzed up to a given radius [127]. Thus, circular fingerprints explore the the atom types and connectivity within a molecule while MACCS keys simply note the presence of a predefined feature. Recently, Ward et. al. used 145 physical attributes – categorized as either stoichiometric attributes, elemental property statistics, electronic structure attributes, or ionic compound attributes – as a set of descriptors [128]. Furthermore, the chemical informatics community has developed thousands of attributes for predicting the properties of molecules [129,130] and the broader development of effective descriptors that can index a large number of molecular structures has been the subject of investigation for decades [125].

Given the enormous pool of candidate representations, the ideal descriptor should, intuitively, uniquely define each material in the data and encode the essential physics and chemistry that influence the target property [116,128]. Additionally, an ideal rep-

resentation follows the same symmetry relations as the prediction target and is invariant to permutations (e.g., ordering of the atoms), translation, and rotation [92, 131]. This is especially relevant as numerous investigations have shown that selection of the molecular descriptor can influence model accuracy more than the choice of the ML algorithm [132, 133]. While recent advances have produced models which are covariant by construction, [134–136] conventional ML models, which are widely applied, are limited by the need to properly select molecular features – a process that that requires considerable domain expertise – and to approximate permutation invariances.

Once a suitable set of molecular descriptors has been selected alongside the property of interest most QSPR (or QSAR) problems fall squarely in the realm of supervised machine learning. Due to the vastness of literature in this field [117], it is likely not an exaggeration to state that every known statistical learning model has, at some point, been applied to a chemoinformatics problem. In a review paper, Mitchell [118] cites the use of neural network, random forest, support vector machine, k-nearest neighbor, and naïve Bayes algorithms within the field and still notes the application of at least a dozen more esoteric algorithms.

Deep learning (DL), often considered a subset of ML, is distinguished as a set of specialized algorithms constructed of layers of nonlinear processing units. The interconnected layers form a network capable of learning rich data representations [91, 92]. Each layer in a deep neural network (DNN) transforms its input to increase the selectivity and invariance of the representation [137]. With multiple layers extremely delicate features can be abstracted and the network can universally approximate any arbitrary mapping [138]; with the caveat that large or augmented data sets may be required to learn symmetries in the input data if the input features or network are not explicitly invariant. Thus, DNNs can learn from “raw” data without labor-intensive feature extraction and the resulting model can remain relatively free from bias introduced by the choice of descriptor. For example, Jha et. al. developed a DNN capable of predicting formation enthalpies using only elemental compositions as the input [82]. Their approach outperformed all conventional ML methods; including those with selected physical attributes.

Additionally, Yang. et. al. introduced a graph convolutional model that consistently matched or outperformed models using fixed molecular descriptors [139].

Convolutional neural networks (CNNs), a type of deep learning network, are particularly apt at image processing tasks such as object detection and segmentation. Attempting to hand-craft a robust set of feature detectors designed to evaluate the presence of an object is a difficult task. For example, accommodating slight changes in object orientation, shading, and location require the user to encode additional detection rules. CNNs overcome these issues by learning local patterns (inducing spatial invariance) in the data and further connecting these patterns hierarchically to learn increasingly complex and abstract concepts [93]. Recently, CNNs have been used to successfully explore microstructure-property linkages [94–96, 140]. Additionally, while CNNs are most commonly associated with analysis of 2D image data, they can be applied to image-like arrays or data with local correlations; such as 1D signal and sequence data or 3D video or volumetric image data. Notably: Stepniewska-Dziubinska et. al. created a novel 3D CNN to estimate the binding affinity of ligand-receptor complexes where each complex was represented with 19 features on a 3D grid [141]; Torng et. al. demonstrated that 3D CNNs applied to structure-based protein analysis outperformed models using hand-engineered descriptors [142]; Kajita et. al. increased prediction performance over proposed theoretical descriptors using a 3D CNN on a voxel descriptor encoded from electron density [143]; and Zhou et. al. employed a 3D CNN to learn the relationship between the quasi-local electron density and exchange-correlation potential [144].

Machine learning for prediction of detonation properties and select other properties of interest to energetic materials has recently been investigated by Barnes et. al. [145,146] and Elton et. al. [147]. Their respective works demonstrated ML’s capability to learn chemical information from carefully selected molecular representations – even on small datasets.

Historically, processing a novel energetic material from conception to fielding has required a monumental investment in both time and money; often decades of research and millions of dollars. In the search for the next generation of explosives and propellants, a seemingly infinite number of molecules can be proposed as potential candidates. Of

these candidate molecules relatively few will meet the required performance criteria upon production to render them of practical utility. Due to the cost required to evaluate new materials — even at early research stages — screening methods and heuristics capable of discerning feasible molecules for further study are of great practical importance.

In this work, we develop a convolutional neural network capable of directly parsing the 3D electronic structure of a molecule described by spatial point data for electron charge density and electrostatic potential concatenated into a 4D tensor. We stress that the input does not contain additional information such as atomic nuclei types, positions, etc. The network is jointly trained on over 20,000 molecules that are potentially energetic materials to predict total electronic energy, dipole moment, HOMO-LUMO gap, crystal density, solid phase heat of formation, Chapman-Jouguet (C-J) detonation velocity, C-J pressure, and C-J temperature. In particular, crystal density, heat of formation and C-J detonation velocity, pressure, and temperature are known key indicators of energetic performance.

The target properties used for training are computed using a physics-based workflow, however, it is emphasized that the learned model, nor the model training process, relies on any physics-informed features and the prediction accuracy is a consequence of learning representations on the "raw" data. Previous works have stated that the electron density of an entire molecule is "an impractical amount of information to provide a neural network" and have opted to use sub-regions of the charge density as inputs to deep neural networks [148]. While preparing this manuscript, we became aware of a preprint that seeks to predict electron densities using a DNN where the only input is an approximate electron density [149]. Otherwise, to our knowledge, this work demonstrates the first use of complete 3D electronic structure for machine learning of molecular properties.

## Nomenclature

The following naming nomenclature is used in subsequent equations unless otherwise specified.

$\sigma$	Standard deviation of a sample
$f$	A neuron identification number
$k$	An element of the set of targets
$K$	The set of target outputs; herein: {total electronic energy, dipole moment, HOMO-LUMO gap, crystal density, solid phase heat of formation, C-J detonation velocity, C-J detonation pressure, C-J temperature}
$m$	A numbered orientation, or rotation, of the input tensor
$n$	An single entry, or example, in the data set
$N$	The number of examples, or items, in the data set
$\theta$	The learned model weights
$\mathbf{x}$	Tensor representation of an instance in the data; the electronic structure of a single molecule
$y_{n,k}$	True value of target $k$ for example $n$
$\hat{y}_{n,k}^{(m)}$	Predicted (by the model) value of target $k$ using orientation $m$ for example $n$
$\hat{y}_{n,k}^{(\bar{m})}$	Final prediction of target $k$ for example $n$ (averaged over orientations)

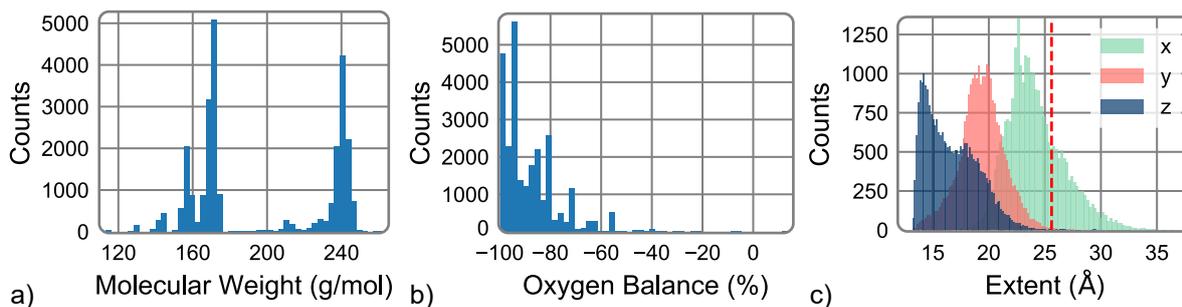


Figure 7.1. Distribution of a) molecular weight, b) oxygen balance, and c) molecular extent for the data set of 26,265 selected molecules. Molecular extent is simply defined as the length of the bounding box in each direction (x, y, z) as output by Gaussian when describing spatial point data. For c, the red dotted line represents the length of each edge in the interpolated bounding box.

## 7.3 Methodology

### 7.3.1 Data Curation

Typically, CNNs have millions of training parameters. In order to mitigate overfitting a large training data set is preferable. In this work, molecules were procured from the GDB database [150–153] and initially screened to consider only those with ‘energetic potential’ (candidate explosives) according to oxygen balance (OB)

$$\text{OB} = -1600 \frac{2N_C + \frac{N_H}{2} - N_O}{\text{MW}} \quad (7.1)$$

where  $N_C$ ,  $N_H$ ,  $N_O$  represent the number of carbon, hydrogen, and oxygen atoms present in the molecule and MW is the molecular weight.

This screening resulted in a reduced data set of 26,265 potentially energetic molecules. The distribution of molecular weight and oxygen balance of this data set are shown in Figures 7.1a and 7.1b, respectively.

Beyond selecting molecules of interest, convolutional neural networks require that each molecule have an associated “input” and “output”. In order to bypass feature selection, we have proposed that the CNN learn a mapping directly from the electronic

structure of a molecule – described by 3D spatial point data for charge density and electrostatic potential stacked into a 4D tensor – to target material properties of interest. For energetic materials, these target properties are total electronic energy, dipole moment, HOMO-LUMO gap, crystal density, solid phase heat of formation, Chapman-Jouguet (C-J) detonation velocity, C-J detonation pressure, and C-J temperature. The electronic structure and target properties are not listed in the GDB database and must be calculated.

Currently, physics-based workflows can accurately estimate many material properties and performance metrics of a candidate molecule. A simple diagram of the physics-based workflow used by CCDC Army Research Lab (ARL) at Aberdeen Proving Ground is shown in Figure 7.2. This framework begins with a sketch of a molecule, which is represented in the computer with its associated SMILES string. The molecule SMILES string is pre-processed with RDKit [154] to generate 1000 random conformers which are each energy-minimized using the MMFF94 force field. The atomic coordinates of the minimum energy conformer are used as an initial geometry and these coordinates are randomly perturbed to break any potential symmetry. Using this atomic coordinate representation, the electron density surrounding the molecule and the corresponding electrostatic potential were calculated using Density Function theory (DFT) at the B3LYP/6-31G\*\* level as implemented in the Gaussian 16 package. [155]. Gaussian outputs many molecular properties; those used in this work are the molecule’s total electronic energy, HOMO-LUMO gap, dipole moment, and 3D point grids of electron density and electrostatic potential. The 3D electron density and electrostatic potential point grids may be further leveraged to predict the material crystal density and solid phase heat of formation via analysis as developed by Byrd, Rice, and Hare [156–158] (diagrammed here as “ARL Toolkit”). With estimated values for the crystal density and solid phase heat of formation, the workflow can be further extended to estimate Chapman-Jouguet (C-J) detonation velocity, C-J detonation pressure, and C-J temperature using the thermochemical code Cheetah v. 8.0 [159].

For this work, the physics-based workflow was applied to the 26,265 energetic molecules to generate a database of electrostatic potentials, charge densities, and target material

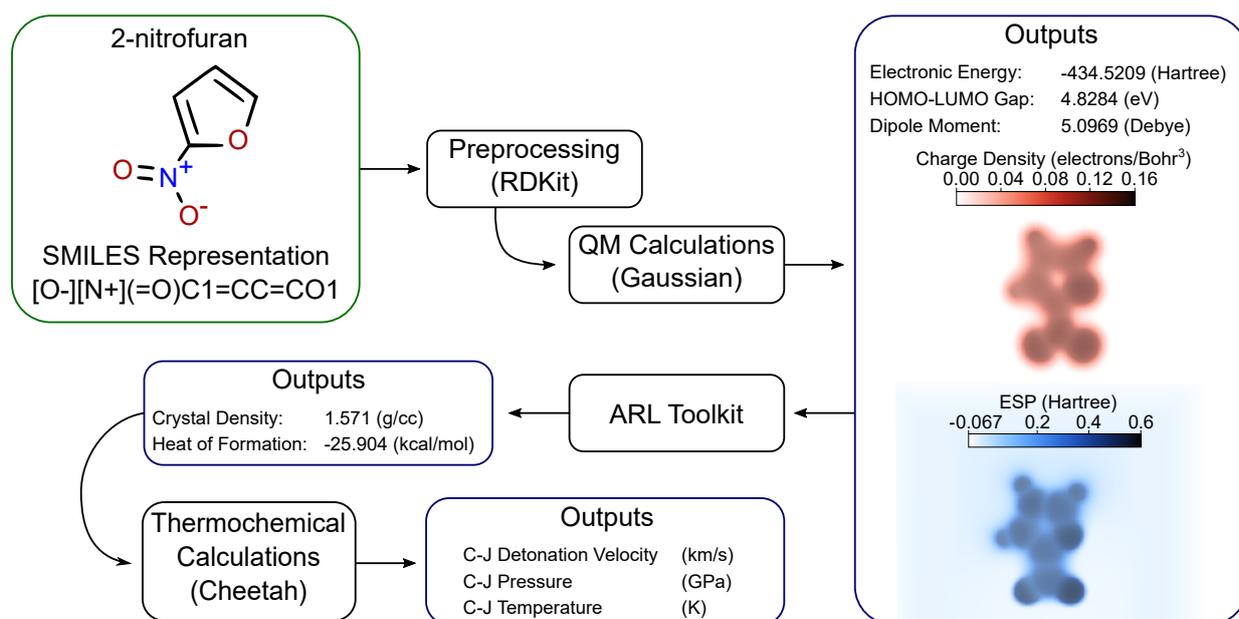


Figure 7.2. Physics-based workflow. A sample molecule 2-nitrofurane is originally represented by a SMILES string. The string is processed by RDKit to produce atomic coordinates for a conformation near the molecule's minimum total electronic energy. Gaussian 16 is called and outputs, among other things, estimates of the molecule's total electronic energy, HOMO-LUMO gap, dipole moment, and 3D point grids of electron density and electrostatic potential. The ARL Toolkit further manipulates the electronic density and electrostatic potential point grids to estimate the molecule's crystal density and solid phase heat of formation. With these properties, the thermochemical code Cheetah v. 8.0 is used to estimate Chapman-Jouguet (C-J) detonation velocity, C-J detonation pressure, and C-J temperature.

properties. Each molecule was processed using 40 cores (Intel Xeon E5-2698v4 Broadwell) requiring 20 to 180 minutes of wall time. A reduced version of this database has been provided as Supporting Information.

### 7.3.2 Input Standardization

Although the major goal of this work is to make predictions on the "raw" input data, some pre-processing is required. Notably, the 3D spatial point grids produced

by Gaussian, also known as cube files, are not constant in either number of points or spacing between points for every molecule (at least under default settings). Additionally, Gaussian orients the molecule such that the direction requiring the most points to resolve its structure is defined as the x-axis. Similarly, the y-axis is chosen so that it has at least as many points as the z-axis. Thus, the resulting 3D spatial point grids are cuboids with side lengths and point spacing varying between molecules. The side lengths provide a naive measure of molecular "extent" and the distributions of the Gaussian grid side lengths are shown in Figure 7.1c. While there are algorithmic adaptations to make use of dissimilar input sizes the simplest approach is to standardize the input shape.

We elected a standard grid of 64 by 64 by 64 points with a point spacing of 0.4 Å; creating a cube with side lengths of 25.6 Å. These parameters were chosen in order to adequately bound *most* of the molecules considered while maintaining small point spacing (higher structure resolution) and a minimal total number of points (to reduce the size of the model input). For each molecule, standard cubes were centered on the rectangular electron density and electrostatic potential point grids and the values at each standard point were calculated using linear interpolation. It should be noted that an artifact of this choice is that some molecules were "clipped" in the x direction as demonstrated by the red dotted line in Figure 7.1c.

