Contents lists available at ScienceDirect

Nuclear Inst. and Methods in Physics Research, A

journal homepage: www.elsevier.com/locate/nima

Full Length Article

Accelerator tuning method using autoencoder and Bayesian optimization

Yasuyuki Morita [a,*], Takashi Washio [b], Yuta Nakashima [c]

[a] RIKEN Nishina Center for Accelerator-Based Science, 2-1 Hirosawa, Wako, Saitama, 351-0198, Japan
[b] The Institute of Scientific and Industrial Research, Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, 567-0047, Japan
[c] Institute for Datability Science, Osaka University, Techno-alliance bldg. C503, 2-8, Yamadaoka, Suita, Osaka, 565-0871, Japan

ARTICLE INFO

Keywords:
Accelerator
Machine learning
Bayesian optimization
Autoencoder

ABSTRACT

During beam-accelerator operation, a large number of parameters need to be tuned. In recent years, tuning methods based on machine learning have been extensively studied. Bayesian optimization (BO) has attracted considerable attention as an excellent method for accelerator tuning. However, its applicability is limited by the number of parameters that can be tuned. In this study, we propose an optimization method that combines autoencoder and BO to tune a large number of parameters. We verified it using beam transport simulations. We confirmed a higher tuning effect in a shorter time than when using only BO. The proposed method is expected to speed up the accelerator operation and provide comprehensive tuning.

1. Introduction

In recent years, considerable activity has been devoted to the development of beam-accelerator operation control methods using machine learning [1,2], particularly Bayesian optimization (BO) [3]. BO is expected to improve the beam transmission of the beamline and beam intensity of the ion source in a short time. However, the number of parameters that BO can handle is limited. As the number of parameters increases, the time required to achieve optimization increases. To address this issue, some methods to reduce the number of parameters based on physical simulation models in advance have been developed [4]; however, this is not easy to achieve in accelerator physics, where the results may not be in perfect agreement with the simulation. Therefore, the development of BO-based control methods has been limited to partial optimization of accelerators. To overcome this problem, we propose a new method that uses autoencoders (AEs) [5]. In a study conducted by Dr. Iwasaki at KEK [6], variational AE (VAE) [7] was used to visualize the accelerator conditions. However, in the present study, AE was used. This is because the objective is not to classify in the latent-variable space as in VAE. Instead, preventing search omissions and avoiding local maxima are easy if the variables are not divided too finely for optimization within the latent-variable space. Dimensionality reduction using AE and optimization by BO within the latent-variable space are expected to efficiently optimize a large number of parameters in a short time. The proposed method is expected to reduce tuning time.

2. Method

In this study, we investigated the impact of dimensionality reduction using an AE and Bayesian optimization (BO) in the latent-variable space. In this method, first, AE was constructed using the accelerator operating parameters as the input. A neural network (NN) was trained to reproduce the operating parameters of the input accelerator at the output. The latent variable $\mathbf{Z}$ is extracted from the middle layer of the NN. In this method, the dimensions of $\mathbf{Z}$ are set to be smaller than the dimensions of the input accelerator operating parameters. $\mathbf{Z}$ was set by determining the smallest dimension that could accurately reproduce the input values in the output layer. Subsequently, we made a copy of the decoder part of AE that recovers the input data from the latent-variable value, checks the recovery accuracy after training, and saves the model. This model was used to recover and optimize the accelerator operating parameters in the latent-variable space. BO was used to search in the latent-variable space, and the decoder was used to reproduce the accelerator operating parameters corresponding to the point in the latent space searched by BO. By applying this method, the number of parameters that need be handled can be reduced to fewer than the actual accelerator operating parameters. This enables the efficient optimization of a large number of parameters.

3. Beam transport simulation

Beam transport simulations were performed to verify the effectiveness of this method.

