Using Machine Learning to Optimise Atomic Physics Experiments

Alfie Renn
Level 4 Project, MPhys Physics
Supervisors: Prof Simon Cornish, Dr Sarah Bromley
Department of Physics, Durham University

April 19, 2022

Machine learning is emerging as a highly effective tool for automating common optimisation tasks in atomic physics laboratories. This work implements Gaussian Process machine learning in such a laboratory at Durham University. We aim to understand the difficulties that come with connecting a machine learner to physical experiments and to contrast the efficacy of machine learning with manual optimisation. We also contrast the optimisation speed of machine learning compared to the classical optimisation algorithm differential evolution. In this work we investigate the machine learning technique of Gaussian processes by integrating the python package M-LOOP (Machine-Learning Online Optimisation Package) into two separate atomic physics experiments. Firstly, a simple experiment is devised involving the use of two waveplates to remove an elliptical polarisation from a laser beam. With this setup, it is found that Gaussian processes are a more powerful tool than classical regression, especially when there are unknown systematic errors in play. Secondly, the density of cold atomic clouds of Cesium is maximised via the optimisation of an evaporation sequence. In particular, the atomic scattering length is a variable parameter, allowing novel control of the experiment not available when using other alkali atoms. With both experiments, the machine learning approach achieves a solution much faster than the classical alternative, and the solutions returned by the machine learner are more optimal than the human optimisation. Overall, it is demonstrated that machine learning is straightforward to implement in atomic physics laboratories, and that it provides efficient goal-oriented experimental optimisation.

Contents

1 Introduction and overview of online optimisation

Machine learning has been used to characterise large datasets for a long time [1–3], but use of online optimisation is something which is increasing in use in experimental physics and engineering [4–7], and especially within atomic physics [8–11]. For example, work has been done in Durham laboratories using online optimisation to facilitate the alignment of lasers into optical fibers [12]. Online optimisation involves computational control of an experiment, with a feedback loop allowing the computer to repeat the experiment, and to optimise a particular experimental output, such as laser intensity. This involves finding the minimum or maximum value of this output, for which we need optimisation routines.

There are a number of classical routines available to someone wishing to find a minimum in a dataset, such as differential evolution — a genetic algorithm with high versatility — which is described in more detail in section 2. However, machine learning techniques are becoming more prevalent, such as Gaussian process regression. A Gaussian process is a Bayesian statistical technique able to represent an N-dimensional function, and by describing an experiment with a cost (e.g., laser intensity) one can model this cost-function over the many input parameters and find its minimum. This is the operating principle of Gaussian process machine learning.

Given a set of parameters which describe a model of the experiment, represented in figure 1 as a set of sliders, we can create a Gaussian Process. This can be used to estimate the cost of the experiment for a set of input parameters, and by comparing this to the result from running the real experiment, the parameters can be tweaked via a feedback loop. With enough repeats, the model will accurately represent the experiment and thus a minimum of cost can be found, representing a favourable set of experimental parameters which will result in an optimised experiment.

1.1 Motivation

Machine learning is a quickly growing topic of research (illustrated in figure 23 in the appendix). This report is an exploration of how useful machine learning in the context of online optimisation can be in an optics-laboratory setting, and specifically how one can use M-LOOP to straightforwardly automate the optimisation of experimental parameters.

In the laboratory in Durham, M-LOOP was introduced into a simple polarisation experiment in order to gain insight into online optimisation itself, and it was also introduced into a complex experiment, to test if computer optimisation could improve the performance of the experiment more than a human optimiser could. This report outlines how the online optimisation process works, the machine learning routine it uses, and a classical routine to compare this to. Also outlined are the simple and complex experiments mentioned above and finally the results of this optimisation.

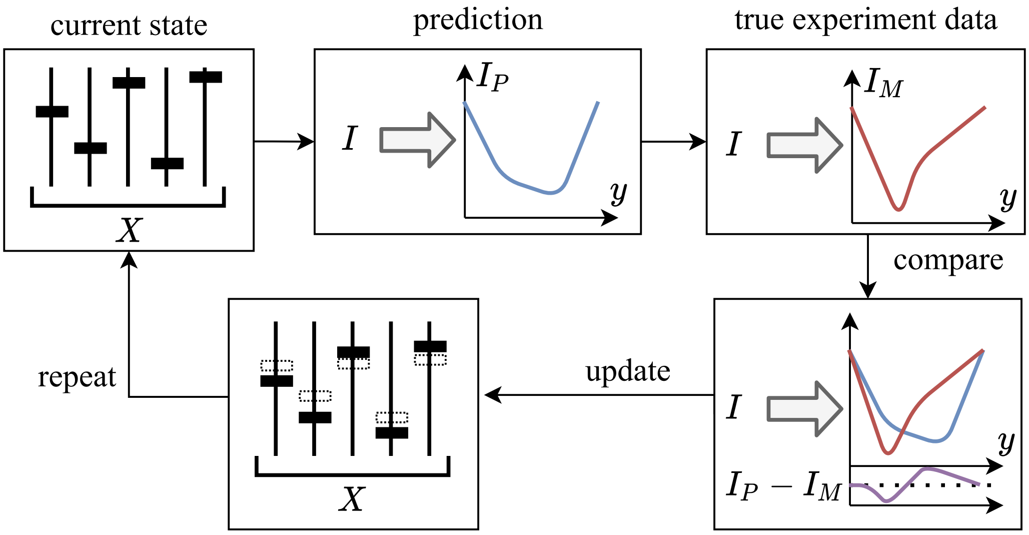

Figure 1: A description of a machine learning online optimisation process. Here the machine learner is represented by a set of parameters X, which make a prediction I_P of an experimental output over an input range y. By conducting the actual experiment and obtaining a measurement I_M, and comparing this with the prediction (purple line), the machine learner's parameters are updated to improve future predictions. This process is repeated to further improve the machine learner until it reaches a desired level of accuracy.

2 Classical and machine learning online optimisation

The first step to using online optimisation with an experiment is that the experiment must be computer-controllable. Such an automatic experiment is initially described, giving context to the following sections which describe the classical (differential evolution) and machine learning (Gaussian process) online optimistaion algorithms used in the rest of this work.

2.1 Experimental context: the rotation mount

In this section, a simple one-dimensional experiment is layed out, of which the setup is shown in figure 3. The experiment is described thusly: linearly polarised light (i.e., laser light) is shone through a rotatable $\lambda/2$ waveplate, which causes a rotation of the polarisation of the light. This rotated light then passes through a polarising beamsplitter before being incident upon a photodetector. Thus, the photodiode exhibits a maximum intensity when the waveplate is oriented such that the linearly polarised light is rotated to be aligned with the polarising beamsplitter. There is then minimum intensity when the opposite is true, corresponding to a difference in rotation of 45°.

Experimentally, the waveplate is rotated using a mechanical system composing of a worm-drive and a hollow geared housing. A schematic of this is shown in figure 2. By changing the number of teeth on the central gear, or the winding of the worm drive, the gear ratio can be changed — and therefore the rotational precision. In this report, two rotation mounts are used, with angular precisions of 0.1° and 0.05° per step of rotation [14]. Using these devices, we

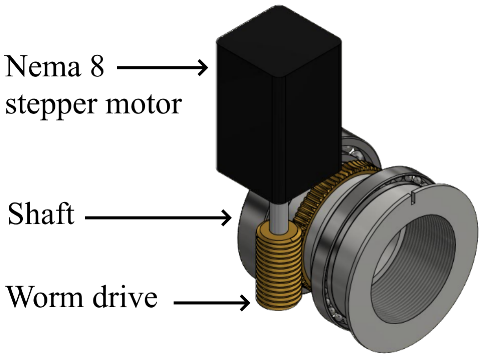

Figure 2: The rotation mount. Waveplates are inserted into the central housing, which is rotated when the stepper motor is powered, turning the worm drive. The stepper motor is controlled via PSoC microelectronics [13], connected to a PC. Schematic from [14].

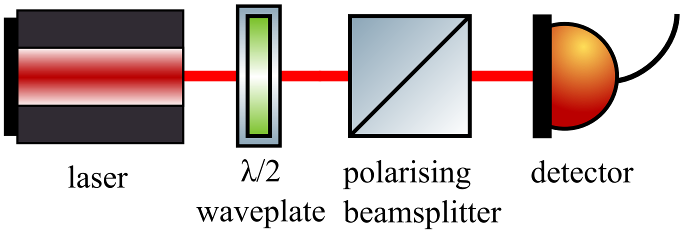

Figure 3: The simple experimental setup. A linearly polarised laser beam passes through a half-waveplate, and the intensity of the resulting laser in one linearly polarised direction is measured by a photodiode detector. The half-waveplate is mounted in a rotation mount from figure 2, allowing automatic control of it.

can control a simple experiment, allowing online optimisation to be tested simply and quickly, before attempting the optimisation of a more complex experiment.

To aid the following explanations of online optimisation algorithms, it is useful to have an analytical description of the experiment to be optimised. With a more complex experiment (e.g., section 4) an analytical solution is not derivable, but with polarisation optics one can be devised. The laser intensity I into the photodiode as a function of waveplate rotation θ for the setup in figure 3 is given by (derivation found in section 3),

$$I \propto \sin^2(2\theta). \tag{1}$$

Equation 1 is shown as a curve in figure 4, where a zero-degree rotation corresponds to the minimum intensity, i.e., when the polarisation of the laser light is anti-aligned with the beamsplitter cube. This figure will be used to provide context to the online optimisation processes in the remainder of this section.

2.2 Differential evolution: an adaptive brute-force approach

Differential evolution is a method of evolutionary optimisation, which is able to find the minimum of a multidimensional function without requiring any gradient information [15]. The algorithm involves a group of candidate solutions, which are evolved by testing new mutated candidates and replacing the old candidates with the new ones which result in an improved quality metric. Over time, the population evolves to become a group of good quality individuals, allowing a global optimum to be reliably found.

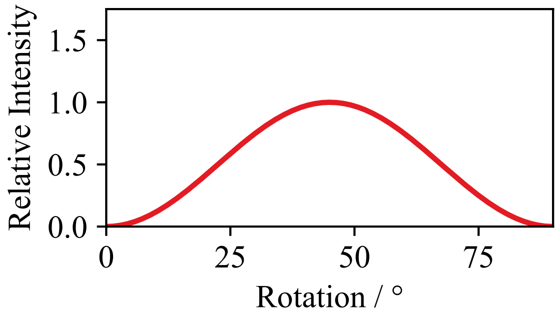

Figure 4: The relative intensity against the rotation angle of the half-waveplate. Zero intensity is chosen to be at zero degrees rotation. The functional form of the intensity curve is from equation 1.

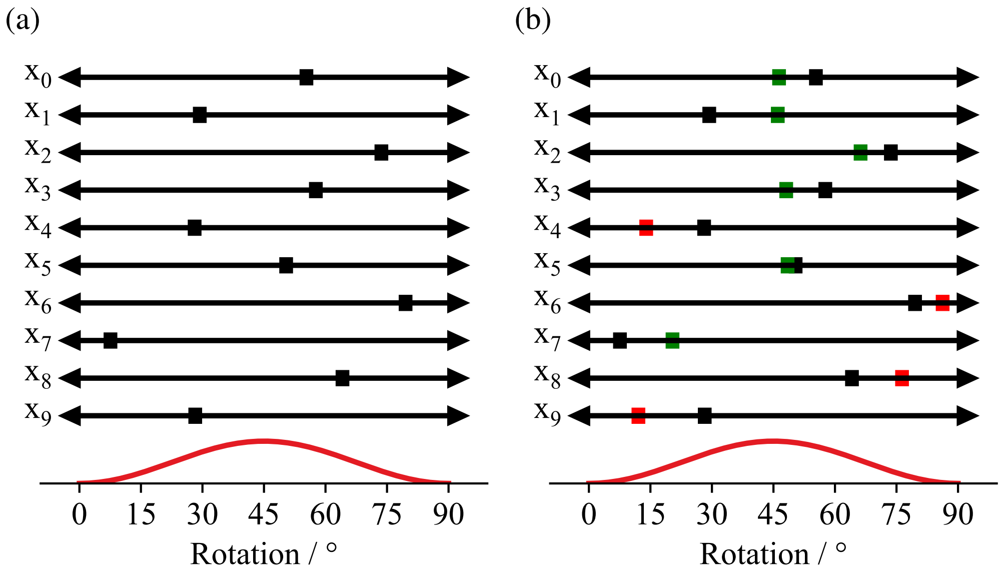

Figure 5: An illustration of the differential evolution process. Each individual x_i in the total population $\{\vec{x}\}$ mutates their rotation by a random amount. If the resulting intensity (red solid line) is higher, this new rotation value is retained (green squares). If not, it is discarded (red squares). This figure shows one iteration of the differential evolution process.

The sliders in figure 5 represents a set of candidates for the experiment outlined in section 2.1, where each candidate contains a rotation value between 0°and 90°. The measure of quality is the intensity of the light passing through the beamsplitter for a particular angle, shown at the bottom of figure 5 (a copy of figure 4). One iteration of the algorithm is then as follows [15]:

For each individual:

1. Mutate the individual by a random amount (see section 2.2.1 for more information on this).

2. Obtain a quality measure for the mutated individual.

3. Replace the old individual with the mutated individual if the quality measure is better (shown in green on figure 5).

By repeating this process, the candidate solutions are evolved, and progress towards each generation being on average better than the last.

As well as differential evolution not requiring gradient information, it is also simplified in that it only needs three control parameters which the user can control. These are described in section 2.2.1. It is because of this simplistic efficacy that we will use differential evolution as a classical optimisation technique to compare against our machine learning technique of Gaussian processes.

