

Reinforcement learning for real-time luminosity optimization in colliders

R. Mamutov^{✉,*}, G. Baranov[✉], and A. Gerasev[✉]

*Budker Institute of Nuclear Physics SB RAS, Novosibirsk, 630090, Russia
and Synchrotron Radiation Facility—Siberian Circular Photon Source “SKIF”
Boreskov Institute of Catalysis SB RAS, Koltsovo, 630559, Russia*



(Received 29 August 2025; accepted 8 December 2025; published 29 December 2025)

This work introduces a reinforcement learning algorithm designed for real-time luminosity optimization in collider experiments. The neural network architecture is selected from multiple candidates through systematic evaluation of their training performance. Prior to training, the input data undergo multistage preprocessing before being fed into the neural network. Our approach combines off-line pretraining on historical accelerator data with online fine-tuning during operation. By processing accelerator measurements over multisecond timescales, the reinforcement learning model dynamically adjusts the magnetic structure to maintain luminosity stability under changing beam conditions. The autonomous nature of the method eliminates the need for manual intervention, enhancing both operational efficiency and beam stability in long-term operation. Experimental validation on the VEPP-4M collider demonstrates the feasibility of the approach and provides a foundation for future development and deployment in accelerator systems.

DOI: [10.1103/71jq-vbl6](https://doi.org/10.1103/71jq-vbl6)

I. INTRODUCTION

One of the challenges in operating colliders is maintaining optimal luminosity, which determines both particle collision efficiency and, consequently, the effectiveness of scientific experiments. Tuning a collider is a nontrivial and iterative procedure because the collider is a complex machine with many sources of distortions and dynamical effects. High collider performance requires not only achieving optimal luminosity but also sustaining it throughout physics experiments. This demands continuous and precise tuning of the machine.

Throughout physics experiments, the optimal state of the collider and the beam collision conditions evolve. Luminosity degradation can occur due to various factors, including field variations, magnet drifts, beam parameter fluctuations (leading to shifts in the tune working point), thermal distortions, beam current effects, and other dynamic influences. These conditions necessitate continuous collider tuning, making luminosity optimization a multifaceted and challenging problem.

Collider tuning can be categorized into two primary approaches. The first focuses on optics correction and optimization procedures [1–16] conducted before physics

data taking. Using a test beam, the collider optics is adjusted to the target configuration to achieve the desired behavior. The second approach, employed with colliding beams, involves luminosity fine-tuning or scanning [17–29]. This work focuses on the latter, enhanced by a novel machine learning (ML) approach.

Traditional control methods, which rely on rigid algorithms and manual intervention, may struggle to adapt promptly to changing conditions. Existing automated luminosity fine-tuning methods also exhibit limitations, as briefly discussed in [30]. In contrast, ML approaches [31–40] offer a promising alternative by enabling the autonomous discovery of optimal machine states, control strategies in real time and novel optimization methods for accelerators. For instance, a reinforcement learning (RL) approach has demonstrated success at BEPCII [30]. Thus, ML-based luminosity optimization presents significant potential to enhance accelerator performance, minimize downtime, and improve experimental efficiency. Additionally, another significant advantage of ML-based approaches is the ability to find latent dependencies that are difficult for operators to identify manually. Trained on extensive datasets and exposed to diverse scenarios, ML models can rapidly make tuning decisions and discover new optimal collider states.

To test and validate the reinforcement learning approach, we applied it to the VEPP-4M collider [41]. It is an interesting use case for such a collider since it has operation stability issues that cannot be handled by traditional algorithms. VEPP-4M is an electron-positron collider operating in the 0.9–6 GeV energy range. High Energy Physics (HEP) experiments are carried out with the KEDR

*Contact author: R.Mamutov@inp.nsk.su

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

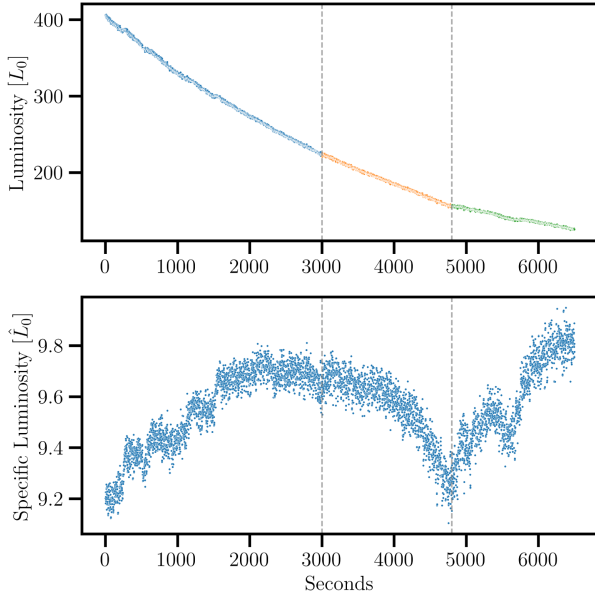


FIG. 1. The absolute and specific (normalized by beam currents) values of small-angle luminosity for an experimental run. There are three periods with the different specific luminosities. The statistical error is comparable to the point size. $L_0 = 10^{28} \text{ cm}^{-2} \text{ s}^{-1}$, $\dot{L}_0 = 10^{28} \text{ cm}^{-2} \text{ s}^{-1} \text{ mA}^{-2}$.

universal magnetic detector [42]. By 2018, a series of measurements had been performed for the precise determination of fundamental particle parameters [43], hadron cross-section measurement (0.92–1.9 GeV) [44] and narrow resonances searches [45]. After 2018, the HEP program includes hadron cross-section measurement (2.3–3.5 GeV), $\Upsilon(1S)$ meson searches, and gamma-gamma physics luminosity run. There are also synchrotron radiation research activities exploiting five experimental stations, accelerator physics research, etc. VEPP-4M operates in decay mode, resulting in luminosity degradation with decreasing beam currents. Depending on the beam energy and initial currents, a single luminosity experiment may last from 15 min up to 3 h. Without continuous tuning, a 3-hour collider operation is affected by dynamic distortions, and optimal luminosity degrades. Ideally, the specific luminosity (normalized by beam currents, Fig. 1) should remain constant during an experiment or even improve due to beam size reduction. However, to restore the optimal configuration, one would need to stop the experiment and reset the operation mode, which is unacceptable. In this context, continuous tuning addresses this challenge effectively.

II. REINFORCEMENT LEARNING

Alongside supervised and unsupervised learning [46], reinforcement learning [47] has emerged as a powerful branch of machine learning for optimizing complex systems where traditional control methods prove inadequate. In particle accelerator physics, RL offers new insights and advantages for real-time beam control and decisions, orbit and

TABLE I. RL methods classification.

Method	Policy	Online/offline	Action space	Model-free
Q learning	Off	On/off	Discrete	Yes
DQN	Off	On/off	Discrete	Yes
SARSA	On	On	Discrete	Yes
DDPG	Off	On/off	Continuous	Yes
TD3	Off	On/off	Continuous	Yes
SAC	Off	On/off	Continuous	Yes
PPO	On	On	Continuous	Yes
CQL	Off	Off	Discrete/continuous	Yes

optics correction, and performance optimization [34,48–51]. In the RL framework, an agent (control algorithm) interacts with the environment (accelerator systems) by taking actions (i.e., adjusting control knobs—linear combinations of settings that have a certain effect on the machine) to maximize a reward (i.e., beam or accelerator performance metrics), thus learning through trial and errors. Unlike other machine learning paradigms, RL agents learn through trial-and-error interactions with the accelerator environment, making them particularly suitable for adaptive control in nonstationary conditions. In the context of luminosity optimization, reinforcement learning is a natural choice for such a problem because the optimal state of the collider is not known in advance and constantly changes due to dynamic effects; the primary objective is to maximize luminosity by adjusting control knobs; the higher the luminosity, the greater the reward for the agent in terms of luminosity gains. The classification of RL algorithms [52] is presented in Table I.

The choice of the algorithm depends on several key considerations. The table compares and summarizes the features of various algorithms for luminosity optimization applications. Online algorithms are not suitable as model training during experimental runs poses safety risks. Model-based algorithms require prior knowledge of the accelerator environment. Off-policy methods offer flexibility by learning from data generated by different policies.

A. Deep Q learning

The Deep Q -Network (DQN) algorithm [47,53,54] was selected for this application due to its effective balance of simplicity and performance for complex systems like colliders. As a value-based, off-policy algorithm, DQN combines Q learning with deep neural networks to approximate the Q function, making it suitable for high-dimensional state spaces. This algorithm is also off-policy: it learns about the greedy strategy, while following a behavior distribution that ensures adequate exploration of the state space. In practice, the behavior distribution is often selected by ϵ -greedy strategy that exploits the model with probability $1 - \epsilon$ and selects a random action with probability ϵ to explore the environment. However, one of the limitations of the algorithm is that it works in the discrete action space, leading to inaccurate control knob values. On the other

hand, decreasing the action step may be inefficient in terms of magnet currents so that small steps will not change the collider state, so a balance is needed. Almost all reinforcement learning algorithms involve estimating value functions that estimate how good for the agent to perform a certain action in a certain state. The result of the Q function (action-value function) is Q value representing the expected total reward an agent will receive if it takes action a in initial state s , following the optimal policy π . It is given by

$$\begin{aligned} Q(s, a) &= \mathbb{E}_\pi[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t = s, a_t = a] \\ &= \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a \right], \end{aligned} \quad (1)$$

where r_t is a reward an agent receives at each time step t ; γ is the discount factor $\in [0, 1]$, which determines how much future rewards are valued compared to immediate rewards; $\mathbb{E}_\pi$ denotes the expected value of a random variable under policy π , i.e., the probability $\pi(a|s)$ of taking action a when in state s .

Usually, DQN exploits two neural networks to approximate the Q function. Unlike supervised learning methods that compare predictions to ground-truth labels, DQN must estimate its own target values for training. However, the weight updates are performed by constructing a loss function using temporal difference targets derived from the well-known equation. The DQN update rule is based on Bellman's equation for Q learning [55], and the target value for the Q function is calculated as

$$y = r + \gamma \max_{a'} Q(s', a'; \theta^-), \quad (2)$$

where r is the immediate reward obtained after taking action a in state s ; s' denotes the subsequent state transitioned from s ; a' represents a potential action in state s' ; θ^- parameterizes the target network. The target network computes Q values for all possible actions in s' , from which we select the maximum value. It is worth mentioning that the targets are dynamic—they depend on the network's weights θ^- during training. This differs fundamentally from supervised learning, where targets are static and predetermined, and no such bootstrapping from the model's own predictions occurs. This recursive, self-referential nature of the Q learning update enables learning without external supervision and introduces unique stability challenges.