It is a common practice to scale all the inputs to the minimum and maximum values found within the data set. However, for the case at hand, when a 3D spatial grid point is near the center of an atom, the electron density and electrostatic potential can acquire very high values relative to the rest of the data points. Thus, a simple min-max or z-score standardization would effectively eliminate fine differences electronic structure. To resolve this issue we bounded the maximum value that the electron density and electrostatic potential 3D spatial point grids can take to the 99<sup>th</sup> percentile of the data set. These limits were 0.16 and 0.6 for the electron density and electrostatic potential, respectively.

An example of this process applied to the electron density of 2-nitrofurans is provided in Figure 7.3. For both subfigures 3a and 3b, the left hand side is the original representation of the electron charge density as produced by Gaussian and the right hand side is

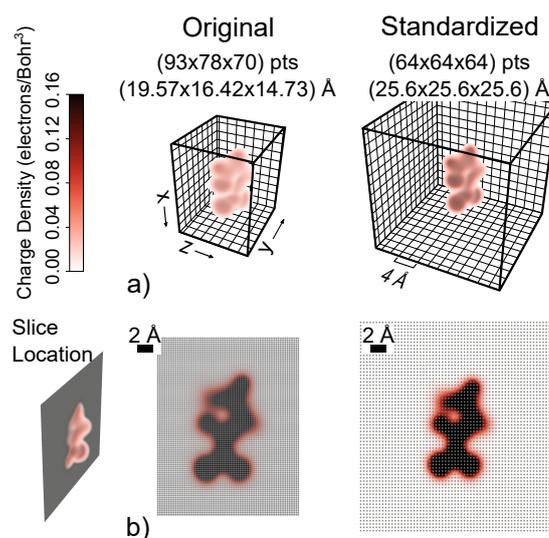


Figure 7.3. Vector graphic of the electron charge density 3D spatial point grid of an example molecule, 2-nitrofurán, before and after standardization. a) The original (output by Gaussian) and standardized grid shapes. b) A 2D slice showing the location of the grid points (grey dots) before and after interpolation. Note that the colorbar applies to both subfigures.

the representation after processing. Figure 7.3a shows the 3D spatial grid extent while Figure 7.3b displays a 2D slice and the grid point locations (grey dots) before and after interpolation.

The resulting standardized input consists of two identically shaped (64x64x64) 3D spatial point grids, one for electron charge density and one for the electrostatic potential, stacked for form a 4D tensor of shape (64x64x64x2) which can be viewed as a 3D image with two channels.

### 7.3.3 Approximating Rotational Invariance

The standardized electrostatic potential and charge density arrays used as inputs to the model are cubic. Consequently, a 90 degree rotation about an axis also produces a valid input array. For cubes, via a series of 90 degree rotations, 24 unique orientations can be produced. Because the original orientation of these input arrays is arbitrarily

determined by Gaussian, the data set was augmented to include all 24 unique orientations of the electrostatic potential and charge density arrays. Including these rotations aids the model in learning approximately rotationally invariant features [141] and increases model generalizability. It should be noted that typical usage of the term “data augmentation” among DL literature implies that a different (usually randomly perturbed) representation of each input example is fed to the model *once* per epoch, however, here we distinguish that all 24 orientations of each molecule were included during each epoch – effectively increasing the total size of the data set to 625,560 examples.

### 7.3.4 Model Architecture

Many choices are made when designing a deep network architecture; such as: number of layers, type of activation functions used, size of convolutional kernels, type and magnitude of regularization. To select the optimal model, many architectures must be trained, evaluated, and compared. We tried three architectures: an Inception-like [160] network, a ResNet-like [161] network, and a "standard" network. With the initial hyperparameters the Inception and ResNet networks failed to reduce the loss function and were not further pursued. The standard network successfully minimized the loss function. This network was composed of sequential 3x3x3 convolutional kernels and 2x2x2 max pooling layers followed by multiple fully connected layers. Additionally, 8 fully connected layer blocks (referred to as FC heads) are appended, each one reducing down to a single neuron, to make a prediction on each target quantity producing a multi-target network. With the exception of the final output neurons (one for each target variable) both the convolutional kernels and fully connected layers were followed by the rectified linear unit (ReLU) activation. The ReLU activation function was selected to speed up the training process. This architecture, show in Figure 7.4, shares a convolutional base which serves to greatly reduce the amount of inputs seen by the final eight fully connected layer blocks. Additionally, the joint learning provides a means for the network to learn a richer set of representations.

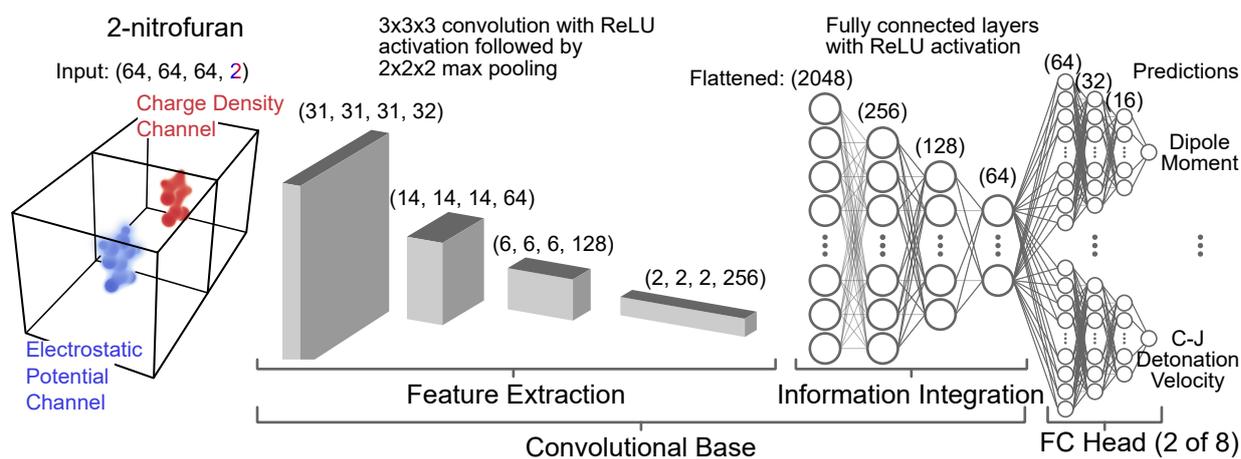


Figure 7.4. Selected model architecture. An example molecule, 2-nitrofurane, is represented by a standardized input consisting of two stacked 3D spatial point grids each of shape  $(64, 64, 64)$  – one for the electron charge density and one for the electrostatic potential – to form a 4D tensor of shape  $(64, 64, 64, 2)$ . A convolutional kernel of shape  $(3, 3, 3)$  is applied to the input followed by ReLU activation function and a  $(2, 2, 2)$  max pooling operation. This process is repeated increasing the number of “filters” while decreasing the data shape. The result is flattened and passed to a series of fully connected layers, each followed by ReLU activation. Finally, an additional 8 sets of fully connected layers (again followed by ReLU activations) are used to reduce the output down to 8 neurons (with linear activation) – one for each target quantity.

### 7.3.5 Training

For training and evaluation, the set of molecules (26,265) was scaffold split using the deepchem package [162] into a training set composed of 21,012 molecules, a validation set with 2,626 molecules, and a test set containing 2,627 molecules (an 80/10/10 split). Scaffold splitting partitions the molecules based on their two-dimensional structural frameworks – intentionally forming dissimilar train, validation, and test sets – creating a greater challenge for the learning algorithm than a random split. [163, 164] The success of the splitting procedure was verified manually by inspecting the two-dimensional structures of the molecules in their respective sets. Examples of molecules in the training, validation, and test sets are shown in Figure 7.5. While only a small subset of the 26,265 molecules can be visualized here, Figure 7.5 demonstrates that each group contains structurally similar molecules while remaining distinct from the other groups.

The aforementioned network was modelled in Keras [98]. Beyond trying three different architectures no hyperparameter optimization was conducted. The Adam optimizer was used with hyperparameters: learning rate = 0.001,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , AMSGrad = False, and a batch size of 64 examples per minibatch.

The loss function  $L$  to be minimized was defined as the weighted summation of squared errors summed over each target quantity shown by:

$$L(\theta) = \sum_k \frac{1}{\sigma_k^2} \sum_{n=1}^N \sum_{m=1}^{24} \left( y_{n,k} - \hat{y}_{n,k}^{(m)}(\mathbf{x}; \theta) \right)^2. \quad (7.2)$$

For each target variable, the squared errors  $y_{n,k} - \hat{y}_{n,k}^{(m)}(x; \theta)^2$  are summed over all examples  $n$  and molecular orientations  $m$  and weighted by the inverse of the variance of the target quantity  $\frac{1}{\sigma_k^2}$ . The weighting term  $\frac{1}{\sigma_k^2}$  is computed only considering the training set. This weighting helps ensure that the model does not preferentially learn one target property relative to the others. Explicit inclusion of the model weights  $\theta$  and the electronic structure tensor  $\mathbf{x}$  in Equation 7.2 is meant to emphasize that the loss is a function of the model weights and that the model prediction  $\hat{y}_{n,k}^{(m)}$  receives, as input, the tensor representation of the molecule and is parameterized by the model weights. The loss function was monitored throughout training and was evaluated on the test set at

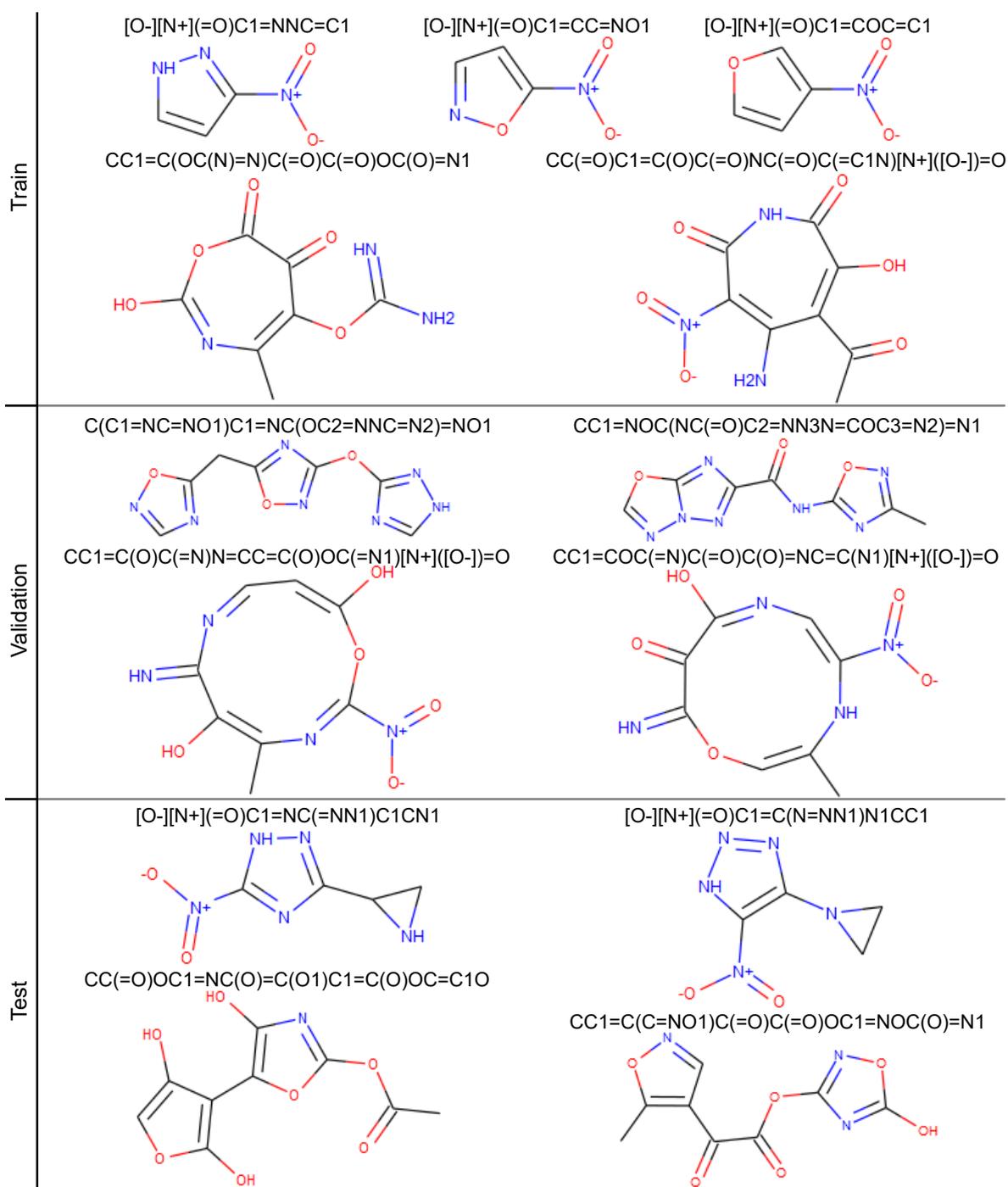


Figure 7.5. Examples of the two-dimensional structural frameworks of molecules present in the train, validation, and test sets after scaffold splitting.

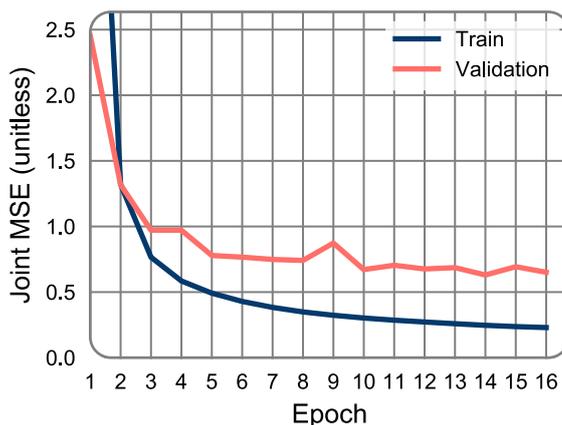


Figure 7.6. Loss history of the model during training. The joint MSE is the loss function described in Eq. 7.2. During each epoch the model encountered  $21,012 \cdot 24 = 504,288$  examples.

the end of each epoch. The model was trained for 16 epochs using a NVIDIA Tesla P100 GPU and the completion of each epoch required roughly two hours. The complete loss history is provided in Figure 7.6. The model corresponding to the 14<sup>th</sup> epoch was selected because it had the lowest loss on the validation set.

## 7.4 Results and discussion

### 7.4.1 Rotational Invariance

If, by viewing multiple orientations of an input, the network learned approximately rotationally invariant features then the convolutional base output activations ( $a$ ) would have the same value for each orientation. These activations correspond to the final dense layer in the “information integration” section of the network. This layer has 64 neurons, which are arbitrarily assigned identifications ( $f$ ) 1-64. The activation value for each neuron and each orientation is plotted for two molecules – one from the training set and one from the validation set – in Figure 7.7.

