MACHINE
LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
9 December 2022

REVISED
5 April 2023

ACCEPTED FOR PUBLICATION
18 April 2023

PUBLISHED
27 April 2023

PAPER

Towards automatic setup of 18 MeV electron beamline using machine learning

Francesco Maria Velotti[1,*], Brennan Goddard[1], Verena Kain[1], Rebecca Ramjiawan[1], Giovanni Zevi Della Porta[1] and Simon Hirlaender[2]

[1] CERN, Geneva, Switzerland
[2] University of Salzburg, Kapitelgasse 4/6, 5020 Salzburg, Austria
[*] Author to whom any correspondence should be addressed.

E-mail: francesco.maria.velotti@cern.ch

Keywords: automatic, electron, beamline, reinforcement learning, reward dangling

Abstract

To improve the performance-critical stability and brightness of the electron bunch at injection into the proton-driven plasma wakefield at the AWAKE CERN experiment, automation approaches based on unsupervised machine learning (ML) were developed and deployed. Numerical optimisers were tested together with different model-free reinforcement learning (RL) agents. In order to avoid any bias, RL agents have been trained also using a completely unsupervised state encoding using auto-encoders. To aid hyper-parameter selection, a full synthetic model of the beamline was constructed using a variational auto-encoder trained to generate surrogate data from equipment settings. This paper describes the novel approaches based on deep learning and RL to aid the automatic setup of a low energy line, as the one used to deliver beam to the AWAKE facility. The results obtained with the different ML approaches, including automatic unsupervised feature extraction from images using computer vision are presented. The prospects for operational deployment and wider applicability are discussed.

1. Introduction and motivation

The AWAKE experiment [1] at CERN's Super Proton Synchrotron (SPS) uses proton-driven plasma wake-fields to accelerate an 18 MeV electron witness bunch to about 2 GeV over a distance of 10 m. Efficient capture and acceleration relies on precise delivery of a dense low-energy e^- bunch to the correct location in space and time in the plasma. The e^- beam brightness (intensity divided by transverse beam size) and position are therefore critical to the performance of the overall facility, as evidenced by the experiments performed in 2018 [2].

The low energy 18 MeV c^{-1} e^- beamline requires time-consuming and frequent optimisation, given its high sensitivity to initial conditions, to equipment settings and to the bunch momentum distribution, as well as inherently less predictable environmental effects like temperature, magnetic history and the pulsing of the adjacent 400 GeV c^{-1} proton beamline. The commissioning of the e^- beamline highlighted the criticality of the magnetic element modelling and of the incoming beam energy jitter [3].

Some of the contributions to beam quality degradation are completely random or uncontrollable, with timescales ranging from seconds to hours or even days. Time-consuming and sometimes non-reproducible manual tuning of the e^- source and beamline parameters were needed to satisfy the experiment requirements.

The main fluctuations are observed on the transverse beam quality and are believed to be caused by chromatic aberrations and optical mismatch at the injection point. Problems of this type for low-energy lines are known [4, 5], although large variations usually also affect the longitudinal beam delivery. A similar multi-objective optimisation is needed, e.g. position, angle, emittance and charge delivered, with a large number of free tuning parameters.

To reduce the time and personpower effort needed for setting up, and to improve the stability and also potentially the absolute performance reach, model-free ML automation approaches were investigated.

At CERN the usage of numerical optimisation for modelling and design is not new—the most representative example is the different algorithms available in the MADX design tool. The exploitation of these algorithms on real operating machines is, on the contrary, very new and only in the recent years, the full exploitation of numerical optimisers and high level controllers has started. One of the first examples to bridge between the modelling and operation world for the CERN accelerators is reported in [6]. In other facilities, recent works showed how numerical optimisers can significantly speed up the tuning of particle accelerators [7], also when combined with machine learning (ML) techniques.

Furthermore, many recent studies [8–10] propose ideas and applications to different accelerator and facilities, spanning from deep neural networks to convolutional ones to control and optimise accelerator performance. For instance, in [11], the authors proposed a neural network based controller for the optimisation of the longitudinal distribution of the their electron bunch using a modified version of the adaptive extremum seeking controller. One of the main conclusions, which is also consistent with our experiments, is that the time-varying nature of most of the accelerator components makes it non-trivial the operational deployment of pre-trained neural networks. For this reason, the approach of reinforcement learning (RL), assuming that all dominant observable are included in the problem, represents a suitable candidate to deal with time-varying, high-dimensional and non-linear systems of particle accelerators.

RL has already been successfully applied in different scientific domains [12–14], showing incredible potential for problems where it is possible to clearly quantify the final objective and reduce the description on how to do that. For example, the usage of RL to control the plasma on the tokamak à configuration variable [15] showed how it is indeed possible to use RL to control a highly non-linear system and still have the freedom to define a high-level control objective for the problem.

On these notes, a very promising approach is detailed in [16], where model-based RL is applied to control the free-electron laser (FEL) on the Free Electron laser Radiation for Multidisciplinary Investigations at Elettra Sincrotrone Trieste. The authors applied RL to the trajectory correction of the seeding laser and the electron beam to increase the FEL radiation. They proposed extremely sample efficient algorithms that can provide a solution to short horizons control, like ours. Such algorithms are of significant interest also for the application discussed in the paper, and indeed this could represent a possible future step towards the full automation of the beamline under analysis.

Surrogate models have also been already developed in accelerator facilities [17–19], to both speed up complex simulations or to produce online models for quick machine tuning. Also, the usage of deep neural networks to create hybrid models using simulations and data [20, 21] showed how machine data can improve the quality of purely simulation-based surrogate models.

The work reported in this paper extends the results presented in [22, 23]. Here we go a step further and we apply Deep RL (DRL) to a non-convex problem, which is what is needed for daily operation of this beamline, making it a real-world example. As it will be clearer in the text, the response we are aiming to control is highly non-linear and populated with many local maxima. Also, the high dimensionality of the Action space makes the problem iteration eager. Here we propose the exploitation of low-dimensional encoding, which is provided by the surrogate model architecture chosen, to provide an unsupervised state description to the RL agent, as this is a key requirement for the success of RL agent learning process. In our experiments, we tested both *explicit* state encoding, using the output of analytical Gaussian image fits, and *implicit* state (or feature extraction) encoding, where the encoder from a trained variational auto-encoder (VAE) gives a representation of the image in a low-dimensional $\mathbb{Z}$ latent space, which was then used directly as implicit state information for the RL agent. This automatic unsupervised feature extraction could be critical for RL applications where explicit state feature description and extraction is difficult (very high dimensional problems) or impossible, for instance in the observation of Schottky spectra.

Acting on the initial matching triplet and the low-energy solenoids, the beam brightness was optimised, using as observation the image of the beam on a beam monitoring screen (BTV [24]) at the entrance of the plasma cell, figure 1. To test and deploy the different types of optimisation agents, an interface to the SPS control system was used together with the generic OpenAI Gym [25] environment framework for the ML tasks. A surrogate model using computer vision in an VAE trained on the machine data was an important part of the work, allowing fast *in-silico* testing and tuning of different algorithms and approaches without beam time.

For the optimisation of the beam brightness, the two different approaches investigated were numerical optimisers and RL. For the numerical optimisers, by varying the equipment parameters the algorithm aims to maximise or minimise an objective function calculated from the BTV image, which it must perform each time it is used. For RL, the agent aims to learn the response of the system during a training phase, such that it can quickly move to the optimum in the subsequent deployment.

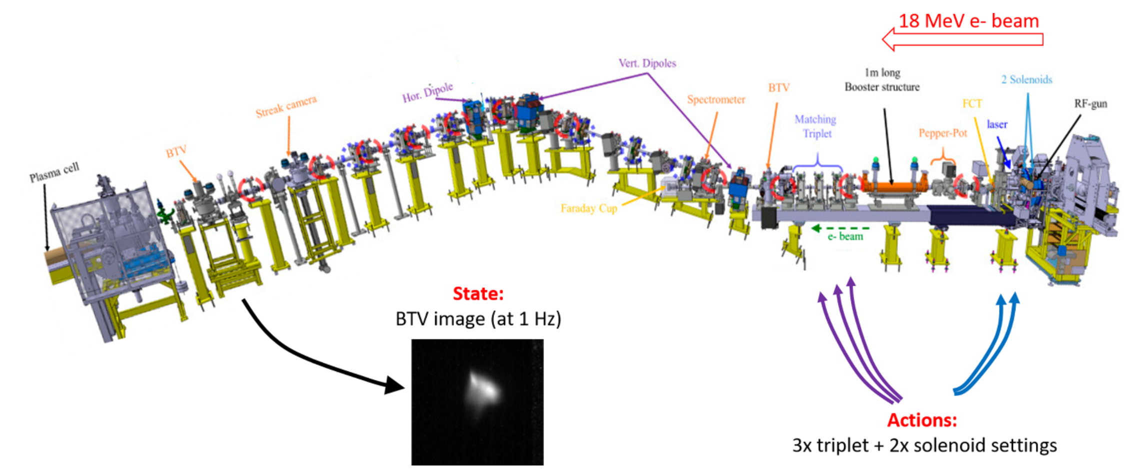


Figure 1. AWAKE beamline showing location of the matching devices (actions) and the observation BTV.