Thus, the loss function is defined as the mean squared error (MSE) between the predicted Q value, $Q(s, a; \theta)$ and the target value y computed via the target network:

$$\mathcal{L}(\theta) = \mathbb{E}[(y - Q(s, a; \theta))^2], \quad (3)$$

θ represents the trainable parameters of the main Q network. The expectation $\mathbb{E}_{s,a}$ is taken over state-action pairs. The target network is different from the main network

being updated in every step. The weights of the target are updated less frequently to provide a stable optimization process and to reduce correlation between the two networks.

However, the vanilla DQN algorithm has several notable limitations that can affect its performance. These include the moving target problem caused by frequently changing Q value estimates, an inherent overestimation bias in the max operator, and relatively slow convergence rates, etc. To reduce such factors, there are numerous variations of DQN with improvements which can be found in [56,57].

B. Network architecture

Architecture refers to how data are transformed as it passes through the network. To proceed, we must define the input and output data for our problem. The algorithm's expected output is either the value of a knob to adjust for increasing the collider's luminosity, or a sequence of knob values that yield incremental improvements over time, or a series of actions.

Initially, the model can be designed to predict a single scalar value. However, this prediction is based on input data representing the collider's current state. The state can be described by a set of parameters, such as magnet setpoints, power supply current/voltage measurements, and integral parameters, which may be measured at a specific moment, averaged over a period, or represented as a time series over an interval. The first option may lead to an inaccurate representation since the accelerator operates in a stochastic environment. Alternatively, the input features, computed as time-averaged parameters, form an N -dimensional vector (where N is the total number of parameters). However, averaging over a period results in loss of dynamic behavior and localized feature information. On the other hand, feeding raw time-series data into the model preserves temporal structure and volatility information, which is critical in such a dynamic environment.

In this case, the input data are structured as an $N \times M$ matrix, where N is the sequence length (number of time steps) and M is the number of parameters. For estimation purposes, typical values are $N \sim 10$ time points and $M \sim 100 - 500$ features. A straightforward baseline model is the multilayer perceptron (MLP) [58–60], which is easy to train and deploy. As an alternative, modern architectures [61–64], working with time series, could be most effective for data with structured and ordered sequences. These include recurrent neural networks (RNNs), transformers, 1D convolutional neural networks (1D-CNNs), and other sequence models (i.e., linear recurrences, state-space models, etc.) [65,66]. Due to a growing interest in large-language models [67], the transformer architecture becomes a more popular and advanced model compared to RNNs and 1D-CNN [68]. The transformer architecture outperforms RNN in terms of training performance and predictive power, while 1D-CNN may be trained faster but loses in finding complex feature dependencies [69,70].

III. DATA PREPARATION

DQN relies heavily on historical experience for stable training due to its reinforcement learning nature. However, training such a model in real time during an experiment is highly impractical. Without sufficient historical data, the agent may converge to suboptimal policies and fail to explore the environment effectively. In the context of particle colliders, the exploration-exploitation trade-off becomes particularly problematic during online training due to high operational costs and beam loss risks. It significantly deteriorates the experiment efficiency since an agent has to investigate the action space making non-optimal decisions and thereby reducing luminosity. Additionally, DQN requires extensive training data to achieve effective performance. All this makes the online on-policy approach difficult to implement. As an alternative, we consider an offline off-policy approach for pretraining the model. While the resulting policy may not be fully optimal, it provides a reasonable starting point without the risks of online exploration. Off-policy training exploits data collected in the past or by another policy.

Training data can be categorized into two types: simulation-based and experiment-based (e.g., historical from the database). Accurately modeling collider luminosity—accounting for all physical effects present in a real machine—is a nontrivial challenge. Hence, for this study, the dataset was extracted from a facility operation database, which records measurements at 1-s intervals from 2015 to the present. The VEPP-4 database comprises multiple tables, each corresponding to a distinct parameter type. Some data types are presented in (Table II).

The VEPP-4 database [71] utilizes the PostgreSQL [72] as its relational database management system and provides programmatic access through SQL timestamp-based queries, enabling researchers to retrieve accelerator parameters. Also, the VEPP-4M control system [73,74] is based on EPICS [75] providing real-time access to accelerator parameters through the standard EPICS Channel Access protocol. So one can get up-to-date parameters or adjust accelerator setpoints easily. Values bound to EPICS' process variables are synchronized with the database records. There is neither delay in setting the value nor in

measuring the luminosity. Each entry is recorded once per second.

The data preprocessing pipeline for analysis and model training consists of the following steps: (i) Invalid data removal: elimination of corrupted, missing, or physically impossible measurements; (ii) feature selection: identification and retention of the most relevant parameters based on correlation analysis; (iii) temporal synchronization: alignment of database records to ensure consistent timestamps across all parameters; (iv) feature engineering: construction of derived features; (v) data normalization: standardization of parameter values using the standard score; (vi) RL trajectory annotation: labeling of state-action-reward transitions; and (vii) action discretization: implementation of 12 possible control actions (fixed-step $\uparrow/\downarrow$ adjustments for each knob).

A. RL trajectory annotation

In RL, trajectory annotation is the process of labeling sequential data generated by an agent's interaction with an environment. It involves structuring raw observational data into meaningful components (S -states, A -actions, R -rewards, S' -next states) to create training samples for RL algorithms. In this sense, we label collider operational data with $SARS'$ transitions.

During all years of physical experiments, a legacy luminosity optimization algorithm was continuously in operation. This algorithm adjusts six parameters ("knobs") sequentially and observes the resulting change in the target variable, luminosity. The parameter descriptions are summarized in Table III. The set of knobs includes four individual skew quadrupoles (knob 1–4), an electrostatic beam separator (knob 5), and two quadrupoles combined into one knob (knob 6). The algorithm follows a straightforward principle: for each knob, a discrete fixed step adjustment is made, data are collected over a 7-s interval, the increment in absolute value of small-angle luminosity is calculated, and the knob's setpoint is updated. If no luminosity improvement is observed or when the number of changes for one knob reaches 10, the algorithm switches to the next knob. This process iterates through all six knobs cyclically, completing one full cycle in approximately 3 min. All algorithm decisions are recorded in the database as regular setpoint

TABLE II. Data categories presented at the VEPP-4 database.

Table	Parameters
Accelerator elements setpoints	Current, voltage
Elements measurements	Current, voltage
Temperature indication	T water, air, elements
Vacuum indication	Average vacuum, local vacuum
Global parameters	Luminosity, beam lifetime, beam currents, tunes, energy
Beam diagnostics data	Beam orbits, beam sizes

TABLE III. Knobs description.

Knob	Role
1. Skew quadrupole SEQ2	Global coupling correction
2. Skew quadrupole SEQ3	Global coupling correction
3. Skew quadrupole NEQ2	Global coupling correction
4. Skew quadrupole NEQ3	Global coupling correction
5. Electrostatic plates EU5	Beam steering at IP
6. Quadrupole pair S(N)EG1	Waist position control

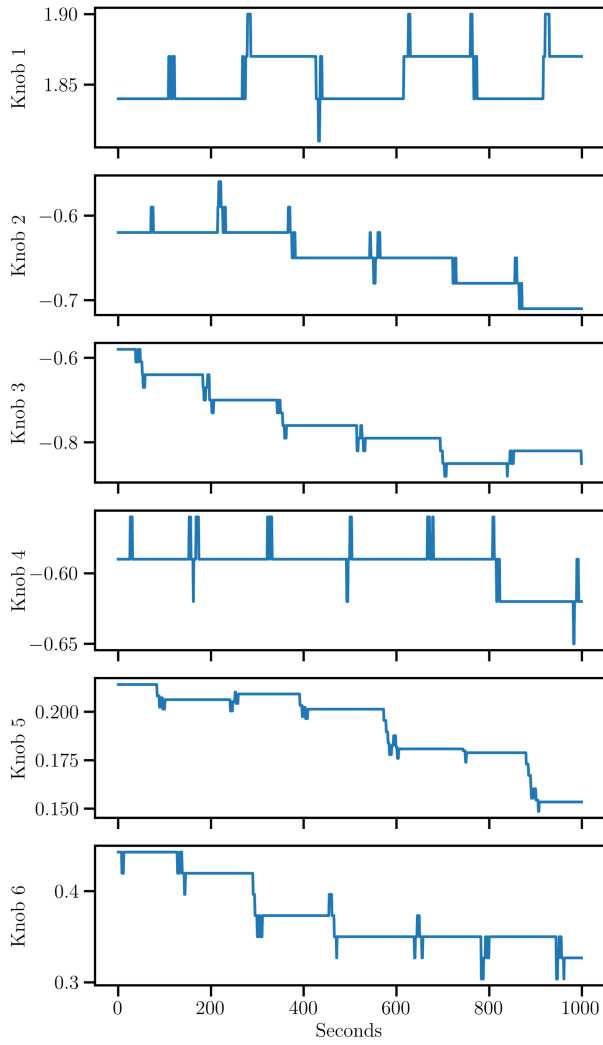


FIG. 2. Control knob values during luminosity optimization with the legacy algorithm for an arbitrary run. Only one knob is changed every 7-s interval.

changes for this set of knobs. These historical data enable performance evaluation of the program and provide *SARS'* transitions suitable for training sample generation. Figure 2 shows the values of six knobs every second over a period of time during optimization. It can be seen that some knobs exhibit values that fluctuate within a relatively narrow “channel” throughout the experiment. The other knobs (knob 3, knob 5, and knob 6) display a clear and slow trend over time. This observed drift may be the signature of the drifting optimum of the collider. Moreover, it shows that the optimization process is coupled as different knobs are used for distinct but interrelated tasks.

