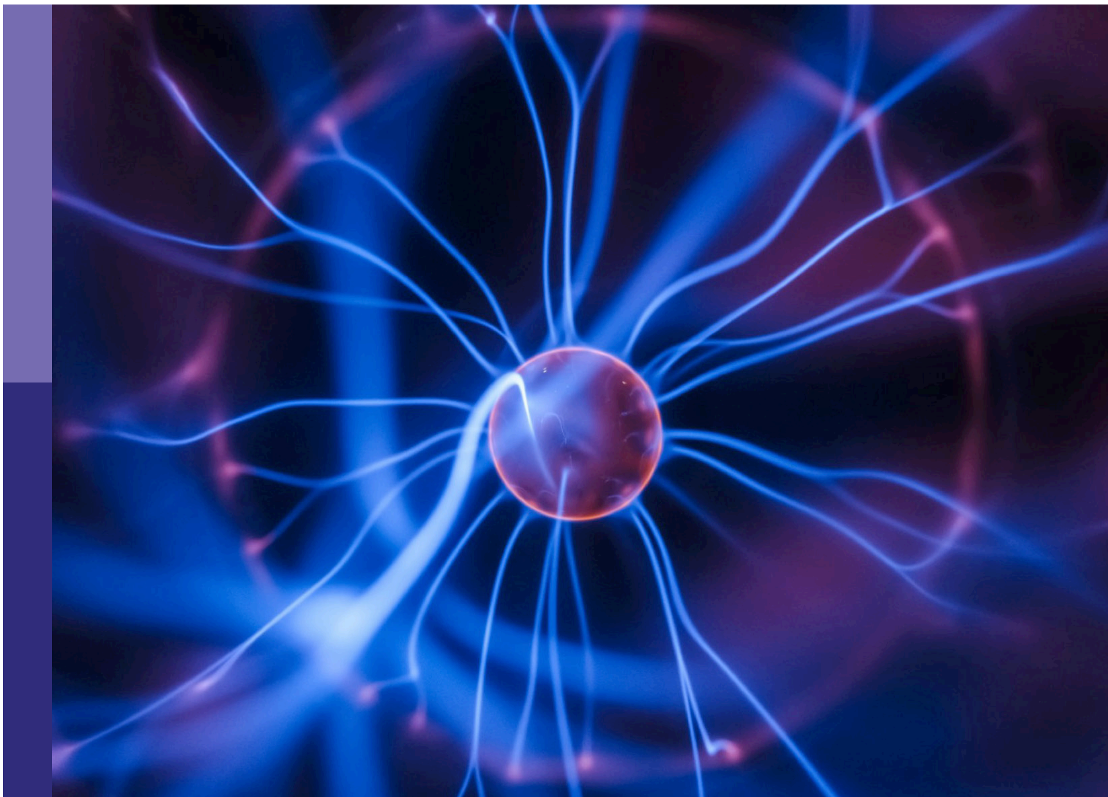
Research Topics

Application of artificial intelligence and machine learning to accelerators

Edited by
Robert Garnett, Alexander Scheinker, Dean Lee
and Andreas Adelmann

Published in
Frontiers in Physics



ISSN 1664-8714
ISBN 978-2-8325-3774-9
DOI 10.3389/978-2-8325-3774-9

About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

Application of artificial intelligence and machine learning to accelerators

Topic editors

Robert Garnett — Los Alamos National Laboratory (DOE), United States
Alexander Scheinker — Los Alamos National Laboratory (DOE), United States
Dean Lee — Michigan State University, United States
Andreas Adelmann — Paul Scherrer Institut (PSI), Switzerland

Citation

Garnett, R., Scheinker, A., Lee, D., Adelmann, A., eds. (2023). *Application of artificial intelligence and machine learning to accelerators*. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-8325-3774-9

Table of contents

ORIGINAL RESEARCH
published: 25 April 2022
doi: 10.3389/fphy.2022.875889

Input Beam Matching and Beam Dynamics Design Optimizations of the IsoDAR RFQ Using Statistical and Machine Learning Techniques

Daniel Koser[1]*, *Loyd Waites*[1], *Daniel Winklehner*[1], *Matthias Frey*[2,3], *Andreas Adelmann*[3] and *Janet Conrad*[1]

[1]Laboratory for Nuclear Science, Massachusetts Institute of Technology, Cambridge, MA, United States, [2]Mathematical Institute, University of St Andrews, St Andrews, UK, [3]Laboratory for Simulation and Modelling, Paul Scherrer Institut, Villigen, Switzerland