In this figure, inactive neurons ( $a = 0$ ) are designated with the color white. As seen, not all neurons are activated; indicative that some features learned by the network are not

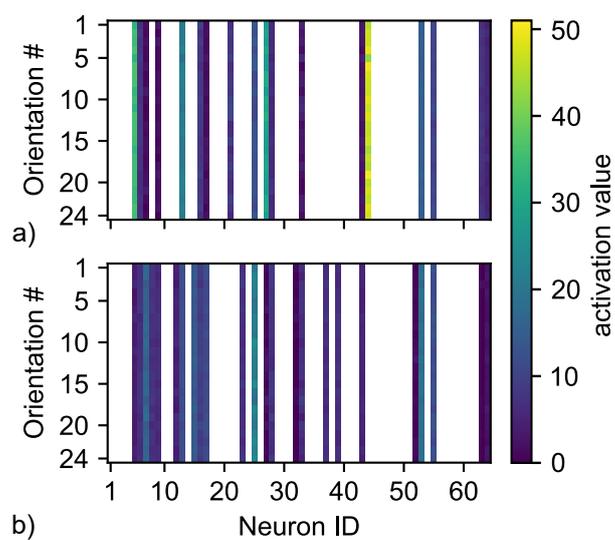


Figure 7.7. Activations ( $a$ ) for each orientation of molecules a) 2-nitrofuran (training set), and b) CC1=COC(=N)C(=O)C(O)=NC=C(N1)[N+]([O-])=O (validation set, shown in Figure 7.5). White indicates an inactive neuron; i.e.  $a = 0$ .

present in the given molecule. Orientation-to-orientation dependence of feature detection is an unwanted attribute. While it is difficult to quantify the degree of orientation independence, we state that Figure 7.7 represents that the activated neurons are very likely to display a similar response across rotations.

#### 7.4.2 Orientation Correlation

Given that the array of extracted features is not perfectly constant between orientations for a given molecular input, we wished to explore the correlation between input orientations. The error between the true value and predicted value for each target quantity, example, and orientation is

$$\epsilon_{n,k}^{(m)} = y_{n,k} - \hat{y}_{n,k}^{(m)}. \quad (7.3)$$

The errors were grouped according to target quantity such that, for a given target, the errors can be considered a matrix with  $N$  rows (one for each molecule) and 24 columns (one for each orientation). For each target quantity the orientation columns were compared pairwise using both the Pearson and Spearman correlation metrics to check for linear and non-linear relationships. An example of the error correlation strengths between orientations is given in Figure 7.8a for the C-J detonation pressure target. The error correlation strength matrices for the remaining target quantities were very similar to that of C-J detonation pressure and are not shown here. However, the maximum error correlation coefficient for each target and the pair of orientations that produced it are listed in Table 7.1. It is noteworthy that the lowest correlation coefficient (Pearson or Spearman) for any target variable was 0.339; that is, there were no negative correlation relationships. A few orientation pairs are repeated in Table 7.1, but since no single pair of orientations is consistently dominant throughout the target quantities these relationships were not further examined.

Strong correlation of errors between orientations would be another indicator of rotational invariance simply because each orientation would make the same prediction and thus the same error. The orientation pairs for each target exhibit moderate strength;

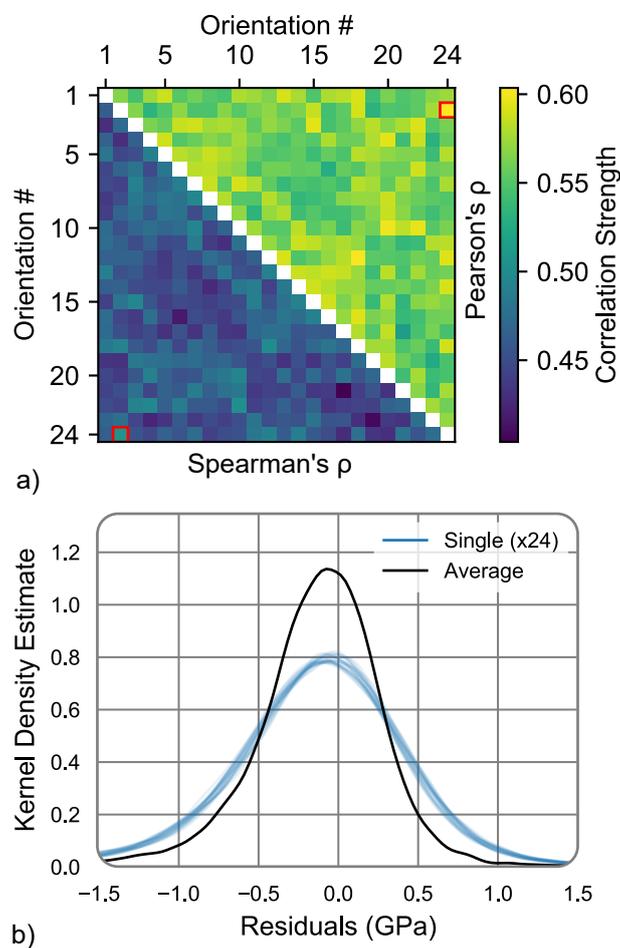


Figure 7.8. Prediction errors of C-J detonation pressure. a) Matrix displaying the correlation strength of prediction error between pairs of input tensor orientations. The upper triangle of the matrix specifies the correlation using the Pearson metric and the lower triangle of the matrix uses the the Spearman correlation metric. The red squares highlight the orientation pairs with the highest correlation using either the Pearson or Spearman method. b) Kernel density estimates of the distributions of C-J detonation pressure prediction error for each input tensor orientation (24) and their average.

Table 7.1. Maximum values for the Pearson (P) and Spearman (S) correlation coefficients for each target quantity and the associated pair of orientations that produced it.

Target	max(P) (orient. pair)	max(S) (orient. pair)
Total Electronic Energy	0.601 (14, 15)	0.537 (15, 14)
HOMO-LUMO Gap	0.643 (5, 8)	0.623 (20, 4)
Dipole Moment	0.558 (6, 7)	0.485 (24, 23)
Crystal Density	0.557 (4, 20)	0.525 (20, 4)
Solid Phase Heat of Formation	0.519 (11, 14)	0.482 (14, 11)
C-J Temperature	0.530 (15, 16)	0.501 (16, 15)
C-J Detonation Pressure	0.604 (2, 24)	0.507 (24, 2)
C-J Detonation Velocity	0.564 (2, 24)	0.521 (22, 10)

while this does not demonstrate perfect rotational invariance it does suggest that prediction error can be decreased overall by taking the average of the predicted values for each orientation. That is, the final prediction for target  $k$  on molecule  $n$  is  $\hat{y}_{n,k}^{(\bar{m})} = \frac{1}{24} \sum_{m=1}^{24} \hat{y}_{n,k}^{(m)}$ . This reduction in error is demonstrated in Figure 7.8b; again using the C-J detonation pressure target as an example. In this figure the Gaussian kernel density estimates (KDE) of the prediction errors for each orientation are plotted alongside the KDE of the error after averaging the predictions. For all orientations – and the average – the error distributions were assumed to be normal. Under this assumption the optimal KDE bandwidth (bw) is that which minimizes the mean integrated square error and is provided as

$$\text{bw} = \left( \frac{4\hat{\sigma}^5}{3N} \right)^{\frac{1}{5}}. \quad (7.4)$$

For a given target quantity, the KDE bandwidth was held constant for all orientations (and the average) by fixing  $\hat{\sigma}$  to the average standard deviation of the errors at each orientation – described mathematically as  $\hat{\sigma} = \frac{1}{24} \sum_{m=1}^{24} \sigma_{\epsilon}^{(m)}$ . Thus, differences in error distributions in Figure 7.8b are not due to varying bandwidths in the KDEs. KDE plots for the errors of the remaining targets were all similar in form to the C-J detonation pressure – including a reduction in error when using the average orientation prediction – and are not displayed here for brevity.

### 7.4.3 Prediction Accuracy

Using the average orientation prediction  $\hat{y}_{n,k}^{(\bar{m})}$  as the final model prediction on target  $t$  for molecule  $n$ , the model prediction is plotted against the true value for each target in Figures 7.9-7.12. Because the train, validation, and test sets consist of 21,012, 2,626, and 2,627 molecules, respectively, these are easily cluttered. To promote visualization the plot domain is binned hexagonally and the quantity of points contained within a bin is represented by color. Additionally, measures of prediction accuracy, namely, the root mean squared error (RMSE) and  $R^2$ , are provided with each plot. Of the model predictions, the HOMO-LUMO gap target has the lowest performance with a test set

$R^2$  of 0.838. Otherwise the lowest test set  $R^2$  reported is 0.943 on the crystal density target indicating excellent model prediction capabilities.

It is worth restating that the various target properties were calculated at different stages in the physics-based model workflow (Figure 7.2). The molecule’s total electronic energy, HOMO-LUMO gap, and dipole moment were first calculated via DFT using Gaussian. Afterwards, the crystal density and solid phase heat of formation were calculated using the ARL toolkit. Finally, the Chapman-Jouguet (C-J) detonation pressure, velocity, and temperature were computed using Cheetah. While targets produced in this manner are themselves estimates and should be further validated by comparison to experimental data where available, we note that the machine-learned model performance does not deteriorate when making predictions on target quantities calculated at later stages of the physics-based workflow. Greater interrogation of the machine-learned model is required to distinguish if the model is learning direct relationships between a molecule’s electronic structure and late-stage properties such as the C-J detonation velocity or if the model, due to the joint training procedure, is capitalizing on correlations between these late-stage properties and other target quantities. In either case, a major advantage of the machine-learned model is its ability to provide simultaneous predictions on a wide array of targets which otherwise would require the coupling of multiple physics-based theories and codes.

#### 7.4.4 Comparison to ECFP4 and Random Forest

Extended-connectivity fingerprints (ECFPs) are a representation of molecular structure. [165] In particular, ECFP4 perceives elements up to 4 bonds around each atom. Using deepchem [162], we produced ECFP4 vectors of length 1024 for each molecule in the training, validation, and test sets. These descriptors served as the input for a joint random forest (RF) model – implemented with scikit-learn [27] – with the same eight target outputs as previously used in the CNN. Although scikit-learn allows for custom weighting of loss and scoring functions when evaluating models, this feature is not currently accessible during model training. As such, the targets were first normalized by

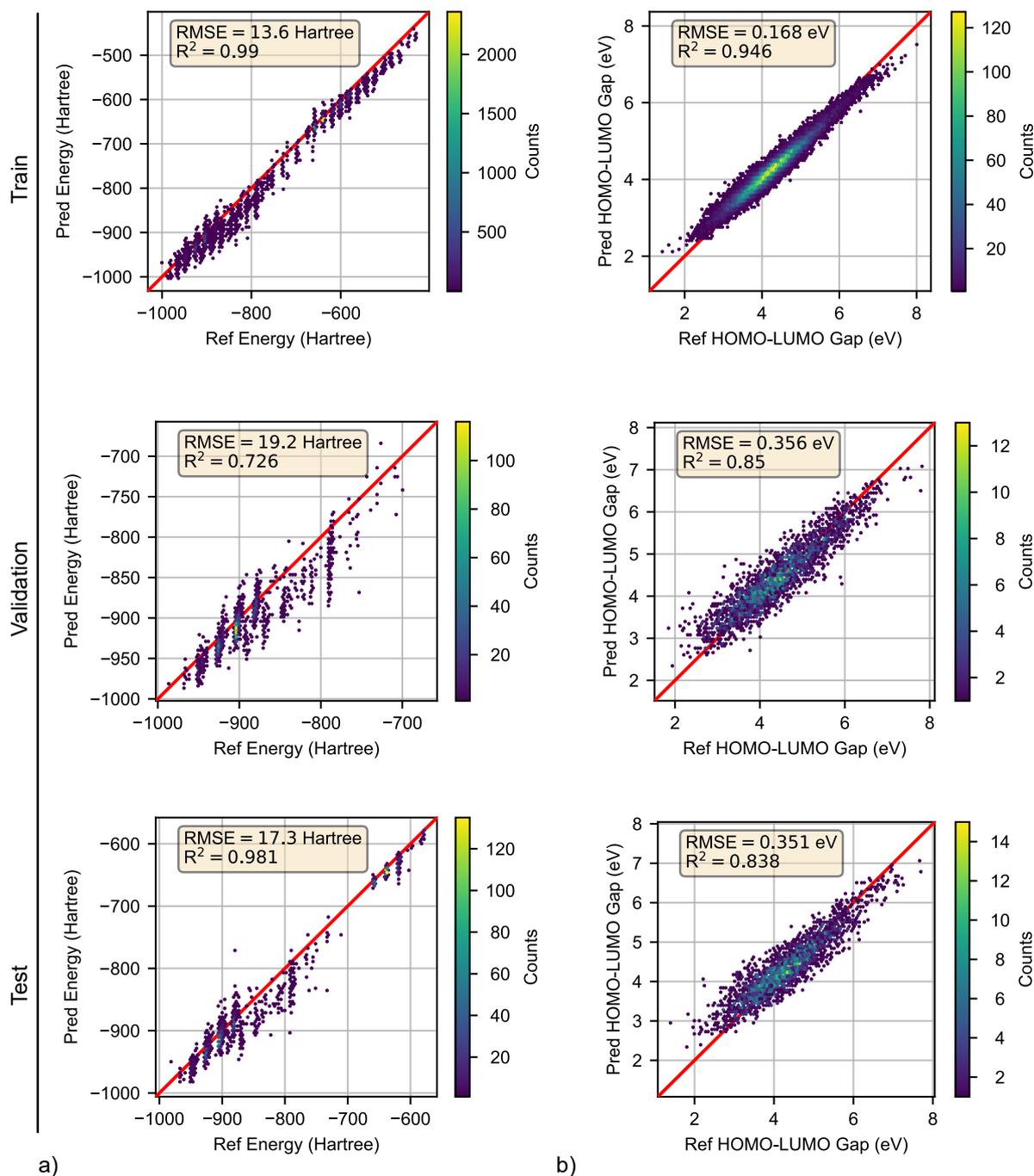


Figure 7.9. Predicted value (CNN) vs. actual value (physics-based workflow estimate). For both subfigures the top plot references the training set, the middle plot consists of the validation set, and the bottom plot contains the test set. The red line indicates an exact prediction (predicted equals actual). Each subfigure corresponds to a different target output: a) total electronic energy, b) HOMO-LUMO gap.

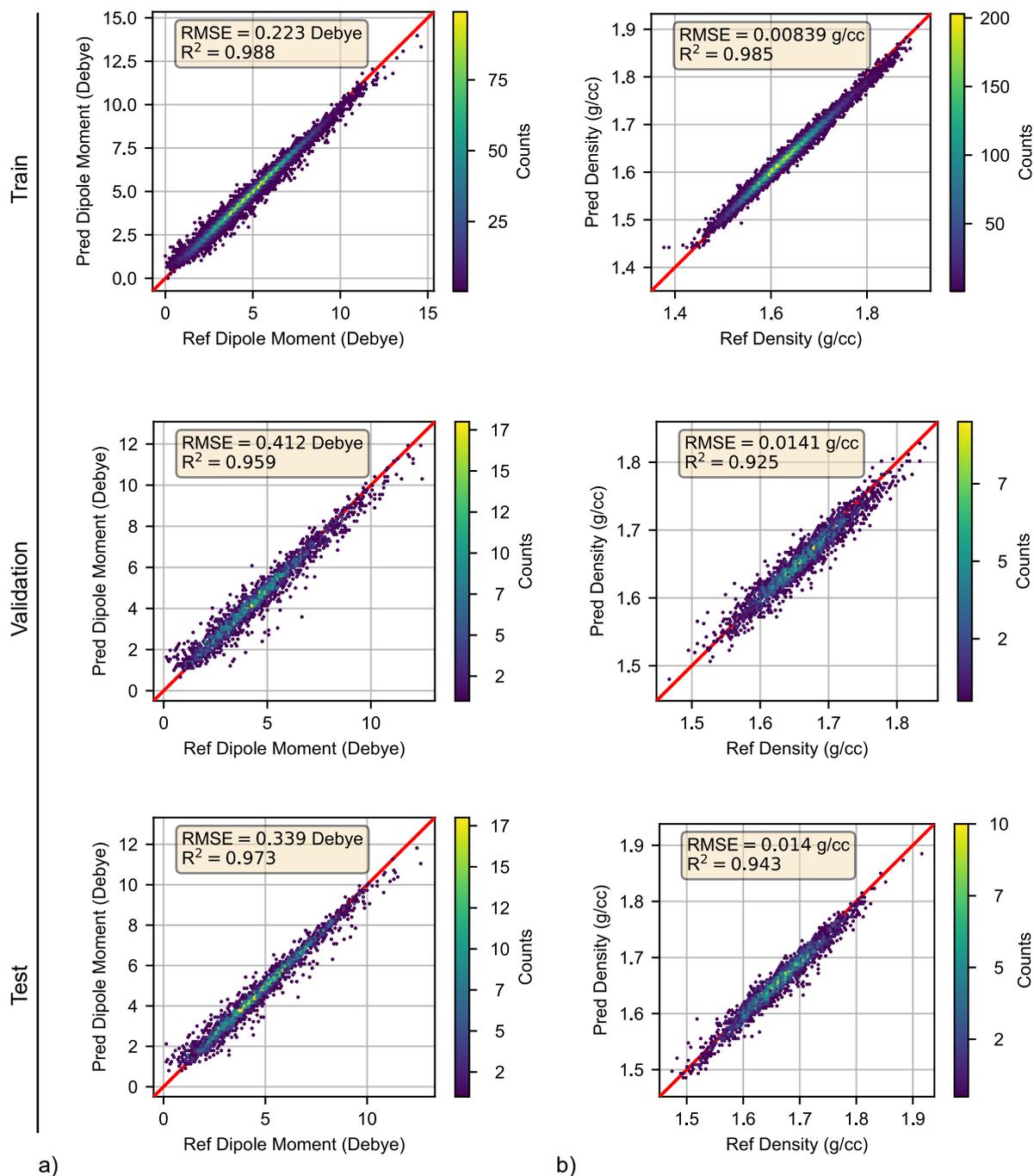


Figure 7.10. Predicted value (CNN) vs. actual value (physics-based workflow estimate). For both subfigures the top plot references the training set, the middle plot consists of the validation set, and the bottom plot contains the test set. The red line indicates an exact prediction (predicted equals actual). Each subfigure corresponds to a different target output: a) dipole moment, b) crystal density.