In this context, the RL components are defined as follows: S is a collider snapshot for each second during a 7-s period before a knob is changed, i.e., an action A is taken; the performed action results in a next collider state S' observed over the next 7 s and a small-angle luminosity increment R , calculated as the difference between the

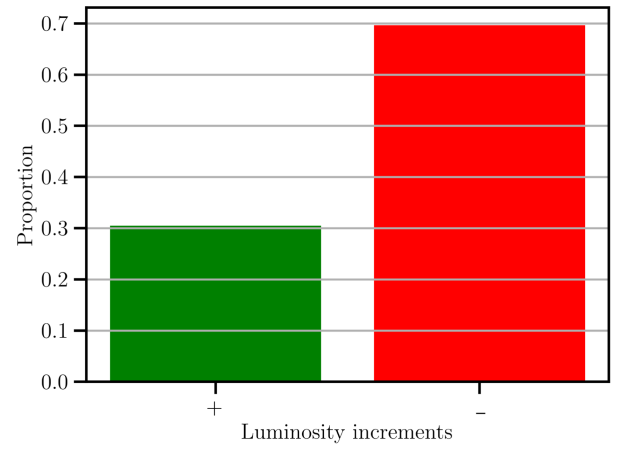


FIG. 3. Luminosity changes without online optimization. “+” indicates positive luminosity increments, while “-” indicates negative luminosity increments. The luminosity increments are calculated as the difference between the 7-s averaged readings before and after the action is applied.

averaged values for 7 s before and after the action. This operation of the legacy algorithm demonstrates that a discrete action space, and consequently DQN, is well suited for our problem.

Figure 3 illustrates the distribution of luminosity increments over 7-s window without online tuning. The observed asymmetry toward negative values reflects the luminosity decay inherent in normal collider operation.

B. Data filtering

The dataset also contains anomalies that require removal. These anomalies originate from multiple sources, including subsystem malfunctions, where some subsystems were nonoperational during an experiment while their readings were still recorded; measurement errors, such as outliers caused by electronic noise or sensor inaccuracies; and software issues in the control or database logging systems.

To address these issues, the filtering procedures were implemented: (i) Nonphysical values were removed, including negative values for strictly positive parameters and records showing constant parameter values throughout an experiment; (ii) experiments with insufficient durations (fewer than two complete cycles of the legacy optimization program) were excluded; (iii) a luminosity threshold of $10 \times 10^{28} \text{ cm}^{-2} \text{ s}^{-1}$ was applied; experiments failing to exceed this threshold were discarded; (iv) parameter values below 10^{-6} were treated as zero, as they were considered physically insignificant; (v) for missing values, setpoint gaps were filled with zeros; and (vi) for measured parameters, experiments with over 30% missing or zeroed data were deemed anomalous and excluded.

Outlier detection and removal were performed using the interquartile range method with linear interpolation. This approach was applied both at the individual

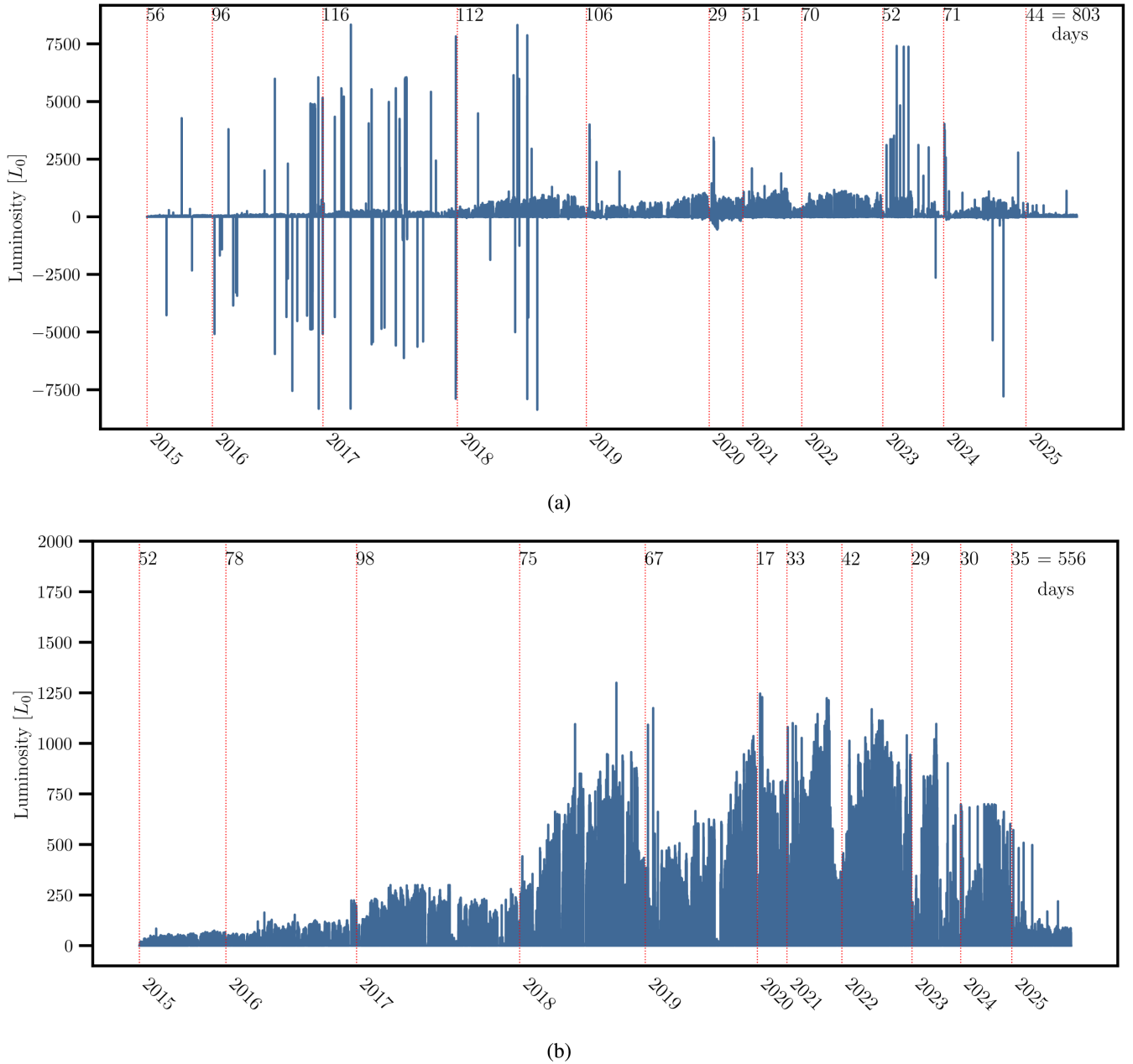


FIG. 4. Luminosity measured per second over several years before and after filtering. $L_0 = 10^{28} \text{ cm}^{-2} \text{ s}^{-1}$. (a) Not filtered. (b) Filtered.

experiment level and globally across all valid experiments, using thresholds of 10% for the percentile value and 0.5 for the scale factor. The same approach was employed to handle parameter gaps through linear interpolation. Figure 4 shows the effect of the luminosity data filtering.

For vacuum-related measurements, complete daily datasets were sometimes missing. These gaps were filled using monthly average vacuum values.

Furthermore, the full dataset contains redundant information that is unnecessary for describing the accelerator state or training the model. The first source of redundancy originates from data irrelevance. The database contains

extraneous data because the accelerator operates in two modes: as a collider and as a synchrotron radiation source. We separated the collider operation data from synchrotron radiation data using operational flags, including the electrostatic alignment flag and experiment initiation markers. Collider-related data were tagged with unique experiment identifiers based on the date.

While one might assume that including more parameters would enhance RL model performance, excessive features can actually degrade results [76–78]. Many parameters exhibit strong correlations as they represent similar physical effects, while others provide only marginal

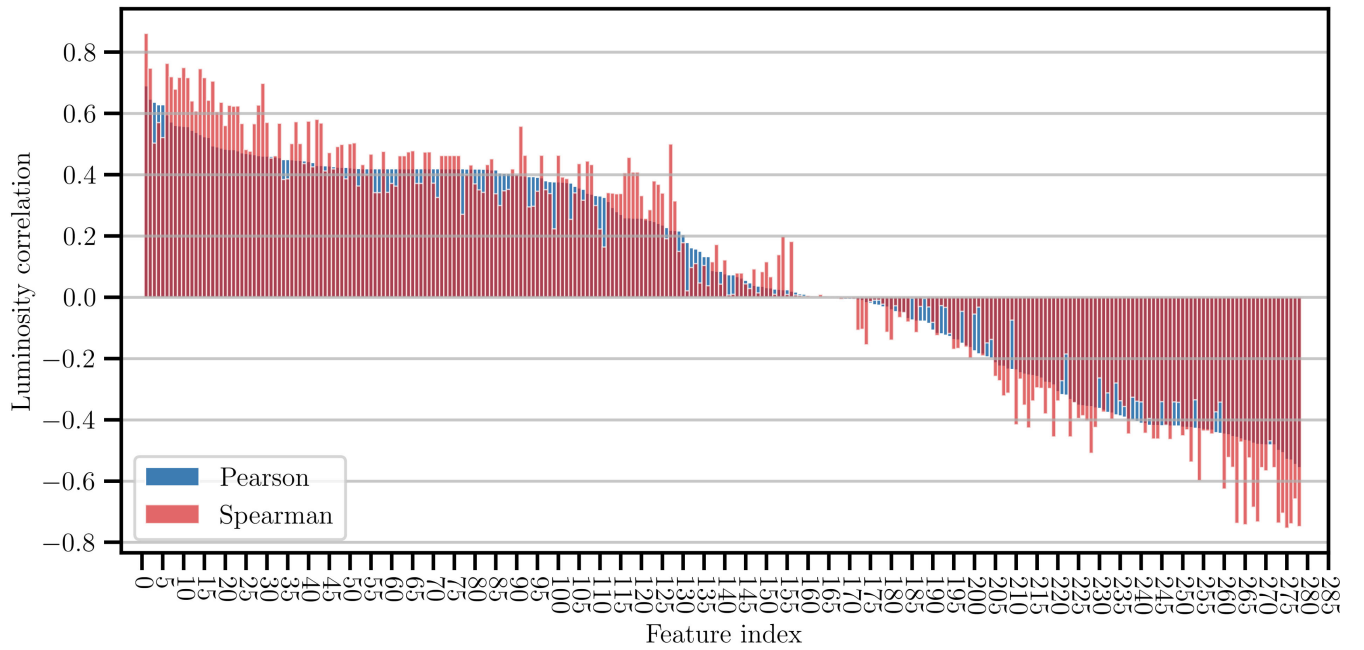


FIG. 5. Correlation of features with luminosity. Positively correlated features increase with luminosity, while negatively correlated features decrease with luminosity.

contributions to the accelerator state description. Such parameters are less informative and should be skipped. A correlation analysis was conducted to identify features most strongly associated with the target parameter (luminosity). Only the most significant features were retained for training. This is achieved by analyzing parameter importance—evaluating both their overall impact on the accelerator and their specific influence on the target metric. Notably, the final luminosity value depends not only on the collider’s magnetic structure but also on beam-beam effects and collective phenomena. While increasing particle numbers collision rates (and thus luminosity), the particles’ self-induced fields modify their dynamics, ultimately reducing the limiting luminosity. All these factors must be carefully considered.

