# Applying machine learning optimization methods to the production of a quantum gas

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
28 August 2019

REVISED
16 December 2019

ACCEPTED FOR PUBLICATION
19 December 2019

PUBLISHED
25 February 2020



PAPER

# Applying machine learning optimization methods to the production of a quantum gas

A J Barker[1], H Style[1], K Luksch[1], S Sunami[1], D Garrick[1], F Hill[2], C J Foot[1] and E Bentine[1]

[1] Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, United Kingdom
[2] DeepMind, 6 Pancras Square, London, N1C 4AG, United Kingdom

**E-mail:** adam.barker@physics.ox.ac.uk

**Keywords:** ultracold quantum matter, machine learning, artificial neural networks, Bose–Einstein condensates

## Abstract

We apply three machine learning strategies to optimize the atomic cooling processes utilized in the production of a Bose–Einstein condensate (BEC). For the first time, we optimize both laser cooling and evaporative cooling mechanisms simultaneously. We present the results of an evolutionary optimization method (differential evolution), a method based on non-parametric inference (Gaussian process regression) and a gradient-based function approximator (artificial neural network). Online optimization is performed using no prior knowledge of the apparatus, and the learner succeeds in creating a BEC from completely randomized initial parameters. Optimizing these cooling processes results in a factor of four increase in BEC atom number compared to our manually-optimized parameters. This automated approach can maintain close-to-optimal performance in long-term operation. Furthermore, we show that machine learning techniques can be used to identify the main sources of instability within the apparatus.

## 1. Introduction

Recent developments in artificial intelligence and machine learning have provided tools with which a computer can now outperform the analytic capability of a human, particularly when data sets are large or when a system relies on many free parameters [1]. The application of machine learning methods has led to dramatic advances in many scientific fields and contexts, such as supply chain forecasting and healthcare [2, 3]. Machine learning is also well suited to the optimization of a complex experimental apparatus [4–6]. As compared to a human, a major advantage of many machine learning methods is that the chosen learner has no preconceptions for how the parameters should affect the final result, and is therefore objectively guided purely by the actual data. As a result, a machine learner is able to find counter-intuitive solutions that a trained experimentalist may overlook [5].

In this paper, we apply three different machine learning algorithms to optimize an atomic physics experiment. Our apparatus is designed to produce a Bose–Einstein condensate (BEC), a quantum-mechanical state of matter which occurs when bosonic particles accumulate in their lowest energy (ground) quantum state [7]. Bose–Einstein condensation in a dilute atomic vapor was first realized in 1995, resulting in the award of the Nobel Prize in 2001 [8, 9]. Since then, ultracold atomic vapor experiments have been used to investigate a wide range of physical phenomena, including quantum many-body physics [10], quantum-mechanical phase transitions [11, 12] and superfluid turbulence [13].

To observe the BEC phase transition in dilute gas experiments, extremely low temperatures of tens of nanokelvin are typically required. The techniques used to reach these ultracold temperatures usually include a combination of optical cooling and forced evaporative cooling [8, 9, 14]. Implementing these cooling processes requires the precise sequencing of time-varying magnetic and optical fields using a control computer. We parametrize these fields by defining their values at specific times, and refer to these definitions as the 'settings' that describe a given sequence. The parameter space that describes a typical experimental sequence is large and locating the optimal experimental settings using exhaustive, brute-force searches is unfeasible.

Given the large parameter space, analytic models are often used to predict the optimal experimental settings. Well-established theory exists to explain several of the typical stages common to cold-atom apparatuses. For example, the cooling of atoms by the radiation forces exerted by laser light has been investigated for decades [15] and forced evaporative cooling in optical or magnetic traps is routinely used [16]. The theories describing these stages of cooling contain approximations and, furthermore, the apparatus can suffer from unknown imperfections or external perturbations. These limitations are usually mitigated by employing further manual optimization of the experimental settings after using the theoretical optimum predictions as a starting point.

Recently, machine learning techniques have been applied in the field of ultracold quantum matter to optimize individual laser cooling [5, 17] and evaporative cooling [4, 18] stages, achieving significant improvements in the performance of these apparatuses. The optimizations in each case were performed on a subset of the atomic cooling processes [19, 20], and did not consider the changeover between each process, which increases the likelihood that the entire cooling sequence will become trapped in a local optimum.

In this paper, we present the results of a simultaneous optimization of all atomic cooling stages involved in our experimental sequence. Additionally, we compare the efficacy and rate of convergence of three common algorithms when applied to our optimization problem. The exact nature of what constitutes an optimized quantum gas experiment depends on the user's requirements. For example: quicker experiments with a higher repetition rate produce a greater amount of data in a given time; lower temperatures of the atomic cloud can improve the precision of spectroscopic measurements [21]; a larger atom number or higher peak density can improve the signal-to-noise ratio when imaging the BEC. Here, our chosen metric for optimization consists of maximizing the atom number in a BEC, unless stated otherwise.

We define our methods of optimization in section 3 and implement these using an open-source software package (Machine Learning Online Optimization Package (M-LOOP)) [22], which has previously been used to optimize evaporative cooling elsewhere [4]. The improvements in experimental performance that result from the optimization of several cooling stages, both individually and collectively, are then presented. We utilize one particular optimization method to identify experimental settings which most strongly affect the result [4]; this also highlights likely sources of instability within the experiment. Finally, we modify our optimization metric to minimize the sequence time required to produce a BEC, which is desired when performing tasks such as optical alignment, or to collect more data when atom number is not a priority.

## 2. The experimental apparatus

We now describe our experimental apparatus and the several stages of trapping and cooling of an atomic vapor which lead to the production of a BEC [21, 23]. An outline of the apparatus and optimization scheme is illustrated in figure 1.

### 2.1. Producing a BEC

First, atoms are laser cooled in a magneto-optical trap (MOT) [25], which collects a fraction of atoms from a room temperature vapor and cools them to around the Doppler limit (146 $\mu$K for $^{87}$Rb) [26]. After fully loading the MOT, the trapped atoms are subjected to a sudden compression and further cooling during a 'compressed' MOT (cMOT) stage, which acts to further reduce the temperature by roughly an order of magnitude [27, 28]. The efficiency of the cooling and compression is dependent on many factors which include the detuning of the laser light from the atomic resonance and the strength of the applied magnetic field. The cold cloud is loaded into a magnetic quadrupole trap and transported to an ultra-high vacuum region by physically translating the field-producing coils. Subsequently, evaporative cooling is performed to further reduce the cloud temperature.

Evaporative cooling can be understood from the following arguments. Atoms in a gas at a finite temperature occupy a distribution of energies, as described by the Maxwell–Boltzmann distribution [29]. Evaporative cooling is performed by selectively ejecting the highest-energy atoms, which reduces the average energy of the remaining atoms. The trapped atoms then rethermalize through collisions, which re-establishes a Maxwell–Boltzmann distribution characterized by a lower temperature [30]. In our case, evaporation is performed by the application of a weak radiofrequency (RF) field, colloquially referred to as a 'knife'[3], which removes atoms with energy above a threshold determined by the frequency of the applied knife.

Evaporation is first performed in a magnetic quadrupole trap and later in a time-averaged orbiting potential (TOP) trap [9, 31]. The quadrupole trap is implemented using a pair of coaxial current-carrying coils to produce a magnetic quadrupole field that confines the atoms. After the RF knife is applied to the trapped atoms, the frequency is slowly reduced; as the evaporation stage progresses, this reduces the threshold energy at which atoms are removed and thus reduces the cloud temperature.

[3] The applied RF field effectively cuts away the high energy tail of the Maxwell–Boltzmann distribution, hence it is termed a 'knife'.

