Continual Learning for Particle Accelerators

Kishansingh Rajput[1, 3], Malachi Schram[1], Willem Blokland[2], Alexander Zhukov[2], Sen Lin[3]
[1]Thomas Jefferson National Accelerator Facility, Newport News, VA, USA 23606
[2]Oak Ridge National Laboratory, Oak Ridge, TN, USA
[3]Department of Computer Science, University of Houston, Houston, TX, USA 77024

Abstract

Particle accelerators operate under dynamically changing conditions, which often lead to data distribution drifts. These drifts pose significant challenges for Machine Learning (ML) models, which typically fail to maintain performance when faced with such non-stationary data. In particle accelerators, the primary sources of data drifts include changes in accelerator settings and non-measured parameters such as machine degradation. Previous research has proposed conditional models to handle multiple beam configurations effectively; however, it is challenging to train ML models on all possible configuration settings. Additionally, conditional models alone can not address performance degradation caused by drifts due to non-measured factors. These limitations contribute to a significant gap between ML development and its deployment in real-world operational settings. To bridge this gap, in this paper, we identify some of the key areas within particle accelerators where continual learning can help mitigate drift-induced performance degradation. In addition, we present a real use case where memory-based continual learning has been employed to demonstrate stable performance on conditional Auto-Encoder model when switching between different beam settings.

1 Introduction

Particle accelerators are critical in both fundamental research and applied sciences, enabling high-precision experiments in physics, medicine, and industry [1]. The operation of these complex systems requires precise tuning of thousands of interconnected components. Traditional diagnostics and model-based control are challenged by the high dimensionality of settings, sensitivity of beam dynamics, and the lack of comprehensive physics models for certain phenomena.

Machine learning (ML) has emerged as a promising approach for accelerator applications such as anomaly detection [2, 3], optimization [4, 5], real-time control [6, 7], surrogate modeling[8, 9, 10], digital twins [11], and virtual diagnostics [12]. ML can improve performance, automate tuning, and provide predictive capabilities beyond traditional methods. However, long-term deployment remains rare. A primary barrier is data and concept drift, shifts in input distributions, and input–output relationships due to measurable changes (for example, beam settings) and unmeasured factors (for example, component degradation). Such drift degrades ML models trained under stationary assumptions, often necessitating retraining, which can be slow and risk catastrophic forgetting of prior knowledge [13]. To partially address this challenge, conditional modeling techniques that incorporate configuration settings as input have emerged, improving prediction accuracy under diverse conditions [14, 15]. However, it is very challenging to anticipate all possible future configurations before running the experiments. In addition, conditional model alone can not address drifts arising from non-measured factors, necessitating the development of adaptive learning strategies. Continual learning (CL) [16] offers a pathway to sustained model performance in dynamic environments such as particle accelerators by enabling adaptation to new data while retaining past knowledge. CL explicitly mitigates forgetting through methods such as memory replay, architectural expansion, regularization,

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

gradient projection, ensembles, and meta-learning [13]. Its potential has been demonstrated in fields from computer vision to robotics [17, 18], and there are developments to run CL on edge hardware [19] but its application to accelerators is largely unexplored.

To fill this void, this paper seeks to present a concise review of the challenges and opportunities in continual learning as it applies to particle accelerators. Our analysis aims to identify key constraints and unique requirements of deploying continual learning in particle accelerator applications to pave a way for future advancements in CL for particle accelerators. In addition, we demonstrate a use case where continual learning is successfully applied to maintain stable performance of a Conditional Auto-Encoder (CAE) model at Spallation Neutron Source (SNS) accelerator when switching between different beam settings.

2 Continual Learning is Critical to ML for Particle Accelerators

ML methods are increasingly getting popular for addressing various complex tasks within particle accelerators [20]. Although these applications have shown strong potential in short-term studies, their long-term deployment is hindered by *data distribution drift* and requires advance CL integration. There are different CL techniques, and their suitability depends on application-specific constraints, including latency requirements, data availability, task boundary clarity, and tolerance for forgetting. Complex machines such as particle accelerators have various constraints depending on underlying application. To pave a way to CL development in this domain, we provide a brief summary of four major applications within accelerators along with their challenges and constraints and match them with proposed CL techniques as follows.