The selection of input parameters for the model was based on correlation analysis between the predictors and the target variable [79,80], though principal component analysis could provide additional insights [81,82]. The correlation diagram is depicted in Fig. 5. Two tests were conducted. Pearson correlation measures linear relationships between variables, while Spearman correlation measures monotonic relationships, which can be nonlinear. Both metrics are useful since luminosity may have a nonlinear dependence on features. Parameters with a modulus below 0.2 were dropped as negligible. This threshold was chosen based on empirical guidelines [83,84] and the physical considerations: some correctors are rarely used in orbit correction and thus minimally affect luminosity; two wigglers are used for synchrotron radiation source mode, etc. While parameters like total beam current (derived from the sum of four bunch

currents) or vacuum pump parameters were deemed redundant. The figure also demonstrates that the two barplots have many in common. Some coefficients differ, but the general trend is similar. Higher values of Spearman coefficients compared to high Pearson coefficients suggest a nonlinear behavior. Low values for both methods indicate a weak relationship, which is observed for values below 0.2. To confirm that the parameters were negligible, we additionally plotted the dependence of luminosity on each parameter and verified that the plots show no clear trends.

The dataset for the correlation diagram covers 10 years of facility operation, though recent data may better reflect current luminosity conditions. Table IV lists the top 10 parameters most strongly affecting the small-angle

TABLE IV. Top 10 important features highly correlated with the small-angle luminosity. The absolute value of Pearson correlation coefficient R is used. Parameters 4, 8, 9, and 10 represent the current values of the power supplies.

Feature	$ R $
1. Total beam current	0.690
2. Positron bunch 2 current	0.647
3. Specific luminosity	0.637
4. SEQ2 current	0.629
5. Average vacuum	0.595
6. Positron bunch 1 current	0.572
7. Electron bunch 2 current	0.561
8. SIX1 current	0.560
9. NRZ6-2 current	0.558
10. SIM1 current	0.557

luminosity. Among the key parameters, SEQ2 (a final-focus skew quadrupole used for online optimization) has a high impact on luminosity and appears as expected, while SIX1, NRZ6-2, and SIM1 (dipole correctors located far away from the final focus) represents a more surprising finding and has a high correlation value. The influence of the correctors may arise from dynamic aperture reduction via local orbit distortion or through vacuum leaks, which are known operational issues. While critical parameters like electron bunch 1 current, beam energy, and lifetime do not appear in the top 10, they are certainly important features.

C. Feature generation

To enhance the dataset for model training, several useful features were generated: (i) temporal features: sine and cosine transformations of time components (seconds, minutes, hours, day of week, day of month, and month) to capture both short-term and long-term seasonality patterns. These help account for operational variations, such as reduced predictability during summer months due to component overheating; (ii) operational duration metrics: elapsed time (in minutes) since accelerator remagnetization and new experiment start, quantifying system drift over time; (iii) specific luminosity: calculated as luminosity normalized by beam currents, i.e., $L/(I_{e1}I_{p1} + I_{e2}I_{p2})$; and (iv) logarithmic transformations: applied to the absolute value of small-angle luminosity measurements.

D. Sampling

The dataset covers the 10-year period from 2015 to 2025, containing approximately 300 GB of data across 70 million records. This period includes various collider operational modes, encompassing different energy levels and luminosity regimes. For an effective reinforcement learning application, proper sample labeling and training dataset construction according to standard RL methodology are crucial.

1. State representation

The environment state is characterized by accelerator parameters and beam characteristics sampled at 1-s intervals. While there are numerous parameters describing the accelerator's state, using all of them is impractical for several reasons. First, many parameters exhibit negligible influence on luminosity. Including these could dilute the learning process. Second, some parameters change very slowly or remain constant during the experiment, while others are highly dynamic and have a dominant effect on the system. Third, including an excessive number of parameters increases model complexity, making hyperparameter tuning and training more challenging. Moreover, the size of the dataset must grow proportionally with the number of parameters to maintain sufficient variability and representativeness of the

TABLE V. Input features describing the collider's state. Integral parameters include beam energy, lifetime, etc. Generated features refer to those described in Sec. III C.

Feature type	Number
Elements settings	224
Beam orbits and sizes	234
Integral parameters	11
Generated features	17

data. The scaling laws are well documented in machine learning literature [85–88].

As mentioned earlier, the legacy optimization program operates on a 7-s cycle: it collects data, computes the absolute luminosity change, and implements control adjustments. Following this principle, we structured our samples by averaging 7-s groups of sequences. This approach smooths out noise and creates stable, meaningful samples for training.

An alternative approach employs a 7-s time series for each parameter as the state representation, which will be shown later. This sequential representation could better capture temporal dependencies and dynamics in the system, potentially improving model performance.

Consequently, the model input accepts two representations: an N -dimensional vector and an $7 \times N$ matrix. The complete table of features is provided in Table V.

2. Action representation

The output vector is represented as a 12-dimensional Boolean vector (one-hot encoded), where each nonzero element corresponds to either an upward or downward step of a specific knob. The step size was empirically chosen to produce observable luminosity changes while maintaining stability. This structure is naturally suited for discrete action spaces, where the model learns to adjust an individual parameter at a time. From legacy program operation, it can be observed that multiple consecutive unidirectional parameter changes (monotonic increases/decreases), i.e., any arbitrary knob may be altered multiple times in a row in one direction. In this case, there is also an option to use parametric action DQN (discretized parameterized action space, multiscale discrete action DQN) [89–91]. $S_0 \rightarrow A_0 \rightarrow \dots \rightarrow A_{n-1} \rightarrow S_n$ transition can be compressed into $S_0 \rightarrow A_0^n \rightarrow S_n$, where n is an index of the last state and also a multiplier for the action discretization. Let A_0 is associated with a knob step a , so the final discretization for this knob is $\{-n * a; -a; a; n * a\}$. This discretization may be extended if there are transitions with other multiplier values. Moreover, one can vary several knobs simultaneously to form a combined action, but it requires transition statistics for model training. Creating hybrid actions can speed up achieving the collider's optimal state and remains a subject for future investigation.

3. Reward calculation

The VEPP-4M detector provides several independent luminosity measurements. We exploit the so-called small-angle luminosity—i.e., collisions that occur at very small angles relative to the incoming beam direction. The objective for the agent is to maximize the luminosity over either short or long timescales. The reward is feedback for the agent, showing the success of its actions. It is calculated as the difference between the average luminosity over the 7-s intervals before and after an action is applied. Luminosity and other parameters measurements are updated every second, while each agent interaction takes 7 s for measurement and 1 s to adjust knob values.

IV. MODEL TRAINING AND TESTING

A key feature of the proposed approach is a two-stage training scheme. The model is initially trained on historical collider operation data obtained from the database, which helps form an initial control policy, reduces training time, and avoids inefficient actions. The model is further being fine-tuned in real time during the experiments.

We evaluate two networks with different architectures (see Sec. II B): an MLP and a transformer. Each architecture has a corresponding target network that updates slowly as described in Sec. II A. The proposed MLP consists of three fully connected layers with ReLU activation functions. The input layer processes features of dimension 486, while the hidden layer transforms the data with dimension 512. The output layer produces Q value predictions with 12 units, from which we select the maximum one according to the policy. Dropout layers ($p = 0.1$) are applied after each hidden layer to mitigate overfitting. The target networks do not have the dropout layers since they are used for the stable Q -value estimation. The learning rate and the weight decay were set to 3×10^{-4} and 10^{-5} , respectively, during the training process. Adam optimizer was used. The batch size was equal to 128, with the total dataset size being $\sim 300,000$. Soft update parameter for the target network τ was set to 0.01. The discount factor γ used in the Q value update expression was set to 0.5 to focus on immediate rewards. The number of epochs was 2000, but as was stated earlier, there are no explicit metrics to evaluate the training process as there are in supervised learning. However, by indirect means, we can terminate the training process when the average Q values stabilize near the expected threshold [see Eq. (4) below] of cumulative expected reward.

The transformer-based Q -network architecture processes state sequences of length 7 through a single transformer encoder layer with 2 attention heads and 256-dimensional embeddings. The model first projects input states into a 256-dimensional space using linear embeddings, then adds learned positional encodings for sequence awareness. The transformer processes these embeddings using masked self-attention with dropout ($p = 0.1$) and layer normalization,

followed by a position-wise feed-forward network with dimension 512. After mean pooling the sequence dimension, a final linear layer outputs Q values for all 12 actions. We trained the model using AdamW (learning rate = 3×10^{-4} and weight decay = 10^{-5}) with batch size 128, employing target network soft updates with $\tau = 0.01$. The discount factor γ was set to 0.5 for future reward estimation.

Hyperparameters tuning process is always challenging and time consuming. The selection of parameters was done using a grid search.

We also evaluated our models in the simulated environment. The environment simulates the collider and produces luminosity values based on input parameters representing the machine state. The virtual collider is based on neural networks similar to those described above. The network accepts the same state vector as input and predicts luminosity values instead of Q values. The state representation is determined by the collider's parameters, including six control knobs. The objective is for both models to optimize the virtual collider's performance by adjusting these control knobs under various initial conditions.

This process is not part of the training phase but rather serves as a validation step to assess the models' performance in a simulated environment and select the most efficient one. This selection is necessary because, in subsequent real collider operations, only one model can be used due to limited beamtime availability. And at the stage of experimental validation, testing different trained models would be resource consuming.

For the environment networks, we modified the training procedure with the only difference—using luminosity as the target value. And the optimization objective was to minimize the loss between predicted and actual luminosity values.

For the model validation, we executed 1000 test scenarios with different seeds and randomized initial state vector values. The models were tasked with exploring and determining control knob settings that yield maximum luminosity in each scenario and give the greatest rewards. Each agent finds the virtual collider's optimal state by varying six control knobs. Each episode resets the state parameters to the initial values. Every next episode, the agent finds the optimum faster. It demonstrates the convergence speed comparison for each model. Figure 6 illustrates the cumulative rewards obtained by models during luminosity optimum search. The transformer model demonstrates faster convergence compared to the MLP model. The observed performance difference may stem from either better fine-tuning of the transformer model or its more advanced architecture.

While the virtual collider closely mimics the real machine, the primary goal of this synthetic optimization task was not to achieve real-world performance, but to compare the convergence properties and exploration behavior of the two model architectures within a reinforcement learning context.