3.1. Simulation condition

This method was validated using simulations assuming a WSS beam transport line at the Research Center for Nuclear Physics (RCNP), Osaka

[*] Corresponding author.
E-mail address: yasuyuki.morita@riken.jp (Y. Morita).

https://doi.org/10.1016/j.nima.2023.168730
Received 2 June 2023; Received in revised form 11 September 2023; Accepted 25 September 2023
Available online 28 September 2023

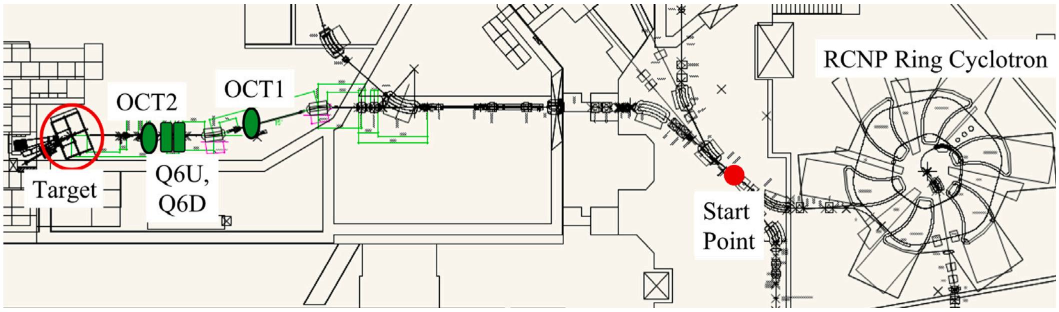

Fig. 1. Schematic of the beamline used in the simulation.

University. The nonlinear magnetic field of the octupole magnets was used to deform the beam shape from a Gaussian distribution to a "hollow beam", [8] which has high beam intensity at the "edge" of beam distribution on the target. Although simulation calculations are possible for hollow beams, the size and shape of the target beam have not been analytically solved. The only way to obtain the desired shape and size is to change the K-value of the magnet based on the simulation results and repeat the calculation until these objectives are achieved. Moreover, multiple combinations of K-values can result in a similar shape. This is similar to the trial-and-error process of actual accelerator operation. Therefore, in this study, a hollow beam simulation was used for verification. This beamline irradiates the muon-producing target of the DC muon source ”MuSIC” [9]; the target irradiation with a hollow beam was the focus of this investigation. The first focal point of the beam extracted from the RCNP ring cyclotron was considered as the initial position for the beam-transport simulation. The RMS beam radius and emittance at the initial position were 0.5 mm and 0.5 π mm mrad, respectively. The ion species used was a proton beam with 392 MeV energy. The simulation was performed by solving the equation of motion for a single particle, and four million particles were generated according to a Gaussian distribution as the initial distribution. This calculation used the same simulator [10] as used in the Takasaki Advanced Radiation Research Institute of the Quantum Science and Technology Agency (QST). Moreover, it is reported that the results are sufficiently correct by comparative experiments with actual experimental results at the QST [10]. In this study, the simulation was applied to the RCNP WSS course, and the results are discussed. The WSS course with octupole electromagnets and the positions of the electromagnets to be adjusted are shown in Fig. 1. The rms envelope of the beam from the starting point to OCT1 is shown in Fig. 2. A total of eight quadrupole magnets and two dipole magnets on the beamline existed where this envelope calculation was performed, all of which were treated as fixed values in this study.

3.2. Optimization parameters

First, K values for each magnet were randomly set within the ranges shown in Table 1, and beam-transport simulations were used to generate data for AE training. Although the data were created by simulation, we plan to use past operation data to for AE learning when using the data in actual beamlines. Four magnets were used as the optimization parameters: two octupole magnets (upstream: OCT1, downstream: OCT2) and two quadrupole magnets between the octupole (upstream: Q6U, downstream: Q6D). By changing the K value of these magnets, the shape and size of the beam irradiating the target can be changed.

The optimization was judged as good or bad based on the amount of the beams irradiating at the edges of the target. Because the muon production target was cylindrical with a radius of 20 mm, we tuned it to maximize the beam ratio (R) within 17.5 mm < r < 20.0 mm to the total beam intensity. Here, r represents the radial position of the beam.

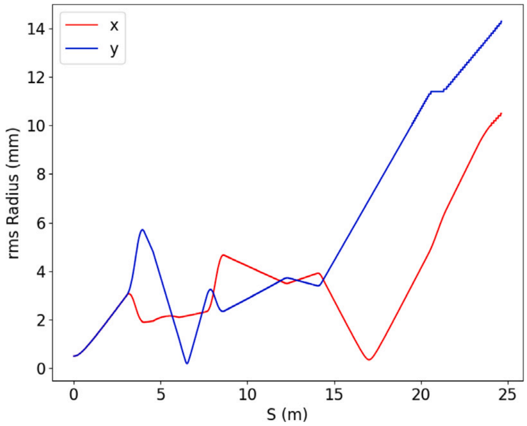

Fig. 2. The rms envelope of the beam from the starting point to OCT1.