The relevant performance metrics for both types of approach are the sample efficiency (number of interactions needed with the machine for the algorithm to converge) and the final beam brightness achieved. The RL agents, based on the Markov assumption, have the advantage of not needing to repeat the exploration phase, once the underlying dynamics have been learned, but unlike the optimisers will not perform well on subsequent deployment if the underlying dynamics of the system are non stationary.

To facilitate hyper-parameter optimisation, agent selection and investigate transfer learning, the decoder of the trained VAE was also used to generate a full synthetic model of the system. This model is able to encode and decode images to and from a latent space $\mathbb{Z}$ using an additional predictor neural network to ensure the correspondence between $\mathbb{Z}$ with the equipment setting configuration $\mathbb{C}$. In this way, it can replace the real beamline to help tune and test any algorithms.

This paper introduces the AWAKE e$^-$ beamline with its operational challenges, and explains the technique for matching for maximum beam brightness at the injection point. The methodology for the implementation of the different optimisers and RL agents is presented, together with the VAE. The construction of the synthetic model is briefly described. The performances of the different approaches deployed on AWAKE are compared, including comparison with the synthetic model results. Technical aspects such as implicit versus explicit state representation are addressed, including a technique the authors developed for overcoming the inherent difficulty with RL reward shaping which simplified and stabilised training and improved overall sample efficiency. Finally, future work and the prospects for operational deployment and wider applicability are discussed.

All the code developed to train the models, define the beamline environment, deploy in operation and produce a surrogate of the full electron line described in this paper is available on GitLab.

1.1. AWAKE electron transfer line and optics

The AWAKE experiment uses a 400 GeV c^{-1} proton transfer line to transport the drive beam from the SPS to the plasma cell. The 18 MeV e$^-$ beam is produced in a side-gallery and needs to be fed into the same plasma cell with high delivery precision. An initial S-band RF photo injector produces a 200 pC e$^-$ beam at around 5 MeV c^{-1} which is then accelerated in a travelling wave accelerating structure up to about 18 MeV c^{-1} [26]. Two low energy solenoids are used just after the photo injector to focus the beam inside the accelerating structure and hand it over to the transfer line. The latter is equipped with an initial and final matching quadrupole triplets to ensure losses transport and final focus at the plasma cell entrance.

The existing tunnel geometry was an important constraint on the optics design of the e$^-$ line. In figure 2 the optics of the e$^-$ beamline is shown. It comprises two achromatic sections: one vertical dog-leg and one 60° horizontal bend to go from the RF gun to the plasma cell, as well as matching elements. Due to the difference in the vertical slope between the tunnel of the RF gun and the plasma cell, the vertical dispersion is matched to zero locally at the merging point but with a finite dispersion angle.

In order to ensure capture of the injected e$^-$ in the plasma accelerating structure, the beam size at the entrance of the plasma cell has to be 250 μm in both planes, with as high intensity as possible. Due to the strong bends and quadrupoles in the beamline, the chromatic aberrations significantly degrade the deliverable beam quality [3]. This is accentuated by the shot-to-shot momentum jitter observed from the e$^-$ source.

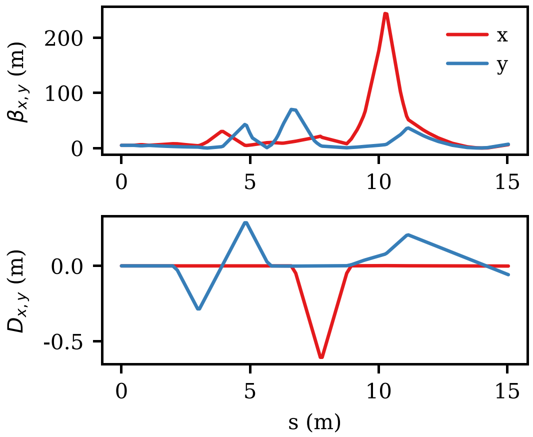


Figure 2. Electron beamline linear optics functions. Top: beta functions for both horizontal and vertical plane. Bottom: dispersion functions.

The AWAKE e$^-$ line commissioning [3] showed that most of the non-reproducibility is due to variations in the beam from the source, which changes the matching needed to the target. This translates into lengthy setting up to produce the required beam parameters, which must be repeated every time the source is restarted.

2. Optics matching

The beam brightness and position need to be controlled at the injection point. The dependence on high order aberration of the optics plus the variations in the initial conditions excludes purely analytical matching of the target beam size σ^* using only linear optics, as a global solution. In fact, analytical matching is possible only if the initial conditions are close enough to the design.

The last beam screen (Beam TV - BTV) is located 0.8 m upstream of the injection point, as installation inside the plasma cell was not possible. The spot brightness can only be maximised at this screen. An optics trim then needs to be applied to move the optics waist to the required location. Accurate knowledge of the beam optics functions is thus fundamental.

With low energy e$^-$ the usage of multiple screens for single-shot optics measurement is impossible as the beam is completely disrupted by the screen. For the AWAKE e$^-$ line an ad-hoc measurement optics was developed which presents a global minimum at a specific longitudinal location. The brightness is then maximised on the final screen, using only the upstream triplet. The global minimum is moved from this screen to the injection point using only four quadrupoles in the final part of the line, and leaving the initial matching quadrupole strengths unchanged.

In theory the dispersive contribution should be taken into account, but this is not so important for our specific application, as the initial dispersion is very close to zero. If the beam is well centred in the initial triplet [3], no significant contribution from the variation of the first three quadrupoles is expected to the dispersion functions.

The displacement of the focal point was fully tested in simulations. Although not yet deployed experimentally, the results and methods shown in this paper are not linked to the success of this methodology. For the future, the installation of a BTV in the plasma volume at the injection point is under investigation, which would remove the need for this step.

The beam quality at the end of the line is also optimised with the low energy solenoids before the accelerating structure, which help to minimise the emittance produced. This can be achieved using the following penalty function r_σ,

$$r_\sigma = \sqrt{(\sigma_x - \sigma_x^*)^2 + (\sigma_y - \sigma_y^*)^2} \tag{1}$$

choosing σ^* using the lowest achievable emittance from the gun (as previously measured, i.e. 0.9 mm mrad). This ensures that the target function has a single global minimum in the 5-dimensional Action space. The validity of this approach was tested in simulation using a model including known non-linear effects. Figure 3 shows that the evolution of r_σ as a function of beam initial conditions, i.e. $(\beta,\alpha)_{x,y}$, and for different initial emittances. As the figure of merit is in each case a convex function with a clear minimum (red dot), r_σ can be incorporated into a target function to ensure the matching of the beam produced from the source to the

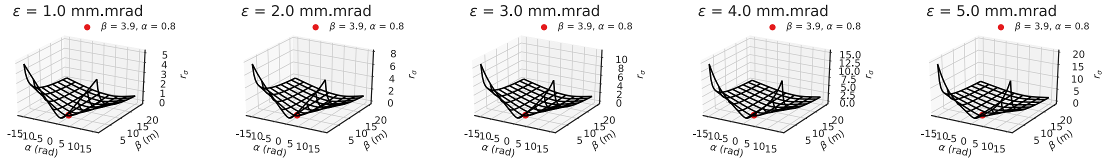


Figure 3. Scan of sensitivity of matched beam spot-size to initial optics conditions, for different emittances.

transfer line. The same minimum location is also found in initial optics space for different emittances, which shows that the approach should be insensitive to emittance variation.

3. ML framework and methodology

Thanks to the high repetition rate of the AWAKE e$^-$ source, a large set of ML algorithms can be explored with the matching approach described above. The python Open AI gym environment framework was used as a standard, with the pyJAPC [27] library for interfacing between the ML algorithms and the CERN control system. The TensorFlow (version 1.14) back-end was used for the machine vision and VAE. Different numerical optimisers and Stable Baselines [28] RL agents were explored for both explicit and implicit state representation. Bound Optimization BY Quadratic Approximation ([29]) and Twin Delayed Deep Deterministic Policy Gradient algorithm (TD3) [30] gave the best results in the two classes.

3.1. DRL

RL is a branch of ML that deals with learning from interaction with an environment. In RL, an agent observes the state of the environment, chooses an action to perform, and receives a reward (or penalty) based on the outcome of the action. The goal of the agent is to maximize its cumulative reward over time by learning a policy that maps states to actions. In the next sections, we refer to cumulative reward as the sum of the rewards obtained in each episode.

One of the challenges of RL is dealing with high-dimensional and complex state and Action spaces, such as those encountered in robotics, computer games, natural language processing or accelerators. DRL is an approach that combines RL with deep neural networks to learn policies that can handle such complex domains. DRL leverages the representation power and generalization ability of deep neural networks to learn from raw sensory inputs and discover useful features for decision making.