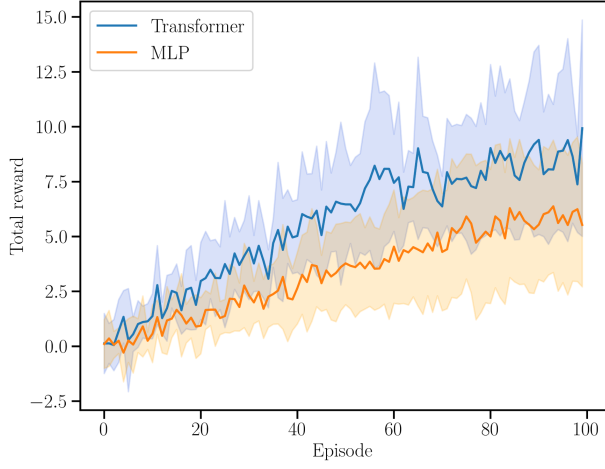


FIG. 6. Luminosity optimization in a virtual collider using six control knobs, evaluated with two different models. The results were obtained by averaging over 1000 scenarios with randomized initial conditions.

Our results demonstrated that the transformer-based agent exhibited more efficient exploration and faster convergence. Therefore, it will be used in the further experimental validation.

V. EXPERIMENTAL VALIDATION OF THE MODEL

The VEPP-4M collider’s 2024–2025 physics program focused on collecting statistics at the J/ψ resonance (~ 3.1 GeV). To maintain operational safety and prevent beam loss, the training dataset also includes the most recent data, and the model was tested before exploitation at the real machine. Moreover, several safeguards were implemented, including physical limits on control knob values; real-time luminosity drop detection; and automatic rollback to previous stable settings when needed.

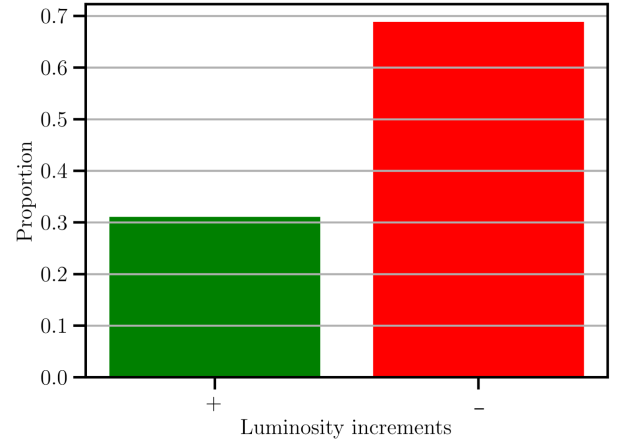
The new program operated for 3 months. To ensure the model accurately reflected the collider’s state, we initialized it with $\epsilon = 0.9$ and gradually reduced this to $\epsilon = 0.2$ over 50 iterations. This approach is performed for the first few runs at the beginning of the shift, and then we keep $\epsilon = 0.2$ fixed. Moreover, the model employs a dequeline replay buffer which collects and stores training samples for model fine-tuning. The replay buffer stores the most recent 10,000 training samples. Also, we set $\gamma = 0.5$ to prioritize short-term luminosity maximization. According to Eq. (1), the series of total reward converges because $\gamma < 1$ and can be bounded by the following sum:

$$Q(s, a) = \frac{\bar{r}}{1 - \gamma}, \quad (4)$$

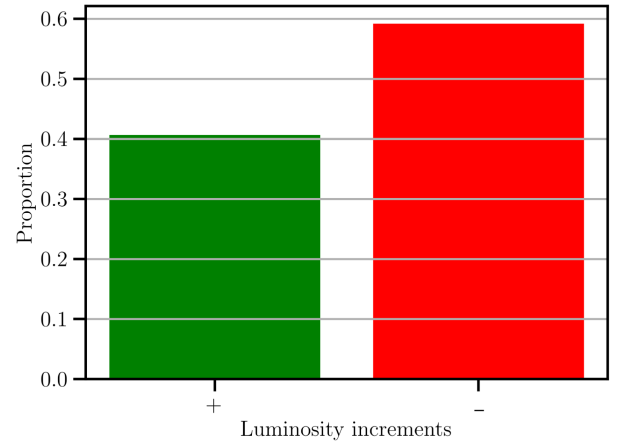
where $\bar{r}$ is the expected value of the reward (luminosity increment). Setting $\bar{r} = 0.7$, one obtains $Q(s, a) \approx 1.4$. It means that the agent disregards long-term action

consequences, focusing instead on the first 3–5 steps. Long-term luminosity maximization proves ineffective given the environment’s inherent noise and unpredictability—random fluctuations inevitably disrupt any long-term strategy. An example of a long-term strategy is the agent gets a greater reward after the first action and gets a total smaller reward over two actions, while it can get a smaller reward after the first action and get a total greater reward over two actions.

To compare the performance of the new and legacy algorithms, we analyze some operation statistics from that 3-month period. Figure 7(a) shows the distribution of “good” and “bad” decisions performed by the legacy program. It can be seen that almost 70% of actions led to luminosity decrease, which can be interpreted as an ineffective algorithm because it tests knob adjustments in both directions to locate optima, employing a brute-force



(a)



(b)

FIG. 7. Algorithm comparison: “+” stands for “good” actions leading to positive luminosity increments, while “−” stands for “bad” actions leading to negative luminosity increments. The luminosity increments are calculated as the difference between the 7-s averaged readings before and after each action. (a) Old algorithm. (b) New algorithm.

strategy and spending time. Specifically, it goes through a cycle and iterates over the knobs in a certain order. The same histogram for the new algorithm is presented in Fig. 7(b). In contrast, the algorithm makes fewer erroneous decisions and does not spend time on exploration, leading to an increase in actions with more frequent positive gains in luminosity. The luminosity increments are calculated as the difference between the 7-s averaged readings before and after the action is applied. One can estimate the performance enhancement by comparing the luminosity integral over several experiments, average specific luminosity or relative luminosity, using the following expression:

$$\frac{dL_t}{L_t} = \frac{\frac{\sum_{k=t+1}^{t+N} L_k}{N} - \frac{\sum_{k=t-N}^t L_k}{N}}{\frac{\sum_{k=t-N}^t L_k}{N}}, \quad (5)$$

where L is the luminosity measurement; t is the moment of time when the action was applied; N is a time window in seconds; so $t - N$ represents a moment of time N seconds ago.

The increased number of successful actions directly correlates with more frequent positive luminosity increments. Figure 8 compares reward histograms for three cases: no online tuning, legacy program operation, and new program operation. The results for all three scenarios were collected concurrently within the same 3-month operation. Data were collected for each scenario sequentially in a rotating manner. The overall operational mode remained

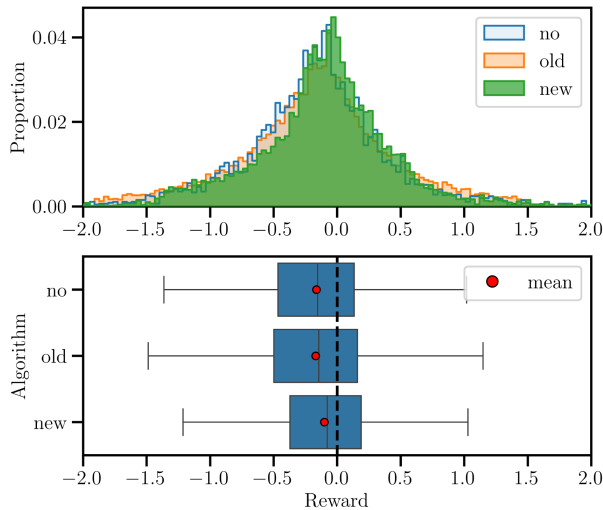


FIG. 8. Algorithm comparison: “no” stands for no tuning during the experiments, “old” stands for tuning with the legacy algorithm, “new” stands for tuning with the DQN algorithm. Each histogram represents a reward distribution. The reward is calculated as the difference between the 7-s averaged luminosity readings before and after each action. In the case with no tuning, the reward is simply the difference between two consecutive 7-s averaged luminosity readings. The box plots below provide summary statistics of these distributions.

constant during this period, ensuring identical experimental conditions. The phases with no tuning include periods where the algorithms had already brought the system to a near-optimal state. So the performance without tuning is nearly equivalent to that with the legacy algorithm. While the legacy algorithm may provide initial benefits, particularly when determining collider optima controlled by a single knob requiring several adjustments, It does not provide any improvements in the long term. Its convergence is slow and inefficient. It cannot keep up with the collider’s quasistatic optimum and makes many harmful steps on average, but is able to find highly suboptimal knob values given the current system state. The legacy algorithm is based on active search, and its performance drops are the direct costs of its inefficient exploration strategy. On the other hand, the histogram of the new algorithm’s performance has a systematic bias toward positive rewards. It indicates that it gains more positive rewards on average. However, each histogram mean value is negative since the downward luminosity trend cannot be compensated.

Figure 9 shows the evolution of knob values during optimization for the two control methods. Old algorithm behavior can be interpreted as chasing an optimum. This optimum drifts over time, likely due to beam degradation processes. The algorithm is constantly chasing this optimum, smoothly and monotonically changing parameters to compensate for the deteriorating conditions. This results in a trend where the knob values appear to follow the decreasing luminosity trend. However, this is a reactive process that always lags behind the true optimum, producing the observed monotonically decreasing curves. In contrast, the DQN algorithm operates differently. Instead of monotonically following the drifting optimum, it exhibits a nonmonotonic, oscillatory behavior in its parameter adjustments.