Table 1
Range of setting K values for each magnet for the test data.

Magnet	Minimum	Maximum	Step width
OCT1	-7000 (m^{-4})	0 (m^{-4})	25 (m^{-4})
OCT2	0 (m^{-4})	7000 (m^{-4})	25 (m^{-4})
Q6U	-1 (m^{-2})	0 (m^{-2})	0.005 (m^{-2})
Q6D	0 (m^{-2})	1 (m^{-2})	0.005 (m^{-2})

4. Machine learning for optimization

The effectiveness of the optimization method using AE and BO was verified via hollow beam simulations. The first step involves learning the AE. AE was built using TensorFlow [11], which is a Python library.

4.1. Model of autoencoder

We prepared AE to represent the latent-variable space and a decoder for tuning to recover the K values from the latent variables. The encoder part of the AE has four input values, six intermediate layers, and two-dimensional output as the latent-variable space. The activation functions for the intermediate layers were all ReLU functions, and for the layer outputting the latent-variable space, the activation functions were hyperbolic tangents. The input values were trained by normalizing the original K values, which range from zero to one. Normalization of the K value for each magnet was performed as shown in Eq. (1).

$$\begin{aligned} Input_{Octupole} &= \frac{K_{Octupole}+7000}{14000} \\ Input_{Quadrupole} &= \frac{K_{Quadrupole}+1}{2} \end{aligned} \tag{1}$$

For the decoder portion, six intermediate layers were used with the encoder portion and ReLU function. In addition to the decoder, an intermediate layer using the six-layer ReLU function was separately

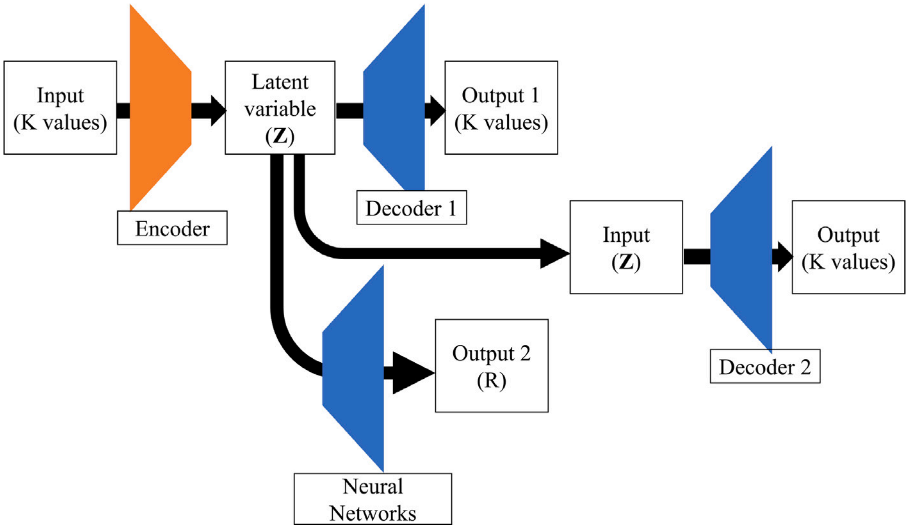

Fig. 3. Schematic of the AE model. In addition to the regular AE, a neural network (NN) that separately predicts R from the latent-variable space was used, and a decoder was used for tuning magnets.

incorporated. Multiple models were created for the number of layers, and the one with the smallest loss function was selected. The ReLU function is expressed in Eq. (2).

$$f(x) = \begin{cases} 1 & (x \geq 0) \\ 0 & (x < 0) \end{cases} \tag{2}$$

This was used to predict the value of R from the latent variables. The aforementioned process limits the latent variable such that it is not completely unrelated to the value of R, and we believed that the value of R could also be predicted. Therefore, as AE, we built a model with encoder and decoder 1 (OutPut1) to recover the K value of the electromagnet, and a neural network (OutPut2) to predict R. In this case, we have built a model that can also predict R. This is intended for use in another analysis of the correlation between K-values and R. Therefore, Output2 was not used for the optimization in this study. In addition to the AE model, we prepared decoder2 that outputs the set values of the electromagnets using the latent-variable space as the input for tuning. Because the previous model estimates R in the decoder part, we prepared another model for tuning without predicting R.