One of the popular algorithms for DRL is TD3, which is an extension of Deep Deterministic Policy Gradient (DDPG). TD3 is designed for continuous control tasks, where the agent has to choose a real-valued action at each step. TD3 improves upon DDPG by addressing two issues: overestimation bias and policy overfitting.

3.2. Action space

For all synthetic and machine tests described, the Action space consisted of the two solenoid and the three quadrupole currents, as indicated in figure 1. In the OpenAI gym agents, the values are normalised to a range of ±1, while in reality the solenoid currents are 0–400 A and the quadrupoles −100 to 100 A. The Action space is therefore bounded to the physical limits of the different circuits, such that no other limits or penalties are needed for the agents. Although the line runs at 10 Hz, some time is needed between setting and acquiring which is dealt with in the generic OpenAI environment, which limits the rate at which scans can be made to about 0.5 Hz.

3.3. Observation

The observation is the BTV image, figure 1, from which we derive both the single objective function (brightness) and the state for the RL agents. The BTV image provides a 256 × 256 pixel 8-bit grey-scale array, from which beam profiles, intensity and more complex information can be extracted at 1 Hz. The images were down-sampled to 128 × 128 pixels, which was the dimensionality for the VAE encoder and decoder used in the synthetic model. The lost of information arising from the down-sampling the BTV image was assessed by comparing the estimation of the beam size on both original and down-sampled version, showing no significant difference.

3.4. Optimiser objective function

All results reported here were obtained optimising a single objective function. Studies made using multi-objective optimisation with an extremum-seeking optimiser have been reported separately [23].

The optimiser tries to minimise the objective function, which should therefore be large and positive when far from the optimum, and small when close to the ideal solution. The objective function used for the optimisers was defined by combining of two contributions:

$$r_i = \frac{1}{i_0}\sum_{j,k} a_{jk} - i_0 \tag{2}$$

$$r_\sigma = r_0 - \frac{1}{r_{\max}}\sqrt{(\sigma_x - \sigma_x^*)^2 + (\sigma_y - \sigma_y^*)^2} \tag{3}$$

where $\sum_{i,j} a_{ij}$ is the measured sum of all pixel values and σ_x^* and σ_y^* were both set to 0.1 mm to represent a target minimum beam size, i_0 was set at a numerical value of 1.3×10^6 slightly above the maximum ever recorded sum of pixels; r_0 and $r_{\max}$ were set to 0.25 mm and 3.0 mm respectively, as minimum and maximum spot sizes. The 0.25 mm is the target beam size from the experiments and 3 mm is taken sufficiently large not to impact the convergence of the numerical algorithms. The two contributions are then put together to for the actual penalty function for the optimiser as:

$$r_o = -1[r_\sigma \alpha_s + r_i(1-\alpha_s)] \tag{4}$$

where α_s represent the weight for the beam size contribution. It was empirically found that 20% weighted contribution from the image amplitude r_a over all pixels and 80% from the beam size r_σ led to convergence to solutions that avoid high losses along the line. Several tests were carried out with different sharing and a sharing of 80–20 showed the best performance without leading to solutions that are not operationally valid. In fact, the function is designed to encourage simultaneous high intensity and small beam size, as this would favourite configurations with high losses along the line.

3.5. Generative VAE for synthetic model

A VAE [31] based on computer vision convolutional neural networks was used to generate synthetic BTV images from the real AWAKE data in an unsupervised manner. This was then used both for state encoding from the BTV images, as well as building a synthetic model (digital twin) of the AWAKE beamline. The basic AE is a pair of neural networks consisting of an encoder, an information bottleneck, and a decoder. The loss function is built of two parts: the reconstruction accuracy which measures how close the decoded data is to the original data, and a divergence term which measures how the information contained in the latent encoding differs from a Gaussian distribution. The pair of networks try to reconstruct the original data as accurately as possible, passing through the low-dimensional information bottleneck.

The VAE uses an additional random term added to the encoded latent space coordinate. Even with limited discrete training data this has the effect of producing a continuous variation of encodings in the latent space, ideal for state variables which we expect to be continuous with changes in the actions for our system. In our experiments we tried different VAE flavours, settling for the β-VAE with loss function of the form:

$$L(\theta,\phi,\beta) = -E_{z\approx q_\phi(\mathbf{z}|\mathbf{x})}\log p_\theta(\mathbf{x}\mid\mathbf{z}) + \beta D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\mid\mid p_\theta(\mathbf{z})) \tag{5}$$

as from [32], where the first term represents the reconstruction loss and the second one is the Kullback–Leibler divergence (KL) which pushes the probability distribution of decoder and encoder to be as similar as possible to a Gaussian distribution. The KL term is weighted with the β parameter which can be considered an additional hyper-parameter to choose (examples of produced images are shown in Annex figure 15). To complete the full synthetic model, a densely connected neural network was used as surrogate model to make the correspondence between the (labelled) Action space and the latent space $\mathbb{Z}$ encoding. The overall architecture for training the VAE and Predictor is shown in figure 4, with the synthetic model shown in figure 5.

These networks were trained on machine data, from a grid-scan made in the 5D Action space. This was a lengthy one-off process needing the accumulation of some 1500 valid images to train the VAE, but allowed the efficient and comprehensive off-line training and hyper-parameter optimisation for comparison of different agents, objective functions, state encoding and reward shaping for the different agents investigated. The size of the grid search was constrained by the beam availability, but it was sufficient to train VAE models capable to reconstruct the measured images with sufficient accuracy to be able to extract a meaningful lower representation.

The lengthy grid search was made to train the surrogate model but could theoretically be used to find the optimal working point. But it would not be a viable method as the beamline input conditions vary quite significantly from day to day (which also impacted the suitability of the single RL agent approach, as addressed in the discussion).

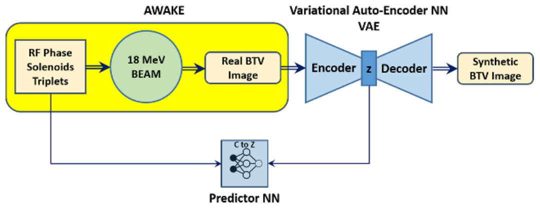


Figure 4. Architecture for training VAE and Predictor networks used for AWAKE synthetic model and RL encoding.

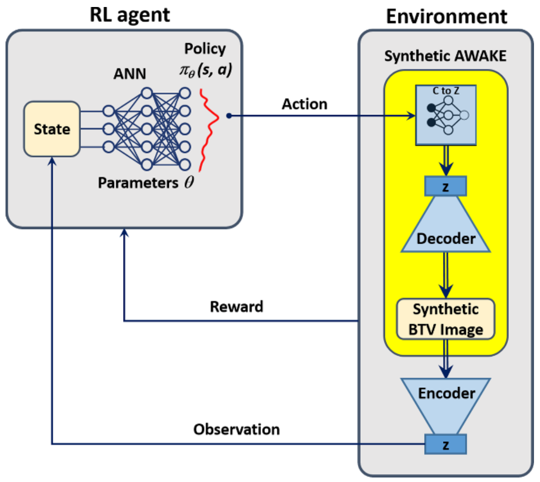


Figure 5. Use of AWAKE synthetic model for RL testing. Note that the decoder and encoder networks do not need to come from the same VAE model.

3.6. RL state space

As previously introduced, RL agents need a state space based on the observation, as well as an action space which changes the state. Two approaches for state representation were investigated. For explicit state extraction, the parameters σ_x, σ_y, μ_x and μ_y were obtained from numerical fitting of a 2D Gaussian to the BTV image and the intensity extracted from the pixel sum. For implicit state extraction from the image we used the computer vision encoder network from a trained VAE and fed the latent vector $\mathbb{Z}$ directly to the RL agent as the state representation. In this case it is important to note that the elements of the state vector do not correspond directly to individual physics parameters. This automatic unsupervised feature extraction could be essential for problems where the explicit extraction of state features is difficult or impossible, or where some hidden state features are to be expected. Given that some of the profiles obtained from the AWAKE BTV are highly non-Gaussian (see Annex figure 16(a)), it was hoped that this implicit approach would help more completely capture the underlying image dynamics.

Since most of the information encoded in a low-dimensional latent space corresponds to the position of the beam spot, we experimented testing a centring algorithm to produce the correct sized data array centred about the brightest part of the image. The results were rather similar and we finally ran almost all experiments without explicitly correcting for the beam position on the screen. This was valid in the context of optimising for the beam brightness—separate studies have shown that the beam position stabilisation can be treated separately [23], using dipole correctors and position monitors.

