ISBN: 978-3-95450-248-6 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2025-THYD1

REINFORCEMENT LEARNING IN PARTICLE ACCELERATORS

A. Santamaria Garcia* †, University of Liverpool, Liverpool, United Kingdom
C. Xu‡, Karlsruhe Institute of Technology, Karlsruhe, Germany
A. Eichler§, J. Kaiser, Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany
S. Hirlaender, Paris Lodron Universität Salzburg, Salzburg, Austria

Abstract

Reinforcement learning (RL) is a unique learning paradigm inspired by the behaviour of animals and humans to learn to solve tasks autonomously. Learning occurs through interactions with an environment, exploring, and evaluating strategies under various conditions. RL excels in complex environments, can handle delayed consequences, and is able to learn solely from experience without access to an explicit model of the system. This makes RL particularly promising for particle accelerators, where the dynamic conditions of particle beams and accelerator systems require continuous adaptation, and modelling is challenging. Although RL applications are emerging in accelerator physics and showing promising results, their widespread introduction faces critical challenges. Among the main obstacles are the effective formulation of control problems, training, and the deployment of solutions in real systems. This paper provides an overview of the potential of RL in accelerator applications, highlighting current challenges and future research directions.

INTRODUCTION

Machine learning (ML) has already been identified as a key technology that can considerably extend and advance the capabilities of particle accelerators and must be included in their future design [1,2]. There are many popular uses of ML methods in particle accelerators, for example in detection and prediction tasks, but optimisation remains one of the most studied and applied areas, where Bayesian optimisation (BO) and RL have emerged as leading paradigms [3–5].

BO builds probabilistic models of the objective function by efficiently sampling the environment and exploits these models to guide the search for optimal solutions in a data-efficient manner. However, if the conditions under which the model was constructed change, the optimisation performance may degrade until a new model is built from fresh observations. Furthermore, BO suffers from inherent limitations in high-dimensional spaces due to the breakdown of distance-based kernel functions, leading to poor generalisation, over-smoothing, and unreliable acquisition optimisation. In contrast, RL used as an optimiser can adapt dynamically to changing environments, scale better to high-dimensional problems, consider delayed consequences, and often converges more rapidly once training is complete. These advantages come at the cost of extensive training through many interactions with the system (simulated or real), depending on the complexity of the problem, and a significant engineering effort to tune the algorithm, in contrast to the more direct deployment possible with BO. Beyond static optimisation tasks, RL also offers the ability to perform closed-loop control by learning policies that map observed states to actions in real time. This enables it to handle dynamic, partially observable, or stochastic systems where conventional optimisation approaches fall short. As a result, RL is increasingly being explored in complex control applications, including fusion reactor control, robotics, industrial process control, energy systems, finance, healthcare, or autonomous driving. These capabilities position RL as a powerful framework for adaptive, goal-directed behaviour in complex environments, such as those in particle accelerators, which can also be exploited for optimisation. This work is not intended as an exhaustive review of the literature, but rather as a reflective overview of the practical and conceptual considerations in using RL for particle accelerator control. A complete introduction to the theory of RL can be found in Ref. [6].

DEEP RL

RL is a computational approach to learning from interaction, which emerged in the 1980s by incorporating concepts from trial-and-error learning, optimal control, and temporal difference methods. Tabular solution methods were the dominant solution strategy in early RL, where the value function is explicitly represented in a table or array. Therefore, these methods are only practical for finite and relatively small Markov decision processs (MDPs), and are ill suited to real world problems, which are typically modelled as continuous and infinite partially observable Markov decision processs (POMDPs). In a POMDP, the agent receives an observation, which provides only partial and potentially noisy information about the true state of the environment. While the underlying system dynamics are Markovian with respect to the true state, decision-making from observations alone requires memory to track the evolving belief. Although optimal solutions to POMDPs exist in theory, they are typically computationally intractable in practice due to the high dimensionality of the belief space and the cost of planning under uncertainty. This is why policy and value functions must be approximated for real-world environments, where we look for policies that are robust, generalisable, and sample-efficient. Tabular methods were quickly replaced by function approximation methods, such as linear or polynomial approximators. While they

* ansantam@liverpool.ac.uk
† also at the Cockcroft Institute, Warrington, United Kingdom
‡ now at Argonne National Laboratory, Lemont, IL, USA
§ also at Hamburg University of Technology, TUHH, Germany