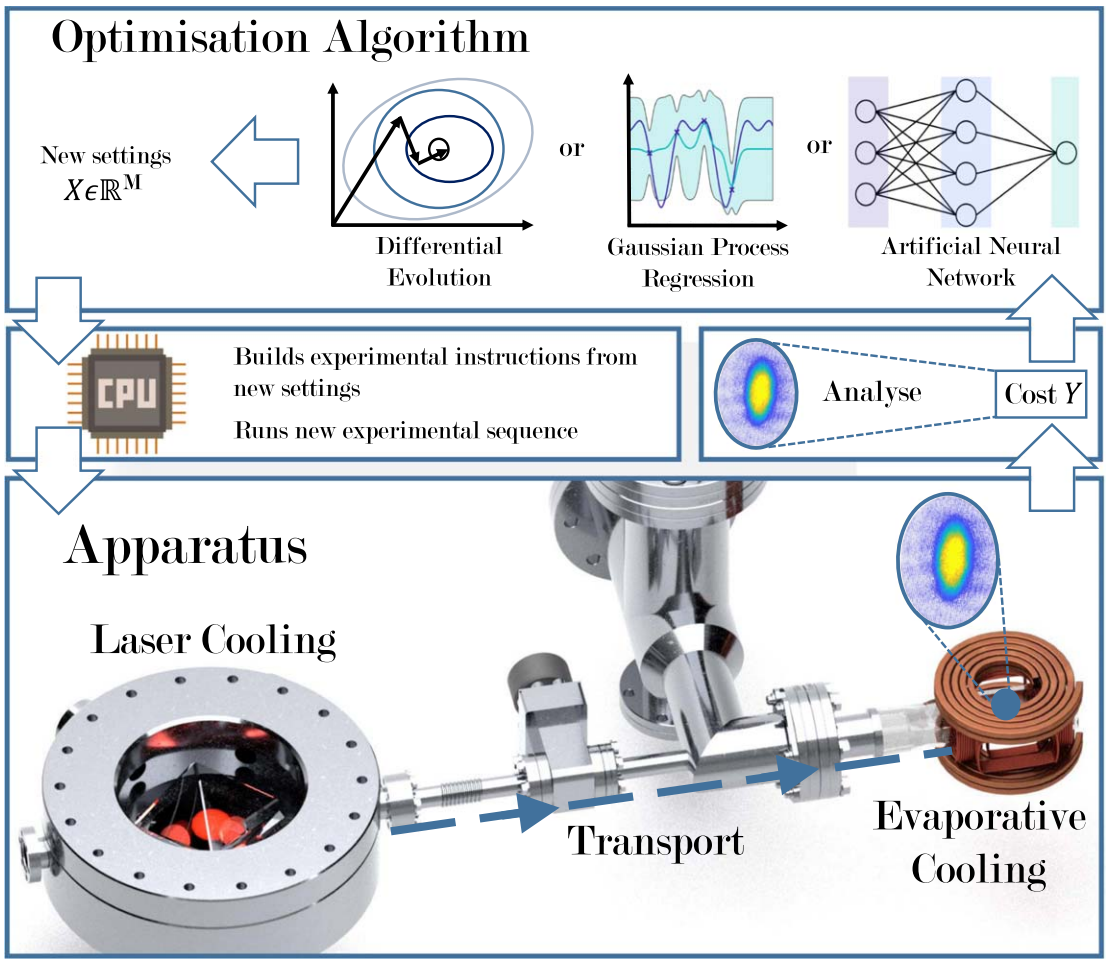


**Figure 1.** The experimental apparatus and the optimization loop. The atomic gas is initially trapped and laser cooled by a combination of laser light and magnetic fields. The trapped cloud is then transported to an ultra-high vacuum region where evaporative cooling is performed. An image of the resulting cloud is taken using absorption imaging [24] and is analyzed to evaluate the cost, which is calculated from the atom number in the cloud. The cost is passed to the chosen optimization algorithm which produces a new set of experimental settings. These settings are processed by a control computer into a new set of instructions and used in the next sequence.

After a first stage of evaporative cooling, atoms are loaded into the TOP trap. This typically occurs once a temperature of 1 $\mu$K is reached. The magnetic field of the TOP trap combines a static quadrupole field with a rotating bias field that lies in the horizontal plane, and is of the form $B = B_x \cos(\omega t)\hat{\mathbf{e}}_x + B_y \sin(\omega t)\hat{\mathbf{e}}_y$, where $\omega = 2\pi \times 7$ kHz is the field rotation frequency, $\hat{\mathbf{e}}_{x,y}$ are the Cartesian axes in the horizontal plane and $B_x$ and $B_y$ are the amplitudes of the quadratures of the field. $B_x$ and $B_y$ can be individually controlled to produce an elliptically polarized field, with the ellipticity expressed by $\epsilon = B_y/B_x - 1$. Further evaporative cooling proceeds in the TOP trap using the RF knife as before. Overall, the evaporative cooling processes in both traps are described by a number of settings which vary in time and include: the quadrupole coil current $I_Q$, the RF knife frequency and, in the case of the TOP trap, the amplitude and ellipticity of the TOP field.

The experimental settings are processed by the control computer in order to direct the apparatus during the sequence. By adjusting these settings between successive sequences, we are able to optimize the production of a BEC.

### 2.2. Observing a BEC

After all stages of cooling have been completed, the atomic cloud is released from the trap. The cloud undergoes a period of free fall, during which it expands ballistically, before an image is taken [24]. This 'time-of-flight' (TOF) expansion allows us to observe the momentum distribution of the cloud. The expansion dynamics of a gas in the quantum regime are distinct from those of a thermal gas [30]. This difference produces a bimodal spatial distribution of atoms after TOF: the BEC component is responsible for a dense 'core' of atoms which lies within a broader 'pedestal' of thermal atoms. This bimodal distribution is evidence that a BEC has been produced. The absorption image is analyzed to determine properties of the cloud, such as the atom number, which are used in the calculation of the cost.

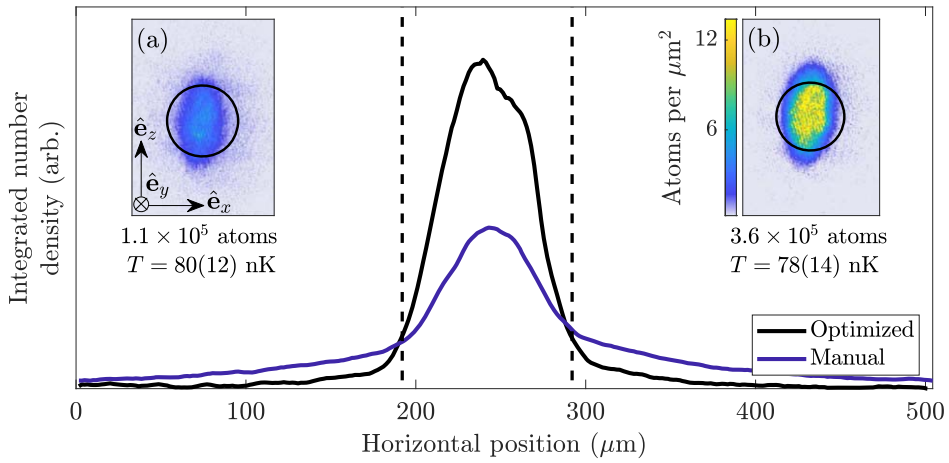


**Figure 2.** Illustration of the results of the optimization via machine learning versus manual optimization. We compare the absorption images, showing the atomic density integrated along the imaging direction $\hat{\mathbf{e}}_y$, of a manually-optimized BEC (a) and a BEC where the evaporative cooling stages of both the quadrupole and TOP traps have been optimized using the GP method (b). Both inset images show the region (black line) in which the counted atom number contributes to the cost. The total atom numbers and temperatures of the clouds are shown below the images. The plot shows a direct comparison of the images integrated along the direction of gravity, $\hat{\mathbf{e}}_z$. The horizontal extent of the region of interest, within which atoms are counted in the cost function, is indicated by the dashed lines.

### 2.3. Machine learning methods

The goal of optimization is to identify the global optimum within a parameter space. In our experiment, the parameter space is that spanned by $M$ experimental settings (currents, voltages, timings etc). A point in this parameter space is given by a vector of experimental settings $\mathbf{X} \in \mathbb{R}^M$[4]. Each point in space has an associated cost $Y = f(\mathbf{X}) \in \mathbb{R}$, generated by a cost function $f(\mathbf{X})$ [32]. The cost function quantifies the desirability of a measured outcome, and is used to steer the optimization.

