

# Learning to Do or Learning While Doing: Reinforcement Learning and Bayesian Optimisation for Online Continuous Tuning

Jan Kaiser,<sup>\*</sup> Annika Eichler, Oliver Stein, Willi Kuropka, Hannes Dinter,  
Frank Mayet, Thomas Vinatier, Florian Burkart, and Holger Schlarb  
*Deutsches Elektronen-Synchrotron DESY, Germany*

Chenran Xu,<sup>†</sup> Andrea Santamaria Garcia, and Erik Bründermann  
*Karlsruhe Institute of Technology KIT, Germany*

(Dated: 5 June 2023)

Online tuning of real-world plants is a complex optimisation problem that continues to require manual intervention by experienced human operators. Autonomous tuning is a rapidly expanding field of research, where learning-based methods, such as Reinforcement Learning-trained Optimisation (RLO) and Bayesian optimisation (BO), hold great promise for achieving outstanding plant performance and reducing tuning times. Which algorithm to choose in different scenarios, however, remains an open question. Here we present a comparative study using a routine task in a real particle accelerator as an example, showing that RLO generally outperforms BO, but is not always the best choice. Based on the study's results, we provide a clear set of criteria to guide the choice of algorithm for a given tuning task. These can ease the adoption of learning-based autonomous tuning solutions to the operation of complex real-world plants, ultimately improving the availability and pushing the limits of operability of these facilities, thereby enabling scientific and engineering advancements.

## I. INTRODUCTION

Complex real-world plants are instrumental in facilitating scientific and technological progress. For their successful operation, it is critical that these facilities achieve predefined performance metrics. These are reached through online tuning, i.e. the optimisation of the plant and its subsystems towards a desired system state. Tuning these systems is a challenging optimisation problem due to the non-linear and often dynamic correlations among a large number of tuning parameters. Moreover, the inherent noise in real-world measurements, the time-consuming data acquisition, and the high costs associated with system downtime make the tuning of real-world systems particularly challenging.

To date, online tuning continues to be performed manually, relying on the experience of expert human operators. This leads to suboptimal solutions that are labour intensive to attain and difficult to reproduce.

To reduce downtime and push the limits of their operational capabilities, efforts are made to develop autonomous plant tuning solutions. Existing approaches can improve optimisation results, reproducibility, and reliability for some tuning tasks, but come with their own drawbacks. For example, while grid search and random search are reliable and highly reproducible approaches, they require a large number of samples. As a result, these methods become impractical in the real world, where the cost per sample may be high. Other approaches from the field of numerical optimisation can reduce the number of required samples and have been successfully applied to tuning tasks [1, 2]. While these approaches show promising results, their performance drops as the number of tuning dimensions increases due to the so-called *curse of dimensionality* [3]. Furthermore, many of these methods are sensitive to noise [2], which is omnipresent in real-world measurements.

Learning-based methods have emerged as promising solutions capable of sample-efficient, high-dimensional optimisation under real-world conditions. Bayesian optimisation (BO) [4] is one such learning-based method, that has recently risen in popularity. In BO, the number of samples required for a successful optimisation is reduced by learning a surrogate model of the objective function at the time of optimisation. Another promising approach is *optimiser learning* [5–8], where the function predicting the next sampling point is learned before the application. A powerful instance of optimiser learning is the use of a neural network that is trained via Reinforcement Learning (RL) [7, 8], allowing for the automated discovery of optimisation algorithms. In this paper, we call the resulting optimisation algorithm Reinforcement Learning-trained Optimisation (RLO). As continued optimisation of a dynamic function can be considered to be equivalent to control, we consider RLO to be equivalent to RL-based control. Both RLO and BO are very actively researched and applied to a variety of real-world plants such as particle accelerators [9–22],

---

<sup>\*</sup> jan.kaiser@desy.de

<sup>†</sup> chenran.xu@kit.edu

fusion reactors [23–25], optical and radio telescopes [26–28], chemical reactions [29], additive manufacturing [30], photovoltaic power plants [31], spacecraft [32, 33], airborne wind energy systems [34], telecommunication networks [35] and grid-interactive buildings [36, 37], amongst others. In each of these fields, both RLO and BO have achieved excellent tuning results at high sample efficiency. To the best of our knowledge, however, there is no previous work conducting a detailed comparison of RLO and BO for online continuous optimisation of real-world plants.

In this work, we study RLO and BO for tuning a subsystem of a particle accelerator and compare them in terms of the achieved optimisation result and their convergence speed. In the field of particle accelerators, both methods are gaining notable attention and have led to significant improvements [9–22]. To ensure the reliability of our results, we combine a significant number of simulations with real-world measurements. Based on the results of our study, we ascertain the advantages and disadvantages of each tuning method and identify criteria to guide the choice of algorithm for future applications.

## II. RESULTS

In this study, we consider as a benchmark a recurring beam tuning task which is ubiquitous across linear particle accelerators and frequently performed during start-up and operation mode changes, where the goal is to focus and steer the electron beam on a diagnostic screen. While this task can be very time-consuming, it is also well-defined, making it suitable as a proof-of-concept application for RLO and BO. For the study, we use the specific magnet lattice of a section of the ARES (Accelerator Research Experiment at SINBAD) particle accelerator [38, 39] at DESY in Hamburg, Germany, one of the leading accelerator centres worldwide. From here on, we refer to this section as the *accelerator section*. An illustration of the accelerator section is shown in Fig. 1. Further details on ARES and the accelerator section are given in Section IV A. The lattice of the accelerator section is in downstream order composed of two quadrupole focusing magnets  $Q_1$  and  $Q_2$ , a vertical steering magnet  $C_v$ , a third quadrupole focusing magnet  $Q_3$ , and a horizontal steering magnet  $C_h$ . Downstream of the magnets there is a diagnostic screen capturing a transverse image of the electron beam. A Gaussian distribution  $\mathbf{b} = (\mu_x, \sigma_x, \mu_y, \sigma_y)$  is fitted to the observed image, where  $\mu_{x,y}$  denote the transverse beam positions and  $\sigma_{x,y}$  denote the transverse beam sizes. The goal of the tuning task is to adjust the quadrupole magnets’ field strengths  $k$  and steering magnets’ steering angles  $\alpha$  to achieve a target beam  $\mathbf{b}'$  chosen by a human operator. We denote the *actuators*, here the magnet settings to be changed by the algorithm, as  $\mathbf{u} = (k_{Q_1}, k_{Q_2}, \alpha_{C_v}, k_{Q_3}, \alpha_{C_h})$ . The optimisation problem can be formalised as minimising the objective

$$\min O(\mathbf{u}) = \min D(\mathbf{b}, \mathbf{b}'), \quad (1)$$

which for the benchmark tuning task is defined as the difference  $D$  between the target beam  $\mathbf{b}'$  and the observed beam  $\mathbf{b}$ . The observed beam  $\mathbf{b}$  is determined by the beam dynamics, which depend on the actuators  $\mathbf{u}$ , and environmental factors, such as the magnet misalignments and the incoming beam to the accelerator section. Together with the target beam  $\mathbf{b}'$ , these define the *state* of the environment. With most real-world tuning tasks, not all of the state can be observed, i.e. it is *partially observable*. In the case of the benchmark task, the magnet misalignments and the incoming beam cannot be easily measured or controlled, and are therefore part of the environment’s hidden state. As a measure of difference between the observed beam  $\mathbf{b}$  and the target beam  $\mathbf{b}'$ , we use the mean absolute error (MAE) defined as

$$D_{\text{MAE}}(\mathbf{b}, \mathbf{b}') = \frac{1}{4} \sum_{i=1}^4 \left| \mathbf{b}^{(i)} - \mathbf{b}'^{(i)} \right|, \quad (2)$$

i.e. the mean of the absolute value of the beam parameter differences over all four beam parameters, where  $\mathbf{b}^{(i)}$  denotes the  $i$ -th element of  $\mathbf{b}$ .

For this study, an RLO policy was trained according to previous work [15] and as described in Section IV D. An implementation of BO with a Gaussian process (GP) model [40], detailed in Section IV E, was specially designed for this study. In addition to the studied RLO and BO solutions, we consider random search and Nelder-Mead Simplex optimisation [41] as baselines for randomised and heuristic optimisation algorithms. They are presented in Sections IV F and IV G, respectively.

### A. Simulation study

For the simulation study, we consider a fixed set of 300 randomly generated environment states, each defined by a target beam  $\mathbf{b}'$ , an incoming beam  $I$  entering the accelerator section from upstream, and transverse misalignments of

