ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-MOPA55

FACILITATING MACHINE LEARNING COLLABORATIONS BETWEEN LABS, UNIVERSITIES, AND INDUSTRY

J. P. Edelen, D. T. Abell, D. L. Bruhwiler, S. J. Coleman, N. M. Cook, A. Diaw, J. Einstein-Curtis,
C. C. Hall, M. Kilpatrick, B. Nash, I. V. Pogorelov, RadiaSoft LLC, Boulder, CO, USA
K. A. Brown, Brookhaven National Laboratory, Upton, NY, USA
C. Tennant, Thomas Jefferson National Laboratory, Newport News, VA, USA
P. Piot, Northern Illinois University, DeKalb, IL, USA
S. Calder, C. Hoffmann, Oak Ridge National Laboratory, Oak Ridge, TN, USA
A. L. Edelen, B. O'Shea, R. Roussel, SLAC, Menlo ParkCA, USA
E.-C. Huang, Los Alamos National Laboratory, Los Alamos, NM, USA

Abstract

It is clear from numerous recent community reports, papers, and proposals that machine learning is of tremendous interest for particle accelerator applications. The quickly evolving landscape continues to grow in both the breadth and depth of applications including physics modeling, anomaly detection, controls, diagnostics, and analysis. Consequently, laboratories, universities, and companies across the globe have established dedicated machine learning (ML) and data-science efforts aiming to make use of these new state-of-the-art tools. The current funding environment in the U.S. is structured in a way that supports specific application spaces rather than larger collaboration on community software. Here, we discuss the existing collaboration bottlenecks and how a shift in the funding environment, and how we develop collaborative tools, can help fuel the next wave of ML advancements for particle accelerators.

INTRODUCTION

In recent years machine learning (ML) has been identified as having the potential for significant impact on the modeling, operation, and control of particle accelerators (for example, see Refs. [1, 2]). While there has been an impressive amount of progress for ML in accelerators, most solutions are not yet fully incorporated into regular operation. This in turn limits the degree to which open questions in robustness, algorithm transfer, uncertainty quantification, and generalization to unseen conditions can be addressed.

At present, researchers are incentivized to prioritize proof-of-concept demonstrations, publish, and move on to the next proof-of-concept. This results in many ML algorithms never being fully tested under a variety of conditions and never integrated into operations. Funding is primarily awarded and structured around new ML methods and advances for specific applications of accelerators (e.g. photon science, high-energy physics, medical accelerators), leaving research and development for community code infrastructure, standards, and cross-application algorithm transfer under-funded. There is a tremendous need for (1) open-source, portable, extensible software, (2) along with common benchmarks and worked examples, and (3) investment in personnel to support for MLOps and DevOps. Accelerator applications and control systems share numerous commonalities across different end use-cases. In the following sections we expand on this further.

ML APPLICATIONS TO ACCELERATORS

To orient the reader, we highlight a few use cases for machine learning.

Neural networks (NNs) have been used to create virtual diagnostics [3–5] that supply operators with diagnostic predictions from other measured data. This can be useful when the diagnostic instrument is destructive and cannot be used continuously during downstream operation, or would update too slowly. Similarly, NNs can be used to create comprehensive fast-executing models of accelerator systems [6–10], using combinations of measurement and simulation data. Uncertainty quantification has been investigated [11, 12]. Adaptive feedback methods have also been combined with static ML models to track changes and enable fine-tuning [10, 13, 14].

Anomaly detection has been specifically highlighted as an area where machine learning can significantly impact operational accelerators [15, 16]. ML tools have been applied to detect anomalies in superconducting magnets at CERN [17], RF cavities at DESY [18–20] and RF cavities at JLab [21]. Additionally, machine learning has been used to identify and remove malfunctioning beam position monitors in the Large Hadron Collider (LHC), prior to application of standard optics correction algorithms [22]. Other efforts have sought to use ML for detection of errors in hardware installation [23]. Analysis of the latent space information using autoencoders has also been demonstrated to improve the ability to identify anomalous behavior in LINACs [24].