There are a number of candidate cost functions that could be used for a cold atom experiment. Figure 2 illustrates an example absorption image, from which a wealth of data may be obtained. The bimodal density distribution after TOF expansion clearly indicates the presence of a BEC, as explained in section 2.2. Consequently, we define our cost function to be proportional to $-\log(\tilde{N})$, where $\tilde{N}$ is the number of atoms within a small region of interest which is chosen to be comparable to the approximate extent of a typical BEC after TOF expansion [30]. Atoms above a threshold momentum are not contained within this region after TOF expansion, and therefore do not contribute to $\tilde{N}$. Further detail justifying our choice of cost function is presented in the appendix. We choose the cost function to be the logarithm of $\tilde{N}$, as the value of $\tilde{N}$ can span several orders of magnitude during the optimization; bad settings may result in no atoms detected above the noise floor, whereas BECs typically contain approximately $10^5$ atoms.

Although analytic functions exist which describe the bimodal distribution, they contain many free parameters. This can make fitting unreliable, especially for the low atom numbers present in the early stages of optimization, and parameters extracted from such fits can throw the learner off course. Our simple cost function is robust against these issues. Physically, our cost function can be interpreted as measuring the population of atoms with momentum close to zero, which increases as the optimization progresses towards producing a BEC. Previous work employed a cost function derived from the fitted width of the cloud, with two repeats of the experiment per experimental settings generation [4].

The optimization feedback loop, outlined in figure 1, can be summarized as follows: the machine learner is configured with an initial $M$-dimensional vector $\mathbf{X}_0$ of experimental settings. We also configure the allowed ranges that each setting can take, to ensure that generated sequences will not damage the apparatus. $\mathbf{X}_0$ is read by an experimental control computer, which defines relevant analog and digital outputs at time steps accordingly. The sequence is run and the resulting image is analyzed to produce a cost $Y_0$. This pair of settings and cost is then used by the chosen optimization algorithm to determine the next settings $\mathbf{X}_*$ to be tested. Each new settings/cost pair updates the learner's knowledge of how the cost depends on each setting [33]. Our problem describes a settings/cost landscape with no initial data and is an example of online optimization. We terminate the optimization after a fixed number of sequences or when no further improvement to the cost has been achieved after 35 sequences.

[4] Although the settings are continuously varying (up to floating-point precision), bounds on each setting are imposed, owing to physical limitations or for safety reasons, hence the set is not strictly $\mathbb{R}^M$.

To implement the optimization routines, we utilize an open-source machine learning toolkit: M-LOOP, which is based on the Python scikit-learn library [4, 22]. This toolkit contains several optimization routines which are described in section 3.

## 3. Optimization methods

We compare the efficacy of three algorithms to optimize our experiment: an evolutionary optimization method (differential evolution (DE)) [34], a regression method based on non-parametric inference (Gaussian process (GP) regression) [35] and a gradient-based (parametric) function approximator (artificial neural network (ANN)) [1].

The optimization methods are tested in the context of non-convex optimization: the cost function described earlier is in general non-convex and thus it is possible that any method may not converge to the global optimum. The likelihood of finding the global optimum can be increased by performing many optimization procedures with varying initial conditions.

We note that the optimization methods are robust to random variations in cost for a given setting. This is appropriate for an experimental apparatus in which random fluctuations are present, either due to variation in the performance of laboratory equipment or because the results depend intrinsically on random processes (e.g. shot noise fluctuations in the atom number). This does not fundamentally prevent the algorithms from finding a good solution, but uncertainty in the cost increases the number of experimental sequences required for the solution to converge.

### 3.1. Differential evolution

Evolutionary algorithms involve several key stages, which are inspired by biological evolution [36]. First, an initial population is generated randomly. New individuals are then produced by mixing features of pre-existing individuals (crossover) and by adding random variation (mutation). Finally, selection is performed by assessing the fitness of new individuals and by replacing the population with the lowest fitness.

In the present work, we use the DE algorithm. In this context, the individuals are settings vectors $\mathbf{X}_i$ and the fitness is the associated cost $Y_i$ of each vector. The initial population is a randomly generated set of $n$ vectors $\{\mathbf{X}_1,\ldots,\mathbf{X}_n\}$ and their experimentally measured costs $\{Y_1, \ldots, Y_n\}$. Mutation produces a new vector $\mathbf{V} = \mathbf{X}_k + (\mathbf{X}_i - \mathbf{X}_j)$, where $\mathbf{X}_i$, $\mathbf{X}_j$ and $\mathbf{X}_k$ are randomly selected vectors [34]. Crossover is achieved by selecting elements randomly from either $\mathbf{X}_i$ or $\mathbf{V}$ to create a new candidate vector $\mathbf{X}_*$. A sequence is then performed using vector $\mathbf{X}_*$ and the value of the cost function $Y_*$ is measured, producing an additional settings/cost pair $\{\mathbf{X}_*, Y_*\}$. Selection is performed by determining whether $Y_* < Y_i$; if so, $\mathbf{X}_*$ replaces $\mathbf{X}_i$ and the process repeats.

The DE method has low computational complexity and requires a small number of vectors from which to begin. However, the simplicity of the method results in slow convergence towards a solution. Nevertheless, we utilize this method to build a set of settings/cost pairs which serves as a starting point to initially train the other optimization methods.

### 3.2. GP regression

Bayesian inference provides us with tools to update a prior hypothesis of a probability distribution based on new data, namely Bayes' rule [35]. In general, a GP is a probability distribution of functions which describe a given dataset. GP regression utilizes Bayes' rule to update this probability distribution given new data [37]. Prior knowledge about a point in parameter space can be invoked in terms of a kernel function; a kernel is a measure of similarity between two inputs separated by a distance in parameter space. A popular choice is the squared-exponential, or Gaussian, distribution kernel $K(\mathbf{X}_i, \mathbf{X}_j)$:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \exp\left\{-\frac{1}{2}\sum_{k=1}^{M}\eta_k(\mathbf{X}_i[k] - \mathbf{X}_j[k])^2\right\}, \tag{1}$$

where $\mathbf{X}_i[k]$ represents the (dimensionless) $k$th element in the vector $\mathbf{X}_i$, the dimensionless parameters $1/\eta_k$ are the characteristic length-scales for each parameter and the summation runs over all settings $k$. The $\eta_k$ are generated when performing GP regression, and provide a measure of how strongly the kernel depends on changes to each of the parameters.

In our context, the function that we fit using GP regression is the mapping between the experimental settings and the experimentally measured cost. Given an existing set of settings/cost pairs $\{\mathbf{X}_i, Y_i\}$, we can estimate the cost (and uncertainty) of any settings $\mathbf{X}_*$ according to the GP fit. We can therefore search for new experimental settings with the lowest predicted cost and iterate within our optimization loop. To facilitate a comparison of $\eta_k$ across all settings, we normalize each $\mathbf{X}[k]$ with respect to the minimum and maximum allowed values for the

$k$th setting. Before applying the GP method, a training set of $2M$ settings/cost pairs is constructed using the DE method.

### 3.3. Artificial neural network

ANNs are an example of a function approximator and take the form of an interconnected network of nodes [1, 38]. An ANN produces a 'black-box' mapping between an input and an output. In our context, the inputs are settings vectors $\mathbf{X}$ and the outputs are the associated costs $Y$. The mapping is determined by the structure and weights of connections in the network, with a connection structure which is intrinsically linear. To incorporate the non-linearity of the cost function, we include the Gaussian error linear unit (GELU) activation function for each node [39]. This continuous function is a popular choice for data which is subject to normally-distributed stochastic variation, which suits our experimental context [40]. In addition, the structure and scale of the ANN must be appropriate for the complexity and size of the vector inputs. We choose a network comprised of 3 hidden layers of 8 fully-connected neurons, inspired by [41], which is sufficient for the number of settings that we optimize in this context (a maximum of 35). An initial training set of $2M$ settings/cost pairs is produced using the DE method.