ISBN: 978-3-95450-248-6 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2025-THYD1

enabled RL to scale beyond tabular settings, their limited expressivity and dependence on manual features constrained their use in complex or high-dimensional tasks. Deep reinforcement learning (DRL) surpassed them by learning rich non-linear representations directly from raw data, enabling breakthroughs in domains such as vision-based control, robotics, and game play [7–9]. In DRL, the policy and value functions are approximated by deep neural networks (DNNs). This significant change was driven by modern computational capabilities and kicked off the development of advanced DRL algorithms [10]. For example, in policy-based RL, a widely used class of modern RL algorithms, the weights of the DNN policy are updated by ascending the gradient of expected cumulative reward, typically estimated using sampled trajectories.

Challenges in Deep RL

Apart from being able to tackle high dimensions, DNNs can learn abstract features and can generalise to states never seen before, since the DNN parameters are shared over all states. However, DRL algorithms introduce a new set of challenges inherited from DNNs. For example, DRL has few to almost no convergence guarantees and introduces additional biases, variance, and stability challenges, while the generalisation capabilities will depend directly on the quantity and quality of the training data. Training stability refers to the ability of the RL agent to learn and improve consistently and predictably over time. One of the main contributors to training instability is using bootstrapped value targets (inherited from temporal difference learning) where the target for one prediction depends on another prediction made by the same network. This can destabilise the learning through feedback loops and the accumulation of errors, and is mitigated by delayed updates. Function approximation bias is also an important factor affecting training stability in DRL. The approximation of the policy and value functions will be influenced by the network architecture, the initialisation of weights (different weights can converge to drastically different policies), the training dynamics (e.g., stochastic gradients), and the distribution of observed data (depending on exploration). *Experience replay* in off-policy algorithms helps mitigate weight initialisation bias by randomly sampling from stored transitions, breaking temporal correlations in the data and stabilising training [9]. Modern on-policy algorithms like Trust Region Policy Optimisation (TRPO) and proximal policy optimisation (PPO) impose different constraints on the gradient update of the policy network (gradient clipping) to improve training stability. Function approximation bias can also be mitigated by ensembling DNNs, which reduces variance and captures uncertainty.

Hyperparameter sensitivity is another critical challenge in DRL due to the high variance in performance across random seeds and the high variance in gradient estimates in policy gradient methods. Without proper hyperparameter tuning the algorithm can be unstable and diverge, prematurely converge to suboptimal behaviour, and fail to explore effectively.

RL IN PARTICLE ACCELERATORS

Despite the challenges presented above, RL has achieved state-of-the-art performance in continuous control tasks, delivering impressive results in complex simulated environments. However, its deployment in real-world systems has been much slower because of the significant differences between both. Particle accelerators are real-world environments that include non-stationary and stochastic systems with partial, filtered, delayed, and noisy signals through diagnostic systems, and often have stringent safety and latency requirements. Additionally, access to experiments on the accelerator is usually limited, imposing a constraint on the time available for training and deployment in the real accelerator. In Ref. [11] the authors identify and review the challenges of deploying RL in real-world systems, which unmistakeably apply to particle accelerators. In the following, we introduce the main challenges in RL deployment in particle accelerators and review a few recent examples from the literature.

Terminology

To ensure clarity moving forward, we outline the distinctions between interaction source (simulation vs. real system), data availability (online vs. offline), and policy consistency (on-policy vs. off-policy). In this paper, we refer to *simulation-based RL* as training agents within a virtual or simulated environment, while *experiment-based RL* involves direct interaction with a real-world or physical system (in our case, a particle accelerator). These categories intersect with the concepts of *online* and *offline* RL, which describe whether the agent is actively collecting new data during learning. In online RL, the agent continually updates its policy based on the experience gathered during training, be it in simulation or on the real system. In contrast, offline RL (or batch RL) learns exclusively from a fixed dataset collected in advance, without any further interaction with the environment. A second, orthogonal distinction exists between *on-policy* and *off-policy* learning. On-policy algorithms update the policy using data collected by the current version of the policy itself, requiring fresh interaction at each iteration (e.g., PPO, A2C). In contrast, off-policy algorithms can learn from data generated by a different behavior policy, including older versions of the current policy or an entirely different agent (e.g., Deep Q-Learning (DQN), Soft Actor Critic (SAC)). This makes off-policy methods especially useful for both replay-based online learning and offline RL.

