ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10889501

Reinforcement Learning for Charged Particle Beam Control to Minimize Injection Mismatch in Particle Accelerators

Thilina Balasooriya
Department of Computer Science
Columbia University
New York, NY
tnb2119@columbia.edu

Shinjae Yoo
Artificial Intelligence Department
Brookhaven National Laboratory
Upton, NY
sjyoo@bnl.gov

Vincent Schoefer
Collider-Accelerator Department
Brookhaven National Laboratory
Upton, NY
schoefer@bnl.gov

Huan-Hsin Tseng
Artificial Intelligence Department
Brookhaven National Laboratory
Upton, NY
htseng@bnl.gov

Yuan Gao
Collider-Accelerator Department
Brookhaven National Laboratory
Upton, NY
ygao@bnl.gov

Weijian Lin
Collider-Accelerator Department
Brookhaven National Laboratory
Upton, NY
wlin1@bnl.gov

Chanaka De Silva
Collider-Accelerator Department
Brookhaven National Laboratory
Upton, NY
desilva@bnl.gov

Abstract—**Particle accelerators are composed of various components, and their properties are finely tuned to optimize certain particle beam qualities as they accelerate. In particular, particle colliders like the Relativistic Heavy Ion Collider (RHIC) at Brookhaven National Lab (BNL) are interested in maximizing luminosity, a measure of the collision rate primarily determined by the beam intensity (number of particles) and its beam size. However, finding and maintaining optimum settings is a time-consuming expert operator activity. This work proposes the use of the Recurrent Proximal Policy Optimization (RPPO), a reinforcement learning algorithm, to find parameters of quadrupole magnet strengths optimizing the beam qualities in the Booster to AGS (Alternating Gradient Synchrotron) section of the RHIC complex.**

I. Introduction

In the field of accelerator physics, machine learning is being increasingly used to optimize control of accelerator subprocesses. Reinforcement Learning (RL) is particularly fit for particle beam control optimization problems due to accessibility to robust simulations and ample performance metrics to determine agent rewards and goals.

The RHIC at BNL focuses on colliding beams of various heavy ions with high luminosity (rate of collision) in order to study the properties of fundamental matter like quarks and gluons. The beam size (denoted as σ) is a measure of the transverse width of a particle beam and is critical to luminosity because colliding narrower (and therefore denser) beams of particles increases the probability of the particles themselves colliding. Charged particle beams are controlled through the RHIC complex with various magnets, electric fields, and other components. This work is concerned specifically with the Booster to Alternating Gradient Synchrotron (BtA) transfer line, a section of the RHIC complex that transfers the particle beam between the Booster and AGS, two circular synchrotron accelerators within the complex. During transfer between synchrotrons, the optical focusing characteristics of the beam must be "matched", (see Sec. II), and will otherwise cause beam size growth [1]. Tuning magnet strengths to minimize optical mismatch is a complex task requiring expert control. Adjusting multiple magnets simultaneously is a high-dimensional challenge, and efficiency is crucial due to limited beam time and noisy performance metrics. Currently, in the BtA, gradient descent algorithms are used in simple simulations to find initial settings, which are then manually fine-tuned for improved efficiency. Previous studies have explored various AI-based approaches to optimizing accelerator control, with Bayesian Optimization (BO) and RL emerging as the most successful methods [2], [3]. BO is generally more sample-efficient and feasible for online learning, while RL, though requiring a simulation, is more suited for continuous control problems [2]. BO has been successfully applied to accelerator optimizations, including photoinjector beamlines [4], laser-plasma accelerators [5], and beam transfer lines [6].

RL, though less explored [2], has also demonstrated success in accelerator control processes, such as beam envelope optimization [7] and transverse beam tuning [8], among others [9]–

[11]. Awal, Hetzel, et al. [12] used a modified Soft-Actor Critic RL algorithm to minimize injection mismatch, achieving more efficient optimization than human experts, though their approach relied on destructive measurements. In the AGS, beam size is monitored using nondestructive but noisier methods [13]. Therefore, this work proposes using Recurrent Proximal Policy Optimization (RPPO) to optimize particle beam control, minimizing injection mismatch with memory-based neural networks that account for daily fluctuations in accelerator operation, using only beam size in the reward function to preserve the beam.