2.2.1 Extension to N dimensions

In contrast to the one-dimensional case above, explained here is the N-dimensional case of differential evolution optimisation [15]. Let $\{\vec{x}_i\}$ be the set of M individuals which are to be optimised. This population size is the first user-controlled parameter. Let $f(\vec{x}_i)$ be the quality measure of a single set of parameters $\vec{x}_i$, where $\vec{x}_i$ is the set of parameters of the i^{th} individual. There are M individuals, and each individual is of size N. For each individual, a mutant vector $\vec{v}$ is created like so,

$$\vec{v} = \vec{x}_{r1} + F(\vec{x}_{r2} - \vec{x}_{r3}), \tag{2}$$

where $\vec{x}_{r1}$, $\vec{x}_{r2}$ and $\vec{x}_{r3}$ are different unique random individuals from the population $\{\vec{x}_i\}$, and F is a scaling factor, the second user-controlled parameter. Each element of the individual vector $\vec{x}_i$ is replaced with the corresponding element from the mutant vector $\vec{v}$ with a probability of CR, where CR $\in [0, 1]$ is the crossover probability, the third and final user-controlled parameter. By using this crossover probability, only some elements of each individual are mutated. Then, a quality metric $f(\vec{v})$ is obtained for the new mutated individual, and if this is better than the original individual, the mutated individual replaces it.

This approach is extendable to any dimensionality of input-variable space, while only requiring three user controlled parameters: the size of the population M; the mutation scaling factor F; and the mutation crossover probability CR. This makes differential evolution a simple but powerful tool for multidimensional optimisation.

2.3 The Gaussian process as a landscape estimator

Gaussian Processes are a Bayesian regression technique, often used an alternative to other machine learning techniques such as neural networks or random forests [16]. They are used to estimate the shape of an unknown function. Without machine learning like this, knowing the form of the analytical function in figure 4 to be proportional to $\sin^2(\theta)$, and given a set of measurements of the function, one could use classical minimisation techniques for regression in order to interpolate between measured points. This would involve creating a function to represent the situation, which in this case would be

$$f(\theta) = A + B\sin^2(C \cdot \theta + D), \tag{3}$$

where A, B, C, and D are constants to be determined by the regression fitting (representing an intensity offset, intensity scaling, angular scaling, and angular offset, respectively). Then, an algorithm such as BFGS [17] would be used to find values for these four parameters, and intensities for any angle θ could be estimated using equation 3. This is an example of regression of a function with a body. However, we are interested in finding a regression method which can predict a function without a body (i.e., a function with only a domain and range). This is done by using a Gaussian Process.

To represent a bodyless function over a domain, the Gaussian Process employs a population of N Gaussian distributions. Each distribution represents a point in the domain of the function, and produces an output across the range of the function. This output is represented by the mean and standard deviation of the distribution, as seen in figure 6a as a series of 'dots-and-sticks'. A Gaussian process has two hyperparameters: the length scale; and the variance. A graphical explanation of the length scale is given by figure 6, while the variance represents the maximum range of representable functions, shown in figure 6 as the standard deviation. Both of these hyperparameters — the variance σ^2 and the length scale l — describe a covariance matrix Σ, where [18,],

$$\Sigma(x_i, x_j) = \sigma^2 \exp\left(-\frac{1}{2}\left(\frac{x_i - x_j}{\ell}\right)^2\right), \tag{4}$$

which describes the correlation between two of the Gaussian distributions x_i and x_j.

Three different covariance matrices are shown in figure 6, with a range of length scales. Changing the variance changes the amplitude of the functions which can be considered, while changing the length scale changes the distance over which the function is correlated with itself, resulting in 'bumpier' functions for shorter length scales and 'smoother' functions for longer ones. Now that a method is available — using these two hyperparameters — to represent an arbitrarily shaped function, we describe the process of evolving the model based on observed data.

We evolve the model (the 'prior distribution') by forming a joint distribution with the prior and the training points, and use conditioning to find a posterior distribution [18]. This process is shown in figure 7.

Firstly, as seen in figure 7a, the prior distribution is created, with an initial covariance matrix and N Gaussian distributions with means μ_{prior}. In the case of figure 7a, $N = 15$. The analytical function, from which the training data is taken, is shown in figure 7a for context.

Next, the evolution begins, firstly by taking a sampling of random points, shown as red squares in figure 7b. These points form a distribution with means μ_{training}. The covariance matrix for prior and training points can be derived from their rotation, or x-position using equation 4. Then, a combined posterior distribution is created according to the following equations [7],

$$\overline{\mu} = \mu_{\text{prior}} - \Sigma_{12}\Sigma_{22}^{-1}\mu_{\text{training}}, \tag{5}$$

$$\overline{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}, \tag{6}$$

where $\overline{\mu}$ and $\overline{\Sigma}$ are the mean and covariance matrix of the posterior distribution, and the Σ_{ij} represent the covariance matrices of the prior or training points, where: $i = j = 1$ is the prior distribution; $i = j = 2$ is the training points; and $i = 1, j = 2$ or $i = 2, j = 1$ are the combined covariance matrices.

The posterior distribution is shown in 'dots-and-sticks' representation in figure 7b. Figure 7c shows the identical situation, but the representation is changed to a mean line and shaded confidence interval. As you can see in figure 7c, after one iteration of the process, it is uncertain

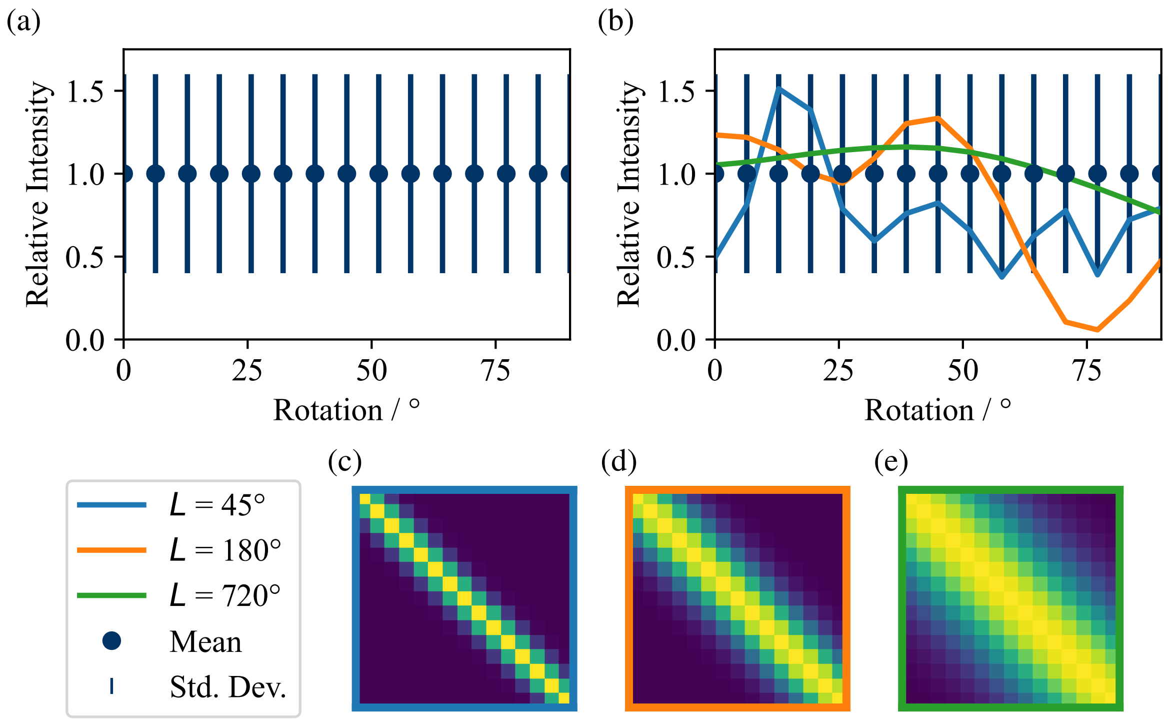

Figure 6: A graphical comparison of the Gaussian process length scale. **(a)** In 1D, a set of means and standard deviations describes the Gaussian process. This can represent any function which passes through these points. A larger variance increases these standard deviations, increasing the range of representable functions. **(b)** Each line represents a random sample from the Gaussian distribution, each with an increasing length scale (blue, orange, and green, respectively). As the length scale increases, the smoothness of the functions represented by the Gaussian process increases. **(c, d, e)** Covariance matrices for each of the length scales. An individual distribution in the Gaussian process is represented by a column in the matrix, and it is related to the other distributions by a strength proportional to the colour of the pixel in this column.

where the optimum of the function is. Thus, to reduce this uncertainty, we conduct another iteration. We choose new training points in the regions with the highest uncertainty, and again perform the same operation above to end up with a new posterior distribution. This is shown in figure 7d, with the second set of training points in green. As you can see, the positional uncertainty of the optimum is greatly reduced. This fitting is performed without knowledge of the body of the function, which is the operational basis behind the Gaussian Process online optimisation that we will be using in this work.

2.3.1 Extension to N dimensions

The Gaussian Process is extendable to a higher dimensionality, and in that situation becomes a lot more powerful relative to classical algorithms. When the parameter space becomes very large, it becomes a lot harder for these classical minimisation algorithms to function optimally as the exploration space is too large. By representing the shape of an exploration space by

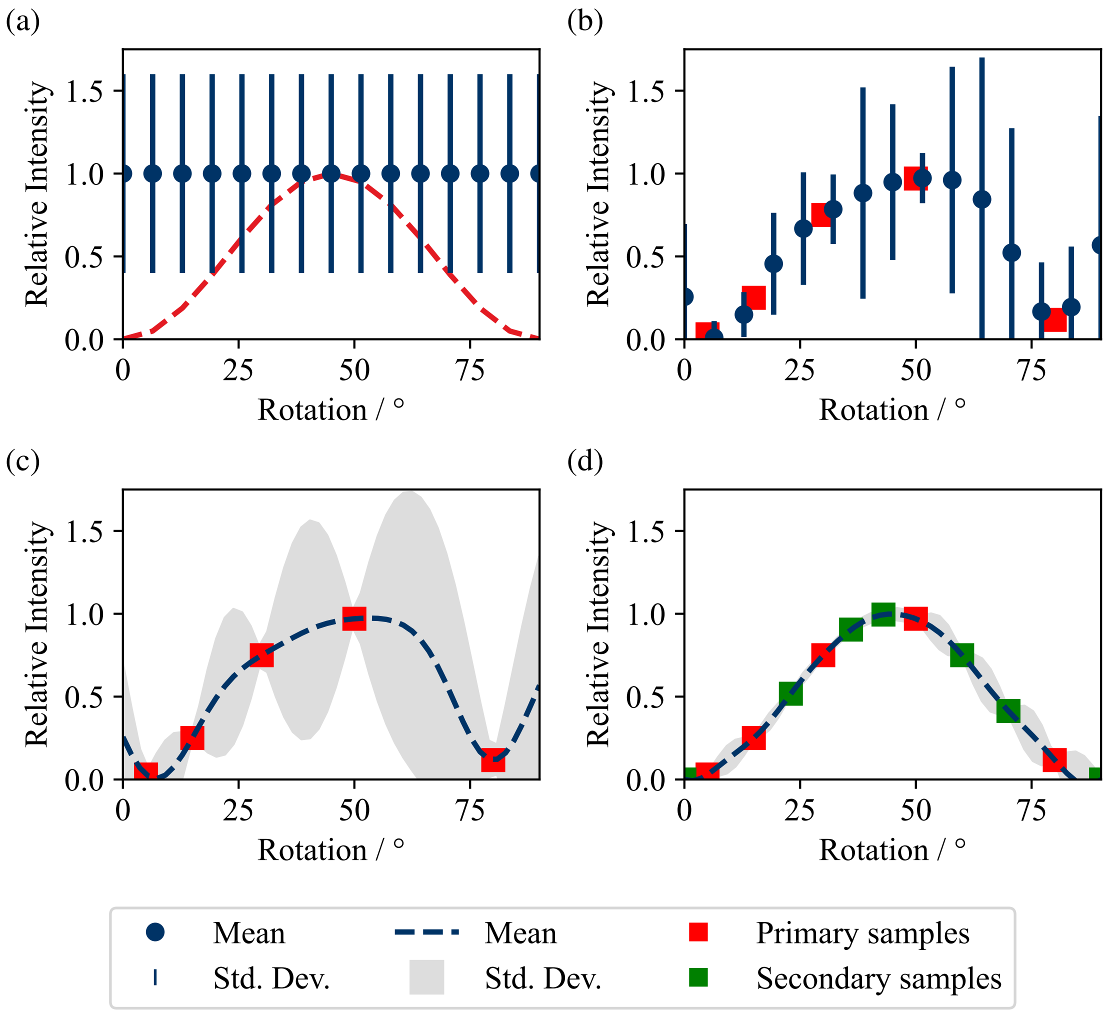

Figure 7: This figure outlines how a Gaussian process (GP) posterior distribution is formed. **(a)** The initial setup. The GP (blue). The test function (red, dashed). **(b)** Measurement of several random training points (red squares) constrains the Gaussian process. The location of the maximum is still uncertain. **(c)** Alternate representation of **b**. **(d)** Measurement of new training points (green squares) further constrains the GP. Now, a maximum of the function can be found with quick, classical minimisation algorithms, such as BFGS [19].

sampling a minimal number of points in the space, Gaussian processes mitigate this problem.

When extended into N-dimensions, Gaussian processes work the same way as in one-dimension, but the mean-vectors and covariance-matrices become higher-dimensional. An example of a 2-dimensional Gaussian Process is shown in figure 24 in the appendix. In higher dimensionality like this, visualisation of Gaussian processes becomes difficult.