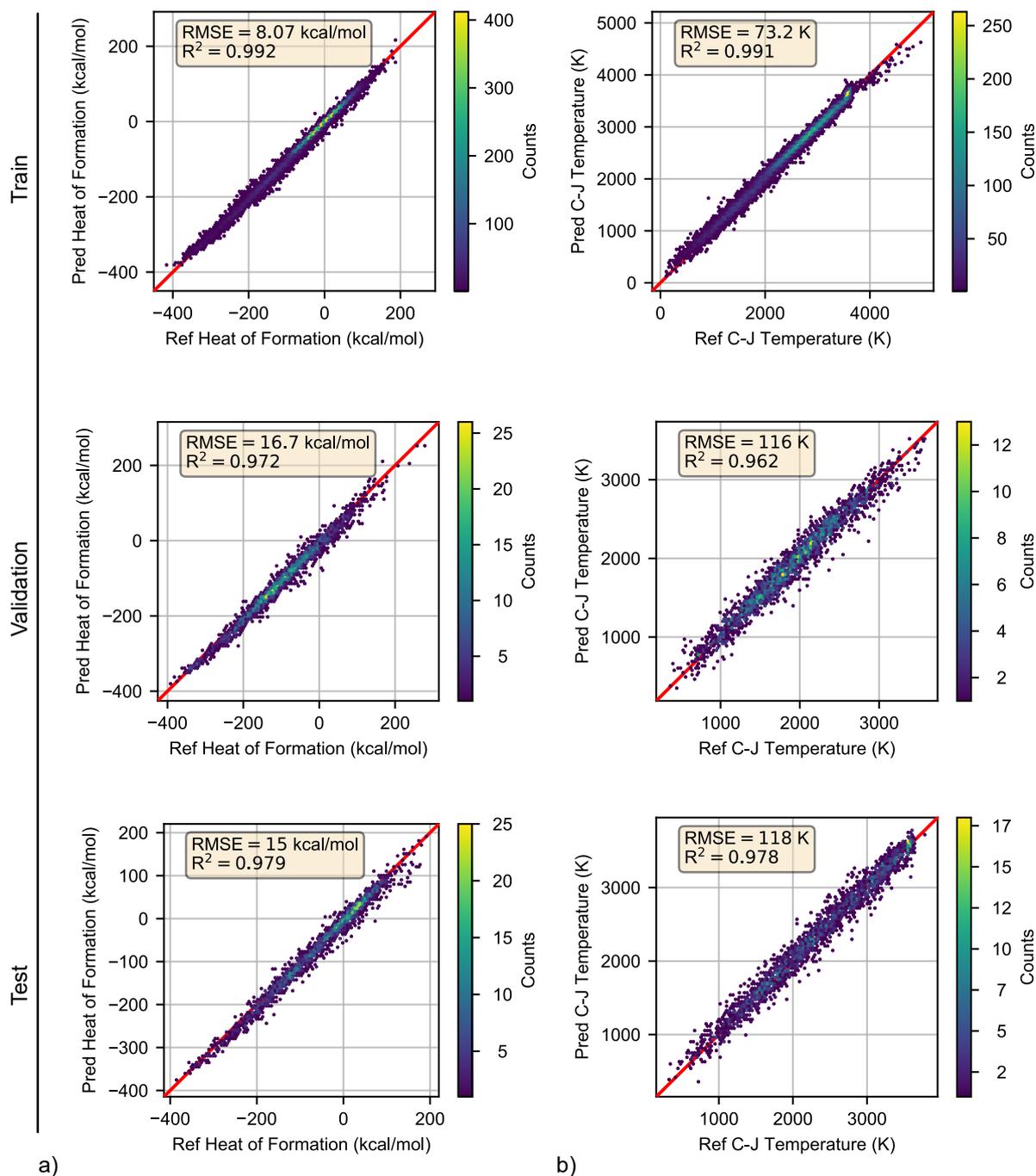


Figure 7.11. Predicted value (CNN) vs. actual value (physics-based workflow estimate). For both subfigures the top plot references the training set, the middle plot consists of the validation set, and the bottom plot contains the test set. The red line indicates an exact prediction (predicted equals actual). Each subfigure corresponds to a different target output: a) solid phase heat of formation, b) Chapman-Jouguet detonation temperature.

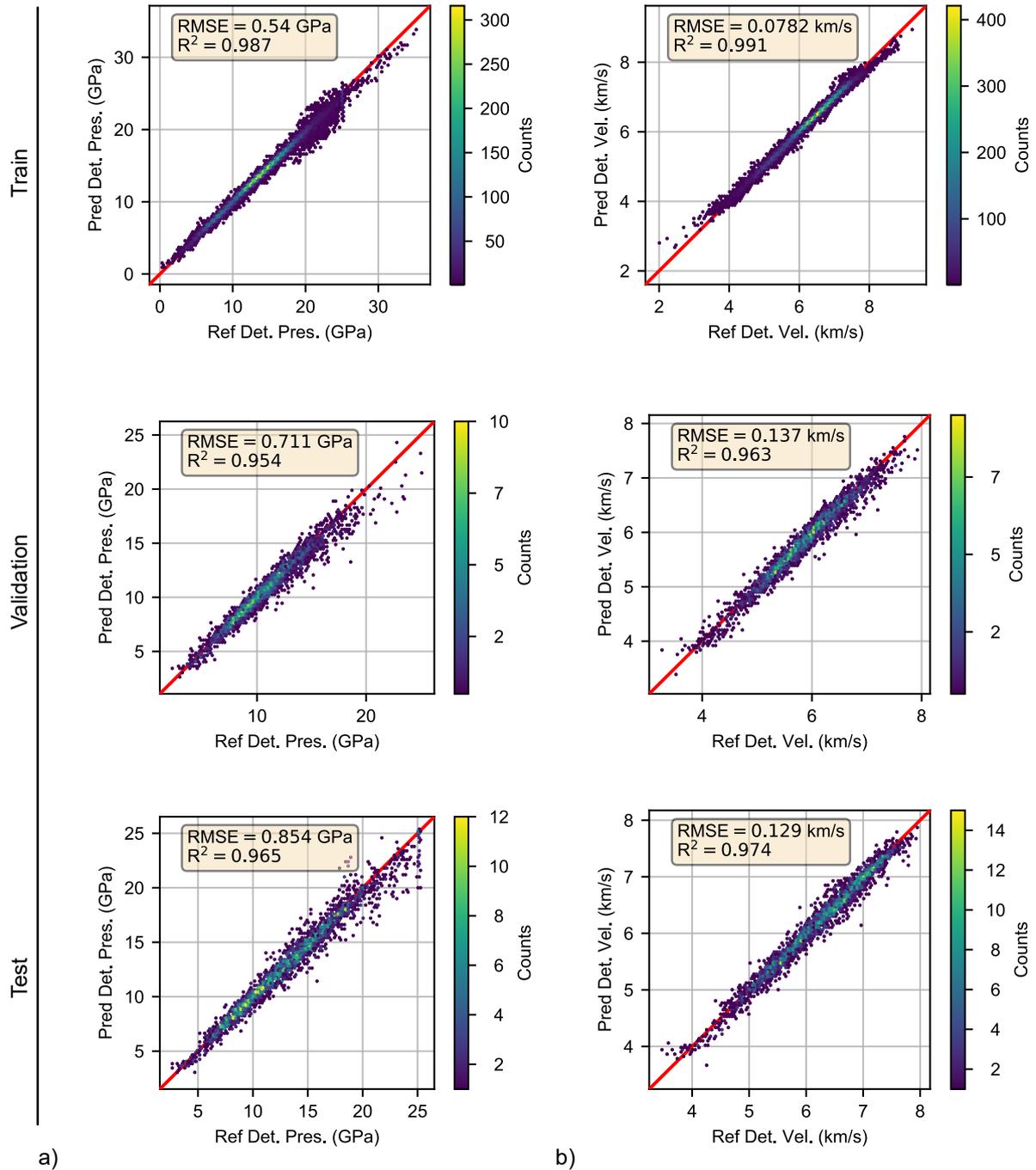


Figure 7.12. Predicted value (CNN) vs. actual value (physics-based workflow estimate). For both subfigures the top plot references the training set, the middle plot consists of the validation set, and the bottom plot contains the test set. The red line indicates an exact prediction (predicted equals actual). Each subfigure corresponds to a different target output: a) Chapman-Jouguet detonation pressure, b) Chapman-Jouguet detonation velocity.

subtracting their respective mean and dividing by their standard deviation. This process ensures that no target is given preferential treatment and is analogous to the use of the weighting factor in Equation 7.2.

The random forest hyperparameters were optimized using hyperopt [99] to explore and exploit the hyperparameter space while evaluating the mean squared error of the validation set on the fitted model over 1000 trials. The initial probability distribution over each hyperparameter that was allowed to vary is given in Table 7.2.

The performance of the tuned, jointly trained random forest model on each task is measured with  $R^2$  and the mean absolute error (MAE). These results are tabulated in Tables 7.3 and 7.4, respectively. While the joint random forest model does fairly well on the training set, its performance is severely degraded on the test set. For example, the crystal density training set  $R^2$  is 0.851 whereas the test set  $R^2$  is 0.326. Interestingly, the total electronic energy validation  $R^2$  and MAE are -2.567 and 39.031 Hartree, respectively. This indicates that model struggled to translate learned features from the training set to the validation set on this target. Because the joint RF is attempting to fit all targets simultaneously we hypothesized that removing the total electronic energy target from the model would improve performance. To this end, using the same training and hyperparameter optimization process, we developed another jointly trained random forest without the total electronic energy target. Additionally, we produced eight independent random forests – one for each target. The only difference being that the independent random forests’ hyperparameter optimization was conducted with 300 trials. The  $R^2$  and MAE scores for these models, alongside the CNN, are also listed in Tables 7.3 and 7.4.

The joint RF without the total electronic energy target and the independent random forests achieved greater performance than the original joint RF model. However, the previously developed 3D CNN outperformed all the ECFP4 RF models on every task. In fact, the MAE for the CNN is typically half (or less) than the MAE for the RF models.

Table 7.2. The random forest hyperparameters that were allowed to vary and their prior distributions. Brackets indicate a discrete distribution with support [lower bound, upper bound, step size] and curly braces indicate a choice between values.

Hyperparameter	Initial Distribution
max_depth	uniform [2, 100, 1]
n_estimators	uniform [5, 200, 5]
min_samples_leaf	uniform [1, 10, 1]
max_features	uniform {None, log <sub>2</sub> , sqrt}

Table 7.3.  $R^2$  scores for all eight prediction tasks broken down by model and set designation (train, validation, or test).

		Total	HOMO-					C-J	C-J
		Electronic	LUMO	Dipole	Crystal	Heat of	C-J	Detonation	Detonation
		Energy	Gap	Moment	Density	Formation	Temperature	Pressure	Velocity
Joint RF	Train	0.951	0.793	0.658	0.851	0.947	0.951	0.925	0.935
	Validation	-2.567	0.418	0.065	0.222	0.713	0.670	0.611	0.644
	Test	0.786	0.402	0.137	0.326	0.763	0.814	0.704	0.739
Joint RF (no energy)	Train		0.867	0.790	0.902	0.964	0.967	0.951	0.957
	Validation		0.422	0.071	0.213	0.718	0.673	0.612	0.646
	Test		0.407	0.148	0.318	0.767	0.818	0.706	0.742
Independent RFs	Train	0.928	0.879	0.768	0.905	0.973	0.973	0.941	0.950
	Validation	-2.472	0.460	0.100	0.322	0.751	0.713	0.627	0.678
	Test	0.784	0.473	0.243	0.426	0.797	0.840	0.718	0.757
CNN	Train	0.990	0.946	0.988	0.985	0.992	0.991	0.987	0.991
	Validation	0.726	0.850	0.959	0.925	0.972	0.962	0.954	0.963
	Test	0.981	0.838	0.973	0.943	0.979	0.978	0.965	0.974

Table 7.4. Mean absolute errors for all eight prediction tasks broken down by model and set designation (train, validation, or test).

		Total	HOMO-					C-J	C-J
		Electronic	LUMO	Dipole	Crystal	Heat of	C-J	Detonation	Detonation
		Energy	Gap	Moment	Density	Formation	Temperature	Pressure	Velocity
		(Hartree)	(eV)	(Debye)	(g/cc)	(kcal/mol)	(K)	(GPa)	(km/s)
Joint RF	Train	17.885	0.252	0.923	0.020	14.957	129.681	0.933	0.154
	Validation	39.031	0.560	1.562	0.035	40.976	260.154	1.536	0.327
	Test	35.629	0.536	1.523	0.038	37.686	259.756	1.859	0.313
Joint RF (no energy)	Train		0.200	0.716	0.016	12.221	104.948	0.744	0.124
	Validation		0.558	1.554	0.036	40.403	258.598	1.529	0.326
	Test		0.532	1.510	0.039	37.331	258.313	1.854	0.313
Independent RFs	Train	20.900	0.182	0.739	0.016	9.585	90.755	0.796	0.120
	Validation	38.065	0.536	1.529	0.033	38.120	244.547	1.514	0.315
	Test	35.628	0.493	1.409	0.035	34.187	241.329	1.802	0.303
CNN	Train	10.260	0.131	0.155	0.006	5.595	57.022	0.351	0.057
	Validation	14.389	0.277	0.292	0.011	12.918	89.516	0.499	0.103
	Test	12.702	0.273	0.232	0.011	11.250	90.029	0.584	0.096

Interestingly, the greatest difference between validation and test set performance measures occurred on the total electronic energy task and was observed in both the RF and CNN models. To explore this difference in greater detail, we plot the histogram of total electronic energy values grouped by set designation in Figure 7.13a. As seen, the validation set molecules all have total electronic energies less than -700 Hartree. While this is unique, there are plenty of instances within the training set with similar values. Using the naive measure of molecular "extent" (as output from Gaussian) to roughly describe a molecule's size, the distribution of extent along the x-axis is given for the training, validation, and test sets in Figure 7.13b. This figure demonstrates that the molecules in the validation set are generally bigger than those in the training and test sets. Recall that, for the CNN, the input standardization process truncates all electronic structure information beyond 25.6 Å. Consequently, inspection of Figure 7.13b reveals that 47.8% of the molecules in the validation set have truncated electronic structures. The percentage of truncated molecules in the training and test sets are 21.5% and 34.5%, respectively. Given that the validation set molecules are generally larger in size, *and* nearly half of these molecules are missing components of their electronic structures, it is impressive that the CNN was reasonably accurate when predicting the total electronic energy ( $R^2=0.726$ , MAE=14.389 Hartree (1.93%)).