In terms of optimization and control, machine learning has been employed in a variety of ways. With limited previous data, Bayesian Optimization (BO) adapts a model during tuning. This has been shown in numerous contexts to provide sample-efficient tuning for accelerators [25–28]. Recent advances in BO for accelerators have enabled tuning that respects learned constraints [27, 28], enforces smooth setting changes, and can handle comprehensive multi-objective optimization (e.g. producing the actual Pareto front on an operational accelerator) [28, 29]. Providing an initial solution from a learned global model and fine-tuning with feedback

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-MOPA55

or local optimizers can help compensate for time-varying changes on accelerators [14, 30, 31]. Machine learning has been applied for optics corrections in the LHC [32], for optimization of FELs [33], trajectory control [34], stabilization of source characteristics in synchrotron light sources [35], and for the improvement of low level RF control systems [36]. Moreover, the application of machine learning based surrogate models have demonstrated orders of magnitude speed up in the optimization of accelerator systems using simulation tools, especially for computationally expensive simulations such as machines with intense space charge [37].

The short review above of recent efforts only scratches the surface of how machine learning has been applied to particle accelerators.

COLLABORATION BOTTLENECKS AND POSSIBLE SOLUTIONS

There are some key opportunities that would greatly benefit the accelerator community in the pursuit of developing machine learning technology and improving the efficiency at which we achieve our solutions. Numerous problems across accelerators are similar enough that they can be tackled with shared algorithms and software infrastructure. At present, disparate groups often end up spending resources solving very similar practical challenges. Major limitations are data sharing, access to curated datasets / examples, and common development platforms. This parallels the recent discussions regarding computational efforts that is covered in detail in the Snowmass white paper [38], which highlights the need to maintain an open review, testing process, and validation / benchmarking procedures to ensure computational reproducability. The community also emphasizes the utility of an ecosystem of codes that utilize open source repositories with worked examples and community based surrogate model training. Similar discussion with historical examples for accelerator simulations can be found in Ref. [39]. Machine learning infrastructure is a subset of this community and would benefit from a parallel approach.

Data Sharing

A major driver of general ML research (e.g. in the computer vision boom of the mid 2010s) was the availability of benchmark data sets for algorithm developers to assess new approaches on. Similarly, open availability of trained models enabled researchers to build on previous developments. Data sharing is also a prerequisite to having worked examples of ML applications to accelerator data and systems.

The accelerator community is in a good position to develop and publish open datasets and models, both for simulations and measurements. There are now journals that publish open data set descriptions, and examples already exist for accelerator data sets [40, 41]. This should help incentivise researchers to spend time on the creation of open data sets. In addition, dedicated funding to support community-developed standards to enable interoperability of models, data, and other software (for example optimization algorithm descriptions) is needed. This would enable greater ease of use and transferrability across applications for data sets and ML solutions.

Data sharing has been a topic of discussion since the "early days" of the current ML efforts in accelerators. Data sharing amongst ourselves would no doubt be fruitful especially as smaller groups develop their own internal datasets for benchmarking and solving specific problems. Both experimental datasets and simulation data sets can often be difficult to obtain and interpret, and well curated data is difficult to come by. As a community we can facilitate this through the development of data standards and provide support for researchers who spend time on data curation. Moreover, we can incentivize researchers to publish their work in journals that include open datasets.

Worked Examples

The community needs a collection of benchmark examples for solving specific problems, similar to those in computer vision. Examples for specific problems solved under different collaborations exist and internally different groups may have a good collection of resources in this area. However a centralized repository that can be utilized regardless of funding source would greatly benefit the community. Similar resources exist for course materials in the accelerator community through the US Particle Accelerator School, including for the course on "Optimization and Machine Learning for Particle Accelerators" [42]. These resources are curated and available for future study and as reference materials. A similar approach for machine learning and moreover, accelerator simulations, examples would be highly valuable to the community.