We utilize the Adam optimization method [42] to update the ANN given new training data. This method is widely used for gradient-based optimization of cost functions with stochastic noise. The method is straightforward to implement and is computationally efficient; the method is also appropriate for problems with very noisy or sparse gradients. In comparison to other classical gradient descent methods, the Adam method utilizes higher-order moments of the gradients of each parameter [43], which often leads to a comparatively faster rate of learning. [42]. We use the trained ANN to search for optimal predicted settings $\mathbf{X}_*$. A sequence is then run using $\mathbf{X}_*$ and the cost $Y_*$ is measured. This settings/cost pair is then used to refine the ANN for future sequences.

## 4. Results

We applied the algorithms presented above to optimize the atomic cooling processes utilized in the production of BECs. We begin by presenting the optimization of evaporative cooling in the quadrupole and TOP traps. This optimization also identified the settings that most strongly affected the cost function. Similarly, we optimized the cMOT laser cooling stage. We combined the sensitive settings in both the laser cooling and evaporative cooling stages to perform a full optimization of all cooling processes involved in the production of a BEC. Finally, we altered the cost function to favor faster sequences, finding settings which produced a BEC of a threshold atom number within the shortest sequence duration.

### 4.1. Optimizing evaporative cooling

Prior to machine learning optimization, our manually-optimized settings produce a BEC of $1.1 \times 10^5$ atoms. This produces a cost of 8.9 when using a circular region of interest of radius 50 $\mu$m located about the cloud center after 23 ms of free fall. These settings can be used as a starting point for machine learning optimization, leading to rapid convergence towards the optimum settings and providing a useful way to quickly retune the experiment.

In order to properly compare the different learners, we instead begin each optimization using completely randomized settings. These initial settings produce no visible atom cloud. Figure 3 shows the cost as a function of experimental run number. The optimization is continued until no further improvement is found within 35 cycles or until a maximum of 180 sequences, which limits the optimization process to a maximum duration of approximately 3 h. We perform one optimization routine for each method.

The GP method converged to a BEC of $3.8 \times 10^5$ atoms after 47 sequences, whereas DE did not converge within the time limit. The ANN method produced a BEC of $3.2 \times 10^5$ atoms after 117 sequences with a rate of convergence which is between those of the GP and DE methods[5]. For both the GP and ANN methods, the optimization procedure resulted in a factor of 3 increase in BEC atom number as compared to the manually-optimized settings. The settings that produced the best cost are shown in table 1 with the original settings used prior to optimization shown in parentheses. Figures 4(a) and (b) illustrate the progression of the experimental settings during the optimization, namely the quadrupole current $I_Q$ and RF knife frequency, respectively. The settings are plotted against the duration of each substage.

The cloud density profile after TOF expansion becomes bimodal as the cost drops below approximately 9.2, indicating the presence of a BEC component. This threshold is achieved after 156 sequences (DE), 14 sequences

[5] For both the GP and ANN methods, the quoted number of sequences does not include the training set of $2M = 70$ sequences which was produced using DE.

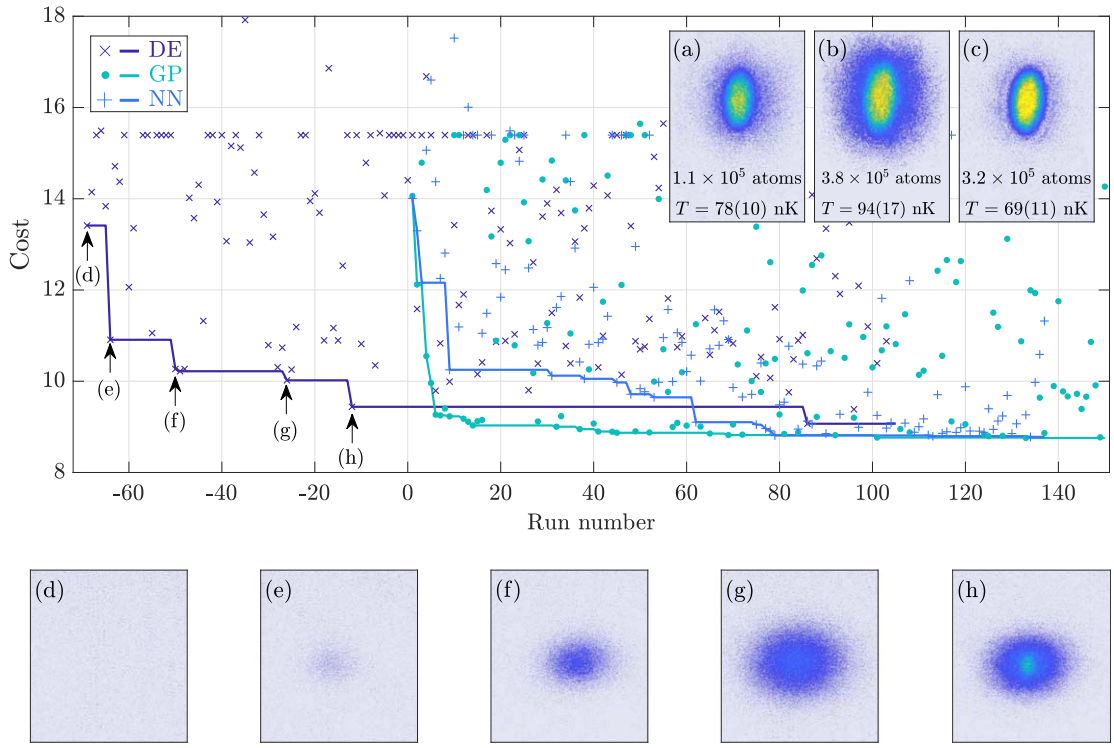


**Figure 3.** Optimizing the quadrupole and TOP evaporative cooling stages, beginning from random initial settings. Data points for the measured cost are illustrated as (DE), (GP) and (ANN). In addition, the moving minimum for each of the three methods is indicated by the solid lines. Inset (a), (b), (c): absorption images of BECs produced using the best settings found for DE, GP and ANN-based optimization, respectively, including the atom number and temperature for each cloud. Sequences used in building the training set using the DE method are labeled with negative numbers. Absorption images below correspond to points labeled (d)–(h). These images illustrate the progression of the optimization from initial settings, for which no trapped atoms were detected, towards finding settings which produce a BEC. The images use the same color scheme as used in figure 2.

**Table 1.** The best settings found for evaporative cooling stages in the magnetic quadrupole (Quad) and TOP traps; these were found using the GP method but are very similar to those found using the ANN method. The values shown define points which are linearly interpolated to produce the evaporation instructions. Numbers in parentheses represent the manually-optimized settings used prior to the optimization. Values shown without brackets were not included in the optimization.

| | Substage | Duration (s) | | $I_Q$ (A) | | RF knife (MHz) | | $B_x$ (G) | | Ellipticity, $\epsilon$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Quad | 0 | 0 | | 323 | (315) | 120 | | | | | |
| | 1 | 18 | | 323 | (315) | 15 | (18) | | | | |
| TOP | 2 | 0 | | 83 | (60) | 26 | (32) | 2.6 | (3.6) | 0 | (0) |
| | 3 | 0.08 | (0.08) | 142 | (131) | 29 | (26) | 19 | (18) | 0 | (0) |
| | 4 | 8.1 | (7.0) | 237 | (226) | 9.1 | (9) | 6.6 | (7.8) | 0.06 | (0) |
| | 5 | 1.1 | (0.8) | 213 | (226) | 10 | (8.5) | 6.3 | (7.8) | −0.15 | (0) |
| | 6 | 1.8 | (1.8) | 249 | (226) | 14 | (7.8) | 9.9 | (7.8) | 0.04 | (0) |
| | 7 | 6.3 | (3.3) | 222 | (226) | 9.5 | (6.7) | 9.0 | (7.8) | 0.09 | (0) |
| | 8 | 5.5 | (1.5) | 200 | (226) | 6.9 | (6.5) | 7.8 | (7.8) | 0.11 | (0) |

(GP) and 75 sequences (ANN). Overall, the relative convergence rates of the methods differ significantly, as expected; the slower convergence of the ANN, as compared to GP, is representative of the large amount of data required to train a fully-connected network. DE proceeds the slowest of the three, which is expected given its simplistic approach to generating subsequent settings. As there is an element of randomness as to how the DE method chooses points to evaluate, it is possible that one model may have chanced upon good settings early in the optimization procedure which then strongly guided its subsequent choices. However, the data sets used to train the GP and ANN methods were comparable and contained settings with minimum costs of 9.4 and 9.6, respectively, giving a fair comparison between the learners.