An investigation of VAE architecture and hyper-parameters was made to achieve stable results. Different flavours of VAE were tried in attempts to produce a more disentangled latent space description, eventually we used a β-VAE flavour [32]. The dimensionality of the $\mathbb{Z}$ space was also studied in terms of suitability for RL state encoding, since a larger value allows more accurate reconstruction, but would intuitively seem likely to complicate the encoding challenge for the RL agent. One useful metric was the eigenmode decomposition of the latent space matrix for all encoded images, figure 6, which allowed us to see how many of the latent dimensions were actually encoding independent information, see for example figure 6.

Tests on the synthetic AWAKE model confirmed that RL agents using a state encoding dimension of 64 or 512 failed to converge. We therefore fixed the latent space dimensionality at 5.

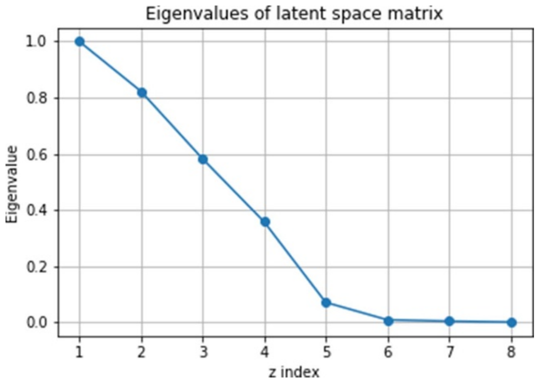


Figure 6. Eigen modes of latent space matrix for encoded images with $\mathbb{Z}$-dimension of 8. It can be seen that only 5 significantly independent image modes exist.

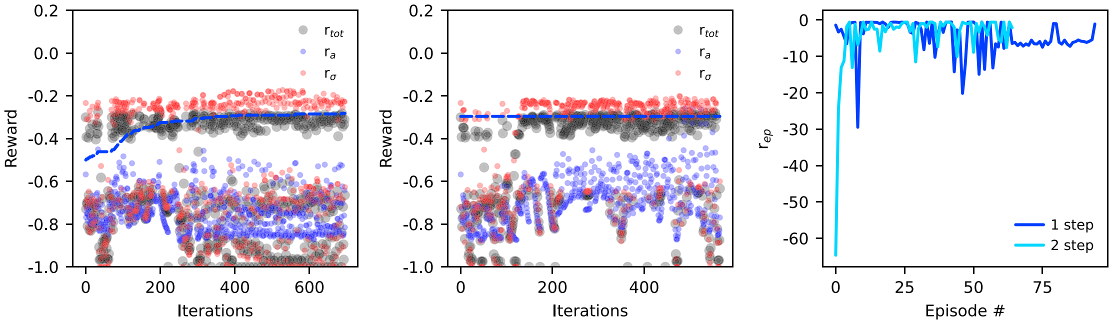


Figure 7. Training of TD3 agent on synthetic AWAKE model. The first figure shows the evolution of the total reward (black) and its two components (red for the beam size and blue for the intensity components) as a function of the number of iterations (interactions with the environment) during the first part of the reward dangling algorithm. The evolution of the reward threshold is shown in blue dashed line. The middle figure shows the evolution of the total reward per episode for the training of the agent with the chosen reward threshold. The right figures shows the evolution of the cumulative reward as a function of the episode during the two stages of training of the reward dangling algorithm.

3.7. RL reward

The RL agent needs a reward which is provided after each action. One important aspect is that the agent aims to maximise the reward over the learning process. Since we are interested in finding a sample-efficient solution which takes as few action steps as possible, our reward function needs to be large and negative when far from the optimum solution, and small but still negative when close to the optimum.

We found that an appropriate choice of reward function per iteration r_i was a very important factor in stable RL agent performance. Our function was constructed to be always negative for realistic observation parameters, and to have a maximum of around 0. This was done combining equations (2) and (3), as done in equation (4) but multiplying it by -1 to obtain suitable reward for our agent:

$$r_i \equiv -r_o. \tag{6}$$

Importantly, the two separate contributions were clipped in the range $[-1, 0]$.

Another aspect that was investigated was the ending-episode reward: this is usually a large positive number which would significantly increase the cumulative reward along the whole episode. It was experimented with and without and found that the difference in terms of number of machine iterations needed for full training was negligible. To compare different techniques to represent the observation space though, the end-episode reward was found to help the metric chosen to classify them, and hence only in this particular case, a positive reward of 20 (chosen arbitrarily large) was assigned to the agent at successful completion of an episode.

3.8. RL episode termination and reward dangling

The correct termination of the RL episode was also a critical factor in the stable performance. For this, we introduced a *reward target* r_t, which when achieved terminated the episode. This introduced a new problem, since the correct setting of this threshold value then also turned out to be an important hyper-parameter. Too low (easy) and the agent would not achieve a good performance, while too high (hard) and the agent would fail to train. Since the reward at the start of the episodes varied unpredictably, and also since the final achievable performance varied depending on the specific run conditions, we needed to develop an automatic way of setting r_t. We opted for *reward dangling*, where we split the RL agent training into two parts—in the first part, r_t was fixed at a very low easy value, typically -0.5. A pseudocode of the procedure is detailed below:

Algorithm 1. Reward dangling

```
**procedure** REWARD DANGLING($\alpha, \gamma$)
    $\alpha = 0.1$
    $\gamma = 0.99$
    $r_t \leftarrow -0.5$
    **while** $i_e < N_{e,max}$ **do**
        $r_f \leftarrow$ Run-episode-training($i_e, r_t$)
        **if** $r_f > r_t$ **then**
            $r_t \leftarrow r_t * \gamma$
        $i_e \leftarrow i_e + 1$
    $r_t \leftarrow r_t * (1 + \alpha)$
    trained-agent $\leftarrow$ Run-training($r_t$)
```

Every time that an episode was successfully concluded, r_t was then increased slightly (by multiplying by a factor γ, typically 0.99). The training then got slightly more difficult, until at some stage with a high r_t the agent failed to train, or takes many iterations.

The training was then repeated with a fixed value of r_t, set at $1 + \alpha$ times of the final value of r_t during the dangling phase. With this approach, we observed stable results despite variations in the final value of the reward per episode.

A final validation run with the trained agent was then used to determine the performance, starting in a random configuration in the Action space. An example of the full reward dangling technique tested on the synthetic AWAKE is shown in figure 7.

Figure 7 shows the full cycle of the reward dangling technique. First, the training of a RL agent continuously changing the target reward (a) and then (b) the final train of the final RL agent with the chosen reward target. It is clear how towards the end of the training the total reward is shifted towards the target, as well as its components in a rather equal manner.

3.9. VAE encoders trained on synthetic image data

One of the drawbacks of training the VAE on the real AWAKE machine is the time needed to acquire the data. Since the state encoding is implicit, we reasoned that training an encoder to respond to artificially-generated images with similar spatial variations could remove the need for this step. We prepared three different data sets.

Firstly a set of about 9'500 real BTV images with distinct action parameter settings were filtered from all 30'000 or so 2019 measurements using Isolation Forest regression—this was necessary as the source could trip randomly during the different data taking campaigns and these anomalous data had to be removed. These were re-scaled to 128 × 128 pixels, divided into 6 sub-datasets of 1'588 images and normalised to the range $[0, 1]$ using the min-max pixel values of the first sub-dataset.

A fully synthetic dataset of 1'588 images was also prepared by combining random numbers of 2D Gaussian with randomly determined amplitude, $\sigma_{x(y)}$, $\mu_{x(y)}$ and tilt angle. Again, all images were then scaled to $[0, 1]$ using the min-max pixel values of this dataset.

Finally, to generate synthetic data from a small training set of real images, we employed a Wasserstein Gradient Penalty GAN (W-GAN) [33], which is a variant of generative adversarial networks that uses a gradient penalty term to enforce the Lipschitz constraint on the discriminator. The W-GAN was trained on 200 images from the real image dataset, which were randomly selected and resized to 128 × 128 pixels.

The architecture of the different VAEs consisted of a encoder and a decoder, both composed of nine convolutional layers and one fully connected layer. The number of filters in each convolutional layer followed the scheme $[1, 1, 2, 2, 4, 4, 8, 8, 16]$, where each number was multiplied by 16 for the encoder. For the decoder, the scheme was instead $[1, 2, 4, 8]$. We used ReLU activation for both the generator and the discriminator.

As an example, we trained the W-GAN for 10 epochs each with a batch size of 16 using Adam optimizer with a learning rate of 5×10^{-4} and $\beta_1 = 0.1$. The training process took approximately 10 h on a NVIDIA Tesla K80 GPU with 12 GB memory. After training, we generated a synthetic dataset of 1'588 images by sampling latent vectors from the same distribution as before and feeding them to the generator. The synthetic images were scaled to the range $[0, 1]$ to match the real images.