Figure 10 illustrates the small-angle luminosity comparison for two control methods. It is shown that the DQN algorithm outperforms the old one over a distance. On average, the integral luminosity improvement achieved by the DQN algorithm is 4% compared to the performance of the legacy algorithm. For a correct comparison, Fig. 11 shows the specific luminosity $\hat{L}$ per run for the equal bunch currents and experiment duration, calculated as an average value, using the expressions below:

$$\hat{L} = \frac{L}{I_{e1} * I_{p1} + I_{e2} * I_{p2}} \propto \frac{1}{\sigma_x \sigma_y}, \quad \langle \hat{L} \rangle = \frac{\int_0^T \hat{L} dt}{T}. \quad (6)$$

As direct, real-time measurement of beam parameters such as the transverse beam sizes and overlap at the interaction point is challenging, this averaged specific luminosity serves as the primary and most reliable indicator of the tuning algorithms’ performance. A higher value signifies a more effective algorithm, as it directly reflects an improved collision rate. However, the results indicate

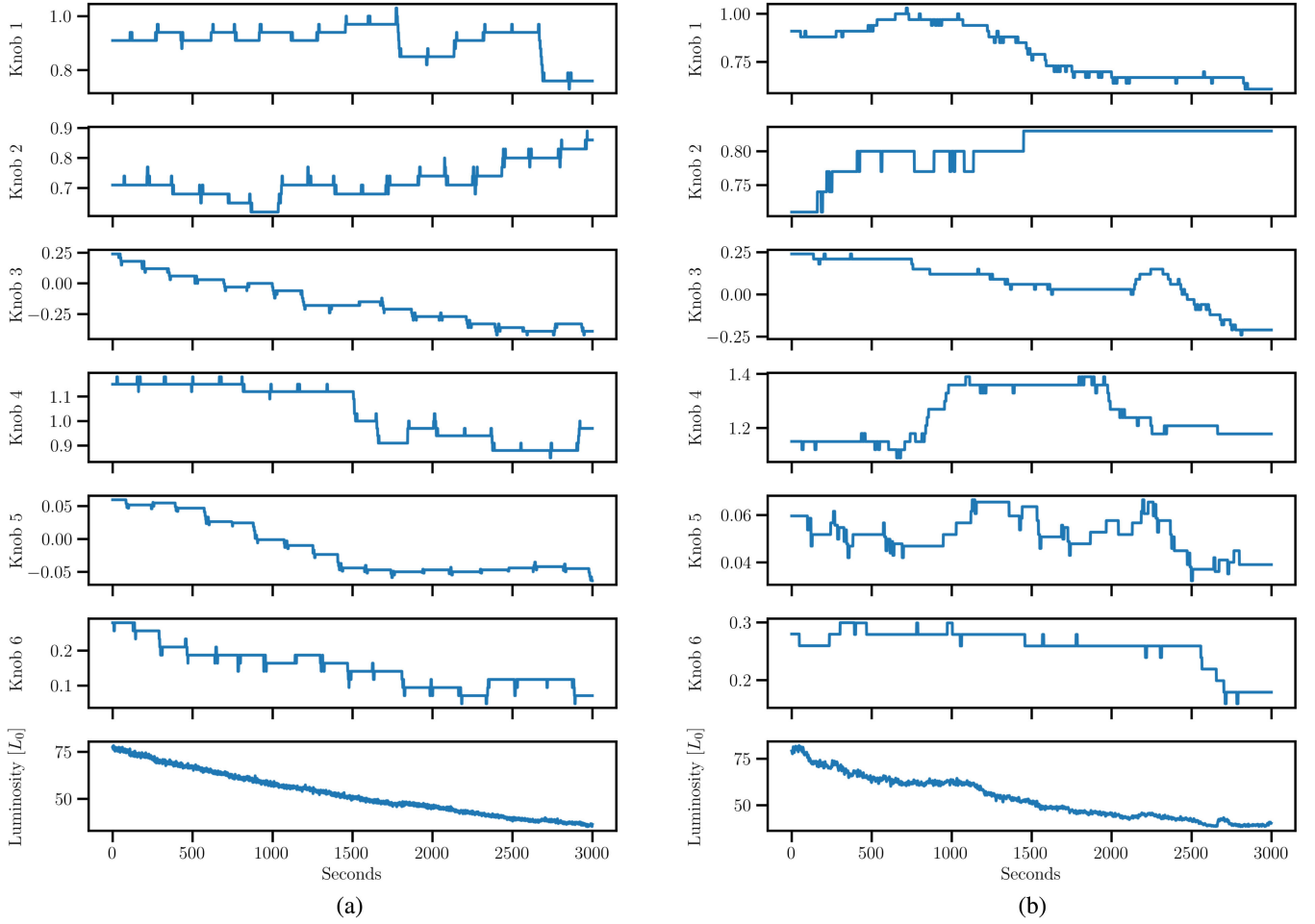


FIG. 9. Evolution of control knob values and luminosity for two experiments optimized by different control methods. $L_0 = 10^{28} \text{ cm}^{-2} \text{ s}^{-1}$. (a) Old algorithm. (b) New algorithm.

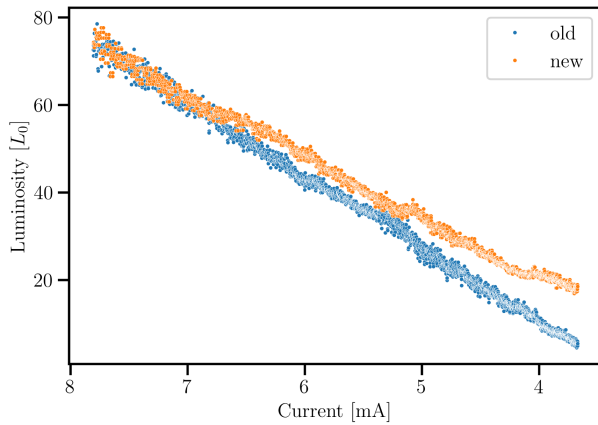


FIG. 10. Small-angle luminosity, averaged over ten runs, for two control methods: “old” refers to tuning with the legacy algorithm and “new” refers to tuning with the DQN algorithm. The statistical error is comparable to the point size. $L_0 = 10^{28} \text{ cm}^{-2} \text{ s}^{-1}$.

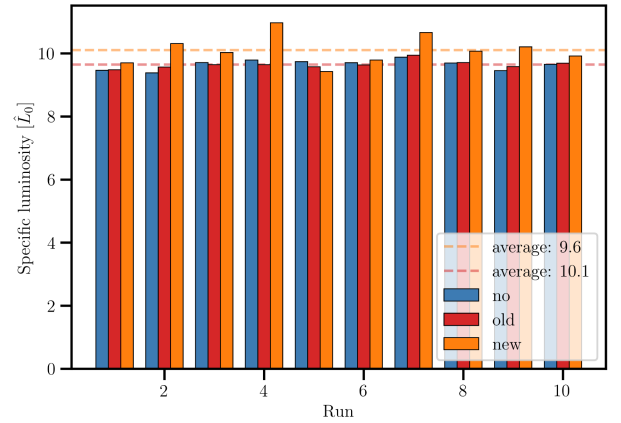


FIG. 11. Average specific small-angle luminosity $\langle \hat{L} \rangle$ over ten consecutive runs. Each run comprises three experiments: “no” stands for no tuning during the experiments, “old” refers to tuning with the legacy algorithm and “new” refers to tuning with the DQN algorithm. The “average” lines refer to the two methods and are highlighted with their respective colors. $\hat{L}_0 = 10^{28} \text{ cm}^{-2} \text{ s}^{-1} \text{ mA}^{-2}$.

potential for further optimization, and further investigation is required.

VI. CONCLUSION

This study demonstrates the successful application of reinforcement learning for real-time luminosity optimization in particle colliders. The proposed DQN algorithm achieves a 4% performance improvement over the conventional method while operating on the same set of control parameters. The study highlights two critical components for successful implementation: pretraining on historical operational data, which provides essential baseline knowledge, and subsequent online fine-tuning that allows the model to adapt to the collider's dynamic state while maintaining up-to-date information about the machine conditions.

The results suggest several directions for future improvements, including hyperparameter optimization, expansion of control parameters, and potential implementation of continuous action spaces for finer control. The number of knobs can be increased to account for various factors and achieve greater luminosity gains. Moreover, additional validation at different energy levels and operational conditions remains necessary.

Methods with continuous action spaces can be considered for more precise tuning, but it should be investigated whether a reduced step value for a discrete action space is less effective. It is also a subject of debate whether the existing training data can be used in a hybrid approach (hierarchical RL [92]). The first level of prediction suggests a knob to change (discrete action space), while the second level predicts the exact value in a continuous space.

However, the research also reveals important constraints and challenges. The significant deviation between simulated models and actual accelerator behavior necessitates conservative control policies to prevent beam loss and maintain operational safety. This requires careful balancing between exploration and exploitation, with particular emphasis on minimizing potentially disruptive actions that could affect experimental efficiency. Overly aggressive actions may lead to beam loss—a potentially dangerous outcome that disrupts experiments. Unfortunately, it is quite difficult to create a fully representative training environment for the RL agent, as the real accelerator's luminosity behavior significantly deviates from simulated models. All possible failure scenarios must be preemptively analyzed, with corresponding action constraints implemented in the decision-making algorithm. The agent should minimize exploratory (and potentially detrimental) actions, as these directly reduce experimental efficiency and data quality. Additionally, adaptation to dynamic conditions is necessary, as the accelerator state changes daily.

The proposed approach shows potential for broader applications in accelerator physics, as the methodology could be adapted to optimize other machine parameters beyond luminosity. While the current implementation

represents an important step forward, further refinement of the model architecture, parameter selection, and training procedures could unlock additional performance gains or help to find new applications. This work contributes to the growing field of intelligent control systems for particle accelerators, demonstrating how reinforcement learning can enhance operational efficiency.

ACKNOWLEDGMENTS

This work was partially supported by the Ministry of Science and Higher Education of the Russian Federation within the governmental order for SRF SKIF Boreskov Institute of Catalysis (Project No. FWUR-2025-0004).