Common Development Platforms

Support is needed to develop standards for data and algorithms that can facilitate interoperability between different software tools and portability between systems. It is well understood that getting a large community to rally behind a single workflow or framework is indeed challenging. However, the amount of resources and time saved once common tools are in place would benefit progress in ML for accelerators tremendously, and it would reduce redundant effort on mundane implementation tasks. Many in the accelerator community note that this investment does come with a risk, as we rely heavily on open-source tools developed outside of accelerators that have their own funding and lifecycle changes. However, focusing on inter-operability of modular software tools with specific uses (for example, one software package for implementations of optimization algorithms, one for handling model execution, one for handling GUI displays based on standard data formats and descriptions) rather than monolithic software packages can help ensure that different pieces can be adjusted independently as the state-of-the-art changes for each.

An example of this from SLAC demonstrates the utility of putting effort into modular software and common standards

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-MOPA55

for data and interfaces. Modular tools for online modeling and ML-based optimization [43, 44] were developed on LCLS and readily transferred to FACET-II (and vice-versa). For online execution of Impact-T simulations and display generation, adapting from LCLS to FACET-II took about an hour of work.

Closely related to this, many accelerators classically have a single overarching control system environment. Moving toward enabling containerization (e.g. in Docker, Singularity) of control system environments can enable a higher level of agility in adjusting different software packages to support new research goals for different research programs at the same accelerator.

It should be emphasized that creation of interoperable, robust, extensible software tools and standards is not trivial. Research and development support is needed to create useful standards, interfaces, and designs. Furthermore, integration with high-performance computing could open up unprecedented applications for real-time, online prediction and control of accelerators. This presents numerous engineering and research challenges that require dedicated funding for development. This is essential for the accelerator community to make full use of ML technology and new computational tools for operation.

Funding Structures and Incentives

It is important to consider not only the technical bottlenecks that exist but institutional bottlenecks. Specifically funding sources and the funding ecosystem can present a significant challenge for large, many institution, collaborations that have diverse funding sources. Laboratory and University efforts tend to center around large funding calls intended to support multiple efforts for several years. Additionally, while laboratory directed research and development funding is available for machine learning infrastructure and ML-ops, these efforts are smaller in scale and focus only on a particular laboratory. Industry efforts tend to focus on the SBIR program which limits the breath of collaboration to one or two institutions for a single project. SBIR funding also limits the distribution of resources due to the nature of the program being a one year concept phase followed by a two year prototype phase. While developments for one science domain application can benefit multiple areas, it can be difficult to get funding for cross-cutting applications like ML. The DOE for example, typically compartmentalizes funding by science application within the various offices underneath the DOE Office of Science. Rather than funding separate efforts in ML for accelerators for each of these areas, the Office of Science could support crosscutting machine learning research. This would enhance our ability to form a larger ecosystem for supporting machine learning research across the accelerator industry.

The incentive structure for researchers also places more emphasis on proof-of-principle results and concepts than it does on bringing methods to the level of maturity needed for deployment. This is a huge problem at present for our field, as there is a large gap between the state of the art indicated by the literature and the types of tools that are available at operational accelerator facilities. As a community we could improve this by better rewarding researchers who take the time to rigorously test ML algorithms and adjusting our standards.

SOFTWARE TOOLS

Facilitation of machine learning collaborations is in a sense dominated by the availability and usefulness of software tools. Groups tend to adopt a software framework for use during a particular collaborative effort. For example sharing code through version control software such as github, or choosing to adopt a ML deployment framework such as MLFlow. There is significant opportunity for the community to develop and adopt common tools and workflows that would improve the transfer of technology between labs, universities, and industry. We highlight some examples of software tools below.

Containers and JupyterLab

JupyterLab and container technology are widely used for scientific computing. One of the key advantages of container technology is the dependencies can be fully specified independent of the operating system. This allows users across platforms to run the same simulations, analysis, model learning, etc, regardless of their particular development environment. Containers deployed via a web browser using JupyterLab are highly versatile and can speed up the rate at which new users can effectively contribute to machine learning research for accelerators.