II. Booster to AGS Transfer Line

A. BtA Background and Components

In an accelerator, there are several types of magnets responsible for different navigational effects on the beam, but this work aims to optimize the quadrupole (or focusing) magnets. As the beam passes through a quadrupole, it focuses the particles in the edge of the beam towards the beam pipe center.

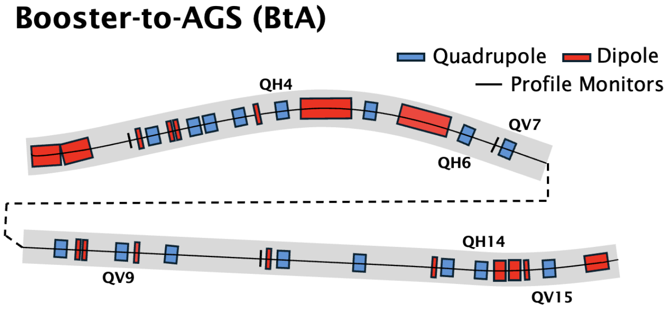

Fig. 1. Booster to AGS transfer line schematic

The BtA is the line that transfers a particle beam from the Booster synchrotron to the AGS and contains 15 quadrupole magnets (among other components). These magnets confine the particles within the beam pipe aperture and transversely shapes the beam to be accepted into the AGS. This work is interested in optimizing the strengths for a subset of these magnets in order to minimize AGS injection mismatch.

B. Injection into AGS - Mismatch Effects

A set of chosen strengths for the quadrupoles in the BtA will affect the particle beam motion (or dynamics). We parameterize the beam motion using optical parameters (β, α, D, D'). The parameters β, α characterize the envelope of the particle trajectories through the BtA and their slopes. The parameters D, D' characterize dispersive effects due to variation in particle momenta [14]. They are defined separately for the horizontal and vertical planes (denoted by subscript x and y). In the BtA, we may assume $D_y = D'_y = 0$, so our beam model is fully characterized by 6 parameters: $(\beta_x, \beta_y, \alpha_x, \alpha_y, D_x, D'_x)$.

This work deals with two main kinds of injection mismatch, 1) amplitude mismatch and 2) dispersion mismatch. The following equations [1] quantify the ratios of beam growth caused by these effects, where σ_0 is the initial beam size and σ_r is the beam size after injection.

1) Amplitude Mismatch:

$$\frac{\sigma^2}{\sigma_0^2} = 1 + \frac{1}{2}\left|\det \Delta J\right| \geq 1 \tag{1}$$

2) Dispersion Mismatch:

$$\frac{\sigma^2}{\sigma_0^2} = 1 + \frac{1}{2}\left(\frac{\Delta D^2 + (\beta \Delta D' + \alpha \Delta D)^2}{\sigma_0^2}\right)\sigma_p^2 \tag{2}$$

where $\Delta D := D^* - D$ denotes the deviation of D from the optimal D^* for injection, and similar for $\Delta D' := D'^* - D'$ and $\Delta J := J^* - J$ (defining $\gamma = (1+\alpha^2)/\beta$),

$$J = \begin{pmatrix} \alpha & \beta \\ -\gamma & -\alpha \end{pmatrix}, \quad J^* = \begin{pmatrix} \alpha^* & \beta^* \\ -\gamma^* & -\alpha^* \end{pmatrix}$$

The goal of this work is to find a set of quadrupole strengths that minimize these ratios Eq. (1), (2) to 1 such that no beam size growth takes place.