This model was constructed with four intermediate layers using the ReLU function, with a two-dimensional latent-variable space as the input layer and four-dimensional set values of the electromagnets as the output layer. Schematics of the AE and decoder models for tuning are shown in Fig. 3.

4.2. Learning of autoencoder

Because several data points were required for learning, the data were created by randomly setting the K values from the range of settings presented in Table 1. In total, 20,000 data points were created for training and divided into 18,000 training- and 2000 test-data points.

However, no correlation was observed because the K values were randomly determined. Therefore, only the results with a beam intensity ratio of 0.15 or higher within the target specified range were used for training. The number of data points satisfying this condition was 575 for the training data and 55 for the test data.

First, the results of the AE study are shown in Fig. 4. The horizontal axis represents the number of epochs, and the vertical axis represents the loss function. The mean squared error (MSE) was used to evaluate the model. MSE is expressed in Eq. (3).

$$MSE = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2, \tag{3}$$

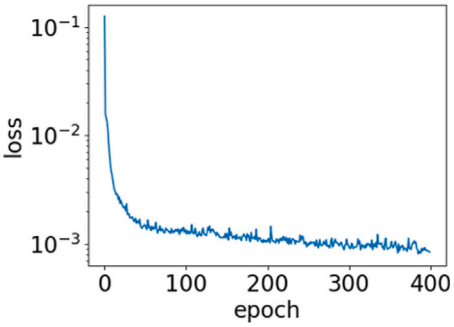

Fig. 4. Results of leaning AE. The horizontal axis is the number of epochs and vertical axis is the loss function (MSE).

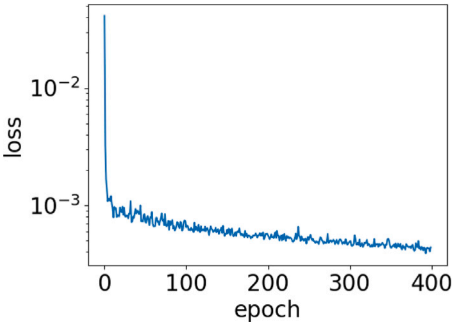

Fig. 5. Results of learning the decoder model for tuning. The horizontal axis is the number of epochs and the vertical axis is the loss function (MSE).

where $\hat{y}_i$ is the input value that is the correct answer, y_i is the output value of AE, and n is the total number of data. These results indicated that the loss function after 400 epochs of learning was sufficiently small and the learning proceeded well.

Subsequently, the decoder model is trained for tuning. The first step in the learning procedure is to input the set values of the electromagnets to the AE learned earlier and to find the latent-variable values. Next, the latent variables were used as the input, and the electromagnet setting values input to the AE were used as answers. The results are presented in Fig. 5.

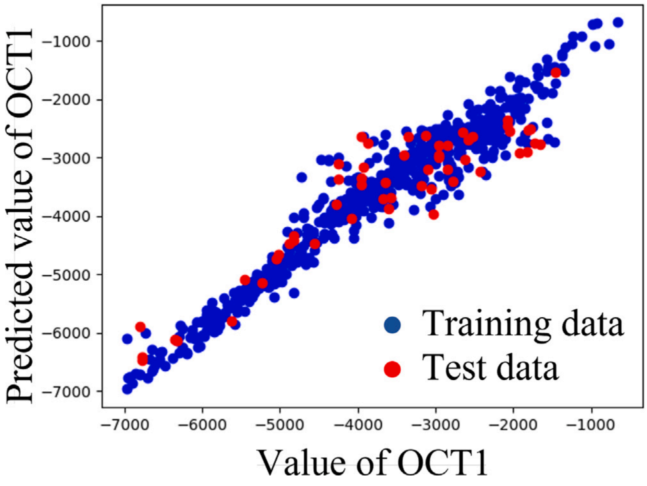

Fig. 6. Relationship between input electromagnet settings and those restored by the learned tuning decoder for OCT1.