2.4 M-LOOP: Machine-Learning Online Optimization Package

Next, a method is required to connect both optimisation routines described here to a physical setup. For this, we will use the python package M-LOOP [8] (Machine-Learner Online Optimization Package), which allows an online optimiser to communicate to an experiment via the

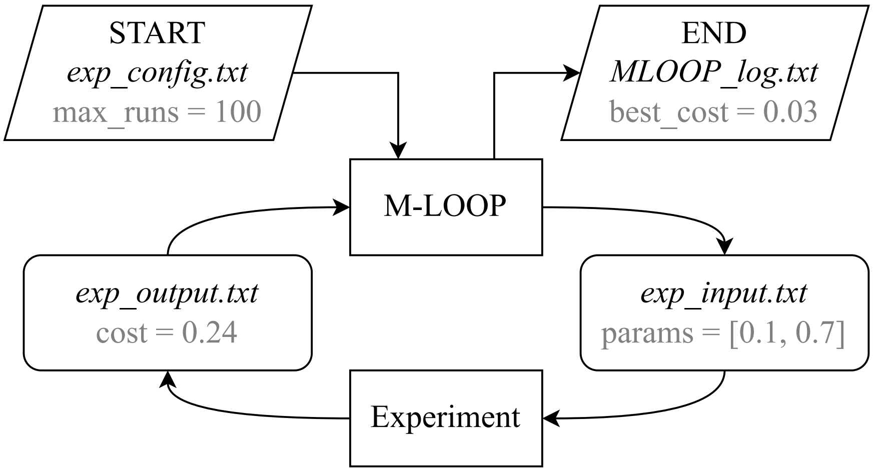

Figure 8: Flow-diagram of an M-LOOP experiment. Where a file is referenced, an example line from the file is shown underneath in grey. The process begins with a configuration file (see 'START'). After this, the iterative learning process begins, which is the main central loop. This includes the experiment being controlled by M-LOOP, with the experimental output being sent back to M-LOOP. Then, to terminate, experimental and machine learner data is saved to a file (see 'END').

filesystem of a computer. A flow-diagram of M-LOOP's operation is shown in figure 8.

To run a machine learning experiment, first the runtime options must be initialised. This includes the choice of parameters such as how many experimental parameters are being optimised, the minimum and maximum bounds of these parameters, the number of experimental repeats which should be conducted, along with other optional parameters [8].

After this, when the program is initialised and begins running, it enters the main loop, shown as the loop in figure 2.4. Here, M-LOOP writes a set of parameters to a file. A computer reads this file and runs the experiment with these parameters. When complete, the experiment then writes an output file containing a single valued cost-function, as well as an optional uncertainty and a flag if the experiment failed to run properly. This cost function represents how well the experiment performed. Then, M-LOOP reads this file, and uses it as a training data point for whichever maching learning method is being used. In this case, this is either differential evolution or a Gaussian Process, as described above in sections 2.2 and 2.3.

Finally, after a specified number of runs, the machine learner terminates. At this point, M-LOOP saves the data collected and parameters associated with the machine learning algorithm, such as the differential evolution individuals or the final Gaussian process posterior distribution, before showing a series of visualisations.

An optimisation experiment can be represented by what happened during the experiments, such as with plots of the cost over time, parameter values over time, or the cost vs parameter. These are used to determine if the machine learning process fitted correctly. An example of these plot is shown in figure 25, taken from the experiment in section 3. Observable is the general minimisation trend over time, in figure 25a, showing the improvement of the minimisation method, and also seen is that the parameters tend towards their optimum values in figure 25c.

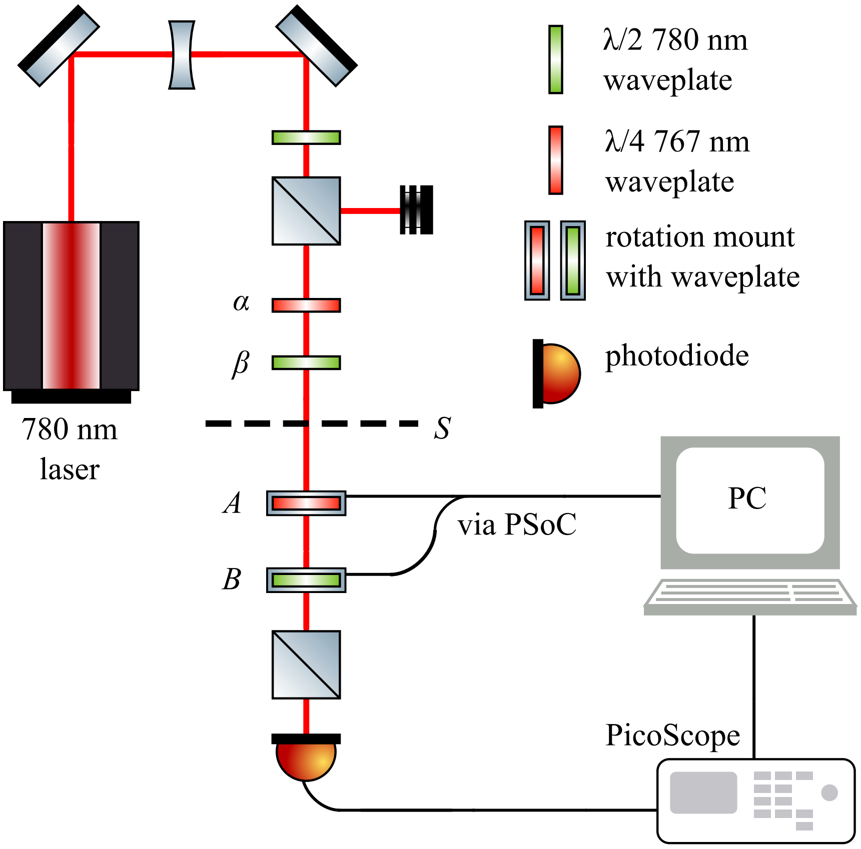

Figure 9: Setup for the simple experiment. A linearly polarised beam is subject to two randomly rotated waveplates α and β, which impose an arbitrary elliptical polarisation on the beam. The optimisation area of the experiment is after the plane S: the waveplates A and B are rotated to resintate a linear polarisation, and are controlled via PSoC microelectronics [13]. The degree of linear polarisaion is measured by proxy as the intensity of the photodiode, using a PicoScope.

One can also observe in figure 25b a near-sinusoidal-shape of the parameters, especially near the minimum, where most of the experiments took place. Finally, figure 25d provides a good showcase of the evolutionary aspect of differential evolution.

M-LOOP will be used as the primary tool in investigating the usefulness machine learning for online optimisation in the laboratory. We will do this using a simple polarisation experiment and a more complex cold atom experiment.

3 A simple experiment: polarisation correction

Building on the experiment in section 2.1, a more complex two-parameter experiment was devised. The experimental setup is shown in figure 9 and is designed to impose an arbitrary polarisation onto a linearly polarised beam, or by the principle of reversibility, to turn any arbitrarily-polarised beam into a linearly polarised one. This effect would be useful for cleaning the output of a laser, or for aligning a beam into a piece of optics which requires a precise polarisation, such as an optical fiber [12].

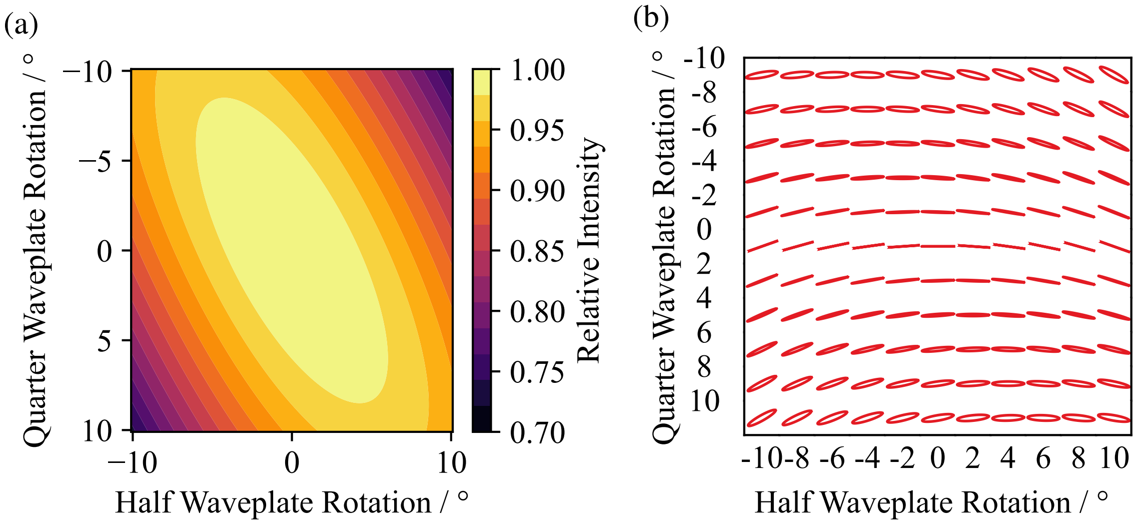

Figure 10: Simulated intensity distributions around the theoretical maximum. **(a)** Theoretical intensity 10° around the maximum for the setup in figure 9, with regard to the rotation of both half- and quarter-waveplates. **(b)** The polarisation of the beam for the angles in **a**, represented as polarisation ellipses [21].

For this simple experiment, we derive an analytical expectation of the cost-landscape, i.e., what the photodiode intensity ought to be based on the rotations of the waveplates. We do this in appendix A using Jones calculus [20]. The intensity I for the half- and quarter-waveplate rotations θ_h and θ_q, respectively, is

$$I \propto \cos^2(2\theta_h) + \cos^2(2[\theta_h - \theta_q]). \tag{7}$$

This relation is used as a model for our system in this section. A contour plot of this equation over the 10° range around $\theta_q = \theta_h = 0°$ is shown in figure 10, with a clear maximum at $\theta_q = \theta_h = 0°$, which will be the maximum sought out by the machine learner.

3.1 Experimental methods

The experimental setup is shown in figure 9. A rotation of 45° on both waveplates was repeated to find an experimental uncertainty of 1 mV (over a range of 3 V), which was passed into M-LOOP as the experimental uncertainty. The landscape was measured for 0°-90° rotation of each waveplate, resulting in a measurement of the total cost landscape. This is shown in figure 12a, and is used to compare a classical regression fit to the landscape to a Gaussian process fit.

3.2 Results of optimisation

This section is a comparison of classical optimisation techniques (differential evolution and function-fitting) with machine learning techniques (the Gaussian process).

Figure 11 shows two optimisation runs, one using differential evolution and one using Gaussian processes. As seen by the rolling mean, the Gaussian process (GP) optimisation reaches an optimum much faster than the differential evolution (DE). Numerically, GP reaches within 1%

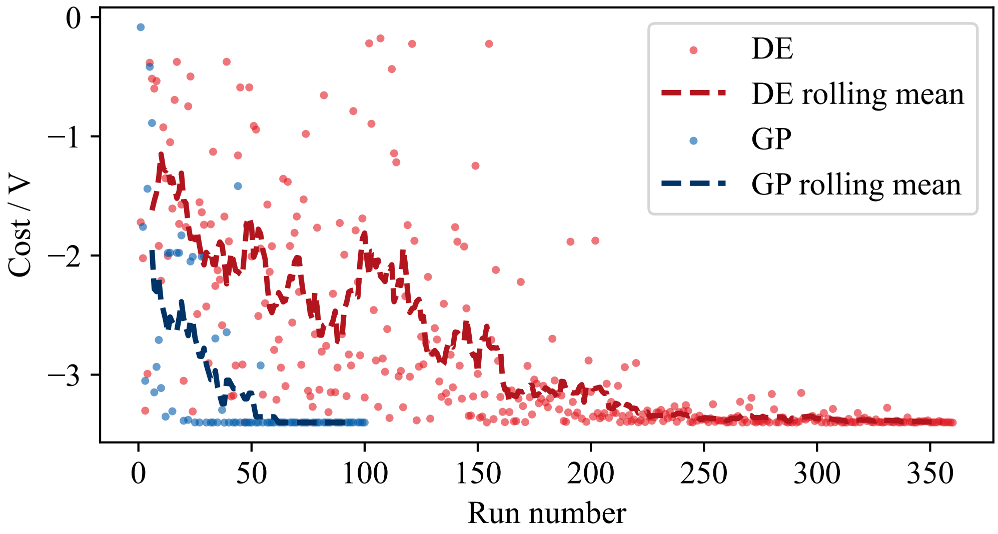

Figure 11: The cost over time for the simple experiment optimisation, comparing differential evolution (DE) to Gaussian Processes (GP). Each point plotted represents one experimental repetition, and the dashed lines are rolling means over 10 runs. GP reaches 5 (1)% of its final minimum in 30 (50) runs. DE reaches 5 (1)% of the same in 163 (247) runs.

of the final optimum after 50 iterations, and DE achieves the same goal within 247 iterations. As with the DE in figure 8, there are 30 individuals. In this case, that might be too large a number, since we know the landscape is only two-dimensional. However, we still need a reasonable number of individuals since the evolution algorithm relies on having a large population to avoid local minima. We know from figure 12a that the landscape has at least one local minimum as well as the global minimum. So, even reducing DE to only 10 individuals (a 3× decrease in population, so a 3× decrease in time), the GP would still have a faster optimisation.