III. Methodology

A. Simulation using Bmad and Tao

To simulate the behavior of the BtA we use Bmad [15], a toolkit for simulating charged particle dynamics, together with Tao, a general purpose simulator. We use existing lattice files that describe the beam dynamics of the BtA and dynamically modify them using PyTao, a python library that interfaces with Tao. Specifically, we use it to modify quadrupole magnet strengths, described using current in Amps. Tao provides measurements for each of the optical parameters inside of the AGS, and these measurements are used with Eq. (1), (2) to calculate a simulated beam size in each transverse direction (σ_x and σ_y) and resultant beam size σ_r.

$$\begin{gathered}\frac{\sigma_x^2}{\sigma_{0x}^2} = 1 + \frac{1}{2}|\det \Delta J_x| + \frac{\Delta D_x^2 + (\beta_x \Delta D'_x + \alpha_x \Delta D_x)^2}{2\,\sigma_{0x}^2}\sigma_{px}^2 \\ \sigma_y^2 = \sigma_{0y}^2\left(1 + \frac{1}{2}|\det \Delta J_y|\right) \\ \sigma_r := \sqrt{\sigma_x^2 + \sigma_y^2}, \qquad \sigma_0 := \sqrt{\sigma_{0x}^2 + \sigma_{0y}^2}\end{gathered} \tag{3}$$

Operationally reasonable values are used for nominal beam values σ_{0x}, $\sigma_{px}$and σ_{0y}. We assume perfect knowledge of σ_0 with no measurement error because we can make an arbitrary number of measurements upstream from injection to reduce the measurement uncertainty.

B. Magnet Sensitivity Study: Feature selection for RL

The beam size growth is governed by six independent optical parameters β_x, β_y, α_x, α_y, D_x, and D'_x, as expressed by a function η (composite function of components of Eq. (3)):

$$\rho := \frac{\sigma_r}{\sigma_0} = \eta(\beta_x, \beta_y, \alpha_x, \alpha_y, D_x, D'_x) = \xi(15 \text{ magnets}) \tag{4}$$

Since beam size also depends on 15 quadrupole magnets via ξ (describing the simulation process), these magnets are not independent. We aim to reduce the system to a set of 6 magnets to provide a new function $\widetilde{\xi}$ approximating the functional range of ξ such that,

$$\rho = \xi(15 \text{ magnets}) \cong \widetilde{\xi}(\text{some } 6 \text{ magnets}) \tag{5}$$

Though exact equality is unlikely, we seek 6 magnets that capture most of the range. Using a feature selection method based on the GENFEATWEIGHT [16], we identify the 6 magnets that account for the most variance in the optical parameter space. We assume the magnets independently affect the output and compute the Jacobian matrix $J = \frac{\partial O}{\partial M}$, where O represents output parameters and M the magnets. Ridge regression is then applied to identify the most significant magnet weights, selecting the top six. The process is described by **Algorithm 1** below. The optimal $\alpha^* = 7.54 \times 10^{-8}$ for ridge regression is found using the L-curve method [17]. We conclude through this process that the most influential quadrupoles are,

$$(\text{QV7, QV9, QV15, QH4, QH6, QH14}) = (c_1, \dots, c_6) = \mathbf{c} \tag{6}$$

(see Fig. 1) which are used in the states of the RL environment.

Algorithm 1 Feature Selection Algorithm (adapted from GENFEATWEIGHT in [16])

```
**Input:** $J \in \mathbb{R}^{m \times n}$, Jacobian matrix
**Output** $\vec{w} \in \mathbb{R}^m$, importance weightings for $m$ features
$U\Sigma V^\top \leftarrow \textsc{Svd}\,(J)$
$X \leftarrow \arg\min_{X=(x_{i,j})} \|J^\top X - V\|_F^2 + \alpha \sum_{i,j} |x_{i,j}|^2$
$\forall i \in [m]\,, w_i = \max_{1 \le h \le n} |x_{i,h}|$
```

C. RL Formulation and Gym Environment

To apply RL for BtA particle beam control, the Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, \gamma, \mathcal{P}, \mathcal{R})$ is chosen as follows. Define the state space $\mathcal{S} = \{(\mathbf{c}; \sigma_r) \in \mathbb{R}^7\}$ with $\mathbf{c}$ in Eq. (6) and σ_r from Eq. (3), action space $\mathcal{A} = \{(a_1, \dots, a_6) \,|\, a_i \in [-5, 5]\}$, and the transition (probability) $P : S \times A \times S \to \mathbb{R}^+$,