Challenge 1: Learning from Limited Samples

Collecting data samples is usually expensive in simulation and prohibitive in the real accelerator due to limited beam time availability and the slow generation of data due to low repetition rates. This is why developing sample efficient algorithms is one of the main research directions in RL, where sample efficiency (or sample complexity) refers to the number of interactions with the environment required to achieve a certain level of performance during the decision

ISBN: 978-3-95450-248-6 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2025-THYD1

making process. Improving sample efficiency decreases the training costs of RL algorithms for both simulation and experiment-based training and makes them more accessible.

The sample efficiency of RL algorithms varies significantly depending on their design: on-policy methods such as PPO [12] and REINFORCE [13] tend to require large amounts of interaction due to their inability to reuse past data, while off-policy algorithms like DQN [9] and SAC [14] achieve better sample efficiency through experience replay and bootstrapped value estimation. Model-based and offline approaches further improve sample efficiency by leveraging learned dynamics models or pre-collected datasets to reduce dependence on costly environment interactions. While algorithms like SAC have a high sample efficiency and robust performance in continuous control tasks, they often come with practical challenges that make them harder to use in real-world applications. SAC is particularly sensitive to hyperparameters, including learning rates, target smoothing coefficients, and entropy regularisation terms, and its performance can degrade significantly outside well-tuned settings. These difficulties are compounded in real-world or high-fidelity simulation settings, where a misconfigured SAC algorithm can waste expensive samples due to unstable training, negating its theoretical efficiency. Similarly, model-based reinforcement learning (MBRL) methods, while even more sample-efficient in theory, tend to be brittle in practice. Their performance is tightly coupled to the quality of the learned dynamics model, which is difficult to calibrate, and their training pipelines typically involve many hyperparameters like model updates, planning horizons, rollout lengths, and uncertainty thresholds that interact in complex ways [15]. As a result, both SAC and MBRL algorithms demand significant expertise, experimentation time, and tuning effort, making them difficult to deploy reliably for non-specialists or in time-constrained scientific projects. In contrast, less sample-efficient algorithms like PPO are often preferred in practice as they are more robust to hyperparameter choices and easier to deploy.

Simulation-based Training A common way of working with RL algorithms for accelerators is to perform simulation-based training and later deploy in the real accelerator. This is safer and easier than experiment-based training, but introduces a new challenge: ensuring that the policy trained in simulation can generalise and perform reliably when deployed on the real system, a problem commonly referred to as the *sim2real* gap. The sim2real gap encapsulates the challenge of generalising from idealised or simplified simulations to the complex, noisy, and often unpredictable dynamics of the real world. There are several promising ways to train a robust policy, where *robustness* refers to the agent's ability to maintain high performance under uncertainty or changes in the environment. Robustness is explicitly targeted during the training phase. One of them is domain randomisation (DR) [16], where the agent interacts with a set of environments covering a wide range of possible scenarios in the training process. For example, in optics simulations, the magnet errors and misalignments are often not included or outdated, and particle tracking simulations often use the same Gaussian initial distribution. To account for these variances in real accelerator conditions, one can apply DR by randomising incoming beam parameters and magnet misalignments during training across a distribution of simulated environments. This approach was used in Ref. [4], where a Twin Delayed DDPG (TD3) agent trained under such conditions achieved robust performance when deployed on the real accelerator. By exposing the agent to a wide range of perturbations during training, the real-world environment effectively becomes just another variation within the distribution it has already experienced. In Ref. [17] the authors employ an improved version of TD3, a model-free, off-policy RL algorithm. They performed a robust simulation-based training by augmenting the observation space with BPM signal trends, capturing temporal dynamics and reducing the impact of partial observability. They also adopted a physics-informed network architecture incorporating convolutional layers to extract relevant spatial features. These changes enhanced policy generalisation across beam species, lattice configurations, and hardware imperfections, allowing to bridge the sim2real gap.

Another promising avenue is meta-RL, also called "learning to learn". Meta-RL trains an agent to acquire the ability to rapidly adapt to new but related tasks by leveraging prior experience, mimicking the fast learning capabilities of humans. Rather than learning a single policy tailored to one environment, the objective is to learn an adaptable policy that can quickly specialise to new tasks using only limited interaction or fine-tuning. Meta-RL promotes generalisation across related environments, and is particularly well-suited for sim2real transfer or settings where online learning is essential. However, these benefits come at a cost since meta-training is computationally expensive, often requiring a large number of diverse tasks and significant environment interaction. In Ref. [18, 19] meta-RL is used for trajectory optimisation on the AWAKE electron line in simulation, showing a considerable improvement in performance over classical and DR training.