DATA AVAILABILITY

The data that support the findings of this article are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

-
- [1] X. Huang, *Beam-Based Correction and Optimization for Accelerators* (Taylor & Francis, Boca Raton, 2020), 10.1201/9780429434358.
 - [2] R. Tomás, M. Aiba, A. Franchi, and U. Iriso, Review of linear optics measurement and correction for charged particle accelerators, *Phys. Rev. Accel. Beams* **20**, 054801 (2017).
 - [3] F. Zimmermann, Measurement and correction of accelerator optics, in *Beam Measurement* (World Scientific, Singapore, 1999), pp. 21–107.
 - [4] J. Safranek, Experimental determination of storage ring optics using orbit response measurements, *Nucl. Instrum. Methods Phys. Res., Sect. A* **388**, 27 (1997).
 - [5] V. Sajaev, V. Lebedev, V. Nagaslaev, and V. Valishev, Fully coupled analysis of orbit response matrices at the FNAL Tevatron, in *Proceedings of the 21st Particle Accelerator Conference, Knoxville, TN, 2005* (IEEE, Piscataway, NJ, 2005), pp. 3662–3664.
 - [6] T. Persson, F. Carlier, J. C. De Portugal, A. G.-T. Valdivieso, A. Langner, E. H. Maclean, L. Malina, P. Skowronski, B. Salvant, R. Tomás *et al.*, LHC optics commissioning: A journey towards 1% optics control, *Phys. Rev. Accel. Beams* **20**, 061002 (2017).
 - [7] A. Romanov, D. Berkaev, P. Y. Shatunov *et al.*, Round beam lattice correction using response matrix at VEPP-2000, in *Proceedings of the International Particle Accelerator Conference, Kyoto, Japan* (2010), p. 4542, <https://proceedings.jacow.org/IPAC10/papers/thpe014.pdf>.
 - [8] R. Mamutov, G. Baranov, P. Piminov, S. Sinyatkin, and D. Lipoviy, VEPP-4 M linear optics correction using orbit response matrices, *J. Instrum.* **19**, P04026 (2024).
 - [9] V. Lebedev, V. Nagaslaev, A. Valishev, and V. Sajaev, Measurement and correction of linear optics and coupling

- at Tevatron complex, *Nucl. Instrum. Methods Phys. Res., Sect. A* **558**, 299 (2006).
- [10] H. Sugimoto, Y. Ohnishi, A. Morita, and H. Koiso, Superkekb optics tuning and issues, in *Proceedings of 65th ICFA Advanced Beam Dynamics Workshop High Luminosity Circular e^+e^- Colliders, eeFACT'22, Frascati, Italy* (2022), pp. 35–41, <https://proceedings.jacow.org/eeFACT2022/papers/tuxat0103.pdf>.
- [11] A. Romanov, D. Edstrom Jr, F. Emanov, I. Koop, E. Perevedentsev, Y. A. Rogovsky, D. Schwartz, and A. Valishev, Correction of magnetic optics and beam trajectory using loco based algorithm with expanded experimental data sets, [arXiv:1703.09757](https://arxiv.org/abs/1703.09757).
- [12] Y. Alexahin and E. Gianfelice-Wendt, Determination of linear optics functions from turn-by-turn data, *J. Instrum.* **6**, P10006 (2011).
- [13] R. Jones, Measuring tune, chromaticity and coupling, [arXiv:2005.02753](https://arxiv.org/abs/2005.02753).
- [14] L. J. Wang and J. Y. Tang, Luminosity optimization and leveling in the super proton–proton collider, *Radiat. Detect. Technol. Methods* **5**, 245 (2021).
- [15] M. Dohlus, G. Hoffstaetter, M. Lomperski, and R. Wanzenberg, Report from the HERA taskforce on luminosity optimization: Theory and first luminosity scans, Technical Report No. DESY-HERA-03-01, DESY, 2003.
- [16] G. Faletti, Optimisation of LHC integrated luminosity, Master's thesis, University of Bologna, 2021.
- [17] J. Ögren, A. Latina, R. Tomás, and D. Schulte, Tuning the compact linear collider 380 GeV final-focus system using realistic beam-beam signals, *Phys. Rev. Accel. Beams* **23**, 051002 (2020).
- [18] Y. Funakoshi, M. Masuzawa, K. Oide, J. Flanagan, M. Tawada, T. Ieiri, M. Tejima, M. Tobiyama, K. Ohmi, and H. Koiso, Orbit feedback system for maintaining an optimum beam collision, *Phys. Rev. ST Accel. Beams* **10**, 101001 (2007).
- [19] M. STEERING, Luminosity optimization using automated IR steering at RHIC, in *Proceedings of European Particle Accelerator Conference, Lucerne, Switzerland* (2004), <https://proceedings.jacow.org/e04/PAPERS/MOPLT163.PDF>.
- [20] S. White, R. Alemany-Fernandez, M. Lamont, and H. Burkhardt, First luminosity scans in the LHC, Technical Report No. CERN-ATS-2010-096, 2010.
- [21] S. White and H. Burkhardt, Luminosity optimization, in *Evian 2010 Workshop on LHC Commissioning*, <https://cds.cern.ch/record/1281618/files/p71.pdf>.
- [22] S. White, Strategy for luminosity optimisation, in *2nd Evian Workshop on LHC Beam Operation* (2010), https://cds.cern.ch/record/1359182/files/SW_6_03.pdf.
- [23] J. Payet, A. Chancé, O. Napoly, D. Uriot, and S. Auclair, Automatic luminosity optimization of the ILC head-on beam delivery system, in *Proceedings of the 2007 IEEE Particle Accelerator Conference (PAC)* (IEEE, New York, 2007), pp. 1988–1990.
- [24] F. Follin, R. Alemany-Fernandez, and R. Jacobsson, Online luminosity optimization at the LHC, *Energy* **6000**, 7000 (2013), <https://proceedings.jacow.org/ICALEPCS2013/papers/thppc123.pdf>.
- [25] M. Hostettler, R. Alemany, A. Calia, F. Follin, K. Fuchsberger, G. Hemelsoet, M. Gabriel, A. Gorzawski, M. Hruska, D. Jacquet *et al.*, Online luminosity control and steering at the LHC, in *ICALEPCS Proceedings* (2017), <https://cds.cern.ch/record/2305966/files/tush201.pdf>.
- [26] H.-F. Ji, Y. Jiao, S. Wang, D.-H. Ji, C.-H. Yu, Y. Zhang, and X.-B. Huang, Feasibility study of online tuning of the luminosity in a circular collider with the robust conjugate direction search method, *Chin. Phys. C* **39**, 127006 (2015).
- [27] X. Huang, J. Corbett, J. Safranek, and J. Wu, An algorithm for online optimization of accelerators, *Nucl. Instrum. Methods Phys. Res., Sect. A* **726**, 77 (2013).
- [28] S. Gierman, S. Ecklund, R. Field, A. Fisher, P. Grossberg, K. Krauter, E. Miller, M. Petree, K. G. Sonnad, N. Spencer *et al.*, New fast dither system for PEP-II, in *Proceedings of the European Particle Accelerator Conference* (2006), p. 3029, <https://proceedings.jacow.org/e06/PAPERS/THPCH100.pdf>.
- [29] A. S. Fisher, S. Ecklund, R. Field, S. Gierman, P. Grossberg, K. Krauter, E. Miller, M. Petree, K. Sonnad, N. Spencer *et al.*, Commissioning the fast luminosity dither for PEP-II, in *Proceedings of the 2007 IEEE Particle Accelerator Conference (PAC)* (IEEE, New York, 2007), pp. 4165–4167.
- [30] J. Fan and J. Wang, Online reinforcement learning control for maintaining an optimum beam collision on an electron-positron collider, *Phys. Rev. Accel. Beams* **27**, 122801 (2024).
- [31] R. Roussel, A. L. Edelen, T. Boltz, D. Kennedy, Z. Zhang, F. Ji, X. Huang, D. Ratner, A. S. Garcia, C. Xu *et al.*, Bayesian optimization algorithms for accelerator physics, *Phys. Rev. Accel. Beams* **27**, 084801 (2024).
- [32] 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**, 124801 (2020).
- [33] J. Kaiser, C. Xu, A. Eichler, A. Santamaria Garcia, O. Stein, E. Bründermann, W. Kuroepka, H. Dinter, F. Mayet, T. Vinatier *et al.*, Reinforcement learning-trained optimisers and Bayesian optimisation for online particle accelerator tuning, *Sci. Rep.* **14**, 15733 (2024).
- [34] V. Kain, S. Hirlander, B. Goddard, F. M. Velotti, G. Z. Della Porta, N. Bruchon, and G. Valentino, Sample-efficient reinforcement learning for CERN accelerator control, *Phys. Rev. Accel. Beams* **23**, 124801 (2020).
- [35] J. Kaiser, O. Stein, and A. Eichler, Learning-based optimisation of particle accelerators under partial observability without real-world training, in *International Conference on Machine Learning, Baltimore, Maryland, USA* (PMLR, 2022), pp. 10575–10585, <https://proceedings.mlr.press/v162/kaiser22a.html>.
- [36] X. Chen, Y. Jia, X. Qi, Z. Wang, and Y. He, Orbit correction based on improved reinforcement learning algorithm, *Phys. Rev. Accel. Beams* **26**, 044601 (2023).
- [37] T. Boltz, M. Brosi, E. Bründermann, B. Haerer, P. Kaiser, C. Pohl, P. Schreiber, M. Yan, T. Asfour, and A.-S. Müller, Feedback design for control of the micro-bunching instability based on reinforcement learning, in *CERN Yellow Reports: Conference Proceedings* (2020), Vol. 9,