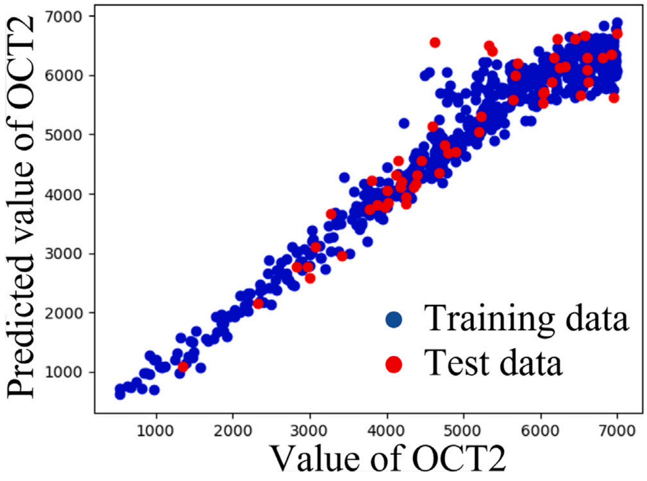

Fig. 7. Relationship between input electromagnet settings and those restored by the learned tuning decoder for OCT2.

Table 2
RMS value of the absolute error of the output value of AE for each magnet.

OCT1	OCT2	Q6U	Q6D
241	255	0.017	0.021

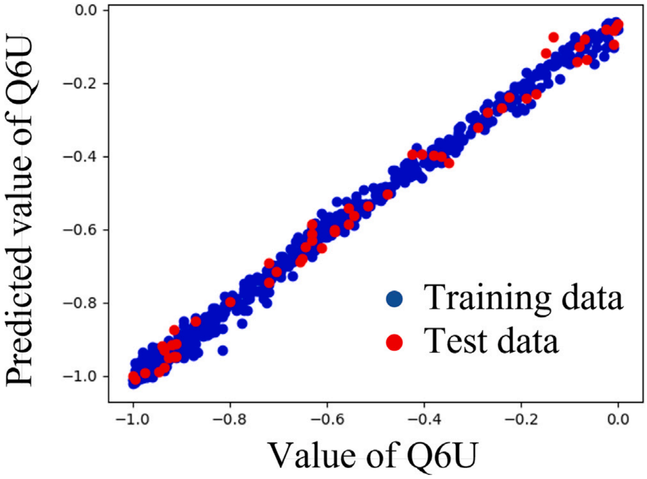

Fig. 8. Relationship between input electromagnet settings and those restored by the learned tuning decoder for Q6U.

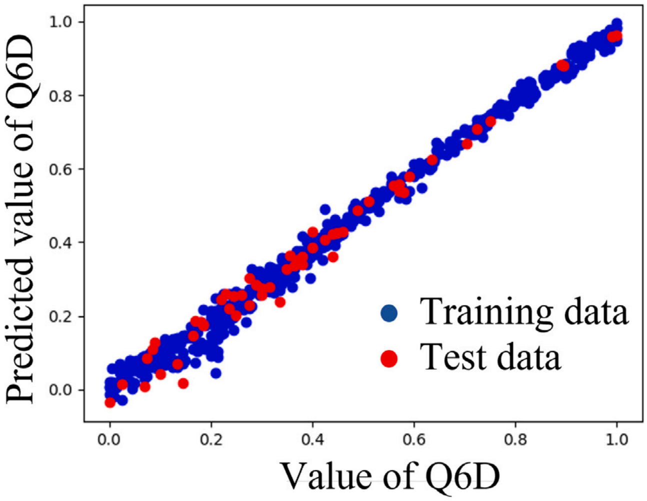

Fig. 9. Relationship between input electromagnet settings and those restored by the learned tuning decoder for Q6D.

In addition, the loss function was sufficiently small at 400 epochs, confirming that learning had proceeded. The relationships between the input electromagnet settings and those restored by the learned tuning decoder are shown in Figs. 6,7,8,9. The blue dots in the figure represent the results of the training data, and the red dots represent the results of the test data. The results indicate that the values were close to the input values although they were not the same as the restored values.

The RMS value of the absolute error of the output value of AE for each magnet is listed in Table 2. This error is approximately a 10% change in R from 0.33 to 0.30, which is sufficiently acceptable. Because dimensionality reduction was performed on the two-dimensional latent-variable space, each latent variable was used as the horizontal and vertical axes. The z-axis (color bar) represents the value of R. The output layer of the latent variable with the hyperbolic tangent as the activation function confirms that the values fall between −1 and 1. The time required for training after data preparation is a few minutes. In this case, the data were prepared by simulation; however, when applying the system to actual equipment, it is not necessary to prepare new data if past operation data can be utilized.