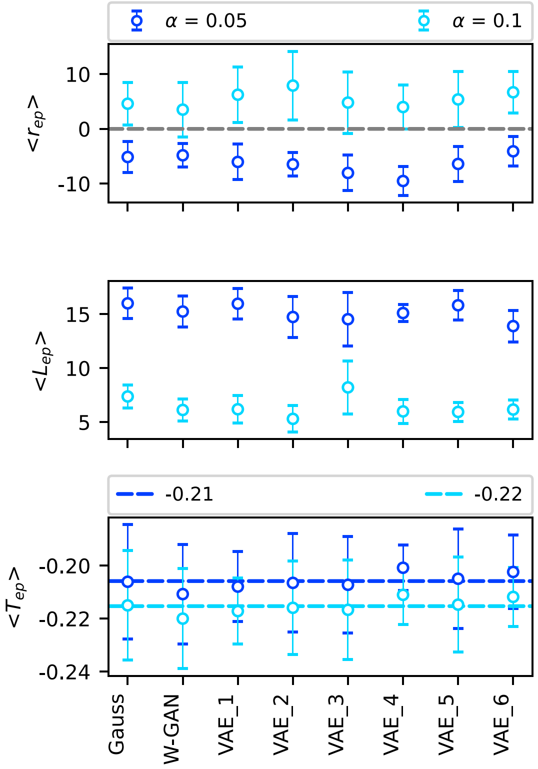


Figure 8. Comparison of six different β-VAE, W-GAN and β-VAE trained on fully fake data, using the synthetic beamline environment described in the previous sections. The top plots shows the average total reward per episode collected by a trained agent when deployed in realistic conditions. The middle plot shows the average episode length and the bottom one, the target reward used for the final training after reward dangling. In all these plots, the average is done over the validation test, assigning an additional positive reward of 10 if the agent managed to overcome the threshold. The error bars represent the standard deviation of the results taken over ten different trained agents.

All other VAEs were trained before and completely independent of the RL agents. A brief hyper-parameters optimisation was also done for some of the VAEs, mainly intended to find the optimal learning rate and the size of the different layers.

The eight datasets (six real data, two synthetic) were used to train different $\beta-$VAEs, from which the encoder circuits were used in the synthetic AWAKE model (figure 5) to compare the results.

Both sets of synthetic images are significantly different from the real data, but could be well reconstructed by the $\beta-$VAE trained on each dataset. Examples are shown in the Annex, figures 16(b) and (c). RL tests were made with each encoder on the synthetic AWAKE model, using the TD3 agent. It should be noted that the synthetic AWAKE model was based on data taken in 2018, i.e. using a different set of images to those used for the real image datasets described above.

A full comparison of the above described encoders, together with the classic explicit representation of the state space, is shown in figure 8, where essentially no difference can be seen on the choice of the encoder type. In the figure, the effect of α can be clearly seen, where all the trained agents with slightly larger α succeed to pass the final reward target, at the cost of a very slightly lower instantaneous reward.

A key finding from this study is that the reconstruction quality of the VAE has a negligible effect on the learning performance of RL agents. This finding has significant implications for reducing the training time and data requirements for the VAE, which are the major bottlenecks in the overall process (excluding the online RL agent training). A possible explanation for this finding is that the VAE only needs to encode the image into a lower-dimensional latent space that captures the salient features of the physical state.

4. Results on AWAKE beamline

4.1. Numerical optimisation tests

The usage of numerical optimisers has been explored and some of the main results already published in [23]. A large set of algorithms were tested and most of them showed successful results, although still needing a large number of iterations (larger than 100 in most cases) and ensuring that a global minimum search was in

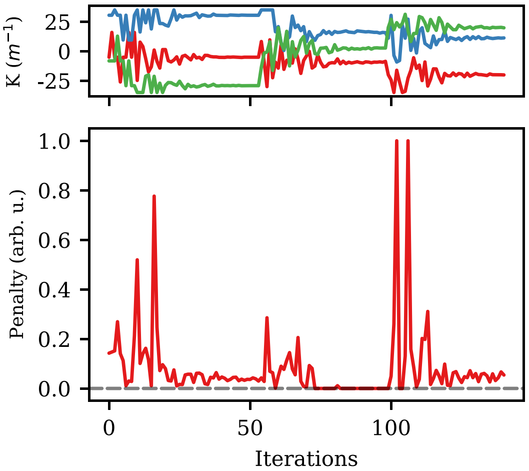


Figure 9. Optimisation result for BOBYQA algorithm using only the first three quadrupoles of the beamline. The global minimum is represented by the grey dashed line.

place. An example of a successful optimisation is shown in figure 9. In this example, the target beam size parameters were obtained using only 3 degrees of freedom, i.e. the initial quadrupole triplet, but very similar results were obtained also for 5, as detailed in [23]. All of the experiments succeeded when the beam size requested was indeed a global minimum for the line. If the requested size was larger, convergence was not achieved, as the only quadrupoles used in the optimisation procedure are the initial three and not those responsible for the final focus. The impact of the initial conditions is nevertheless very significant and cannot be neglected. As these are not stable, the optimisation procedure needs to be regularly run to obtain the design beam size at the BTV or plasma cell.

4.2. RL tests

For the RL tests the metrics used to measure the performance were the cumulative reward per episode $r_{\mathrm{ep}} = \sum_i^{\mathrm{ep}} r$, the length of episode L_{ep} in number of interactions, and the final weighted reward of the last iteration r_{ep}.

After extensive tuning of hyperparameters for different agents and the development of the reward dangling approach using the synthetic AWAKE model (every training was in the order of 2 min), tests were made with the real AWAKE machine. The parameters that needed the main number of evaluations were α_s, learning rate, and the penalty to assign in case of attempt to beyond the allowed Action space. The Stable Baselines RL agent 'TD3' worked reliably and learned the problem dynamics. The system was then tried with the implicit state extraction, using different versions of encoder trained on the various real and synthetic datasets described above. The RL agents also converged in a similar time to the explicit state versions, showing that the RL state can be successfully auto-encoded in an unsupervised manner, as shown in figures 10 and 11.

Equally importantly, for applications where sample efficiency of the overall method is important, we demonstrated that encoders trained with fully or partially (W-GAN) synthetic data were also effective for state encoding. This is illustrated in figure 10 and compared with fully explicit state description and with an implicit one but trained on real data. The training time is rather similar in the terms of total machine interactions, which is about 600 iterations for the first stage where the target is adapted to the agent performance and about 450 for the actual agent training, which adds up to about 20 h of beam time. In some cases, training was ended after a total of 600 machine calls to reduce the time needed, if the agent had converged to its target performance. The episode length reached after training for all three different ways of encoding the states is rather similar and less than 10 machine iterations in all cases.

Another metric to show the evolution of the training (after choosing the target reward with the *reward dangling* algorithm) of the TD3 agent is $\Delta r_{\mathrm{ep}} \equiv r_e - r_0$, which gives a magnitude of the improvement on the environment made by the agent after a reset. In figure 12(a) it is clear that after 10 episodes all the agents manage to always improve the performance of the beamline.

The performance of the three different agents trained on the real beamline are summarised in figure 12(b). For each agent, the delta reward between start and end of the episode is plotted as a function of the episode number used to perform the validation. In this situation the agents trained are free to operate on the beamline in the context of an episode after the random reset of the actions.

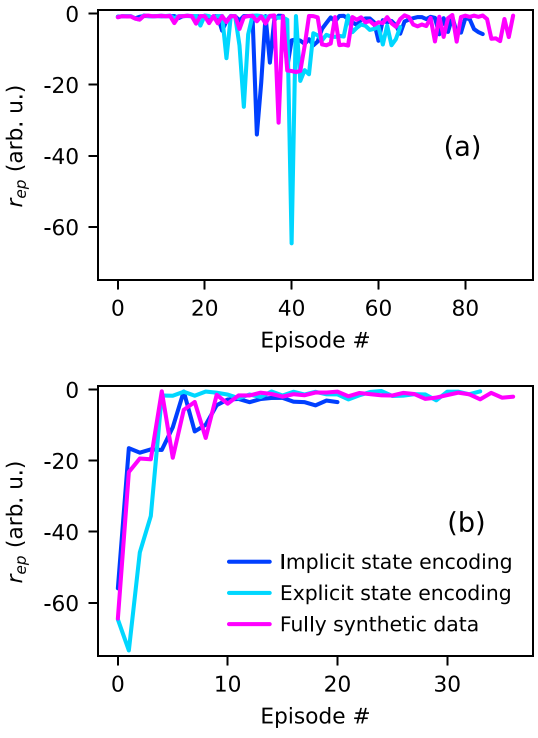


Figure 10. Reward per episode for two-step training process for TD3 RL agent on real AWAKE machine. (a) Initial stage of *reward dangling* and (b) actual training of TD3 agent with the maximum reward target found. The training were performed using three different state representation: implicit state encoding with VAE trained on real images, explicit state encoding and implicit state encoding using fully synthetic data for VAE training.

