Statistical and Machine Learning Methods for Quantum Measurements with Single Photon Emitters

A Thesis Submitted to the Committee on Graduate Studies

in Partial Fulfillment of the Requirements of the Degree of Master of Science

in the Faculty of Arts and Science

TRENT UNIVERSITY

Peterborough, Ontario, Canada

Materials Science M.Sc. Graduate Program

January 2025

Abstract

Statistical and Machine Learning Methods for Quantum Measurements with Single Photon Emitters

Dylan G. Stone

With wide applications ranging from quantum communication and metrology to biomedicine, single photon sources in solid-state hosts have become a major area of study. Here, we focus on three applications: nanothermometry, optically detected magnetic resonance (ODMR), and second order autocorrelation. We present novel statistical and machine learning (ML) approaches to extract information from experimental and simulated data and benchmark these methods against traditional inference-based statistical approaches. We found that compared to traditional inference-based methods ML algorithms can: i) predict temperatures at the nanoscale with greater accuracy and with less calibration points than traditional fitting methods; ii) identify the resonance peaks in ODMR spectra with factors ~1.3x and ~4.7x better accuracy and resolution and achieved equal or better performance with ~5x less data; and iii) have the potential to parse second order autocorrelation data more efficiently. ML algorithms are thus powerful tools for quantum sensing techniques.

Keywords: colour centers, diamond, hexagonal boron nitride, quantum emitters, nanoscale sensing, optical thermometry, optically detected magnetic resonance, second order autocorrelation, machine learning, multi-feature linear regression, clustering.

Acknowledgments

I'd like to begin by thanking my supervisor, Dr. Carlo Bradac. Your help throughout this process has been immense and I am so grateful for your guidance, patience, and support. You provided me with so many opportunities, beginning in undergrad, which have led me to where I am today. Thank you. I would also like to thank my committee members, Bill Atkinson, Nisha Agarwal, and Sharif Sadaf, for their expertise and insight.

I would not have been able to complete this project without the funding and support provided to me by NSERC, through the Canadian Graduate Scholarship – Masters, the Ontario Graduate Scholarship, and the various funding provided by Trent University. I would also like to thank the Digital Research Alliance of Canada for allowing me to use their compute clusters for data processing.

Thank you to my friends and cohort, both back home and in Peterborough. Your support and encouragement were both needed and appreciated. Finally, I'd like to thank my family; my mom, dad, and sister. Thank you for supporting me throughout all of my education. You were all my rock, I couldn't have done it without you.

Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Definition
ANN	Artificial Neural Network
CA	Clustering Algorithm
CNN	Convolutional Neural Network
EPV	Events Per Variable
FWHM	Full Width at Half Maximum
GeV	Germanium Vacancy
hBN	Hexagonal Boron Nitride
ISC	Intersystem Crossing
L-M	Levenberg-Marquardt
LR	Linear Regression
LSE	Least Squares Error
MF-LR	Multi-Feature Linear Regression
ML	Machine Learning
MW	Microwave
ND	Nanodiamond
NN	Neural Network
ODMR	Optically Detected Magnetic Resonance
PL	Photoluminescence
PSB	Phonon Side Band
ReLU	Rectified Linear Unit
SF	Standard Fit
SiV	Silicon Vacancy
V_B	Boron Vacancy
VC	Voting Classifier
WCSS	Within Cluster Sum of Squares
ZPL	Zero Phonon Line

1 Introduction

Single photon quantum sources in solid-state hosts have become prime candidates for hardware systems for many advanced quantum technologies.[1–4] Examples of such atom-like quantum emitters include those in diamond,[5–9] silicon carbide,[10] hexagonal boron nitride (hBN),[11–16] zinc oxide,[17] rare earth ions in solids,[18–20] and optically active donors in silicon[21,22]. Such samples are attractive because they often contain spin states that can be readily manipulated and read out, have long coherence times, room temperature operation, can create entangled states, and have spin-dependent optical transitions allowing for spin-photon interfacing and long-distance optical transmission of quantum information.[23,24] There are a wide range of relevant applications for such emitters including quantum communication[25–28] and computation,[29–32] quantum simulation,[33,34] and quantum metrology and sensing.[1,35,36] Additionally, these emitters, along with organic dyes, fluorescent proteins, polymers, and inorganic nanoparticles such as quantum dots, gold nanostructures, and upconversion nanoparticles, have been used as nanothermometers.[37–42] These nanoscale probes have incurred an additional range of applications from fields such as nanomedicine,[7,39,43–49] microfluids,[50] and nanoelectronics[51] with practical realizations from temperature driven gene expression[52,53] and cancer therapy[54,55] to temperature management in high-power microelectronics[56]. Many of these solid-state hosts are technology ready, with potential devices being able to leverage well established nanofabrication techniques from the semiconductor industry and could be integrated into on-chip electronic, magnetic, and photonic nanostructures.[57–59]

Despite the variety and versatility of all these materials, there are still a number of drawbacks when applying them in practical situations. In this study we focus on improving three techniques that either apply or prepare solid-state emitters for use in real world applications: nanothermometry, optically detected magnetic resonance (ODMR) and the second order autocorrelation function. We utilize two specific solid-state emitters in our investigation: nanodiamonds (NDs) and hBN, both with colour center defects.

Nanothermometry has become a powerful tool for characterizing and understanding submicrometric systems whose performance and dynamics are tied to the temperature of the sample, enabling us to measure and control temperature at the nanoscale.[15] In the search for the perfect nanoscale probes, one major drawback persists: many of the currently studied probes are relative rather than absolute, meaning they require individual calibration against reference systems of known temperatures. This can be rather time consuming in practical applications where many probes may be used, as each require their own individual calibration. Furthering this, probes should ideally be calibrated both ex situ and in situ as the environment the probe is placed in can have a significant impact on their optical properties, sometimes referred to as thermal equivalent noise.[37,60]

ODMR has become a well-established and powerful technique with applications including the measuring of electric and magnetic fields, temperature, strain and pressure, and electron and nuclear spins, all with remarkably high sensitivity and nanoscale resolution. Despite this, one of the major drawbacks of ODMR is its reliance on lengthy acquisitions. In practical measurements, accuracy and resolution depend on your ability to identify resonance peaks in the ODMR spectra. Therefore, in order to achieve enough photoluminescent (PL) contrast (i.e. a relatively large enough difference in photon counts

on and off resonance), you have to scan the microwave (MW) signal over a range of frequencies (hundreds of MHz) with dwell times between ~10^{-3}-10^{2}s for every frequency bin.

Solid-state quantum emitters have recently reached near ideal single-photon characteristics for use in quantum information technology. This application requires the efficient identification of bright, stable, single photon emitters.[61] The second order autocorrelation function (or simply autocorrelation or autocorrelation technique) has become a popular technique for the identification of single emitters within a sample filled with anywhere from single emitters to large ensembles.[61–64] Similar to ODMR, in practical scenarios the ability to accurately distinguish between single emitters and ensembles relies on the ability to accurately find and measure the relative dip in PL counts, this time near zero delay. Depending on the efficiency of the system this can be a lengthy process and must be repeated for every optically active site one wishes to categorize.

To overcome these barriers, we focus on the application of machine learning (ML) algorithms to gain improved performance over traditional methods in major benchmarking metrics such as accuracy and resolution, and/or to reduce the required amount of data that is needed to achieve equivalent or better results. In this study we explore a number of ML algorithms, including Multi-Feature Linear Regression (MF-LR), Artificial Neural Networks (ANNs), and K-means++ clustering, with a focus on MF-LR and a Custom Algorithm based on K-means++.

For nanothermometry, we propose an all-optical nanothermometry technique based on fluorescent nanodiamonds and a machine learning MF-LR algorithm. Similar

multiparametric studies have been conducted as shown here [65], however our approach has a few key differences. Our samples are co-doped and host two different emitters, germanium vacancy (GeV) and Silicon vacancy (SiV) centers, in high concentrations (~$5x10^{14}cm^{-3}$). This allows for the concurrent monitoring of potentially twice as many temperature-dependent observables or features (intensity, zero-phonon line position, and full width at half maximum) as both the GeV and SiV centers are excited simultaneously with the same laser. We show that with as few as five nanodiamonds and as little as three calibration temperatures, our MF-LR algorithm can make temperature predictions with a resolution of ~1K $Hz^{-1/2}$ on any uncalibrated ND. Although this is a modest resolution, it overcomes one of the major problems of requiring calibration for every individual probe. We also show that the accuracy and resolution can be improved by increasing the number of calibration points used in model training. During setup, the model can also select the best subset of features to use automatically, to achieve the best possible results. Finally, we also deliberately deviate from traditional studies where accuracy, resolution, and sensitivity are determined from the ND used for training/calibration. Instead, we estimate their generalized values from the uncalibrated nanothermometers. These are thus a lower-bound, rather than a best-case scenario, for practical applications where it is difficult or unfeasible to calibrate each nanosensor before use.

For ODMR, we demonstrate a method based on data clustering that can extract resonance frequencies from ODMR spectra with a factor ~1.3x better accuracy and ~4.7x better resolution than traditional methods based on statistical inference. Additionally, to achieve equivalent accuracy and resolution, our model can determine resonance frequencies with ~5x fewer data points, dramatically improving the efficiency. This allows for a dramatic

reduction in time spent acquiring data. It is also shown to work more reliably for noisy and scarce datasets, which is particularly attractive for real-world acquisition and analysis. This model has the additional bonus of not requiring any training or calibration beforehand, removing the need for the creation of training sets.

For autocorrelation, although results were not found for this study, we discuss examples of current successes in the use of ML algorithms against traditional statistical inferencing methods, which align nicely with the improvements in both performance and efficiency shown with our other investigations. On top of the requirement to have acceptable signal-noise ratios for peak locating, many models require lots of training data to perform acceptably, to this end we also discuss a data simulation technique that allows for the creation of synthetic data that mirrors that of a real-world acquisition. This enables the creation of nearly limitless training data for which the parameters are known with perfect precision for easy benchmarking.

2 Theory

2.1 Colour Centers in Diamond and hBN

In this section we will look at the optical properties of defects in both diamond and hexagonal Boron Nitride (hBN), known as *color centers* or *vacancy centers*. We will discuss the structures of each material, their energy level schemes, and briefly touch on the idea of single versus multiphoton emission. In the following three subsections we will explore specific techniques for probing these color centers to gain information about local temperature (2.1.1), local magnetic fields (2.1.2), and determining the relative number of color centers present in a region (2.1.3), respectively.

Structure

A pure diamond structure consists of repeating three-dimensional unit cells known as *diamond cubic unit cells*, consisting entirely of carbon atoms.[66] In such a lattice, each carbon atom is bonded to four other carbon atoms. Diamond however, is also host to extrinsic defects consisting of complexes of foreign atoms. For instance, *Figure 2.1* shows two cartoon images of silicon (SiV) and germanium vacancy centers (GeV) in a diamond lattice. Each coloured sphere represents a different atom: purple for carbon, pink for silicon, and green for germanium. The white spheres indicate vacant sites where, in a pure diamond lattice, carbon atoms would reside. In the case of SiVs and GeVs, the foreign atom sits in between two adjacent vacant carbon sites. There are many other vacancy centers in diamond being studied, with the most well known being the nitrogen vacancy center, which occupies the location of one of the two adjacent vacant carbon atoms directly.[8,15,67] In this work we dealt with samples containing SiVs and GeVs, therefore they will be the focus of this study.

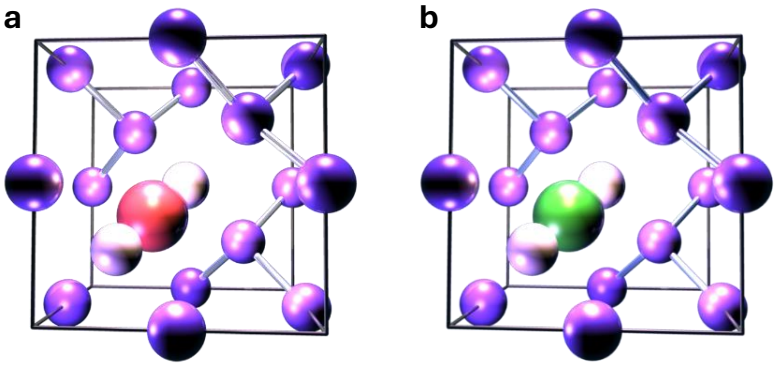

Figure 2.1: Diamond lattice with vacancy centers. **a)** silicon vacancy center and **b)** germanium vacancy center. The purple spheres represent carbon atoms, pink represents silicon, green

represents germanium, and the white spheres represent vacancies, where carbon atoms would normally sit.

Unlike nanodiamonds, hBN has a two-dimensional hexagonal lattice, where bulk material exists in sheets similar to graphite. *Figure 2.2* shows a cartoon image of the lattice structure where orange circles represent nitrogen atoms, purple represents boron atoms, and outlined circles represent vacancies. In a pure lattice, the elements are arranged such that a boron atom is never bonded to another boron atom, likewise for nitrogen. In this work we studied a specific impurity in hBN, the boron vacancy center, which consists of a missing boron atom within the lattice. Other vacancies in hBN exist and are being studied, including nitrogen vacancies, substitutions with elements such as carbon, and vacancies adjacent to substitutions.[11]

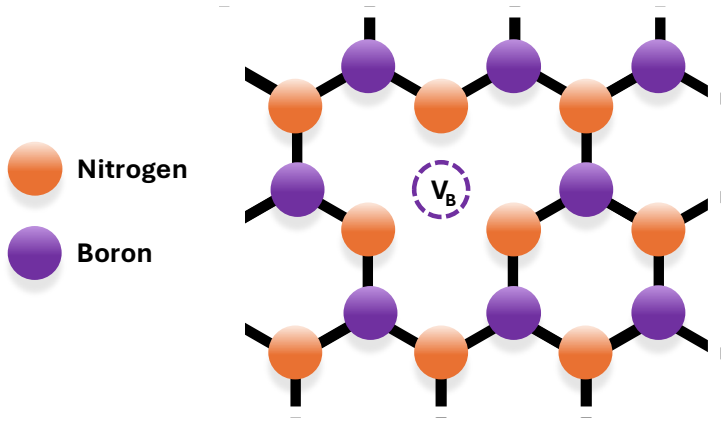

Figure 2.2: The atomic structure of hBN with a Boron vacancy center (V_B). The boron vacancy consists simply of a missing boron atom within the lattice, with no replacement element taking its place.

Energy Level Scheme

Both the SiV and GeV in diamond and V_B in hBN share similar energy level schemes consisting of a ground state, excited state, and intermediate metastable state. These defects have triplet ground and excited states ($S = 1$), with singlet metastable states ($S = 0$), as shown in *Figure 2.3*. The electronic transitions directly between the ground and excited states are spin conserving, and hence a transition from a $m_s = 0, +1,$ or -1 sublevel must end up in the matching spin sublevel in the destination. However, the alternative path provided by the metastable state, known as *intersystem crossing* (ISC), is non-spin conserving. The ISC transitions are spin-selective, in this case the shelving rate from the excited $m_s = 0$ sublevel is much lower than that for the $m_s = \pm 1$ sublevels. Additionally, from the singlet metastable state, decay occurs preferentially towards the ground $m_s = 0$ state. The combination of these two factors are integral to the technique discussed in section 2.1.2, and are discussed in detail there.[15]

Its important to note that the emission frequency differs from that of the absorption frequency due to the energy lost (gained) to phonon emission (absorption), the case of emission is depicted in *Figure 2.3*. The emission of a photon at a lower (higher) energy than that of absorption is known as Stokes (anti-Stokes) excitation. This process is discussed further in section 2.1.1.[68]

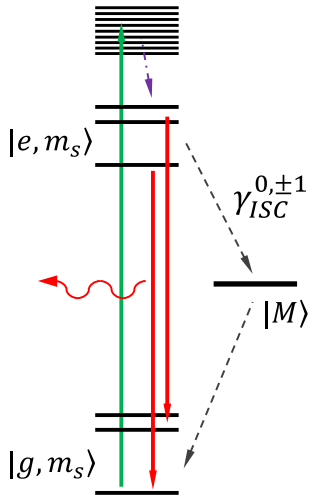

Figure 2.3: Energy level scheme of SiV and GeV in diamond and V_B in hBN. The green arrow denotes excitation via photon absorption from any of the ground triplet states to their corresponding triplet excited state (spin conserving). The purple arrow indicates decay from the conduction band before photon emission. The red arrows indicate relaxation via photon emission from excited states to ground states. The blacked dashed arrows indicate non-radiative (in the range being measured) decay to the metastable state from any of $|e, m_s = 0, \pm1\rangle$ to the ground state, preferentially $m_s = 0$.

2.1.1 Nanothermometry

In this section we will focus on the use of vacancy centers as nanoscale thermometers by exploiting the temperature-dependent nature of their optical emissions. The related results in this work are based off SiVs and GeVs in nanodiamonds and hence will be the focus of this section. For a broader discussion of the overall energy level scheme see section 2.1.

Excitation Paths

There are two main excitation paths taken, Stokes and anti-Stokes, depicted in *Figure 2.4*. Stokes excitation involves the absorption of a higher energy photon, with respect to the Zero Phonon Line (ZPL), transitioning from the lowest vibronic state in the ground

electronic state, to a higher vibronic state in the electronic excited state. This 'excess' energy is then lost to phonon emission, transitioning to the lowest vibronic state. Electronic relaxation then occurs, resulting in the emission of a photon with a lower energy than what was originally absorbed. Anti-Stokes excitation instead requires the absorption of phonon(s) in addition to a photon to make the jump to the excited electronic state. First phonon(s) are absorbed, transitioning to the first vibronic state in the ground electronic state, then a lower energy photon (with respect to the ZPL) is absorbed making the transition to the lowest vibronic state in the excited electronic state. Electronic relaxation then occurs, resulting in an emitted photon of higher energy than that of the absorbed light.[68]

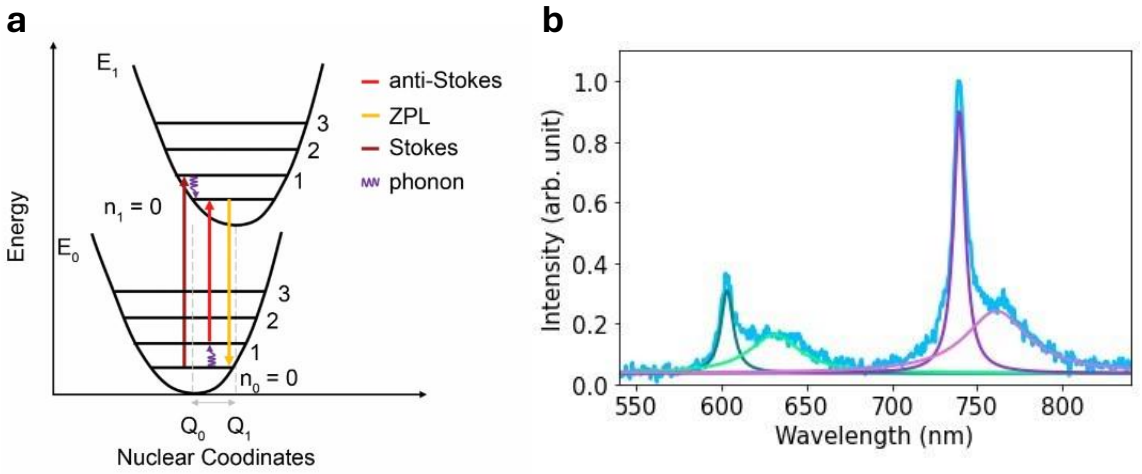

Figure 2.4: Energy level schemes and resulting emission spectrum of colour centers. **a)** Cartoon representation of the electronic and vibrational energy levels in samples with vacancy centers. E_0 and E_1 represent the ground and first excited state respectively. The levels $n_{0,1}$ represent the vibrational levels within the two electronic levels. The Stokes excitation involves the absorption of a higher energy photon, with respect to the ZPL emission, with the excess energy being lost to phonon emission. The anti-Stokes excitation involves the absorption of phonon(s) to meet the

energy requirements to jump to the first electronic excited state, and hence absorb a photon of lower energy compared to the ZPL emission. [Reprinted with permission from Toan Trong Tran et. al., "Anti-Stokes excitation of solid-state quantum emitters for nanoscale thermometry", Science Advances, (2019)]. **b)** Example emission spectrum of a typical nanodiamond with two vacancy centers (ex: SiV and GeV, shown in purple and green respectively). The two narrow spikes (dark green and purple) are the ZPLs as shown in **(a)**. The two wider peaks (lighter green and purple) are known as the phonon side bands (PSB) and result from emissions from transitions between various vibronic states. [Reprinted with permission from Dylan G. Stone et. al., "Diamond Nanothermometry Using a Machine Learning Approach", ACS Optical Materials, (2023)].