4.3. Bayesian optimization

Fig. 10 demonstrates how the training and test data are distributed in the latent-variable space.

BO was performed using two parameters in the latent-variable space as variables to maximize R. GPyOpt [12], a library for Python, was used to perform BO. Matern 5/2 [13], the default value of GPyOpt, was used as the kernel function in this verification. The acquisition function is the upper confidence bound (UCB), defined by 4σ as the effective area. The tuning cycle was repeated 200 times: eight times for the initial data and 192 times for the optimization. The design was constructed to terminate the optimization by reverting to the parameter with the largest R.

As shown in Fig. 10, the latent variables did not utilize the entire range from −1 to 1 in the current training data. If AE utilizes an unlearned range, the predicted electromagnet setpoints may not be appropriate. Therefore, we limited the range and performed optimization. In Fig. 10, the latent variables are distributed within the range listed in Table 3. Based on this result, the possible setting ranges of the

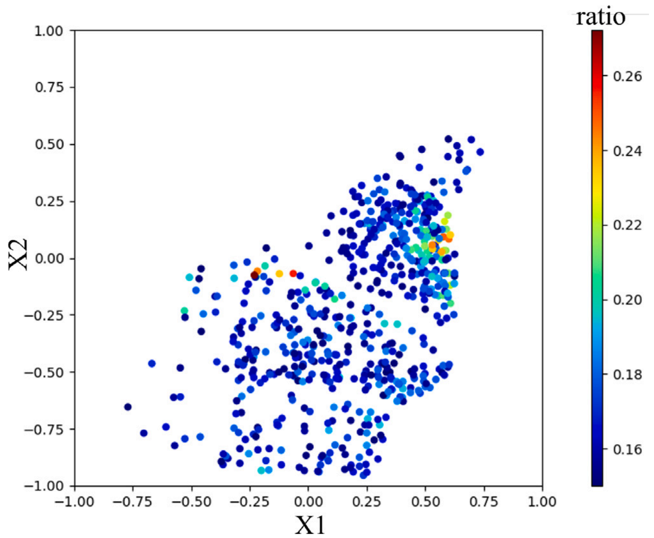

Fig. 10. Latent variable spatial distribution of all training world data.

Table 3
Distribution of latent-variable space of training data.

	Minimum value	Maximum value
X1	−0.770	0.697
X2	−0.955	0.521

Table 4
Parameter-optimization range for Bayesian optimization.

	Minimum value	Maximum value
X1	−0.800	0.700
X2	−0.975	0.525

BO parameters are determined. Outside the range shown in Table 3, the data are not trained for this time and it is unknown how it will behave. By setting the range shown in Table 4 as the settable range of BO parameters, optimization was made possible only within the range where learning was conducted.

5. Results of optimization

To verify the effectiveness of this method, we verified the beam distribution after tuning it to determine whether the shape was suitable for our purpose. To verify the effect of dimensionality reduction by AE, we compared the results of this method with those of normal BO without AE. The number of tuning cycles for this study was 200 (approximately 1.5 h). This was determined as a realistically acceptable beamline tuning time.

5.1. Beam distribution after tuning

The beam distribution at the parameter when R was the largest among the 200 tuning cycles in the BO is shown in Fig. 11. The value of R in each cycle and the highest value up to that point are shown in Fig. 12.

Here, R is determined by the amount of the beam in the range from 17.5 mm to 20.0 mm in radius. Fig. 11 demonstrates that the peaks of the beam were approximately within this range and the value of R was 0.30. However, not all the peaks fell within this range, and some were slightly out of range because 200 tuning cycles are insufficient to achieve fine optimization. However, the fact that such a good result was obtained for this number of tuning cycles out of more than 3 billion parameter combinations in the learning parameter range indicates that the tuning was performed at an astonishingly fast rate. Therefore, although there is still a possibility that R could be increased by increasing the number of tuning cycles, optimization using this method was sufficiently successful.