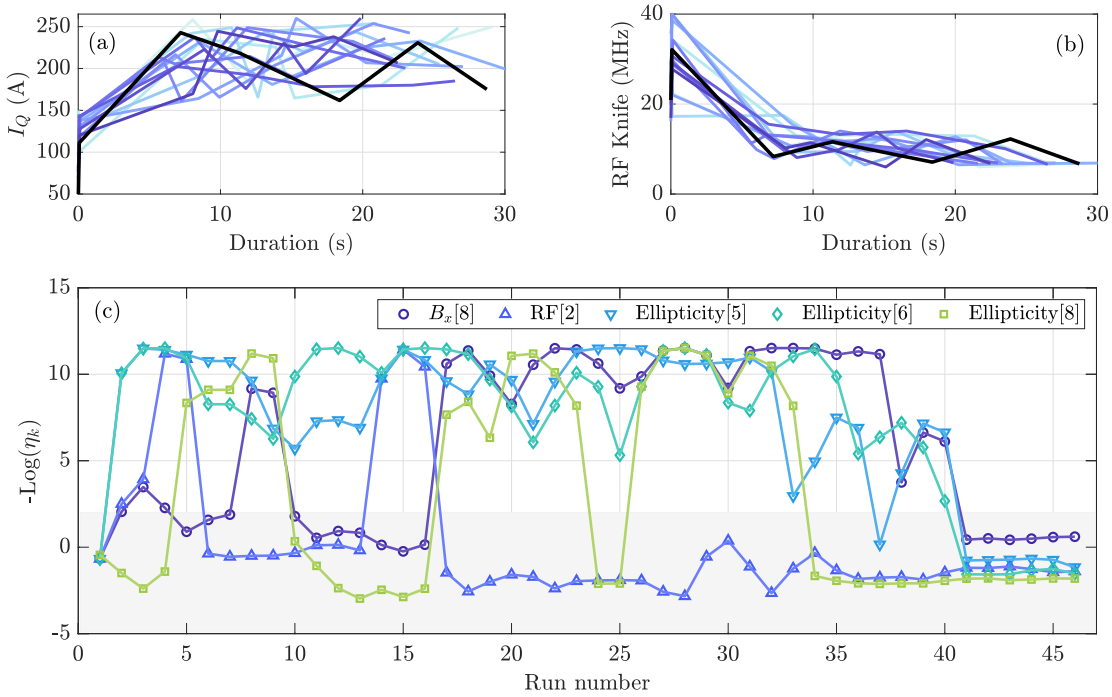


**Figure 4.** (a) and (b) illustrate the progression of the quadrupole current $I_Q$ and RF knife frequency settings during the TOP substages, respectively, as produced during the optimization. The settings are plotted against the duration of the evaporative cooling stage. Darker colors indicate settings produced later into the optimization procedure. The minimum cost is achieved with the settings shown in black. (c) Shows the evolution of the five most sensitive $\eta_k$ as more data is added during the optimization routine. The settings shown are indexed by a number which indicates the substage of the evaporative cooling stage, as given in table 1. The shaded area indicates the region of $-\mathrm{Log}(\eta_k) < 2$ within which a setting is deemed sensitive.

After only a few hours, the result of the optimization procedure is a greater than threefold improvement in the BEC atom number, as compared to a BEC produced with settings which have been manually optimized over many years. This improvement is illustrated in figures 2(a) and (b), which show BECs produced using the evaporation settings before and after optimization, respectively. Figure 2 also demonstrates the merit of defining a small region of interest, which discounts atoms from the broader, thermal fraction of the atomic distribution.

### 4.2. Sensitivities

We utilize the cost landscape fitted by the GP learner to determine the sensitivity of the different settings. As detailed in section 3.2, the variables $\eta_k$ provide a measure of how steeply the cost function varies about the predicted minimum with respect to each setting. As a heuristic indicator of sensitivity, we define a setting to be sensitive if the associated $\eta_k$ is greater than $\exp(-2)$. Figure 4 illustrates the convergence of $\eta_k$ as more data is added during the evaporative cooling optimization. For clarity, only the five most sensitive settings are shown.

We find that the cost is highly sensitive to the final amplitude of the magnetic field in the TOP trap ($B_x$[7]), as well as the initial (RF[1]) and final (RF[7]) radiofrequencies of the knife. This can be understood as follows: the initial frequency determines the threshold energy above which atoms are ejected from the trap; this frequency must be sufficiently high so as not to immediately cut away a large number of atoms when the RF knife is first turned on when evaporation begins. A combination of final knife frequency and TOP amplitude determines the final, lowest energy cut in the evaporation ramp. If this is too high, the cloud is hotter and fewer atoms accumulate within the region-of-interest after TOF. If this is too low, the evaporation sequence unneccesarily ejects atoms which would otherwise have contributed to the BEC component. The cost is also sensitive to the final RF knife cut in the quadrupole trap, as this determines the temperature of the atomic cloud when it is loaded into the TOP trap.

Surprisingly, we find the cost is highly sensitive to the TOP field ellipticity during certain substages. This setting was fixed to 0 during previous manual optimization, as this was expected to yield the best results. From observations of cloud positions when trapped in the quadrupole or TOP trap, we have determined that the rotation axis of the TOP field is not perfectly aligned with the symmetry axis of the quadrupole field, which increases the displacement between the energy minima of an atom in these two traps. In addition to any center-of-mass motion of the cloud, which may be induced as the cloud is transferred from the quadrupole to the TOP trap, other multipole oscillations in the cloud may be excited. We postulate that a non-zero ellipticity in the TOP field provides an asymmetric confinement force, which may help to eradicate or damp excitations in the cloud

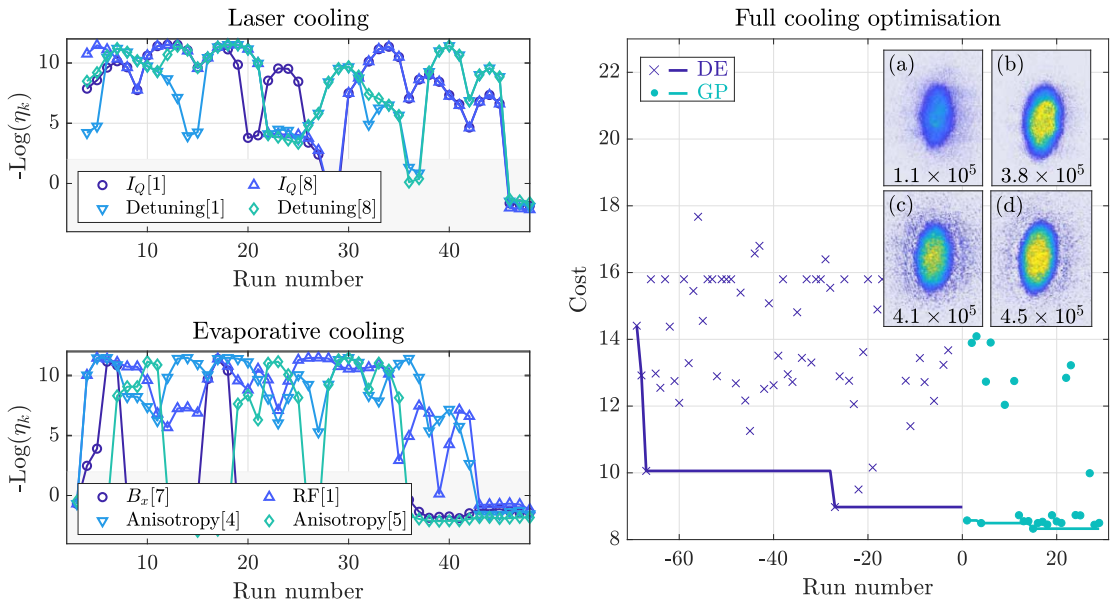


**Figure 5.** Cost versus run number for optimization of all cooling processes, using the GP method. Laser cooling panel: evolution of the 4 most sensitive $\eta_k$ during optimization of laser cooling (cMOT). Evaporative cooling panel: evolution of the 4 most sensitive $\eta_k$ during optimization of evaporative cooling. Right: cost versus run number for the overall cooling optimization, using the 18 most sensitive settings, for the initial DE method (×) and the subsequent GP method (•). The moving minimum cost is illustrated by the solid line. Sequences used in building the training set using the DE method are again labeled with negative numbers. Insets: (a)—manually-optimized BEC, (b)—BEC following evaporative cooling optimization, (c)—BEC following laser cooling optimization and (d)—BEC following the full cooling optimization. The temperature, as extracted from a fitting procedure, of (d) is 69(6) nK. Images use the same color scheme as that used in figure 2 and values within the insets correspond to the total atom numbers.