MLOps and DevOps

Training and deploying models involves several forms of version control for consistent results. This includes tracking datasets and data transformations, tying those to specific model architectures and experiments, and tracking model performance and parameters for each run. Several open source tools have been developed for this in the machine learning space, including data revisioning with DVC[1], model runs and serving with MLflow[2], and workflows with Apache Airflow[3] or kubeflow[4], and commercial solutions including: AWS SageMaker, AzureML, and NVIDIA Triton. Managing this process is known as machine learning operations (MLops), which involves tracking the workflows, inputs, and outputs of a process throughout experimentation and development. Even in the most turn-key of deployments, custom scripting and tooling is necessary to support an MLops framework due to the variety of possible endpoints and training frameworks. This includes support for both PyTorch and TensorFlow training scripts and models, while also being able to run inference tasks on CPUs, GPUs, and specialized and programmable endpoints such as TPUs and FPGAs.

[1] https://dvc.org/
[2] https://mlflow.org/
[3] https://airflow.apache.org/
[4] https://www.kubeflow.org/

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-MOPA55

A deployment on operational accelerators has specialized needs, as there are real time systems, high level control loops, and a desire to limit the need for expert intervention. This offers the opportunity to specialize the workflows for specific use cases, while also requiring a more robust development and versioning system.

Cloud-Based Tools

Tools that take advantage of cloud computing and browser based technology are an important technological development for fostering collaborations. Overleaf for example allows users across computer platforms to develop documents in a friendly environment. For scientific computing there have been recent developments in browser-based technology that enable the sharing of information in a similar way. For example, the scientific gateway called `Sirepo` [45], uses an open source cloud computing framework of the same name [46–48]. While `Sirepo` traditionally supports the simulation of particle accelerators and adjacent technologies using for example: MAD-X, `elegant`, OPAL, or Synergia, and Synchrotron Radiation Workshop (SRW) for physical optics. Users can build simulations and share the configurations via a web-link. Recently the Sirepo framework was expanded to include a machine learning and data analysis application referred to as `Activait`. Users can import a dataset and perform a wide range of analysis tasks including machine learning model development. Users can then share these results in the same manor as with the physics simulations.

Modular and Interoperable Tools and Standards

Separating software development by the specific task, when coupled with standards for interfaces and data, can help aid inter-operability and extensibility. Numerous examples exist, but here we highlight a few that are used at SLAC together for online modeling, online optimization, offline design optimization, algorithm prototyping before beam time with surrogate models, and data set creation for ML-based system models.

openPMD [49] is a standard way of defining a particle distribution, making it possible to exchange these more readily between different simulation codes and analysis tools. Distgen [50] enables arbitrary creation of distributions and uses openPMD. The LUME modeling tools [44] enable interfacing and execution of particle accelerator simulation codes in a standard way, making it easier to do start-to-end modeling with different combinations of accelerator codes. Data is output in a standard format, making it easy to use similar data processing codes for different systems and types of simulations. Xopt [43] is a framework for defining optimization algorithms and logging results, and has been used extensively at multiple accelerators for doing online optimization with the real machine and offline optimization with simulations and surrogate models.

CONCLUSIONS

There are many varied efforts to deploy ML tools across the accelerator community both domestically and abroad. The community as a whole would benefit from adopting a shared set of interoperable modular tools and standards. Moreover, making available example simulation and experimental datasets would lower the barrier to entry for newcomers and strengthen our ability to perform more cutting edge ML research. Here we have provided some initial thoughts on how to shape and encourage such collaborations.

REFERENCES

[1] A. L. Edelen, S. G. Biedron, B. E. Chase, D. Edstrom, S. V. Milton, and P. Stabile, "Neural networks for modeling and control of particle accelerators," *IEEE Trans. Nucl. Sci.*, vol. 63, no. 2, pp. 878–897, 2016. doi:10.1109/TNS.2016.2543203

[2] A. Edelen *et al.*, "Opportunities in machine learning for particle accelerators," Fermilab, Batavia, IL, USA, Tech. Rep. FERMILAB-PUB-19-017-AD, 2017. doi:10.48550/ARXIV.1811.03172

[3] C. Emma, A. Edelen, M. J. Hogan, B. O'Shea, G. White, and V. Yakimenko, "Machine learning-based longitudinal phase space prediction of particle accelerators," *Phys. Rev. Accel. Beams*, vol. 21, no. 11, p. 112 802, 2018. doi:10.1103/PhysRevAccelBeams.21.112802