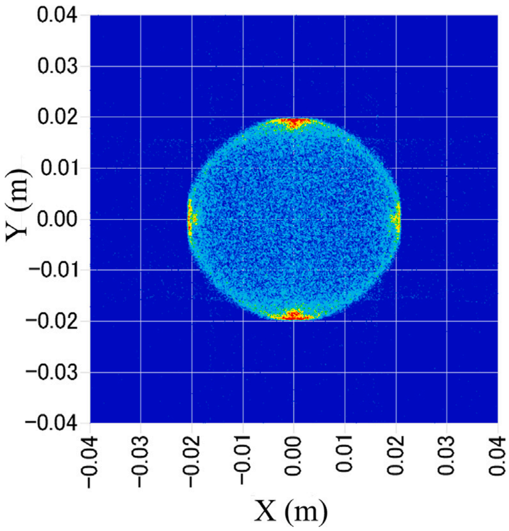

Fig. 11. Distribution of the beam at the parameter when R was the largest among the 200 tuning cycles in the Bayesian optimization.

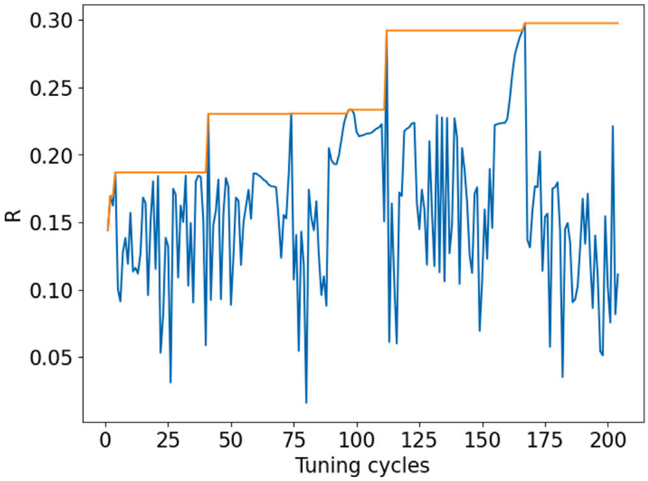

Fig. 12. The value of R in each cycle and the highest value up to that point.

5.2. Comparison with the case without autoencoder

To verify whether the current method was more effective than using only BO, a comparison was made with BO, which optimizes four parameters without AE. The validation was performed by comparing the current method used in combination with AE and BO, using only BO with the same range as the AE training data shown in Table 1 (4para_wide), and a narrower selection of quadrupole magnets (Q6U, Q6D) (4para_narrow) limited to two quadrupole magnets (Q6U, Q6D). Table 5 lists the selectable ranges of Q6U and Q6D for 4para_narrow.

Ten trials of optimization tuning were performed for each of the three methods. The results are summarized in Fig. 13 and Table 6. Fig. 13 plots the number of trials on the horizontal axis and R after the optimization of the vertical axis. The blue dots indicate the current method, which is a combination of AE and BO; red dots indicate 4para_narrow; green dots indicate 4para_wide. Table 6 lists the average

Table 5
Range of quadrupole electromagnet settings for 4para_narrow. The set value is the K value corresponding to the magnetic field gradient.

Magnet	Minimum value	Maximum value	Step width
Q6U	−0.4 (m^{-2})	0 (m^{-2})	0.1 (m^{-2})
Q6D	0 (m^{-2})	0.4 (m^{-2})	0.1 (m^{-2})

Table 6
Average value of R after optimization with each method.

AE and BO	4para_wide	4para_narrow
0.31	0.18	0.23

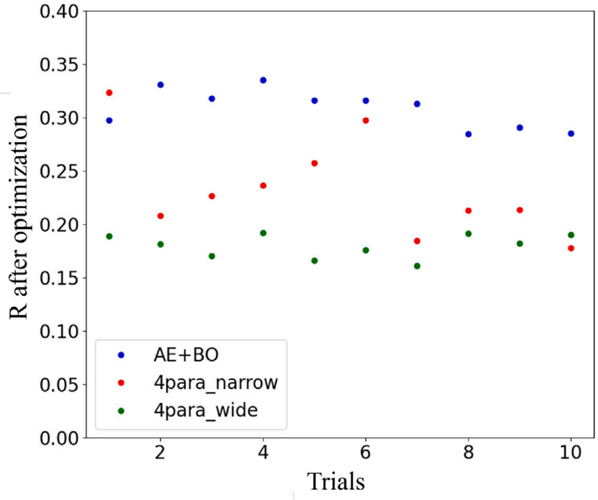

Fig. 13. Value of R after optimization for each method. The horizontal axis denotes the number of optimization tuning trials, and the vertical axis the value of R after optimization. The blue dots are the result of this combined AE and Bayesian optimization method. Red dots are the results of 4para_narrow. The green dots are the results of 4para_wide.

values of R over ten trials for each method. Fig. 13 indicates that the method using AE and BO together converges stably to an R of approximately 0.3. As shown in Table 6, the average value is 0.31. However, for 4para_wide, R did not reach 0.2 in most cases, even after the same 200 optimization tuning cycles. The average value is only 0.18. The method combining AE and BO reached an average of 66% higher R than the optimization with only BO. From this, the method can reach a better solution faster than direct four-parameter optimization.