We present a novel machine learning-based approach to generate fast-executing virtual radiofrequency quadrupole (RFQ) particle accelerators using surrogate modelling. These could potentially be used as on-line feedback tools during beam commissioning and operation, and to optimize the RFQ beam dynamics design prior to construction. Since surrogate models execute orders of magnitude faster than corresponding physics beam dynamics simulations using standard tools like PARMTEQM and RFQGen, the computational complexity of the multi-objective optimization problem reduces significantly. Ultimately, this presents a computationally inexpensive and time efficient method to perform sensitivity studies and an optimization of the crucial RFQ beam output parameters like transmission and emittances. Two different methods of surrogate model creation (polynomial chaos expansion and neural networks) are discussed and the achieved model accuracy is evaluated for different study cases with gradually increasing complexity, ranging from a simple FODO cell example to the full RFQ optimization. We find that variations of the beam input Twiss parameters can be reproduced well. The prediction of the beam with respect to hardware changes, e.g., the electrode modulation, are challenging on the other hand. We discuss possible reasons for that and elucidate nevertheless existing benefits of the applied method to RFQ beam dynamics design.

Keywords: radio frequency quadrupole, beam dynamics design, beam matching, virtual accelerator, isodar, surrogate modelling, neural network, polynomial chaos expansion

OPEN ACCESS

Edited by:
Frank Franz Deppisch,
University College London,
United Kingdom

Reviewed by:
Gianluca Valentino,
University of Malta, Malta
Baoxi Han,
Oak Ridge National Laboratory (DOE),
United States

****Correspondence:***
Daniel Koser
dkoser@mit.edu

Specialty section:
This article was submitted to
High-Energy and Astroparticle Physics,
a section of the journal
Frontiers in Physics

Received: *14 February 2022*
Accepted: *22 March 2022*
Published: *25 April 2022*

Citation:
Koser D, Waites L, Winklehner D, Frey M, Adelmann A and Conrad J (2022) Input Beam Matching and Beam Dynamics Design Optimizations of the IsoDAR RFQ Using Statistical and Machine Learning Techniques. Front. Phys. 10:875889. doi: 10.3389/fphy.2022.875889

1 INTRODUCTION

Machine Learning (ML), using statistical methods and Neural Networks (NNs), is quickly becoming a staple of modern computational physics. Their highly successful application in computer vision [1], [2] and the establishment of many software packages that are widely available and standardized (e.g., TensorFlow [3] and Keras [4]) has led to attempts to use ML in almost all fields of science. Particle accelerator physics is no exception, although ML is not as well-established here as in other fields. A few examples of ML in accelerator physics are given in the following. Arguably, the best-established use of ML is image analysis using convolutional neural networks (CNNs). CNNs can be used in beam diagnostics for the analysis of the output of emittance scanners, optical fibers, residual gas monitors,

TABLE 1 | Basic parameters of the IsoDAR-RFQ, corresponding to the previously developed baseline beam dynamics design and the preliminary RF/mechanical design.

Design parameter	Value
RF frequency (MHz)	32.8
Design ion	H_2^+
Design beam current (mA)	6.5
Duty cycle	Cw
Input/output energy (keV)	15 / ~70
Inter-vane voltage (kV)	20.1
Beam transmission (%)	97.3
Trans. input emittance (π mm mrad)	0.30
Trans. output emittance (π mm mrad)	0.34
Long. output emittance (π keV deg)	40.2
Tank diameter (cm)	28
electrode length (cm)	136.5
RF power (kW)	~3.6
Shunt impedance (kΩm)	154

and reconstruction of beam pulse structure [5]. The SwissFEL was tuned using Bayesian optimization [6, 7]. Bayesian optimization, using Gaussian Process models was also used for the Linac Coherent Light Source (LCLS) [8].Another very promising technique, that is also the subject of this paper, is surrogate modelling. We describe the method in detail later. In short, a fast-executing model of a complex system can be produced by training a NN or using Polynomial Chaos Expansion (PCE) on a set of high-fidelity simulations. This fast-executing Surrogate Model (SM) can then be used in an optimization scheme or for on-line feedback during run-time. Some examples of successful use of surrogate models in particle accelerator optimization are given in [9], [10–12], which have demonstrated speedups of one to several orders of magnitude compared to conventional techniques.