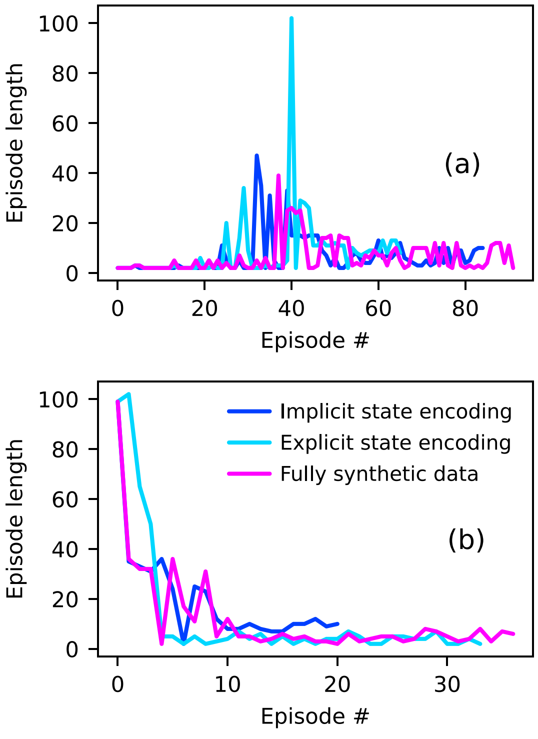


Figure 11. Episode length for two-step training process for TD3 RL agent on real AWAKE machine. (a) Initial stage of *reward dangling* and (b) actual training of TD3 agent with the maximum reward target found, as for figure 10.

The final beam spot obtained was of very good quality, as shown in the projections plotted in figure 13, when using implicit state encoding from an VAE trained on synthetic data.

The main problem encountered with the RL approach was the longer-term stability of the beamline. Although the trained agent performed well if tested during a few hours of the training, the results were less

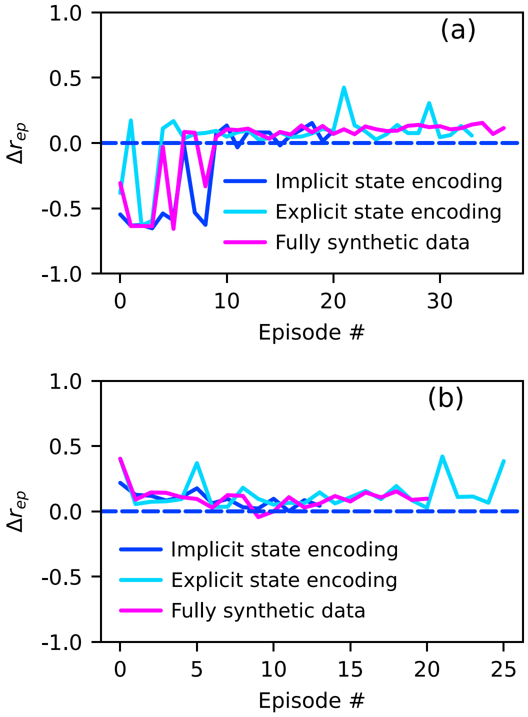


Figure 12. Evolution of $\Delta r_{\rm ep}$ during training (a) and validation (b) for three different state encoding.

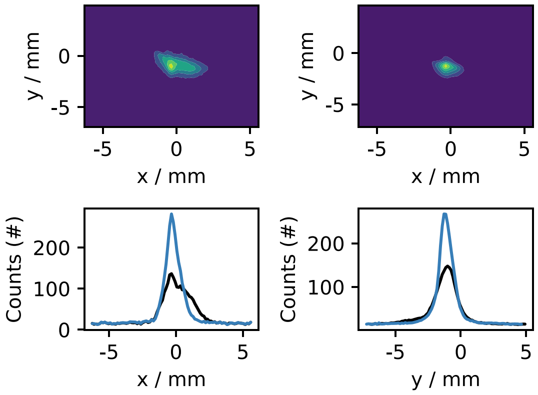


Figure 13. Beam size measurements for one episode after training of TD3 agent using implicit state description and VAE trained on synthetic data. (Top-left) Measurements at the start of the episode after randomly reset the action settings. (Top-right) BTV image after the agent has acted is shown. (Below) Horizontal (black) and vertical (blue) projections of the images above.

good when tested some weeks or months later. This is shown in figure 14, where different versions of state encoding were tested. In these experiments, the maximum number of iterations in the environment was fixed to 100. In some cases, the agent trained could still solve the control problem but in a larger number of steps. In others, the trained agent failed to converge, indicating that the problem dynamics had shifted outside of the valid training data space. There was no obvious time correlation with the performance of the agents but more with the routine tasks performed. The contributing factors are likely to be the ‘hidden’ action variables which change the beam spot distribution and hence the encoded state—this is not surprising given the adjustments made to the source including the laser power, alignment, synchronisation and RF phasing which are all empirically adjusted before a new run, or indeed at the start of each day during a running period.

Due to lack of availability of the beamline, no statistically significant experiments could be performed to assess the resilience to hidden variables of the different state encoding techniques, but from first observations the fully synthetic state encoding seems to be the less resilient to these response changes.

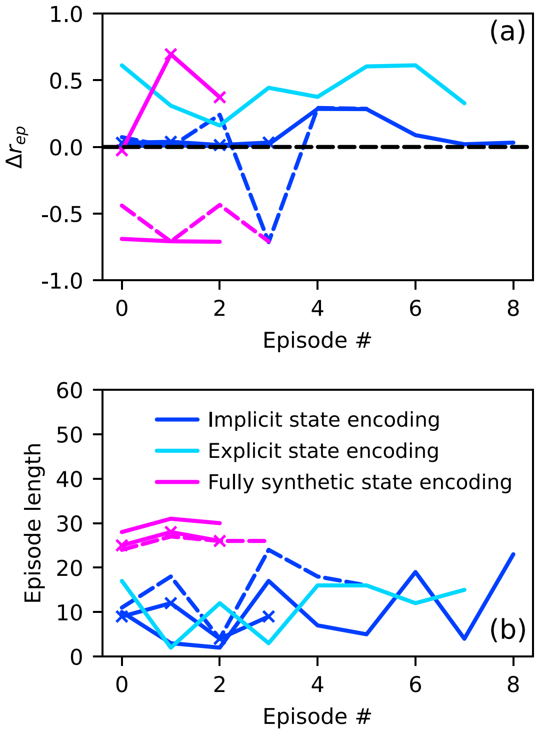


Figure 14. Evolution of Δr_{ep} (a) and episode length (b) during validation for three different state encoding and different moments in time (indicated in same colour but with different line style). It is clear that the performance of the agents are far from what obtained at training time, mainly in the time needed to re-establish the target performance of the line.

5. Discussion

5.1. Usage of RL agents in operation

The training of RL agents using implicit or explicit state encoding proved to be successful regardless on the encoding chosen and on the initial state of the system—it was in fact possible to train a large number of RL agents on different days, where the source states (unknown to the agents) were changing either on purpose or randomly.

The main problem though, was that trained agents were very difficult to reuse days after their training, showing very poor performance. We believe that this behaviour is to be attributed to the unknown states of the source and basically to the change of the mapping between actors, states and reward that follows. A clear solution would be to include more state dimensions describing the source and the possible parameter configuration, but unfortunately the beam time available was not sufficient to test also this state parameterisation.

These results are not suitable for operational deployment of the trained RL agents, but clearly showed their potential. For this reason, it was decided to rely on numerical optimisers to provide the required beam quality to the experiment and continue the studies on the usage of RL agents with a larger state description. Also, profiting for the insight proposed in [34], the possibility to use physics informed Gaussian process to speed up the numerical optimisation is under investigation.

One way to address the temporal variation of the learned response is to add a temporal dimension to the dataset, e.g. converting the data into time-series and then use a recurrent neural network instead of a fully connected one. This would require the ability to train the RL agent on data that capture the change of this latent temporal dimension. This approach would be similar to the one presented in [35], but with a recurrent neural network as the model and an RL-based policy instead of a Bayesian optimization one.

5.2. Performance reach

Numerical optimisers could take up to a few hundred of iterations to achieve a suitable beam configuration for the experiment, strongly depending on the initial conditions of the optimisation. RL agents, instead, could perform this task in just a handful of action steps, if the mapping at training time is preserved. Clearly this would mean a huge improvement in the usage of machine time, although work it is still needed to include the hidden states that are causing the variation of the mapping in time.

The full process is done today all manually and using linear optics approximations to treat the line initial condition changes. Already the deployed environment and the tested numerical optimisers would provide a speed up in setting up time and possibly more reproducible conditions for the experiment. Studies are still ongoing to fully deploy operationally this method.

A possible extension to the manual optimiser launching is to use model-free adaptive feedback systems, like extremum seeking [36], to maintain the optimal value found via numerical optimisation even after drift of the settings. This is of course only possible in case the drift are slow with respect to the probing frequency.

Moreover, a possible expansion of the synthetic model to capture the time-varying response and additional data to improve the VAE models would allow for a possible model-based RL, giving the possibility to update the surrogate with the new data points obtained during the deployment of the real line.