that would otherwise affect the efficiency of subsequent evaporative cooling. The ellipticity of the TOP field can also help to counter-balance any asymmetry in the quadrupole field. These unexpected results produced by the optimization procedure, which at first seem counterintuitive, can provide hints to the experimentalist as to where imperfections might exist in the apparatus.

### 4.3. Multiple stage optimization of laser cooling and evaporative cooling

The previous section shows that GP regression is the most rapidly converging of the methods tested in our experimental context. For the remainder of this paper, we therefore focus on this method. We use the GP method to optimize the cMOT stage, approximately doubling the number of laser cooled atoms produced. In figure 5, the Laser cooling panel illustrates the convergence of the four most sensitive $\eta_k$ of this stage as the optimization progresses.

The number of sequences required for the GP method to converge increases with the number of settings. In addition, the computation time scales as the cube of the number of costs over which the GP fits. Given this, it is advantageous to reduce the number of settings as far as is reasonable without compromising the outcome of the optimization. Determining the most sensitive settings allows simultaneous optimization of all cooling stages (laser cooling in the cMOT, evaporative cooling in the quadrupole trap and TOP traps) in a reasonable time. We again utilize the DE method to produce a training set of $2M = 36$ settings/cost pairs. The values of insensitive settings in the laser cooling and evaporative cooling stages were fixed to the best values found during the separate optimization of each stage.

Using the GP learner and by optimizing only the sensitive settings, we are able to produce a BEC from random initial settings after only 12 experimental sequences (following the 36 runs used to build the training set). The optimization produces a BEC with an atom number of $4.5 \times 10^5$, which is greater than atom numbers produced in the optimization of the cooling stages separately. Figures 5(a)–(d), illustrates the improvements in BEC atom number after we have optimized the stages individually and collectively. This faster optimization routine, using only the sensitive settings, can be used to perform quick and regular re-optimization to keep an experimental apparatus tuned up to the best of its capability.

### 4.4. Tailoring the cost function

A maximized atom number in the BEC is often desirable and this motivated our earlier choice of cost function. However, depending on the scenario, other quantities may be of greater importance. For example, when performing alignment of optical elements, it is more useful to maximize the repetition rate of the experiment.

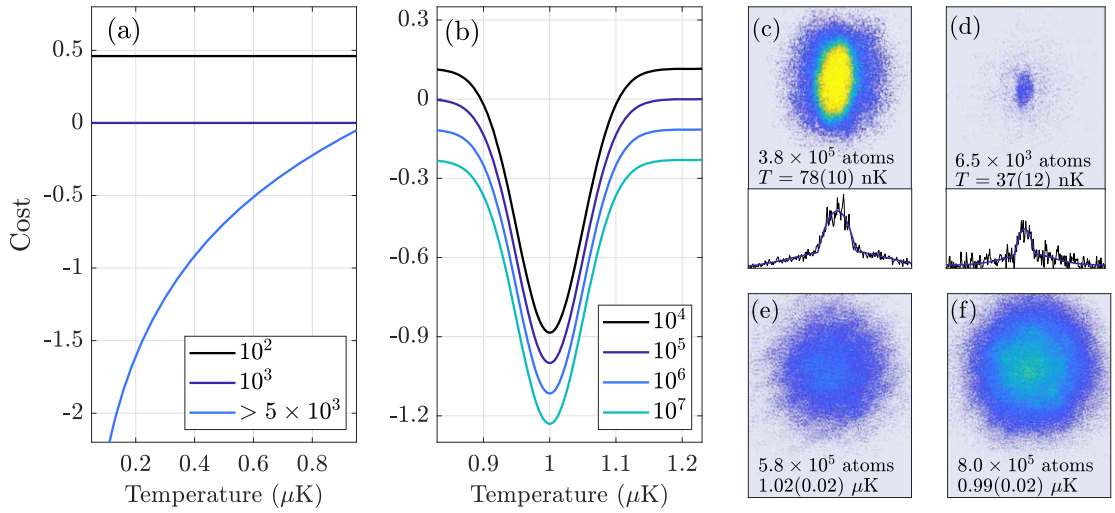


**Figure 6.** Results of optimizations detailed in sections 4.4.2 and 4.4.3. (a), (b) Cost versus atom number and temperature for the cost functions presented in sections 4.4.2 and 4.4.3, respectively. Values in the legend of each plot indicate the total atom number of the cloud. ((c), (d), upper) Absorption images of BECs produced using the optimization routines presented in sections 4.1 and 4.4.2, respectively. ((c), (d), lower) Bimodal fits (purple) to the atomic density profiles (black) extracted from the absorption images above. (e), (f) 1 $\mu$K clouds produced using a truncated version of the evaporative cooling routine presented in section 4.1 and by the optimization presented in 4.4.2, respectively.

*4.4.1. Minimizing sequence duration*

We use our optimization routine to find settings which produce a BEC of a threshold atom number in the shortest possible time. In general, BEC experiments have sequence durations ranging from a few seconds to minutes, depending on the implementation details of each apparatus. We use the cost function

$$f(\tilde{N}) = -(1 + \arctan(\tilde{N} - \tilde{N}_0)/(1 + \tilde{t}), \tag{2}$$

where $\tilde{N}_0$ is a threshold number of atoms within our region-of-interest, which we choose to correspond to an overall BEC size of $1 \times 10^5$ atoms, and $\tilde{t}$ is the sequence duration. This cost function rewards a short sequence time and penalizes settings which do not produce a BEC of a threshold atom number; there is also little reward for producing a BEC with an excess of atoms. With no other changes to the optimization routine, the optimized settings produce a BEC of $9.6 \times 10^4$ atoms and reduce our overall sequence time from 58 s to 46 s, a time saving of over 20%. This demonstrates the power of online optimization to reconfigure an apparatus to achieve the aims of the user.

*4.4.2. Minimizing temperature*

We use our machine learner to find settings which minimize the temperature of the ultracold gas. Temperature cannot be made arbitrarily low, as $N \to 0$ for $T \to 0$ for an evaporative cooling process, so we incorporate a threshold number $N_0$ into the cost function. We define the cost function $f(N, T)$ to be

$$\begin{aligned} &\text{if} \quad N > N_0 \\ &\qquad f(N, T) = \log(T) \\ &\text{else} \\ &\qquad f(N, T) = -0.2 \log(N), \end{aligned} \tag{3}$$

where $T$ is the cloud temperature and $N$ is the total atom number in the atomic cloud. Figure 6(a) illustrates this cost as a function of atom number and temperature. For sufficient atom numbers, the cost depends only on temperature and encourages the learner to reduce $T$. For smaller atom numbers, the fits from which temperature is inferred can fail, and so only the atom number itself is used to determine the cost. We take $N_0 = 5 \times 10^3$ as a lower bound for the number of atoms in the BEC. The prefactor of 0.2 minimizes the discontinuity in the cost either side of the threshold atom number, which assists our gradient-based method in finding the optimal parameter set.