Next, we compare the GP optimisation with the classical idea of fitting a function to the data with regression. This is only usable for this particular experiment since the functional form of the cost landscape should follow equation 7 from section 3. Thus, we fit the intensity as,

$$I = I_0 + I_A(\cos^2(2\theta_h) + \cos^2(2[\theta_h - \theta_q])), \tag{8}$$

where I_0 is an intensity offset, I_A is an intensity scaling factor, and θ_q and θ_h are the rotations of the quarter- and half-waveplates, respectively, in degrees. These rotations are given by,

$$\begin{aligned} \theta_h &= A_h\Theta_h - \phi_h, \\ \theta_q &= A_q\Theta_q - \phi_q, \end{aligned} \tag{9}$$

where A_h and A_q are scaling factors to convert rotation steps to angular degrees, ϕ_h and ϕ_q are angular offsets, and Θ_h and Θ_q are the rotations of the half- and quarter-waveplates in steps, respectively.

The sum of the square-residuals of the measured data (figure 12a) against the model function (equation 8) was minimised with respect to the fit parameters using the L-BFGS-B algorithm from scipy [19]. The optimum parameters from this optimisation are shown in table 2,

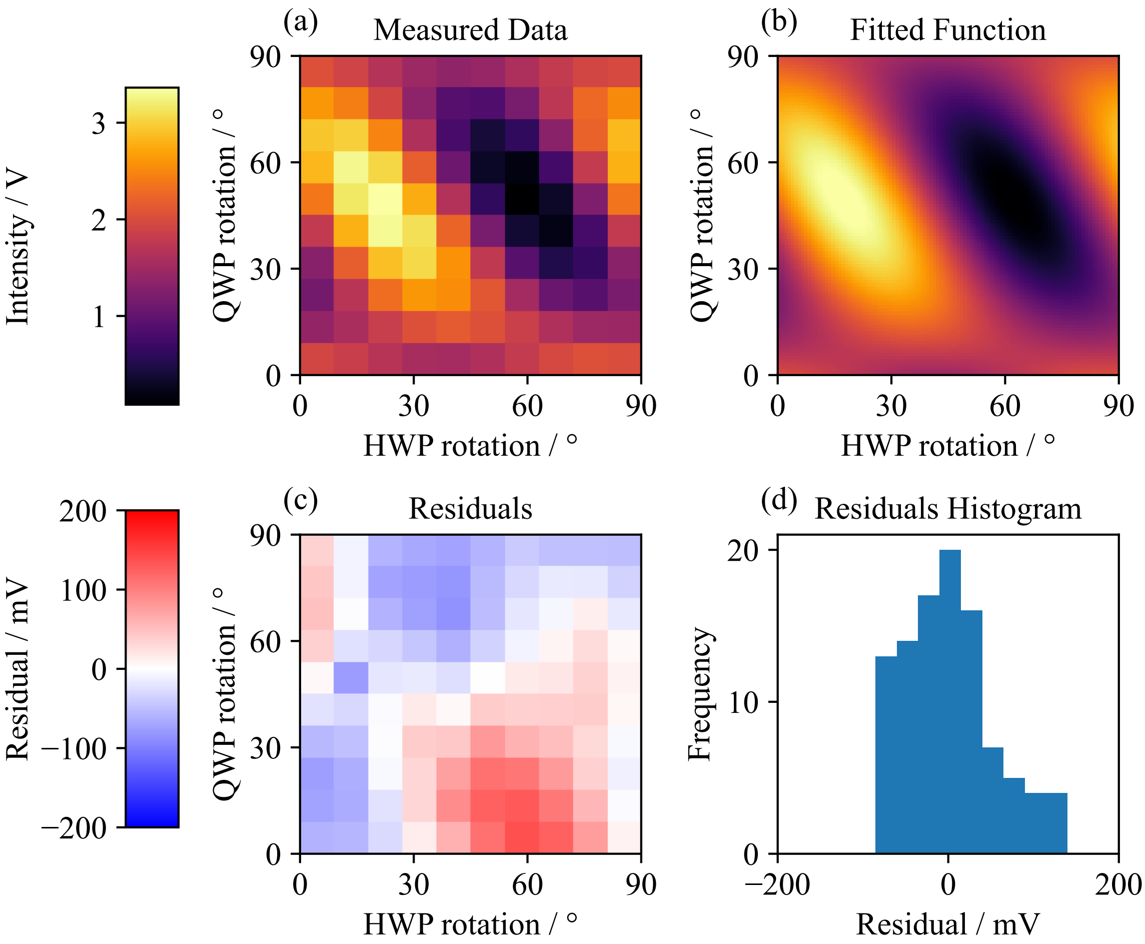

Figure 12: Comparison of measured data to a fitted function derived from residual minimisation. **(a)** 10×10 grid of measured intensities with respect to rotations of both waveplates. **(b)** Equation 8 over the same range with fitted parameters from table 2. **(c)** Residuals, calculated as **a** − **b**. **(d)** Histogram of residuals.

which shows that the two rotation mounts have a precision of 0.05 deg/step and 0.1 deg/step for the half- and quarter-waveplate rotation mount, respectively, as expected. This optimisation is shown in figure 12, showing the measured data, the fitted function with the optimal fit parameters, as well as the residuals to this fit. The residuals display a large-scale pattern, which suggests that the model function does not fully represent the physical experiment. This misrepresentation is a shortcoming of the body-function fitting minimisation technique, which is why we continue in this section to optimise the experiment with a Gaussian process. It is not known why this large-scale pattern arises, but one reason could be the small wavelength-mismatch of the waveplates with the laser light. As seen in figure 9, the quarter waveplate is rated for light 13 nm away from the laser light we are using, due to experimental constraints. This could introduce an error such as incomplete retardation of the light.

Now, we compare a Gaussian process (GP) fit to the body-function fitting above. Since the GP allows us to estimate the shape of a function, we can also calculate residuals from the measured data, which is shown in figure 13. Figures 13a and b look similar to a and b in figure

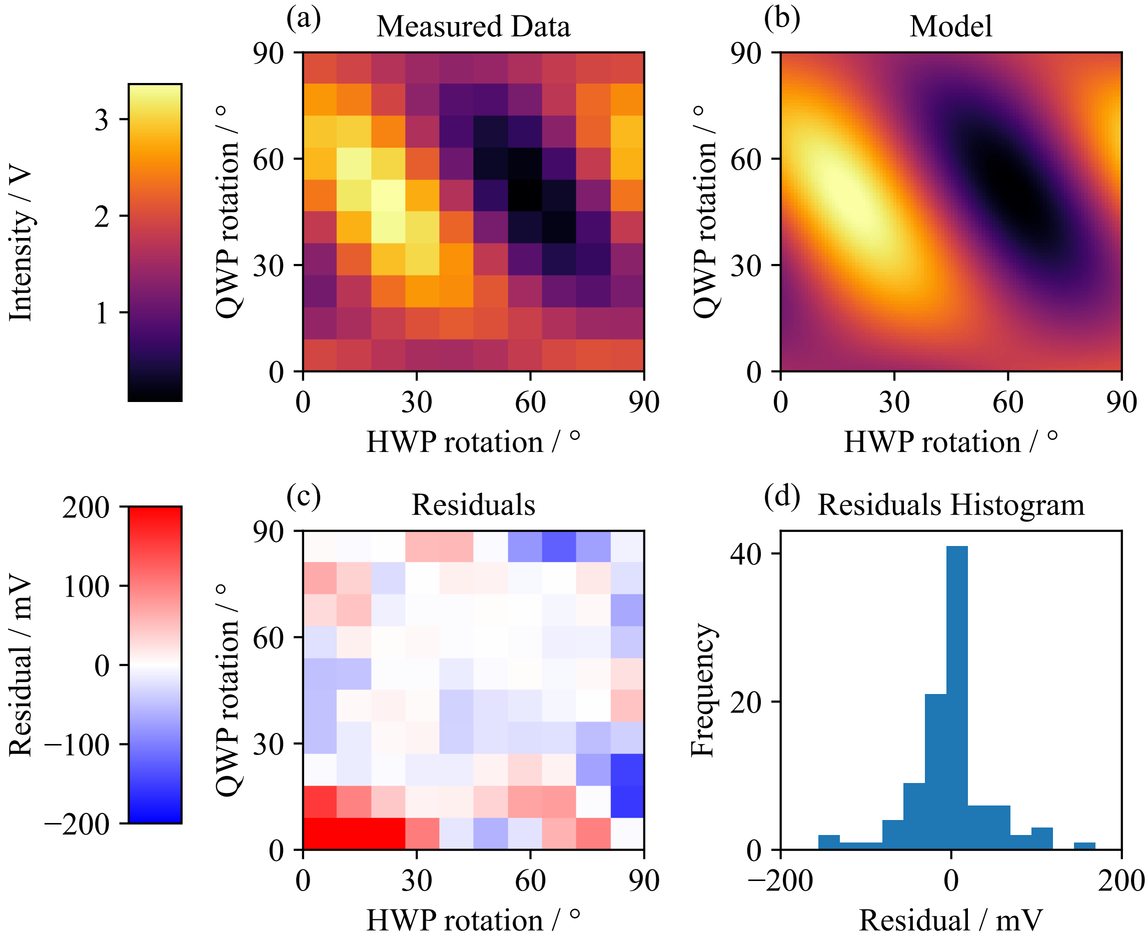

Figure 13: Comparison of measured data to the Gaussian Process fit. **(a)** 10×10 grid of measured intensities with respect to rotations of both waveplates. **(b)** The Gaussian Process estimation of the function. **(c)** Residuals, calculated as **a** − **b**. **(d)** Histogram of residuals.

12, since figures (a) are the same data and figures (b) are two different models attempting to recreate the shape of this data. The GP fit is that it had no knowledge of the shape of the function, and iteratively developed an idea of it. Looking at figure 13c, this meant that the GP did not fail in the same way as the functional fit, as it did not have a systematic error. However, near the edges (i.e., away from the optimum), the GP fit is poorer. This is likely because the algorithm used to explore the space was finding the maximum, so the precise shape of the function around the minimum areas was 'unimportant' for the GP to know, so long as it knew the approximate shape. Because of this, the residual histogram for the GP had a smaller variance, but a larger range than the one for body-function fitting.

Figure 14 shows a 3D comparison of the body-function and GP fitting. On this plot, it is more visually obvious that the GP does not the corner of the plot well.

The sum of the squared residuals for the body-function fitting is 300 V^2, compared to a larger value of 591 V^2 for the GP fit. This suggests the body-function fit is better, but the GP fit is better around the intensity maximum, while the bad fit around the corners inflates the residuals for the GP.

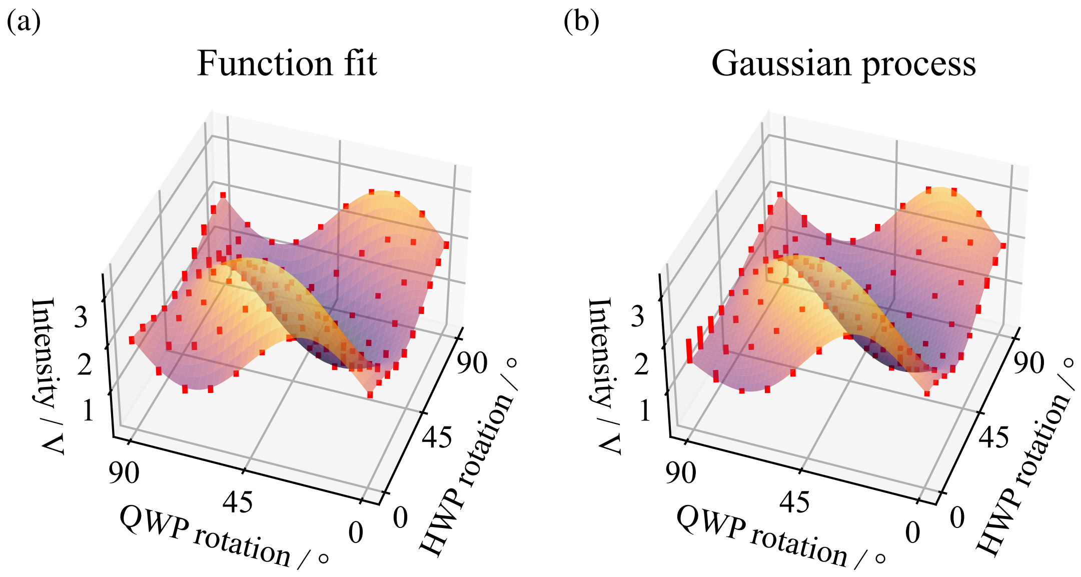

Figure 14: A 3D visualisation of the intensity distribution as the rotation angle is changed. Red lines represent measured points and their distance to the fitted function. **(a)** The body-function fit. **(b)** The Gaussian Process fit.

In conclusion, the GP fit allows for fitting of systematic errors, but body-function fitting allows parameter values to be determined with an error. So far we have only considered a simple experiment, but we will see that GPs become even more valuable when used with a more complex experiment.

4 A complex experiment: cold cesium

4.1 Cold atom experiments: a dive into evaporative cooling

A Bose-Einstein Condensate (BEC) describes a quantum-mechanical state of matter arising when many bosons accumulate in their lowest energy level [22]. In this work we are interested specifically in bosonic atoms creating a BEC [23]. The first BECs were observed in 1995, resulting in the award of the Nobel Prize in 2001 [24, 25], while Cesium BECs — the atom species used in this work — were developed later on [26, 27]. The experimental setup used in this work is shown in figure 15. The experiment includes components for processes such as of loading the MOT from the 2D MOT, optical molasses, and degenerate Raman sideband cooling [28–30], processes which cool the atoms by orders of magnitude. These processes are outside the scope of this work; instead we focus on the optimisation of a stage near the end of the experimental sequence: evaporative cooling [31].