Anomaly Detection models, such as CAE and conditional variational-AE (CVAE), contrastive learning methods, and recurrent networks, are used to identify equipment faults, errant beams, and other abnormal conditions [15, 3, 21]. In operation, these models often need to run in near real time and require rapid adaptation to avoid ML downtime when online model can not provide reliable predictions. Constraints include low latency, limited storage for past data, and, in the case of supervised methods, sparse labeled anomalies. For scenarios with mixed known and unknown drift sources, hybrid approaches combining *meta-learning* (to group operating regimes) with *regularization-based incremental updates* [22, 23] can achieve fast adaptation while tolerating limited forgetting. At a high level regularization based methods achieve knowledge retention by penalizing updates to weights that are relevant to previously learned tasks. In addition, sparse sampling based replays can also be employed to limit forgetting similar to the use case presented in this paper.

Optimization and Control applications include Bayesian optimization (BO), Deep Reinforcement Learning (DRL), and Model Predictive Control (MPC) as applied to tuning beam parameters, optimizing injection, and maintaining stability under varying conditions [4, 5, 11, 24]. For online control, models must adapt with minimal downtime and often need to preserve knowledge across multiple operating configurations. BO may benefit from periodically updated surrogate models via *memory-based replay* [25], while DRL and MPC controllers can integrate *regularization* or *gradient projection* [26] methods to retain policy effectiveness across drifts. Meta-reinforcement learning is well-suited for accelerators that cycle through recurring operational modes as shown in a recent study [27].

Surrogate Models and Digital Twins emulate accelerator behavior for training control algorithms, pre-emptive maintenance, and scenario testing [11]. Surrogate models often run offline and can store historical data, where *memory-based CL* with *priority replay* [28] is recommended to minimize forgetting and maintain fidelity over long timescales. On the other hand, Digital Twins need to continuously adapt to evolving physical conditions, mostly in online mode. For systems experiencing continual expansion of operating space, *architecture-based methods* (e.g., progressive neural networks [29]) can be combined with replay to add capacity without sacrificing prior accuracy.

Virtual Diagnostics predict unmeasured beam properties in real time from other sensor data, or map the measurements in a more convenient format for operators enabling non-destructive monitoring [30, 31, 32]. Such applications involving real-time decision-making require low-latency adaptation similar to anomaly detection. For recurring settings, *ensemble-based methods* or *architecture-based growth* can maintain specialized models per configuration, fine-tuned from a shared base model. A *time-prioritized memory buffer* can be used to retain only recent and relevant data distributions for adaptation, reducing compute cost.

Across all these applications, CL adoption requires robust infrastructure for monitoring, drift detection, and automated model updating, integrated with accelerator control systems. By tailoring CL strategies to operational constraints, ML systems can transit from short-lived prototypes to persistent, adaptive tools supporting the long-term reliability and autonomy of particle accelerators. More details can be found in a recent review at [13].

3 Memory-based Continual Machine Learning at SNS Accelerator

The Spallation Neutron Source (SNS) accelerator facility at Oak Ridge National Laboratory (ORNL) delivers a 60 Hz pulsed 1.3 GeV proton beam at 1.8 MW, making it the world's highest power proton accelerator. The beam is accelerated in a superconducting linear accelerator and accumulated in a ring to form a very short and intense pulse with up to 1.6×10^{14} protons. Ongoing efforts at SNS are leveraging ML to predict anomalies before they occur to pre-emptively abort the beam and potentially avoid damages due to beam loss and potentially reduce longer downtime, more details on this effort can be found in [3, 15]. Data drift is a significant challenge in maintaining ML performance, as it can lead to unstable model predictions causing a significant increase in false positives. A previous study [15] has proposed to use conditional ML models–Conditional Siamese Neural Network (CSNN), and CVAE to mitigate performance degradation due to drift in the data caused by beam setting changes and improve the overall performance. However, there is a strong need for adaptive methods, i.e., continual model update to account for beam settings that are not included in the training data, and drifts caused by other sources. As such, we have leveraged memory-based CL to achieve stable performance on a CAE model under data drifts. The information on the data set and sensors can be found at [15].