The Franck-Condon principle plays an important role in the process of these various transition paths. According to this principle, with respect to the nuclei of the lattice, the absorption of a photon and subsequent electronic transition happens instantaneously, meaning that the nuclei have no time to adjust their positions. Therefore, the probability of transition relates to the overlap of the wavefunctions in each vibrational level for fixed nuclear positions. This restriction of wavefunction overlap gives us preferential transitions for absorption. For example, it is far more likely that the Stokes line (*Figure 2.4*) will land on the second vibronic state ($n_1 = 1$) than the first ($n_1 = 0$). After excitation, the nuclei readjust their positions creating vibrations in the lattice in the form of emitted phonons.[18]

Temperature Dependence

The key to utilizing these nanodiamonds as nanoscale thermometers is their spectral dependence on the temperature of the diamond and, due to diamond's ability to transfer

thermal energy efficiently, the temperature of the local environment. As shown in *Figure 2.5*, the spectrum of a vacancy-containing nanodiamond varies with the temperature of the diamond. The key parameters that are monitored during these temperature variations are the position, full width at half maximum (FWHM), and amplitude of both the ZPL (main spike), and phonon sideband (PSB) (wider peak).[7]

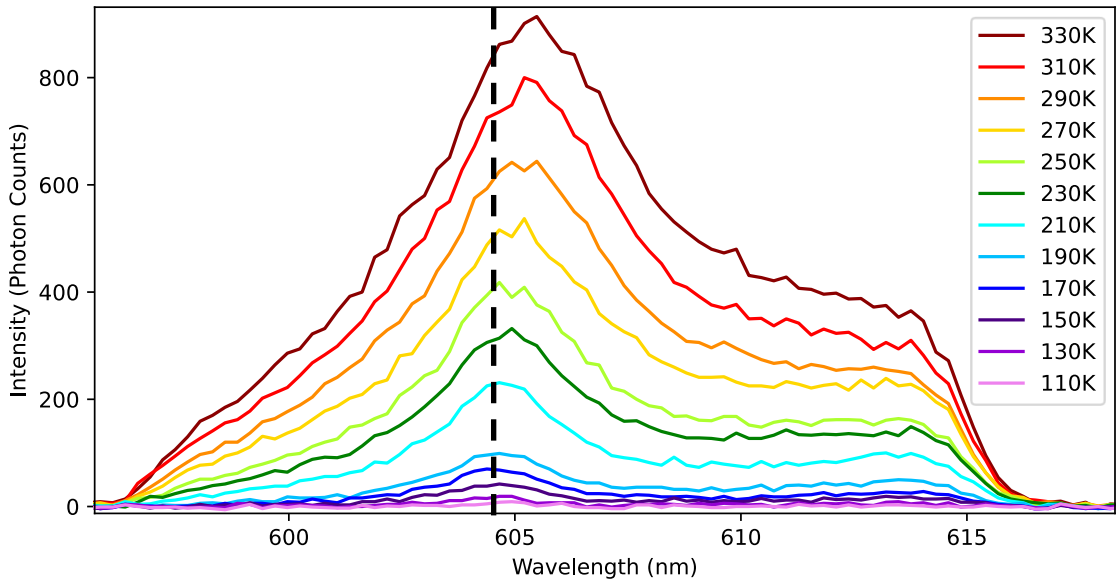

Figure 2.5: Temperature dependence of diamond colour center. This is an example image to help visualize the temperature dependence of the diamond colour center's emission spectrum. The black dashed line is added for visual aid. As the temperature increases from 110K to 300K, the spectrum shifts to the right (change in ZPL), the amplitude increases, and the FWHM increases. (This particular image is of a GeV in diamond, anti-Stokes excitation via a 637nm laser).

In this work, we specifically looked at samples containing two different vacancies which were simultaneously excited via the Stokes process, giving a total of twelve physical quantities. Stokes excitation was used in this case as we were able to achieve a better signal, however anti-Stokes excitation scales exponentially with temperature and can be an attractive option depending on the environment being probed.[68]

2.1.2 Optically Detected Magnetic Resonance in Colour Centers

In this section we will discuss a technique known as *optically detected magnetic resonance* (ODMR) which is a spin-based quantum sensing technique used for a range of applications such as measuring nanoscale electric and magnetic fields, strain and pressure, temperature, and even individual electron and nuclear spins. ODMR is applicable for a wide range of colour centers, including those found in both nanodiamonds and hBN. However, the results gathered in this study focused on hBN samples with Boron vacancies (V_B), therefore they will be the example used in this discussion.[15]

As mentioned in detail in section 2.1, the electronic structure of hBN consists of a triplet ground and excited state, as well as a singlet metastable state. Direct transitions between the sublevels in the ground and excited states are spin conserving, but ISC to the metastable state is not. Furthermore, electrons in the excited $m_s = \pm 1$ sublevel are far more likely to take the ISC path than the $m_s = 0$ sublevel. Finally, decay from the metastable state to the ground state preferentially occurs to the ground $m_s = 0$ sublevel, meaning the system can be optically pumped such that (mostly) all spin states are $m_s = 0$, resulting in higher photoluminescence (PL) counts since ISC transitions are non-radiative. These three factors are the main mechanisms ODMR uses to extract information from the samples and constitute the 'OD' part of the name.[15,69]

The latter half of the name, Magnetic Resonance, can then be used to selectively force the emitters into the $m_s = \pm 1$ sublevels. Typically with ODMR measurements, the aim is to identify the energy gaps between the $m_s = 0$ and ± 1 states. This is done by applying a microwave (MW) field to the sample that sweeps a predefined range of frequencies. Since

the transition involving ISC is non-radiative, when the MW field is resonant with the transition frequencies, a dip in the total PL counts is observed. Depending on if the $m_s = \pm 1$ states are degenerate or not, there will be one or two dips in the PL counts that indicate the frequency gaps between these sublevels and the lower $m_s = 0$ sublevel.

Figure 2.6 shows an example of an ODMR spectrum for a nitrogen vacancy (NV) center in diamond with an applied magnetic field. The applied field causes Zeeman splitting, raising the degeneracy of the $m_s = \pm 1$ and changing the relative energy gaps between them and the $m_s = 0$ sublevel. This shift in energy gap changes the resonance frequency, which appears as an increasing separation in the dips seen in the figure.

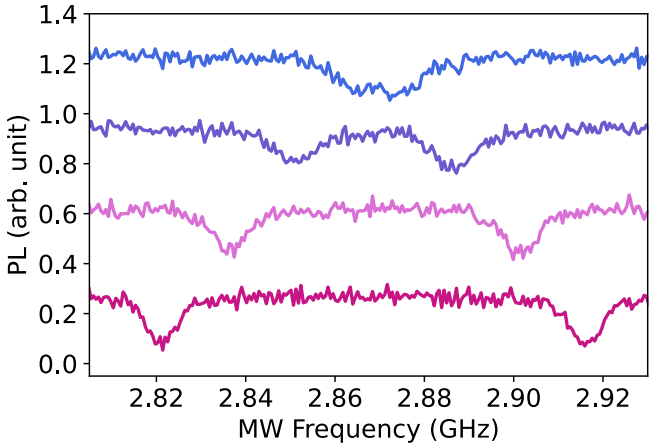

Figure 2.6: Applied magnetic field influence on ODMR peak separation. This is an example ODMR spectra of a room temperature NV center in diamond under the effect of an applied magnetic field at different strengths. The characteristic ODMR dips can be seen when the MW field is on resonance with the transition energy between sublevels. The application of a magnetic field causes Zeeman splitting, shifting the dips further apart as the field strength is increased (see main text for detail). [Reprinted with permission from Dylan G. Stone et. al., “Fast

characterization of optically detected magnetic resonance spectra via data clustering", ACS Journal of Physical Chemistry C, (2024)].

2.1.3 Autocorrelation

In this section we will discuss the Autocorrelation technique, which measures the second order autocorrelation, $g^{(2)}(t)$, over a predefined period of time. For this work the technique was employed to determine if an optically active site in our material was a single colour center or a cluster of many centers.

Figure 2.7 shows an example of an Autocorrelation measurement for an optically active region in an hBN sample. The inset depicts a simplified lab setup for taking such a measurement known as the Hanbury-Brown-Twiss autocorrelator. The light emitted from the active source is passed through a 50/50 beam splitter and sent to one of two detectors. When a detector receives a photon, a timer is started and continues counting until the second detector receives a photon. This time delay is then plotted in a histogram of counts versus time delay, with each detection being added to its corresponding time bin. Practically speaking the detectors must be arbitrarily labelled as 'start' and 'stop', for this reason negative time can be counted if the 'stop' detector receives a photon first.[61]

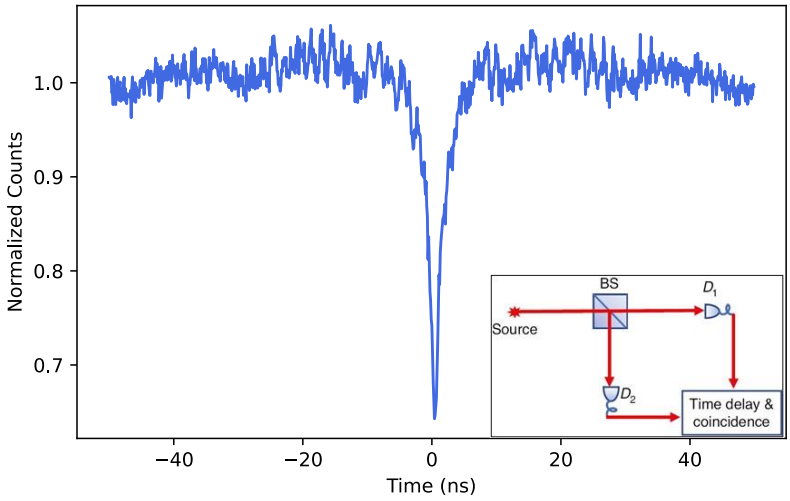

Figure 2.7: Autocorrelation plot with a simplified lab diagram inset. The example plot is a histogram with 1000 time bins. A count is added to a bin when the difference in time between detections of the start and stop detector fall within that time window. The inset depicts a simplified lab setup, where the light from a photon source (i.e. our colour center) is separated by a 50/50 beam splitter and sent to the two detectors. One detector is arbitrarily chosen to be the 'start' and the other the 'stop'. In this context, a negative time indicates that the 'stop' detector received a photon before the 'start' detector.

The main point of interest in such a plot is the dip at zero time. For a photon to be emitted, the colour center must be excited and then allowed to relax, which takes a non-zero amount of time to occur. Therefore, if the active site contains only a single emitter, in an ideal measurement we would expect there to be zero counts in the smallest time bin, as it is physically impossible for the emitter to release two photons simultaneously. If there is a non-zero number of counts in the lowest bin, then there must be multiple emitters present. The contrast of the dip gives some information about the relative number of emitters present. For example, if a few emitters are present, the dip may be significantly less deep but still easily visible with long integration times, whereas a large cluster of

emitters would produce little to no visible dip. Practically speaking, there is always some amount of noise present, so a threshold must be chosen to differentiate between one or more emitters. Typically, a value of 0.5 is chosen as the threshold for single emitters.[61,70]

2.2 Machine Learning

2.2.1 General Overview

This section gives an overview of the main machine learning (ML) methods relevant to this work. This section will outline the major categories of machine learning algorithms, with the following subsections giving a general idea of the function of each of the methods used. The specific use cases will be discussed in more detail in section 4.

Machine learning, sometimes referred to as automated learning, is a subfield of the broader field of Artificial Intelligence. The goal of ML is to develop a collection of algorithms which can learn from experience (i.e. past data) in order to make informed predictions about, or classify, unseen data. The exact mechanisms vary between algorithms, but in general involve the learning of hidden parameters that map input data to a given output or classification.[71–74]

There are two main types of predictive models in ML, *classification* and *regression* algorithms. Classification focuses on determining the membership of a given item to a specific group or *class*. A well-known example is that of image recognition using the famous MNIST dataset, which contains 70,000 images of handwritten numbers between zero and nine, and their *label*, i.e. what number they are. The goal of the algorithm in this case would be to classify each image by assigning the dataset a single label from the list of classes, zero through nine.[75,76]

Regression algorithms on the other hand focus on determining a numerical value, typically output value(s), from input value(s). They do this by learning the mapping from input(s) to output(s) within a specified level of accuracy. Statistical estimators such as standard deviation from the mean or variance are common ways to define the accuracy of a given mapping. [73,75]

Aside from the type of prediction made, ML approaches are often divided into three main categories: *supervised*, *unsupervised*, and *reinforcement* learners. Each of these categories describe the general process by which a given algorithm learns from the data provided and makes predictions. This work will focus on the differences between supervised and unsupervised learning, as the models discussed later on fit within these categories.[71,73] For a general description on reinforced learning, see *Machine and quantum learning for diamond-based quantum applications*.[73]

Supervised Learning

Supervised learners use a learning-by-example strategy, where the algorithm is provided with a *training set* which contains some number of labelled points $\{(\boldsymbol{x}_i, y_i)\}$. Here, $\boldsymbol{x}_i$ is an n-dimensional vector containing the data inputs, which are referred to as *features,* and y_i is the corresponding output or *label*. During training, both the features and labels are known, and the algorithm works to find the mapping from one to the other, such that a label, y_i, can be predicted from given feature(s), $\boldsymbol{x}_i$. To validate the training of the algorithm, a subset of the data is reserved for testing, referred to as a *testing set*. This set is one which the model in question has not seen before, and similarly contains a number of labelled points. This time, only the features from the set are provided, and the algorithm must make predictions on the correct labelling of each of the feature(s). Then,

the known labels can be compared to the predictions to quantify the accuracy of the model and whether further training is needed.[71,73,74] Two such examples of supervised learners are discussed in this work, *Multi-Feature Linear Regression*, and *Artificial Neural Networks* (sometimes simply referred to as *Neural Networks*).

Unsupervised Learning

Unsupervised Learning on the other hand focuses on finding underlying patterns and structure in the data provided. Unsupervised learners are provided only with the features and some form of rule defined by a user. These types of algorithms can be suitable for creating labels in unlabelled data, such that supervised learners can be implemented and trained on said labels. A common example of such learners are clustering algorithms. Data is grouped into clusters based on parameters such as minimizing the mean distance between the cluster and its members.[71,73] One such example, *Kmeans++*, will be discussed in this work.

For the following sub-chapters, specific variables may be used for the general equations discussed to allow for an easier comparison to their specific use in later chapters. Variables will be chosen such that they pair nicely with the corresponding chapters that will reference them.

2.2.2 Multi-Feature Linear Regression

Multi-Feature Linear Regression (MF-LR), sometimes referred to as multiple linear regression, is a supervised learning technique that predicts a label based on a weighted sum of features. Formally, the data consists of instances, i.e. feature-label pairs $(\boldsymbol{x}_i, T_i)$, $i = 1, 2, 3, \dots, N$, where $\boldsymbol{x}_i \in \mathbb{R}^n$, $T_i \in \mathbb{R}$ and N is the number of datasets or feature-label

pairs. The goal is to find the function $\tau(\boldsymbol{x})$ that best maps the features to the labels, while minimizing the cost function. The predicted outputs are given by:

$$\tau_i(\boldsymbol{x_i}) = w_{i1}x_{i1} + w_{i2}x_{i2} + w_{i3}x_{i3} + \cdots w_{in}x_{in} + b_i \tag{2.1}$$

Where i refers to the i^{th} prediction, $\boldsymbol{w}$ refers to the weights, and b is the bias. This formula is often simplified to:

$$\tau_i(\boldsymbol{x_i}) = \sum_{j=1}^{n} w_{ij}x_{ij} \tag{2.2}$$

Where $w_{i0} = b_i$ and $x_{i0} = 1$. This is the generalized form of the simpler slope-intercept Linear Regression (LR) where the number of features, n, is greater than 1. [7,73,77]

A common cost function for LR is the least squares error (LSE) cost function, given by:

$$LSE = \sum_{i=1}^{N} (\tau_i - T_i)^2 \tag{2.3}$$

Where τ_i are the predicted values, and T_i are the known target values, and i is once again a given data set or feature-label pair.[7,73,77] The weights $\boldsymbol{w}_i$ are determined by minimizing equation 2.3 using algorithms such as the *gradient descent* algorithm.[78]

2.2.3 Artificial Neural Networks

Artificial Neural Networks (ANNs) or *Neural Networks* (NNs) for short, are supervised learning techniques that are designed to resemble the structure of the neurons in the human brain, and the complex connections they form between one another. The network is made up of layers, each of which contain multiple artificial neurons or nodes, which each accept an input and produce an output. The general structure of an NN begins with

an input layer, where the nodes receive the features directly, and ends with an output layer where final predictions or classifications are returned. Between these layers are the so-called *hidden layers*, which are responsible for the data processing and come in many forms. The types of hidden layers present determine the type of NN (**Figure *2.8***).[73,79]

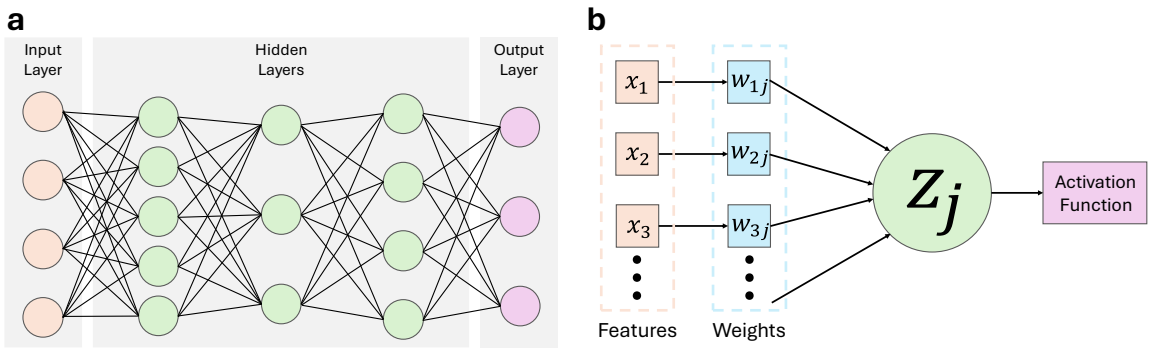

Figure 2.8: Visualization of Artificial Neural Network Structure. **a)** A cartoon depiction of the general structure of an ANN where each node in a given layer connects to each node in the following layer. Features are passed to the input layer, processed in the hidden layers, and their corresponding predictions are returned at the output. Note that each of the layers can have any number of nodes in them, the number of nodes were chosen at random. **b)** A more detailed view of the structure for a single node, depicting the weighted sum of the outputs from a previous layer's node (see 2.4).