Evaporative cooling is visually explained in figure 16. Evaporative cooling begins with the realisation of a dipole trap — two perpendicular laser beams impose a trapping potential on a group of atoms, confining atoms with a low enough temperature in the trap. This is discussed more in [32]. A 1D cut of the potential shown in figure 16a. The second and third panels show the process of evaporative cooling — the trap is made shallower, allowing the hotter atoms to

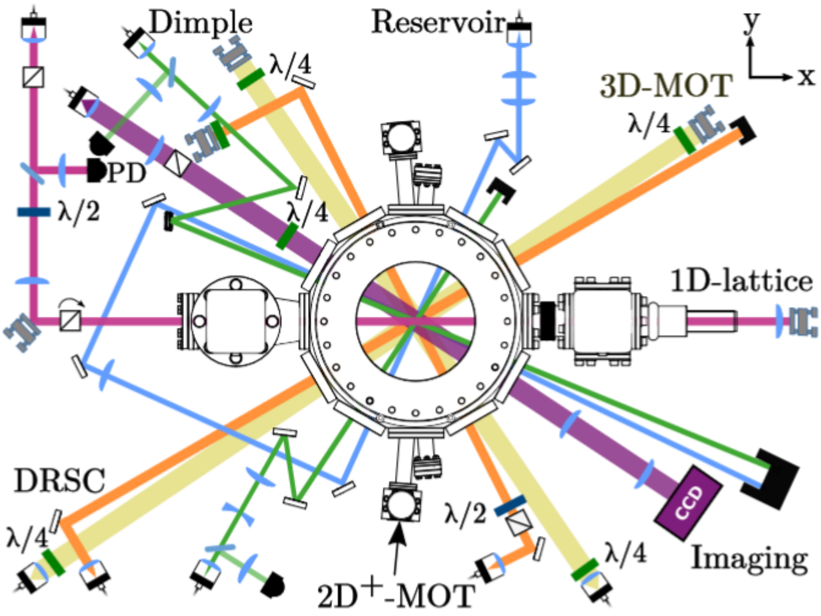

Figure 15: Experimental setup for the cold atom experiment used in section 4. Figure originally from [28]. We change only the Dimple lasers, which are two lasers which create a dipole trap for the atoms [32]. There is also a bias coil around the central vacuum chamber, providing a tunable magnetic field, which changes the scattering length of the atoms.

evaporate, and the remaining atoms are allowed to rethermalise to a colder temperature. Overall, this results in a reduction in the number of trapped atoms, but the atoms remaining in the trap are at a much higher density. As well as how quickly to perform this process (in order to allow time to rethermalise but not too much time that it becomes inefficient), other problems which affect the cooling rate must be considered. The first is that in the case of Cesium, the scattering length can be changed via the application of a magnetic field [33], which influences how often scattering events occur during the evaporation. Figure 17 shows this tunable scattering length as well as indicating the magnetic field for which the scattering length is zero. With this control, three-body loss must be considered. This process describes when three atoms interact in a scattering event which creates a diatomic molecule and an energetic third atom, meaning that all three atoms are lost from the trap. The probability of this process is controlled via the scattering length, which is difficult to quantify [34]. Thus, the magnetic field is a novel parameter that we are able to control, which has an optimal value which is difficult to work out analytically. This difficulty in manually finding an optimal value provides a strong motivation for using M-LOOP and machine learning to optimise the evaporative cooling process.

4.2 Previous work with other species of atoms

There is a wide range of work using machine learning to improve atomic physics experiments, but it has primarily used Rubidium [8, 10, 35, 36], or more rarely, Thulium [11]. These works have provided different insights and novel results, as discussed below.

The work listed uses different approaches when it comes to parametrisation of the experi-

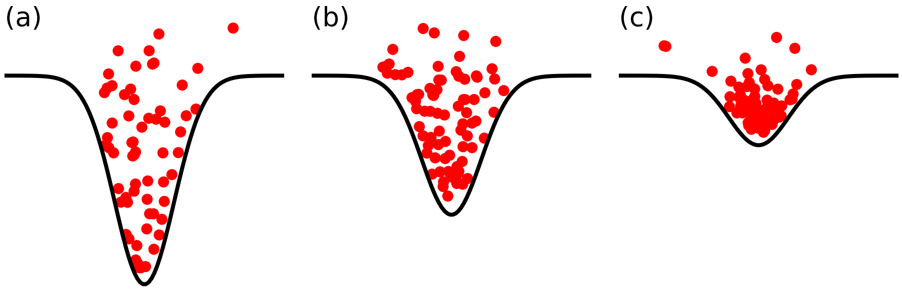

Figure 16: An illustration of evaporative cooling. As the depth of a trap is lowered, hotter atoms escape, and the remaining atoms rethermalise, resulting in the loss of some atoms, but more dense, colder final atomic cloud.

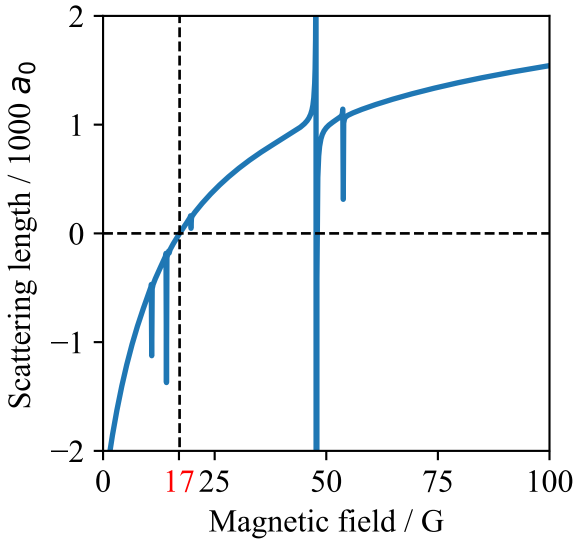

Figure 17: The scattering length of Cesium in terms of applied magnetic field, in terms of the Bohr radius a_0. The scattering length crosses zero at 17.1 G, allowing novel investigation into phenomena at positive, negative, or zero scattering length [27].

ment (i.e., turning the evaporation sequence into a series of 1-dimensional variables to be optimised), and contrasting ideas for describing the quality of the experiment with a cost function. The parametrisations used are linear ramps between points [8, 10, 11, 36], flat fragments [35], as well as polynomial approximations [8]. In this work we will use linear ramps as these remain computationally simple while allowing a large amount of freedom in the shape of the overall ramp.

In terms of cost functions, other work varies a lot more. Some work uses single-value cost functions like the peak optical depth (density) [35], or the average height of the atomic cloud edges [8], whereas other work sums the total number of atoms in a small region of the final cloud [10, 36]. Overall, the cost function is a difficult decision as it should be a metric which has high sensitivity — to allow the experiment to be well optimised — but also should have a large range of measurement. This is because a machine learner will attempt many experiments with parameters far from their optimums, resulting in poor-quality experiments. Thus, a cost-function should be able to measure these bad experiments.

The concluding opinion of other work is that a machine learner optimises a sequence much faster than the manual or classical optimiser case, but useful predictions could rarely be made

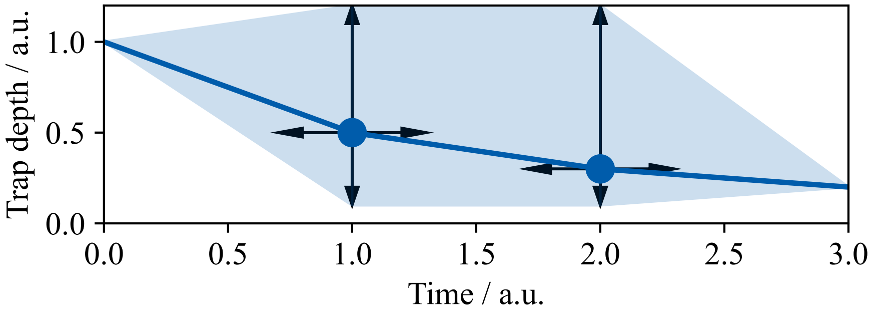

Figure 18: An example experimental evaporation sequence, where the trap depth is reduced over time from a larger to a smaller value. Blue dots represent adjustable points, with a finite vertical range of trap depths. The blue shaded area represents the region containing every possible shape of evaporation ramp creatable by adjusting these points.

from the final optimisation results [8, 10, 11, 35, 36]. The faster optimisation speed is evidenced in [8], as the number of runs taken to optimise the experiment was 20 for the machine learner (Gaussian process) and 145 for the classical optimiser (Nelder-Mead). They also claim that for some of their parameters, the machine learner fit them well and predictions could be made about the cost, but for other parameters it was impossible to tell. Often in this way, a machine learner makes it difficult to draw conclusions about the optimisation.

Because of these difficulties, a machine learner provides the best use case in a goal-oriented setting, by providing a repeatable, high-quality experiment setup which can be used for another scientific goal, in contrast to performing science and drawing conclusions from the machine learning process itself. In [10], it is argued that their investigation proves that they can perform short optimisations to account for experimental drift. For example, by running a one-hour optimisation at the start of each day, or performing longer runs every few weeks to account for long-term drift. This describes a goal-oriented approach and examples a great use case for machine learners in atomic physics.

4.3 Experimental methods

4.3.1 Experimental sequence control

In contrast to figure 16 which gives an overview of the evaporative cooling sequence, figure 18 shows a more continuous version. The trap depth is reduced via a few linear ramps, providing the evaporation required. In figure 18 this is shown in terms of trap depth, whereas in the experiment we will control the power of two independent, cross-aligned trapping lasers.

With our experiment, there is voltage-control of intermediate laser powers and magnetic fields (e.g., the blue points in figure 18) and the time duration of experimental stages. This means that the duration of each linear ramp can be changed, as well as the gradient of the ramp. This is parametrised by having three time parameters, one per stage of the sequence (i.e., between each set of two vertices in figure 18), and two voltage parameters per trapping laser controlling the intermediate voltages (i.e., the height of the central points in figure 18).

The result is a total of 3 time + 4 voltage parameters (2 for each trapping laser), totalling 7 parameters. Also controllable is the constant magnetic field for each timestep, which adds another 3 parameters for a total of 10. This parametrisation is the same which is used by manual optimisation, and so our main assessment for the machine learner is whether — with the same freedom as the manual case — it can do better than manual optimisation.

We can also limit the freedom of the experiment by having fewer ramps. The case described above is labelled as the '3-ramp' case, as it has three linear fragments. We can remove one or two points from the middle of the sequence to turn it into the 2- or 1-ramp case, respectively. These will be used to determine the freedom that the optimiser needs to achieve a comparable result to the manually optimised case.

4.3.2 Absorption imaging

To provide a quality measure for the experiment, the atom cloud resulting from a particular evaporation sequence is imaged using absorption imaging [37]. This is performed by capturing three images in the vacuum chamber: one with the atoms illumated by resonant laser light (a); one of just the laser light (b); and a background image (c).

Image c is a background image and is subtracted from the other two images to remove a common offset, resulting in the corrected images $a - c$ and $b - c$.

The intensity $I(x, y)$ of image $a - c$ (of the atom cloud) will follow [30],

$$I(x, y) = I_0(x, y)e^{-n(x,y)\sigma_{\text{tot}}}, \tag{10}$$

where $I_0(x, y)$ is the initial intensity, $n(x, y)$ is the density of the atomic cloud at position x, y, and σ_{tot} is the total absorption cross-section of the cloud. This is the Beer-Lambert law which is accurate provided that the probe does not saturate the atoms [37].

By definition, image $b - c$ is $I_0(x, y)$. By comparing it with image $a - c$, the atomic density n can be computed as [37],

$$n(x, y) = \frac{1}{\sigma_{\text{tot}}} \ln\left(\frac{a-c}{b-c}\right) = \frac{1}{\sigma_{\text{tot}}}\text{OD}(x, y), \tag{11}$$

where OD is the optical depth of the image.

To model the thermal clouds evident in the experiment, the density distribution is fitted to a model 2D Gaussian function [37],

$$n(x, y) = n_0 \exp\left(-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2}\right), \tag{12}$$

where the fitting parameters used are: n_0, the peak atom density; x_0 and y_0, the central position of the cloud; and σ_x and σ_y, the Gaussian widths of the cloud in the x- and y- directions.

The peak optical depth OD can be found as the peak of the fit from equation 12. The total atom number N can be obtained by summing the pixels in the image as [37],

$$N = \int\int n(x, y)dxdy = \frac{1}{\sigma_{\text{tot}}}\int\int \text{OD}(x, y)dxdy. \tag{13}$$

4.3.3 Cost function as a quality descriptor

The cost function used for the experiment is an efficiency factor γ, where,

$$\gamma = \frac{\log(\frac{\mathrm{OD}}{\mathrm{OD}_0})}{\log(\frac{N}{N_0})}, \tag{14}$$

where OD and N are as described above, and OD_0 and N_0 are the initial values of optical depth and atom number prior to the evaporation sequence. This efficiency factor γ is a measure of the relative increase in optical depth (i.e., density) versus the relative decrease in atom number. By minimising this factor, the experiment is improved by increasing the gain in optical depth per atom lost.

An error in this efficiency factor was determined by running the experiment evaporation ramp 30 times and computing the standard error. This standard error was 0.07, with typical values of 0.7 for the efficiency factor, for a relative error of around 10%.

4.4 Results of optimisation

The experiment was optimised with several degrees of freedom, finally being given the same freedom as the manual optimiser. Firstly, the impact of the magnetic field on the result of the experiment was verified, followed up by optimising the experiment with 1, 2, and 3 ramps, to investigate the effect this freedom had on its optimisation. Finally, the 3-ramp optimisation was compared with the results from the manual optimisation. These experiments will be further discussed in this section.