It is important to note that the conditional input vector includes 21 beam setting parameters, which also serve to define the task boundaries. In this study, any change in one or more of these parameters is treated as a distinct task. Consequently, we have 31 unique beam setting vectors, corresponding to 31 separate tasks derived from real experimental data collected at the SNS accelerator in October 2024. While some tasks exhibit similar characteristics and could potentially be grouped together, others display distinctly different behaviors. Nonetheless, we chose to define tasks strictly based on beam settings, as these changes are explicitly known. In our experiment, we replicated online deployment scenario at the SNS accelerator, with tasks presented in the sequence that they appeared in operation. Particularly, we focus on addressing drifts caused by known changes in the beam settings.

3.1 Methods

The CAE model used in this study comprises a conditional encoder and a conditional decoder. The encoder consists of four convolutional blocks, each containing a 1D convolutional layer followed by a max-pooling layer. Each convolutional layer uses 128 filters with kernel size of 6, except for the final layer, which has 16 filters. All max-pooling layers use a pooling size of 2. The output from the last convolutional layer is flattened and concatenated with the conditional input before feeding to three dense layers each with 128 nodes followed by a final dense layer with 256 nodes. As such, latent vector with size 256 is produced. The latent vector is concatenated with conditional input before feeding to decoder's three dense layers each using 128 nodes followed by a dense layer with 625 nodes (for proper output shape). These layers are followed by four pairs of 1D convolutional layers and up-sampling layers, each convolutional layer uses 128 filters with kernel size of 6, and up-sampling layers use size 2. All the layers uses ReLU activation function, except final decoder layer, that uses linear. In total, the model has 2.3M learnable parameters.

We train the CAE in a semi-supervised manner using only normal data, enabling it to reconstruct normal patterns accurately. Anomalies, deviating from this distribution, yield higher reconstruction errors, making the CAE suitable for anomaly detection. We used Mean Absolute Error (MAE) as loss function and Adam Optimizer with initial learning rate of 10^{-4}. We also used learning rate reduction on validation loss plateau and early stopping with patience of 5, and 20 epochs respectively. In traditional replay techniques, all samples are retained in memory for future training sessions on new tasks, which can lead to increased storage requirements and longer training times for large numbers of tasks. In real-time decision making processes such as particle accelerator anomaly prediction, fast model training is desired when data drifts to minimize ML downtime. To address this, we use the reconstruction error distribution from the CAE model to pick representative samples and maintain a small subset per task for replay. The sample section technique is described in Algorithm 1. This approach achieves a balanced subset of samples spanning different error distributions while avoiding over-representation of either low-error or high-error examples in the memory buffer per task.

Algorithm 1 Replay Sample Selection with Quantile-Based Sampling on Reconstruction Error

Input: Reconstruction errors $\{d_i\}_{i=1}^{N}$, number of partitions n, buffer size M
Output: Replay buffer $\mathcal{B}$ of size M
Initialize buffer $\mathcal{B} \leftarrow \emptyset$, $Q_0 \leftarrow \min(d_i)$ **for** $j \leftarrow 1$ **to** n **do**
  $Q_j \leftarrow Quantile(\{d_i\}, q_j = j/n)$; $I_j \leftarrow \{d_i \mid Q_{j-1} < d_i \leq Q_j\}$ **if** $I_j \neq \emptyset$ **then**
    Sample $s_j \sim Uniform(I_j)$; $\mathcal{B} \leftarrow \mathcal{B} \cup \{s_j\}$
  end
end
if $|\mathcal{B}| > M$ **then**
  Resample buffer using above steps with the following inputs: reconstruction errors (d) = $\mathcal{B}$, number of partitions $n = M$, and buffer size $M = M$
end
return $\mathcal{B}$