In an NN each node is connected to every node in the next following layer through an associated weight. Similar to the MF-LR algorithm, the weights are determined by training the algorithm with known feature-label pairs or datasets. Formally, the input for a given node is the weighted sum of all the outputs from the nodes in the previous layer, which is given by:

$$z_j = \sum_i w_{ij} x_i \qquad (2.4)$$

Where z_j is the input for the j^{th} node on the current layer, x_i are the outputs of each of the nodes from the previous layer, and w_{ij} are the corresponding weights for each of the connections between the previous layer's nodes and the j^{th} node in the current layer (see *Figure 2.8*). [73,79]

Before the output of the j^{th} node is passed on to the nodes in the next layer, the output z_j is passed through an *activation function*. There is a large selection of available activation functions with some of the most common being the *Rectified Linear Unit* (ReLU), and sigmoid functions, such as the hyperbolic tangent and the logistic functions.[79,80]

At the output layer, a loss function, such as the LSE (2.3) is used to determine the error in the network's prediction compared to the true (known) value from each feature-label pair. The weights are then updated by computing the gradient of the loss function with respect to each of the weights such that the overall loss is minimized. This updating process continues backwards through the network in a process known as *backpropagation*. The methods for updating the weights, computing the loss, and even the general structure of the network and types of layers included can vary greatly from network to network depending on the application, however these details go beyond the scope of this research.[73,79–81]

2.2.4 K-means++

K-means++ is an unsupervised clustering technique which clusters data into k groups. K-means++ is a modified version of the more general *k-means* clustering algorithm, often referred to as *Lloyd's algorithm*, where extra steps are taken when choosing the initial cluster centers to help avoid converging to local minima.[82,83]

The ultimate goal of the k-means algorithm is to minimize the distance between the features, $\boldsymbol{x} = [x_1, x_2, \dots, x_n]$, and their assigned cluster's centre or *centroid*, $C = \{c_1, c_2, \dots, c_k\}$ by modifying the values in C. The formula is given by:

$$\phi = \sum_{j=1}^{n} \min_{c \in C} (|x_i - c|^2) \tag{2.5}$$

Where ϕ is known as the *inertia* or *within-cluster sum-of squares*.[82,83]

There are three main steps in the k-means algorithm:

1. Choose the initial k centroids. This is typically done by randomly selecting points from $\boldsymbol{x}$.
2. For each centroid c_i, let C_i be all values in $\boldsymbol{x}$ that are closer to c_i than they are to c_j, for all $j \neq i$.
3. For all $c \in C$, set c_i to equal the mean of all points assigned to C_i from step 2. This is given by $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$.

The difference between the previous and current centroids is computed to monitor its change. Steps two and three are repeated until the centroids stop changing, or more practically, when the change in the centroids position drops below a predefined threshold.[82,83]

This process will always converge, however depending on the initial centroids this may be to a local minimum. To help address this issue, k-means++ modifies the initialization stage to make the centroids more distant from one another.[82,83] This is done by using what is known as D^2 *weighting*, see ref [82] for more detail.

2.3 Traditional Fitting

2.3.1 General Overview

Since the focus of this work is to compare standard fitting approaches to that of machine learning (ML) methods, we will focus on highlighting the subtle difference between them. We will use Multi-Feature Linear Regression (MF-LR) as our ML model to compare to, as at first glance it is not obvious how this technique differs from a more traditional curve fitting approach such as Levenberg-Marquardt fitting.

Traditional fitting techniques rely on predefined, project-specific, models with unknown parameters that must be found, in order to fit the model to a given dataset. Such models typically require *complete data*, i.e. sets of data that are large enough to allow the model to converge to a result within some target level of accuracy. In Contrast, ML models make predictions using general-purpose learning methods without the need for a specified model. These general-purpose models aim to find patterns in anything from *sparse data* (sets of data too small for traditional fitting to converge) to large unwieldy data.[70,84]

In this study we focus on the use of the *Levenberg-Marquardt algorithm* (L-M), as well as a modified version of L-M known as the *Trust Region Reflective Algorithm*.[85]

2.3.2 Levenberg-Marquardt Algorithm (Trust Region Reflective Algorithm)

The well known *Levenberg-Marquardt algorithm* (L-M) is a statistical inferencing technique which fits a predefined model, $f(x, \boldsymbol{\beta})$, to a given set of data, (x_i, y_i) pairs, using *Nonlinear Least Squares Minimization*. The function being minimized is given by:

$$F(\boldsymbol{\beta}) = \sum_{i=1}^{n} |y_i - f(x_i, \boldsymbol{\beta})|^2 \qquad (2.6)$$

Where $f(x_i, \boldsymbol{\beta})$ is the model being fit to the data, and $\boldsymbol{\beta}$ is a vector containing the parameters of the given model. For example, in a 2D linear model $\boldsymbol{\beta}$ would contain the slope and intercept of the line.[86,87]

In this work, the Python library SciPy Optimize was used to implement the L-M fitting. In cases where boundaries were given for the various parameters being fit, the library uses a modified version of L-M fitting known as *Trust Region Reflective Algorithm* ('trf'). For the purposes of this work, the main difference is the 'Reflected Boundaries' which constrain the parameters to the boundaries given. If the parameters leave this boundary, they are *reflected* back into the predefined limits. See the SciPy documentation for more details.[85]

3 Methods

3.1 Generalized Accuracy, Resolution, and Sensitivity

This section outlines the definitions of the performance metrics used to compare the nanothermometry results in section 4.1, as well as the considerations made when designing our testing methodology. Our characterization is based on a series of key elements:[7]

i. In order to make a relative comparison between our multifeatured model and all other established single feature models, we used our own measurements rather than literature values. This was to ensure that the comparison was independent of a particular measurement system's detection efficiency, signal-to-noise ration, resolution, etc.

ii. The values reported for accuracy, resolution, and sensitivity in section 4.1 are what we refer to as *generalized* values. This approach follows a few key steps. First, we build each model using known calibration (training) data consisting of feature-label pairs. Second, we feed each model its corresponding feature(s) from unseen testing data and get in return its prediction for the label (in our case the temperature). Using this we determine with what accuracy, resolution, and sensitivity the model can predict the true label (true temperature) of the sample (see iv). This approach differs from traditional ones because these generalized figures of merit are not estimated from the data used to determine the models themselves (training sets), but rather from the data they are tested on.

iii. As we explain in section 4.1, for each model we find $A_{N=1}$, $R_{N=1}$, $S_{N=1}$, $A_{N=5}$, $R_{N=5}$, and $S_{N=5}$. In this case $N = 1$ refers to the models that were trained on one nanodiamond (ND) and tested on the remaining. In our study six different samples were used. The values shown in *Figure 4.3* are averages of all combinations of 1-training/5-testing. $N = 5$ refers to models that were trained on five NDs and tested on the remaining ND. Again, the results displayed in the figure are averages, this time of all 5-training/1-testing combinations.

iv. While determining the accuracy, resolution, and sensitivity of each model, we considered the true temperatures to be that which is measured by the cryostat. A more rigorous, yet impractical, definition would require absolute knowledge of the temperature of the samples rather than that from a reference instrument.

Accuracy

Accuracy is defined as the absolute difference between the measured (average) value and the "true" value of our observable, in this case temperature. The "true" value is the value measured by the cryostat, while the measured value is the value predicted by the given model as described in ii).

Resolution

Resolution in nanothermometry is generally defined as the standard deviation of the measured observable, σ, multiplied by the square root of the measurement integration time, t_m. In our case the resolution comes from the standard deviation in the predicted values as per points ii) and iii). For convenience our integration times were $t_m = 1s$, so the resolution is simply σ.

Sensitivity

Relative sensitivity is often defined as $|(\partial O/\partial T)/O|$, where O is the measured observable and T is our temperature. Sensitivities were calculated as per point iii) above. For the multifeatured model, which uses multiple observables to make its prediction, a weighted linear combination of single feature sensitivities was used. In this case our model had up to $n = 5$ single features x_{ij}. The sensitivity of such a model for each i^{th} dataset is described as:

$$S_{MFLR_i} = \sum_{j=0}^{n} \alpha_{ij} \frac{\partial x_{ij}/\partial T}{x_{ij}} \tag{3.1}$$

Where $s_{ij} = (\partial x_{ij}/\partial T/x_{ij})$ is the sensitivity for an individual feature, which are weighted by the coefficient α_{ij}. This coefficient is determined by the coefficients in equations 2.1 or 2.2 via normalization:

$$\alpha_{ij} = \frac{w_{ij}}{\sum_{j=1}^{n} w_{ij}} \tag{3.2}$$

Here the index j runs over the number of features used in the multifeatured model, while i indicates the dataset for models that train on multiple samples (as per point iii).

3.2 Standard Fitting – ODMR

For the ODMR portion of this study, the main point of comparison for our ML models was a standard fit of an ODMR spectrum, modelling each of the peaks as Lorentzian functions. This was implemented using the function curve_fit() from the scipy.optimize() library.[88] The Lorentzian function used was:

$$L(x) = \frac{\mathrm{ab}^2}{\mathrm{b}^2 + (2\mathrm{x} - 2\mathrm{c})^2} \tag{3.3}$$

Where a is the amplitude, b is the full width at half maximum (FWHM), c is the peak position, and x, in our case, is the frequency in MHz. The parameters are determined using nonlinear least squares minimization[89] similar to that of equation 2.3.

To fit the (normalized) ODMR spectrum, we subtracted two Lorentzian functions from one:

$$ODMR\ Fit = 1 - \frac{\mathrm{ab}^2}{\mathrm{b}^2 + (2\mathrm{x} - 2\mathrm{c})^2} - \frac{\alpha\beta^2}{\beta^2 + (2\mathrm{x} - 2\gamma)^2} \tag{3.4}$$

Where the Greek letters α, β, γ are the amplitude, FWHM, and peak position of the second peak.

3.3 Synthetic ODMR Data

In order to have a large set of data to work with in this study, synthetic ODMR data was generated using Python and some of its libraries (see Appendix B.2). This data was also crucial to precisely quantify the accuracy of the models used as we know with complete certainty what the true parameters of the datasets are. The available real experimental data was used to determine an appropriate range of values for each of the key parameters used to generate the data. The data generation involves several formulas as well as NumPy's binomial function to simulate a real experiment. The first of these formulas determines the probability of a successful count for a specific frequency value:

$$S_i(f) = \frac{p_i}{1 + \left(\frac{f - x_{0i}}{w_i}\right)^2} \tag{3.5}$$

Where $S(f)$ is the probability of success at the frequency value f (MHz), p_i is the probability of success at resonance (i.e. the contrast for the peak), x_{0i} is the location of the peak (MHz, resonance point), and w_i is the width of the peak (MHz). Since the ODMR spectra in question have two peaks, that are each Lorentzian in shape, this calculation is done for each peak, where $i = 1,2$ denotes peaks 1 and 2.

Next, the probability $S_i(f)$ is used in NumPy's binomial function to determine the number of successful counts that are "detected" for that frequency bin:

$$counts_i = np.binomial(n_t, S_i(f), 1)[0] \tag{3.6}$$

Where n_t refers to the number of tries per frequency bin. This equation is written in the format of the Python code, the 1 refers to the shape of the function output (i.e. how many values are returned), and the [0] ensures that we are only assigning the value returned to the $counts_i$ variable.

These counts per frequency bin are then collected into an array:

$$s_i = [counts_{i,0}, counts_{i,1}, \dots, counts_{i,nb}\,] \tag{3.7}$$

Where n_b is the number of bins that the frequency range is divided into.

Finally, the simulated ODMR spectrum is calculated using the following:

$$Simulated\ ODMR\ Spectrum = \frac{n_t - (s_1 - s_2)}{n_t} \tag{3.8}$$

You can see with this process that, as with a real experiment, each frequency bin is probed one at a time with a given probability of a successful count being detected. As with a real experiment, as the frequency approaches the resonance values, the number of counts per bin declines producing the signature dips in measured counts associated with ODMR (see section 2.1.2 for a description of this process).

Table 3.1: Parameter ranges used to simulate the synthetic ODMR data used in this study

Parameter	**Variable**	**Value/Range**	**Variance between Peaks**	**Units**
Frequency	f	[3000, 4000]	-	MHz
Number of bins	n_b	199	-	-
Peak location (resonance point) ($i = 1\ or\ 2$)	x_{0i}	[3200, 3800]	-	MHz
Peak width ($i = 1\ or\ 2$)	w_i	[30, 130]	Up to +5%	MHz
Probability of success at resonance ($i = 1\ or\ 2$)	p_i	[0.03, 0.06]	Up to +65%	-

3.4 Synthetic Autocorrelation Data

Similar to the ODMR data, Autocorrelation data was generated with the intent on comparing ML models to that of standard fitting approaches. By using synthetically generated data, instead of real experimental data, the parameters of data can be known with perfect precision, allowing for precise measurements of accuracy. The foundation of this simulation comes from the works of Kudyshev et. al., specifically the supplemental information from the paper *Rapid Classification of Quantum Sources Enabled by Machine Learning*.[61]

The process of creating the synthetic data follows that of a real experiment, where each data point is added to its corresponding time-delay bin, one at a time (see section 2.1.3 for details on autocorrelation measurements). Whether or not a measurement occurs depends on a probability function that aims to model the probability of measuring a second count after a given time delay. The probability functions and their parameters were taken from ref [61].

For the first time bin, the probability is given as:

$$P_{n_0} = Rr + R(1-r)\left[1 - (1+a)e^{-\frac{1}{4}\delta t \lambda_1} + ae^{-\frac{1}{4}\delta t \lambda_2}\right] \tag{3.9}$$

Where $R = (20N_{bins})^{-1}$ is the average co-detection rate per bin, $r = 1 - \sqrt{1 - g^{(2)}(0)}$ is the fraction of photonic background in the total emission, and δt is the bin width. The remaining variables $\lambda_1 = \gamma_{EG} + \gamma_{GE}$, $\lambda_2 = \gamma_{MG} + \frac{\gamma_{EM}\gamma_{GE}}{\lambda_1}$, and $a = \frac{\lambda_2}{\gamma_{MG}} - 1$ are combinations of decay rates for various electronic transitions (see *Table 3.2*).

For the remaining bins, the following probability function is used:

$$P_n = Rr + R(1-r)\left[1 - (1+a)e^{-\left|n-n_0+\frac{1}{2}\right|\delta t\lambda_1} + ae^{-\left|n-n_0+\frac{1}{2}\right|\delta t\lambda_2}\right] \qquad (3.10)$$

Where n is the bin number.

In order to simulate the data, the function is given a $g^{(2)}(t=0)$ value along with an integration time. It then computes the probability for a second count in the first bin and compares that to a randomly generated value between 0 and 1, if the value is less than the probability a detection occurred and we add one to the first bin, otherwise we move onto the next bin. The process is repeated until either (a) a detection occurs or (b), a preset maximum time delay limit is reached (i.e the max value on the horizontal axis). In both instances, the bin number is reset (i.e. the 'counter' starts over) and we repeat the process. This occurs continuously until the overall time reaches the predefined integration time for the 'measurement'. For more details, see Appendix B.3.

Table 3.2: Parameters used to simulate Autocorrelation datasets (taken from ref [61])

Parameter	**Variable**	**Value**	**Units**
Bin width	δt	2.34	ns
Number of bins	N_{bins}	215	-
Radiative excited-state decay rate	γ_{EG}	20	MHz
Excitation rate	γ_{GE}	20	MHz
Non-radiative shelving rate	γ_{EM}	10	MHz
Non-radiative de-shelving rate	γ_{MG}	7	MHz

3.5 Custom Clustering Algorithm

For part of this work we developed a custom-written *Clustering Algorithm* (CA) in order to analyze optically detected magnetic resonance (ODMR) spectra that is based on k-

means clustering and uses the corresponding function from the open source machine learning library scikit-learn as its foundation.[90] The goal of this algorithm is to determine the two microwave (MW) resonance values by locating their corresponding peaks in an ODMR spectrum. The algorithm involves two main steps:

1) 'Vertically' clustering the 1D array containing photoluminescence (PL) counts into k_v *rows*.
2) 'Horizontally' clustering the 1D array containing the microwave (MW) frequency data into two *columns*.