4.4.1 Experimental novelty: tunable scattering length via the magnetic bias field

Since Cesium has not before been used in machine learning optimisation experiments, we had the novel ability to change the magnetic bias field, and therefore the scattering length. A simple experiment involving optimising only the magnetic field was performed, the results of which are shown in figure 19. There is a visible minimum in the negative of the final optical depth, i.e., a maximum in the density of the atom cloud. The optimised minimum is at a magnetic field of 22.9 ± 1.8 G, where the uncertainty is the distance between the mean line minimum and the magnetic field of the lower standard deviation bounding dashed lines for the minimum cost. This is within error of the manually optimised value. To decrease this uncertainty, the optimisation would have to be run for more time. It is uncertain the magnitude of effect this would have on the uncertainty since the points in the centre of figure 19 have a spread comparable to the fitted standard deviation. Thus, if more measurements were made, they may also have the same deviation since the experiment was not identical every time it was run.

However, this optimisation was enough evidence to support that the magnetic field had a measurable effect on the results of the experiment, so in further optimisations the magnetic field value was changed.

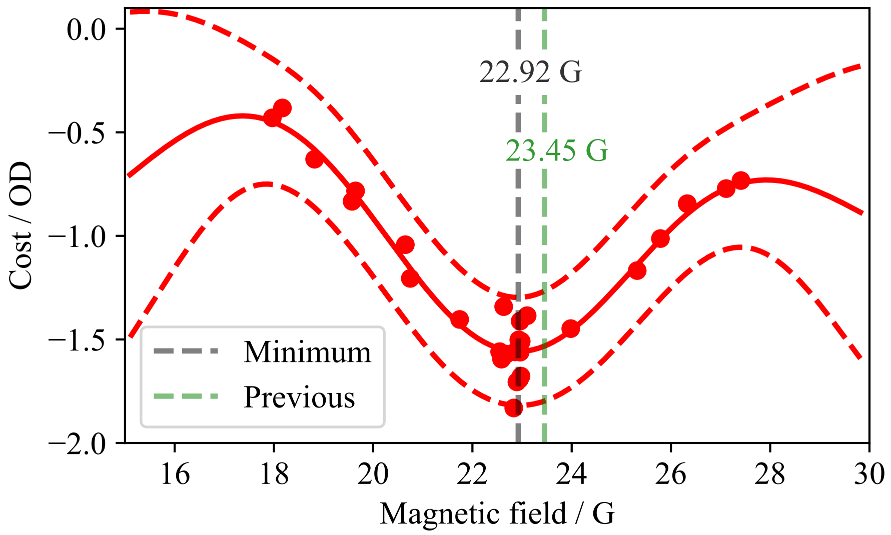

Figure 19: A fitted Gaussian process of the final OD (density) of the atomic cloud versus the magnetic field value after 30 iterations. The solid red line is the estimated cost versus magnetic field value, and the shaded red lines are one standard deviation from this cost. The minimum of the solid red line is shown by the dashed grey line, and the manually-optimised magnetic field value is shown by the dashed green line.

4.4.2 'Ramping' up the complexity

The experiment was optimised using M-LOOP with 1, 2, and 3 linear ramps during the evaporation stage, with each increase in the number of ramps increasing the freedom of ramp-shape while also increasing the number of parameters required to describe this shape. The different ramp experiments are described as thus:

- In the 1 ramp case, there are 2 parameters: the time of the ramp and the magnetic field.
- In the 2 ramp case, there are 6 parameters: 2 times of the ramps, 2 magnetic fields (1 per ramp) and 2 laser voltages (1 per trapping laser, labelled A and B).
- In the 3 ramp case, there are again an additional 4 parameters: 1 more time for the third ramp, 1 more magnetic field for the third ramp, and 2 more laser intensities for the time between the second and third ramp. This makes a total of 10 parameters.

The shapes of each of these ramps can be seen in figure 20, which shows the final optimised ramps.

Because of this increase in parameters with an increase in ramps, each successive optimisation routine required more repeats and thus more time to optimise the experiment. For each of 1, 2, and 3 ramps the optimisations were run for 40, 100, and 400 iterations, respectively. This took a time of 15 minutes, 40 minutes, and 160 minutes, respectively. The latter end of this timescale reaches an impractical limit for a daily optimisation. If a shorter optimisation time was required, one would have to reduce the number of parameters, for example by omitting the

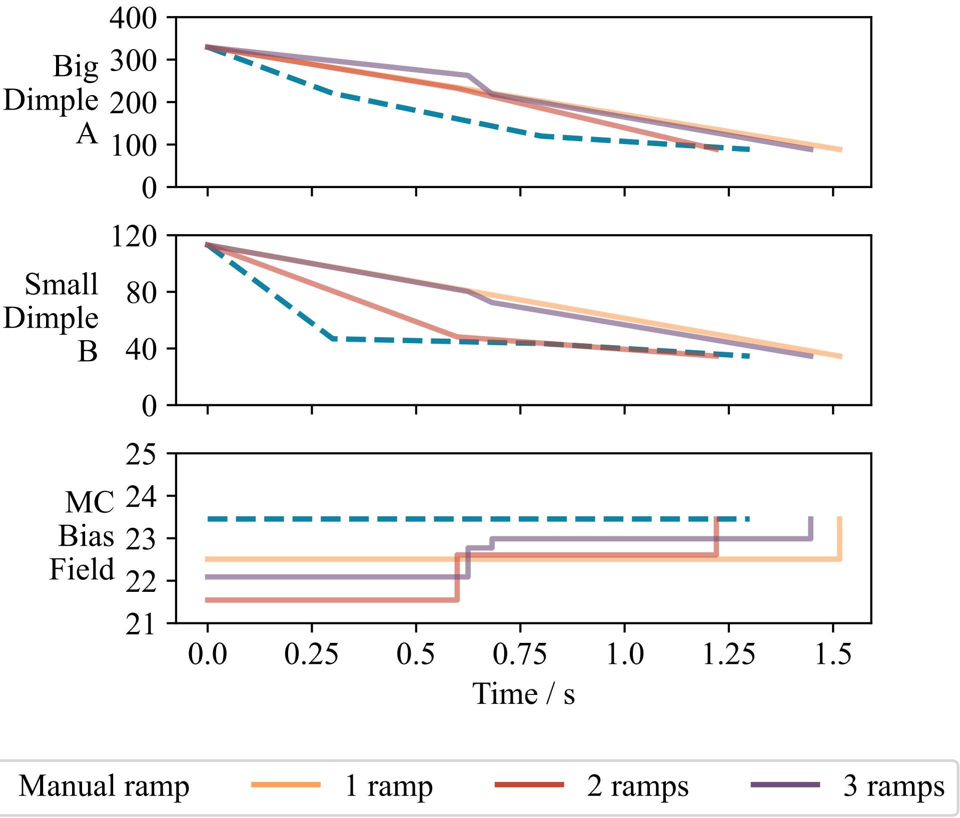

Figure 20: A comparison of the optimised sequences for 1 (orange), 2 (red), and 3 (purple) linear ramps. The dashed blue line shows the default, manually optimised sequence. The ramps are shown for the two trapping lasers A and B (top, middle), and the magnetic bias field (bottom).

optimisation of some parameters such as the timesteps. Alternatively, as we will now discuss, the number of ramps could be reduced while still achieving a desirable optimum solution.

The completion of the optimisation resulted in a set of predicted optimal parameters for each number of ramps. These are shown in figure 20. The sequences from this optimisation were repeated 30 times each, in order to compare their consistency and final cost. This includes a repeat of the manually optimised sequence, and also includes 10 repeats of the initial parameters. These repeats are shown in figure 21. The experiment begins in the bottom right of the plot (grey), and after the evaporation ramp moves to the top left. The goal of optimisation is therefore to increase the gradient of this movement, i.e., to increase the gain in optical depth (i.e., density) while decreasing the number of atoms lost. As seen by comparison with the manual case, the 2-, and 3-ramp case achieved this. They had very similar optical depths to the manual case, but the 3-ramp case achieved a higher atom number for the same optical depth. The manual case ended up with an atom number of $(1.46 \pm 0.10) \times 10^5$ and an optical depth of 1.60 ± 0.16, compared to the 3-ramp atom number of $(1.70 \pm 0.11) \times 10^5$ with an optical depth of 1.61 ± 0.17, where the uncertainty is the standard deviation of the repeated experiments. This data is also tabulated in table 1, for all the ramps. This increase in atom number is a definite increase for the three-ramp case, which proves that it was able to optimise the experiment to a

Run	Atom Number N / $\times 10^5$	Optical Depth OD
Manual	1.46 ± 0.10	1.60 ± 0.16
1 Ramp	1.54 ± 0.12	1.36 ± 0.13
2 Ramp	1.65 ± 0.16	1.58 ± 0.17
3 Ramp	1.70 ± 0.11	1.61 ± 0.17

Table 1: A comparison of statistics of the atom number and optical depths of the experiments resulting from the experimental sequences in figure 20. Values tabulated are the mean of the experimental runs, and the uncertainties are the standard deviations of the runs, which are also shown on figure 21.

greater quality than manual optimisation. However, it is not clear *why* the optimised solution performs better, which is a common problem in optimisation experiments [10, 36].

Another question is if there is a trend in the magnetic field. For the 2- and 3-ramp cases, the magnetic field begins low, and then increases, respecting an increase in the scattering length (see figure 17). The optimum magnetic field in the 1-ramp case appears to be the average magnetic field between these other cases. A conclusion cannot be made without more data, but it is possible that the magnetic field has a large enough impact on the evaporation that changing it throughout can provide more novel and efficient evaporation, in contrast to it being constant throughout, as is the case for the manually optimised experiment.

A final question is how close each optimisation routine got to the theoretical optimum solution, which they each would have reached given infinite optimisation time. The time to give an optimiser is a difficult question to answer before it has been tested — in this work the best attempts were made based on previous optimisation runs. Due to time constraints, it was not possible to repeat an optimisation routine, so there is a possibility that some routines did not reach their true optimum. Because of this, the values from table 1 are lower bounds, particularly for the 1- and 2-ramp cases. Regardless, machine learning saves time when compared to manual or classical optimisation, which we will now evidence.

4.4.3 Comparison of machine learner and classical routine

Figure 22 shows the cost-vs-iteration for the 3 ramp optimisation experiment. Since the nature of Gaussian processes (GP) allows us to estimate the landscape, the data from this optimisation was used to simulate a differential evolution (DE) optimisation of the same experiment. Due to the DE simulation relying on the Gaussian process as a landscape-estimator, we cannot compare the resulting optimum parameters found. However, we can use it to compare the optimisation time of a classical and a machine learning algorithm. This is done graphically in figure 11. The GP takes 255 runs to reach within 5% of its optimum found rolling mean-cost, or 258 runs to get within 1%. This is contrasted with DE which takes 640 runs to achieve a 5% proximity to its final optimum, and 913 runs to achieve a 1% proximity. There is a clear distinction between the methods, showing that the optimisation gradient of the GP is much steeper than that of the DE. Thus, machine learning opens up far more efficient avenues for optimisation of this experiment.

The GP run was ended after 400 iterations due to experimental time constraints. It is uncertain what the result of further GP optimisation would be. It is hoped that if allowed more time,

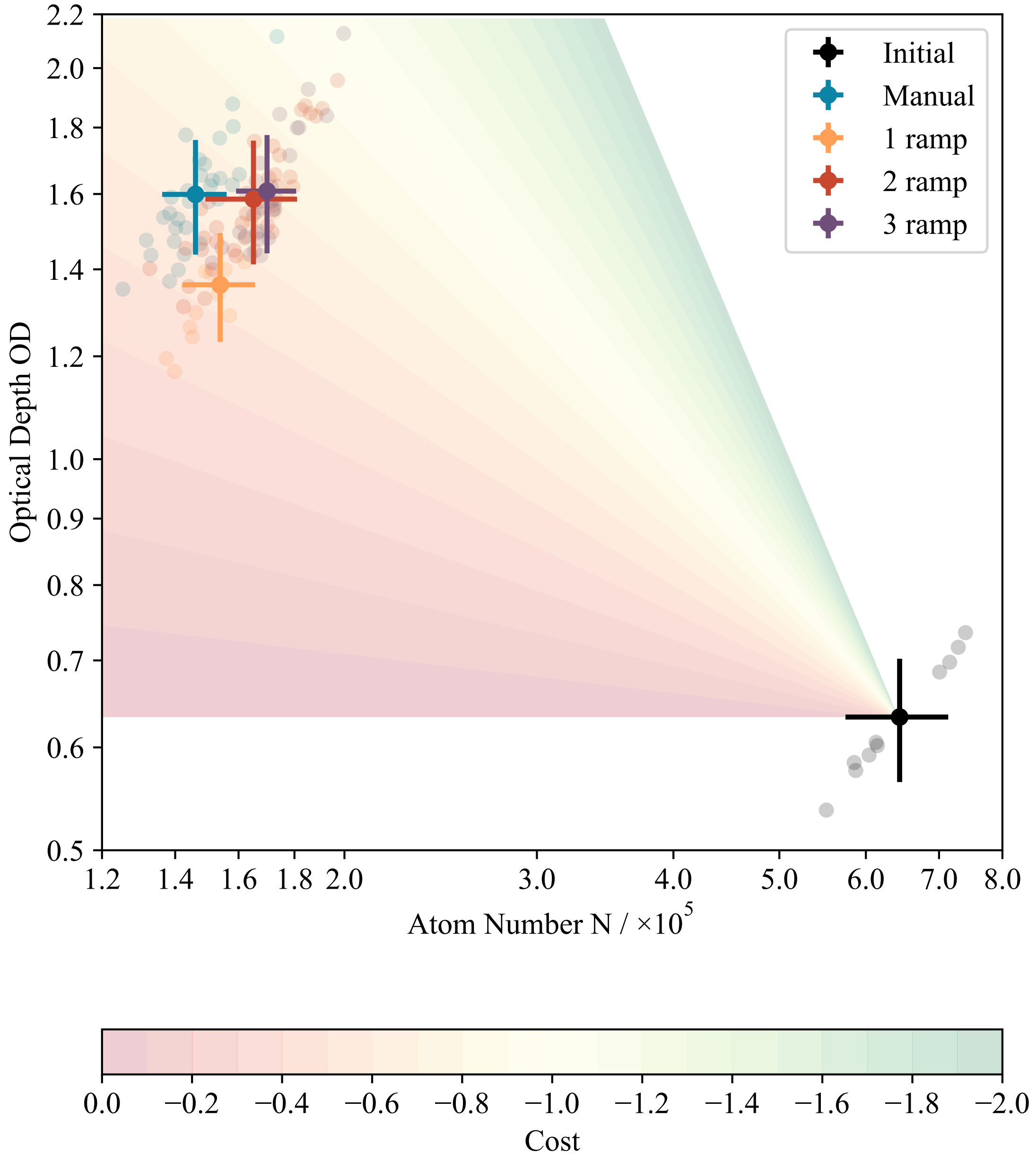

Figure 21: A comparison of the results of the evaporations using the different optimised sequences from figure 20. The start of the evaporation is shown in the bottom right of the figure. More efficient evaporation begets a larger increase in optical depth and a smaller decrease in atom number, i.e., moving towards the upper right. This is represented by the cost contours, which show the value of the cost function 14 at each point on the figure.

the GP would become a better estimator at the landscape of the experiment, and create a more robust solution. This is a topic of further study.