5.3. Applicability to other systems and outlook

The tools and methodologies presented in this paper are rather generic and the application to the AWAKE transfer line case could be seen as a first proof-of-principle. The unsupervised state encoding via auto-encoders could be used in many other systems and domains of accelerators. Also, the methodology presented to assess the optimal reward threshold for RL agents training can be equally applied to any other RL training case were the episode termination threshold strongly drives the training speed and success rate. For example, this is under study for the training of RL agents to automatically extract information from Schottky spectra in the CERN Low Energy Ion Ring [37].

The full methodology as described could also have other applications in transfer lines with very sensitive final focus. Depending on the available instrumentation, the same basic methodology, but, for instance, using multiple screens for more accurate optics estimation, could be envisaged. The technique presented in this paper could be applied almost entirely, basically changing only the source of image.

Also, the concept of a synthetic model developed as a tool to fully replace a real accelerator can directly be applied to many other machines and facilities. The benefit of having a full offline surrogate of the real machine significantly simplified the tuning of the RL agents and comparison of other techniques.

Looking at further development, the access to multiple beam observations could open the way to a full phase-space reconstruction using lower projection needed than classic tomography. This would allow a more detailed state description and hence a much more robust system to unknown variables and drifts.

6. Conclusions

The feasibility of automatic optimisation of the AWAKE e^- beamline brightness based on ML techniques has been successfully explored using a variety of approaches. The performance of suitable numerical optimisers was demonstrated, despite the non-convex nature of the overall problem. A number of other useful techniques have been developed and tested, including successful RL agent training, a generative synthetic model using β-VAE for offline testing and hyperparameter optimisation, the use of implicit unsupervised RL state encoding with a β-VAE encoder based on computer vision, the training of these encoder networks with fully synthetic data and the development of automatic reward target value setting for RL episode termination.

The limitations of RL were also reached for this specific configuration, where the variations from the e^- source meant that the trained RL agents could not reproducibly be deployed over long time scales. The work showed the importance of capturing all the generative factors in the observation of the RL state space, as well as including all the corresponding actions which are used to correct the performance. Given the high repetition rate of the AWAKE e^- beamline, the simpler optimisation approach is the one which will be deployed operationally. Nevertheless, the RL paradigm with a learned response remains relevant for applications like injection into LHC, where each sample is much more expensive to take, and the repetition rate is orders of magnitude lower.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgment

The stimulating real-world environment provided by the CERN ‘Machine Learning Coffee’ forum [38] proved invaluable for presenting and discussing progress and results. We also benefited greatly from the dedication of the entire AWAKE project team, the CERN Equipment groups and the SPS operations team.

Particular thanks go to E Gschwendtner for her continued enthusiastic support of these activities, as well as to L Verra and S Dobert for keeping the beamline running when we needed it.

Appendix A. Synthetic images for different $\mathbb{Z}$ dimensionalities

Random samples of real, VAE recovered and fully synthetic 128 × 128 pixel images for different Z dimensionalities. The images in the top rows are the original BTV measurements, those in the lower rows have been generated through the full synthetic AWAKE model from the labelled action values, using the predictor and decoder networks.

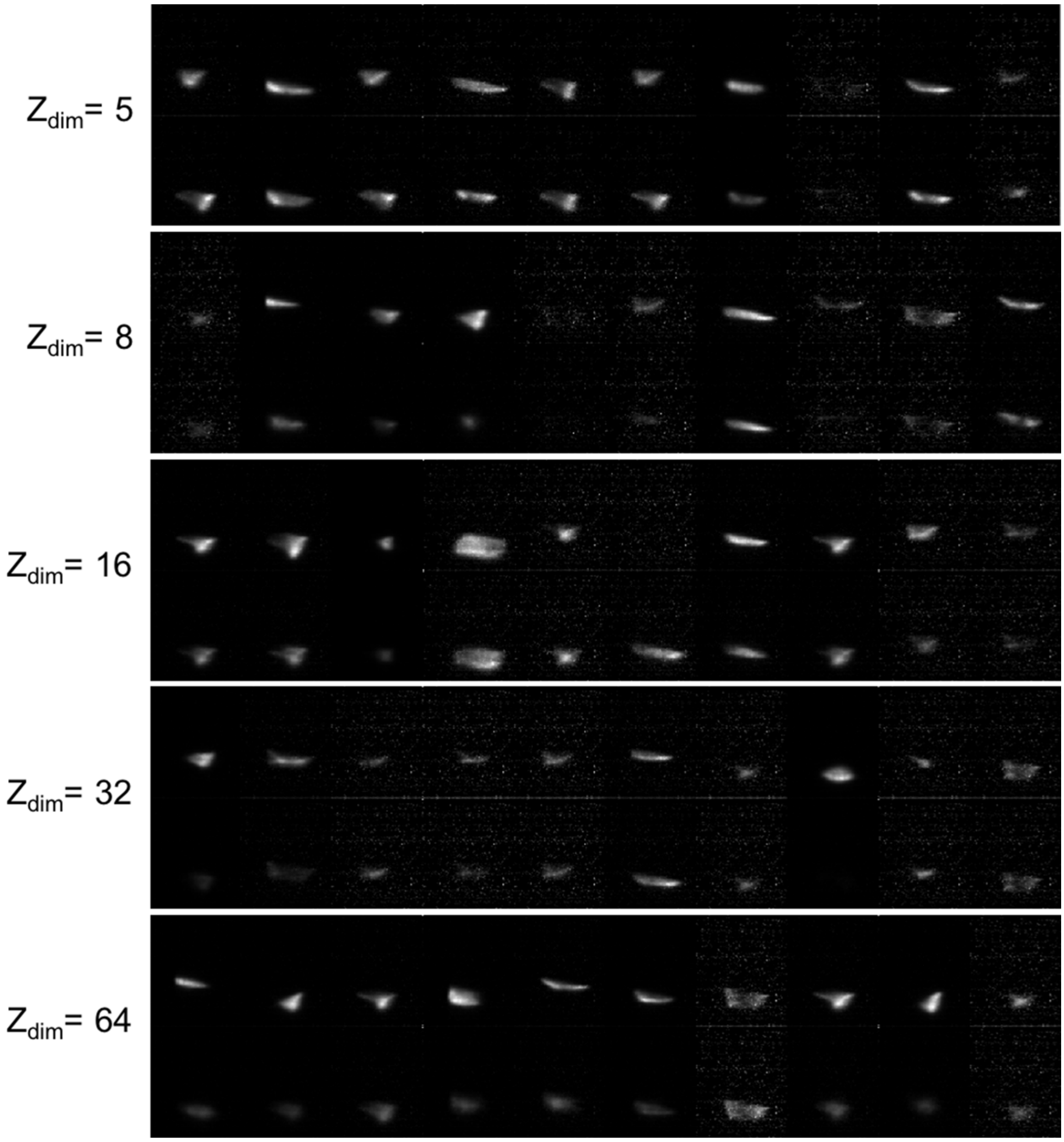
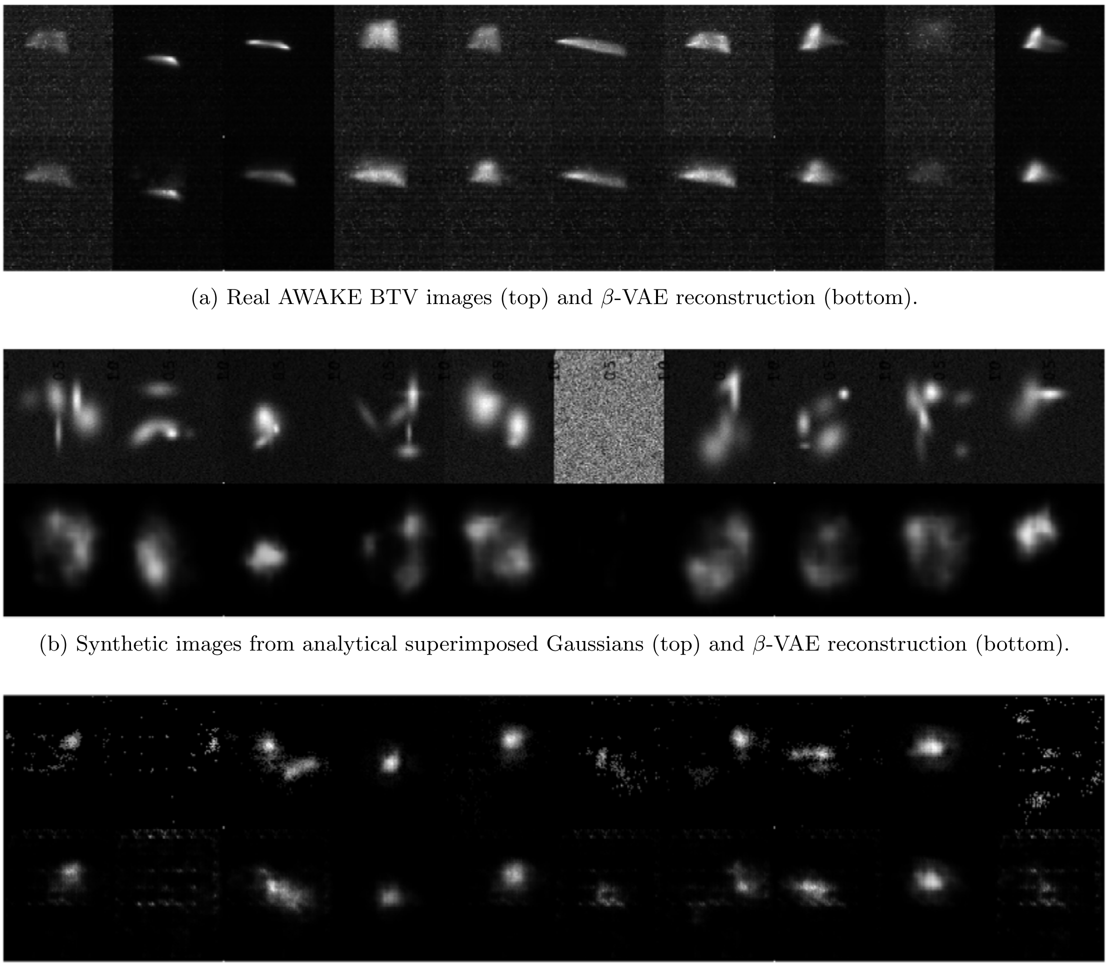