To our knowledge, ML has not yet been applied to the design of radiofrequency quadrupole (RFQ) linear accelerators. Here we report our recent results using surrogate modelling to create virtual RFQ models that can be used in several ways:

- Uncertainty Quantification (UQ) [13] of the RFQ with respect to input beam variations or RFQ settings during run-time.
- Prediction of output beam parameters from a given set of input beam parameters. The SM becomes a virtual accelerator, ideal as tuning and commissioning aid.
- Design and optimization of the RFQ hardware. Based on the success as a virtual accelerator, we also tested the SM technique as a hardware optimization tool.

The findings in this paper are fully transferable to other RFQs.

1.1 Particle Physics Motivation for This Work

The motivation for this work lies in the IsoDAR project [13]; [14,15], a proposed search for exotic neutrinos. These are hypothesized cousins to the three known standard model neutrinos and could explain anomalies seen in the neutrino oscillation experiments of the past 3 decades [16].

To reach discovery-level sensitivity ($>5\sigma$) in 5 years of running, IsoDAR requires a 10 mA cw proton beam at 60 MeV on a neutrino production target. This accelerator (described in Refs. [17], [15,18]) accelerates H_2^+ ions instead of protons and uses a novel RFQ direct injection method [19], [15], in which the beam is aggressively pre-bunched in an RFQ that is embedded axially in the cyclotron yoke and brought very close to the cyclotron median plane. Because of the high beam current, necessarily small diameter (as little yoke iron as possible must be removed), and the difficult matching of the RFQ output to the cyclotron acceptance, we have initiated this study to accurately predict the sensitivity of the RFQ, the output beam parameters, and to optimize the RFQ design beyond the current baseline. In **Table 1**, we list the most important parameters of the IsoDAR RFQ, some of which will be used as design variables (DVARs) and objectives (OBJs) in the reported study.

1.2 The Structure of This Paper

We have structured this manuscript into Methodology, Results, and Discussion. In each section, we describe our work separately for the two applications of the SM: 1. As Tuning and Commissioning Tool; 2. As Design and Optimization Tool. These are the natural applications due to the immense speedup of SMs compared to high-fidelity Particle-In-Cell (PIC) simulations. We also present results for a very simple system—the FODO cell—as a benchmark and to elucidate the basic principles and challenges. In the Results, we show that the SM performs excellently as a tuning tool, but issues arise when we vary the hardware parameters of the RFQ. In the Discussion we elaborate possible aspects relevant for the surrogate model to under-perform when the beam dynamics is affected by hardware (design parameter) changes, e.g., space charge, number of design variables or neural network topology.

1.2.1 The Surrogate Model as Tuning and Commissioning Tool

The first application we present is using the SM as an on-line feedback tool during the commissioning and running of the RFQ direct injection prototype. We envision the SM to provide valuable assistance for the operator to allow quick or automated adjustment of the RFQ and beamline settings with respect to the input beam properties. To this end, in the final application, we will train the SM using simulated input values like the signal of beam position monitors (BPMs), the beam current (from an AC Current Transformer [20]), and beam size (from a wire probe) before the RFQ and predict the signals from similar devices after the RFQ. To test the idea in this manuscript, we use the Twiss parameters [21] of the beam as input.

1.2.2 The Surrogate Model as Design and Optimization Tool

Finding an optimized beam dynamics design often requires a very large number of simulation iterations. This makes the design procedure of RFQs time consuming, especially when completely new solutions to meet the required beam output quality need to be explored. This is sometimes even the case for comparatively fast executing beam dynamics codes like PARMTEQM [22] or

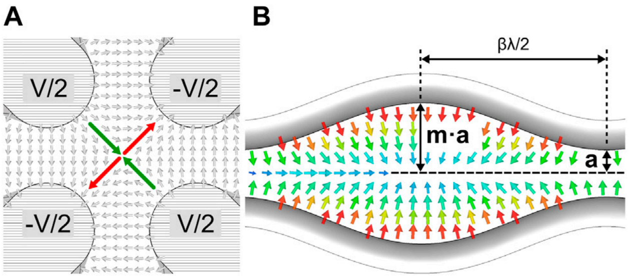


FIGURE 1 | Transverse electric quadrupole field around the beam axis of an RFQ **(A)** with focusing/defocusing plane (green/red) and electrode cell modulation **(B)**, resulting in a longitudinal field component.

RFQGen [23], but is definitely a problem when very time consuming PIC simulations are used as the basis for optimization. Similar to demonstrated successes with cyclotrons and electron accelerators [11,24], we are investigating the use of SMs to perform multiobjective optimization for the RFQ modulation cell parameters, in order to yield minimum beam output emittances (transverse and longitudinal) and maximum transmission.