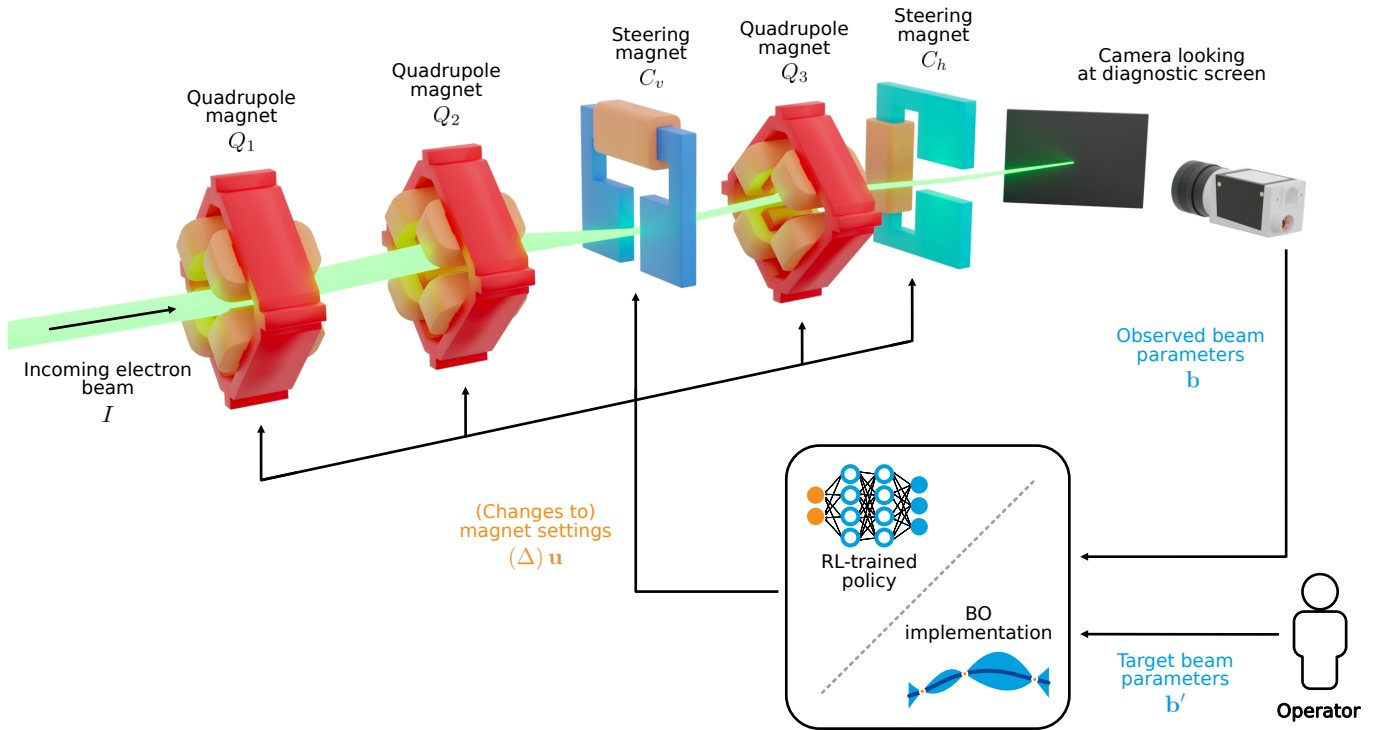


FIG. 1: **Simplified 3D illustration of the considered section of the ARES particle accelerator.** This section consists of three quadrupole magnets and two steering magnets, followed by a diagnostic screen. The measured beam  $b$  and the desired beam  $b'$  are provided to the algorithm performing the tuning. In the case of BO, they are used to compute the objective. In the case of RL, they are provided along with the magnet settings as input to the policy and are used to calculate the reward. Both algorithms output either the next settings to the magnets  $u$  or a change to the magnets  $\Delta u$ .

the quadrupole magnets and the diagnostic screen  $M$ . We refer to these instances of the environment state as *trials*, defined in Eq. (3). The results of the simulation study over RLO and BO, as well as the two baseline algorithms, random search and Nelder-Mead Simplex, are summarised in Table I.

We find that the learning-based algorithms RLO and BO outperform both baselines in terms of the optimisation result, achieving a final beam difference  $D$  at least 6 times smaller. Furthermore, RLO achieves a median final beam difference  $D$  of  $4\ \mu\text{m}$ , which is more than an order of magnitude smaller than the one achieved by BO. The final beam difference achieved by RLO is smaller than the one achieved by BO in 96% of the trials. Note that the final beam difference achieved by RLO is smaller than the measurement accuracy  $\epsilon = 20\ \mu\text{m}$  of the real-world diagnostic screen.

Based on  $\epsilon$ , we construct two metrics to measure the optimisation speed. We define *steps to target* as the number of steps until the observed beam parameters differ less than an average of  $\epsilon$  from the target beam parameters, and *steps to convergence* as the number of steps after which the average of the beam parameters never changes by more than  $\epsilon$ . We observe that RLO always converges and manages to do so close to the target in 88% of trials. BO also converges on almost all trials, but only does so close to the target in 12% of trials, taking about 4 times longer to do so. Figure 2 indicates why: BO explores the optimisation space instead of fully converging toward the target beam. It is possible to suppress this behaviour by using an acquisition function that favours exploitation, but our experiments have shown that such acquisition functions do not perform well with noisy objective functions. If a sample of the objective value was too high as a result of noise, the surrogate model is likely to overestimate the objective near that sample, causing BO to get stuck instead of finding the true optimum. We further observe that RLO converges more smoothly than BO. While this has little effect in simulation, in the real world, smooth convergence has various advantages like limiting wear on the actuators. In the particle accelerator benchmark, smoother convergence limits the effects of magnet hysteresis, an effect where the ferromagnetic core of an electromagnet retains some magnetisation when the current in its coils is removed, reduced, or reversed. As a result of such effects, the objective function may become noisy or even shift, which is why avoiding them through smooth actuator changes generally stabilises the

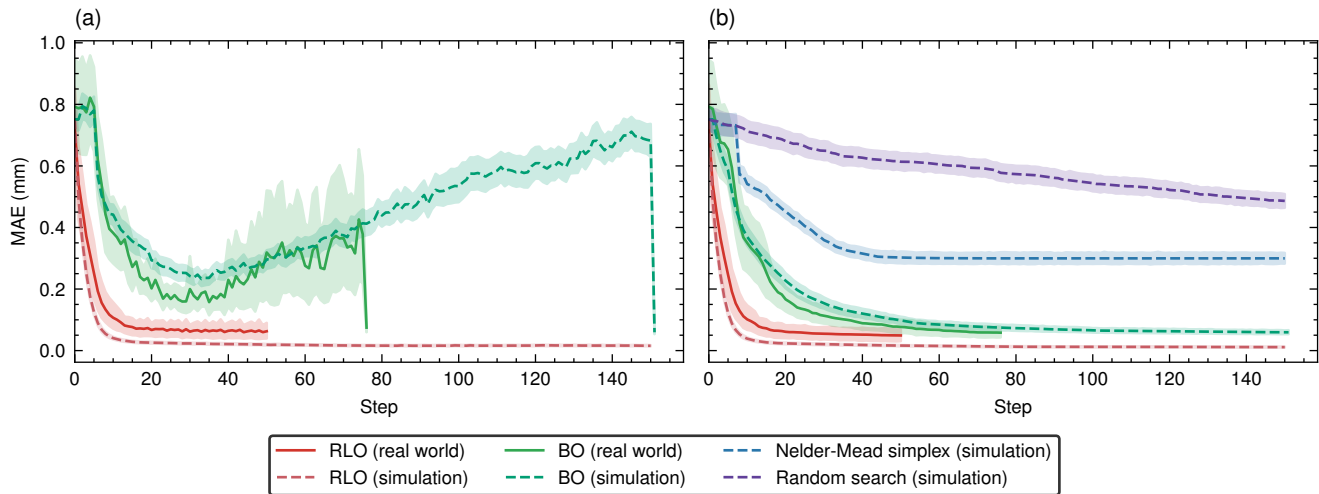


FIG. 2: **Beam difference over time for different optimisation algorithms.** The mean beam difference as the MAE of the beam parameters to the target beam is shown by the solid and dashed lines. The envelopes show the 95 % confidence intervals of the beam differences. **a** shows the beam differences as measured at each step. **b** shows the best beam differences encountered up to each step, i.e. the beam differences that one would return to if the optimisation was terminated in the respective step. Note that on the real plant, this is an estimate, as the beam difference may not be exactly the same for the same set of actuator settings at different times.

objective function and improves reproducibility.

## B. Real-world study

In order to evaluate the methods' ability to transfer to the real world and to verify the results obtained in simulation, we also studied the performance of RLO and BO on the ARES particle accelerator. This part of the study is crucial, as even with accurate simulations, the gap between simulation and the real world is often wide enough that algorithms performing well in simulation cannot be transferred to the real plant [42]. We observed this gap between simulation and experiment in the early stages of training the RL policy, where trained policies performed well in simulation but failed altogether in the real world. Similarly, when implementing BO for the tuning task, implementations tuned for exploitation showed faster and better optimisation in simulation but failed during the experiment under real-world conditions.

Given the limited availability of the real accelerator, we considered 22 trials of the 300 used for the simulation study. The magnet misalignments and the incoming beam on the real accelerator can neither be influenced nor measured during the experiment, so they were considered unknown variables. Before every measurement shift, the incoming beam was aligned with the centres of the quadrupole magnets in order to reduce dipole moments induced when the beam passes through a quadrupole magnet off-centre, which can steer the beam too far off the screen. This adjustment is needed for BO to find an objective signal in a reasonable time. In Section II E, we investigate how the alignment, or the lack thereof, affects the results of this study. The results of the real-world measurements are listed in Table I. Two example optimisations by RLO and BO on the real accelerator are shown in Fig. 3. On the real particle accelerator, just like in the simulation study, we observe that RLO achieves both a better tuning result and faster convergence than BO. This time, RLO outperforms BO on 13 of 22 trials. The gap between the two, however, is not as pronounced in the real world. While all three performance metrics of BO are almost identical between the real world and simulation, the performance of RLO appears to degrade. This is partially due to the measurement accuracy now limiting the achievable beam difference, with the result of RLO being only slightly larger than  $\epsilon$  at  $24.46 \mu\text{m}$ . The degradation of RLO performance may, however, also be an indication that despite the use of domain randomisation the RL policy has slightly overfitted on the simulation. BO does not suffer from this issue as it learns at application time. Note also that to use the available machine study time most effectively, both algorithms were given fewer steps on the real accelerator than in simulation, and that BO was given more steps than RLO.