To fully leverage data-intensive strategies, like DR and meta-RL or just policy-gradient methods, the underlying simulation must support fast and scalable data generation. This has motivated the development of differentiable beam dynamics libraries, like Cheetah [20], and surrogate models based on DNNs that serve as lightweight, trainable environments. In Ref. [21], training was conducted on a custom, differentiable physics-based simulator, tailored to model spill regulation dynamics with realistic variability. In contrast, [22] employed an LSTM-based surrogate model trained directly on real accelerator data to emulate magnet power supply behavior. The work in Ref. [23] utilised a GPU-accelerated physics engine to simulate multi-particle beam transport through a drift-tube linac. Meanwhile, [24] relied on a hand-crafted, physics-informed surrogate environment based on analytic expressions for energy gain, heat load, and

ISBN: 978-3-95450-248-6 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2025-THYD1

arc fault trip rates, providing a structured yet interpretable simulation framework for training and evaluating RL agents.

Experiment-based Training Experiment-based training of RL agents has been demonstrated at multiple accelerators, namely at the FERMI FEL at ELETTRA [15, 25], Linac 4 and AWAKE at CERN [26], and KARA at KIT [27, 28]. In all of the above mentioned cases, the agent learned directly from live samples and required no pre-training in simulation.

The work in Ref. [25] aims to achieve optimal laser alignment from random initial conditions and recovering alignment after drifts. To tackle these, the authors implement two RL algorithms: Q-learning with linear function approximation for the attainment task, and Natural Policy Gradient REINFORCE (NPG-REINFORCE) for drift recovery. Both algorithms proved to be effective in an experiment-based setting. Although REINFORCE is typically considered data-intensive and unsuitable for direct use in physical experiments, its application was feasible here due to the low dimensionality of the task (4 motor voltages), dense and immediate reward signal (FEL intensity), and careful decomposition of the policy into independently controlled action dimensions using Von Mises distributions. Moreover, the use of REINFORCE was restricted to fine-tuning in the vicinity of known working points, effectively functioning as an online adaptation mechanism rather than a full learning-from-scratch algorithm.

The same control problem was tackled in Ref. [15] with three different algorithms. The authors tested NAF2, a stable and sample-efficient model-free variant, and two model-based algorithms, AE-DYNA-SAC and ME-TRPO, trained on Bayesian ensemble models of system dynamics. Despite the traditionally high data requirements of RL, all algorithms succeeded in optimizing the intensity of the FEL in a few steps per episode and using fewer than 1100 real-world data points. The model-based methods demonstrated superior sample efficiency and asymptotic performance, while NAF2 exhibited faster convergence in terms of steps per episode. Success was attributed to the short task horizon, low-dimensional state-action space, and uncertainty-aware modelling.

The work in Ref. [26] presents the real-world deployment of a sample-efficient RL agent using the Normalized Advantage Function (NAF) algorithm for beam trajectory correction at CERN's AWAKE and LINAC4 accelerators. At AWAKE, the agent was trained directly on the machine over ~200 episodes (~350 interactions). During validation, the trained agent consistently reduced trajectory errors to below 2 mm RMS in 1–2 steps. At LINAC4, the agent learned a 16-dimensional control task in ~90 episodes (~300 interactions), reaching <1 mm RMS in up to 3 steps during validation. NAF offered sufficient sample efficiency for real-world deployment due to the structured nature of the task (low-dimensional state and action spaces, dense and smooth reward signals, and short episode horizons) allowing the agent to learn effective control policies within a few hundred interactions. However, its performance was sensitive to hyperparameter choices, which required careful tuning, particularly in safety-constrained environments like LINAC4.