[4] A. Hanuka *et al.*, "Accurate and confident prediction of electron beam longitudinal properties using spectral virtual diagnostics," *Sci. Rep.*, no. 1, p. 2945, 2022. doi:10.1038/s41598-021-82473-0

[5] A. Sanchez-Gonzalez, V. Bapst, K. Cranmer, and P. Battaglia, "Hamiltonian graph networks with ode integrators," 2019. doi:10.48550/ARXIV.1909.12790

[6] A. L. Edelen, S. Biedron, S. V. Milton, and J. P. Edelen, "First Steps Toward Incorporating Image Based Diagnostics into Particle Accelerator Control Systems Using Convolutional Neural Networks," in *Proc. NAPAC'16*, Chicago, IL, USA, Oct. 2016, pp. 390–393. doi:10.18429/JACoW-NAPAC2016-TUPOA51

[7] L. Gupta, A. Edelen, N. Neveu, A. Mishra, C. Mayes, and Y.-K. Kim, "Improving surrogate model accuracy for the LCLS-II injector frontend using convolutional neural networks and transfer learning," *Mach. Learn: Sci. and Tech.*, vol. 2, no. 4, p. 045 025, 2021. doi:10.1088/2632-2153/ac27ff

[8] A. Edelen, N. Neveu, M. Frey, Y. Huber, C. Mayes, and A. Adelmann, "Machine learning for orders of magnitude speedup in multiobjective optimization of particle accelerator systems," *Phys. Rev. Accel. Beams*, vol. 23, no. 4, p. 044 601, 2020. doi:10.1103/PhysRevAccelBeams.23.044601

[9] J. Zhu, Y. Chen, F. Brinker, W. Decking, S. Tomin, and H. Schlarb, "High-fidelity prediction of megapixel longitudinal phase-space images of electron beams using encoder-decoder neural networks," *Phys. Rev. Appl.*, vol. 16, no. 2, p. 024 005, 2021. doi:10.1103/PhysRevApplied.16.024005

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-MOPA55

[10] A. Scheinker, F. Cropp, S. Paiagua, and D. Filippetto, "An adaptive approach to machine learning for compact particle accelerators," *Sci. Rep.*, vol. 11, no. 1, pp. 1–11, 2021. doi:10.1038/s41598-021-98785-0

[11] O. Convery, L. Smith, Y. Gal, and A. Hanuka, "Uncertainty quantification for virtual diagnostic of particle accelerators," *Phys. Rev. Accel. Beams*, vol. 24, no. 7, p. 074 602, 2021. doi:10.1103/PhysRevAccelBeams.24.074602

[12] A. A. Mishra, A. Edelen, A. Hanuka, and C. Mayes, "Uncertainty quantification for deep learning in particle accelerator applications," *Phys. Rev. Accel. Beams*, vol. 24, no. 11, p. 114 601, 2021. doi:10.1103/PhysRevAccelBeams.24.114601

[13] A. Scheinker, "Adaptive machine learning for robust diagnostics and control of time-varying particle accelerator components and beams," *Information*, vol. 12, no. 4, p. 161, 2021. doi:10.3390/info12040161

[14] A. L. Edelen, J. P. Edelen, S. Biedron, S. Milton, and P. van der Slot, "Using neural network control policies for rapid switching between beam parameters in a free-electron laser," in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA*, vol. 8, 2017.

[15] D. Marcato *et al.*, "Machine learning-based anomaly detection for particle accelerators," in *2021 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2021, pp. 240–246. doi:10.1109/CCTA48906.2021.9658806

[16] J.P. Edelen and N.M. Cook, "Anomaly detection in particle accelerators using autoencoders," *arXiv preprint arXiv:2112.07793*, 2021. doi:10.26024/P6MV-EN77

[17] M. Wielgosz, A. Skoczeń, and M. Mertik, "Using lstm recurrent neural networks for monitoring the lhc superconducting magnets," *Nucl. Instrum. and Methods in Phys. Res. Sect. A*, vol. 867, pp. 40–50, 2017. doi:10.1016/j.nima.2017.06.020