In the first step, the number of clusters must be provided ahead of time to the k-means algorithm. The optimal number of clusters is chosen using a custom, quantitative version of the *elbow method*, where the so-called inertia (i.e. the sum of squared distances of samples to their closest cluster center, weighted by the sample weights[90]) is evaluated and plotted against the number of clusters chosen (see section 3.6 for more details). For our datasets, the optimal number of 'vertical' clusters, which we will refer to as *rows*, was always $k_v = 4$, and hence this was used for each dataset we analysed. After this vertical clustering is complete, the lowest row in the previous step is clustered once again into $k_h = 2$ 'horizontal' clusters, which we will refer to as *columns*, this time using the MW frequency data. The centroid of these two columns is then extracted and used as the prediction for the two peaks. *Figure 3.1* gives a visual representation of this process.[15]

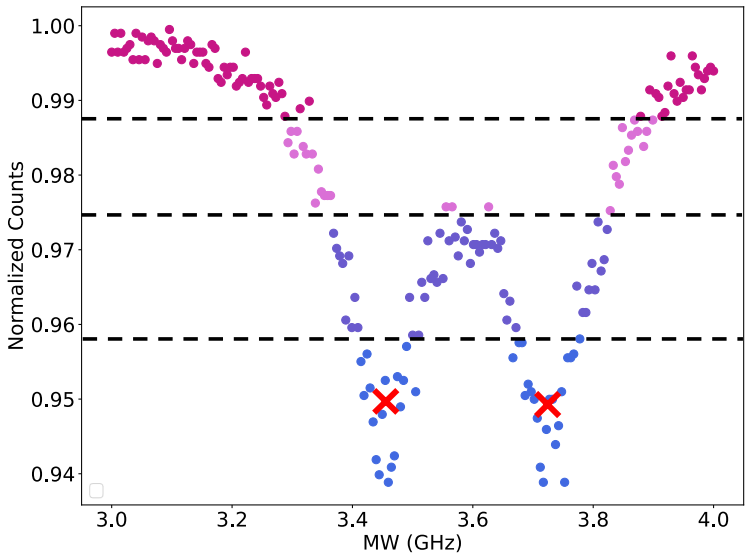

Figure 3.1: A qualitative depiction of the custom CA. The superimposed horizontal dashed lines were added to clearly show the distinct rows the data was separated into after the first step, represented by the different coloured points. In this instance, the blue points in the lowest row would be passed to the second step, where they are clustered into two columns. The superimposed red X's were added to represent the location of the centroids of the two columns, which would be used as the peak predictions (see main text for more detail).

Additionally, there are also a set of *conditions* that help the second step correctly identify which rows should be included in the second round of clustering depending on the spectrum's structure. There were three types of data explored in this study: **a)** clearly separated peaks with a difference in PL contrast of up to 65% ('asymmetric' data), **b)** clearly separated peaks with a difference in PL contrast of up to 15% ('symmetric' data), and **c)** overlapping peaks. Examples of each of these cases are shown in *Figure 3.2*.[15]

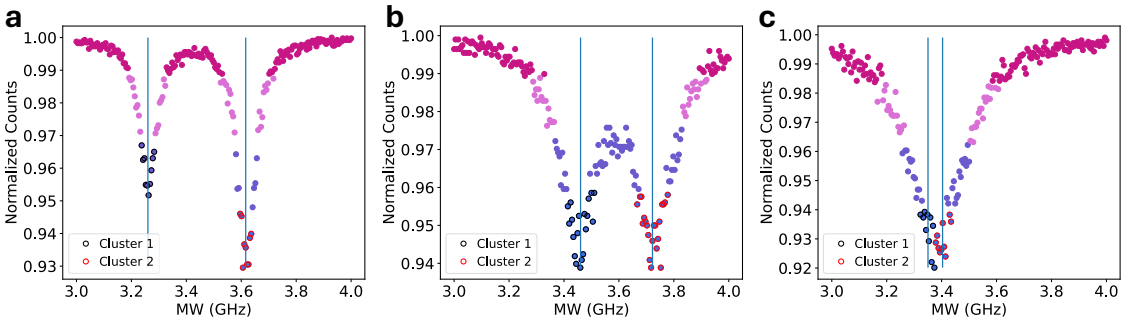

Figure 3.2: Examples of the three possible ODMR data archetypes considered in this work. **a)** PL contrast of up to 65% (asymmetric). **b)** PL contrast of up to 15% (symmetric). **c)** partial-to-fully overlapping peaks. The black and red circles labelled as 'Cluster 1' and 'Cluster 2' are the points that got passed onto the second step in the CA (see main text for detail). [Reprinted with permission from Dylan G. Stone et. al., "Fast characterization of optically detected magnetic resonance spectra via data clustering", ACS Journal of Physical Chemistry C, (2024)].

To differentiate between the three archetypes, two *conditions* were implemented to check the structure of the data. *Condition 1* compares two quantities, the change in the column width (in this case the average of both column widths) and the change in the distance between the column centres (the distance between cluster centroids) when transitioning from including i) only the lowest row, to ii) the lowest two rows in the second horizontal clustering step. This condition was designed specifically to differentiate between *Figure 3.2a* and *Figure 3.2b*, where including only the lowest row in the second step would entirely miss one of the two peaks present. The rationale is that if both peaks were present in the lowest row, we would expect that including the second lowest row would somewhat equally increase the width of the columns, as well as the distance between them, since the peaks widen asymmetrically. Conversely, if one of the peaks was not present in the lowest row, including the second lowest row should dramatically change the distance between the column centres with relatively minimal change to the column

widths. *Figure 3.3* shows a comparison between capturing and missing the second peak depending on which rows are included.[15]

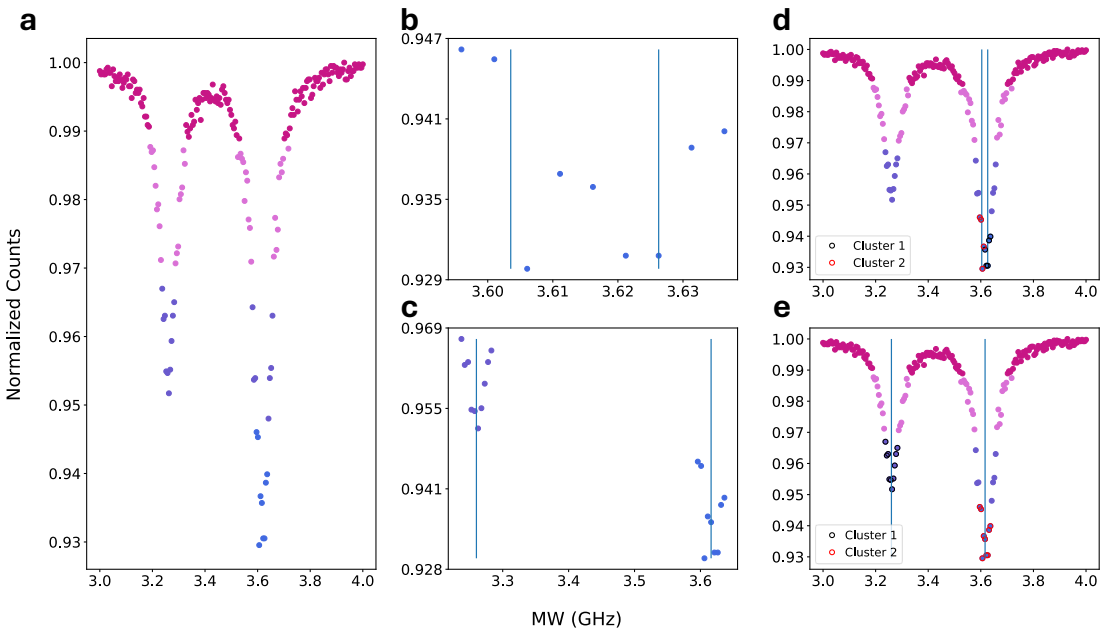

Figure 3.3: Visualization of the process to select the correct peaks. **a)** The raw data is colour coded; each colour indicates a different row as identified by the first clustering step. **b-c)** Subset of data that is passed to step two not using **(b)** and using **(c)** the conditions, respectively. **d-e)** Resulting peak prediction based on not using **(d)** and using **(e)** the conditions, respectively. [Reprinted with permission from Dylan G. Stone et. al., "Fast characterization of optically detected magnetic resonance spectra via data clustering", ACS Journal of Physical Chemistry C, (2024)].

To illustrate this process, we use *Figure 3.3* as an example. Here only one peak, located at ~3.62 GHz is present in the lowest row. If only these points are passed to the second step, the algorithm identifies the peak locations as being close to the 3.62 GHz (~3.60 GHz and ~3.63 GHz) shown in panel (b). If now we include the second lowest row in the second clustering step (panel (c)), the new peak predictions become ~3.26 GHz and ~3.62GHz,

dramatically changing compared to the column widths. With this in mind, we can define *condition 1* to be:

$$\frac{next\ distance/current\ distance}{next\ width/current\ width} \leq 1 \qquad (3.11)$$

in order to determine if the second lowest row should be included. If the ratio is less than or equal to one, then including the second lowest row does not reveal any previously missed peaks*. The specific choice of 1 for the threshold for *condition 1* was found to work well with the specific artifacts present in our experimental datasets. The values were chosen by comparing the ratios of the distance-to-width change for datasets that were known to miss peaks due to large PL contrast, and those that were known to have low PL contrast and always catch both. This value can be easily modified for datasets that differ greatly from those in this study.[15]

Although *condition 1* eliminated the issue of missing peaks in the datasets, including unnecessary rows can have a negative impact on the accuracy of the peak predictions due to the asymmetric broadening of the peaks as more rows are included. This broadening then shifts the location of the cluster centroids away from the 'true' peak location. To avoid using unnecessary rows in step two of the CA, we included a *condition 2*, which checks if either of the previous columns are contained within either of the new columns. In practice, it is sufficient to only check whether one of the previous columns is contained

* Note: the model currently identifies missed peaks one row above, the code can be easily modified to check any number of rows above for missing peaks if the data contains extremely large PL contrasts, at the cost of run time.

within either of the new columns. For leniency, *condition 2* only checks if at least 80% of the data points are present:

$$\{Previous\ columns\} \subset^{0.8} \{Current\ Column\} \tag{3.12}$$

Here we use $\subset^{0.8}$ to denote a fraction subset. If the condition is met, the current column that contains the previous two columns is instead replaced with them, such that it only includes points from the lowest row. In *Figure 3.3c*, the blue cluster on the right would have also included the purple points from the row above, however, since the previous two columns (*Figure 3.3b*) are contained within the new column on the right, this column is replaced by them. We are now only including the second lowest row for the missed peak in step two of the CA, as shown in *Figure 3.3c.*[15]

The last type of structure considered was that of partial to fully overlapping peaks as shown in *Figure 3.2c*. typically overlapping peaks will satisfy *condition 1*, skipping over *condition 2* entirely, and the need for separate considerations are unnecessary. However, in the rare instance that the second lowest row is used, but *condition 2* is not met, we know we must be dealing with the overlapping structure. Due to the asymmetric broadening of the peak(s), as we include more rows the point in which the clump of points is divided can shift, meaning that all previous columns will not be (approximately) fully contained within either of the new columns. If this occurs, the CA simply uses the lowest row for step two.[15]

It is important to highlight that in cases such as *Figure 3.2c*, due to the way the second step of clustering occurs, the algorithm will always return two separate peak locations, even if the true peaks are completely overlapping. In our CA, this becomes a source of

uncertainty (reducing the accuracy and precision) in the predictions of the resonance points.[15] This pitfall proves to be insignificant when compared to traditional methods, as will be discussed in Section 4.

3.6 Custom Elbow Method

In order to use Kmeans clustering you must first choose the number of clusters to group the data into (see section 2.2.4). One popular technique, which is the basis for our custom method is the *elbow method*. Similar to other popular methods such as the *silhouette analysis*[91,92], the *elbow method* aims to compare the distance between a cluster's points and its centroid using a quantity known as the *inertia*, or the *within-cluster-sum-of-squares* (WCSS). The WCSS is defined as the sum of the square distances between each point and their centroid:

$$WCSS_k = \sum_{x_i \in C_k} |x_i - C_{0,k}|^2 \tag{3.13}$$

Where C_k is the k^{th} cluster, x_i are the points within the k^{th} cluster, and $C_{0,k}$ is the centroid of the k^{th} cluster.

The *elbow method* involves clustering the data for a range of cluster numbers k, and plotting the WCSS or *inertia* against the number of clusters chosen. The optimal number of clusters is then chosen visually by selecting the value at the "elbow" of the plot, or the point in which the slope goes from decreasing steeply to being fairly leveled out. [91,93] *Figure 3.4* shows an example of such a plot. Typically, these plots contain the results for one dataset, and hence have no error bars for the *inertia* values. In this case, our goal was to look at the average *inertia* versus number of clusters plotted for a range of samples to

find the optimal number of clusters for a given type of data (see k_v in section 3.5). Thus, the optimal number of clusters is chosen based on the average optimal number, which is then rounded to the nearest whole number.

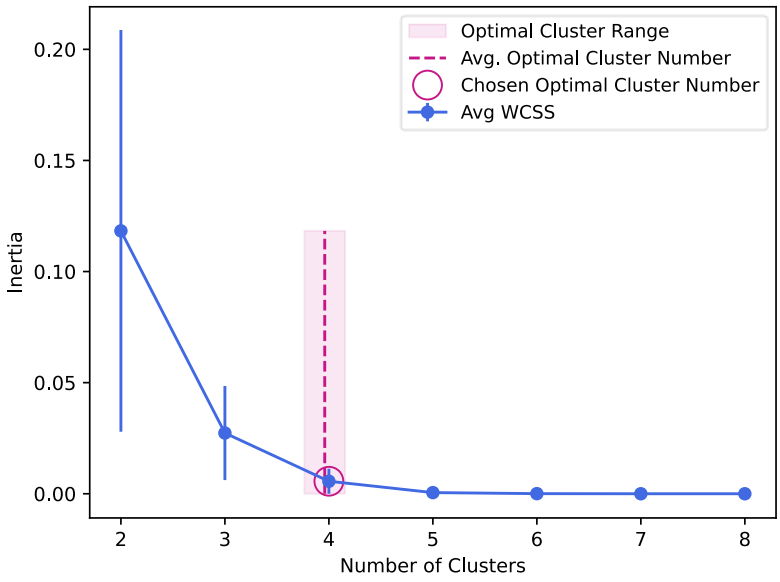

Figure 3.4: An example plot of the *elbow method*. This plot contains an averaged *inertia* of 10 samples to determine the optimal cluster number (k_v) for a group of datasets. The dashed line corresponds to the average of the optimal cluster numbers, with its corresponding range. The closest whole number is chosen as the optimal cluster number for the set of samples.

As mentioned, this approach is typically a visual one, and requires a manual selection of optimal cluster numbers for each plot. Since we are dealing with hundreds of thousands of datasets, this method was modified using gradient calculations in order to automatically select the elbow point without continuous human input[†]. In this case, we

[†] It is important to note that this modification is highly specific to the data we worked with and should not be used in general. However, similar modifications could be made to work with different types of data.

explore clusters numbers from $k = [2,8]$, as shown in *Figure 3.4*. The formula used to determine the optimal number was:

$$k_{optimal} = \min \left| \frac{dI}{dk} - \left[\left(max\left(\frac{dI}{dk}\right) - min\left(\frac{dI}{dk}\right) \right) * 0.3 + min\left(\frac{dI}{dk}\right) \right] \right| \qquad (3.14)$$

Where I is an array of *inertia* values for each value of k, and the max and min refer to the maximum values in the array. This formula was found heuristically and was one of a handful that were created and tested to fill this role. Equation 3.14 was continuously modified and tested against a range of samples and compared to the results chosen by qualitative inspection until consistent agreement was reached. After both visual observation as well as countless applications of equation 3.14 to the datasets used in this study, $k = 4$ was found to be the optimal value for all datasets.

4 Results and Discussion

4.1 Thermometry

For the nanothermometry portion of this work we used nanodiamonds (NDs) which contained both germanium vacancy (GeV) and silicon vacancy (SiV) centers in high concentrations, $\sim 5 * 10^{14} cm^{-3}$ (see ref [7] for description on sample synthesis). Specifically, we looked at six such samples in this study. The photoluminescence of the samples was measured using a custom confocal microscope integrated with an open-loop, temperature-controlled cryostat (see ref [7] for specifics). The NDs were optically excited using a single continuous-wave laser at 532nm using Stokes excitation, producing spectra like those shown in *Figure 2.4b.* In that example, the zero-phonon line (ZPL) and phonon side band (PSB) at 80°C for GeV was ~603.5 and ~631.2 nm, and ~739.7 and ~761.4 nm

for the SiV center. This combination of colour centers is ideal for this study as their spectra are far enough apart to be easily distinguishable, allowing for them to be easily isolated during analysis. Section 2.1 discusses in detail the energy level scheme as well as excitation and emission for colour centers.[7]

From the spectra for both colour centers we look at three physical quantities for both the ZPL and PSB: intensity, full width at half maximum (FWHM), and peak position, giving us a total of 12 physical temperature-dependent quantities which can be monitored simultaneously[‡] (see section 2.1.1 for discussion of the temperature-dependent nature of these quantities). *Figure 2.5* gives a qualitative example of how the spectrum of a colour center changes with changing temperature[§]. The figure clearly shows the change in intensity, FWHM, and overall position as the temperature is varied. *Figure 4.2a-d* shows quantitative dependence of four of the 12 quantities on temperature, by plotting their values compared to the temperature of the sample. Spectra for all six samples used in this study were acquired over a range of 25-80°C in increments of 5°C, giving a total of 12 temperatures. In order to extract these quantities from the spectra, we fit the sum of two Lorentzian functions (two copies of equation 3.3) to the data using the curve_fit() function from the SciPy Optimize library (see section 3.2 for a similar procedure but with ODMR data). An example of such a fitting procedure is depicted in *Figure 4.1*.[7]

[‡] Note: for this study we looked at 10 of the 12 physical quantities, excluding the intensities of the two PSBs

[§] Note: this figure is purely for illustrative purposes and is not based on the samples used in this portion of the study.

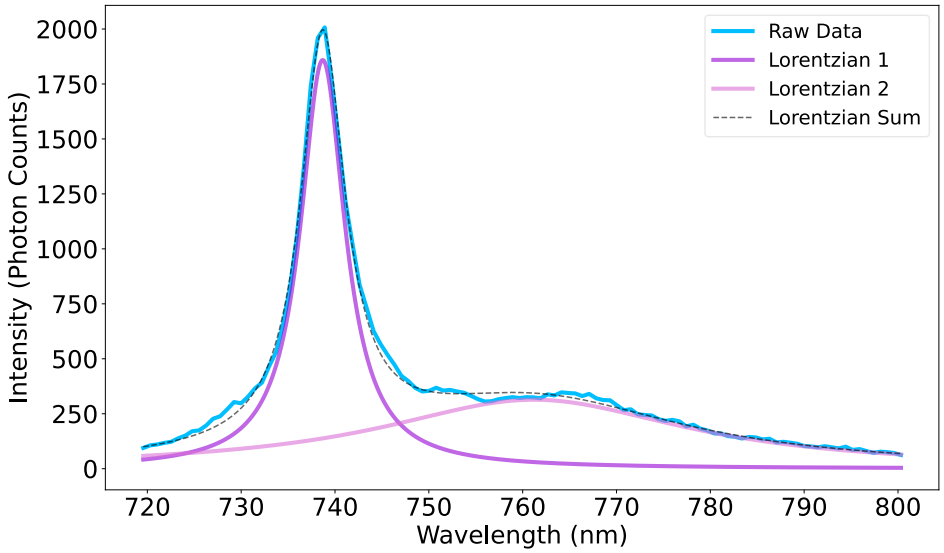

Figure 4.1: Example of a double Lorentzian Fit on an SiV emission spectrum. The two Lorentzian fits labelled "Lorentzian 1" and "Lorentzian 2", when summed together, produce the dashed line that fits the raw data.