Challenge 2: Partial Observability

Partial observability is a fundamental challenge in applying RL to real-world systems like particle accelerators, where direct access to the full system state is rarely possible. It can manifest through limited sensor coverage, noise, hidden dynamics, and coupled subsystems. When not addressed, it may lead to unstable policies or poor generalisation. However, its impact can be mitigated through thoughtful observation design, incorporation of temporal information, or control architectures that rely on fast, reactive feedback. As demonstrated by Ref. [17], incorporating temporal trends in beam position measurements improves policy robustness and generalisation in sim2real transfer, while Ref. [28] shows that a high-frequency control rate minimises the risk of losing important dynamics between decisions. Nevertheless, the limitations of policies using fully-connected neural networks (NNs) under partial observability can be seen when applied to rapidly changing nonlinear instabilities [27], highlighting the need for more advanced strategies such as recurrent architectures or attention mechanisms to provide the agent with a rich enough representation of the underlying dynamics to anticipate the instability progression. In contrast to highly dynamic instability control problems, partial observability was not a major bottleneck in Refs. [15, 25, 26] due to the combination of a stable physical process, frequent and informative observations, a well-structured state representation, and low-frequency decision-making.

Challenge 3: Safety

Safety in RL is especially critical in real-world, high-stakes applications such as particle accelerator control, where unsafe actions during training or deployment can lead to beam loss and equipment damage. In this context, safety concerns arise in two main areas: the agent must apply a safe policy that consistently avoids harmful actions, and it must conduct safe exploration while interacting with the environment to learn effectively [29]. These challenges are amplified in high-dimensional or continuous state spaces, where guaranteeing safety across all possible states is often infeasible. While techniques such as shielding, reward shaping, and uncertainty-aware planning have been developed to address these issues, they come with trade-offs between safety, optimality, and sample efficiency. Although soft safety using negative rewards has been demonstrated in other fields, such as fusion reactor control [30], the RL community has yet to fully solve the problem of hard safety guarantees, especially during exploration. While action space constraints can be a simple and intuitive way to improve safety in experiment-based training, they do not guarantee overall system safety, especially in the presence of coupled dynamics, delayed effects, or partially observable states. As a result, direct experiment-based training remains difficult in safety-critical tasks (e.g. with high-power beams), and simulation-based

ISBN: 978-3-95450-248-6 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2025-THYD1

approaches combined with conservative deployment strategies are still the preferred approach. Some work addressing safe RL in accelerators can be found here [31].

Challenge 4: Real-time Inference

Conventional RL deployments, where the policy runs on a CPU, can typically support control frequencies around 100 Hz, which is sufficient for many routine optimisation tasks. However, some real-world applications demand much faster response times, often within microseconds to milliseconds, to regulate fast-evolving or unstable dynamics. Meeting these ultra-fast control requirements imposes strict constraints on latency and computational overhead, which requires hardware acceleration such as FPGAs or AI-specific inference engines. The work in Ref. [28] presents a RL control platform for particle accelerators that achieves true real-time performance by deploying a PPO-trained policy on AI Engine architecture, delivering inference latencies as low as 770 ns and a control loop period of just 2.7 µs. This is achieved through an *experience accumulator* scheme that buffers real-time interaction data during live operation of the control loop, enabling PPO training to be conducted on a separate CPU while maintaining ultrafast inference on-chip. This framework was also tested in the control of the microbunching instability [27], which requires actions at a frequency of the order of the synchrotron motion. It is a key innovation that makes experiment-based training with real hardware feasible under stringent timing requirements.

Summary

Figure 1 serves as a visual summary of the main points covered in this section.

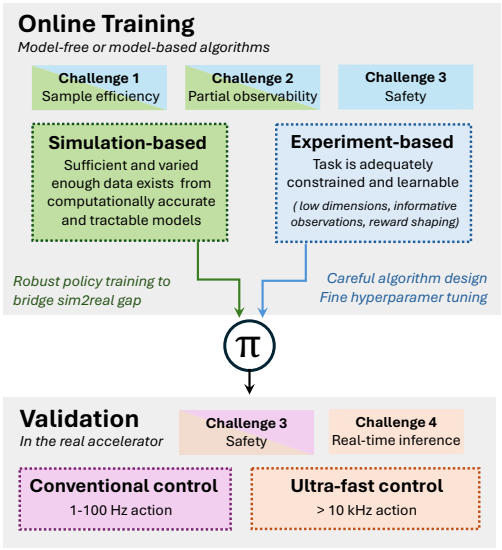

Figure 1: Overview of the main challenges of RL deployment in particle accelerators.