$$P(\mathbf{s}^{(t)}, \mathbf{a}^{(t)}, \mathbf{s}^{(t+1)}) := \delta\left(\mathbf{c}^{(t)} + \mathbf{a}^{(t)}, \mathbf{c}^{(t+1)}\right) \tag{7}$$

where $\mathbf{c}^{(t)}, \mathbf{c}^{(t+1)}$ are the magnetic current components of $\mathbf{s}^{(t)}, \mathbf{s}^{(t+1)} \in \mathcal{S}$ respectively, and $\mathbf{s}^{(t)}, \mathbf{s}^{(t+1)}$ are states of time step t and $t+1$, $\gamma \in [0, 1)$, $\mathcal{R} : \mathcal{S} \to \mathbb{R}$ is a reward defined and discussed later in Eq. 8, and $\delta(\cdot, \cdot)$ is a Kronecker delta function in Eq. (7), which essentially depicts the interaction of the state and the action by the additive manner: $\mathbf{c}^{(t+1)} = \mathbf{c}^{(t)} + \mathbf{a}^{(t)}$, or sometimes we denote $\mathbf{a}^{(t)} = \Delta\mathbf{c}^{(t)} := \mathbf{c}^{(t+1)} - \mathbf{c}^{(t)}$. That is, an action *only* interacts with the first 6 components of a state. Technically speaking, σ_r is not independent from the magnetic current $\mathbf{c}$, but empirical investigations suggest that the inclusion of σ_r is helpful for seeking optimal configurations.

An agent (or a policy) $\pi : \mathcal{S} \to \mathcal{A}$ yields a trajectory starting from $s^{(1)}$ in $\mathcal{S}$: $\mathbf{s}^{(1)} \to \mathbf{s}^{(2)} \to \cdots \to \mathbf{s}^{(T)}$ where the trajectory is also called an *episode* in RL, and the length of the episode is T. For this work, $T = 10$ is set. While the original objective of the RL is to find an optimal agent π^*, the focus of our interest lies more in achieving the optimal final state $\mathbf{s}^*$. In the end, we seek the states $\{\mathbf{s}^{(t)}\}$ that satisfy the condition $\pi(\mathbf{s}^{(t)}) = \mathbf{a}^{(t)} \to 0$ as $t \gg 1$ and the optimal states are defined as $\mathbf{s}^* = \lim_{t \gg 1} \mathbf{s}^{(t)}$.

Notably, the choice of our MDP imposes the constraint $|\mathbf{c} - \mathbf{c}^{(1)}| \le 50$ Amp for any $\mathbf{c}$ such that the beam settings are not drastically unsuitable for use in the real machine.

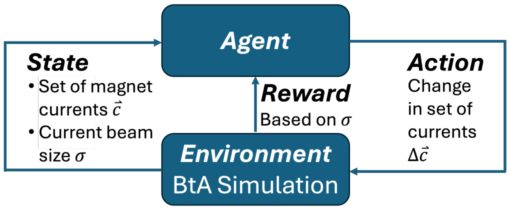
Agent
State
• Set of magnet currents $\vec{c}$
• Current beam size σ
Reward
Based on σ
Action
Change in set of currents $\Delta\vec{c}$
Environment
BtA Simulation

Fig. 2. BtA RL formulation diagram

Environment: The RL environment was developed with the Gymnasium package (formerly OpenAI Gym) [18] and integrates the PyTao BtA simulation. To ensure robustness and scalability to real-world scenarios, we apply domain randomization, a technique commonly used to facilitate simulation-to-reality transfer [12], [19]. Variations in weather, minor maintenance, and other random factors can affect beam dynamics. We simulate these effects by randomizing the initial optical parameters O with noise from a triangular distribution $\mathcal{T}\left((1-h)O,\, O,\, (1+h)O\right)$, where h is set to cause up to 20% beam size growth (a reasonable estimate).

Typical measurement errors in the AGS, are simulated using Gaussian noise $\mathcal{N}(0, \mu\, \sigma_0)$ added to each σ_r measurement in the simulation, where μ represents the noise percentage. To mitigate noise effects, we use a 15% trimmed mean of 20 samples for each measurement.