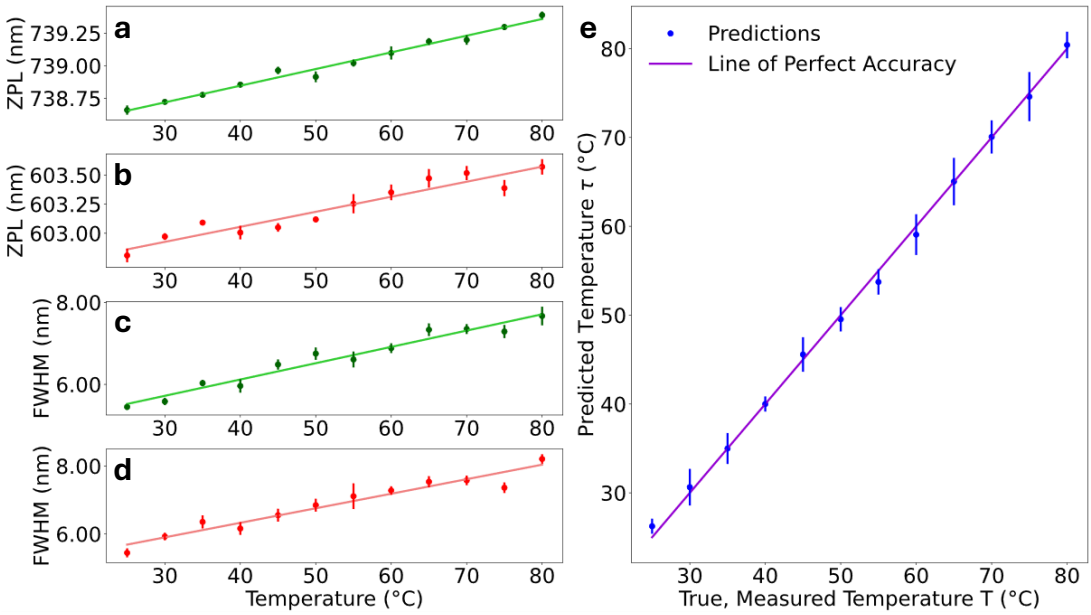

Figure 4.2: Temperature-dependence of SiV and GeV photoluminescence spectra. **a-d)** Temperature-dependent changes of the ZPL emission wavelength **(a,b)** and of the FWHM **(c,d)** for the SiV (green) and GeV (red) colour centers, respectively. The lines represent the linear fits

of the data. **e)** The accuracy of the Multi-Feature Linear Regression model shown as the difference between the true (i.e. measured) temperature (x-axis) and the models predicted value (y-axis). The straight line represents the "perfect" accuracy, i.e. where the prediction is exactly equal to the measured value. Note: while the error bars in **(e)** appear greater than those in **(a-d)**, they are a factor ~1.2x smaller. The relative errors are ~3.3% in **(e)** compared to ~3.9% in **(a-d)**. [Reprinted with permission from Dylan G. Stone et. al., "Diamond Nanothermometry Using a Machine Learning Approach", ACS Optical Materials, (2023)].

The chosen machine learning algorithm for this portion of the study was Multi-Feature Linear Regression (MF-LR) (see section 2.2.2). The reasoning is two-fold. For one, there is a subtle but important distinction between machine learning approaches and traditional statistical inferencing that is explained in detail in section 2.3.1. Briefly, the key difference is in machine learning's ability to handle smaller scarce datasets better than traditional fitting. Although the 12 parameters, which will from here on be referred to as *features*, are known to be non-linearly dependent on temperature, they can be approximated as such over the relatively small temperature interval we are investigating (25-80°C). This gives us our second reason, being that the application of linear regression in this case works out of the box and requires no pre-linearizing of the data. For larger ranges however, the data can be pre-linearized using the functional dependences on temperature if it is known (which is usually the case for spectroscopic features of diamond colour centers and other optical nanothermometers) *Figure 4.2a-d* shows the four best, i.e. the most linear, of the 12 features over our temperature range. Here we plot the specific values of either ZPL position or FWHM for the SiV and GeV centers for one of the six samples versus the temperature the value corresponds to. The error bars were calculated by taking the standard deviation of the values for all six samples at each

temperature point to illustrate the variance from sample to sample and how well it linearly predicts temperature. Panel e) on the other hand shows the difference between the temperature predicted by the MF-LR model (y-axis) and the true temperature measured by the cryostat (x-axis). The predicted points plotted are the average of six different train-test runs, with the error bars being the standard deviation of the predictions between the six runs. The details on the training and testing of the MF-LR will be discussed in detail later, the main take away here is that the model can, at the very least qualitatively, predict the temperature of the sample as-well or better than the best four of the 12 features can on their own. Further down we will look at the specific quantitative performance metrics for each of the single features compared to the MF-LR model.[7]

In this portion of the study we had three main goals: **1)** create a general purpose model that can be trained on a set of samples once, and then used on any similar samples afterwards without needing to do any calibration, **2)** achieve better performance using the MF-LR model than that of standard single feature approaches, and **3)** do so with fewer data points reducing the amount of data collection required. To test our first goal, we devised a set of training and testing regiments to see how well the model could predict temperatures for samples it was not trained on (for more specifics on training and testing machine learning models, see section 2.2). There were two main approaches to testing the MF-LR model against the single feature competitors, the first being to train each model on $N = 1$ of the six samples, and test its performance using the remaining five, unseen samples. We did this for every combination of 1-training/5-testing sets giving us a total of 30 tests. The second approach involved training each of the models on $N = 5$ of the six samples, and testing its performance on the remaining, unseen sample. We again did this

for every combination, this time for all 5-training/1-testing sets, giving us a total of 6 tests. Note that here the 'model' for the single features consists of a single-feature linear regression, i.e. equation 2.1 with only a single weight.[7]

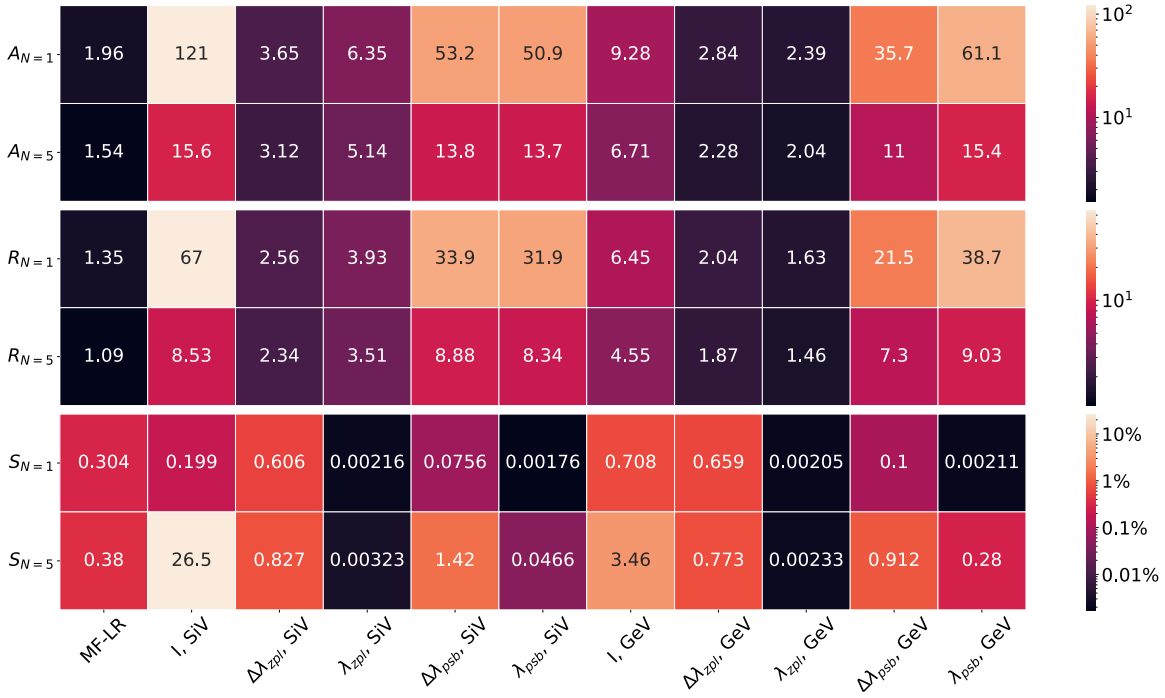

Figure 4.3: Heatmap of the performance of all tested nanothermometry models. Three generalized performance metrics are shown: Accuracy (A), Resolution (R), and Sensitivity (S). For each metric we show results for each of the two training regiments: 1-training/5-testing sets ($N = 1$) and 5-training/1-testing set ($N = 5$), which are averages of all possible training/testing combinations. Our MF-LR model is compared to the 10 single feature fits tested: ZPL intensity (I), ZPL width ($\Delta\lambda_{zpl}$), ZPL position (λ_{zpl}), PSB width ($\Delta\lambda_{psb}$), and PSB position (λ_{psb}) for both SiV and GeV. To more easily see the difference between values, a logarithmic colour scale is used where darker colours indicate better (i.e. smaller) accuracies and resolutions, while lighter colours indicate better (i.e. higher) sensitivities. The MF-LR model displays the best accuracy and resolution, and a moderate sensitivity, beating out the SiV's λ_{zpl}, which takes second place in

terms of accuracy and resolution. Many other models display significantly larger sensitivities, however often with poor accuracy and resolution.

Figure 4.3 shows a heatmap of the three *generalized* performance metrics, Accuracy (A), Resolution (R), and Sensitivity (S) for our MF-LR model as well as all 10 of the single feature fits we tested. We specify that these are the *generalized* versions of these metrics, as the normal literature versions of each, by definition, do not apply to this type of testing. In section 3.1 we discuss in detail exactly how these variants are defined and calculated. The important point here is that these metrics convey the same information as their normal counterparts, but for situations where the testing data is different from the 'calibration' (i.e. training) data. For each metric we have results for both training regiments: 1-training/5-testing sets ($N = 1$), and 5-training/1-testing sets ($N = 5$). Thus the $N = 1$ case can be seen as a lower bound, as the models are only trained on a single dataset and tested on other sets which they have not seen before. It is important to note that all of the values presented here are averages across all the combinations of training/testing.[7]

Analyzing *Figure 4.3* we can see that the first two of our goals have been accomplished: (2) the MF-LR model displays both better accuracy (and resolution) than the leading single feature models, and (1) does so while being applied to NDs for which it has not been trained on, with no extra calibration being required. The MF-LR model displays accuracies of $A_{N=1} = 1.96\text{K}$ and $A_{N=5} = 1.54\text{K}$, and resolutions of $R_{N=1} = 1.35\text{K Hz}^{-1/2}$ and $R_{N=5} = 1.09\text{K Hz}^{-1/2}$. It is followed in performance by the single feature model tracking the temperature-dependence of the SiV's ZPL, which has accuracies of $A_{N=1} = 2.39\text{K}$ and $A_{N=5} = 2.04\text{K}$, and resolutions of $R_{N=1} = 1.63\text{ K Hz}^{-1/2}$ and $R_{N=5} = 1.46\text{ K}$

$Hz^{-1/2}$. Overall, the multi-feature, multi-ND (i.e. $N = 5$) approach produces better accuracies by a factor of ~1.3 to 10.1x, and better resolutions by a factor of ~1.3 to 8.3x. The improvement in accuracy is likely due to the averaging effect affecting the noise.[7]

In order to investigate our third goal, reducing the amount of data required to achieve acceptable accuracies and resolutions, we devised a setup where the models are trained on increasingly fewer data points to gauge how much data is required for 'acceptable' levels of accuracy and resolution. This investigation is motivated by the fact that acquiring calibration curves is a time-consuming step for many nanothermometry methods in practical settings. *Figure 4.4* shows the results of this investigation, where accuracy and resolution are plotted against the number of data points used to train the models. To illustrate the improvements over typical approaches, where individual NDs are calibrated using a single temperature-dependent feature, we included in the figure the best MF-LR model ($N = 5$), along with the four best single feature models trained on single samples ($N = 1$). As with the previous figure, all points in this figure are averages over all the training/testing combinations, with the shaded areas being the standard deviations in these combinations. As one would expect, in general, increasing the number of data points produces better (i.e. lower) accuracies and resolutions for all models. However, for the best performing models, the accuracy and resolution does not improve significantly after more than 3 training points. This is highly advantageous for practical cases, as in this situation, only one fourth of the data was required to get nearly the same results as using the full dataset. For instance, the MF-LR model increases significantly (~33%) going from 2 to 3 training points, but only increases by a modest ~5% going from 3 to 6 points. Similarly, the resolution of the MF-LR model increased by ~38% going from 2 to 3

points, but only ~11% from 3 to 6 points. Thus, in accordance with our third goal, we have shown that it is possible to produce similarly accurate results with significantly less calibration data.[7]

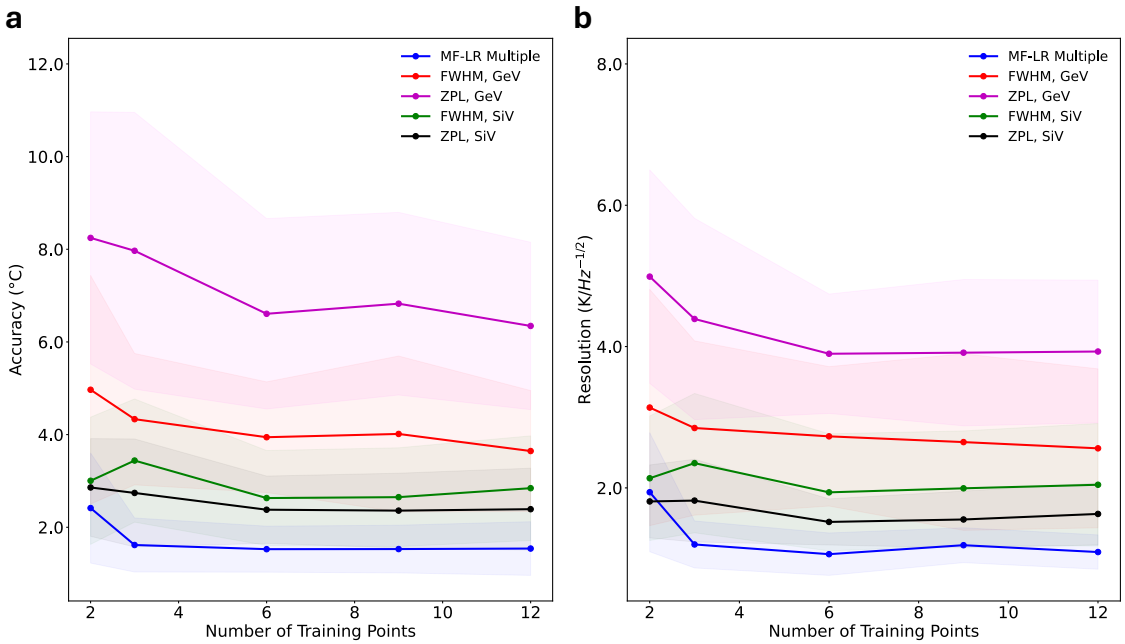

Figure 4.4: Performance of MF-LR model compared to four best single feature models. Accuracy **(a)** and resolution **(b)** of the MF-LR model (Multiple NDs) compared to the four best performing single feature models as a function of the number of data points used to train each of them. As expected, using all 12 of the available data points for training results in the better (i.e. lower) accuracies and resolutions. However, and interestingly, the MF-LR produce relatively good results for both figures of merit with as little as 3 training points, with only a small improvement increasing to 4 and above. Each point shown is an average of all the combinations of training and testing, with the shaded regions around each curve depicting the standard deviations of the averages for each point. [Reprinted with permission from Dylan G. Stone et. al., "Diamond Nanothermometry Using a Machine Learning Approach", ACS Optical Materials, (2023)].

Up until this point we have explained that the MF-LR model uses multiple features from the spectra of the colour centers to make its temperature predictions but have not specifically addressed which of these features were used. Different combinations of features can produce different results, so it was important for us to determine which would result in the best accuracy and resolution in our tests. To determine the best features to include, a simple script was written that tested all 1013 combinations of using 2-10 features in the MF-LR model. From these the best can easily be selected and used moving forward. In our case, the best values of $A_{N=1}$, $R_{N=1}$, and $S_{N=1}$were obtained using the FWHM ($\Delta\lambda_{zpl}$) of the GeV, and ZPL position (λ_{zpl}) and FWHM ($\Delta\lambda_{zpl}$) of the SiV. At the same time, the best results for $A_{N=5}$, $R_{N=5}$, and $S_{N=5}$ came from using the FWHM ($\Delta\lambda_{zpl}$) of the GeV, and intensity (I), ZPL position (λ_{zpl}), FWHM ($\Delta\lambda_{zpl}$), and PSB position (λ_{PSB}) of the SiV.[7]

Some final notes, as with many machine learning algorithms, this MF-LR model has the attractive quality of getter better over time. As more data is fed into the model during training, the better it can predict future unseen data. However, one should be wary of using any trained model, be it linear regression or any other supervised model, outside of the range it was trained in. As with many forms of calibration, you can only be sure of a device or algorithms accuracy when used within the range in which it was calibrated. In our case, these models should not be used significantly outside of the 25 to 80°C temperature range.[7]

Statement of Contributions

The results presented in this section were published[7] with co-authors Carlo Bradac, Yongliang Chen, Evgeny A. Ekimov, and Toan Trong Tran. C.B. and I conceived of the idea of the project and performed the data interpretation and analysis, Y.C. and T.T.T. carried out the experimental measurements, and E.A.E. synthesized the diamond samples. All authors discussed the results and commented on the manuscript.