#### 7.4.5 y-Randomization

In addition to estimating the generalization error by evaluating the model performance on the test set, we compared the performance of the final CNN model to a "straw" model developed with y-randomization. A straw model is a similar variant of the working model but it excludes the working hypothesis by design [166]. Logical reasoning dictates that if the proposed model cannot significantly outperform straw models then it should not be accepted.

For y-randomization, the target labels of the training set are randomly shuffled before the model is trained [167]. This procedure effectively ablates any chemical information available to the model and, by definition, cannot generalize to new or unseen

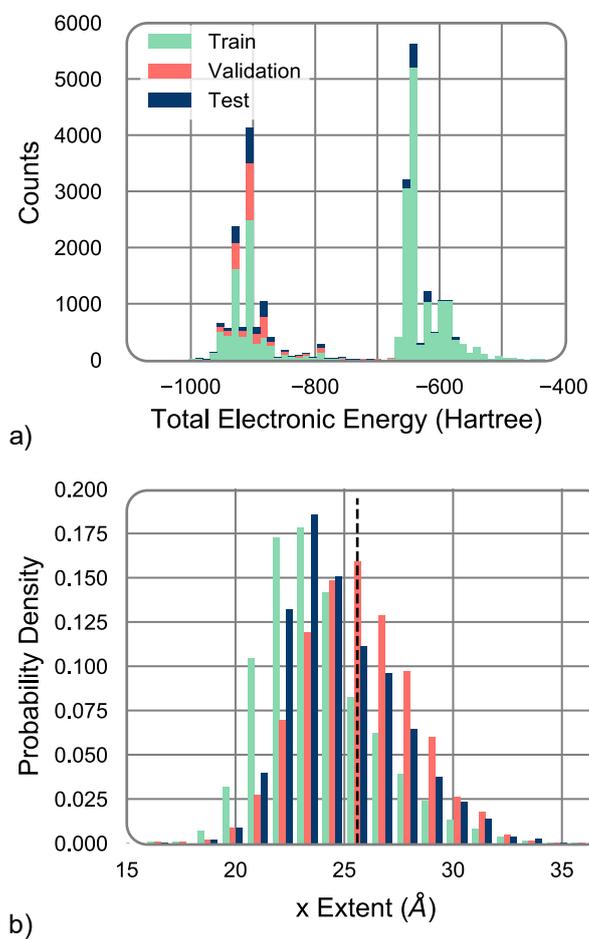


Figure 7.13. a) Stacked histograms of total electronic energy values present in the training, validation, and test sets. b) Probability distributions of extent along the x-axis. The dashed black line represents the outer limit of the input standardization bounding box.

instances [168]. A model trained under these conditions is a straw model wherein any predictive performance it attains on either the training, validation, or test sets can only be attributed to chance correlations between the inputs and targets [167].

Ideally, multiple y-randomized models are produced in order to estimate the expectation and variance of the straw model performance. A statistically significant increase in performance of the proposed model compared to that of the straw models serves as strong evidence that the proposed model learned actual, physically-relevant, information.

Due to the computational costs of training a large model, we only developed one y-randomized variant. Upon completion of training (following the same procedures as the proposed model), inspection of the model revealed that the model learned to simply predict the mean value of the training set for each target. That is, the randomized model proved incapable of learning even chance correlations or noise between the electronic structure of the molecule and the target properties and "settled" for predicting the target mean. By definition, the straw model  $R^2$  scores for both the training and test sets are approximately zero for all targets and are not shown here. Likewise, the root mean squared errors are also omitted for concision as these are simply the standard deviations of the target outputs.

Although only one y-randomized model variant was trained, we submit that the stark contrast in model performance between the straw model and original model provides sufficient validation that the proposed model has extracted physical information from the molecular structures.

#### **7.4.6 Effect of Extent Cutoff**

As noted in Methodology, due to the size of the standardized input cube the electron charge density and electrostatic potential 3D spatial point grids of some molecules will be clipped in the "x" direction (longest principal direction as dictated by Gaussian). In order to compare the effect of this clipping on the target predictions while considering all

target quantities simultaneously a joint scaled error was computed by taking the mean of the weighted absolute errors over the target set

$$\frac{1}{|K|} \sum_{k \in K} \left( \frac{|\epsilon_{n,k}|}{\sigma_k} \right). \quad (7.5)$$

Here the error is taken as the difference between the actual target value and the final model prediction (averaged over orientations);  $\epsilon_{n,k} = y_{n,k} - \hat{y}_{n,k}^{(\bar{m})}$ . The quantity described by Equation 7.5 is herein referred to as the mean scaled absolute error (MSAE) and serves as a measurement of the effective error across all target quantities.

Note that in Equation 7.5, the absolute value of the error is taken so that only the magnitude, and not the direction, of the error is considered. Additionally, the absolute error is scaled by the inverse standard deviation of the true values for the target variable  $k$  ( $\sigma_k$ ) considering the complete data set. Including the train, validation, and test sets in the calculation of  $\sigma_k$  keeps the scaling factor constant between these sets.

The MSAE for each molecule is plotted against the original (pre-standardization) length of the molecule’s 3D spatial point grids in the x direction; shown in Figure 7.14.

As seen, the prediction error is generally higher for those molecules with clipped information and there is some correlation between molecular extent in the x direction and absolute error.

## 7.5 Conclusion and Future Work

We have shown that a 3D convolutional neural network can effectively parse the electron charge density and electrostatic potential of a molecule to make high accuracy predictions on material properties without the need to hand-craft molecular descriptors. The 3D CNN model, without any hyperparameter tuning, outperforms tuned random forest models using ECFP4 descriptors. This model attains excellent generalization error even when making predictions on structurally dissimilar molecules as observed with scaffold-based splitting.

Future work may address opportunities for model improvement; namely: increasing the size of the bounding box so that no electronic structure information is clipped; ap-

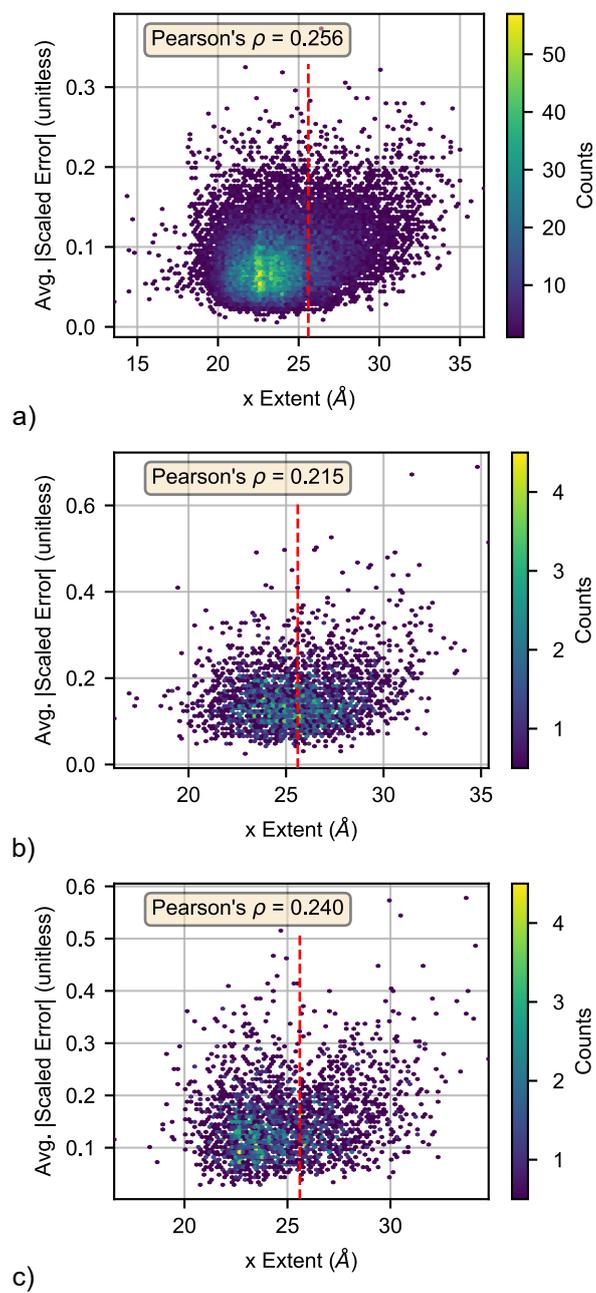


Figure 7.14. Average scaled absolute error vs. input pre-standardization x direction extent for a) training, b) validation, and c) test sets. Red dotted line indicates extent cutoff.

plying data augmentation using any arbitrary, randomly selected, molecular orientation during model training (i.e. not limited to 90 degree rotations); increasing the maximum value limits imposed on the electron density and electrostatic potential. This last point is reinforced by observing that much of the molecule's internal electronic structure was "capped" by these limits being set too low as seen in Figure 7.3b.

Additionally, the model may be coupled with other machine learned models capable of estimating the electronic structure of a molecule so that the physics-based workflow is entirely bypassed – augmenting material screening and accelerating discovery. This outcome could be further improved by adapting the model for sequential design so that new molecules can be proposed for synthesis followed by experimental validation and testing. With transfer learning applied to the convolutional base of the network, the model could be adapted to make predictions on target quantities for which no physics-based workflow exists and where limited experimental data are available; for example, explosive sensitivity testing. Finally, work to quantify the uncertainty of the model predictions will greatly enhance the model's utility.

## 8. CONCLUSION

In order to visualize an emerging hot spot in a real energetic system, the intensity ratio method of phosphor thermography was performed to remotely measure temperature from within a simplified PBX undergoing mechanical excitation. Sylgard 184 was used as the binder in the PBX due to its optical clarity and was not shown to have adverse affects on the temperature measurement or sensitivity. ZnO:Ga phosphor sensitivity and calibration are shown to be dependent on incident fluence and should be considered in future designs. Spatially, the pixel-to-pixel standard deviation was shown to be less than 2% for the prescribed temperatures  $\leq 220$  °C. The temporal, frame-to-frame, standard deviation was not computed, but observed measurement fluctuations are likely attributed to shot-to-shot fluctuations in laser energy and can be corrected in future tests. This work was useful because the spatially and temporally resolved temperature field of an evolving hot spot can be used to shed light on the underlying physical phenomena governing hot spot initiation and growth.

A review of the maximum likelihood estimation process for the normal and logistic latent distributions commonly employed in sensitivity tests was provided. The underlying assumption that the latent distribution is unimodal greatly simplifies the parameter estimation procedure but cannot generalize to more complex latent distributions. An updated estimation procedure applicable to bimodal latent distributions was motivated and detailed.

The influence of simple and realistic pore geometries on the sensitivity of shocked HMX was shown to be significant. Pore orientation and aspect ratio are strong drivers of shock sensitivity. Generally speaking, elliptical pores with their major axis aligned with the incoming shock are the most sensitive while those elliptical pores with the major axis perpendicular to the incoming shock are the least sensitive. Despite their usefulness, the metrics of aspect ratio and orientation alone do not adequately capture

the deviations in shock sensitivity of realistic pore geometries. Machine-learned models trained on the simulated collapse of realistic pore shapes can predict sensitivity with good accuracy but interrogation of these models to understand the exact effect of shape features is difficult. Of the models tested, the convolutional neural network (CNN) acting on a standardized binary image representation of the pore shapes demonstrated the best performance. Furthermore, the shape features extracted by the CNN align with human intuition in that pore shapes of similar appearance are adjacent in feature space and have comparable critical threshold velocities. The data generated by this work and the produced surrogate models are made publicly available through an interactive web application. This work can continue to augment research surrounding pore collapse by using the trained surrogate models to directly compare and investigate the relative importance of pore size and pore shape. Additionally, the collected distribution of critical threshold velocities (relating to pore sensitivity) could be used to inform full-scale stochastic continuum models.

Finally, it was demonstrated that a 3D convolutional neural network can effectively parse the electron charge density and electrostatic potential of a molecule to make high accuracy predictions on material properties without the need to hand-craft molecular descriptors. The 3D CNN model, without any hyperparameter tuning, outperformed tuned random forest models using conventional ECFP4 descriptors. The model attained excellent generalization error even when making predictions on structurally dissimilar molecules as observed with scaffold-based splitting. Promoting advancement, the model can be coupled with other machine learned models capable of estimating the electronic structure of a molecule so that the physics-based workflows are entirely bypassed – augmenting material screening and accelerating discovery. This outcome could be further improved by adapting the model for sequential design so that new molecules can be proposed for synthesis followed by experimental validation and testing. With transfer learning applied to the convolutional base of the network, the model could be adapted to make predictions on target quantities for which no physics-based workflow exists and where limited experimental data are available; for example, the explosive sensitivity testing mentioned throughout this work.

## REFERENCES

## REFERENCES

- [1] Paul W Cooper. *Explosives engineering*. John Wiley & Sons, 2018.
- [2] J. J. Hood. *Philos. Mag.* 6, 371, (1878). 20, 323 (1885).
- [3] Jacobus Henricus Van't Hoff. *Etudes de dynamique chimique*, volume 1. Muller, 1884.
- [4] Svante Arrhenius. Über die reaktionsgeschwindigkeit bei der inversion von rohrzucker durch säuren. *Zeitschrift für physikalische Chemie*, 4(1):226–248, 1889.
- [5] M Menzinger and Re Wolfgang. The meaning and use of the arrhenius activation energy. *Angewandte Chemie International Edition in English*, 8(6):438–444, 1969.
- [6] Frank Philip Bowden and Abraham David Yoffe. *Initiation and growth of explosion in liquids and solids*. CUP Archive, 1985.
- [7] John Edwin Field, GM Swallowe, and SN Heavens. Ignition mechanisms of explosives during mechanical deformation. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 382(1782):231–244, 1982.
- [8] John E. Field. Hot spot ignition mechanisms for explosives. *Accounts of Chemical Research*, 25(11):489–496, 1992.
- [9] Marcellin Berthelot. *Explosives and their power*. J. Murray, 1892.
- [10] W Taylor and A Weale. Conditions for the initiation and propagation of detonation in solid explosives. *Transactions of the Faraday Society*, 34:995–1003, 1938.
- [11] Blaine Asay. *Shock wave science and technology reference library, Vol. 5: Non-shock initiation of explosives*, volume 5. Springer Science & Business Media, 2009.
- [12] S. Ek. *Sixth Symposium (Int.) on Detonation*, page 272.
- [13] LC Smith. Los alamos national laboratory explosives orientation course: Sensitivity and sensitivity tests.[to impact, friction, spark, and shock]. Technical report, Los Alamos National Lab., NM (USA), 1987.
- [14] NA Kholevo. New data on the sensitivity of condensed explosives to mechanical shock. Technical report, JOINT PUBLICATIONS RESEARCH SERVICE ARLINGTON VA, 1968.
- [15] Muhamed Sucasca. *Test methods for explosives*. Springer Science & Business Media, 2012.
- [16] Kirt N Sasser and Thaddeus C Speed. In-process hazard classification of explosives. *Safety Management Services Inc., West Jordan, UT*, 2005.