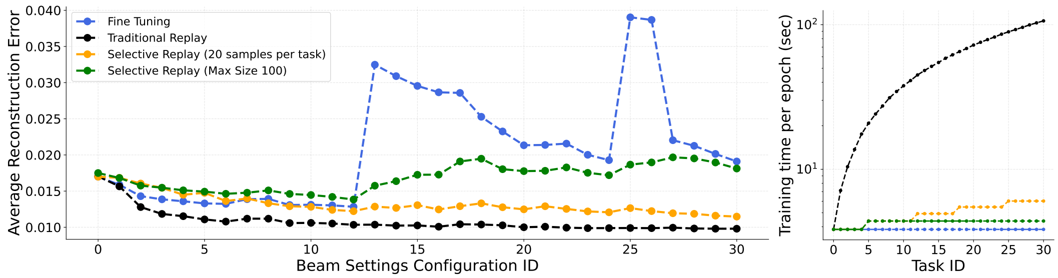
We employ two variations of sample selection based on Algorithm 1. The data samples from normal operation that belong to a single beam setting instance are similar in structure. As such, we implement a growing replay with 20 samples per task with no limit on maximum replay size (i.e., M is virtually infinite). This sample selection strategy grows the replay size linearly with the number of tasks and may attribute to increasing training time as number of tasks increases, however, still expected to be significantly faster than traditional replay. Next, we employ a fixed sized replay with maximum size of 100 while still picking 20 samples per task following Algorithm 1. It is important to note that the selection of size 20, and 100 are mainly for demonstration purposes and can be optimized to balance between training time and model performance based on anticipated number of tasks. All methods weight training losses for each task inversely to task representation in the dataset.

3.2 Results

To offer readers a thorough comparison, we included four different approaches for maintaining the stability of the CAE model during task transitions on the SNS dataset: two variants of selective replay techniques as previously described (with 20 samples per task, one with an unrestricted replay size, and another with a maximum replay size of 100); the traditional replay, which preserves all data samples from preceding tasks; and online fine-tuning, which exclusively utilizes new data for model updates. Currently at SNS, online fine-tuning is being utilized to update the deployed model periodically and it's performance is being assessed in deployment mode. We compare average reconstruction error on test datasets up to given task from all the methods. The test data sets consist of equal number of samples (5000 in this case) to avoid bias.

Our experimental findings, as depicted in Figure 1a, indicate that selective replay techniques can sustain stability without compromising performance while significantly expediting the training process, as illustrated in Figure 1b. Our approach employing selective replay performs similar to the standard replay but utilizes only 20 samples per task, which results in shorter training time, reduced memory usage, and lower computational demands. In contrast, online fine-tuning and fixed-size replay methods achieve approximately constant training time per epoch but are unable to maintain stable performance in the presence of considerable task drift due to catastrophic forgetting.

Ultimately, these results demonstrate that selective replay techniques tailored to the specific application and model can significantly enhance the preservation of long-term stability in machine learning models during particle accelerator operations. It is important to note that our research presents an initial step towards developing tailored continual learning solutions for unique challenges encountered by various accelerator applications, such as selecting the right adaptation method or evaluating metrics.



(a) Average reconstruction error on all previous tasks in the chain

(b) Training time per epoch

Figure 1: Performance and training time comparison among four methods under consideration. The selective replay with 20 samples per task demonstrate similar performance to traditional replay while being much faster in training. Selective replay with max size of 100, and Fine-tuning with no replay shows forgetting due to loss of information.

4 Conclusion

In this paper, we have attempted to bridge the gap between ML solution development and their long term deployment in particle accelerator applications. We have provided a brief summary of opportunities in various accelerator applications where continual learning can help. In addition, we have introduced memory-based replay techniques with representative sample selection which helps mitigate model degradation during task switching and expected to have significantly lower storage requirements and training times compared to traditional replay. We demonstrated the application of continual learning using a real-world example at SNS accelerator where our approach helped maintain stable performance on drifting data. One of the limitation of this work is that it assumes when task changes, however, it can be coupled with a drift detection method to make it more generalized. The future work will include exploring tailored continual learning solutions that address unique challenges and constraints specific to accelerator applications. We plan on extending our existing approach with other methods like meta-learning and adapters. Moreover, we aim to evaluate these techniques using real-world metrics related to anomaly predictions and decision-making processes in particle accelerators.