[18] A. S. Nawaz, S. Pfeiffer, G. Lichtenberg, and H. Schlarb, "Self-organzied critical control for the european xfel using black box parameter identification for the quench detection system," in *2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, IEEE, 2016, pp. 196–201. doi:10.1109/SYSTOL.2016.7739750

[19] A. Nawaz, S. Pfeiffer, G. Lichtenberg, and P. Rostalski, "Anomaly detection for the european xfel using a nonlinear parity space method," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 1379–1386, 2018. doi:10.1016/j.ifacol.2018.09.554

[20] A. Nawaz, S. Pfeiffer, G. G. Lichtenberg, and P. Rostalski, "Anomaly detection for cavity signals-results from the European XFEL," in *Proceedings of the 9th International Particle Accelerator Conference, Vancouver, BC, Canada*, vol. 29, 2018. doi:10.18429/JACoW-IPAC2018-WEPMF058

[21] C. Tennant, A. Carpenter, T. Powers, A. S. Solopova, L. Vidyaratne, and K. Iftekharuddin, "Superconducting radio-frequency cavity fault classification using machine learning at jefferson laboratory," *Phys. Rev. Accel. Beams*, vol. 23, no. 11, p. 114 601, 2020. doi:10.1103/PhysRevAccelBeams.23.114601

[22] E. Fol, J. M. C. de Portugal, and R. Tomás, "Unsupervised Machine Learning for Detection of Faulty Beam Position Monitors," in *Proc. IPAC'19*, Melbourne, Australia, May 2019, pp. 2668–2671. doi:10.18429/JACoW-IPAC2019-WEPGW081

[23] T. Dewitte, W. Meert, E. V. Wolputte, and P. V. Trappen, "Anomaly Detection for CERN Beam Transfer Installations Using Machine Learning," in *Proc. ICALEPCS'19*, New York, NY, USA, Oct. 2019, p. 1067. doi:10.18429/JACoW-ICALEPCS2019-WEMPR010

[24] J. P. Edelen and C. C. Hall, "Autoencoder based analysis of rf parameters in the fermilab low energy linac," *Information*, vol. 12, no. 6, p. 238, 2021. doi:10.3390/info12060238

[25] J. Duris *et al.*, "Bayesian optimization of a free-electron laser," *Phys. Rev. Lett.*, vol. 124, no. 12, p. 124 801, 2020. doi:10.1103/PhysRevLett.124.124801

[26] A. Hanuka *et al.*, "Physics model-informed gaussian process for online optimization of particle accelerators," *Phys. Rev. Accel. Beams*, vol. 24, no. 7, p. 072 802, 2021. doi:10.1103/PhysRevAccelBeams.24.072802

[27] J. Kirschner, M. Mutny, N. Hiller, R. Ischebeck, and A. Krause, "Adaptive and safe bayesian optimization in high dimensions via one-dimensional subspaces," pp. 3429–3438, 2019. doi:10.48550/ARXIV.1902.03229

[28] R. Roussel *et al.*, "Turn-key constrained parameter space exploration for particle accelerators using bayesian active learning," *Nat. Comm.*, vol. 12, no. 1, pp. 1–7, 2021. doi:10.1038/s41467-021-25757-3

[29] R. Roussel, A. Hanuka, and A. Edelen, "Multiobjective bayesian optimization for online accelerator tuning," *Phys. Rev. Accel. Beams*, vol. 24, no. 6, p. 062 801, 2021. doi:10.1103/PhysRevAccelBeams.24.062801

[30] A. Scheinker, A. Edelen, D. Bohler, C. Emma, and A. Lutman, "Demonstration of model-independent control of the longitudinal phase space of electron beams in the linac-coherent light source with femtosecond resolution," *Phys. Rev. Lett.*, vol. 121, no. 4, p. 044 801, 2018. doi:10.1103/PhysRevLett.121.044801

[31] A. Edelen, "Neural networks for modeling and control of particle accelerators," *Dissertation*, Colorado State University. Available at: https://tinyurl.com/37jzwua8.