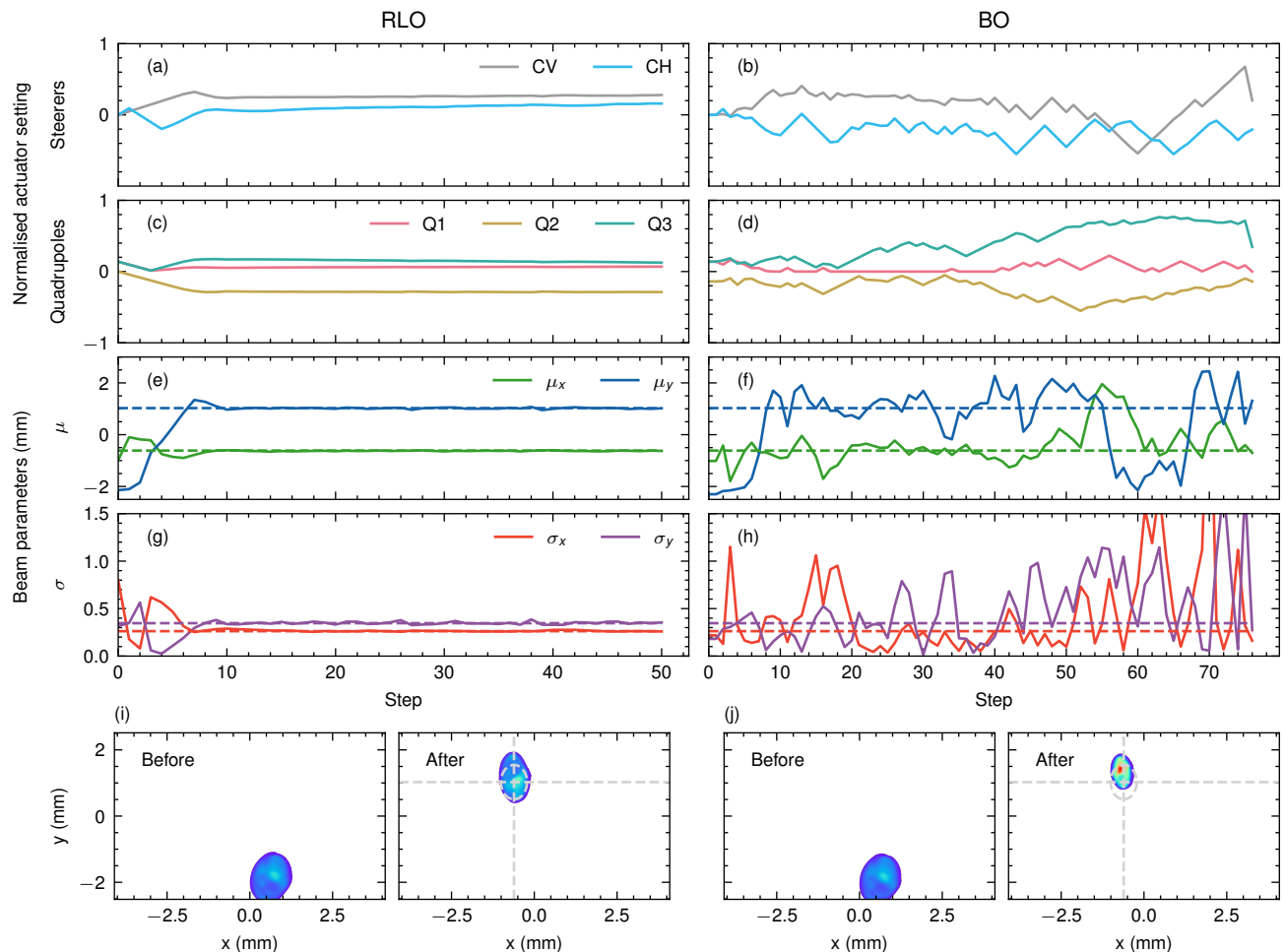


FIG. 3: **Example optimisations on the real particle accelerator.** **a,c,e,g** show one optimisation with RLO. **b,d,f,h** show one optimisation with BO. **a,b** show the steerer settings and **c,d** show the quadrupole magnet settings. **e,f** show the beam positions and **g,h** show the beam sizes. **i,j** show the beam images before and after the optimisation respectively. The target beam size and position are indicated with dashed lines.

### C. Sim2real transfer

The transfer of a method, that works well in a simulation environment, to the real-world is a large part of developing tuning algorithms for facilities such as particle accelerators. The challenges posed by this so-called *sim2real* transfer impact the choice of tuning algorithm.

Successfully transferring a policy trained for RLO to the real ARES accelerator involved a number of engineering decisions detailed in previous work [15] and in Section IV D. While some of the design choices, such as inferring changes to the actuator settings instead of the actuator settings directly, can be applied to other tuning tasks with relative ease, others, such as domain randomisation [43, 44], require specialised engineering for each considered tuning task. Furthermore, all of these require time-consuming fine-tuning to actually achieve a successful zero-shot *sim2real* transfer of a policy trained only in simulation. This is illustrated by the fact that many of the policies trained before the one studied here, performed excellently in simulation while sometimes not working at all on the real ARES accelerator.

On the other hand, BO transfers to the real world with relatively little effort. Once it was sorted out how to best deal with faulty measurements, further discussed in Section II E, most iterations of the BO implementation performed about as well on the real accelerator as they did in simulation. Only some more specialised design decisions, such as tuning the acquisition function strongly towards exploitation, did not transfer as well. The easier *sim2real* transfer of BO is likely owing to the fact that GP model is learned entirely on the real plant and therefore will not overfit to a different objective function that deviates from the one under optimisation.

One issue that may arise when transferring BO or random search from simulation to the real plant is that, while RLO naturally converges toward an optimum and then stays there, meaning that if the optimisation is ended at any time the environment’s state is at, or at least close to, the best-seen position, algorithms like BO and random search are likely to explore further after finding the optimum. It is therefore necessary to return to the best-seen input when the optimisation is terminated. In simulation, this strategy will recover the same objective value, but real-world objective functions are noisy and not always perfectly stationary, e.g. due to slow thermal drifts. As a result, effects such as magnet hysteresis on particle accelerators may shift the objective value when returning to a previously seen point in the optimisation space. In the benchmark tuning task, we experience noisy measurements and magnet hysteresis. We found that for the studied BO trials, the final beam error deviated by a median of  $11\ \mu\text{m}$  and a maximum of  $42\ \mu\text{m}$ . This means that the deviation is usually smaller than the measurement accuracy  $\epsilon$ . At least for the benchmark task, the effect is therefore non-negligible, but also not detrimental to the performance of the tuning algorithms.

#### D. Inference times

The time it takes to infer the next set of actuator settings may also influence the algorithm choice. For the benchmark task, the inference time happens to be negligible, because our benchmarked physical system, specifically the magnets and the beam measurement, is orders of magnitude slower than the inference time. At other facilities, where the physical process takes less time, the time taken for tuning may be dominated by the inference time of the tuning algorithm and there might even be real-time requirements [45].

We measure the average inference times of both algorithms over the 45 000 inferences of the simulation study using a MacBook Pro with an M1 Pro chip running Python 3.9.15. We observe that BO takes an average of 0.7 s to infer the next actuator settings, while RLO is more than three orders of magnitude faster at 0.0002 s. This is because the RLO policy requires only one forward pass of the multilayer perceptron (MLP) with a complexity of  $O(1)$  with respect to the steps taken. By contrast, in each BO inference step, a full optimisation of the acquisition function is performed. This involves inferences with the GP model with complexity  $O(n^3)$ , scaling with the number of steps taken  $n$ . Note that the RLO inference can be significantly sped up by using specialised hardware [46].

#### E. Robustness in the presence of sensor blind spots

In any real system, it is possible to encounter states where the available diagnostics deliver false or inaccurate readings, causing erroneous objective values and observations. Transitions to these states can be caused by external factors as well as the tuning algorithm itself. A good tuning algorithm should therefore be able to recover from these states. In the benchmark tuning task, an erroneous measurement occurs when the electron beam is not visible on the diagnostic screen within the camera’s field of view. In this case, the beam parameters computed from the diagnostic screen image are false, also resulting in a faulty objective value.

We observed that when the beam is not properly observed, RLO can usually recover it in just a few steps. Presumably, the policy can leverage its experience from training to make an educated guess on the beam’s position based on the magnet settings even though faulty beam measurements were not part of its training, where RLO always had access to the correct beam parameters.

In contrast, BO struggles to recover the beam when it is off-screen, as the GP model is learned at application time from faulty observations, resulting in faulty predictions of the objective and acquisition functions. When defining the task’s objective function as only a difference measure from the current to the target beam, falsely good objective values are predicted in the blind spot region of the actuator space and BO converges towards their locations. Our implementation, as described in Section IV E, alleviates this issue by introducing a constant punishment to the objective function when no beam is detected in the camera’s field of view. Nevertheless, the lack of information about the objective function’s topology results in BO taking many arbitrary steps before the beam is by chance detected and the optimisation starts progressing towards the target beam. While more comprehensive diagnostics can help solve this problem, these are often not available.

Because of BO’s insufficient ability to recover from a system state in which there is no informative objective signal, the presented measurements on the real accelerator were taken with the beam aligned to the quadrupole magnets. As a result, the additional dipole moments induced by the quadrupole magnets when increasing the magnitude of their focusing strength are kept minimal, reducing the chance that the beam leaves the camera’s field of view during the initial step of the optimisation. As this alignment would not be performed during nominal operation but may change the observed performance of both algorithms, a study was performed in simulation in order to understand how to interpret the reported results given that the beam was aligned to the centres of the quadrupole magnets before

the optimisation. Both algorithms are evaluated over the same 300 trials as in Section II A. Unlike in the original simulation study, we also simulate erroneous beam parameter measurements when the beam position is detected outside the camera’s field of view. Both algorithms are tested once with the original incoming beam and once with an incoming beam that was previously aligned to the quadrupole magnets. The results are reported in Table I. We conclude that the reported results on the real particle accelerator would be expected to worsen by about 5 % to 33 % for RLO and by 12 % to 121 % for BO if the electron beam had not been aligned to the centres of the quadrupole magnets at the beginning of measurement shifts. This does not change how both algorithms compare to each other.