4.2 Optically Detected Magnetic Resonance

For the optically detected magnetic resonance (ODMR) portion of this work we used a combination of experimental and synthetic datasets. The experimental portion used hexagonal boron nitride (hBN) samples with boron vacancy centers, V_B, specifically negatively charged vacancy centers (see section 2.1 for details on hBN). The colour centers were created using flakes of hBN $\sim 100 * 50 \mu m^2$ in area and ~ 200nm thick which were exfoliated onto SiO_2 substrate. Ensembles of vacancies were then produced by irradiating the material with nitrogen ions. For more details on sample preparation see ref [15]. The samples were optically excited using a 532nm laser on a lab built confocal microscope, with 5mW of laser power, with the vacancies emitting at ~800nm. ODMR was performed using constant wave excitation while a microwave (MW) signal is swept uniformly from 3 to 4GHz. The frequency range was divided into 200 individual bins, with each step involving 1ms of integration time with the MW field active, followed by 1ms without the field serving as a reference for determining the ODMR contrast. A permanent neodymium magnet was used to apply a magnetic field of varying strengths by placing it at specific distances from the sample to frequency shift the $m_s = \pm 1$ spin resonances (see section 2.1.2 for ODMR details). Data was collected for four sets of

magnetic field strengths (i.e. five distances from the sample) on five different excitation locations, giving us a total of 20 different measurements. For each of the 20 measurements, a total of 10s per bin were saved individually as 1ms per bin sweeps (i.e. 10,000 1ms/bin sweeps), this was done to allow us to control, after the fact, the amount of integration time we wished to work with.[15]

Simulated ODMR data was also produced for this study using the experimental data as a template. This data assumes a Lorentzian shape with a full width at half maximum (FWHM) and photoluminescence (PL) contrast (i.e. the amplitude). This was chosen as our experimental data follows a Lorentzian shape, and it allows for a direct quantitative comparison of the performance of the models we wished to test, in this case our custom model and traditional fitting (more on that later). The synthetic data consists of two Lorentzian peaks, each with their own amplitude, FWHM, and peak location. The constraints on the synthetic data were kept to a minimum: resonances were allowed to occur anywhere within the frequency range, and the FWHMs and PL contrasts were independent and allowed to vary randomly within set ranges that were chosen to match those of the experimental data. *Table 3.1* lists the specific values and ranges used for data simulation, and section 3.3 outlines the simulation process. Real physical systems often have more constraints on said parameters. For example, there can often be symmetries in the transitions between $m_s = 0 \leftrightarrow -1$ and $m_s = 0 \leftrightarrow +1$. We deliberately relaxed the constraints on the data to allow for our custom model (more below) to be more general and allow for the types of experimental artifacts that were present in our real data. For instance, our experimental data displays large differences in PL contrast between the two peaks due to the MW amplifier displaying a nonuniform, frequency-dependent response.

Although the various parameters for the simulations are chosen at random within set ranges, we can easily note down these values for each dataset, allowing us to determine important performance metrics such as accuracy and resolution with perfect precision.[15]

For this study, we generated 10^5 data sets for each value of average counts per bin, N. *Figure 4.5d-f)* shows three such data sets spread across the range of average bin counts simulated. Panels *a-b)* show three qualitatively similar experimental datasets for visual comparison. Experimental datasets are labelled using the integration time per bin, τ_b, while the synthetic datasets are labelled using N. The integration time per bin and average counts per bin can be directly related to one another using the efficiency of the experimental setup. For instance, in our case 1ms/bin corresponds to $\sim 10^7$ counts/bin. The absolute counts/bin are not important, the key is the relative number of counts in on-resonance versus off-resonance regions, i.e. the ODMR contrast. For datasets such as those shown in *c)* and *f)*, it is relatively straight forward to use traditional fitting techniques to determine the resonance points. However, for more scarce data such as those shown in *a-b)* and *d-e)*, traditional methods can often fail to converge. It is important to note that while the data simulation mirrors that of a real setup, it does not incorporate a means of adding additional noise to the measurements that arise from detector dark counts. This is most prominent when comparing panels *a)* and *d)*, where *d)* shows more distinct drops from the maximum PL count. Regardless, since the important factor is simply the ODMR contrast, not absolute counts, the simulated data behaves similarly enough that it can be used in this study to benchmark the models quantitatively.[15]

Determining resonance points for scarce data is far more desirable, as it requires less data acquisition time and can handle data with more noise and low contrasts. There are two relevant observations when it comes to data acquisition. Firstly, even with small integration times per bin, total integration times can add up quickly to seconds or tens of seconds per sweep. For instance, the experimental data shown in *Figure 4.5a-b)* took 1.2-16s to acquire. Secondly, it can be onerous to find the resonance peaks in data that exhibit both low ODMR contrast (determined by the photophysics of the emitter via its intersystem crossing rates) as well as low signal-to-noise ratios. In this study, we aimed to create a custom Clustering Algorithm (CA) that, when compared to standard fitting (SF) approaches, **(1)** performs better in both accuracy and resolution, as well as **(2)** being able to do so with less information (i.e. less integration time/fewer counts per bin).[15]

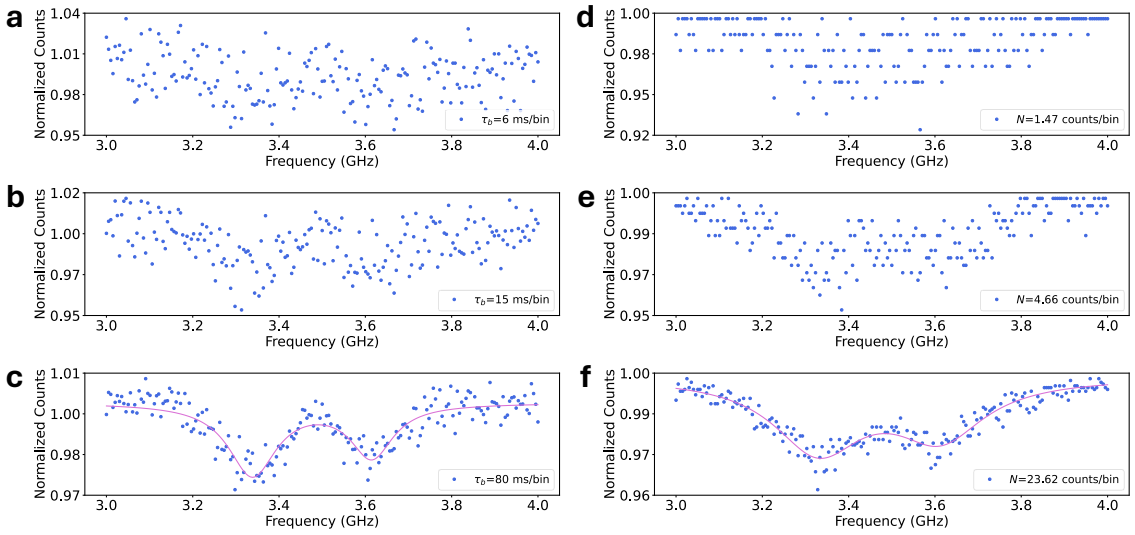

Figure 4.5: Experimental and synthetic ODMR spectra. **a-c)** experimental data obtained from ensembles of V_B centers in hBN. The spectra were acquired by integrating on each frequency bin for **(a)** 6ms/bin, **(b)** 15ms/bin, and **(c)** 80ms/bin; which corresponds to a total integration time of 1.2s, 3.0s, and 16s, respectively. **d-f)** synthetic ODMR data. Spectra are instead defined by

average number of counts/bin, N, equal to **(d)** 1.47 counts/bin, **(e)** 4.66 counts/bin, and **(f)** 23.62 counts/bin. The solid lines correspond to the Lorentzian fits of the data. [Reprinted with permission from Dylan G. Stone et. al., "Fast characterization of optically detected magnetic resonance spectra via data clustering", ACS Journal of Physical Chemistry C, (2024)].

Using data pattern recognition and clustering algorithms is not a new idea and is in fact a well-established method of analysis for magnetic resonance data and for spectral interpretation. These methods are especially common in fields such as (bio)medical science where they are used to reduce the complexity of large datasets.[94,95] In our case we focused on the use of clustering to accurately and rapidly extract information from data that is noisy or scarce, which traditional models may struggle to return accurate results or even fail to converge, rather than reducing the complexity. To do so, we created a custom CA that uses Kmeans clustering (see section 2.2.4) and open source scikit-learn as the foundation of the model. Section 3.5 gives a detailed description of the CA and how it handles specific cases seen in our data. In short, the CA involves two stages of clustering. The first stage involves clustering the data into k_v vertical clusters or *rows*. The number of rows is determined by a custom version of the *elbow method* which is described in detail in section 3.6. Briefly, the elbow method involves plotting the within-cluster-sum-of-squares (i.e. the distance between a cluster centroid and its points) versus the number of clusters used (k_v). The plot typically displays a steep drop then levels out creating an 'elbow' at the optimal cluster number. In our case $k_v = 4$ was the best for all datasets used. Secondly, the lowest row is clustered once more into $k_h = 2$ horizontal clusters or *columns*. The horizontal position of these two centroids is then returned as the predictions of the resonance peak locations. The CA also includes some extra *conditions* that help it

tackle two main scenarios: i) differentiating between datasets with large versus small variances in ODMR contrast between the two peaks (*Figure 3.2a-b*), and ii) cases where the peaks are partially/fully overlapping (*Figure 3.2c*).[15]

We characterize the performance of our CA based on the accuracy and precision, benchmarking them against statistical inferencing methods, specifically an SF of two Lorentzian functions (see section 3.2). We define accuracy as the difference between the true values of the resonance frequencies and the predicted values, and precision as the standard deviation in the accuracy for all 10^5 data sets tested (per N). Practically speaking, precision translates directly to resolution for sensing techniques that map physical quantities such as $\boldsymbol{E}$, $\boldsymbol{B}$, $\boldsymbol{T}$, etc. to relative positions of ODMR resonance frequencies as it dictates the minimum resolvable difference one can measure. For instance, the resolution of our technique mapped to $\boldsymbol{B}$ can be found by dividing our precision by the gradient of the magnetic field, in our case that of the permanent neodymium magnet.[96] This is analogous for the electric field. For mapping to temperature, a similar calculation to what is shown in 3.1 can be done, in this case multiplying our precision by the square root of the integration time per bin. For synthetic data, the true values are known with absolute precision, as they are used to generate the datasets themselves. For real experimental datasets, the true values are determined by acquiring a standard fit of the datasets for long integration times per bin ($\tau_b = 10$s/bin) and high signal to noise ratio (~500). Since our datasets were acquired as individual 1ms/bin sweeps, we can simply sum up all 10,000 sweeps to produce such datasets. The SF in this case fits the data using least square minimization to determine the optimal

parameters for the two Lorentzian functions, where the two peak locations are the resonance frequency predictions.

Figure 4.6d-e) show the performance of the CA compared to the SF, where accuracy is plotted against the number of average counts per bin, N. Panels *a-c)* show examples of the three types of data shapes explored in this study, where peaks have *(a)* large variation in PL contrast, *(b)* small variation in PL contrast, and *(c)* partially or fully overlap with one another. Section 3.5 discusses in detail the inner workings of the CA and explains how the model identifies and handles each case. As one would expect, in general, increasing N produces better (i.e. lower) accuracies and resolutions. The two panels *d-e)* display the exact same information, with *d)* showing a subset of the data with PL contrast variations up to 15%, versus the full data set *(e)* with variations up to 65%. The CA performs similarly in both cases despite the large difference in allowed variation, and achieves better accuracy and resolutions than the SF, aligning with aim (1) for this study. For instance, in *d)* the CA achieves similar or better accuracy than the SF with much fewer (factor $\geq$ 5x) average counts/bin with an overall higher resolution (factor $\geq$ 3x). The CA's robust handling of such small to large variations is due to one of the custom *conditions* imposed to handle such scenarios (see section 3.5 for details). Interestingly, the same cannot be said for the SF, which suffers both worse accuracy and resolution when handling only the subset of data with variations up to 15%. Practically speaking, this makes the CA an even more attractive option for experimental setups where it is unknown if experimental artifacts causing changes in PL contrast are present, as the CA will perform similarly regardless.[15]

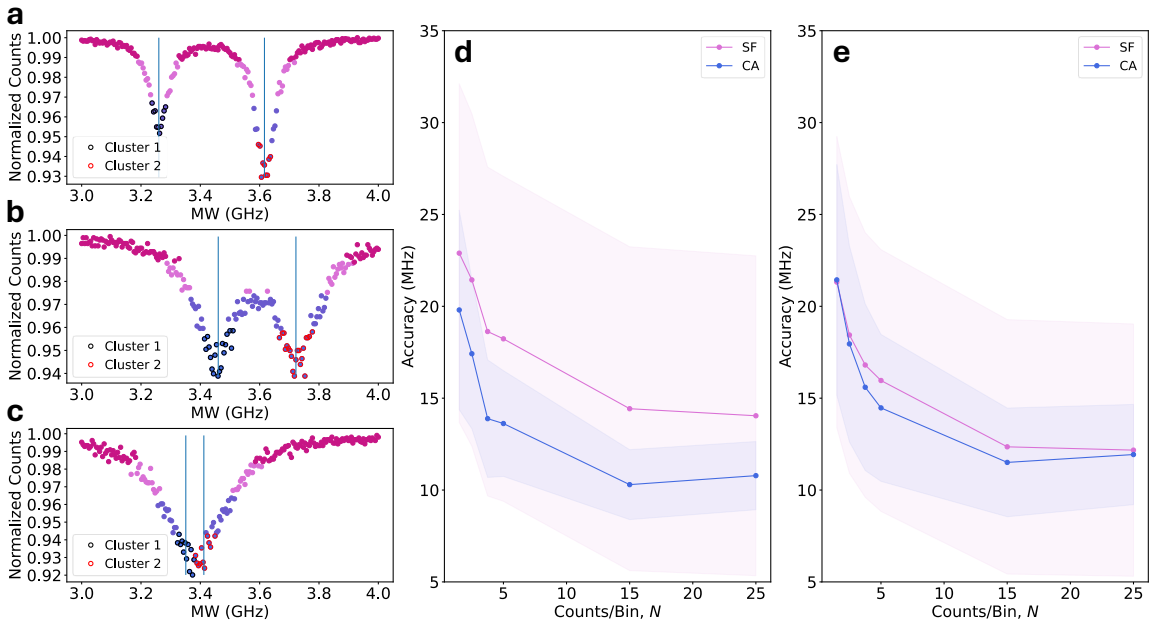

Figure 4.6: Operation and performance of CA and SF. **a-c)** Graphical illustration of the CA applied to the three variations of data explored in this study: **(a)** large variation in PL contrast, **(b)** small variation in PL contrast, and **(c)** partially to fully overlapping peaks. CA assigns data to vertical *rows* (solid colours) and horizontal *columns* (outlines, labelled Cluster 1 and 2). The x-coordinates of the centroids correspond to the resonance frequency predictions (vertical lines). **d-e)** Accuracy of the CA and SF models (data points), with shaded regions indicating the precision (standard deviation in accuracy). Panels **d)** and **e)** differ as they show performance when applied to data sets with variation in PL contrast between the two peaks of up to 15% and 65% respectively. Note, the shaded areas have been divided by five in **d-e)** to improve visual clarity. [Reprinted with permission from Dylan G. Stone et. al., "Fast characterization of optically detected magnetic resonance spectra via data clustering", ACS Journal of Physical Chemistry C, (2024)].

We have shown that the CA can be a powerful tool for speeding up ODMR acquisition and analysis, boasting $\geq$ 5x efficiency over the SF in certain scenarios. Thus, we have

accomplished aim (2), achieving better accuracy with less information (i.e. average counts per bin). It is important to note that as the number of counts per bin and the signal-to-noise ratio increases, the CA and the SF converge at similar accuracies and resolutions. However, the CA outperforms the SF for noisy/scarce data which is attractive as ODMR experiments can be lengthy and subject to experimental artifacts. It is also important to note that, although our data follows a Lorentzian shape, the model is agnostic to the functional shape of the peaks and would work equally well for spectra with Gaussian or Voigt functions, which can be the case if inhomogeneous broadening occurs. This is not the case for the SF, as a functional form must be imposed when curve fitting. Practically, this means the CA can be used with experimental data where it is unclear exactly what functional form is present, given the visual similarities between the three distributions mentioned.[15]

We argue that the superior performance of the CA over the SF for noisy/scarce data comes from the effect of compressing the dimensionality of the data, this by-design effect leads to the CA having a greater effective number of events per variable (EPV). Traditional SF methods work by determining the parameters of a function which is fit to the data set, in our case two Lorentzian functions, by minimizing the sum of square errors between the raw data's y-values and the fitted curves. Here there are six variables to be fitted: FWHM, peak position, and amplitude for both peaks. It is known that SFs falter for *small* datasets, where *small* is defined with respect to the number of EPV to be fitted. As the number of variables increases, so does the required number of points to achieve a good fit. In our case *small* might be defined $\lesssim 25$ since the SF under performs the CA in this range, or it could simply be a value of N for which an acceptable minimum accuracy

is achieved. Regardless of the line drawn, the CA has a greater EPV due to how it operates. The initial stage of vertical clustering compresses the dimensionality of the problem with respect to the y-axis, reducing the problem to that of a two-variable fit along the x-axis. These two variables are simply the two centroids of the remaining data when clustered into two groups. In contrast, for the same data, the SF must determine six variables. This results in the CA achieving improved performance over the SF. These results can be generalized beyond this study, as the CA should outperform the SF for similar multivariate data where the number of sought clusters is less than the number of variables to be fit.[15]

There are two caveats with using the CA: i) this method is only applicable to the so-called classification problems, where the goal is to cluster the data into groups or categories, and ii) the compression of dimensionality inherently leads to a loss of information. In our case this loss of information is acceptable, and in fact has been shown to be advantageous to increase the effective number of EPV, such that the CA can achieve equal or greater performance with less information. However, the SF is richer information-wise, as it returns the FWHMs and amplitudes alongside the peak locations.[15]

Some additional notes, the CA, unlike the Multi-Feature Linear Regression explored earlier, is an *unsupervised* model (see section 2.2.1), meaning that it does not require any training before use. This is desirable in this case as *supervised* models can sometimes require large amounts of training, and ODMR acquisition can be lengthy.[15]

By design, our custom CA always returns predictions for the resonance frequencies even for scarce sets. For demonstration purposes, we emulated 10^5 datasets where the average number of counts per bin were as low as $N = 0.05$. As expected, the CAs accuracy and

resolution were poor, 89 and 109MHz respectively. However, this information is still valuable for dynamic-type ODMR measurements where instead of scanning the MW signal uniformly over the entire frequency range (3-4GHz in our case), you sweep over smaller target windows informed by the CA. This approach is not possible with the SF, as for the $N = 0.05$ case, it failed to converge for over 60% of the 10^5 datasets.[15]

For completeness, results from two other machine learning (ML) models that were tested are included in *Figure A.1*. These models, namely *Multi-Feature Linear Regression* (MF-LR) and *Neural Networks* (NN), were tested during the initial phases of this study as potential competitors to the SF before our CA was created. A description of how each of these models work can be found in sections 2.2.2 and 2.2.3 respectively. Since these models are *supervised learners* (see section 2.2.1), to test their performance, they were first trained on a subset of the 10^5 datasets generated for each value of average bin count (N). The remaining unseen subset was then used to test the performance of both models. After running all three models through our testing, it was clear that MF-LR and NN were not going to be competitive in this application, so other avenues were explored, ultimately leading to the creation of the CA model that this section focuses on. Since the results for the MF-LR and NN were inferior to that of the SF, they were not included in this section.

Statement of Contributions

The results presented in this section were published[15] with co-authors Carlo Bradac, Benjamin Whitefield, and Mehran Kianinia. C.B. and I conceived of the idea of the project and performed the data interpretation and analysis as well as data simulation. B.W. and M.K. collected the experimental ODMR data. All authors discussed the results and commented on the manuscript.