In 4para_narrow, the same 200 optimization tunings may reach the same level of R as the method that uses AE and BO together.

However, the stability was low, sometimes reaching an R that is not significantly different from that of 4para_wide. The average value was 0.23, and the method combining AE and BO reached an average R 30% higher than the optimization with only BO over a narrower parameter range. Therefore, narrowing some of the parameter ranges can be effective in accelerating the process. However, the method using AE and BO in this study was more stable and yielded better solutions. From the above, this method of using AE and BO together can perform optimization tuning for a wider range of parameters faster and more stably than using only BO.

6. Conclusion

We developed a method that uses BO in combination with dimensionality reduction using AE to solve the problem of increasing number of parameters and tuning time during accelerator tuning. In this study, tuning was assumed to maximize the beam fraction R in the range of 17.5 mm–20 mm in radius, and the effect was verified using beam transport simulations of hollow beam generation. The four electromagnetic parameters were dimensionally reduced to two using an AE and optimized in a two-dimensional latent-variable space. Consequently, we tuned the parameters such that R = 0.30 after 200 tuning cycles, which is sufficiently practical. A comparison was made with the case where only BO was used without AE, and we confirmed that a 66% higher average R could be consistently achieved over the same parameter range in same 200 tuning cycles. Compared with BO with a narrower parameter range, it was also confirmed that the stability of finding the optimal solution was improved, reaching an average of 30% higher R. These results indicate that this method, which combines AE and BO, reduces the dimensionality of the number of parameters to be optimized and enables fast and stable optimization tuning.

Data availability

The data that has been used is confidential.

Acknowledgment

This work was supported by RIKEN Special Postdoctoral Researcher Program.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Y. Morita, et al., Developments of control system for ion source using machine learning, J. Phys.: Conf. Ser. 2244 (2022) 012105.

[2] B. Schwenker, L. Herzberg, Y. Buch, A. Frey, A. Natochii, S. Vahsen, H. Nakayama, A neural network for beam background decomposition in Belle II at SuperKEKB, 2023, arXiv preprint arXiv:2301.06170.

[3] M. Pelikan, D.E. Goldberg, E. Cantú-Paz, et al., BOA: The Bayesian optimization algorithm, in: Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99, Vol. 1, Citeseer, 1999, pp. 525–532.

[4] J. Duris, D. Kennedy, A. Hanuka, J. Shtalenkova, A. Edelen, P. Baxevanis, A. Egger, T. Cope, M. McIntire, S. Ermon, et al., Bayesian optimization of a free-electron laser, Phys. Rev. Lett. 124 (12) (2020) 124801.

[5] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[6] M. Iwasaki, Application of the machine learning to the collider experiments, in: Proceedings, International Workshop on Future Linear Colliders, Vol. 28, LCWS2019, Sendai, Japan, 2021.

[7] D.P. Kingma, M. Welling, Auto-encoding variational bayes, 2013, arXiv preprint arXiv:1312.6114.

[8] Y. Yuri, M. Fukuda, T. Yuyama, Transverse profile shaping of a charged-particle beam using multipole magnets-formation of hollow beams, in: J. Phys.: Conf. Ser., 1350, (1) IOP Publishing, 2019, 012115.

[9] MuSIC, URL, https://www.rcnp.osaka-u.ac.jp/RCNPhome/music/index.html.

[10] Y. Yuri, M. Fukuda, T. Yuyama, Formation of hollow ion beams of various shapes using multipole magnets, Prog. Theor. Exp. Phys. 2019 (5) (2019) 053G01.

[11] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, Software available from tensorflow.org, URL https://www.tensorflow.org/.

[12] GPyOpt, URL, https://sheffieldml.github.io/GPyOpt/.

[13] B. Matérn, Spatial Variation, Vol. 36, Springer Science & Business Media, 2013.