### F. Failure modes

With tuning algorithms that are intended to be deployed without supervision to enable the autonomous operation of complex plants, it is important to understand how they might fail. We observe that over the entirety of this study, neither RLO nor BO ever produced a final beam that was worse than the beam before the optimisation. Instead, both algorithms clearly improve the beam in most trials, with only a few trials being outliers where the objective was only slightly improved. It was not possible to identify for either RLO or BO a cause for these outliers. Most likely, they are stochastic in nature, owing to the stochastic components of either algorithm. That is, the RLO policy presumably did not gain enough experience in some regions of the state space because they were not explored as much during training. Similarly, BO may be at a disadvantage when the randomly chosen initial samples are unfavourable. We performed grid scans over target beams for both algorithms in simulation to confirm this through the presence of outliers in random locations of the target beam space. They further show that both algorithms perform worse when tasked with tuning towards large than when tasked with tuning towards a small beam, though this effect is subtle compared to the outliers. The root cause of this observation is presumably the initialisation of the magnets in an focus-defocus-focus (FDF) pattern at the beginning of each optimisation with both RLO and BO. While creating a performance deficit for certain beams compared to others, this initialisation improves the overall performance of both algorithms.

There are two further failure modes that should be discussed for both algorithms. RLO can in rare cases enter an unstable state, in which the policy outputs oscillating actuator settings. These result in the beam parameters oscillating around the target. The cause of these oscillations is yet unknown. We note that the oscillations are also produced by policies trained with different random seeds. BO may fail seriously when the beam is far away instead of just slightly off the screen before the start of the optimisation. In such a case, it can take a long time before the beam is randomly moved into the visible area of the diagnostic screen.

### G. Running as a feedback

Real-world plants may be subject to drifts caused by unmodelled external factors. Moreover, control can be regarded as the continuous optimisation of a dynamic objective function. Consequently, a tuning algorithm that can run as feedback on a dynamic objective function can be used for drift compensation and control in addition to tuning. Thus, a tuning method’s ability to operate as a feedback is an interesting subject of further investigation and could impact algorithm selection.

While BO assumes a static objective function, the benchmarked RLO policy does not rely on memorising previously seen objective values or could alternatively learn to adapt to dynamic objective functions during training. It should therefore be possible to use the policy from RLO as an RL-based feedback controller for a dynamic system. To test this, we ran an optimisation with both methods for 80 steps. After 40 steps, we introduce an instant step-to-step change to the incoming beam, changing the latter such that a different set of actuator settings is required to achieve the same beam on the diagnostic screen. We then observe how the RL policy and our BO implementation react to the upstream change. If the method manages to recover the machine state, it can be considered capable of running as feedback. As can be seen in Table II and Fig. 4, the RL-based controller can in fact recover the target beam in about the same time it took to perform the original optimisation, with the final beam difference being comparable to the one achieved in optimisation with a static incoming beam. The beam achieved by BO when the beam instantly changes during the optimisation is, as expected, significantly worse than it is with a constant beam. After the incoming beam changed, the GP model based on the previous 40 samples is no longer correct, effectively breaking BO.

However, the system changes that feedbacks need to react to are not always fast. Often, they occur slowly over time, such that the controller must track the change in order to hold the system near the desired state after attaining it. We therefore also evaluate the RLO policy as a controller and BO in a setup where the incoming beam changes linearly over the course of 80 steps. The results are listed in Table II. We can see in Fig. 4 that the RL policy is capable of tracking the target beam parameters after attaining them. The reasonably small increase in final MAE can primarily

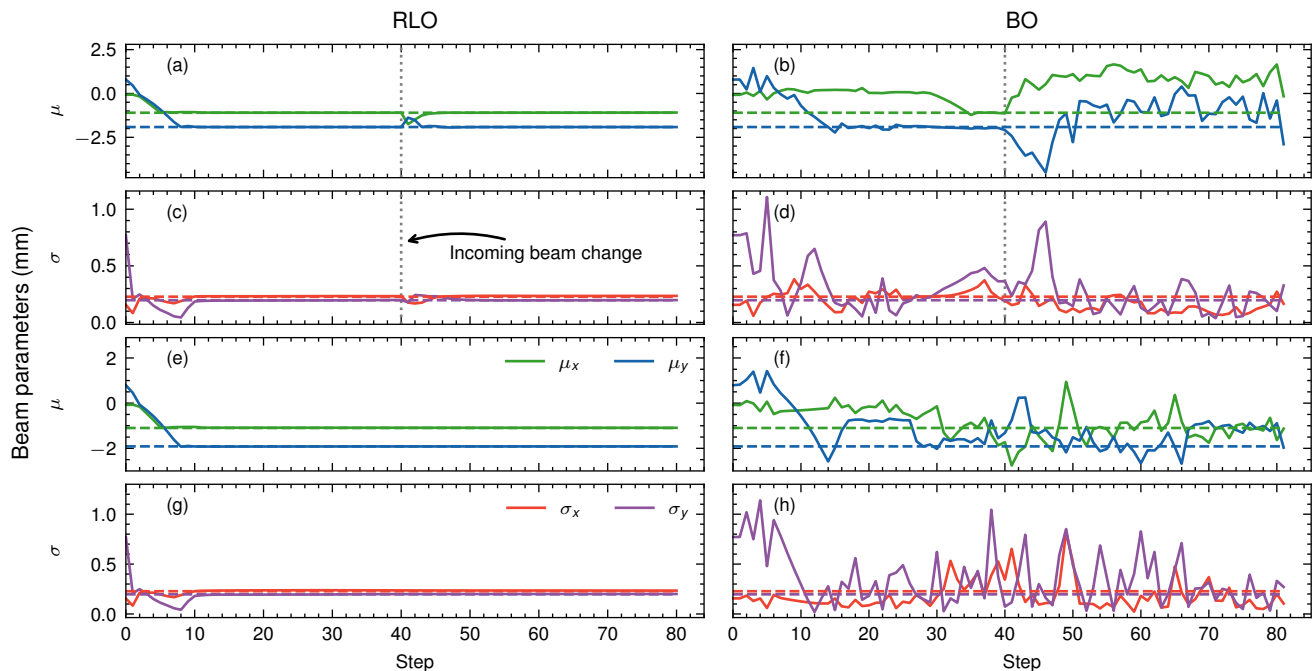


FIG. 4: **RLO and BO optimisers running as feedbacks in simulation.** **a-d**, RLO and BO reacting to an instant change of the incoming beam at step 40, denoted by the vertical dotted lines. **e-h**, the optimisers tracking the optimum with respect to a continuously changing incoming beam. **a,b,e,f** show the evolution of the beam positions  $\{\mu_x, \mu_y\}$ , and **c,d,g,h** show the beam sizes  $\{\sigma_x, \sigma_y\}$ . The horizontal dashed lines denote the target beam parameters respectively.

be explained by the fact that the policy requires a few steps to converge on the desired beam parameters, but in the final step only a single step has passed since the last change to the incoming beam, therefore giving the policy only very little time to correct for the change. As with the instant incoming beam change, BO is not capable of tracking the desired beam parameters. As the incoming beam cannot be included in the GP-model, the learned surrogate is ill-defined, tracking a dynamically changing objective function with a static model. As a result BO optimises an objective function that diverges from the true objective function of the system.

It needs to be mentioned that the slow drifts of the underlying objective function, like the temperature drift of the magnets, can be tackled by adaptive BO with a spatiotemporal GP model [47] or contextual BO [48]. This would require, however, problem-specific implementation and additional engineering effort.

## H. Robustness to actuator failure

In real-world plants, one also has to deal with the potential failure of components such as the actuators used for tuning. It would therefore be beneficial if a tuning algorithm could handle such an actuator failure and recovers the previous state.

We evaluate RLO's and BO's ability to handle both a permanent actuator failure, where the actuator has failed some time before the tuning algorithm was started, and a delayed actuator failure, where the actuator is operational when the tuning starts but fails at a later time during the tuning. Specifically, we simulate the failure of the third quadrupole magnet in the benchmark task, assuming that the magnet's power supply has failed and its quadrupole strength is permanently set to  $0 \text{ m}^{-2}$  after the initial failure. We assume that a failed actuator provides a correct readback to the optimisation algorithm. Table II lists the results. We observe that RLO handles actuator failure well, despite never being trained to do so. When the magnet has failed before the start of the optimisation, RLO finds an optimum, that is almost as good as it would be without the magnet failure, without using the failed magnet. RLO can recover the beam's state when the magnet fails during the optimisation. An example of RLO reacting to an actuator failure during tuning is shown in Fig. 5. BO even improves in performance, as a failed magnet reduces the dimensions of the search space, but despite this, it performs worse than RLO.