FUTURE DIRECTIONS

Generalisable or transferable RL agents are particularly valuable in the context of particle accelerators, which, unlike robots in the field of robotics, are rarely standardized or mass-produced. This lack of uniformity limits collaboration, as trained agents are often specific to a single machine or task. Consequently, most RL applications in accelerator control remain isolated, one-off efforts. However, accelerators of the same class frequently share common design principles, similar lattice structures, and comparable tuning objectives. These structural and functional similarities present an opportunity to develop *lattice agnostic agents* that generalise across machines performing analogous tasks. Some promising first steps are shown in Ref. [32].

Another promising avenue is the use of multi-agent and hierarchical RL approaches, which are natural candidates for high-dimensional, distributed, and tightly coupled control systems, like the ones in large-scale scientific infrastructures. In multi-agent RL, multiple specialised agents can control different subsystems (e.g., beamlines, magnets, diagnostics) while coordinating to achieve a global objective, such as maximising beam quality or optimising energy efficiency. Hierarchical RL provides different levels of abstraction, with high-level policies selecting subgoals or control modes, with low-level controllers translating strategic decisions into concrete control signals, similar to how human operators structure their actions.

Finally, explainable policies will be essential in safety-critical domains, where the black-box nature of DNNs makes it challenging to understand, trust, or validate an agent's behaviour. Techniques for improving interpretability (such as symbolic rule extraction, decision trees, causal attribution methods, and visualisation) can support human operators by revealing underlying control strategies, facilitating system diagnosis, and fostering trust.

CONCLUSIONS

RL is a vibrant and fast-evolving research field with applications that extend well beyond particle accelerators, offering powerful capabilities in many real-world domains. As the field matures, RL has the potential to deliver flexible, adaptive, and high-performing control strategies that outperform traditional methods in complex, dynamic environments. Continued exploration, especially through interdisciplinary collaboration between domain experts and RL practitioners like Ref. [5], will be crucial to unlocking this potential and realising safe and impactful applications in large-scale scientific infrastructures and beyond.

REFERENCES

[1] A. Edelen *et al.*, "Opportunities in machine learning for particle accelerators", *arXiv:1811.03172*, 2018. doi:10.48550/arXiv.1811.03172

[2] A. Santamaria Garcia *et al.*, "How can machine learning help future light sources?", in *Proc. FLS'23*, Luzern, Switzerland, pp. 249–256, 2024. doi:10.18429/JACoW-FLS2023-TH3D3

ISBN: 978-3-95450-248-6 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2025-THYD1

[3] R. Roussel *et al.*, "Bayesian optimization algorithms for accelerator physics", *Phys. Rev. Accel. Beams*, vol. 27, no. 8, p. 084 801, 2024. doi:10.1103/PhysRevAccelBeams.27.084801

[4] J. Kaiser *et al.*, "Reinforcement learning-trained optimisers and Bayesian optimisation for online particle accelerator tuning", *Sci. Rep.*, vol. 14, no. 1, p. 15 733, 2024. doi:10.1038/s41598-024-66263-y

[5] A. Santamaria Garcia *et al.*, "The reinforcement learning for autonomous accelerators collaboration", in *Proc. IPAC'24*, Nashville, TN, pp. 1812–1815, 2024. doi:10.18429/JACoW-IPAC2024-TUPS62

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning, second edition: an introduction*. MIT Press, 2018.

[7] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning", *arXiv:1509.02971*, 2019. doi:10.48550/arXiv.1509.02971

[8] D. Kalashnikov *et al.*, "QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation", *arXiv:1806.10293*, 2018. doi:10.48550/arXiv.1806.10293

[9] V. Mnih *et al.*, "Playing atari with deep reinforcement learning", *arXiv:1312.5602*, 2013. doi:10.48550/arXiv.1312.5602

[10] K. Arulkumaran *et al.*, "Deep reinforcement learning: A brief survey", *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017. doi:10.1109/MSP.2017.2743240

[11] G. Dulac-Arnold *et al.*, "Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis", *Mach. Learn.*, vol. 110, pp. 2419–2468, 2021. doi:10.1007/s10994-021-05961-4

[12] J. Schulman *et al.*, "Proximal policy optimization algorithms", *arXiv:1707.06347*, 2017. doi:10.48550/arXiv.1707.06347

[13] J. Zhang *et al.*, "Sample efficient reinforcement learning with reinforce", *arXiv:2010.11364*, 2020. doi:10.48550/arXiv.2010.11364

[14] T. Haarnoja *et al.*, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor", *arXiv:1801.01290*, 2018. doi:10.48550/arXiv.1801.01290