- [17] SM Walley, JE Field, and MW Greenaway. Crystal sensitivities of energetic materials. *Materials Science and Technology*, 22(4):402–413, 2006.
- [18] DA Nesterenko, VA Garanin, AI Kazakov, AG Korepin, and LB Romanova. Energetic properties and impact sensitivity of crystalline explosives. *Russian Journal of Physical Chemistry B*, 8(5):701–711, 2014.
- [19] Stephen M Walley, John E Field, Richard A Biers, William G Proud, David M Williamson, and Andrew P Jardine. The use of glass anvils in drop-weight studies of energetic materials. *Propellants, Explosives, Pyrotechnics*, 40(3):351–365, 2015.
- [20] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.
- [21] Chollet Francois. *Deep learning with python*, 2017.
- [22] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*, springer series in statistics, 2009.
- [23] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2017.
- [24] Tom Michael Mitchell. *The discipline of machine learning*, volume 9. Carnegie Mellon University, School of Computer Science, Machine Learning . . . , 2006.
- [25] David Robinson. What's the difference between data science, machine learning, and artificial intelligence?
- [26] Brian C Barnes. A primer on machine learning for materials and its relevance to army challenges. Technical report, US Army Research Laboratory Aberdeen Proving Ground United States, 2018.
- [27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [28] J. O. Mares, J. K. Miller, N. D. Sharp, D. S. Moore, D. E. Adams, L. J. Groven, J. F. Rhoads, and S. F. Son. Thermal and mechanical response of pbx 9501 under contact excitation. *Journal of Applied Physics*, 113(8):084904, 2013.
- [29] J. K. Miller, J. O. Mares, I. E. Gunduz, S. F. Son, and J. F. Rhoads. The impact of crystal morphology on the thermal responses of ultrasonically-excited energetic materials. *Journal of Applied Physics*, 119(2):024903, 2016.
- [30] WG Proud. The measurement of hot-spots in granulated ammonium nitrate. In *AIP Conference Proceedings*, volume 620, pages 1081–1086. AIP, 2002.
- [31] WG Proud, IJ Kirby, and JE Field. The nature, number and evolution of hot-spots in ammonium nitrate. In *AIP Conference Proceedings*, volume 706, pages 1017–1020. AIP, 2004.
- [32] DL Woody. Experimental procedures for the observance of shear band formation during low velocity impacts. *Review of scientific instruments*, 63(3):2030–2034, 1992.

- [33] Jesus O Mares, JK Miller, IE Gunduz, JF Rhoads, and Steven F Son. Heat generation in an elastic binder system with embedded discrete energetic particles due to high-frequency, periodic mechanical excitation. *Journal of Applied Physics*, 116(20):204902, 2014.
- [34] Z. A. Roberts, A. D. Casey, I. E. Gunduz, J. F. Rhoads, and S. F. Son. The effects of crystal proximity and crystal-binder adhesion on the thermal responses of ultrasonically-excited composite energetic materials. *Journal of Applied Physics*, 122(24):244901, 2017.
- [35] Sizhu You, Ming-Wei Chen, Dana D Dlott, and Kenneth S Suslick. Ultrasonic hammer produces hot spots in solids. *Nature communications*, 6:6581, 2015.
- [36] Ming Wei Chen, Sizhu You, Kenneth S. Suslick, and Dana D. Dlott. Hot spots in energetic materials generated by infrared and ultrasound, detected by thermal imaging microscopy. *Review of Scientific Instruments*, 85(2), 2014.
- [37] SW Allison and GT Gillies. Remote thermometry with thermographic phosphors: Instrumentation and applications. *Review of Scientific Instruments*, 68(7):2615–2650, 1997.
- [38] A H Khalid, K Kontis, and H-Z Behtash. Phosphor thermometry in gas turbines: Consideration factors. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 224(7):745–755, 2010.
- [39] Marcus Aldén, Alaa Omrane, Mattias Richter, and Gustaf Särner. Thermographic phosphors for thermometry: a survey of combustion applications. *Progress in energy and combustion science*, 37(4):422–461, 2011.
- [40] Jan Brübach, Christian Pflitsch, Andreas Dreizler, and Burak Atakan. On surface temperature measurements with thermographic phosphors: A review. *Progress in Energy and Combustion Science*, 39(1):37–60, 2013.
- [41] Christoph Knappe. *Phosphor Thermometry on Surfaces-A Study of its Methodology and its Practical Applications*. Lund University Publications, 2013.
- [42] N Fuhrmann, J Brübach, and A Dreizler. Phosphor thermometry: A comparison of the luminescence lifetime and the intensity ratio approach. *Proceedings of the Combustion Institute*, 34(2):3611–3618, 2013.
- [43] Gustaf Särner, Mattias Richter, and Marcus Aldén. Two-dimensional thermometry using temperature-induced line shifts of zn: Zn and zn: Ga fluorescence. *Optics letters*, 33(12):1327–1329, 2008.
- [44] Christopher Abram, Benoit Fond, and Frank Beyrau. High-precision flow temperature imaging using ZnO thermographic phosphor tracer particles. *Optics Express*, 23(15):19453, 2015.
- [45] F Abou Nada, Christoph Knappe, Xin Xu, Mattias Richter, and Marcus Aldén. Development of an automatic routine for calibration of thermographic phosphors. *Measurement Science and Technology*, 25(2):025201, 2014.
- [46] Barry T Neyer. More efficient sensitivity testing. Technical report, EG and G Mound Applied Technologies, Miamisburg, OH (USA), 1989.

- [47] Barry T Never. A d-optimality-based sensitivity test. *Technometrics*, 36(1):61–70, 1994.
- [48] Jerome Cornfield and Nathan Mantel. Some new aspects of the application of maximum likelihood to the calculation of the dosage response curve. *Journal of the American Statistical Association*, 45(250):181–210, 1950.
- [49] Abraham Golub and Frank E Grubbs. Analysis of sensitivity experiments when the levels of stimulus cannot be controlled. *Journal of the American Statistical Association*, 51(274):257–265, 1956.
- [50] Wilfrid Joseph Dixon and Am M Mood. A method for obtaining and analyzing sensitivity data. *Journal of the American Statistical Association*, 43(241):109–126, 1948.
- [51] DJ Finney. Probit analysis, cambridge university press. *Cambridge, UK*, 1947.
- [52] Chester Ittner Bliss. The calculation of the dosage-mortality curve. *Annals of Applied Biology*, 22(1):134–167, 1935.
- [53] Chester Ittner Bliss. The calculation of the dosage-mortality curve. *Annals of Applied Biology*, 22(1):149, 1935. Appendix by R. A. Fisher.
- [54] Mervyn J Silvapulle. On the existence of maximum likelihood estimators for the binomial response models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 310–313, 1981.
- [55]
- [56] Peter Rossi. *Bayesian non-and semi-parametric methods and applications*. Princeton University Press, 2014.
- [57] FP Bowden and SD Yoffe. Initiation and growth of explosions in liquids and solids cambridge univ. *Press, 1952*, §§ 3.1-3.2, 4, 1952.
- [58] Frank Philip Bowden and OA Gurton. Initiation of solid explosives by impact and friction: the influence of grit. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 198(1054):337–349, 1949.
- [59] MM Chaudhri and John Edwin Field. The role of rapidly compressed gas pockets in the initiation of condensed explosives. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 340(1620):113–128, 1974.
- [60] JP Dear and JE Field. A study of the collapse of arrays of cavities. *Journal of Fluid Mechanics*, 190:409–425, 1988.
- [61] CS Coffey. Phonon generation and energy localization by moving edge dislocations. *Physical Review B*, 24(12):6984, 1981.
- [62] John E Field. Hot spot ignition mechanisms for explosives. *Accounts of chemical Research*, 25(11):489–496, 1992.
- [63] Tzu-Ray Shan and Aidan P Thompson. Shock-induced hotspot formation and chemical reaction initiation in petn containing a spherical void. In *J. Phys.: Conf. Ser.*, volume 500, page 172009, 2014.

- [64] FM Najjar, WM Howard, LE Fried, MR Manaa, A Nichols III, and G Levesque. Computational study of 3-d hot-spot initiation in shocked insensitive high-explosive. In *AIP conference proceedings*, volume 1426, pages 255–258. American Institute of Physics, 2012.
- [65] David Furman, Ronnie Kosloff, and Yehuda Zeiri. Effects of nanoscale heterogeneities on the reactivity of shocked erythritol tetranitrate. *The Journal of Physical Chemistry C*, 120(50):28886–28893, 2016.
- [66] Reilly M Eason and Thomas D Sewell. Molecular dynamics simulations of the collapse of a cylindrical pore in the energetic material  $\alpha$ -rdx. *Journal of Dynamic Behavior of Materials*, 1(4):423–438, 2015.
- [67] Ryan A Austin, Nathan R Barton, John E Reaugh, and Laurence E Fried. Direct numerical simulation of shear localization and decomposition reactions in shock-loaded hmx crystal. *Journal of Applied Physics*, 117(18):185902, 2015.
- [68] H Keo Springer, Sorin Bastea, Albert L Nichols III, Craig M Tarver, and John E Reaugh. Modeling the effects of shock pressure and pore morphology on hot spot mechanisms in hmx. *Propellants, Explosives, Pyrotechnics*, 43(8):805–817, 2018.
- [69] A Kapahi and HS Udaykumar. Three-dimensional simulations of dynamics of void collapse in energetic materials. *Shock Waves*, 25(2):177–187, 2015.
- [70] Nirmal Kumar Rai and HS Udaykumar. Three-dimensional simulations of void collapse in energetic materials. *Physical Review Fluids*, 3(3):033201, 2018.
- [71] Nirmal Kumar Rai, Martin J Schmidt, and HS Udaykumar. High-resolution simulations of cylindrical void collapse in energetic materials: Effect of primary and secondary collapse on initiation thresholds. *Physical Review Fluids*, 2(4):043202, 2017.
- [72] Tingting Zhou, Jianfeng Lou, Yangeng Zhang, Huajie Song, and Fenglei Huang. Hot spot formation and chemical reaction initiation in shocked hmx crystals with nanovoids: a large-scale reactive molecular dynamics study. *Physical Chemistry Chemical Physics*, 18(26):17627–17645, 2016.
- [73] S Roy, NK Rai, O Sen, DB Hardin, AS Diggs, and HS Udaykumar. Modeling mesoscale energy localization in shocked hmx, part ii: training machine-learned surrogate models for void shape and void–void interaction effects. *Shock Waves*, pages 1–23, 2019.
- [74] George Arthur Levesque and Peter Vitello. The effect of pore morphology on hot spot temperature. *Propellants, Explosives, Pyrotechnics*, 40(2):303–308, 2015.
- [75] Nirmal Kumar Rai, Martin J Schmidt, and HS Udaykumar. Collapse of elongated voids in porous energetic materials: Effects of void orientation and aspect ratio on initiation. *Physical Review Fluids*, 2(4):043201, 2017.
- [76] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- [77] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Goullart, and Tony Yu. scikit-image: image processing in python. <https://doi.org/10.7717/peerj.453>, 2014.

- [78] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [79] Ralph Menikoff and Thomas D Sewell. Constituent properties of hmx needed for mesoscale simulations. *Combustion theory and modelling*, 6(1):103–125, 2002.
- [80] Craig M Tarver, Steven K Chidester, and Albert L Nichols. Critical conditions for impact-and shock-induced hot spots in solid explosives. *The Journal of Physical Chemistry*, 100(14):5794–5799, 1996.
- [81] Thomas D Sewell, Ralph Menikoff, Dmitry Bedrov, and Grant D Smith. A molecular dynamics simulation study of elastic properties of hmx. *The Journal of chemical physics*, 119(14):7417–7426, 2003.
- [82] Dipendra Jha, Logan Ward, Arindam Paul, Wei-keng Liao, Alok Choudhary, Chris Wolverton, and Ankit Agrawal. Elemnet: Deep learning the chemistry of materials from only elemental composition. *Sci. Rep.*, 8(1):17593, 2018.
- [83] Atsuto Seko, Hiroyuki Hayashi, Keita Nakayama, Akira Takahashi, and Isao Tanaka. Representation of compounds for machine-learning prediction of physical properties. *Phys. Rev. B*, 95(14):144110, 2017.
- [84] Harold Moellering and John N Rayner. The dual axis fourier shape analysis of closed cartographic forms. *The Cartographic Journal*, 19(1):53–59, 1982.
- [85] Frank P Kuhl and Charles R Giardina. Elliptic fourier features of a closed contour. *Computer graphics and image processing*, 18(3):236–258, 1982.
- [86] P Markondeya Raj and W Roger Cannon. 2-d particle shape averaging and comparison using fourier descriptors. *Powder technology*, 104(2):180–189, 1999.
- [87] Harold Moellering and John N Rayner. The harmonic analysis of spatial shapes using dual axis fourier shape analysis (dafsa). *Geographical Analysis*, 13(1):64–77, 1981.
- [88] Gösta H Granlund. Fourier preprocessing for hand print character recognition. *IEEE transactions on computers*, 100(2):195–201, 1972.
- [89] Charles T Zahn and Ralph Z Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on computers*, 100(3):269–281, 1972.
- [90] Katherine P Rice, Aaron E Saunders, and Mark P Stoykovich. Classifying the shape of colloidal nanocrystals by complex fourier descriptor analysis. *Crystal growth & design*, 12(2):825–831, 2012.
- [91] Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug Discovery Today*, 23(6):1241–1250, 2018.
- [92] Adam C Mater and Michelle L Coote. Deep learning in chemistry. *J. Chem. Inf. Model.*, 59(6):2545–2559, 2019.
- [93] Chollet Francois. Deep learning with python, 2017.
- [94] Ruho Kondo, Shunsuke Yamakawa, Yumi Masuoka, Shin Tajima, and Ryoji Asahi. Microstructure recognition using convolutional neural networks for prediction of ionic conductivity in ceramics. *Acta Mater.*, 141:29–38, 2017.

- [95] Ahmet Cecen, Hanjun Dai, Yuksel C Yabansu, Surya R Kalidindi, and Le Song. Material structure-property linkages using three-dimensional convolutional neural networks. *Acta Mater.*, 146:76–84, 2018.
- [96] Zijiang Yang, Yuksel C Yabansu, Reda Al-Bahrani, Wei-keng Liao, Alok N Choudhary, Surya R Kalidindi, and Ankit Agrawal. Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets. *Comput. Mater. Sci.*, 151:278–287, 2018.
- [97] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.
- [98] François Chollet et al. Keras. <https://github.com/fchollet/keras>, 2015.
- [99] James Bergstra, Daniel Yamins, and David Daniel Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. 2013.
- [100] O Anatole von Lilienfeld. Quantum machine learning in chemical compound space. *Angew. Chem., Int. Ed.*, 57(16):4164–4169, 2018.
- [101] Connor W Coley, William H Green, and Klavs F Jensen. Machine learning in computer-aided synthesis planning. *Acc. Chem. Res.*, 51(5):1281–1289, 2018.
- [102] Maneesh K Yadav. On the synthesis of machine learning and automated reasoning for an artificial synthetic organic chemist. *New J. Chem.*, 41(4):1411–1416, 2017.
- [103] Igor Iosifovich Baskin, Timur Ismailovich Madzhidov, Igor Sergeevich Antipin, and Alexandre A Varnek. Artificial intelligence in synthetic chemistry: achievements and prospects. *Russ. Chem. Rev.*, 86(11):1127–1156, 2017.
- [104] Connor W Coley, Regina Barzilay, Tommi S Jaakkola, William H Green, and Klavs F Jensen. Prediction of organic reaction outcomes using machine learning. *ACS Cent. Sci.*, 3(5):434–443, 2017.
- [105] Zhenpeng Zhou, Xiaocheng Li, and Richard N Zare. Optimizing chemical reactions with deep reinforcement learning. *ACS Cent. Sci.*, 3(12):1337–1344, 2017.
- [106] Artur M Schweidtmann, Adam D Clayton, Nicholas Holmes, Eric Bradford, Richard A Bourne, and Alexei A Lapkin. Machine learning meets continuous flow chemistry: Automated optimization towards the pareto front of multiple objectives. *Chemical Engineering Journal*, 352:277–282, 2018.
- [107] Sean Ekins. The next era: Deep learning in pharmaceutical research. *Pharmaceutical Research*, 33(11):2594–2603, 2016.
- [108] Dezhen Xue, Prasanna V Balachandran, John Hogden, James Theiler, Deqing Xue, and Turab Lookman. Accelerated search for materials with targeted properties by adaptive design. *Nat. Commun.*, 7:11241, 2016.
- [109] Dipendra Jha, Logan Ward, Zijiang Yang, Christopher Wolverton, Ian Foster, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. Inet: A general purpose deep residual regression framework for materials discovery. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2385–2393. ACM, 2019.