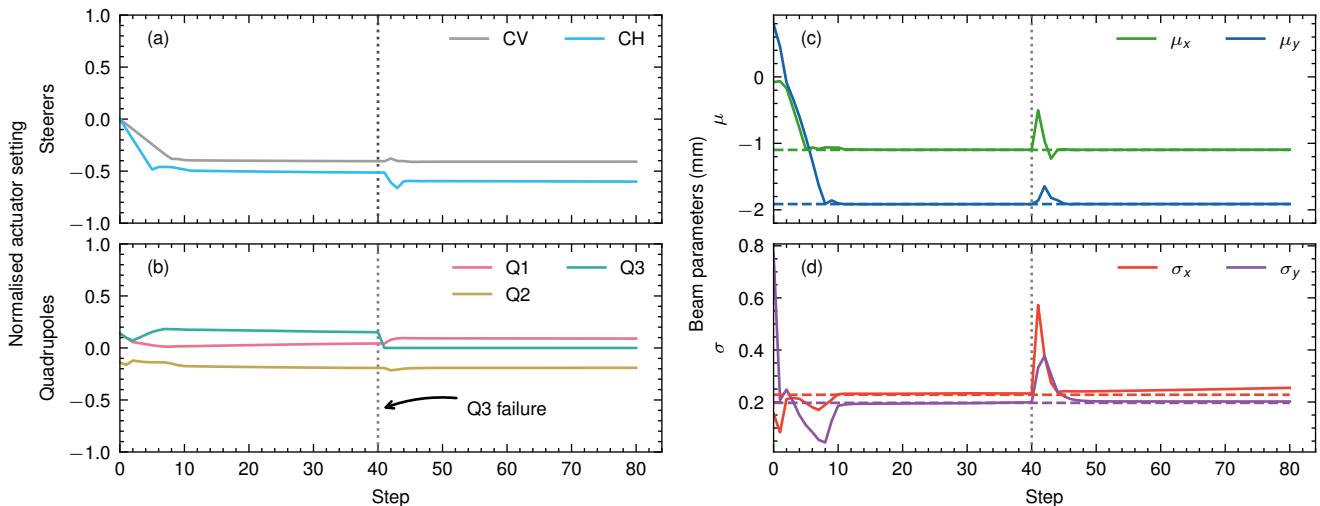


FIG. 5: **RLO reacting to an simulated actuator failure during the optimisation.** The third quadrupole magnet fails in step 40, denoted by the vertical dotted lines. **a** shows the normalised steerer settings and **b** showed the normalised quadrupole strengths, where, when Q3 fails, the strength of the other focusing quadrupole magnet Q2 is quickly increased and the horizontal steering magnet is used to counter the increased change beam position as a result of the changing dipole moments induced by the quadrupole magnets. **c** shows the beam positions  $\{\mu_x, \mu_y\}$  and **d** shows the beam sizes  $\{\sigma_x, \sigma_y\}$ . The dashed lines denote the target beam parameters respectively.

### III. DISCUSSION

The results of our study show that both learning-based optimisation algorithms RLO and BO clearly outperform the baseline methods Nelder-Mead Simplex optimisation and random search. Furthermore, the results indicate that in most cases, RLO is the superior learning-based optimisation method, thanks to its ability to utilise experience acquired before the application time. Nevertheless, BO proves to be a promising alternative for online continuous tuning of complex real-world plants due to its versatility as a black-box optimisation method. In Fig. 6 we illustrate, how both learning-based algorithms relate to each other and the two investigated baselines in terms of different design aspects.

RLO primarily outperforms BO in that it is capable of converging towards the desired plant state faster than BO and closer to the desired state. Furthermore, RLO was found to be more capable of dealing with many of the challenges encountered when working with real-world plants. When presented with false sensor readings, RLO recovers faster than BO. Furthermore, RLO does not continue exploring once the optimum is found, eliminating the problems associated with recovering previously seen objective values in real-world systems. In addition, a trained RLO agent requires no setting of hyperparameters or similar at application time and can therefore be used as a one-click solution by anyone without requiring RL expertise from the user and promising reproducible results. This is in contrast to BO, which is likely to need small hyperparameter adjustments for different instances of the same tuning tasks, thus requiring that a user brings at least some understanding of the chosen BO implementation and its hyperparameters. While not the main focus of this study, policies trained via RL as optimisers may also be used without retraining as controllers, being able to both reach the optimum in a static system and track the optimum in a dynamic system.

The main advantage of BO is the relatively small engineering effort required to deploy it successfully. BO algorithms that adapt hyperparameters automatically during the actual optimisation can be implemented easily and require relatively little hyperparameter tuning between different tuning tasks. In contrast, RLO requires substantial engineering efforts by both RL and domain experts, who must develop a suitable training setup and overcome the sim2real transfer problem. In addition, we observed that both RLO and BO can deal with unexpected situations like actuator failures, thus being robust tuning methods for real-world applications.

The choice of tuning algorithm depends primarily on how much and how often a tuning algorithm is going to be used, and whether the final tuning result and the time saved by a fast tuning algorithm are worth the associated engineering effort. We find that RLO is the overall more capable and faster optimiser, but requires significant upfront engineering. It is therefore better suited to regularly performed tasks, where better tuning results and faster tuning justify the initial investment. For tasks that are only performed a few times, for example on rare occasions during operation or during the commissioning of a system, the engineering effort associated with RLO may not be justified.

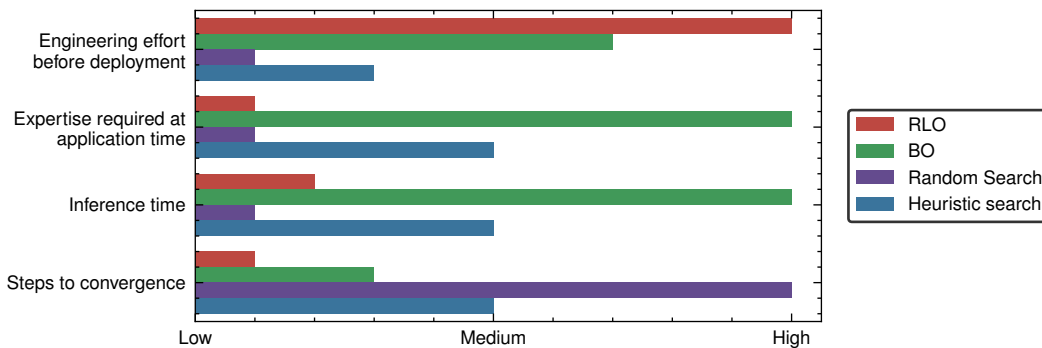


FIG. 6: **Design space for large-scale facility tuning algorithms.** Shows qualitative metrics of comparison for all algorithms considered in our study relative to each other and may aid the decision-making process for choosing one of these algorithms based on criteria specific to the desired application.

Our study has shown that BO, despite not performing as well as RLO, is a valid alternative in such cases.

## IV. METHODS

To collect the data presented in this study, evaluation runs of RLO and BO as well as the baseline methods of Nelder-Mead Simplex and random search were run in simulation and on a real particle accelerator. The following sections introduce the real-world plant used for our study, our experimental setups, and the optimisation algorithms.

### A. ARES particle accelerator section

The ARES (Accelerator Research Experiment at SINBAD) particle accelerator [38, 39], located at Deutsches Elektronen-Synchrotron DESY in Hamburg, Germany, is an S-band radio frequency linac that features a photoinjector and two independently driven travelling wave accelerating structures. These structures can operate at energies up to 154 MeV. The primary research focus of ARES is to produce and study sub-femtosecond electron bunches at relativistic energies. The ability to generate such short bunches is of great interest for applications such as radiation generation by free electron lasers. ARES is also used for accelerator component research and development as well as medical applications.

The accelerator section, known as the *Experimental Area*, is a subsection of ARES, shown in Fig. 7 and made up of two quadrupole magnets, followed by a vertical steering magnet that is followed by another quadrupole magnet and a horizontal steering magnet. Downstream of the five magnets, there is a scintillating diagnostic screen observed by a camera. The power supplies of all magnets can be switched in polarity. The quadrupole magnets can be actuated up to a field strength of  $72 \text{ m}^{-2}$ . The limit of the steering magnets is  $6.2 \text{ mrad}$ . The camera observes an area of about  $8 \text{ mm}$  by  $5 \text{ mm}$  at a resolution of  $2448$  by  $2040$  pixels. The effective resolution of the scintillating screen is ca.  $20 \mu\text{m}$ .

At the downstream end of the section, there is an experimental chamber. This section is regularly used to tune the beam to the specifications required in the experimental chamber or further downstream in the ARES accelerator.

### B. Simulation evaluation setup

In the simulation, a fixed set of 300 randomly generated trials were used to compare the different optimisation algorithms. Each trial is a tuple

$$(\mathbf{b}', M, I) \quad (3)$$

of the target beam  $\mathbf{b}$  that we wish to observe on the diagnostic screen, the misalignments of the quadrupole magnets and the diagnostic screen  $M$ , as well as the incoming beam  $I$  entering the accelerator section. The target beam was generated in a range of  $\pm 2 \text{ mm}$  for  $\mu_x$  and  $\mu_y$ , and  $0 \text{ mm}$  to  $2 \text{ mm}$  for  $\sigma_x$  and  $\sigma_y$ . These ranges were chosen to cover a wide range of measurable target beam parameters, which are constrained by the dimensions of the diagnostic screen.