[15] S. Hirlaender *et al.*, "Model-free and Bayesian ensembling model-based deep reinforcement learning for particle accelerator control demonstrated on the FERMI FEL", *arXiv:2012.09737*, 2022. doi:10.48550/arXiv.2012.09737

[16] OpenAI *et al.*, "Solving Rubik's cube with a robot hand", *arXiv:1910.07113*, 2019. doi:10.48550/arXiv.1910.07113

[17] X. Chen *et al.*, "Orbit correction based on improved reinforcement learning algorithm", *Phys. Rev. Accel. Beams*, vol. 26, no. 4, p. 044 601, 2023. doi:10.1103/PhysRevAccelBeams.26.044601

[18] S. Hirlaender *et al.*, "Deep meta reinforcement learning for rapid adaptation in linear Markov decision processes: applications to CERN's AWAKE project", vol. 1458, 2024. doi:10.1007/978-3-031-65993-5_21

[19] S. Hirlaender *et al.*, "Towards few-shot reinforcement learning in particle accelerator control", in *Proc. IPAC'24*, Nashville, TN, pp. 1804–1807, 2024. doi:10.18429/JACoW-IPAC2024-TUPS60

[20] J. Kaiser *et al.*, "Bridging the gap between machine learning and particle accelerator physics with high-speed, differentiable simulations", *Phys. Rev. Accel. Beams*, vol. 27, no. 5, p. 054 601, 2024. doi:10.1103/PhysRevAccelBeams.27.054601

[21] C. Xu *et al.*, "Beyond PID Controllers: PPO with Neuralized PID Policy for Proton Beam Intensity Control in Mu2e", *arXiv:2312.17372*, 2023. doi:10.48550/arXiv.2312.17372

[22] J. St. John *et al.*, "Real-time artificial intelligence for accelerator control: A study at the Fermilab Booster", *Phys. Rev. Accel. Beams*, vol. 24, no. 10, p. 104 601, 2021. doi:10.1103/PhysRevAccelBeams.24.104601

[23] X. Pang *et al.*, "Autonomous control of a particle accelerator using deep reinforcement learning", *arXiv:2010.08141*, 2020. doi:10.48550/arXiv.2010.08141

[24] J. Colen *et al.*, "Explainable physics-based constraints on reinforcement learning for accelerator controls", *arXiv:2502.20247*, 2025. doi:10.48550/arXiv.2502.20247

[25] N. Bruchon *et al.*, "Basic reinforcement learning techniques to control the intensity of a seeded free-electron laser", *Electronics*, vol. 9, no. 5, 2020. doi:10.3390/electronics9050781

[26] V. Kain *et al.*, "Sample-efficient reinforcement learning for CERN accelerator control", *Phys. Rev. Accel. Beams*, vol. 23, no. 12, p. 124 801, 2020. doi:10.1103/PhysRevAccelBeams.23.124801

[27] L. Scomparin *et al.*, "Preliminary results on the reinforcement learning-based control of the microbunching instability", in *Proc. IPAC'24*, Nashville, TN, pp. 1808–1811, 2024. doi:10.18429/JACoW-IPAC2024-TUPS61

[28] L. Scomparin *et al.*, "Microsecond-latency feedback at a particle accelerator by online reinforcement learning on hardware", *arXiv:2409.16177*, 2024. doi:10.48550/arXiv.2409.16177

[29] J. Garcia *et al.*, "A comprehensive survey on safe reinforcement learning", *J. Mach. Learn. Res.*, vol. 16, no. 42, pp. 1437–1480, 2015. http://jmlr.org/papers/v16/garcia15a.html

[30] J. Degrave *et al.*, "Magnetic control of tokamak plasmas through deep reinforcement learning", *Nature*, vol. 602, pp. 414–419, 2022. doi:10.1038/s41586-021-04301-9

[31] S. Hirlaender *et al.*, "Ultra fast reinforcement learning demonstrated at CERN AWAKE", in *Proc. IPAC'23*, Venice, Italy, pp. 4510–4513, 2023. doi:10.18429/JACoW-IPAC2023-THPL038

[32] C. Xu *et al.*, "Beam trajectory control with lattice-agnostic reinforcement learning", in *Proc. IPAC'23*, Venice, Italy, pp. 4487–4490, 2023. doi:10.18429/JACoW-IPAC2023-THPL029