5 Acknowledgement

This work was supported by the DOE Office of Science, United States under Grant No. DE-SC0009915 (Office of Basic Energy Sciences, Scientific User Facilities program). This manuscript has been authored by Jefferson Science Associates (JSA) operating the Thomas Jefferson National Accelerator Facility for the U.S.Department of Energy under Contract No. DE-AC05-06OR23177. Oak Ridge National Laboratory is operated by UT-Battelle, LLC, under contract DE-AC05-00OR22725. The US government retains, and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan)

References

[1] Ugo Amaldi. The importance of particle accelerators. *Europhys. News*, 31(6):5–9, November 2000.

[2] Yasir Alanazi, Malachi Schram, Kishansingh Rajput, Steven Goldenberg, Lasitha Vidyaratne, Chris Pappas, Majdi I. Radaideh, Dan Lu, Pradeep Ramuhalli, and Sarah Cousineau. Multi-module-based CVAE to predict HVCM faults in the SNS accelerator. *Machine Learning with Applications*, 13:100484, 2023.

[3] Willem Blokland, Kishansingh Rajput, Malachi Schram, Torri Jeske, Pradeep Ramuhalli, Charles Peters, Yigit Yucesan, and Alexander Zhukov. Uncertainty aware anomaly detection

to predict errant beam pulses in the oak ridge spallation neutron source accelerator. *Phys. Rev. Accel. Beams*, 25:122802, Dec 2022.

[4] J. Duris, D. Kennedy, A. Hanuka, J. Shtalenkova, A. Edelen, P. Baxevanis, A. Egger, T. Cope, M. McIntire, S. Ermon, and D. Ratner. Bayesian optimization of a free-electron laser. *Phys. Rev. Lett.*, 124:124801, Mar 2020.

[5] Kishansingh Rajput, Malachi Schram, Auralee Edelen, Jonathan Colen, Armen Kasparian, Ryan Roussel, Adam Carpenter, He Zhang, and Jay Benesch. Harnessing the power of gradient-based simulations for multi-objective optimization in particle accelerators. *Machine Learning: Science and Technology*, 6(2):025018, apr 2025.

[6] Xiaoying Pang, Sunil Thulasidasan, and Larry Rybarcyk. Autonomous control of a particle accelerator using deep reinforcement learning, 2020.

[7] Michael Schenk, Elías F Combarro, Michele Grossi, Verena Kain, Kevin Shing Bruce Li, Mircea-Marian Popa, and Sofia Vallecorsa. Hybrid actor-critic algorithm for quantum reinforcement learning at CERN beam lines. *Quantum Science and Technology*, 9(2):025012, February 2024.

[8] Malachi Schram, Kishansingh Rajput, Karthik Somayaji NS, Peng Li, Jason St. John, and Himanshu Sharma. Uncertainty aware machine-learning-based surrogate models for particle accelerators: Study at the Fermilab Booster Accelerator Complex. *Phys. Rev. Accel. Beams*, 26:044602, Apr 2023.

[9] Kunxiang Sun, Xiaolong Chen, Xiaoying Zhao, Xin Qi, Zhijun Wang, and Yuan He. Surrogate model of particle accelerators using encoder–decoder neural networks with physical regularization. *International Journal of Modern Physics A*, 38(26n27):2350145, 2023.

[10] Jan Kaiser, Chenran Xu, Annika Eichler, and Andrea Santamaria Garcia. Bridging the gap between machine learning and particle accelerator physics with high-speed, differentiable simulations. *Phys. Rev. Accel. Beams*, 27:054601, May 2024.