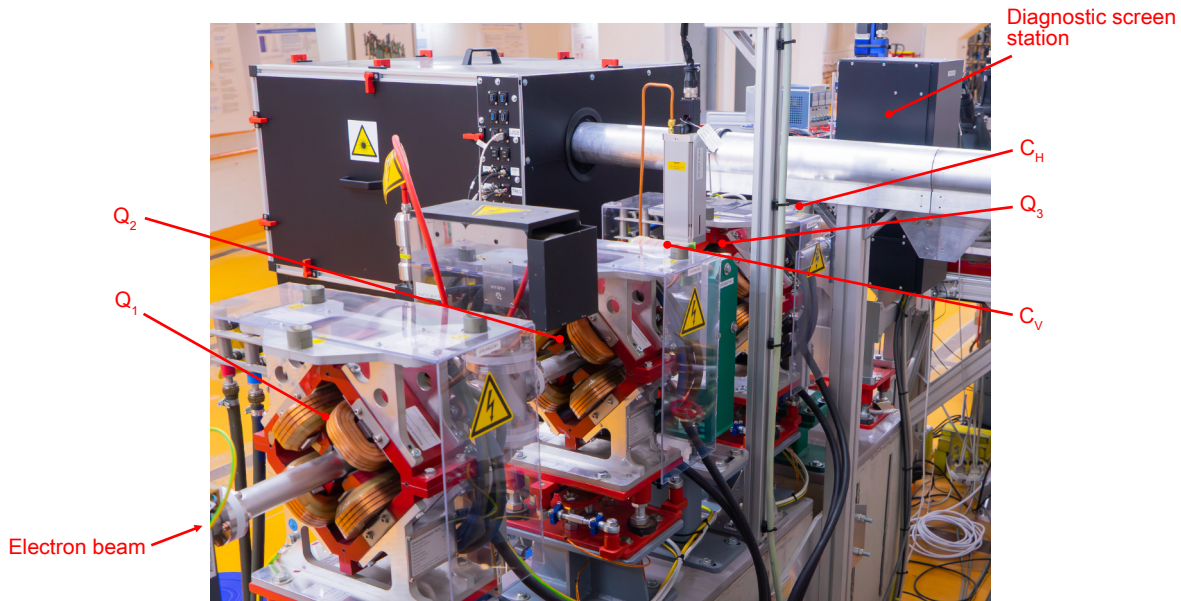


FIG. 7: **Considered accelerator section at ARES.** The electron beam travels downstream from left to right. The five magnets actuated by the optimisation algorithms and the diagnostic screen station are marked.

The incoming beam  $I$  is randomly generated to represent samples from the actual operating range of the real-world accelerator. Both incoming beam and misalignment ranges were chosen to be larger than their estimated ranges present in the real machine.

### C. Real-world evaluation setup

In the real world, the overall machine state was set to an arbitrary normal machine state, usually by leaving it as it was left from previous experiments. This should give a good spread over reasonable working points. The target beams were taken from the trial set used for the simulation study. As the incoming beam and misalignments cannot be influenced in the real world in the same way they can be in simulation, they are left as they are on the real accelerator and considered unknown. Experiments on the real accelerator were conducted on 9 different days over the course of 82 days, running at charges between 2.6 pC and 29.9 pC, and an energy of 154 MeV. To ensure a fair comparison of the tuning methods, we align the beam to the quadrupole magnets at the beginning of each measurement day. This ensures that the beam remains within the camera’s field of view on the diagnostic screen in the initial step, which is also a common operating condition of the accelerator. This reduces the dipole moments produced when increasing the strength of the quadrupole magnets and therefore reduces the likelihood of the beam being steered past the camera’s field of view on the diagnostic screen in the very first step when BO changes the quadrupole strengths. The alignment is not necessarily needed for the RLO as it can recover the beam back into the diagnostic screen camera’s field of view despite receiving erroneous observations.

Transferability of the experiments between simulation and real world as well as RLO and black-box optimisation was achieved through a combination of OpenAI Gym [49] environments an overview of which is shown in Fig. 8. Two different environments were created based on a common parent environment defining the logic of the beam tuning task. One wraps around the *Cheetah* simulation code [50], allowing for fast training and evaluation. The other environment interfaces with the accelerator’s control system. Crucially, both environments present the same interface, meaning that any solution can easily be transferred between the two. While Gym environments are primarily designed for RL policies to interact with their task, the ones used for this work were made configurable in such a way that they can also be interfaced with a BO optimisation. This includes configurable reward formulations and action types that pass actuator settings to the `step` method and have the latter return an objective value via the `reward` field of the `step` method’s return tuple.

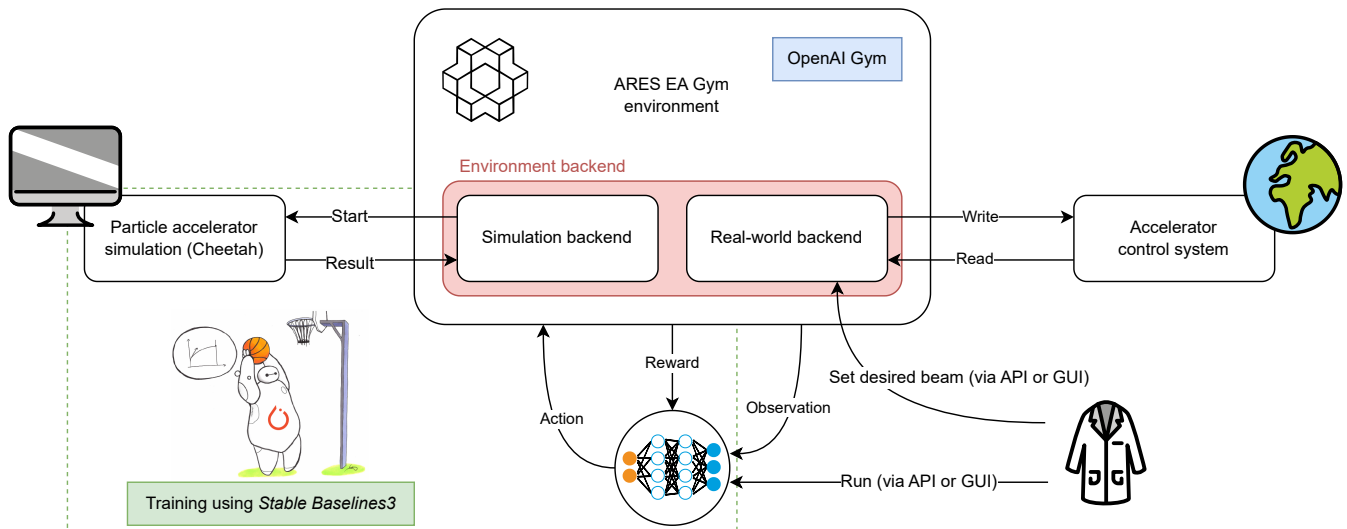


FIG. 8: **Gym environment setup used for the study.** In particular showing how one environment interface facilitates design and training using a simulation of the plant, as well as transferring a developed tuning algorithm to the real plant without modification.

#### D. Reinforcement learning

The RLO implementation used for this study has been introduced in previous work [15]. In this case, an MLP with two hidden layers of 64 nodes each is used as a policy, observing as input the currently measured beam parameters on the diagnostic screen  $\mathbf{b} = (\mu_x, \sigma_x, \mu_y, \sigma_y)$ , the currently set field strengths and deflection angles of the magnets  $\mathbf{u} = (k_{Q_1}, k_{Q_2}, \alpha_{C_v}, k_{Q_3}, \alpha_{C_h})$  and the desired beam parameters  $\mathbf{b}' = (\mu_x, \sigma_x, \mu_y, \sigma_y)$  set by the human operator. The policy then outputs changes to the magnet settings  $\mathbf{a}_t = \Delta\mathbf{u}$ . A normalisation of rewards and observations using a running mean and standard deviation is performed over the training. The outputs are normalised to 0.1 times the magnet ranges of  $\pm 30 \text{ m}^{-2}$  for the quadrupole magnets and  $\pm 2 \text{ mrad}$  for the steering magnets. During training and application, optimisations are started from a fixed FDF setting of the quadrupole triplet, with the strengths  $(k_{Q_1}, k_{Q_2}, k_{Q_3}) = (10, -10, 10) \text{ m}^{-2}$  and both steering magnets set to 0 mrad. The policy is trained for 6 000 000 steps using the Twin Delayed DDPG (TD3) [51] algorithm as implemented by the *Stable Baselines3* [52] package. Training is run in a simulation provided by the *Cheetah* [50] particle tracking code, as limited availability makes training on the real particle accelerator infeasible. Domain randomisation [43] is performed during training. Specifically, the magnet and screen misalignments, the incoming beam and the target beam are randomly sampled from a uniform distribution for each episode. The reward function used for training is

$$R(\mathbf{s}_t, \mathbf{a}_t) = \begin{cases} \hat{R}(\mathbf{s}_t, \mathbf{a}_t) & \text{if } \hat{R}(\mathbf{s}_t, \mathbf{a}_t) > 0 \\ 2 \cdot \hat{R}(\mathbf{s}_t, \mathbf{a}_t) & \text{otherwise} \end{cases} \quad (4)$$

with  $\hat{R}(\mathbf{s}_t, \mathbf{a}_t) = O(\mathbf{u}_t) - O(\mathbf{u}_{t+1})$  and  $O(\mathbf{u}_t)$  being the natural logarithm of the weighted MAE between observed and target beam on the diagnostic screen. The trained policy is deployed zero-shot, i.e. without any further training or fine tuning, to the real world.

#### E. Bayesian optimisation

The BO version used for this study is a custom implementation using the *BoTorch* [53] package. The objective  $O(\mathbf{u})$  to be optimised is defined as

$$O(\mathbf{u}) = -\log(\text{MAE}(\mathbf{b}, \mathbf{b}')) + w_{\text{on-screen}}. \quad (5)$$

The logarithm is used to properly weigh the fine improvement when BO approaches the target beam. A further on-screen reward  $w_{\text{on-screen}} = 10$  is added to the objective when the beam can be observed on the screen, and subtracted

from the objective to penalise the settings when the beam is off the diagnostic screen. To increase the numerical stability of the GP regression, the previous input settings  $\mathbf{u}$  are normalised to  $[-1, 1]$ , projecting the maximum to 1 and the minimum to  $-1$ , and objective values are standardised. The covariance function of the GP models used in this study is the sum of a Matérn-5/2 kernel [54] and a white noise function. The GP hyperparameters, like the length scales and signal noise, are determined dynamically by log-likelihood fits in each step. In each trial, BO is started from the same fixed FDF setting used by RLO. Five random samples are taken to initialize the GP model. Based on the posterior prediction of the GP model, an expected improvement (EI) [55] acquisition function is calculated, which automatically balances the exploration and exploitation of the objective. The next sample is chosen by maximising the acquisition function, where the maximum step sizes are constrained to 0.1 times the total action space. Additionally, the quadrupole magnets are only allowed to vary unidirectionally, i.e. in the FDF setting, so that the time-consuming polarity changes of the quadrupole magnets' power supplies due to the exploration behaviour of BO can be avoided. BO is allowed to run 150 steps in simulation and 75 steps on the real machine, after which we return to the best settings found.