- [110] Mohammad Atif Faiz Afzal, Aditya Sonpal, Mojtaba Haghightlari, Andrew J Schultz, and Johannes Hachmann. A deep neural network model for packing density predictions and its application in the study of 1.5 million organic molecules. *Chem. Sci.*, 10(36):8374–8383, 2019.
- [111] Anand Chandrasekaran, Deepak Kamal, Rohit Batra, Chiho Kim, Lihua Chen, and Rampi Ramprasad. Solving the electronic structure problem with machine learning. *npj Comput. Mater.*, 5(1):22, 2019.
- [112] Alberto Fabrizio, Benjamin Meyer, Michele Ceriotti, Clemence Corminboeuf, et al. Electron density learning of non-covalent systems. *Chem. Sci.*, 2019.
- [113] Raghunathan Ramakrishnan and O Anatole von Lilienfeld. Machine learning, quantum mechanics, and chemical compound space. *arXiv preprint arXiv:1510.07512*, 2015.
- [114] Ghanshyam Pilania, Chenchen Wang, Xun Jiang, Sanguthevar Rajasekaran, and Ramamurthy Ramprasad. Accelerating materials property predictions using machine learning. *Sci. Rep.*, 3:2810, 2013.
- [115] John C Snyder, Matthias Rupp, Katja Hansen, Klaus-Robert Müller, and Kieron Burke. Finding density functionals with machine learning. *Phys. Rev. Lett.*, 108(25):253002, 2012.
- [116] Mojtaba Haghightlari and Johannes Hachmann. Advances of machine learning in molecular modeling and simulation. *Curr. Opin. Chem. Eng.*, 23:51–57, 2019.
- [117] Saeed Yousefinejad and Bahram Hemmateenejad. Chemometrics tools in qsar/qspr studies: A historical perspective. *Chemometrics and Intelligent Laboratory Systems*, 149:177–204, 2015.
- [118] John BO Mitchell. Machine learning methods in chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 4(5):468–481, 2014.
- [119] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: Data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.
- [120] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. "O'Reilly Media, Inc.", 2017.
- [121] R. Todeschini, V. Consonni, and P. Gramatica. 4.05 - chemometrics in qsar. In Steven D. Brown, Romá Tauler, and Beata Walczak, editors, *Comprehensive Chemometrics*, pages 129 – 172. Elsevier, Oxford, 2009.
- [122] Kunal Roy, Supratik Kar, and Rudra Narayan Das. *Understanding the basics of QSAR for applications in pharmaceutical sciences and risk assessment*. Academic press, 2015.
- [123] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, 108(5):058301, 2012.
- [124] Olexandr Isayev, Corey Oses, Cormac Toher, Eric Gossett, Stefano Curtarolo, and Alexander Tropsha. Universal fragment descriptors for predicting properties of inorganic crystals. *Nat. Commun.*, 8:15679, 2017.

- [125] Frank R Burden. Molecular identification number for substructure searches. *J. Chem. Inf. Comput. Sci.*, 29(3):225–227, 1989.
- [126] Logan Ward, Ruoqian Liu, Amar Krishna, Vinay I Hegde, Ankit Agrawal, Alok Choudhary, and Chris Wolverton. Including crystal structure attributes in machine learning models of formation energies via voronoi tessellations. *Phys. Rev. B*, 96(2):024104, 2017.
- [127] Esben Jannik Bjerrum. Fetch the sheriff! i found a fingerprint!
- [128] Logan Ward, Ankit Agrawal, Alok Choudhary, and Christopher Wolverton. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Comput. Mater.*, 2:16028, 2016.
- [129] Roberto Todeschini and Viviana Consonni. *Handbook of Molecular Descriptors*, volume 11. John Wiley & Sons, 2008.
- [130] Kurt Varmuza, M Dehmer, and D Bonchev. *Statistical Modelling of Molecular Descriptors in QSAR/QSPR*. Wiley Online Library, 2012.
- [131] Andrea Grisafi, David M Wilkins, Michael J Willatt, and Michele Ceriotti. Atomic-scale representation and statistical learning of tensorial properties. In *Machine Learning in Chemistry: Data-Driven Algorithms, Learning Systems, and Predictions*, pages 1–21. ACS Publications, 2019.
- [132] S Stanley Young, Fei Yuan, and Mu Zhu. Chemical descriptors are more important than learning algorithms for modelling. *Mol. Inf.*, 31(10):707–710, 2012.
- [133] Luca M Ghiringhelli, Jan Vybiral, Sergey V Levchenko, Claudia Draxl, and Matthias Scheffler. Big data of materials science: Critical role of the descriptor. *Phys. Rev. Lett.*, 114(10):105503, 2015.
- [134] Risi Kondor, Hy Truong Son, Horace Pan, Brandon Anderson, and Shubhendu Trivedi. Covariant compositional networks for learning graphs. *arXiv preprint arXiv:1801.02144*, 2018.
- [135] Risi Kondor. N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials. *arXiv preprint arXiv:1803.01588*, 2018.
- [136] Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In *Advances in Neural Information Processing Systems*, pages 14510–14519, 2019.
- [137] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.
- [138] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- [139] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.*, 59(8):3370–3388, 2019.
- [140] JM Rickman, T Lookman, and SV Kalinin. Materials informatics: From the atomic-level to the continuum. *Acta Mater.*, 2019.

- [141] Marta M Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Pawel Siedlecki. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*, 34(21):3666–3674, 2018.
- [142] Wen Torng and Russ B Altman. 3d deep convolutional neural networks for amino acid environment similarity analysis. *BMC Bioinf.*, 18(1):302, 2017.
- [143] Seiji Kajita, Nobuko Ohba, Ryosuke Jinnouchi, and Ryoji Asahi. A universal 3d voxel descriptor for solid-state material informatics with deep convolutional neural networks. *Sci. Rep.*, 7(1):1–9, 2017.
- [144] Yi Zhou, Jiang Wu, Shuguang Chen, and GuanHua Chen. Toward the exact exchange–correlation potential: A three-dimensional convolutional neural network construct. *The Journal of Physical Chemistry Letters*, 10(22):7264–7269, 2019.
- [145] Brian C Barnes, Daniel C Elton, Zois Boukouvalas, DeCarlos E Taylor, William D Mattson, Mark D Fuge, and Peter W Chung. Machine learning of energetic material properties. *arXiv preprint arXiv:1807.06156*, 2018.
- [146] Brian Barnes, Betsy Rice, and Andrew Sifain. Machine learning of energetic material properties and performance. *Bull. Am. Phys. Soc.*, 2020.
- [147] Daniel C Elton, Zois Boukouvalas, Mark S Butrico, Mark D Fuge, and Peter W Chung. Applying machine learning techniques to predict the properties of energetic materials. *Sci. Rep.*, 8(1):1–12, 2018.
- [148] Kun Yao and John Parkhill. Kinetic energy of hydrocarbons as a function of electron density and convolutional neural networks. *J. Chem. Theory Comput.*, 12(3):1139–1147, 2016.
- [149] Anton V Sinitskiy and Vijay S Pande. Deep neural network computes electron densities and energies of a large set of organic molecules faster than density functional theory (dft). *arXiv preprint arXiv:1809.02723*, 2018.
- [150] Tobias Fink, Heinz Bruggesser, and Jean-Louis Reymond. Virtual exploration of the small-molecule chemical universe below 160 daltons. *Angew. Chem., Int. Ed.*, 44(10):1504–1508, 2005.
- [151] Tobias Fink and Jean-Louis Reymond. Virtual exploration of the chemical universe up to 11 atoms of c, n, o, f: Assembly of 26.4 million structures (110.9 million stereoisomers) and analysis for new ring systems, stereochemistry, physicochemical properties, compound classes, and drug discovery. *J. Chem. Inf. Model.*, 47(2):342–353, 2007.
- [152] Lorenz C Blum and Jean-Louis Reymond. 970 million druglike small molecules for virtual screening in the chemical universe database gdb-13. *J. Am. Chem. Soc.*, 131(25):8732–8733, 2009.
- [153] Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *J. Chem. Inf. Model.*, 52(11):2864–2875, 2012.
- [154] greg Landrum. Open-source cheminformatics software.
- [155] MJ Frisch, GW Trucks, HB Schlegel, GE Scuseria, MA Robb, JR Cheeseman, G Scalmani, V Barone, GA Petersson, H Nakatsuji, et al. Gaussian 16, revision a. 03, gaussian. *Inc., Wallingford CT*, 2016.

- [156] Edward FC Byrd and Betsy M Rice. Improved prediction of heats of formation of energetic materials using quantum mechanical calculations. *The Journal of Physical Chemistry A*, 110(3):1005–1013, 2006.
- [157] Betsy M Rice, Jennifer J Hare, and Edward FC Byrd. Accurate predictions of crystal densities using quantum mechanical molecular volumes. *The Journal of Physical Chemistry A*, 111(42):10874–10879, 2007.
- [158] Betsy M Rice and Edward FC Byrd. Evaluation of electrostatic descriptors for predicting crystalline density. *J. Comput. Chem.*, 34(25):2146–2151, 2013.
- [159] S Bastea, LE Fried, KR Glaeseman, WM Howard, IFW Kuo, PC Souers, and PA Vitello. Cheetah 8.0 thermochemical code. Energetic Materials Center, Lawrence Livermore National Laboratory, 2015.
- [160] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [161] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [162] Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O’Reilly Media, 2019. <https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837>.
- [163] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chem. Sci.*, 9(2):513–530, 2018.
- [164] Damian Leśniak, Sabina Podlowska, Stanisław Jastrzębski, Igor Sieradzki, Andrzej J Bojarski, and Jacek Tabor. Development of new methods needs proper evaluation—benchmarking sets for machine learning experiments for class a gpcrs. *J. Chem. Inf. Model.*, 59(12):4974–4992, 2019.
- [165] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, 50(5):742–754, 2010.
- [166] Kangway V Chuang and Michael J Keiser. Adversarial controls for scientific machine learning, 2018.
- [167] Christoph Rücker, Gerta Rücker, and Markus Meringer. y-randomization and its variants in qspr/qsar. *J. Chem. Inf. Model.*, 47(6):2345–2357, 2007.
- [168] Kangway V Chuang and Michael J Keiser. Comment on “predicting reaction performance in c–n cross-coupling using machine learning”. *Science*, 362(6416):eaat8603, 2018.

## APPENDIX

## A. DERIVATION OF MAXIMUM LIKELIHOOD ESTIMATORS AND INFORMATION MATRIX FOR A MIXTURE OF TWO GAUSSIANS

Maximum likelihood estimators are a poor choice when estimating the parameters for a mixture of Gaussians. However, their derivation is provided here for reference.

A Gaussian (normal) distribution is parameterized by two variables; aptly named mu ( $\mu$ , representing the center of the distribution) and sigma ( $\sigma$ , identifying the spread of the distribution).

The probability density function (pdf) of a single Gaussian distribution is given by

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (\text{A.1})$$

and a mixture of two Gaussian distributions is defined as

$$p(x|\mu_1, \mu_2, \sigma_1^2, \sigma_2^2) = \pi_1 p(x|\mu_1, \sigma_1^2) + \pi_2 p(x|\mu_2, \sigma_2^2) \quad (\text{A.2})$$

where the coefficients  $\pi_1$  and  $\pi_2$  must be greater than or equal to zero and sum to one.

Following the notation used in section 5.1, the following variables and functions are defined:

$$z_1^{(i)} = \frac{L^{(i)} - \mu_1}{\sigma_1} \quad (\text{A.3})$$

$$z_2^{(i)} = \frac{L^{(i)} - \mu_2}{\sigma_2} \quad (\text{A.4})$$

$$f(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \quad (\text{A.5})$$

$$P^{(i)} = \pi_1 \int_{-\infty}^{z_1^{(i)}} f(t)dt + \pi_2 \int_{-\infty}^{z_2^{(i)}} f(t)dt = \pi_1 \int_{-\infty}^{z_1^{(i)}} f(t)dt + (1 - \pi_1) \int_{-\infty}^{z_2^{(i)}} f(t)dt \quad (\text{A.6})$$

where  $L^{(i)}$  is the stimulus level  $L$  applied to test article  $i$ .  $z_1^{(i)}$  and  $z_2^{(i)}$  are the z-scores of stimulus level  $L^{(i)}$  standardized with parameters  $(\mu_1, \sigma_1)$  and  $(\mu_2, \sigma_2)$ .  $f(t)$  is the pdf of a standard normal distribution (Gaussian distribution with  $\mu = 0$  and  $\sigma = 1$ ). And  $P^{(i)}$  is the probability of observing a ‘go’ for a randomly selected test article at stimulus level  $L^{(i)}$ .

Additionally,  $N^{(i)}$  and  $M^{(i)}$  are the number of ‘goes’ and ‘no-goes’ observed at stimulus level  $L^{(i)}$ , respectively.

With this information, the likelihood function for a binary dataset (‘go’, ‘no-go’) at  $n$  stimulus levels is given by

$$L(\mu_1, \mu_2, \sigma_1, \sigma_2, \pi_1) = \prod_{i=1}^n \binom{N^{(i)} + M^{(i)}}{N^{(i)}} (P^{(i)})^{N^{(i)}} (Q^{(i)})^{M^{(i)}} \quad (\text{A.7})$$

where  $Q^{(i)}$  is equal to  $1 - P^{(i)}$ .

The natural logarithm of likelihood function is

$$\log(L) = l = \sum_{i=1}^n \left[ \log \binom{N^{(i)} + M^{(i)}}{N^{(i)}} + N^{(i)} \log(P^{(i)}) + M^{(i)} \log(Q^{(i)}) \right]. \quad (\text{A.8})$$

### First Partial - MLEs

The values of  $\mu_1$ ,  $\mu_2$ ,  $\sigma_1$ ,  $\sigma_2$ , and  $\pi_1$  that produce a maximum value of equation A.8 are the maximum likelihood estimators for those parameters. The maximum value of eq. A.8 occurs when the partial derivatives of eq. A.8 with respect to the parameters are all equal to zero.