As shown in figure 6(d), optimizing for temperature produces a 37(12) nK cloud of around $6.5 \times 10^3$ atoms. The resulting cloud is significantly colder than that produced using the optimization detailed in section 2.3 (figure 6(c)).

*4.4.3. Maximizing atom number at a specific temperature*

Other optimizations could also be conceived, such as maximizing the atom number of a thermal cloud at a specified temperature. We use the cost function

$$f(N, T) = -\exp(-(T/T_0 - 1)^2) - \log(N), \tag{4}$$

where $T$ is the temperature, $T_0 = 1\ \mu$K is the target temperature and $N$ is the total atom number. This cost function favors the production of a 1 $\mu$K cloud with the greatest atom number, and is illustrated in figure 6(b) as a function of temperature for different atom numbers.

Figure 6(f) shows an image of a thermal cloud produced by settings optimized in this way and, for comparison, figure 6(e) shows a 1 $\mu$K cloud produced by truncating the evaporative cooling ramp from table 1. The 1 $\mu$K cloud produced through this new optimization has an atom number that is 38% larger. Another variant of this scheme could be to minimize $T$ for a thermal gas with a specified atom number, which would be possible with only a minor adjustment to the cost function described above.

## 5. Conclusion

The value of machine learning in finding patterns and optima in data which depends on many parameters is apparent across multiple fields of research [40]. In our specific case, machine learning has provided a means for autonomous experimental optimization. We have compared the convergence rate of three optimization methods. Most notably, for the first time, we have optimized all cooling stages involved in a quantum gas experiment simultaneously. The optimization is quick and achieves our aim of increasing the atom number in a BEC, which is beneficial for improved signal-to-noise ratios when measuring atom numbers in future experiments.

We have used the GP method to identify the sensitive settings within each cooling stage. By restricting the attention of the learner to only consider these sensitive settings, it becomes possible to optimize the experiment as a whole with only a small number of sequences. Optimization can be performed within an hour, allowing daily optimization if necessary to maintain peak performance for producing consistent, high-quality data. Long-term drifts which would otherwise degrade the apparatus' performance can thus be easily mitigated, by scheduling regular optimization routines, e.g. once a week.

Certain features of our optimal solutions are counterintuitive: improvements arising from an elliptical TOP field during the evaporative cooling stage were not expected and would not generally be explored by a researcher. These features may indicate underlying physics, or may allude to the presence of imperfections in the experimental apparatus.

One caveat is that the point of convergence, or optimum, may be one for which the length scale of any parameter is extremely short. While we hope to find the global minimum of the cost function, it is of little experimental value if a perturbation from the prescribed experimental settings leads to a sharp response in the cost. The stability of the solution can be evaluated by assessing the average cost over multiple runs for each input and building separate models for both $E[Y]$ and $\mathrm{Var}[Y]$. These can be jointly optimized to produce a solution which not only works to achieve the user's optimization aim but also reduces shot-to-shot fluctuations which limit the resolution of an experiment. In the interest of short optimization routines, we have decided against this approach. We have observed that the optima found are no less stable than the previous, manually optimized values. Even so, shorter optimization routines can be performed more frequently to counter long-term drifts.

Given the desirability of short optimization routines, and as illustrated by the relative rates of convergence between the methods, we conclude that the GP regression method is of greatest utility in our experimental context. Our optimization routine produces a relatively small amount of training data which, consequently, may reduce the suitability of an ANN-based method, as these typically require many thousands of data points to accurately train the network weights.

## Acknowledgments

The authors would like to thank Henry Howard-Jenkins for useful discussions. This work was supported by the EPSRC Grant Reference EP/S013105/1. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for parts of this research. AJB, KL and DG thank the EPSRC for doctoral training funding. The data that support the findings of this study are available from the corresponding author upon reasonable request.

## Appendix. Further detail on the optimization cost function

In our context, the intention of the evaporative cooling stage is to increase the phase-space density (PSD) [30] of an atomic cloud to the critical value required for Bose–Einstein condensation. As detailed in section 2.1, evaporative cooling is performed by ejecting atoms of higher-than-average energy. The remaining atoms rethermalize to form a colder atomic cloud with a momentum distribution that is more strongly peaked around $k = 0$, where $k$ is atomic momentum, with the onset of a macroscopic occupation of $k \approx 0$ at the critical point for Bose–Einstein condensation. Although the temperature of the atomic cloud is favorably reduced, the mechanism of evaporative cooling reduces the total number of atoms as the stage progresses.

Our optimization objective, as described in section 2.3, is to maximize the number of atoms within a small circular region centered on the atomic cloud after TOF expansion. TOF expansion is a popular method of extracting the momentum distribution of an atomic cloud [30]. The bimodal density distribution for a BEC after TOF expansion is well known [8]: the thermal cloud expands to form a broad Gaussian pedestal, while the BEC forms a parabolic profile in the center. This is illustrated in figure 2. Our counting region captures the number of atoms with momentum $k$ less than a threshold $k_c$, where $k_c$ is set by the radius of the region. To ensure our region includes mostly the BEC component, the radius of the counting region is chosen to be comparable to the Thomas–Fermi radius of a BEC of $10^5$ atoms in a trap with similar parameters to our manual-optimized BEC sequence. Figure 2 illustrates that this region predominantly includes the BEC component of the cloud for both the manually-optimized and machine learning-optimized BECs.

We vary the completion percentage of the evaporative cooling stage and extract the PSD from a bimodal fit. As shown in figure A1, the fitted PSD (purple) rises to and above the critical value for Bose–Einstein condensation. We also show the number of atoms contained within a region much larger than the extent of the BEC (180 $\mu$m radius, blue) and a region of comparable size to the BEC (50 $\mu$m radius, black). We divide the number of atoms contained within the larger region by 8 for easier comparison with the smaller region on the same axis.

The number of atoms contained within the smaller region increases with PSD, as the atomic ensemble re-establishes a momentum distribution more strongly peaked around $k = 0$. Additionally, and as expected, the number of atoms contained within the larger region decreases as the stage progresses and as the total atom