The Reward: $\mathcal{R} : \mathcal{S} \to \mathbb{R}$ is defined as the composition of $\widetilde{\mathcal{R}} : \mathbb{R} \to \mathbb{R}$ and the beam ratio ρ such that $\mathcal{R}(\mathbf{s}) = \widetilde{\mathcal{R}} \circ \rho(\mathbf{s})$. Recall $\rho = \frac{\sigma_r}{\sigma_0}$ is related to the initial size σ_0 and the final σ_r, which is the consequence of a state $\mathbf{s}$ by non-trivial simulations $\sigma_r = \sigma_r(\mathbf{s})$. Ideally, we want the optimal result $\rho_* = 1$, and thus, we define a "good" ratio to be in the range $\rho \le 1.15$.

Two main considerations were made in shaping the reward function: **1)** The agent must be able to precisely converge to near-optimal ratios and **2)** not become stranded in the large space of bad policies. The rationale is similar to that of the leaky ReLu activation function [20], which is to aid convergence by stopping neurons from becoming inactive from negative input values. It allows positive values to proportionally influence the output, while negative values still have some gradient to deter neuron death. Similarly, we would like good ratio values ($\rho \le 1.15$) to proportionally and aggressively influence the gradient policy update while worse ratio values still have some gradient to prevent agent confusion. To this end, we design the function $\widetilde{\mathcal{R}}$ on ρ,

$$\widetilde{\mathcal{R}}(\rho) = \begin{cases} -66.67\,(\rho - 1) + 10 & \text{if } \rho \le 1.15 \\ -k\rho + b & \text{otherwise} \end{cases} \tag{8}$$

where the constant b is determined by the continuity condition and k is a small positive value for a gentle slope.

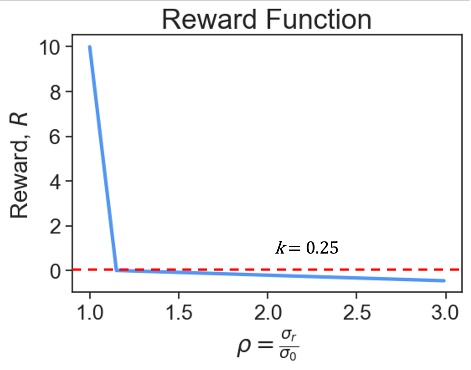

Fig. 3. Piecewise function of $\widetilde{\mathcal{R}}$.

For clarity in evaluation, the range of $\widetilde{\mathcal{R}} \in [0, 10]$ on $\rho \leq 1.15$ is designed to have $\widetilde{\mathcal{R}}(\rho_*) = 10$ when the optimal $\rho_* = 1$ is achieved. Empirically, it is found that $k = 0.25$ is suitable for the agent to learn a descent policy. Other choices of (k, b) are possible.

D. The RL Algorithm

We employ Stable Baselines3 [21] for our RL algorithm, specifically utilizing the Recurrent Proximal Policy Optimization (RPPO) from the sb3-contrib library, which includes community-contributed algorithms based on Stable Baselines3. Vanilla Proximal Policy Optimization (PPO) is an actor-critic deep RL algorithm that uses neural networks to estimate both a value function (evaluating policy value) and a policy function (producing the policy), updating their weights to converge on an optimal policy [22]. RPPO enhances PPO by incorporating Long Short-Term Memory (LSTM) architecture, which improves the model's temporal memory.

RPPO was selected for its LSTM-based architecture in both policy and value networks, enabling the model to track hidden states over time. This allows the agent to use intermediate rewards to infer hidden domain elements (e.g., temperature effects, maintenance) and converge on an improved solution for specific domain instances.

IV. Experiments

A. Simulation setup

The agent was trained across environments with varying measurement noise levels: 0, 5, 10, 15, 20, 25, 30%, logging the mean reward over 100 episodes. Each agent was trained for 750k timesteps (sufficient for convergence) and repeated 5 times with 5 standardized random seeds per noise level for consistency and reproducibility. The RPPO architecture for both value and policy networks consists of 2 LSTM layers with 256 units and 3 hidden layers with 64 neurons. Notable hyperparameters were: learning rate 5×10^{-4}, clip range 0.1, $\gamma = 0.99$, entropy coefficient 0.01, batch size 256, buffer size 1024. Vanilla PPO served as the baseline.