- pp. 227–227, <https://proceedings.jacow.org/ipac2019/papers/mopgw017.pdf>.
- [38] N. Bruchon, G. Fenu, G. Gaio, M. Lonza, F. H. O’Shea, F. A. Pellegrino, and E. Salvato, Basic reinforcement learning techniques to control the intensity of a seeded free-electron laser, *Electronics* **9**, 781 (2020).
- [39] A. Awal, J. Hetzel, R. Gebel, and J. Pretz, Injection optimization at particle accelerators via reinforcement learning: From simulation to real-world application, *Phys. Rev. Accel. Beams* **28**, 034601 (2025).
- [40] D. Y. Wang, H. Bagri, C. Macdonald, S. Kiy, P. Jung, O. Shelbaya, T. Planche, W. Fedorko, R. Baartman, and O. Kester, Accelerator tuning with deep reinforcement learning, in *Workshop on Machine Learning and the Physical Sciences* (2021), https://ml4physicalsciences.github.io/2021/files/NeurIPS_ML4PS_2021_125.pdf.
- [41] P. Piminov, G. Baranov, A. Bogomyagkov, V. Borin, V. Dorokhov, S. Karnaev, K. Y. Karyukina, V. Kiselev, E. Levichev, O. Meshkov *et al.*, VEPP-4 M collider operation at high energy, in *Proceedings of the 12th International Conference on Particle Accelerators, Campinas, SP, Brazil* (2021), pp. 155–158, <https://inspirehep.net/files/1a486893d0aa55d650808f2bc920437b>.
- [42] V. Anashin, V. Aulchenko, E. Baldin, A. Barladyan, A. Y. Barnyakov, M. Y. Barnyakov, S. Baru, I. Y. Basok, I. Bedny, O. Beloborodova *et al.*, The KEDR detector, *Phys. Part Nucl.* **44**, 657 (2013).
- [43] E. B. Levichev, A. N. Skrinksky, Y. A. Tikhonov, and K. Y. Todyshev, High-precision particle mass measurements using the KEDR detector at the VEPP-4 M collider, *Phys. Usp.* **57**, 66 (2014).
- [44] V. Anashin, O. Anchugov, V. Aulchenko, E. Baldin, G. Baranov, A. Barladyan, A. Y. Barnyakov, M. Y. Barnyakov, S. Baru, I. Y. Basok *et al.*, Precise measurement of Ruds and R between 1.84 and 3.72 GeV at the KEDR detector, *Phys. Lett. B* **788**, 42 (2019).
- [45] V. Anashin, V. Aulchenko, E. Baldin, A. Barladyan, A. Y. Barnyakov, M. Y. Barnyakov, S. Baru, I. Y. Basok, O. Beloborodova, A. Blinov *et al.*, Search for narrow resonances in e^+e^- annihilation between 1.85 and 3.1 GeV with the KEDR detector, *Phys. Lett. B* **703**, 543 (2011).
- [46] E. Alpaydin, *Introduction to Machine Learning* (MIT Press, Cambridge, 2020).
- [47] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, 1998), Vol. 1.
- [48] X. Pang, S. Thulasidasan, and L. Rybarcyk, Autonomous control of a particle accelerator using deep reinforcement learning, [arXiv:2010.08141](https://arxiv.org/abs/2010.08141).
- [49] S. Hirlander and N. Bruchon, Model-free and Bayesian ensembling model-based deep reinforcement learning for particle accelerator control demonstrated on the Fermi FEL, [arXiv:2012.09737](https://arxiv.org/abs/2012.09737).
- [50] A. M. de Oñate, N. Bruchon, V. Kain, M. Schenk, and B. R. Mateos, Trajectory steering for DC beams at the CERN SPS using reinforcement learning based on intensity measurements, in *Proceedings of the 16th International Particle Accelerator Conference (IPAC’25), Taipei, Taiwan* (2025), <https://meow.elettra.eu/81/pdf/THPM113.pdf>.
- [51] T. Balasooriya, S. Yoo, V. Schoefer, H.-H. Tseng, Y. Gao, W. Lin, and C. De Silva, Reinforcement learning for charged particle beam control to minimize injection mismatch in particle accelerators, in *Proceedings of the 2025–2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, New York, 2025), pp. 1–5.
- [52] F. AlMahamid and K. Grolinger, Reinforcement learning algorithms: An overview and classification, in *Proceedings of the 2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)* (IEEE, New York, 2021), pp. 1–7.
- [53] J. Fan, Z. Wang, Y. Xie, and Z. Yang, A theoretical analysis of Deep Q-learning, in *Learning for Dynamics and Control* (PMLR, 2020), pp. 486–489, <https://proceedings.mlr.press/v120/yang20a>.
- [54] L. Graesser and W. L. Keng, *Foundations of Deep Reinforcement Learning: Theory and Practice in Python* (Addison-Wesley Professional, 2019).
- [55] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, Playing Atari with deep reinforcement learning, [arXiv:1312.5602](https://arxiv.org/abs/1312.5602).
- [56] M. Sewak, Deep Q network (DQN), double DQN, and dueling DQN: A step towards general artificial intelligence, in *Deep Reinforcement Learning: Frontiers of Artificial Intelligence* (Springer, New York, 2019), pp. 95–108.
- [57] N. Sanghi, *Deep Reinforcement Learning with Python* (Springer, New York, 2021).
- [58] I. D. Mienye and T. G. Swart, A comprehensive review of deep learning: Architectures, recent advances, and applications, *Information* **15**, 755 (2024).
- [59] T. Makosso, A. Almaktoof, and K. Abo-Al-Ez, Review of different types of neural network architectures, *Int. J. Electr. Eng. Appl. Sci.* **7**, 47 (2024).
- [60] H. T. Ünal and F. Başçiftçi, Evolutionary design of neural network architectures: A review of three decades of research, *Artif. Intell. Rev.* **55**, 1723 (2022).
- [61] P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, An experimental review on deep learning architectures for time series forecasting, *Int. J. Neural Syst.* **31**, 2130001 (2021).
- [62] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, *Adv. Neural Inf. Process Syst.* **30**, 5998 (2017).
- [63] A. Sherstinsky, Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network, *Physica (Amsterdam)* **404D**, 132306 (2020).
- [64] J. Kim, H. Kim, H. Kim, D. Lee, and S. Yoon, A comprehensive survey of deep learning for time series forecasting: Architectural diversity and open challenges, *Artif. Intell. Rev.* **58**, 1 (2025).
- [65] J. Sieber, C. Amo Alonso, A. Didier, M. Zeilinger, and A. Orvieto, Understanding the differences in foundation models: Attention, state space models, and recurrent neural networks, *Adv. Neural Inf. Process Syst.* **37**, 134534 (2024). [arXiv:2405.15731](https://arxiv.org/abs/2405.15731).
- [66] F. Stolzenburg, S. Litz, O. Michael, and O. Obst, The power of linear recurrent neural networks, *Neural Comput. Appl.* **37**, 27027 (2025).

- [67] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, A survey of large language models, [arXiv:2303.18223](https://arxiv.org/abs/2303.18223).
- [68] W. Tang, G. Long, L. Liu, T. Zhou, J. Jiang, and M. Blumenstein, Rethinking 1D-CNN for time series classification: A stronger baseline, in *The Tenth International Conference on Learning Representations (ICLR 2022)* (2022), [arXiv:2002.10061](https://arxiv.org/abs/2002.10061).
- [69] A. Zeng, M. Chen, L. Zhang, and Q. Xu, Are transformers effective for time series forecasting?, in *Proceedings of the AAAI Conference on Artificial Intelligence* (2023), Vol. 37, pp. 11121–11128, <https://ojs.aaai.org/index.php/AAAI/article/view/26317>.
- [70] Z. Ali, A comprehensive overview and comparative analysis of CNN, RNN-LSTM and transformer (2024), Available at SSRN: <https://ssrn.com/abstract=5175090>.
- [71] E. Goman, S. Karnaev, O. Plotnikova, and E. Simonov, The database of the VEPP-4 accelerating facility parameters, in *Proceedings of PCaPAC08, Ljubljana, Slovenia* (JACoW, Geneva, Switzerland, 2008).
- [72] The PostgreSQL Global Development Group, PostgreSQL: The world's most advanced open source relational database, <https://www.postgresql.org/>.
- [73] S. E. Karnaev, Control systems for the VEPP-4 accelerator complex and the NSLS-II light source booster synchrotron, Ph.D. thesis, Budker Institute of Nuclear Physics SB RAS, Novosibirsk, Russia, 2017, <https://www.disscat.com/content/sistemy-upravleniya-uskoritelnym-kompleksom-vepp-4-i-busternym-sinkhrotronom-istochnika-si>.
- [74] R. Mamutov and G. Baranov, Software development for automation of particle accelerator magneto-optical structure correction, *Instrum. Exp. Tech.* **67**, S58 (2024).
- [75] EPICS Community, EPICS: Experimental physics and industrial control system, <https://epics-controls.org/>.
- [76] C. F. Worel, The curse of dimensionality: When too many features break your machine learning model, [10.2139/ssrn.5206704](https://ssrn.com/abstract=5206704) (2023).
- [77] I. Ramos-Pérez, J. A. Barbero-Aparicio, A. Canepa-Oneto, Á. Arnaiz-González, and J. Maudes-Raedo, An extensive performance comparison between feature reduction and feature selection preprocessing algorithms on imbalanced wide data, *Information* **15**, 223 (2024).
- [78] L. Budach, M. Feuerpfel, N. Ihde, A. Nathansen, N. S. Noack, H. Patzlaff, H. Harmouch, and F. Naumann, The effects of data quality on ML-model performance, *Inf. Syst.* **132**, 102549 (2025).
- [79] M. A. Hall, Correlation-based feature selection for machine learning, Ph.D. thesis, The University of Waikato, 1999.
- [80] M. M. Mukaka, A guide to appropriate use of correlation coefficient in medical research, *Malawi Med. J.* **24**, 69 (2012), <https://www.ajol.info/index.php/mmj/article/view/81576>.
- [81] F. Song, Z. Guo, and D. Mei, Feature selection using principal component analysis, in *Proceedings of the 2010 International Conference on System Science, Engineering Design and Manufacturing Informatization* (IEEE, New York, 2010), Vol. 1, pp. 27–30.
- [82] F. Rahmat, Z. Zulkafli, A. J. Ishak, R. Z. Abdul Rahman, S. D. Stercke, W. Buytaert, W. Tahir, J. Ab Rahman, S. Ibrahim, and M. Ismail, Supervised feature selection using principal component analysis, *Knowl. Inf. Syst.* **66**, 1955 (2024).
- [83] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences* (Routledge, New York, 2013), [10.4324/9780203771587](https://doi.org/10.4324/9780203771587).
- [84] S. Panjeh, A. Nordahl-Hansen, and H. Cogo-Moreira, Establishing new cutoffs for Cohen's d: An application using known effect sizes from trials for improving sleep quality on composite mental health, *Int. J. Methods Psychiat. Res.* **32**, e1969 (2023).
- [85] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, Scaling laws for neural language models, [arXiv:2001.08361](https://arxiv.org/abs/2001.08361).
- [86] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark *et al.*, Training compute-optimal large language models, [arXiv:2203.15556](https://arxiv.org/abs/2203.15556).
- [87] Y. Bansal, B. Ghorbani, A. Garg, B. Zhang, C. Cherry, B. Neyshabur, and O. Firat, Data scaling laws in NMT: the effect of noise and architecture, in *Proceedings of the International Conference on Machine Learning, Baltimore, Maryland, USA* (PMLR, 2022), pp. 1466–1482, <https://proceedings.mlr.press/v162/>.
- [88] N. C. Thompson, K. Greenewald, K. Lee, G. F. Manso *et al.*, The computational limits of deep learning, [arXiv:2007.05558](https://arxiv.org/abs/2007.05558).
- [89] J. Xiong, Q. Wang, Z. Yang, P. Sun, L. Han, Y. Zheng, H. Fu, T. Zhang, J. Liu, and H. Liu, Parametrized Deep Q-networks learning: Reinforcement learning with discrete-continuous hybrid action space, [arXiv:1810.06394](https://arxiv.org/abs/1810.06394).
- [90] S. R. Sinclair, S. Banerjee, and C. L. Yu, Adaptive discretization in online reinforcement learning, *Oper. Res.* **71**, 1636 (2023).
- [91] J. Cao, L. Dong, and C. Sun, Hierarchical reinforcement learning for kinematic control tasks with parameterized action spaces, *Neural Comput. Appl.* **36**, 323 (2024).
- [92] M. Hutsebaut-Buyse, K. Mets, and S. Latré, Hierarchical reinforcement learning: A survey and open research challenges, *Mach. Learn. Knowl. Extract.* **4**, 172 (2022).