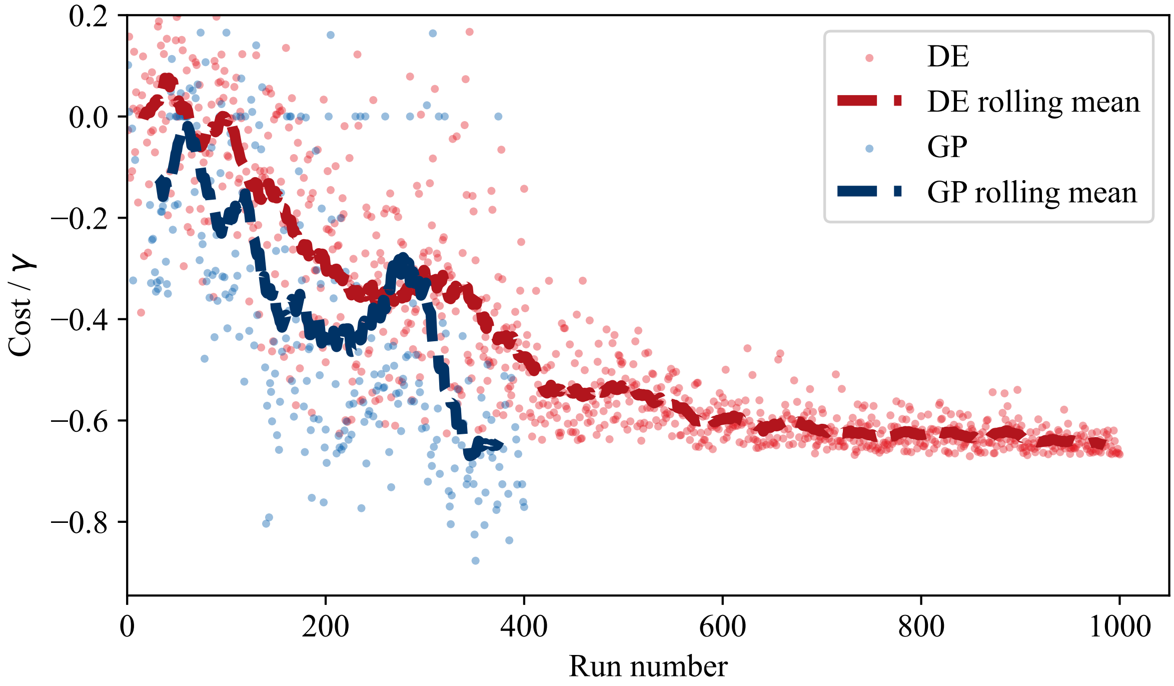

Figure 22: A cost over time comparison for differential evolution (DE) versus Gaussian process (GP) minimisation runs. The GP run is the 3-ramp run, as it was the most complex experiment performed. The DE run is simulated using the predicted landscape from the GP run, as an actual DE run would take an impractical time to complete. Each GP point represents one experimental run, and a rolling mean over 30 points is shown (dashed lines) for GP and DE. The GP optimisation reaches 5 (1)% of optimum in 255 (258) runs, while the DE optimisation reaches 5 (1)% of optimum in 640 (913) runs.

5 Conclusion

In this report, machine learning is found to be a very useful tool for goal-oriented optimisation of automatic experiments, and it is found to be simple to implement in atomic physics laboratories. It is useful for complex setups such as evaporative cooling, while also being beneficial for simpler optical setups. Machine learning is able to improve the output of an experiment to a better level than that of a human optimiser, and the type of experiment which is optimised is often automated in the laboratory, making integration easy.

It was found that Gaussian processes perform decidedly faster than the classical algorithm differential evolution, and that they provide informative estimations of how particular parameters may affect the quality of an experiment. However, the optimisation results can be counterintuitive and hard to explain, especially in the case of a complex experiment. It is not clear when planning an optimisation which parameters or cost functions will result in the best solution — the optimisation simply must be performed to test these parameters. Moreover, the available parameter options to choose from are not obvious. For example, it is unclear what will make a particular experimental cost function easier to optimise than another, or how much freedom to give a machine learner, as more freedom often results in more variables, and so a longer optimisation time. Furthering the latter problem, the experiment must also be describable by a single number, which can often result in losing information on other aspects of the success of an experiment. Finally, it is difficult to infer physical meaning from optimisation results, such as why one particular parameter value is better than another, and thus it is difficult to decide how the optimisation can be improved in future experiments.

It is for these reasons that Gaussian-process online optimisation should be a goal-oriented approach, used to increase the quality of an experiment for further research, rather than conclusions being drawn from the optimisation alone. Online optimisation could be used, for example, to account for short- or long-term experimental drift in atomic physics experiments, or to maximise the density of an atom cloud before a more complex stage after evaporation requiring a very dense collection of atoms.

5.1 Future work

The novelty of a tunable scattering length brought by the use of Cesium in this experiment is a topic requiring further study, including investigation into how changing the magnetic bias field during the evaporation can effect the evaporation efficiency, and how strong this effect is.

For further investigation into evaporation experiments like this, more investigation should be performed into the cost function used. For example, it would be convenient to use a cost function which is proportional to phase-space density, a concept used widely in atomic physics experiments, which is difficult to measure directly but can be approximated with the results of absorption imaging. Also, repeat optimisations should be performed to verify that the optimum evaporation sequences found by machine learning in one optimisation run are the same as found in repeat runs.

Finally, these optimisation runs take a long time on such a complex experiment, and thus investigation into other optimisation routines should be considered. For example, Nelder-Mead is a classical minimisation routine which converges very quickly on local minima. Such an algorithm could be used if the cost-landscape of the experiment has a large global maxima and few to zero local minima. This idea of landscapes being ‘trap-free’ is discussed more in [8]. If the cost landscapes are shaped like this, optimisation routines specialising in finding local minima could be a route into even faster optimisation, whereas machine learning routines provide techniques more robust against local minima and are able to model the cost landscape of an experiment.

Bibliography

[1] D. Kermany, M. Goldbaum, W. Cai, *et al.* *Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning.* Cell, 172(5):1122–1131.e9, February 2018. DOI: 10.1016/j.cell.2018.02.010

[2] J. Patterson and A. Gibson. *Deep Learning: A Practitioner's Approach.* O'Reilly, 2017. ISBN: 978-1-4919-1425-0.

[3] R. Horisaki, R. Takagi, and J. Tanida. *Learning-based imaging through scattering media.* Optics Express, 24:13738, June 2016. DOI: 10.1364/OE.24.013738

[4] D. Nguyen-Tuong and J. Peters. *Local Gaussian process regression for real-time model-based robot control.* In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 380–385, September 2008. DOI: 10.1109/IROS.2008.4650850

[5] D. Nguyen-tuong, J.R. Peters, and M. Seeger. *Local Gaussian Process Regression for Real Time Online Model Learning.* Advances in Neural Information Processing Systems, 21:1193–1200, 2009.

[6] J. Snoek, H. Larochelle, and R.P. Adams. *Practical Bayesian Optimization of Machine Learning Algorithms.* In Advances in Neural Information Processing Systems. Curran Associates, Inc., August 2012. DOI: 10.48550/arXiv.1206.2944

[7] C.E. Rasmussen. *Gaussian Processes in Machine Learning.* In Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch, editors, Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures, Lecture Notes in Computer Science, pages 63–71. Springer, Berlin, Heidelberg, 2004. DOI: 10.1007/978-3-540-28650-9_4

[8] P.B. Wigley, P.J. Everitt, A. van den Hengel, *et al.* *Fast machine-learning online optimization of ultra-cold-atom experiments.* Scientific Reports, 6(1):25890, May 2016. DOI: 10.1038/srep25890

[9] G. Ness, A. Vainbaum, C. Shkedrov, *et al.* *Single-Exposure Absorption Imaging of Ultracold Atoms Using Deep Learning.* Physical Review Applied, 14(1):014011, July 2020. DOI: 10.1103/PhysRevApplied.14.014011

[10] A.J. Barker, H. Style, K. Luksch, *et al.* *Applying machine learning optimization methods to the production of a quantum gas.* Machine Learning: Science and Technology, 1(1):015007, February 2020. DOI: 10.1088/2632-2153/ab6432

[11] E.T. Davletov, V.V. Tsyganok, V.A. Khlebnikov, *et al.* *Machine learning for achieving Bose-Einstein condensation of thulium atoms.* Physical Review A, 102(1):011302, July 2020. DOI: 10.1103/PhysRevA.102.011302

[12] R.S. Mathew, R. O'Donnell, D. Pizzey, *et al.* *The Raspberry Pi auto-aligner: Machine learning for automated alignment of laser beams.* Review of Scientific Instruments, 92(1):015117, January 2021. DOI: 10.1063/5.0032588

[13] R. Ashby. *Designer's Guide to the Cypress PSoC.* Elsevier Science & Technology, Oxford, UNITED STATES, 2005. ISBN: 978-0-08-047714-5

[14] L.A. McArd. *Rotation mount user's guide.* October 2021. Unpublished laboratory resource, available upon request.

[15] R. Storn. *Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces.* page 16, January 1995. DOI: 10.1023/A:1008202821328

[16] A.J. Ogden. *Applying machine learning to find galaxy redshifts*, Master's thesis. April 2022. Unreleased.

[17] A. Mokhtari and A. Ribeiro. *Regularized stochastic BFGS algorithm.* In 2013 IEEE Global Conference on Signal and Information Processing, pages 1109–1112, December 2013. DOI: 10.1109/GlobalSIP.2013.6737088

[18] E. Giné and R. Nickl. *Mathematical Foundations of Infinite-Dimensional Statistical Models.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 2015. DOI: 10.1017/CBO9781107337862

[19] P. Virtanen, R. Gommers, T.E. Oliphant, *et al.* *SciPy 1.0: fundamental algorithms for scientific computing in Python.* Nature Methods, 17(3):261–272, March 2020. DOI: 10.1038/s41592-019-0686-2

[20] R.C. Jones. *A New Calculus for the Treatment of Optical Systems I. Description and Discussion of the Calculus.* JOSA, 31(7):488–493, July 1941. DOI: 10.1364/JOSA.31.000488

[21] E. Collett and B. Schaefer. *Visualization and calculation of polarized light. I. The polarization ellipse, the Poincaré sphere and the hybrid polarization sphere.* Applied Optics, 47(22):4009–4016, August 2008. DOI: 10.1364/AO.47.004009

[22] S.N. Bose. *Plancks Gesetz und Lichtquantenhypothese.* Zeitschrift für Physik, 26(1):178–181, December 1924. DOI: 10.1007/BF01327326

[23] A. Einstein. *Quantentheorie des einatomigen idealen Gases.* In Albert Einstein: Akademie-Vorträge, pages 237–244. John Wiley & Sons, Ltd, 1925. DOI: 10.1002/3527608958.ch27

[24] K.B. Davis, M.O. Mewes, M.R. Andrews, *et al. Bose-Einstein Condensation in a Gas of Sodium Atoms.* Physical Review Letters, 75(22):3969–3973, November 1995. DOI: 10.1103/PhysRevLett.75.3969

[25] M.H. Anderson, J.R. Ensher, M.R. Matthews, *et al. Observation of Bose-Einstein Condensation in a Dilute Atomic Vapor.* Science, July 1995. DOI: 10.1126/science.269.5221.198

[26] T. Weber, J. Herbig, M. Mark, *et al. Bose-Einstein Condensation of Cesium.* Science, 299(5604):232–235, January 2003. DOI: 10.1126/science.1079699

[27] T. Kraemer, J. Herbig, M. Mark, *et al. Optimized production of a cesium Bose–Einstein condensate.* Applied Physics B, 79(8):1013–1019, December 2004. DOI: 10.1007/s00340-004-1657-5