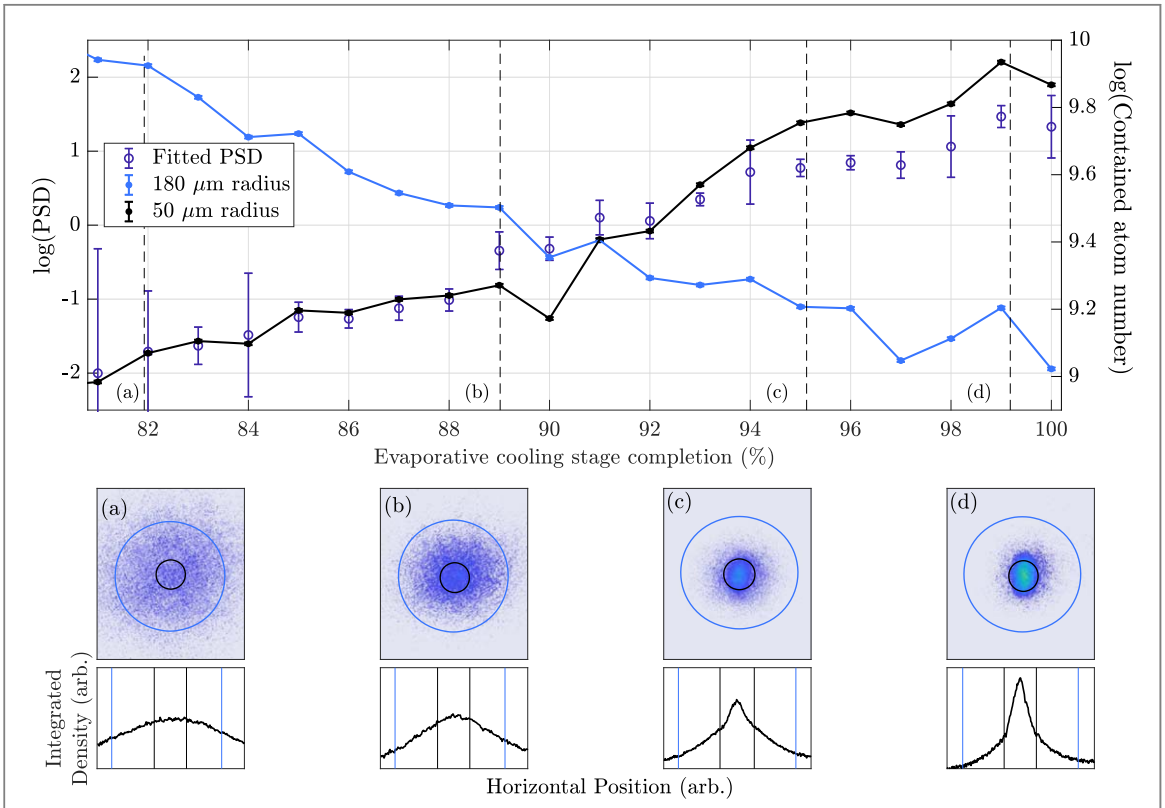


**Figure A1.** Fitted PSD (purple) as a function of evaporative cooling stage progression. Similarly, the number of atoms contained within regions of size 180 $\mu$m (blue) and 50 $\mu$m (black) as a function of evaporative cooling stage progression for the same data. Images ((a)–(d), upper) show atomic density distributions after TOF for 82%, 89%, 95% and 99% stage completion, respectively. The counting regions of radius 180 $\mu$m (blue) and 50 $\mu$m (black) are also illustrated. ((a)–(d), lower) Show the density distribution integrated in the vertical axis plotted against horizontal position. The horizontal extent of the 180 $\mu$m (blue) and 50 $\mu$m counting regions are illustrated by blue and black lines, respectively.

number is reduced. As shown for this data set, extracting PSD from a fitting procedure would provide the equivalent information to a learner as our atom counting method. Nevertheless, when the atomic cloud is faint or not visible, as is the case for earlier stages of the evaporative cooling stage, the fit fails. In contrast, the counting region method relies on fewer parameters and is significantly more robust to small or faint clouds, which are encountered regularly during an optimization routine.

## ORCID iDs

A J Barker https://orcid.org/0000-0003-2574-3081

## References

[1] Patterson J and Gibson A 2015 *Deep Learning : A Practitioner's Approach* 1st edn (Newton, MA: O'Reilly Media Inc.)
[2] Min H 2010 *Int. J. Logist. Res. Appl.* **13** 13–39
[3] Kermany D S 2018 *Cell* **172** 1122–31.e9
[4] Wigley P B *et al* 2016 *Sci. Rep.* **6** 25890
[5] Tranter A D, Slatyer H J, Hush M R, Leung A C, Everett J L, Paul K V, Vernaz-Gris P, Lam P K, Buchler B C and Campbell G T 2018 *Nat. Commun.* **9** 4360
[6] Seif A, Landsman K A, Linke N M, Figgatt C, Monroe C and Hafezi M 2018 *J. Phys. B: At. Mol. Opt. Phys.* **51** 174006
[7] Einstein A 1924 Königliche Preußische Akademie der Wissenschaften. Sitzungsberichte, pp 261–67
[8] Davis K B, Mewes M O, Andrews M R, van Druten N J, Durfee D S, Kurn D M and Ketterle W 1995 *Phys. Rev. Lett.* **75** 3969–73
[9] Anderson M H, Ensher J R, Matthews M R, Wieman C E and Cornell E A 1995 *Science* **269** 198–201
[10] Bloch I, Dalibard J and Nascimbène S 2012 *Nat. Phys.* **8** 267–76
[11] Hadzibabic Z, Krüger P, Cheneau M, Battelier B and Dalibard J 2006 *Nature* **441** 1118–21
[12] Greiner M, Mandel O, Esslinger T, Haensch T W and Bloch I 2002 *Nature* **415** 39–44
[13] Navon N, Gaunt A L, Smith R P and Hadzibabic Z 2016 *Nature* **539** 72–5
[14] Bradley C C, Sackett C A, Tollett J J and Hulet R G 1995 *Phys. Rev. Lett.* **75** 1687–90
[15] Cohen-Tannoudji C, Dupont-Roc J and Grynberg G 1998 *Atom-Photon Interactions* (New York: Wiley) (https://doi.org/10.1002/9783527617197)
[16] Ketterle W and Druten N V 1996 *Adv. At., Mol., Opt. Phys.* **37** 181–236
[17] Toscano J, Wu L Y, Hejduk M and Heazlewood B R 2019 *J. Phys. Chem.* A **123** 5388
[18] Geisel I, Cordes K, Mahnke J, Jöllenbeck S, Ostermann J, Arlt J, Ertmer W and Klempt C 2013 *Appl. Phys. Lett.* **102** 214105
[19] Lausch T, Hohmann M, Kindermann F, Mayer D, Schmidt F and Widera A 2016 *Appl. Phys.* B **122** 112
[20] Rohringer W, Bücker R, Manz S, Betz T, Koller C, Göbel M, Perrin A, Schmiedmayer J and Schumm T 2008 *Appl. Phys. Lett.* **93** 264101
[21] Harte T L, Bentine E, Luksch K, Barker A J, Trypogeorgos D, Yuen B and Foot C J 2018 *Phys. Rev.* A **97** 013616
[22] Hush M R 2019 M-LOOP (https://m-loop.readthedocs.io/en/latest/index.html)
[23] Gildemeister M, Nugent E, Sherlock B E, Kubasik M, Sheard B T and Foot C J 2010 *Phys. Rev.* A **81** 031402
[24] Foot C J 2005 *Atomic Physics* (Oxford: Oxford University Press)
[25] Raab E L, Prentiss M, Cable A, Chu S and Pritchard D E 1987 *Phys. Rev. Lett.* **59** 2631–4
[26] Steck D A 2001 Rubidium 87 D Line Data (http://steck.us/alkalidata) Revision 2.2.1 (Accessed: 21 November 2019)
[27] Sherlock B E, Gildemeister M, Owen E, Nugent E and Foot C J 2011 *Phys. Rev.* A **83** 043408
[28] Sheard B T 2011 Magnetic transport and Bose–Einstein condensation of rubidium atoms *Thesis* (https://ora.ox.ac.uk/objects/uuid:dedece2b-c33a-415b-9d6b-570263042797)
[29] Blundell S and Blundell K M 2010 *Concepts in Thermal Physics* (Oxford: Oxford University Press)
[30] Pethick C J and Smith H 2008 *Bose–Einstein Condensation in Dilute Gases* 2nd edn (Cambridge: Cambridge University Press)
[31] Bentine E 2018 Atomic mixtures in radiofrequency-dressed potentials *Thesis* (https://ora.ox.ac.uk/objects/uuid:b3a77b79-230b-4b61-92f5-1ebf0794f490)
[32] Glover F and Kochenberger G A 2003 *Handbook of Metaheuristics* (New York: Springer)
[33] Jaillet P and Wagner M R 2018 *Online Optimisation* (Berlin: Springer)
[34] Storn R and Price K 1997 *J. Glob. Optim.* **11** 341–59
[35] Seeger M 2004 *Int. J. Neural Syst.* **14** 69–106
[36] Vikhar P A 2016 Evolutionary algorithms: a critical review and its future prospects *2016 ICGTSPICC* (Piscataway, NJ: IEEE) pp 261–5 (http://ieeexplore.ieee.org/document/7955308/)
[37] Rasmussen C E and Williams C K I 2006 *Gaussian Processes for Machine Learning* (Cambridge, MA: MIT Press) (http://gaussianprocess.org/gpml/)
[38] Schmidhuber J 2015 *Neural Netw.* **61** 85–117
[39] Hendrycks D and Gimpel K 2016 arXiv:1606.08415
[40] Kalantre S S, Zwolak J P, Ragole S, Wu X, Zimmerman N M, Stewart M D and Taylor J M 2019 *npj Quantum Inf.* **5** 6
[41] Sheela K G and Deepa S N 2013 *Math. Probl. Eng.* **2013** 1–11
[42] Kingma D P and Ba J 2014 arXiv:1412.6980
[43] Ruder S 2016 arXiv:1609.04747