Note that this designed mostly using a simulation before deploying it to the real accelerator. This was done in an effort to reduce the amount of beam time needed for development.

## F. Nelder-Mead simplex

The Nelder-Mead Simplex optimisation [41] was implemented using the *SciPy* [56] Python package. The initial simplex was tuned in a random search of 405 samples to the one that performed best across the set of 300 trials. Nelder-Mead is allowed to run for a maximum of 150 steps. After 150 steps or after early termination the simplex might perform better if it returns to the final sample, but as it is generally converging, it does not necessarily need to. The objective function optimised by Nelder-Mead is the MAE of the measured beam parameters to the target beam parameters.

## G. Random search

For the random search baseline, we sample random magnet settings from the constrained space of magnet settings. Constraint in this case means that we limit the space to a range commonly used during operations, instead of the full physical limits of the magnets. The latter limits are almost an order of magnitude larger than anything ever used in operation. At the end of the optimisation, we return to the best example found.

## DATA AVAILABILITY

The data generated for the presented study is available at <https://doi.org/10.5281/zenodo.7853721>.

## CODE AVAILABILITY

The code used to conduct and evaluate the presented study is available at <https://github.com/desy-ml/r1-vs-bo>.

## ACKNOWLEDGEMENTS

This work has in part been funded by the IVF project InternLabs-0011 (HIR3X) and the Initiative and Networking Fund by the Helmholtz Association (Autonomous Accelerator, ZT-I-PF-5-6). All figures and pictures by the authors are published under a CC-BY7 license. The authors thank Sonja Jaster-Merz and Max Kellermeier of the ARES team for their great support during shifts as well as always insightful brainstorming. In addition, the authors acknowledge support from DESY (Hamburg, Germany) and KIT (Karlsruhe, Germany), members of the Helmholtz Association HGF, as well as support through the *Maxwell* computational resources operated at DESY and the *bwHPC* at SCC, KIT.

## AUTHOR CONTRIBUTIONS

J.K., C.X., A.S.G., A.E. and E.B developed the concept of the study. A.E., E.B and H.S. secured funding. J.K. developed and trained the RL agent with support from O.S. C.X. designed the implementation of BO. J.K. ran the simulated evaluation trials and took the real-world data. J.K. evaluated the measured data. C.X. provided substantial input to the evaluation. A.E. and A.S.G. provided input on the evaluation of the measured data. W.K., H.D., F.M. and T.V. assisted the data collection as ARES operators. F.B. assisted the data collection as ARES machine coordinator. W.K., H.D., F.M., T.V. and F.B. contributed their knowledge of the machine to the implementation of both methods. J.K. wrote the manuscript. C.X. provided substantial edits to the manuscript. J.K. created the presented figures with input from C.X., O.S. and F.M. All authors discussed the results and provided edits and feedback on the manuscript.

## COMPETING INTERESTS

The authors declare no competing interests.

- 
- [1] Bergan, W. F. *et al.* Online storage ring optimization using dimension-reduction and genetic algorithms. *Physical Review Accelerators and Beams* **22**, 054601 (2019). URL <https://link.aps.org/doi/10.1103/PhysRevAccelBeams.22.054601>.
- [2] Huang, X., Corbett, J., Safranek, J. & Wu, J. An algorithm for online optimization of accelerators. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **726**, 77–83 (2013). URL <https://www.sciencedirect.com/science/article/pii/S0168900213006347>.
- [3] Bellman, R. *Dynamic Programming* (Princeton University Press, 1957).
- [4] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P. & de Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE* **104**, 148–175 (2016).
- [5] Andrychowicz, M. *et al.* Learning to learn by gradient descent by gradient descent. In *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016)* (2016).
- [6] Chen, T. *et al.* Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research* **23**, 1–59 (2022). URL <http://jmlr.org/papers/v23/21-0308.html>.
- [7] Li, K. & Malik, J. Learning to optimize. In *International Conference on Learning Representations* (2017). URL <https://openreview.net/forum?id=ry4Vrt5gl>.
- [8] Li, K. & Malik, J. Learning to optimize neural nets (2017). Preprint available at <http://arxiv.org/abs/1703.00441>.
- [9] Boltz, T. *et al.* Feedback design for control of the micro-bunching instability based on reinforcement learning. In *CERN Yellow Reports: Conference Proceedings*, vol. 9, 227–227 (2020).
- [10] Bruchon, N. *et al.* Basic Reinforcement Learning Techniques to Control the Intensity of a Seeded Free-Electron Laser. *Electronics* **9** (2020).
- [11] Duris, J. *et al.* Bayesian Optimization of a Free-Electron Laser. *Physical Review Letters* **124** (2020).
- [12] Hanuka, A. *et al.* Online tuning and light source control using a physics-informed Gaussian process. In *Proceedings of the 33rd Conference on Neural Information Processing Systems* (2019).
- [13] Jalas, S. *et al.* Bayesian optimization of a laser-plasma accelerator. *Physical Review Letters* **126** (2021).
- [14] Kain, V. *et al.* Sample-efficient reinforcement learning for CERN accelerator control. *Physical Review Accelerators and Beams* **23**, 124801 (2020). URL <https://link.aps.org/doi/10.1103/PhysRevAccelBeams.23.124801>.
- [15] Kaiser, J., Stein, O. & Eichler, A. Learning-based optimisation of particle accelerators under partial observability without real-world training. In Chaudhuri, K. *et al.* (eds.) *Proceedings of the 39th International Conference on Machine Learning*, vol. 162 of *Proceedings of Machine Learning Research*, 10575–10585 (PMLR, 2022). URL <https://proceedings.mlr.press/v162/kaiser22a.html>.
- [16] Madysa, N. *et al.* Automated intensity optimisation using reinforcement learning at LEIR. In *Proceedings of the 13th International Particle Accelerator Conference (IPAC2022)*, 941–944 (2022).
- [17] McIntire, M., Cope, T., Ermon, S. & Ratner, D. Bayesian Optimization of FEL Performance at LCLS. In *Proceedings of the 7th International Particle Accelerator Conference* (2016).
- [18] O’Shea, F. H., Bruchon, N. & Gaio, G. Policy gradient methods for free-electron laser and terahertz source optimization and stabilization at the FERMI free-electron laser at Elettra. *Physical Review Accelerators and Beams* **23**, 122802 (2020).
- [19] Pang, X., Thulasidasan, S. & Rybarczyk, L. Autonomous control of a particle accelerator using deep reinforcement learning. In *Proceedings of the Machine Learning for Engineering Modeling, Simulation, and Design Workshop at Neural Information Processing Systems 2020* (2020). URL <http://arxiv.org/abs/2010.08141>.
- [20] Shalloo, R. J. *et al.* Automation and control of laser wakefield accelerators using Bayesian optimization. *Nature Communications* **11**, 1–8 (2020).
- [21] St. John, J. *et al.* Real-time artificial intelligence for accelerator control: A study at the Fermilab Booster. *Physical Review Accelerators and Beams* **24**, 104601 (2021).