[32] E. Fol, J. M. C. de Portugal, G. Franchetti, and R. Tomás, "Optics Corrections Using Machine Learning in the LHC," in *Proc. IPAC'19*, Melbourne, Australia, May 2019, pp. 3990–3993. doi:10.18429/JACoW-IPAC2019-THPRB077

[33] F. O'Shea, N. Bruchon, and G. Gaio, "Policy gradient methods for free-electron laser and terahertz source optimization and stabilization at the fermi free-electron laser at elettra," *Phys. Rev. Accel. Beams*, vol. 23, no. 12, p. 122 802, 2020. doi:10.1103/PhysRevAccelBeams.23.122802

[34] V. Kain *et al.*, *Phys. Rev. Accel. Beams*, vol. 23, no. 12, p. 124 801, 2020. doi:10.1103/PhysRevAccelBeams.23.124801

[35] S. Leemann *et al.*, "Demonstration of machine learning-based model-independent stabilization of source properties in synchrotron light sources," *Phys. Rev. Lett.*, vol. 123, no. 19, p. 194 801, 2019.

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-MOPA55

doi:10.1103/PhysRevLett.123.194801

[36] J. A. D. Cruz, S. Biedron, M. Martinez-Ramon, S. Sosa, and R. Pirayesh, *Studies in applying machine learning to LLRF and resonance control in superconducting RF cavities*, 2019. doi:10.48550/arXiv.1910.07648

[37] L. Yang, D. Robin, F. Sannibale, C. Steier, and W. Wan, "Global optimization of an accelerator lattice using multiobjective genetic algorithms," *Nucl. Instrum. Meth. Phys. Res., Sect. A*, vol. 609, no. 1, pp. 50–57, 2009. doi:10.1016/j.nima.2009.08.027

[38] S. Biedron *et al.*, *Snowmass21 Accelerator Modeling Community White Paper*, 2022. doi:10.48550/ARXIV.2203.88335

[39] D. Sagan *et al.*, "Simulations of future particle accelerators: Issues and mitigations," *J. Instr.*, vol. 16, no. 10, T10002, 2021. doi:10.1088/1748-0221/16/10/t10002

[40] D. Kafkes and J. St. John, "Boostr: A dataset for accelerator control systems," *Data*, vol. 6, no. 4, p. 42, 2021. doi:10.3390/data6040042

[41] A. Carpenter, C. Tennant, and L. Vidyaratne, "Jefferson laboratory c100 superconducting radio-frequency cavity fault data, 2020," Thomas Jefferson National Accelerator Facility (TJNAF), Newport News, VA, USA, Tech. Rep., 2020. doi:10.14462/MLFaultClassifier/1616675

[42] R. Lehe, *Optimization and machine learning for particle accelerators*, US Particle Accelerator School, 2022. https://slaclab.github.io/USPAS_ML/

[43] E. A. Nanni *et al.*, "C 3 demonstration research and development plan," *arXiv preprint*, 2022. doi:10.5281/zenodo.5559141

[44] C. E. Mayes *et al.*, "Lightsource Unified Modeling Environment (LUME), a Start-to-End Simulation Ecosystem," in *Proc. IPAC'21*, Campinas, Brazil, May 2021, pp. 4212–4215. doi:10.18429/JACoW-IPAC2021-THPAB217

[45] "The Sirepo scientific gateway." (2021), https://sirepo.com

[46] D. L. Bruhwiler *et al.*, "Knowledge Exchange Within the Particle Accelerator Community via Cloud Computing," in *Proc. IPAC'19*, Melbourne, Australia, May 2019, pp. 3548–3551. doi:10.18429/JACoW-IPAC2019-THPMP046

[47] M. S. Rakitin *et al.*, "Sirepo: An open-source cloud-based software interface for x-ray source and optics simulations," *J. Synchrotron Radiat.*, vol. 25, no. 6, pp. 1877–1892, 2018. doi:10.1107/S1600577518010986

[48] "The Sirepo framework for scientific cloud computing." (2021), https://github.com/radiasoft/sirepo

[49] A. Huebl *et al.*, "Openpmd 1.0.0: A meta data standard for particle and mesh based data," 2015. doi:10.5281/zenodo.33624

[50] *Githuib: Distgen particle distribution generator*. https://github.com/ColwynGulliford/distgen