The first order partial derivatives of  $l$  with respect to  $\mu_1$ ,  $\mu_2$ ,  $\sigma_1$ ,  $\sigma_2$ , and  $\pi_1$  are given by

$$\frac{\partial l}{\partial \mu_1} = \sum_{i=1}^n \left[ N^{(i)} \frac{\partial(\log(P^{(i)}))}{\partial P^{(i)}} \frac{\partial P^{(i)}}{\partial z_1^{(i)}} \frac{dz_1^{(i)}}{d\mu_1} + M^{(i)} \frac{\partial(\log(Q^{(i)}))}{\partial Q^{(i)}} \frac{\partial Q^{(i)}}{\partial z_1^{(i)}} \frac{dz_1^{(i)}}{d\mu_1} \right] \quad (\text{A.9})$$

$$\frac{\partial l}{\partial \mu_2} = \sum_{i=1}^n \left[ N^{(i)} \frac{\partial(\log(P^{(i)}))}{\partial P^{(i)}} \frac{\partial P^{(i)}}{\partial z_2^{(i)}} \frac{dz_2^{(i)}}{d\mu_2} + M^{(i)} \frac{\partial(\log(Q^{(i)}))}{\partial Q^{(i)}} \frac{\partial Q^{(i)}}{\partial z_2^{(i)}} \frac{dz_2^{(i)}}{d\mu_2} \right] \quad (\text{A.10})$$

$$\frac{\partial l}{\partial \sigma_1} = \sum_{i=1}^n \left[ N^{(i)} \frac{\partial(\log(P^{(i)}))}{\partial P^{(i)}} \frac{\partial P^{(i)}}{\partial z_1^{(i)}} \frac{dz_1^{(i)}}{d\sigma_1} + M^{(i)} \frac{\partial(\log(Q^{(i)}))}{\partial Q^{(i)}} \frac{\partial Q^{(i)}}{\partial z_1^{(i)}} \frac{dz_1^{(i)}}{d\sigma_1} \right] \quad (\text{A.11})$$

$$\frac{\partial l}{\partial \sigma_2} = \sum_{i=1}^n \left[ N^{(i)} \frac{\partial(\log(P^{(i)}))}{\partial P^{(i)}} \frac{\partial P^{(i)}}{\partial z_2^{(i)}} \frac{dz_2^{(i)}}{d\sigma_2} + M^{(i)} \frac{\partial(\log(Q^{(i)}))}{\partial Q^{(i)}} \frac{\partial Q^{(i)}}{\partial z_2^{(i)}} \frac{dz_2^{(i)}}{d\sigma_2} \right] \quad (\text{A.12})$$

$$\frac{\partial l}{\partial \mu_2} = \sum_{i=1}^n \left[ N^{(i)} \frac{\partial(\log(P^{(i)}))}{\partial P^{(i)}} \frac{\partial P^{(i)}}{\partial \pi_1} + M^{(i)} \frac{\partial(\log(Q^{(i)}))}{\partial Q^{(i)}} \frac{\partial Q^{(i)}}{\partial \pi_1} \right]. \quad (\text{A.13})$$

The derivatives in equations A.9 to A.13 are obtained as

$$\frac{\partial \log(P^{(i)})}{\partial P^{(i)}} = \frac{1}{P^{(i)}} \quad (\text{A.14})$$

$$\frac{\partial \log(Q^{(i)})}{\partial Q^{(i)}} = \frac{1}{Q^{(i)}} \quad (\text{A.15})$$

$$\frac{\partial P^{(i)}}{\partial z_1^{(i)}} = \pi_1 f(z_1^{(i)}) \quad (\text{A.16})$$

$$\frac{\partial P^{(i)}}{\partial z_2^{(i)}} = (1 - \pi_1) f(z_2^{(i)}) \quad (\text{A.17})$$

$$\frac{\partial Q^{(i)}}{\partial z_1^{(i)}} = -\pi_1 f(z_1^{(i)}) \quad (\text{A.18})$$

$$\frac{\partial Q^{(i)}}{\partial z_2^{(i)}} = -(1 - \pi_1) f(z_2^{(i)}) \quad (\text{A.19})$$

$$\frac{\partial z_1^{(i)}}{\partial \mu_1} = \frac{-1}{\sigma_1} \quad (\text{A.20})$$

$$\frac{\partial z_2^{(i)}}{\partial \mu_2} = \frac{-1}{\sigma_2} \quad (\text{A.21})$$

$$\frac{\partial z_1^{(i)}}{\partial \sigma_1} = \frac{-(L^{(i)} - \mu_1)}{\sigma_1^2} \quad (\text{A.22})$$

$$\frac{\partial z_2^{(i)}}{\partial \sigma_2} = \frac{-(L^{(i)} - \mu_2)}{\sigma_2^2} \quad (\text{A.23})$$

$$\frac{\partial P^{(i)}}{\partial \pi_1} = I = \int_{-\infty}^{z_1^{(i)}} f(t)dt - \int_{-\infty}^{z_2^{(i)}} f(t)dt \quad (\text{A.24})$$

$$\frac{\partial Q^{(i)}}{\partial \pi_1} = -I = -\int_{-\infty}^{z_1^{(i)}} f(t)dt + \int_{-\infty}^{z_2^{(i)}} f(t)dt. \quad (\text{A.25})$$

Substituting equations A.14 – A.25 into equations A.9 – A.13 yields

$$\frac{\partial l}{\partial \mu_1} = \sum_{i=1}^n \left[ N^{(i)} \frac{1}{P^{(i)}} \pi_1 f(z_1^{(i)}) \frac{-1}{\sigma_1} + M^{(i)} \frac{1}{Q^{(i)}} \pi_1 f(z_1^{(i)}) \frac{1}{\sigma_1} \right] \quad (\text{A.26})$$

$$\frac{\partial l}{\partial \mu_2} = \sum_{i=1}^n \left[ N^{(i)} \frac{1}{P^{(i)}} (1 - \pi_1) f(z_2^{(i)}) \frac{-1}{\sigma_2} + M^{(i)} \frac{1}{Q^{(i)}} (1 - \pi_1) f(z_2^{(i)}) \frac{1}{\sigma_2} \right] \quad (\text{A.27})$$

$$\frac{\partial l}{\partial \sigma_1} = \sum_{i=1}^n \left[ N^{(i)} \frac{1}{P^{(i)}} \pi_1 f(z_1^{(i)}) \frac{-(L^{(i)} - \mu_1)}{\sigma_1^2} + M^{(i)} \frac{1}{Q^{(i)}} \pi_1 f(z_1^{(i)}) \frac{(L^{(i)} - \mu_1)}{\sigma_1^2} \right] \quad (\text{A.28})$$

$$\frac{\partial l}{\partial \sigma_2} = \sum_{i=1}^n \left[ N^{(i)} \frac{1}{P^{(i)}} (1 - \pi_1) f(z_2^{(i)}) \frac{-(L^{(i)} - \mu_2)}{\sigma_2^2} + M^{(i)} \frac{1}{Q^{(i)}} (1 - \pi_1) f(z_2^{(i)}) \frac{(L^{(i)} - \mu_2)}{\sigma_2^2} \right] \quad (\text{A.29})$$

$$\frac{\partial l}{\partial \pi_1} = \sum_{i=1}^n \left[ N^{(i)} \frac{1}{P^{(i)}} I - M^{(i)} \frac{1}{Q^{(i)}} I \right] \quad (\text{A.30})$$

## Second Partial - Information Matrix

The information matrix is defined as the negative of the expectation of the Hessian of the log likelihood function with respect to the parameters. That is,

$$INF = -E[H(l)]. \quad (\text{A.31})$$

The Hessian of the log likelihood function is a matrix composed of the second partial derivatives of said function taken with respect to the parameters.

$$H = \begin{bmatrix} \frac{\partial^2 l}{\partial \mu_1^2} & \frac{\partial^2 l}{\partial \sigma_1 \partial \mu_1} & \frac{\partial^2 l}{\partial \mu_2 \partial \mu_1} & \frac{\partial^2 l}{\partial \sigma_2 \partial \mu_1} & \frac{\partial^2 l}{\partial \pi_1 \partial \mu_1} \\ \frac{\partial^2 l}{\partial \mu_1 \partial \sigma_1} & \frac{\partial^2 l}{\partial \sigma_1^2} & \frac{\partial^2 l}{\partial \mu_2 \partial \sigma_1} & \frac{\partial^2 l}{\partial \sigma_2 \partial \sigma_1} & \frac{\partial^2 l}{\partial \pi_1 \partial \sigma_1} \\ \frac{\partial^2 l}{\partial \mu_1 \partial \mu_2} & \frac{\partial^2 l}{\partial \sigma_1 \partial \mu_2} & \frac{\partial^2 l}{\partial \mu_2^2} & \frac{\partial^2 l}{\partial \sigma_2 \partial \mu_2} & \frac{\partial^2 l}{\partial \pi_1 \partial \mu_2} \\ \frac{\partial^2 l}{\partial \mu_1 \partial \sigma_2} & \frac{\partial^2 l}{\partial \sigma_1 \partial \sigma_2} & \frac{\partial^2 l}{\partial \mu_2 \partial \sigma_2} & \frac{\partial^2 l}{\partial \sigma_2^2} & \frac{\partial^2 l}{\partial \pi_1 \partial \sigma_2} \\ \frac{\partial^2 l}{\partial \mu_1 \partial \pi_1} & \frac{\partial^2 l}{\partial \sigma_1 \partial \pi_1} & \frac{\partial^2 l}{\partial \mu_2 \partial \pi_1} & \frac{\partial^2 l}{\partial \sigma_2 \partial \pi_1} & \frac{\partial^2 l}{\partial \pi_1^2} \end{bmatrix} \quad (\text{A.32})$$

The matrix  $H$  is symmetric so only the terms given by the upper or lower diagonal need specified to determine the entire matrix.

Determining the second partial derivatives of the log likelihood function, and then applying the expectation operator and negating the subsequent expression, is straightforward - albeit tedious. As an example the full derivation of the term  $-E \left[ \frac{\partial^2 l}{\partial \mu_1^2} \right]$  is given here:

$$\frac{\partial^2 l}{\partial \mu_1^2} = \sum_{i=1}^n \left[ N^{(i)} \frac{-\pi_1}{\sigma_1} \frac{\partial \left( \frac{f(z_1^{(i)})}{P^{(i)}} \right)}{\partial \mu_1} + M^{(i)} \frac{\pi_1}{\sigma_1} \frac{\partial \left( \frac{f(z_1^{(i)})}{Q^{(i)}} \right)}{\partial \mu_1} \right] \quad (\text{A.33})$$

The expanded derivatives in equation A.33 are

$$\frac{\partial \left( \frac{f(z_1^{(i)})}{P^{(i)}} \right)}{\partial \mu_1} = \frac{1}{P^{(i)}} \frac{\partial f(z_1^{(i)})}{\partial \mu_1} + f(z_1^{(i)}) \frac{-1}{(P^{(i)})^2} \frac{\partial P^{(i)}}{\partial \mu_1} \quad (\text{A.34})$$

$$\frac{\partial \left( \frac{f(z_1^{(i)})}{Q^{(i)}} \right)}{\partial \mu_1} = \frac{1}{Q^{(i)}} \frac{\partial f(z_1^{(i)})}{\partial \mu_1} + f(z_1^{(i)}) \frac{-1}{(Q^{(i)})^2} \frac{\partial Q^{(i)}}{\partial \mu_1} \quad (\text{A.35})$$

Continuing the application of the chain rule

$$\frac{\partial f(z_1^{(i)})}{\partial \mu_1} = \frac{\partial f(z_1^{(i)})}{\partial z_1^{(i)}} \frac{\partial z_1^{(i)}}{\partial \mu_1} = -z_1^{(i)} f(z_1^{(i)}) \frac{-1}{\sigma_1} \quad (\text{A.36})$$

$$\frac{\partial P^{(i)}}{\partial \mu_1} = \frac{\partial P^{(i)}}{\partial z_1^{(i)}} \frac{\partial z_1^{(i)}}{\partial \mu_1} = \pi_1 f(z_1^{(i)}) \frac{-1}{\sigma_1} \quad (\text{A.37})$$

$$\frac{\partial Q^{(i)}}{\partial \mu_1} = \frac{\partial Q^{(i)}}{\partial z_1^{(i)}} \frac{\partial z_1^{(i)}}{\partial \mu_1} = -\pi_1 f(z_1^{(i)}) \frac{-1}{\sigma_1} \quad (\text{A.38})$$

Substituting equations A.36 – A.37 into equations A.34 and A.35 and then further substituting those equations into A.33 produces

$$\begin{aligned} \frac{\partial^2 l}{\partial \mu_1^2} = & \sum_{i=1}^n \left[ N^{(i)} \frac{-\pi_1}{\sigma_1} \left( \frac{1}{P^{(i)}} \frac{z_1^{(i)} f(z_1^{(i)})}{\sigma_1} + \frac{f(z_1^{(i)}) r f(z_1^{(i)})}{(P^{(i)})^2 \sigma_1} \right) \right. \\ & \left. + M^{(i)} \frac{\pi_1}{\sigma_1} \left( \frac{1}{Q^{(i)}} \frac{z_1^{(i)} f(z_1^{(i)})}{\sigma_1} - \frac{f(z_1^{(i)}) r f(z_1^{(i)})}{(Q^{(i)})^2 \sigma_1} \right) \right]. \end{aligned} \quad (\text{A.39})$$

Applying the expectation operator will only affect the  $N^{(i)}$  and  $M^{(i)}$  variables.  $N^{(i)}$  will become  $(N^{(i)} + M^{(i)})P^{(i)}$  and  $M^{(i)}$  will become  $(N^{(i)} + M^{(i)})Q^{(i)}$ . Therefore, after *some* elimination and regrouping of terms

$$-E \left[ \frac{\partial^2 l}{\partial \mu_1^2} \right] = \sum_{i=1}^n (N^{(i)} + M^{(i)}) \frac{\pi_1^2 f(z_1^{(i)})^2}{\sigma_1^2} \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right). \quad (\text{A.40})$$

The other terms in both the Hessian and Information matrix are derived similarly. Defining the information matrix

$$INF = -E[H(l)] = \begin{bmatrix} a_{11} & a_{12} & \dots \\ \vdots & \ddots & \\ a_{51} & & a_{55} \end{bmatrix} \quad (\text{A.41})$$

the elements of the information matrix are:

$$\begin{aligned} a_{11} &= \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \frac{\pi_1^2 f(z_1^{(i)})^2}{\sigma_1^2} \\ a_{12} &= \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \frac{\pi_1^2 f(z_1^{(i)})^2}{\sigma_1^2} z_1^{(i)} \\ a_{13} &= \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \pi_1 (1 - \pi_1) \frac{f(z_1^{(i)}) f(z_2^{(i)})}{\sigma_1 \sigma_2} \end{aligned}$$

$$a_{14} = \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \pi_1(1 - \pi_1) \frac{f(z_1^{(i)})f(z_2^{(i)})}{\sigma_1\sigma_2} z_2^{(i)}$$

$$a_{15} = \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \frac{f(z_1^{(i)})}{\sigma_1} (-\pi_1) I$$

$$a_{22} = \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \frac{\pi_1^2 f(z_1^{(i)})^2}{\sigma_1^2} (z_1^{(i)})^2$$

$$a_{23} = \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \pi_1(1 - \pi_1) \frac{f(z_1^{(i)})f(z_2^{(i)})}{\sigma_1\sigma_2} z_1^{(i)}$$

$$a_{24} = \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \pi_1(1 - \pi_1) \frac{f(z_1^{(i)})f(z_2^{(i)})}{\sigma_1\sigma_2} z_1^{(i)} z_2^{(i)}$$

$$a_{25} = \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \frac{f(z_1^{(i)})}{\sigma_1} (-\pi_1) I z_1^{(i)}$$

$$a_{33} = \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \frac{(1 - \pi_1)^2 f(z_2^{(i)})^2}{\sigma_2^2}$$

$$a_{34} = \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \frac{(1 - \pi_1)^2 f(z_2^{(i)})^2}{\sigma_2^2} z_2^{(i)}$$

$$a_{35} = \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \frac{f(z_2^{(i)})}{\sigma_2} (-1)(1 - \pi_1) I$$

$$a_{44} = \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \frac{(1 - \pi_1)^2 f(z_2^{(i)})^2}{\sigma_2^2} (z_2^{(i)})^2$$

$$a_{45} = \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) \frac{f(z_2^{(i)})}{\sigma_2} (-1)(1 - \pi_1) I z_2^{(i)}$$

$$a_{55} = \sum_{i=1}^n (N^{(i)} + M^{(i)}) \left( \frac{1}{P^{(i)}} + \frac{1}{Q^{(i)}} \right) I^2$$