[28] A. Ratkata, P.D. Gregory, A.D. Innes, *et al. Measurement of the tune-out wavelength for ^{133}Cs at 880 nm.* Physical Review A, 104(5):052813, November 2021. DOI: 10.1103/PhysRevA.104.052813

[29] K. Dieckmann, R.J.C. Spreeuw, M. Weidemüller, *et al. Two-dimensional magneto-optical trap as a source of slow atoms.* Physical Review A, 58(5):3891–3895, November 1998. DOI: 10.1103/PhysRevA.58.3891

[30] W. Ketterle, D.S. Durfee, and D.M. Stamper-Kurn. *Making, probing and understanding Bose-Einstein condensates.* In Proceedings of the International School of Physics ”Enrico Fermi”, course CXL, April 1999. DOI: 10.48550/arXiv.cond-mat/9904034

[31] W. Ketterle and N.J. Van Druten. *Evaporative Cooling of Trapped Atoms.* In Benjamin Bederson and Herbert Walther, editors, Advances In Atomic, Molecular, and Optical Physics, volume 37, pages 181–236. Academic Press, January 1996. DOI: 10.1016/S1049-250X(08)60101-9

[32] R. Grimm, M. Weidemüller, and Y.B. Ovchinnikov. *Optical dipole traps for neutral atoms.* In Advances In Atomic, Molecular, and Optical Physics, vol 42, pp. 95–170, Academic Press, February 1999. DOI: 10.48550/arXiv.physics/9902072

[33] G.F. Gribakin and V.V. Flambaum. *Calculation of the scattering length in atomic collisions using the semiclassical approximation.* Physical Review A, 48(1):546–553, July 1993. DOI: 10.1103/PhysRevA.48.546

[34] T. Shobu, H. Yamaoka, H. Imai, *et al. Optimized evaporative cooling for sodium Bose-Einstein condensation against three-body loss.* Physical Review A, 84(3):033626, September 2011. DOI: 10.1103/PhysRevA.84.033626

[35] A.D. Tranter, H.J. Slatyer, M.R. Hush, *et al. Multiparameter optimisation of a magneto-optical trap using deep learning.* Nature Communications, 9(1):4360, October 2018. DOI: 10.1038/s41467-018-06847-1

[36] I. Nakamura, A. Kanemura, T. Nakaso, *et al. Non-standard trajectories found by machine learning for evaporative cooling of ^{87}Rb atoms.* Optics Express, 27(15):20435–20443, July 2019. DOI: 10.1364/OE.27.020435

[37] A. Guttridge. *Photoassociation of Ultracold CsYb Molecules and Determination of Interspecies Scattering Lengths.* PhD Thesis, Durham University, 2018. URL: http://etheses.dur.ac.uk/12817

[38] J. Tricot, B. Maltzan. *arXiv dataset and metadata of 1.7M+ scholarly papers across STEM*. Cornell University. Accessed April 2022. URL: https://www.kaggle.com/datasets/Cornell-University/arxiv.

[39] I. Hughes and T. Hase. *Measurements and Their Uncertainties: A Practical Guide to Modern Error Analysis.* Oxford University Press, Incorporated, Oxford, UNITED KINGDOM, 2010. ISBN: 978-0-19-157656-0

A Jones Calculus Polarisation Derivation

Equation 7 is derived as follows, using Jones calculus [20]. Our input light vector $\vec{\mu}$ is,

$$\vec{\mu} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \tag{15}$$

representing horizontally polarised light. Then, the Jones matrices for a quarter-waveplate ($\mathbf{M}_{\text{QWP}}$), a half-waveplate ($\mathbf{M}_{\text{HWP}}$), and a linear polariser ($\mathbf{M}_{\text{LP}}$) are,

$$\mathbf{M}_{\text{QWP}} = e^{-i\pi/4} \begin{pmatrix} \cos^2\theta + i\sin^2\theta & (1-i)\sin\theta\cos\theta \\ (1-i)\sin\theta\cos\theta & \sin^2\theta + i\cos^2\theta \end{pmatrix}, \tag{16}$$

$$\mathbf{M}_{\text{HWP}} = e^{-i\pi/2} \begin{pmatrix} \cos^2\theta - \sin^2\theta & 2\cos\theta\sin\theta \\ 2\cos\theta\sin\theta & \sin^2\theta - \cos^2\theta \end{pmatrix}, \tag{17}$$

$$\mathbf{M}_{\text{LP}} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \tag{18}$$

where θ is the angle between the waveplate fast axis and the common horizontal axis of the waveplates. To propagate our initial light vector $\vec{\mu}$ through the system of optics, we use the matrix multiplication,

$$\vec{\eta} = \mathbf{M}_{\text{LP}} \cdot \mathbf{M}_{\text{HWP}} \cdot \mathbf{M}_{\text{QWP}} \cdot \vec{\mu}, \tag{19}$$

where $\vec{\eta}$ is the Jones vector of the light after the system of optics. Then, $\vec{\eta}$ is,

$$\vec{\eta} = e^{-3i\pi/4} \begin{pmatrix} (i\sin^2\theta_q + \cos^2\theta_q)\cos(2\theta_h) + \frac{1}{4}(1-i)(\cos(2\theta_h - 2\theta_q) - \cos(2\theta_h + 2\theta_q)) \\ 0 \end{pmatrix}, \tag{20}$$

where θ_q and θ_h are the rotations of the quarter- and half-waveplates, respectively. Since the elements in the Jones vectors represent electric fields, the intensity I of the light is given by the square $\vec{\eta}$,

$$\begin{aligned} I &= \frac{1}{2}I_0(\quad \cos^2(2\theta_h) + \cos^2(2[\theta_h - \theta_q])), \\ I &\propto \qquad \cos^2(2\theta_h) + \cos^2(2[\theta_h - \theta_q]), \end{aligned} \tag{21}$$

where I_0 is the intensity of the incident light.

Parameter	Optimised value
I_0 / mV	48.0 ± 0.1
I_A / mV	1686 ± 1
A_h / deg/step	0.050 ± 0.002
A_q / deg/step	0.100 ± 0.004
ϕ_h / deg	27 ± 6
ϕ_q / deg	50 ± 5

Table 2: The optimised parameters for the fit of equation 8 to figure 12a. Each parameter has an associated error derived from the square-root of the diagonal elements of the covariance matrix returned by the fit [39].

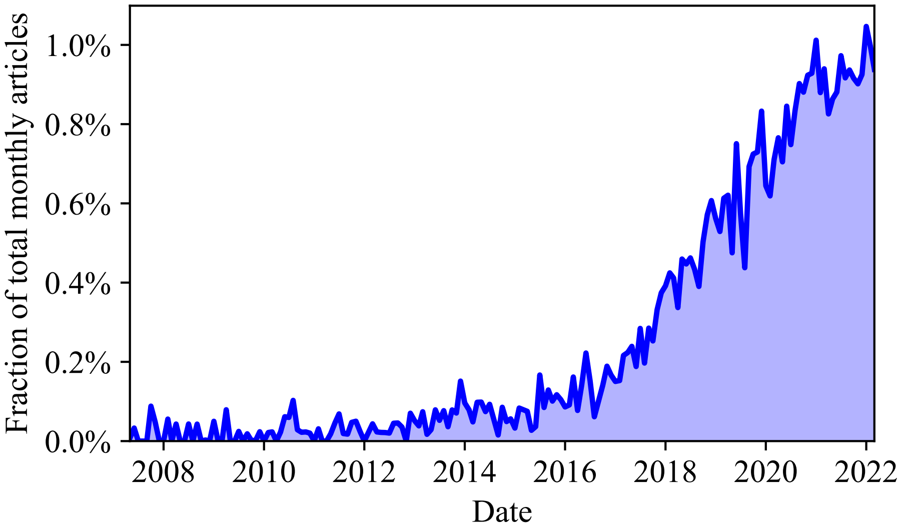

Figure 23: A summary of the relative abundance of machine learning papers on the open-access academic archive arXiv.org. The blue line shows the relative percentage over time of papers containing the phrase ‘machine learning’ in the title. Data used from [38].

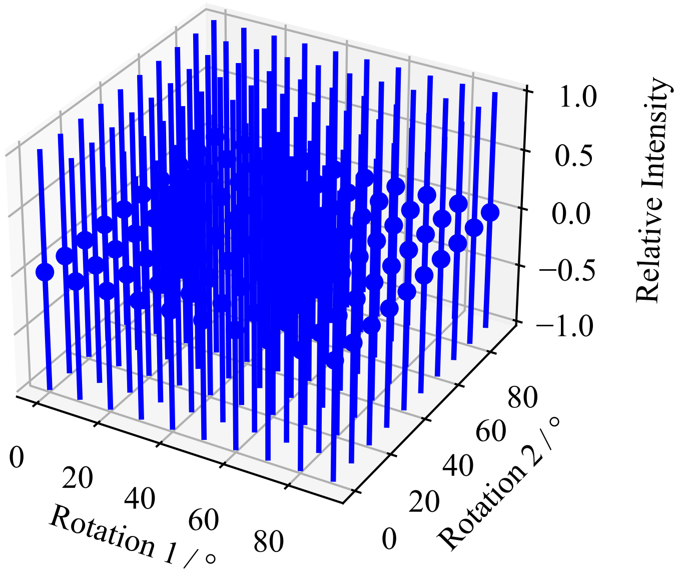

Figure 24: A representation of a 2-dimensional Gaussian process. The second parameter is the rotation of an additional waveplate.

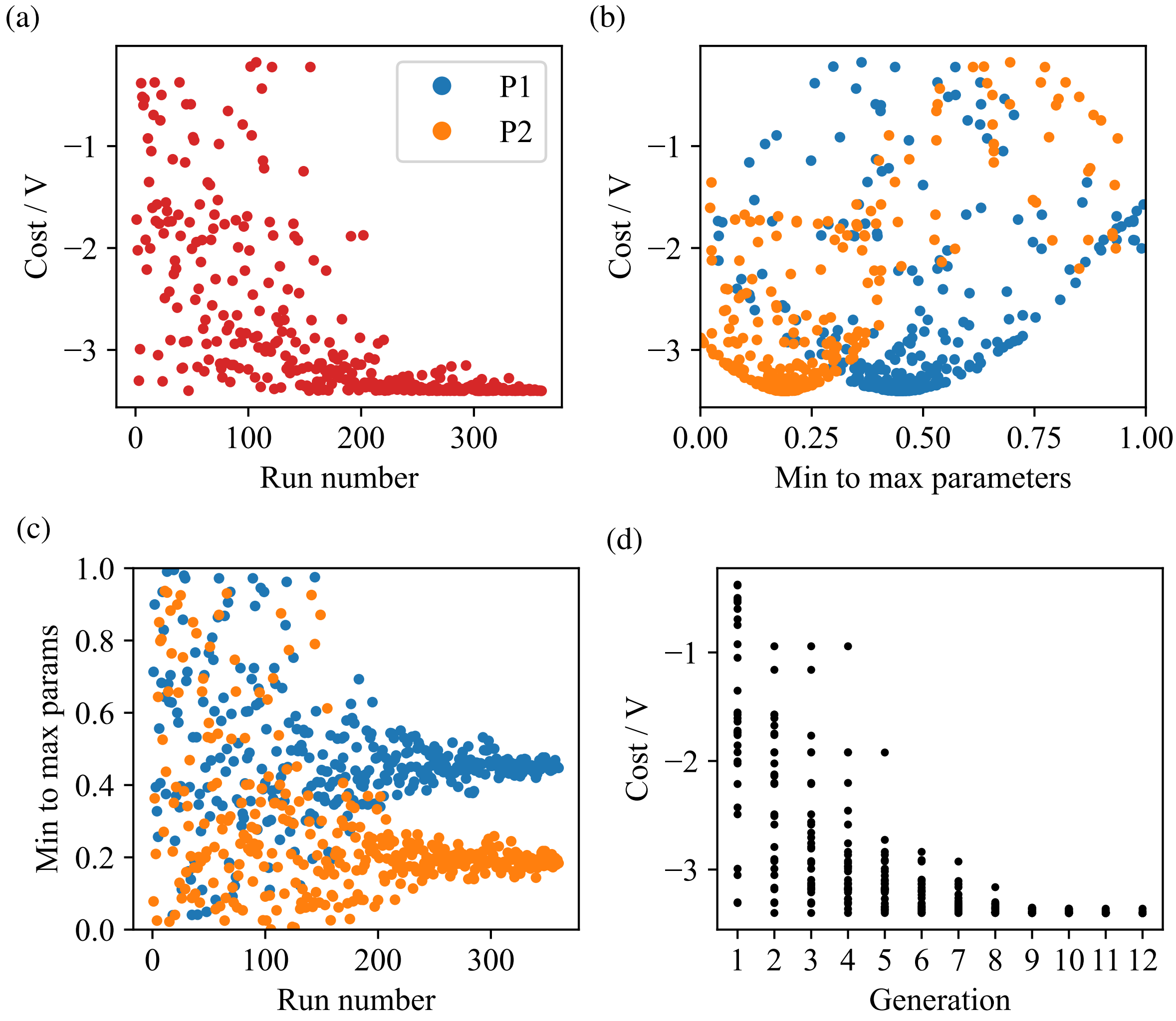

Figure 25: An example of the output of M-LOOP, for a differential evolution optimisation of the experiment from section 3. **(a)** The cost of the experiment over time. The downward trend shows the experiment is improving. **(b)** Each run of the experiment is also plotted as a cost vs parameter graph. One can see more exploration is performed nearer the optimum (i.e., the bottom) of this plot. **(c)** The parameter values over time, which as the experiment improves, converge to their optimal values. **(d)** The evolution of the 30 individuals in the differential evolution process over time, which as the process evolves, converge to the minimum experimental cost.