4.3 Next Steps—Autocorrelation

For the Autocorrelation work that was done, a combination of experimental and simulated data was produced. The experimental portion used hexagonal boron nitride (hBN) samples with negatively charged boron vacancy centers, V_B. Section 2.1 gives a detailed description of the relevant aspects of hBN with vacancy centers for this study. The samples contained many ensembles of colour centers randomly dispersed throughout. Optical excitation was achieved using a 532nm continuous wave laser with a custom-built confocal microscope. The laser used operates from 1-100mW, with different laser powers being used depending on the ensemble being measured, often closer to ~20mW. A piezoelectric stage was used to precisely move the stage for both sample scanning and ensemble locating. Samples were scanned in rows of 50x1μm creating a 2D 50x50μm slice of the surface in order to locate colour centers. *Figure 4.7* shows an example of one such optical map, where the map axes represent the spatial location in micrometers, and the colour represents the number of photon counts measured within a given $1x1$μm section (colour bar is approximate, see figure caption). In general, larger, brighter regions tend to indicate large ensembles of emitters, while smaller, dimmer regions tend to indicate smaller ensembles or even single emitters.

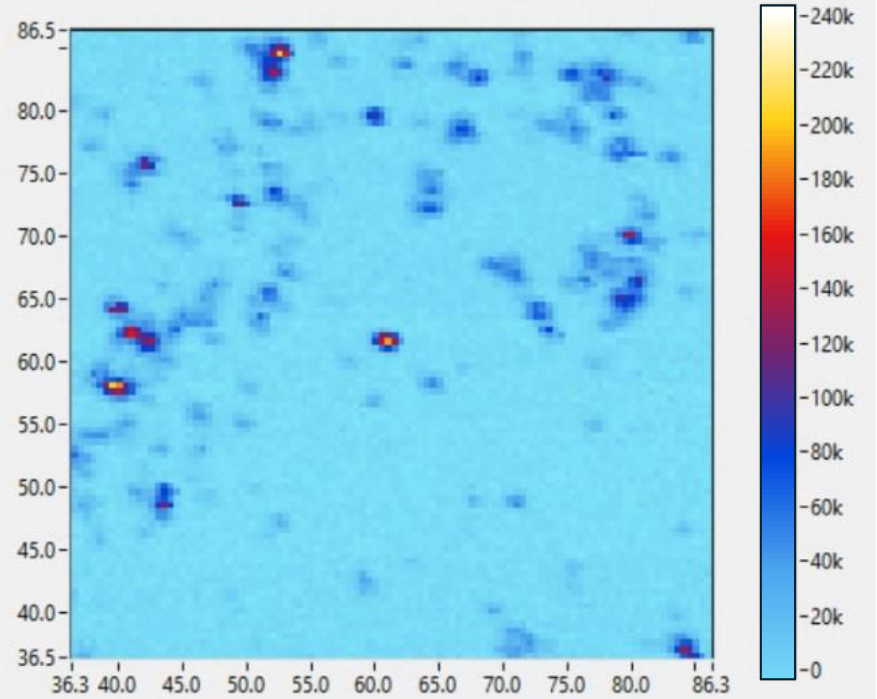

Figure 4.7: Example of a 2D optical sample map (slice). The map axes show spatial position in micrometers, while the colours represent the number of photon counts measured at each $1x1\mu m$ square. In general, larger, brighter regions indicate large ensembles of emitters, while smaller, dimmer regions indicate smaller ensembles/single emitters. Note: the colour bar on the right was superimposed on the image as the original image output did not retain the colour gradient. Therefore, photon counts shown in the map are only approximately equal to their corresponding region in the colour bar. Regardless, relative photon counts can still be compared.

The goal of autocorrelation experiments in this context are to determine whether or not a bright region in the colour map is a single emitter, or an ensemble of many emitters. The motivation behind this is the applications of single photon emitters in quantum information technologies, where such individual emitters can be used as photonic qubits.[61] The process of determining whether an optically active site in a material is a single emitter involves using the Autocorrelation technique, which is described in detail

in section 2.1.3. Briefly, due to the non-zero amount of time required for an optical transition to occur in the material, there must be a non-zero amount of delay between two consecutive photon emissions from the same emitter (i.e. multiple photons cannot be emitted from one emitter simultaneously). This fact is key to this technique, as two detectors (*start* and *stop*) are used to measure the time delay between photon emissions for a given region in the sample. If the number of pairs of photon counts are plotted vs the time delay (histogram) between the two events, we should expect, in a perfect scenario, to see zero events that occurred simultaneously. However, due to dark counts and other sources of error we instead see a significantly lower, but non-zero number of counts around zero-delay. Such a plot is shown in *Figure 2.7*, where a large dip can be seen around the zero-delay mark, which slowly rises and levels out for larger delays. If we are instead looking at a region with multiple emitters, we would expect to see significantly more counts around zero-delay, with very large ensembles having perceptively no dip whatsoever. Practically, one must decide on a threshold with which determines whether a given optically active region hosts more than one emitter. This threshold depends on the setup used and the amount of noise present, however normalized counts below 0.5 are often chosen to be indicative of a single emitter.[61]

Similarly to the ODMR portion of this study, synthetic data was simulated to both increase the amount of available data, as well as have complete knowledge of the parameters for quantitative performance comparisons. Increasing the amount of available data is particularly attractive for *supervised learners*, as these types of models often require large amounts of training data where the ‘answer’ (label) being learned is known. Data simulation allows for practically limitless training data without the need for lengthy

data acquisition, which can often be one of the biggest bottlenecks for supervised machine learning models. The transition rates between various electronic levels are also well known and understood for colour centers in diamond and hBN and can be used to generate datasets to imitate specific colour centers accordingly. A description of the simulation process can be found in section 3.4. *Table 3.1* gives examples of values used from reference [61], which can be easily changed to simulate specific colour centers in various materials. Briefly, the simulation is designed to follow that of a real-world experiment, where each data point in the histogram is added to its corresponding time bin one at a time. For each time bin, the probability of seeing a second photon is computed and compared to a randomly generated value between 0 and 1. If the random value is less than the probability, we add one to the corresponding time bin, otherwise a detection did not occur and we move on to the next bin. This continues until either a second detection occurs, or the end of the monitored time window is reached, and the timer starts over for a new pair of events. In order to simulate more than one emitter, we simply generate a random number and compare to the probability n times, where n is the number of emitters, for each time bin. *Figure 4.8* shows examples of simulated autocorrelation data, where the simulation run time increases from *a-f)*. In this context the simulation run time can be related directly to real-world integration time through the experimental setups efficiency and the parameters used for the simulation.

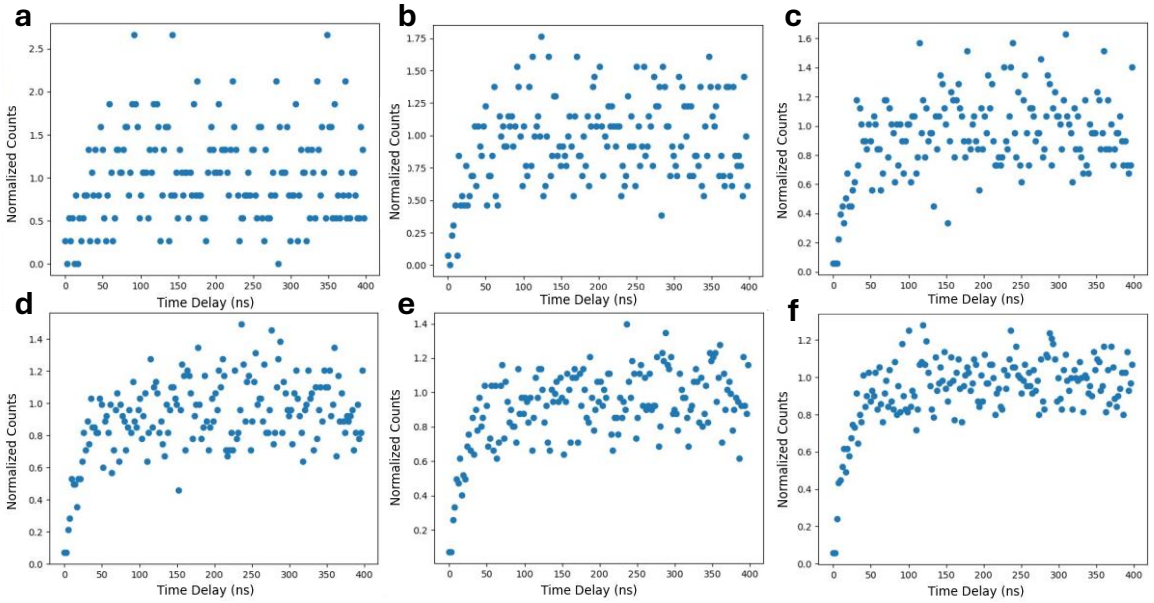

Figure 4.8: Examples of simulated autocorrelation plots for various run times. Here the simulation run time, which can be related to real-world experimental integration time, increases from **(a-f)**. These plots were generated using most of the values given in *Table 3.2*. The specific parameters used are not important, the key point here is simply to illustrate the progression from scarce random looking data to more structured data. Note that here we only simulate the positive side of the time axis, as the positive and negative regions in a real plot give the same information as the only difference is which of the two detectors went off first. However, the simulation can easily be modified to generate both regions if desired.

After producing our data generating code, this section of our work was paused in order to focus on ODMR and was ultimately left unfinished. However, similar approaches to those used in the nanothermometry and ODMR studies can also be applied here. The next steps for the autocorrelation study would involve testing out a handful of machine learning (ML) algorithms against traditional methods such as curve fitting. For instance, Kudyshev et. al. demonstrated a significant improvement over traditional Levenberg-Marquardt (L-M) fitting using *Convolutional Neural Networks* (CNN) and *Voting Classifiers* (VC),

achieving better accuracy than the L-M approach while also using less data. For instance, in one of their tests, the CNN and VC boast mean accuracies of 98% and 95%, respectively, over the L-Ms meager 55% which is barely better than the flip of a coin. They also demonstrated strong performance when it came to decreasing amounts of available data, with the CNN and VC dropping to at most 70% compared to the L-Ms 20%.[61] For context, a CNN is a variant of the more basic NN described in section 2.2.3, with additional hidden layers that slide a smaller matrix (so-called 'filter') along the inputted matrix and calculate the values in the resulting output matrix using the mathematical operation *convolution*. This model is most notably used for image recognition.[81,97] Voting Classifiers on the other hand receive predictions from several other ML models, and 'vote' on the correct answer.[61]

5 Conclusion

Single photon quantum emitters in solid-state hosts have become a major area of research due to their wide range of applications from biomedicine to quantum information technologies. Despite ongoing research, many of these techniques suffer from practical drawbacks from requiring copious amounts of calibration, to lengthy data acquisition times. Our goal was to mitigate these shortcomings by improving performance, speed, and efficiency of data analysis. We specifically developed a set of novel techniques based on machine learning (ML) and data simulation to achieve this and we showcase their effectiveness on three examples.

In our nanothermometry study we presented an all-optical nanothermometry technique that uses fluorescent nanodiamonds which co-host two colour centers: silicon and germanium vacancies. Our technique utilizes Multi-Feature Linear Regression (MF-LR)

and is compared to those of traditional single feature models. Our motivation was mostly practical, aiming to achieve the best accuracy and resolution with the least amount of calibration points, and simultaneously, the best combination of temperature-dependent features that should be used. We found that with a subset of only 3-5 features, we can achieve an accuracy of 1.5K and a resolution of 1.1K $Hz^{-1/2}$ using a MF-LR algorithm trained on as few as 3 calibration temperatures on a set of 5 nanodiamonds. Due to the nature of the model and our analysis, this same technique can be easily applied to any other nanothermometer with at least two temperature-dependent observables or features.

In our optically detected magnetic resonance (ODMR) study we presented a custom clustering algorithm (CA) that efficiently analyzes ODMR data from quantum emitters. The algorithm is based on the standard K-means algorithm, with additional modifications to handle a wide variety of ODMR spectra. Using a combination of experimental and synthetic datasets, we showed that our model has a factor ~1.3x better accuracy and ~4.7x better resolution with a ~5x increase in efficiency when compared to traditional fitting models such as the Levenberg-Marquardt method. Our CA is therefore a powerful tool for improving accuracy, speed, and efficiency of quantum sensing applications which are based on determining the resonance frequencies of spin defects in ODMR spectra, especially those which are noisy or scarce.

In our autocorrelation study we presented avenues for ML applications on autocorrelation measurements for determining whether an optically active site is a single emitter. Although no results were produced, we laid out the groundwork for supervised ML model training through the ability to generate datasets based on the well-known electronic transition rates in the samples discussed. We also discussed two models, Convolutional

Neural Networks and Voting Classifiers, which have shown to perform considerably better than traditional fitting, based on statistical inference such as the Levenberg-Marquardt method, aligning with our findings for both our nanothermometry and ODMR studies.

Overall, we have demonstrated that ML shows promise for improving all manor of experimental analysis; from improvements to performance metrics such as accuracy, resolution, and sensitivity to dramatic reductions in the amount of data acquisition required to achieve results equal to—or better than—those of traditional methods. With similar levels of information and noise, our ML models outperformed their traditional, statistical inferencing counterparts, specifically in cases with increased noise or data scarcity, allowing for greater flexibility and general applicability in real world experimental research and analysis.

6 Bibliography

1. Awschalom, D. D., Hanson, R., Wrachtrup, J. & Zhou, B. B. Quantum technologies with optically interfaced solid-state spins. *Nature Photon* **12**, 516–527 (2018).

2. Atatüre, M., Englund, D., Vamivakas, N., Lee, S.-Y. & Wrachtrup, J. Material platforms for spin-based photonic quantum technologies. *Nat Rev Mater* **3**, 38–51 (2018).

3. Shiue, R.-J. *et al.* Active 2D materials for on-chip nanophotonics and quantum optics. *Nanophotonics* **6**, 1329–1342 (2017).

4. Aharonovich, I., Englund, D. & Toth, M. Solid-state single-photon emitters. *Nature Photon* **10**, 631–641 (2016).

5. Doherty, M. W. *et al.* The nitrogen-vacancy colour centre in diamond. *Physics Reports* **528**, 1–45 (2013).

6. Bradac, C., Gao, W., Forneris, J., Trusheim, M. E. & Aharonovich, I. Quantum nanophotonics with group IV defects in diamond. *Nat Commun* **10**, 5625 (2019).

7. Stone, D. G., Chen, Y., Ekimov, E. A., Tran, T. T. & Bradac, C. Diamond Nanothermometry Using a Machine Learning Approach. *ACS Appl. Opt. Mater.* **1**, 898–905 (2023).

8. Becker, J. N. & Neu, E. Chapter Seven - The silicon vacancy center in diamond. in *Semiconductors and Semimetals* (eds. Nebel, C. E., Aharonovich, I., Mizuochi, N. & Hatano, M.) vol. 103 201–235 (Elsevier, 2020).

9. Iwasaki, T. *et al.* Germanium-Vacancy Single Color Centers in Diamond. *Sci Rep* **5**, 12882 (2015).

10. Castelletto, S. & Boretti, A. Silicon carbide color centers for quantum applications. *J. Phys. Photonics* **2**, 022001 (2020).

11. Kim, S. H. *et al.* Color Centers in Hexagonal Boron Nitride. *Nanomaterials* **13**, 2344 (2023).

12. Tran, T. T., Bray, K., Ford, M. J., Toth, M. & Aharonovich, I. Quantum emission from hexagonal boron nitride monolayers. *Nature Nanotech* **11**, 37–41 (2016).

13. Gottscholl, A. *et al.* Spin defects in hBN as promising temperature, pressure and magnetic field quantum sensors. *Nat Commun* **12**, 4480 (2021).

14. Gong, R. *et al.* Coherent dynamics of strongly interacting electronic spin defects in hexagonal boron nitride. *Nat Commun* **14**, 3299 (2023).

15. Stone, D. G., Whitefield, B., Kianinia, M. & Bradac, C. Fast Characterization of Optically Detected Magnetic Resonance Spectra via Data Clustering. *J. Phys. Chem. C* **128**, 13147–13154 (2024).

16. Gottscholl, A. *et al.* Initialization and read-out of intrinsic spin defects in a van der Waals crystal at room temperature. *Nat. Mater.* **19**, 540–545 (2020).

17. Morfa, A. J. *et al.* Single-Photon Emission and Quantum Characterization of Zinc Oxide Defects. *Nano Lett.* **12**, 949–954 (2012).

18. Ter-Mikirtychev, V. V. Optical Spectroscopy of Rare-Earth Ions in the Solid State. in *Encyclopedia of Spectroscopy and Spectrometry (Third Edition)* (eds. Lindon, J. C., Tranter, G. E. & Koppenaal, D. W.) 481–491 (Academic Press, Oxford, 2017).

19. Utikal, T. *et al.* Spectroscopic detection and state preparation of a single praseodymium ion in a crystal. *Nat Commun* **5**, 3627 (2014).

20. Eichhammer, E., Utikal, T., Götzinger, S. & Sandoghdar, V. Spectroscopic detection of single Pr3+ ions on the 3H4−1D2 transition. *New J. Phys.* **17**, 083018 (2015).

21. Mi, X. *et al.* A coherent spin–photon interface in silicon. *Nature* **555**, 599–603 (2018).

22. Tosi, G., Mohiyaddin, F. A., Huebl, H. & Morello, A. Circuit-quantum electrodynamics with direct magnetic coupling to single-atom spin qubits in isotopically enriched 28Si. *AIP Advances* **4**, 087122 (2014).

23. Togan, E. *et al.* Quantum entanglement between an optical photon and a solid-state spin qubit. *Nature* **466**, 730–734 (2010).

24. Hanson, R. & Awschalom, D. D. Coherent manipulation of single spins in semiconductors. *Nature* **453**, 1043–1049 (2008).

25. Pfaff, W. *et al.* Unconditional quantum teleportation between distant solid-state quantum bits. *Science* **345**, 532–535 (2014).

26. Hensen, B. *et al.* Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres. *Nature* **526**, 682–686 (2015).

27. W. J. Munro, K. Azuma, K. Tamaki and K. Nemoto, Inside Quantum Repeaters. *IEEE Journal of Selected Topics in Quantum Electronics*, **21**, 78-90 (2015).

28. Bernien, H. *et al.* Heralded entanglement between solid-state qubits separated by three metres. *Nature* **497**, 86–90 (2013).

29. Lavroff, R. H. *et al.* Recent Innovations in Solid-State and Molecular Qubits for Quantum Information Applications. *J. Phys. Chem. A* **125**, 9567–9570 (2021).

30. Wolfowicz, G. *et al.* Quantum guidelines for solid-state spin defects. *Nat Rev Mater* **6**, 906–925 (2021).

31. Gardas, B., Dziarmaga, J., Zurek, W. H. & Zwolak, M. Defects in Quantum Computers. *Sci Rep* **8**, 4539 (2018).

32. Weber, J. R. *et al*. Quantum computing with defects. *Proceedings of the National Academy of Sciences* **107**, 8513-8518 (2010).

33. Cai, J., Retzker, A., Jelezko, F. & Plenio, M. B. A large-scale quantum simulator on a diamond surface at room temperature. *Nature Phys* **9**, 168–173 (2013).