- [22] Xu, C. *et al.* Bayesian optimization of the beam injection process into a storage ring. *Phys. Rev. Accel. Beams* **26**, 034601 (2023). URL <https://link.aps.org/doi/10.1103/PhysRevAccelBeams.26.034601>.
- [23] Degraeve, J. *et al.* Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature* **602**, 414–419 (2022).
- [24] Seo, J. *et al.* Feedforward beta control in the KSTAR tokamak by deep reinforcement learning. *Nuclear Fusion* **61** (2021).
- [25] Seo, J. *et al.* Development of an operation trajectory design algorithm for control of multiple 0D parameters using deep reinforcement learning in KSTAR. *Nuclear Fusion* **62** (2022).
- [26] Guerra-Ramos, D., Trujillo-Sevilla, J. & Rodríguez-Ramos, J. M. Towards piston fine tuning of segmented mirrors through reinforcement learning. *Applied Sciences (Switzerland)* **10** (2020).
- [27] Nousiainen, J. *et al.* Toward on-sky adaptive optics control using reinforcement learning: Model-based policy optimization for adaptive optics. *Astronomy and Astrophysics* **664** (2022).
- [28] Yatawatta, S. & Avruch, I. M. Deep reinforcement learning for smart calibration of radio telescopes. *Monthly Notices of the Royal Astronomical Society* **505**, 2141–2150 (2021).
- [29] Zhou, Z., Li, X. & Zare, R. N. Optimizing chemical reactions with deep reinforcement learning. *ACS Central Science* **3**, 1337–1344 (2017).
- [30] Deneault, J. R. *et al.* Toward autonomous additive manufacturing: Bayesian optimization on a 3D printer. *MRS Bulletin* **46**, 566–575 (2021).
- [31] Abdelrahman, H., Berkenkamp, F., Poland, J. & Krause, A. Bayesian optimization for maximum power point tracking in photovoltaic power plants. In *2016 European Control Conference (ECC)*, 2078–2083 (Institute of Electrical and Electronics Engineers Inc., 2016).
- [32] Xiong, Y., Guo, L., Huang, Y. & Chen, L. Intelligent thermal control strategy based on reinforcement learning for space telescope. *Journal of Thermophysics and Heat Transfer* **34**, 37–44 (2020).
- [33] Xiong, Y., Guo, L. & Tian, D. Application of deep reinforcement learning to thermal control of space telescope. *Journal of Thermal Science and Engineering Applications* **14** (2022).
- [34] Baheri, A., Bin-Karim, S., Bafandeh, A. & Vermillion, C. Real-time control using Bayesian optimization: A case study in airborne wind energy systems. *Control Engineering Practice* **69**, 131–140 (2017).
- [35] Maggi, L., Valcarce, A. & Hoydis, J. Bayesian optimization for radio resource management: Open loop power control. *IEEE Journal on Selected Areas in Communications* **39**, 1858–1871 (2021).
- [36] Ding, X., Du, W. & Cerpa, A. E. MB2C: Model-based deep reinforcement learning for multi-zone building control. In *BuildSys 2020 - Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 50–59 (Association for Computing Machinery, Inc, 2020).
- [37] Nweye, K., Liu, B., Stone, P. & Nagy, Z. Real-world challenges for multi-agent reinforcement learning in grid-interactive buildings. *Energy and AI* **10** (2022).
- [38] Panofski, E. *et al.* Commissioning results and electron beam characterization with the S-band photoinjector at SINBAD-ARES. *Instruments* **5** (2021).
- [39] Burkart, F. *et al.* The ARES Linac at DESY. In *Proceedings of the 31st International Linear Accelerator Conference (LINAC'22)*, no. 31 in International Linear Accelerator Conference, 691–694 (JACoW Publishing, Geneva, Switzerland, 2022). URL <https://jacow.org/linac2022/papers/thpojo01.pdf>.
- [40] Rasmussen, C. E. & Williams, C. K. I. *Gaussian Processes for Machine Learning* (The MIT Press, 2005). URL <http://www.gaussianprocess.org/gpml/>.
- [41] Nelder, J. A. & Mead, R. A simplex method for function minimization. *The Computer Journal* **7** (1965).
- [42] Dulac-Arnold, G., Mankowitz, D. & Hester, T. Challenges of real-world reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning* (2019).
- [43] Tobin, J. *et al.* Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 23–30 (2017).
- [44] OpenAI *et al.* Solving Rubik’s cube with a robot hand (2019). Preprint available at <https://arxiv.org/abs/1910.07113>.
- [45] Wang, W. *et al.* Accelerated deep reinforcement learning for fast feedback of beam dynamics at KARA. *IEEE Transactions on Nuclear Science* **68**, 1794–1800 (2021).
- [46] Scomparin, L. *et al.* KINGFISHER: A Framework for Fast Machine Learning Inference for Autonomous Accelerator Systems. In *Proc. 11th Int. Beam Instrum. Conf. (IBIC'22)*, no. 11 in International Beam Instrumentation Conference, 151–155 (JACoW Publishing, Geneva, Switzerland, 2022). URL <https://jacow.org/ibic2022/papers/mop42.pdf>.
- [47] Nyikosa, F. M., Osborne, M. A. & Roberts, S. J. Bayesian optimization for dynamic problems (2018). Preprint available at <https://arxiv.org/abs/1803.03432>.
- [48] Krause, A. & Ong, C. Contextual Gaussian process bandit optimization. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F. & Weinberger, K. (eds.) *Advances in Neural Information Processing Systems*, vol. 24 (Curran Associates, Inc., 2011). URL <https://proceedings.neurips.cc/paper/2011/file/f3f1b7fc5a8779a9e618e1f23a7b7860-Paper.pdf>.
- [49] Brockman, G. *et al.* OpenAI Gym (2016).
- [50] Stein, O., Kaiser, J. & Eichler, A. Accelerating linear beam dynamics simulations for machine learning applications. In *Proceedings of the 13th International Particle Accelerator Conference* (2022). URL <https://github.com/desy-ml/cheetah>.
- [51] Fujimoto, S., van Hoof, H. & Meger, D. Addressing function approximation error in actor-critic methods (2018). Preprint available at <https://arxiv.org/abs/1802.09477v3>.
- [52] Raffin, A. *et al.* Stable Baselines3 (2019). URL <https://github.com/DLR-RM/stable-baselines3>.
- [53] Balandat, M. *et al.* BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33* (2020). URL <https://proceedings.neurips.cc/paper/2020/hash/>

f5b1b89d98b7286673128a5fb112cb9a-Abstract.html.

- [54] Matérn, B. *Spatial Variation*, vol. 36 (Springer New York, 1986), 2 edn. URL <http://link.springer.com/10.1007/978-1-4615-7892-5>.
- [55] Jones, D. R., Schonlau, M. & Welch, W. J. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization* **13**, 455–492 (1998).
- [56] Virtanen, P. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **17**, 261–272 (2020).
- [57] Eichler, A. *et al.* First steps toward an autonomous accelerator, a common project between DESY and KIT. In *12th International Particle Accelerator Conference : virtual edition, May 24th-28th, 2021, Brazil : proceedings volume / IPAC2021. Ed.: R. Picoreti*, 2182–2185 (JACoW Publishing, 2021). 54.11.11; LK 01.
- [58] Huang, X. *et al.* Development and application of online optimization algorithms. In *Proc. North Amer. Part. Accel. Conf (NAPAC;), Chicago*, 1–5 (2016).
- [59] Olsson, D. K. *et al.* Online optimisation of the MAX-IV 3 GeV ring dynamic aperture. *Proc. IPAC2018* **2281** (2018).
- [60] Pang, X. & Rybarczyk, L. Multi-objective particle swarm and genetic algorithm for the optimization of the LANSCE linac operation. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **741**, 124–129 (2014). URL <https://www.sciencedirect.com/science/article/pii/S0168900213017464>.
- [61] Tian, K., Safranek, J. & Yan, Y. Machine based optimization using genetic algorithms in a storage ring. *Phys. Rev. ST Accel. Beams* **17**, 020703 (2014). URL <https://link.aps.org/doi/10.1103/PhysRevSTAB.17.020703>.
- [62] Tomin, S. *et al.* Progress in automatic software-based optimization of accelerator performance. In *Proceedings of the 7th International Particle Accelerator Conference* (2016).
- [63] Zhang, Z. Badger: The Ocelot Optimizer rebirth. Tech. Rep., SLAC National Accelerator Lab., Menlo Park, CA (United States) (2021).
- [64] Zhang, Z. *et al.* Badger: The missing optimizer in ACR. In *Proceedings of the 13th International Particle Accelerator Conference (IPAC 2022)* (2022). URL <https://slac-ml.github.io/Badger>.



TABLE I: Performance of optimisation algorithms on the benchmark tuning task.

Optimiser	Final beam difference ( $\mu\text{m}$ )		Steps to target			Steps to convergence		
	Median	Mean	Median	Mean	Success rate	Median	Mean	Success rate
<i>Simulation (infinite diagnostic screen, no beam alignment) <math>\rightarrow</math> Section II A</i>								
Random search	460	$490 \pm 200$	-	-	0.0 %	33	$51 \pm 51$	100.0 %
Nelder-Mead simplex	270	$300 \pm 160$	56	$56 \pm 0$	0.3 %	29	$28 \pm 15$	100.0 %
RLO	4	$11 \pm 19$	9	$16 \pm 20$	88 %	7	$9 \pm 11$	100.0 %
BO	45	$60 \pm 55$	52	$61 \pm 34$	12 %	32	$42 \pm 31$	99.7 %
<i>Real world (finite diagnostic screen, beam aligned to quadrupole magnets) <math>\rightarrow</math> Section II B</i>								
RLO <sup>ab</sup>	24	$29 \pm 48$	15	$17 \pm 11$	41 %	10	$10.4 \pm 5.5$	100.0 %
BO <sup>ac</sup>	44	$58 \pm 41$	52.5	$52.5 \pm 5.5$	10 %	33	$33 \pm 16$	100.0 %
<i>Simulation (finite diagnostic screen, no beam alignment) <math>\rightarrow</math> Section II E</i>								
RLO	4	$13 \pm 40$	8	$13 \pm 13$	85 %	7	$8.2 \pm 6.8$	99.3 %
BO	38	$61 \pm 96$	45	$45 \pm 14$	10 %	28	$32 \pm 15$	83.3 %
<i>Simulation (finite diagnostic screen, beam aligned to quadrupole magnets) <math>\rightarrow</math> Section II E</i>								
RLO	4	$10 \pm 18$	8	$13 \pm 13$	87 %	7	$7.3 \pm 5.2$	99.7 %
BO	23	$28 \pm 19$	40	$40 \pm 13$	35 %	25	$26 \pm 10$	97.0 %

The results reported in simulation have been collected over 300 different trials, for each of which the algorithms were given 150 steps for the optimisation. Because of limited beam time availability at the real accelerator, 22 trials<sup>a</sup> were selected for the real-world evaluations, RLO was given 50 steps<sup>b</sup> for the optimisation and BO was given 75 steps<sup>c</sup> for the optimisation. The final beam difference is computed as the MAE over the beam parameters at the end of the optimisation according to Eq. (2). Steps to target are computed as the number of steps it took until an MAE to the target beam smaller than the measurement accuracy of  $\epsilon = 20 \mu\text{m}$  was achieved. Steps to convergence are defined as the number of steps after which the improvement of the best seen MAE remains smaller than the measurement accuracy  $\epsilon$ . For each metric, we report the median and the mean with standard deviation. Both steps to target and steps to convergence metrics are reported only over trials for which the target or convergence, respectively, were achieved before the optimisation was terminated. We therefore also report success rates, that is the proportion of trials for which the target or convergence, respectively was successfully achieved before termination.

TABLE II: Results of feedback and actuator failure studies.

Optimiser	Normal	Feedback		Magnet failure	
		Instant	Continuously	Before	During
RLO	16	18	80	40	16
BO	73	350	250	45	50

The values reported are mean final beam differences as MAEs from the observed beam parameters to the target beam parameters in  $\mu\text{m}$ . All optimisations reported in this table were terminated at 80 steps. We therefore also report results for normal tuning runs where the upstream beam changes or magnet failures were introduced.