Figure 15. Example of real (top rows) and images produced by the VAE (bottom rows) for different dimension of the latent space.

Appendix B. Real and synthetic images for VAE state encoder training

Random samples of 128 × 128 pixel images used to train VAE for image encoding (top rows) and recovered VAE images (bottom rows) for the real and two synthetic datasets.



(a) Real AWAKE BTV images (top) and β-VAE reconstruction (bottom).

(b) Synthetic images from analytical superimposed Gaussians (top) and β-VAE reconstruction (bottom).

(c) Synthetic images from Wasserstein GP-GAN (top) and β-VAE reconstruction (bottom).

Figure 16. Example of real images, analytically and W-GAN produced images are shown on top rows. Examples of images reconstructed with the VAE are instead shown on bottom rows.

ORCID iD

Francesco Maria Velotti https://orcid.org/0000-0001-7815-6011

References

[1] AWAKE collaboration 2013 AWAKE design report: a proton-driven plasma wakefield acceleration experiment at CERN, *Technical Report* CERN-SPSC-2013-013, SPSC-TDR-003 (Geneva: CERN)

[2] AWAKE collaboration 2018 Acceleration of electrons in the plasma wakefield of a proton bunch *Nature* **561** 363

[3] Bracco C *et al* 2019 Systematic optics studies for the commissioning of the awake electron beamline *10th Int. Partile Accelerator Conf. IPAC2019 (Melbourne, Australia)*

[4] Gavalda X N Multi-Objective genetic based algorithms and experimental beam lifetime studies for the synchrotron SOLEIL storage ring *PhD Thesis* Université Paris-Saclay (available at: https://tel.archives-ouvertes.fr/tel-01385576)

[5] Kirschner J *et al* 2019 Bayesian optimisation for fast and safe parameter tuning of SWISSFEL *39th Free Electron Laser Conf. (FEL2019) (Hamburg, Germany)*

[6] Fuchsberger K 2011 Novel concepts for optimization of the CERN large hadron collider injection lines *Dissertation* Technische Universität Wien

[7] Duris J *et al* 2020 Bayesian Optimization of a free-electron laser *Phys. Rev. Lett.* **124** 124801

[8] Edelen A L, Biedron S G, Chase B E, Edstrom D, Milton S V and Stabile P 2016 Neural networks for modeling and control of particle accelerators *IEEE Trans. Nucl. Sci.* **63** 2

[9] Edelen A *et al* 2016 Neural network model of the PXIE RFQ cooling system and resonant frequency response *7th Int. Particle Accelerator Conf. (IPAC)*

[10] Edelen A *et al* 2018 First steps toward incorporating image based diagnostics into particle accelerator control systems using convolutional neural networks *2nd North American Particle Accelerator Conf. (NAPAC)*

[11] Scheinker A, Edelen A, Bohler D, Emma C and Lutman A 2018 Demonstration of Model-independent control of the longitudinal phase space of electron beams in the linac-coherent light source with femtosecond resolution *Phys. Rev. Lett.* **121** 044801

[12] Xie Z *et al* 2018 Feedback control for Cassie with deep reinforcement learning *2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)* pp 1241–6

[13] Akkaya I *et al* 2019 Solving Rubik's cube with a robot hand (arXiv:1910.07113)
[14] Bellemare M G, Candido S, Castro P S, Gong J, Machado M C, Moitra S, Ponda S S and Wang Z 2020 Autonomous navigation of stratospheric balloons using reinforcement learning *Nature* **588** 77–82
[15] Degrave J *et al* 2022 Magnetic control of tokamak plasmas through deep reinforcement learning *Nature* **602** 414–9
[16] Hirlaender S and Burchon N 2020 Model-free and bayesian ensembling model-based deep reinforcement learning for particle accelerator control demonstrated on the FERMI FEL (arXiv:2012.09737)
[17] Ogren J, Gohil C and Schulte D 2021 Surrogate modeling of the CLIC final-focus system using artificial neural networks *J. Instrum.* **16** 05012
[18] Edelen A, Neveu N, Frey M, Huber Y, Mayes C and Adelmann A 2020 Machine learning for orders of magnitude speedup in multiobjective optimization of particle accelerator systems *Phys. Rev. Accel. Beams* **23** 044601
[19] Kranjcevic M , Riemann B, Adelmann A and Streun A 2021 Multiobjective optimization of the dynamic aperture using surrogate models based on artificial neural networks *Phys. Rev. Accel. Beams* **24** 014601
[20] Edelen A *et al* 2018 Neural network virtual diagnostic for the FAST low energy beam line *Proc. IPAC 2018* (Vancouver: JACoW) p WEAF040
[21] Edelen A *et al* 2018 Neural network based approaches to the modeling and control of particle accelerators *Proc. of IPAC 2018* (Vancouver: JACoW) p THYGBE2
[22] Kain V Hirlander S, Goddard B, Velotti F M, Della Porta G Z, Bruchon N and Valentino G 2020 Sample-efficient reinforcement learning for CERN accelerator control *Phys. Rev. Accel. Beams* **23** 124801
[23] Scheinker A , Hirlaender S, Velotti F M, Gessner S, Della Porta G Z, Kain V, Goddard B and Ramjiawan R 2020 Online multi-objective particle accelerator optimization of the AWAKE electron beam line for simultaneous emittance and orbit control *AIP Adv.* **10** 055320
[24] Burger S *et al* A new control system for the CERN TV beams observation (available at: https://cds.cern.ch/record/1123677)
[25] Brockman G *et al* 2016 OpenAI Gym (arXiv:1606.01540)
[26] Pepitone K *et al* 2018 The electron accelerators for the AWAKE experiment at CERN—baseline and future developments *Nucl. Instrum. Methods Phys. Res.* A **909** 102–6
[27] CERN python module *pyjapc* (available at: https://gitlab.cern.ch/scripting-tools/pyjapc)
[28] Hill A *et al* 2018 Stable baselines, github repository (available at: https://github.com/hill-a/stable-baselines)
[29] Powell M J D 2009 The BOBYQA algorithm for bound constrained optimization without derivatives *Technical Report* NA2009/06 (Cambridge: Department of Applied Mathematics and Theoretical Physics)
[30] Fujimoto S *et al* 2018 Addressing function approximation error in actor-critic methods (arXiv:1802.09477v3 [cs.AI])
[31] Kingma D and Welling M 2014 Auto-encoding variational bayes (arXiv:1312.6114v10 [stat.ML])
[32] Burgess C P 2018 Understanding disentangling in β-VAE (arXiv:1804.03599v1 [stat.ML])
[33] Gulrajani I *et al* 2017 Improved training of wasserstein GANs (arXiv:1704.00028v3 [stat.ML])
[34] Hanuka A Huang X, Shtalenkova J, Kennedy D, Edelen A, Zhang Z, Lalchand V R, Ratner D and Duris J 2021 Physics model-informed Gaussian process for online optimization of particle accelerators *Phys. Rev. Accel. Beams* **24** 072802
[35] Kuklev N *et al* 2022 Online accelerator tuning with adaptive bayesian optimization *5th North American Particle Accelerator Conf. (NAPAC2022) (Albuquerque, NM, USA)*
[36] Hamza M 1966 Extremum control of continuous systems *IEEE Trans. Autom. Control* **11** 182–9
[37] Madysa N *et al* Automated intensity optimisation using reinforcement learning at LEIR *IPAC'22*
[38] (Available at: https://indico.cern.ch/category/14287/