34. Buluta, I., & Nori, F. Quantum Simulators. *Science* **326**, 108-111 (2009).

35. Vaidya, S., Gao, X., Dikshit, S., Aharonovich, I. & Li, T. Quantum sensing and imaging with spin defects in hexagonal boron nitride. *Advances in Physics: X* **8**, 2206049 (2023).

36. Degen, C. L., Reinhard, F. & Cappellaro, P. Quantum sensing. *Rev. Mod. Phys.* **89**, 035002 (2017).

37. Bradac, C., Lim, S. F., Chang, H.-C. & Aharonovich, I. Optical Nanoscale Thermometry: From Fundamental Mechanisms to Emerging Practical Applications. *Advanced Optical Materials* **8**, 2000183 (2020).

38. Brites, C. D. S. *et al.* Thermometry at the nanoscale. *Nanoscale* **4**, 4799–4829 (2012).

39. del Rosal, B., Ximendes, E., Rocha, U. & Jaque, D. In Vivo Luminescence Nanothermometry: from Materials to Applications. *Advanced Optical Materials* **5**, 1600508 (2017).

40. Jaque, D. & Vetrone, F. Luminescence nanothermometry. *Nanoscale* **4**, 4301–4326 (2012).

41. Quintanilla, M. & Liz-Marzán, L. M. Guiding Rules for Selecting a Nanothermometer. *Nano Today* **19**, 126–145 (2018).

42. Liang, Z., Wu, J., Cui, Y., Sun, H. & Ning, C.-Z. Self-optimized single-nanowire photoluminescence thermometry. *Light Sci Appl* **12**, 36 (2023).

43. Carrasco, E. *et al.* Intratumoral Thermal Reading During Photo-Thermal Therapy by Multifunctional Fluorescent Nanoparticles. *Advanced Functional Materials* **25**, 615–626 (2015).

44. Tsuji, T., Ikado, K., Koizumi, H., Uchiyama, S. & Kajimoto, K. Difference in intracellular temperature rise between matured and precursor brown adipocytes in response to uncoupler and β-adrenergic agonist stimuli. *Sci Rep* **7**, 12889 (2017).

45. Tsai, P.-C. *et al.* Measuring Nanoscale Thermostability of Cell Membranes with Single Gold–Diamond Nanohybrids. *Angewandte Chemie* **129**, 3071–3076 (2017).

46. Sotoma, S., Epperla, C. P. & Chang, H.-C. Diamond Nanothermometry. *ChemNanoMat* **4**, 15–27 (2018).

47. Nakano, M. & Nagai, T. Thermometers for monitoring cellular temperature. *Journal of Photochemistry and Photobiology C: Photochemistry Reviews* **30**, 2–9 (2017).

48. Zhou, H., Sharma, M., Berezin, O., Zuckerman, D. & Berezin, M. Y. Nanothermometry: From Microscopy to Thermal Treatments. *ChemPhysChem* **17**, 27–36 (2016).

49. Bai, T. & Gu, N. Micro/Nanoscale Thermometry for Cellular Thermal Sensing. *Small* **12**, 4590–4610 (2016).

50. Wu, J., Kwok, T., Li, X. *et al.* Mapping three-dimensional temperature in microfluidic chip. *Sci Rep* **3**, 3321 (2013).

51. Yue, Y. & Wang, X. Nanoscale thermal probing. *Nano Reviews* **3**, 11586 (2012).

52. Kamei, Y. *et al.* Infrared laser–mediated gene induction in targeted single cells in vivo. *Nat Methods* **6**, 79–81 (2009).

53. Kumar, S. V. & Wigge, P. A. H2A.Z-Containing Nucleosomes Mediate the Thermosensory Response in *Arabidopsis*. *Cell* **140**, 136–147 (2010).

54. Schroeder, A. *et al.* Treating metastatic cancer with nanotechnology. *Nat Rev Cancer* **12**, 39–50 (2012).

55. O'Neal, D. P., Hirsch, L. R., Halas, N. J., Payne, J. D. & West, J. L. Photo-thermal tumor ablation in mice using near infrared-absorbing nanoparticles. *Cancer Letters* **209**, 171–176 (2004).

56. Chen, Y. *et al.* Optical Thermometry with Quantum Emitters in Hexagonal Boron Nitride. *ACS Appl. Mater. Interfaces* **12**, 25464–25470 (2020).

57. Engineering, N. A. of. *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2002 NAE Symposium on Frontiers of Engineering*. (National Academies Press, 2003).

58. Sartison, M., Ibarra, O. C., Caltzidis, I., Reuter, D. & Jöns, K. D. Scalable integration of quantum emitters into photonic integrated circuits. *Mater. Quantum. Technol.* **2**, 023002 (2022).

59. Kane, B. E. A silicon-based nuclear spin quantum computer. *Nature* **393**, 133–137 (1998).

60. Labrador-Páez, L. *et al.* Reliability of rare-earth-doped infrared luminescent nanothermometers. *Nanoscale* **10**, 22319–22328 (2018).

61. Kudyshev, Z.A. *et al.* Rapid Classification of Quantum Sources Enabled by Machine Learning. *Adv. Quantum Technol* **3**, 2000067 (2020).

62. Fishman, R. E. K., Patel, R. N., Hopper, D. A., Huang, T.-Y. & Bassett, L. C. Photon-Emission-Correlation Spectroscopy as an Analytical Tool for Solid-State Quantum Defects. *PRX Quantum* **4**, 010202 (2023).

63. Neu, E., Agio, M. & Becher, C. Photophysics of single silicon vacancy centers in diamond: implications for single photon emission. *Opt. Express, OE* **20**, 19956–19971 (2012).

64. Tchernij, S. D. *et al.* Single-Photon-Emitting Optical Centers in Diamond Fabricated upon Sn Implantation. *ACS Photonics* **4**, 2580–2586 (2017).

65. Choi, S., Agafonov, V. N., Davydov, V. A. & Plakhotnik, T. Ultrasensitive All-Optical Thermometry Using Nanodiamonds with a High Concentration of Silicon-Vacancy Centers and Multiparametric Data Analysis. *ACS Photonics* **6**, 1387–1392 (2019).

66. Chan, V. W. L., Pisutha-Arnond, N. & Thornton, K. Phase-field crystal model for a diamond-cubic structure. *Phys. Rev. E* **91**, 053305 (2015).

67. Lühmann, T. *et al.* Screening and engineering of colour centres in diamond. *J. Phys. D: Appl. Phys.* **51**, 483002 (2018).

68. Tran, T. T. *et al.* Anti-Stokes excitation of solid-state quantum emitters for nanoscale thermometry. *Science Advances* **5**, eaav9180 (2019).

69. Neumann, P. *et al.* Excited-state spectroscopy of single NV defects in diamond using optically detected magnetic resonance. *New J. Phys.* **11**, 013017 (2009).

70. Landry, G. & Bradac, C. Efficient characterization of blinking quantum emitters from scarce data sets via machine learning. *Mater. Quantum. Technol.* **4**, 015403 (2024).

71. Berry, M. W., Mohamed, A. & Yap, B. W. *Supervised and Unsupervised Learning for Data Science*. (Springer International Publishing, Cham, 2020).

72. Shalev-Shwartz, S. & Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*. (Cambridge University Press, 2014).

73. Stone, D. G. & Bradac, C. Machine and quantum learning for diamond-based quantum applications. *Mater. Quantum. Technol.* **3**, 012001 (2023).

74. Chapelle, O., Schölkopf, B., & Zien, A. *Semi-Supervised Learning*. (MIT Press, 2010).

75. Kirchner, J., Heberle, A. & Lowe, W. Classification vs. Regression - Machine Learning Approaches for Service Recommendation Based on Measured Consumer Experiences. *IEEE World Congress on Services* 278-285 (2015).

76. Chauhan, R., Ghanshala, K. K. & Joshi, R. C. Convolutional Neural Network (CNN) for Image Detection and Recognition. *First International Conference on Secure Cyber Computing and Communication* 278–282 (2018).

77. Murphy, K. P. *Machine Learning: A Probabilistic Perspective*. (MIT Press, 2012).

78. Lubis, F. F., Rosmansyah, Y. & Supangkat, S. H. Gradient descent and normal equations on cost function minimization for online predictive using linear regression with multiple variables. *International Conference on ICT For Smart Society* 202–205 (2014).

79. Bishop, C. M. Neural networks and their applications. *Review of Scientific Instruments* **65**, 1803–1832 (1994).

80. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).

81. Albawi, S., Abed Mohammed, T. & Alzawi, S. Understanding of a Convolutional Neural Network. *International Conference on Engineering and Technology* 1-6 (2017).

82. Arthur, D. & Vassilvitskii, S. K-means++: the advantages of careful seeding. *Society for Industrial and Applied Mathematics* 1027–1035 (2007).

83. Kmeans++ Documentation - Scikit-Learn. *scikit-learn* https://scikit-learn/stable/modules/clustering.html.

84. Bzdok, D., Altman, N. & Krzywinski, M. Statistics versus machine learning. *Nature Methods* **15**, 233–234 (2018).

85. scipy.optimize.least_squares — SciPy v1.13.1 Manual. https://docs.scipy.org/doc/scipy-1.13.1/reference/generated/scipy.optimize.least_squares.html#scipy.optimize.least_squares.

86. Moré, J. J. The Levenberg-Marquardt algorithm: Implementation and theory. in *Numerical Analysis* (ed. Watson, G. A.) 105–116 (Springer, Berlin, Heidelberg, 1978).

87. Ranganathan, A. The Levenberg-Marquardt Algorithm. *Tutoral on LM algorithm* **11** 101-110 (2004).

88. Optimization and root finding (scipy.optimize) — SciPy v1.14.0 Manual. https://docs.scipy.org/doc/scipy/reference/optimize.html.

89. curve_fit Documentation - scipy.optimize. *SciPy* https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html.

90. KMeans. *scikit-learn* https://scikit-learn/stable/modules/generated/sklearn.cluster.KMeans.html.

91. Saputra, D. M., Saputra, D. & Oswari, L. D. Effect of Distance Metrics in *Determining K-Value in K-Means Clustering Using Elbow and Silhouette Method*. 341–346 (Atlantis Press, 2020).

92. Shutaywi, M. & Kachouie, N. N. Silhouette Analysis for Performance Evaluation in Machine Learning with Applications to Clustering. *Entropy* **23**, 759 (2021).

93. Gao, M. *et al.* Identification Method of Electrical Load for Electrical Appliances Based on K-Means ++ and GCN. *IEEE Access* **9**, 27026–27037 (2021).

94. Lindon, J. C., Holmes, E. & Nicholson, J. K. Pattern recognition methods and applications in biomedical magnetic resonance. *Progress in Nuclear Magnetic Resonance Spectroscopy* **39**, 1–40 (2001).

95. Dimitriadou, E., Barth, M., Windischberger, C., Hornik, K. & Moser, E. A quantitative comparison of functional MRI cluster analysis. *Artificial Intelligence in Medicine* **31**, 57–71 (2004).

96. Balasubramanian, G. *et al.* Nanoscale imaging magnetometry with diamond spins under ambient conditions. *Nature* **455**, 648–651 (2008).

97. Liao, Y.-W. *et al.* Deep-Learning-Enhanced Single-Spin Readout in Silicon Carbide at Room Temperature. *Phys. Rev. Appl.* **17**, 034046 (2022).

A Additional ODMR Results

This section contains additional results from the ODMR section (4.2) that were not included in the main results as they did not meet our minimum requirements and were ultimately abandoned in favour of alternatives.

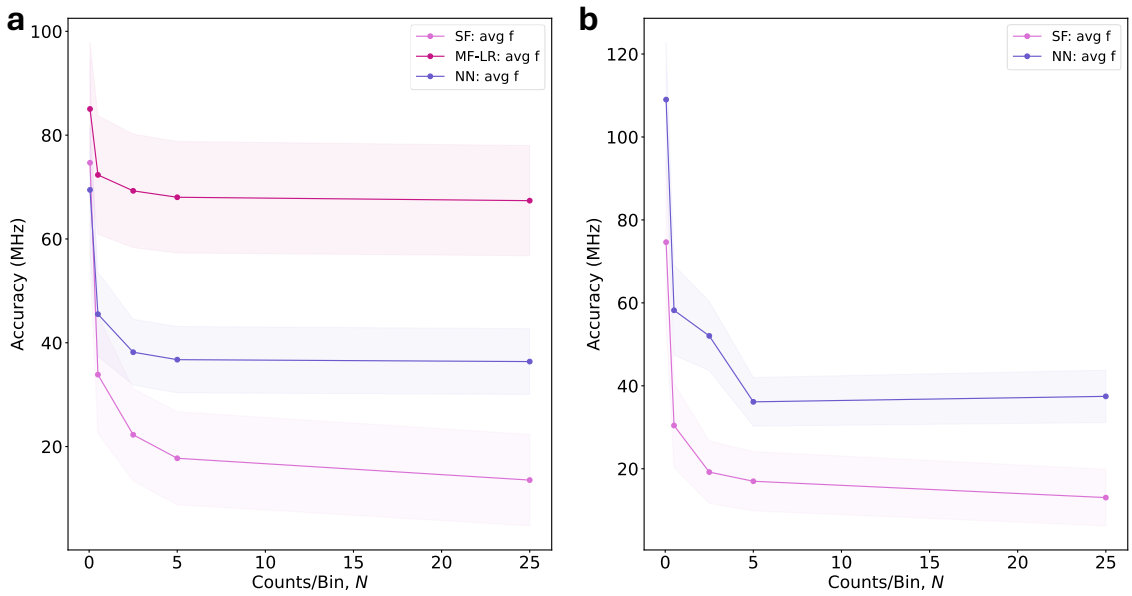

Figure A.1: Operation and performance of SF, MF-LR, and NN for ODMR. **a-b)** Accuracy of each model (data points), with the shaded regions indicating the precision (standard deviation in accuracy). The shaded areas have been divided by five to improve visual clarity. Panels **a)** and **b)** differ as they show performance when applied to data sets with variation in PL contrast between the two ODMR peaks of up to 15% and 65% respectively. Note: original testing involved exploring a much wider range of N then what is shown in section 4.2, therefore the SF shown here was a previous run on different but similarly generated datasets, using the same SF model. Also, as mentioned in the main results section, the SF model struggled to converge for large portions of tested data for very small values of N, therefore the results shown include only those that managed to converge.

B Code and Discussion

This study used Python v3.9 and a myriad of accompanying libraries to perform data simulations and analysis. The following sections contain a discussion of modules and libraries used in this study as well as some of the source code or references to where it can be accessed.

B.1 Modules and Libraries

Many modules and libraries were imported into Python during this study, the following is a brief list of some of the major libraries and how they were used:

1. **NumPy** and **Pandas** were used for basic data handling, processing, and mathematical operations with arrays and data frames, respectively.
2. **Matplotlib.pyplot** and **Seaborn** were used to generate line graphs and heatmaps respectively.
3. **SciPy's optimize.curve_fit()** was used for all traditional statistical inferencing based curve fitting such as the ODMR fit in section 3.2.
4. **Scikit-learn** was the source of all the machine learning algorithms used in this study.
    i. **sklearn.model_selection.train_test_split** was used for any *supervised* models to randomly separate the data into training and testing sets.
    ii. **sklearn.linear_model.LinearRegression** was used for the MF-LR algorithm, with **sklearn.model_selection.mean_square_error** being used as the loss function.
    iii. **sklearn.cluster.Kmeans** was used in the creation of the custom clustering algorithm.

iv. **Sklearn.neural_network.MLPRegressor** was used for the ANN algorithm, with **sklearn.model_selection.GridSearchCV** incorporated to help select the best hyperparameters for the network.

B.2 Custom CA and Synthetic ODMR Data Access

A publicly available GitHub repository has been created which contains the source code for the custom clustering algorithm along with its accompanying ODMR data simulation code. The repository contains information about how the functions work, some basic instructions on how to use it, as well as some example simulated data. All of this can be found here: https://github.com/DylanStone2000/Fast-characterization-of-optically-detected-magnetic-resonance-spectra-via-data-clustering

B.3 Synthetic Autocorrelation Code

The following Python function was designed based on the work of Kudyshev et. al. to simulate the real-world experimental process of taking a second order autocorrelation measurement. The function contains many variables that can be customized to simulate specific kinds of emitters using their known transition rates. For the sake of illustration, the variables have been set to those provided in *Table 3.2*. In its current state it is setup to take in a run time (i.e. how long in seconds you want the simulation to run for), a list/array of $g^2(t = 0)$ values to simulate the data around, as well as the number of emitters. The function can be easily modified to allow for multiple run times and/or number of emitters to help automate the creation of many datasets if one wishes. An explanation of the simulation can be found in section 3.4.

```
def Sim_g2(g0, run_time, num_emitters=1):
    """
    This function generates synthetic autocorrelation datasets. N datasets are
    generated, where N is the length of the g0 array/list. The function can
    be easily modified to accept multiple run times or number of emitters
    instead/aswell.

    Parameters
    ----------
    g0 : array/list of float64s
        List of values to use for the g(t=0) values when generating data.
    run_time : int32
        How many seconds to run the simulation loop for.
    num_emitters : int32, optional
        The number of emitters in the simulated dataset. The default is 1.

    Returns
    -------
    data : DataFrame
        DataFrame containing columns of the bin counts for each run.

    """

    tmax = 4*10**(-7)    # s
    dt = 2.34*10**(-9)   # s
    Nbins = 215
    gammaEG = 2*10**7    # Hz or 50 ns
    gammaGE = gammaEG
    gammaEM = 10**7      # Hz
    gammaMG = 7*10**6    # Hz
    R = 1/(20*Nbins)
    n0 = 0

    lam1 = gammaEG + gammaGE
    lam2 = gammaMG + gammaEM*gammaGE/lam1
    a = lam2/gammaMG - 1

    n = 0

    data = pd.DataFrame()

    for val in g0:
        r = 1 - np.sqrt(1 - val)
        bin_counts = np.zeros(171) # tmax/dt ~ 171 (rounded up)
        start_time = time.time()

        while (time.time() - start_time) < run_time:
            if n < tmax/dt:
                if n == 0:
                    prob = R*r + R*(1-r)*(1 - (1+a)*np.exp(-(1/4)*dt*lam1)\
                                          + a*np.exp(-(1/4)*dt*lam2))
                else:
                    prob = R*r + R*(1-r)*(1 - (1+a)* \

                           np.exp(-np.abs(n-n0+0.5)*dt*lam1) + \
```

```
                        a*np.exp(-np.abs(n-n0+0.5)*dt*lam2))

            # if there are multiple emitters, give it multiple chances
            for i in range(num_emitters):
                if np.random.uniform(0,1) < prob:
                    bin_counts[n] += 1
                    n = 0

            else:
                n += 1
        else:
            n = 0

    print(val)
    data[str(val)] = bin_counts

return data
```