[11] Diana Kafkes and Malachi Schram. Developing Robust Digital Twins and Reinforcement Learning for Accelerator Control Systems at the Fermilab Booster. In *12th International Particle Accelerator Conference* , 5 2021.

[12] 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*, 21:112802, Nov 2018.

[13] Kishansingh Rajput, Sen Lin, Auralee Edelen, Willem Blokland, and Malachi Schram. Outlook towards deployable continual learning for particle accelerators. *Machine Learning: Science and Technology*, 6(3):031001, jul 2025.

[14] Alexander Scheinker. Conditional guided generative diffusion for particle accelerator beam diagnostics. *Scientific Reports*, 14(1):19210, Aug 2024.

[15] Kishansingh Rajput, Malachi Schram, Willem Blokland, Yasir Alanazi, Pradeep Ramuhalli, Alexander Zhukov, Charles Peters, and Ricardo Vilalta. Robust errant beam prognostics with conditional modeling for particle accelerators. *Machine Learning: Science and Technology*, 5(1):015044, mar 2024.

[16] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[17] Zhen Wang, Liu Liu, Yiqun Duan, Yajing Kong, and Dacheng Tao. Continual learning with lifelong vision transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 171–181, 2022.

[18] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion*, 58:52–68, 2020.

[19] Eugenio Ressa, Alberto Marchisio, Maurizio Martina, Guido Masera, and Muhammad Shafique. Tinycl: An efficient hardware architecture for continual learning on autonomous systems, 2025.

[20] Auralee Edelen and Xiaobiao Huang. Machine learning for design and control of particle accelerators: A look backward and forward. *Annual Review of Nuclear and Particle Science*, 74(Volume 74, 2024):557–581, 2024.

[21] Majdi I. Radaideh, Chris Pappas, Jared Walden, Dan Lu, Lasitha Vidyaratne, Thomas Britton, Kishansingh Rajput, Malachi Schram, and Sarah Cousineau. Time series anomaly detection in power electronics signals with recurrent and convlstm autoencoders. *Digital Signal Processing*, 130:103704, 2022.

[22] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, and et al. Overcoming catastrophic forgetting in neural networks. In *Proceedings of the National Academy of Sciences*, volume 114, pages 3521–3526. National Academy of Sciences, 2017.

[23] R. Aljundi and et al. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 2490–2499, 2018.

[24] Simon Hirländer, Sabrina Appel, and Nico Madysa. Data-driven model predictive control for automated optimization of injection into the sis18 synchrotron: Data-driven model predictive control for automated optimization of injection into the sis18 synchrotron. May 2024. 15th International Particle Accelerator Conference : 15th International Particle Accelerator Conference ; Conference date: 19-05-2024 Through 24-05-2024.

[25] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[26] Jingyang Qiao, Zhizhong Zhang, Xin Tan, Yanyun Qu, Wensheng Zhang, Zhi Han, and Yuan Xie. Gradient projection for continual parameter-efficient tuning, 2024.

[27] S. Hirlaender et al. Towards few-shot reinforcement learning in particle accelerator control. In *Proc. IPAC'24*, number 15 in IPAC'24 - 15th International Particle Accelerator Conference, pages 1804–1807. JACoW Publishing, Geneva, Switzerland, 05 2024.

[28] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay, 2016.

[29] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks, 2022.

[30] F. Jafarinia et al. Simulation study on a virtual diagnostics concept for x-ray pulse characterisation. In *Proc. 14th International Particle Accelerator Conference*, number 14 in IPAC'23 - 14th International Particle Accelerator Conference, pages 1813–1815. JACoW Publishing, Geneva, Switzerland, 05 2023.

[31] Claudio Emma, Auralee Edelen, Adi Hanuka, Brendan O'Shea, and Alexander Scheinker. Virtual diagnostic suite for electron beam prediction and control at facet-ii, 2021.

[32] Owen Convery, Lewis Smith, Yarin Gal, and Adi Hanuka. Uncertainty quantification for virtual diagnostic of particle accelerators. *Phys. Rev. Accel. Beams*, 24:074602, Jul 2021.