B. Training Convergence

In Fig. 4, RPPO demonstrates stable convergence to near-optimal values (maximum episode reward is 100) for increasing levels of measurement noise. We observe higher variation in the mean reward as measurement noise is increased but a limited effect on the overall convergence behavior.

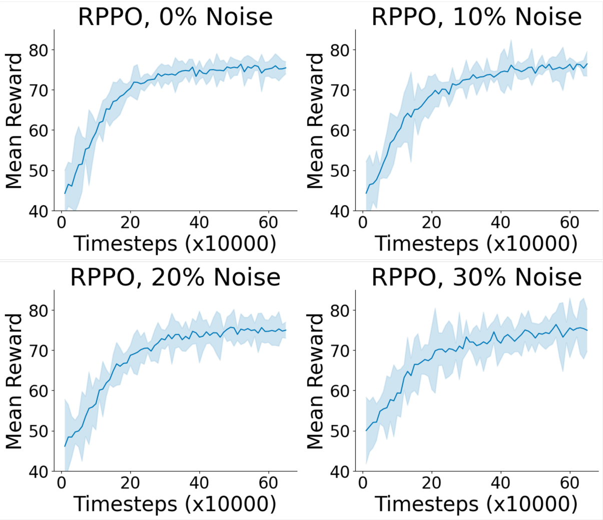

Fig. 4. Mean reward for different noise levels (0, 10, 20, 30%) with variance bands.

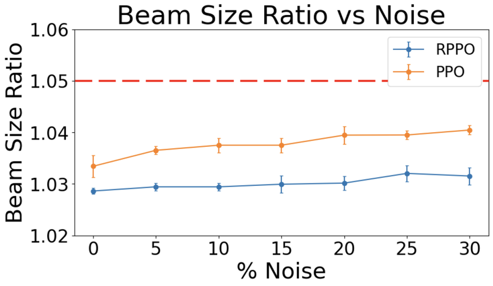

Fig. 5. Effect of measurement error on evaluation beam size

C. Noise Resilience

Fig. 5 shows the minimal impact of measurement noise, with all results within The red line represents a 5% beam size increase (ratio of 1.05), the practical limit below which improvements are indistinguishable due to instrumentation error. Measuring performance against this benchmark ensures results are both meaningful and near the practical optimum. Vanilla PPO achieved less optimal beam size ratios, highlighting the value of memory-based training for learning BtA hidden states.

V. Conclusion

In this work, we demonstrate the use of RL and RPPO architecture for robust optimization of the Booster to AGS transfer line for accelerator tuning. Based on data from previous RHIC runs, emittance at the AGS injection point (end of the BtA) increases by 20-25% within 2-3 days without expert intervention. A robust RL system capable of autonomously limiting beam size growth to below 5% (Fig. 5) outperforms state-of-the-art methods, preserving beam quality and enhancing luminosity at downstream collision points.

VI. Acknowledgment

This project was funded in part by the Columbia University Rabi Scholars Research program. This work was also supported by the U.S. DOE, under award DE-SC-0012704 and BNL's LDRD #24-061. This research used resources of the NERSC, under Contract No. DE-AC02-05CH11231 using NERSC award HEP-ERCAP0033786.

References

[1] D.A. Edwards and M.J. Syphers, *An Introduction to the Physics of High Energy Accelerators*, Wiley Series in Beam Physics and Accelerator Technology. Wiley, 2008.

[2] J. Kaiser, C. Xu, and A. Eichler, "Reinforcement learning-trained optimisers and bayesian optimisation for online particle accelerator tuning," vol. 14, no. 1, pp. 15733, Jan 2024.

[3] Verena Kain, Simon Hirlander, Brennan Goddard, Francesco Maria Velotti, Giovanni Zevi Della Porta, Niky Bruchon, and Gianluca Valentino, "Sample-efficient reinforcement learning for cern accelerator control," *Phys. Rev. Accel. Beams*, vol. 23, pp. 124801, Dec 2020.

[4] Boltz Tobias, Martinez Jose L., and Xu Connie et. al, "More sample-efficient tuning of particle accelerators with bayesian optimization and prior mean models," *Arvix*, May 2024.

[5] A. Ferran Pousa, S. Jalas, M. Kirchen, A. Martinez de la Ossa, M. Thévenet, S. Hudson, J. Larson, A. Huebl, J.-L. Vay, and R. Lehe, "Bayesian optimization of laser-plasma accelerators assisted by reduced physical models," *Phys. Rev. Accel. Beams*, vol. 26, pp. 084601, Aug 2023.

[6] Yasuyuki Morita, Takashi Washio, and Yuta Nakashima, "Accelerator tuning method using autoencoder and bayesian optimization," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 1057, pp. 168730, 2023.

[7] David Wang, Harpriya Bagri, Calum Michael Macdonald, Spencer Kiy, Paul Jung, O. Shelbaya, Thomas Planche, Wojciech T. Fedorko, Rick Baartman, and Oliver Kester, "Accelerator tuning with deep reinforcement learning," 2021.

[8] Jan Kaiser, Oliver Stein, and Annika Eichler, "Learning-based optimisation of particle accelerators under partial observability without real-world training," in *Proceedings of the 39th International Conference on Machine Learning*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, Eds. 17–23 Jul 2022, vol. 162 of *Proceedings of Machine Learning Research*, pp. 10575–10585, PMLR.

[9] Andrea Santamaria Garcia, Annika Eichler, Chenran Xu, Jan Kaiser, Luca Scomparin, Michael Schenk, Sabrina Pochaba, and Simon Hirlaender, "The reinforcement learning for autonomous accelerators collaboration," *JACoW*, vol. IPAC2024, pp. TUPS62, 2024.

[10] Simon Hirlander, Lukas Lamminger, Sabrina Pochaba, Jan Kaiser, Chenran Xu, Andrea Santamaría García, Luca Scomparin, and Verena Kain, "Towards few-shot reinforcement learning in particle accelerator control," 05 2024.

[11] Linh Nguyen, Kevin Brown, Michael Costanzo, Yuan Gao, Margaret Harvey, James Jamilkowski, John Morris, and Vincent Schoefer, "A Physics-Based Simulator to Facilitate Reinforcement Learning in the RHIC Accelerator Complex," *JACoW*, vol. ICALEPCS2023, pp. FR2AO04, 2023.

[12] Awal Awal and Jan Hetzel et. al, "Injection optimization at particle accelerators via reinforcement learning: From simulation to real-world application," *Arvix*, June 2024.

[13] H. Weisberg, E. Gill, P. Ingrassia, and E. Rodger, "An ionization profile monitor for the brookhaven ags," *IEEE Transactions on Nuclear Science*, vol. 30, no. 4, pp. 2179–2181, 1983.

[14] E.D Courant and H.S Snyder, "Theory of the alternating-gradient synchrotron," *Annals of Physics*, vol. 3, no. 1, pp. 1–48, 1958.

[15] D. Sagan, "Bmad: A relativistic charged particle simulation library," *Nucl. Instrum. Meth.*, vol. A558, no. 1, pp. 356–359, 2006, Proceedings of the 8th International Computational Accelerator Physics Conference.

[16] Hao Huang, Shinjae Yoo, and Shiva Prasad Kasiviswanathan, "Unsupervised feature selection on data streams," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, New York, NY, USA, 2015, CIKM '15, p. 1031–1040, Association for Computing Machinery.

[17] D. Calvetti, S. Morigi, L. Reichel, and F. Sgallari, "Tikhonov regularization and the l-curve for large discrete ill-posed problems," *Journal of Computational and Applied Mathematics*, vol. 123, no. 1, pp. 423–446, 2000, Numerical Analysis 2000. Vol. III: Linear Algebra.

[18] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[19] Joshua Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," *CoRR*, vol. abs/1703.06907, 2017.

[20] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li, "Empirical evaluation of rectified activations in convolutional network," *CoRR*, vol. abs/1505.00853, 2015.

[21] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.