2 METHODOLOGY

2.1 Surrogate Modelling

Surrogate models are cheap alternatives to reduce the computational complexity of multiobjective optimizations as already shown in the context of particle accelerators in [25]. We chose neural networks and polynomial chaos expansions to replace the high-fidelity RFQ model codes. These methods are explained in the following subsections. More detailed introductions can be found in the listed references and the references contained therein.

2.1.1 Polynomial Chaos Expansion

The principle of the polynomial chaos expansion (PCE) relies on the orthogonality of the multivariate polynomials Ψ_i. The high-fidelity model $m(x)$ with input vector $x \in \mathbb{R}^d$ and $d \geq 1$ is approximated by

$$m(x) \approx \hat{m}(x) = \sum_{i=1}^{P} c_i \Psi_i(\xi) = \sum_{i=1}^{P} c_i \prod_{j=1}^{d} \psi_j(\xi_j) \tag{1}$$

where

$$P = \frac{(p+d)!}{p!d!} \tag{2}$$

is the total number of monomials determined by the expansion truncation order p and the dimensionality of the system d. The vector $\xi = (\xi_1, \ldots, \xi_d)$ represents the input vector that is mapped onto the support of the univariate polynomials ψ_j. The type of the univariate polynomials of the jth dimension depends on the distribution of the corresponding input dimension. For example, uniformly distributed dimensions are approximated by Legendre polynomials and normally distributed dimensions by Hermite polynomials.

There are multiple methods to obtain the expansion coefficients c_i with different requirements on the number of training points N. Commonly used methods are orthogonal projection, regression and Bayesian. In the case of the projection method, the number of training points grows exponentially with the dimension, i.e.,

$$N = (p+1)^d. \tag{3}$$

Regression and Bayesian approaches have no strict requirements, but according to [26] an optimal number of samples is given by

$$N = (d-1)P. \tag{4}$$

A benefit of PCE based surrogate models is the evaluation of Sobol' indices [26], a measure of global sensitivity of the output on the input. The first-order Sobol' index, also known as main sensitivity, quantifies the effect of a single input dimension. The total effect of an input dimension, that also includes all correlations with other dimensions, is denoted as total sensitivity.

We also refer the interested reader to the following literature [26] (and the references therein). Many PCE literature references can also be found in the bibliography of [27].

2.1.2 Artificial Neural Networks

The term "Artificial Neural Network" (ANN) refers to a broad class of methods within Machine Learning (ML) that share the common property of consisting of many interconnected processing units that are used to transform data. The first of such a hierarchy of layers, consists of an affine linear function T: $\mathbb{R}^n \rightarrow \mathbb{R}^m$, defined as $T(x) := Wx + b$, where $W = (a_{ij}) \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $n, m \in \mathbb{N}$. W and b are commonly referred to as the weights and biases of the ANN. The second is an activation function σ: $\mathbb{R} \rightarrow \mathbb{R}$, which is typically nonlinear. Many variants of σ exist, in this work we use the rectified linear unit $\sigma(x) = \max(0, x)$.

The activation function is applied in an element-wise manner, hence a vector activation function σ: $R^n \rightarrow R^n$ can be defined. Now we are able to define a continuous function $f(x)$ by a composition of linear transforms T^i and activation functions σ, i.e.,

$$f(x) = T^k \circ \sigma \circ T^{k-1} \circ \cdots \sigma \circ T^1 \circ \sigma \circ T^0(x), \tag{5}$$

with $T^i(x) = W_i x + b_i$. W_i are initially undetermined matrices and b_i initially undetermined vectors and $\sigma(\cdot)$ is the element-wise activation function. The values of W_i and b_i are randomly initialized and adjusted during "training" using an optimization algorithm to maximize some performance metric.

Such an ANN is called a $(k+1)$-layer ANN, which has k hidden layers. Denoting all the undetermined coefficients (e.g., W_i and b_i) in **Eq. 5** as $\theta \in \Theta$, where θ is a high dimensional vector and Θ is the span of θ, the ANN representation of a continuous function can now be viewed as

$$f = f(x; \theta). \tag{6}$$

Let $\mathbb{F} = \{f(\cdot, \theta) | \theta \in \Theta\}$ denote the set of all expressible functions by the ANN parameterized by $\theta \in \Theta$, then $\mathbb{F}$ provides an efficient way to represent unknown continuous functions.

Approximation properties of neural network can be found in [28], [29], where the authors studied approximation properties

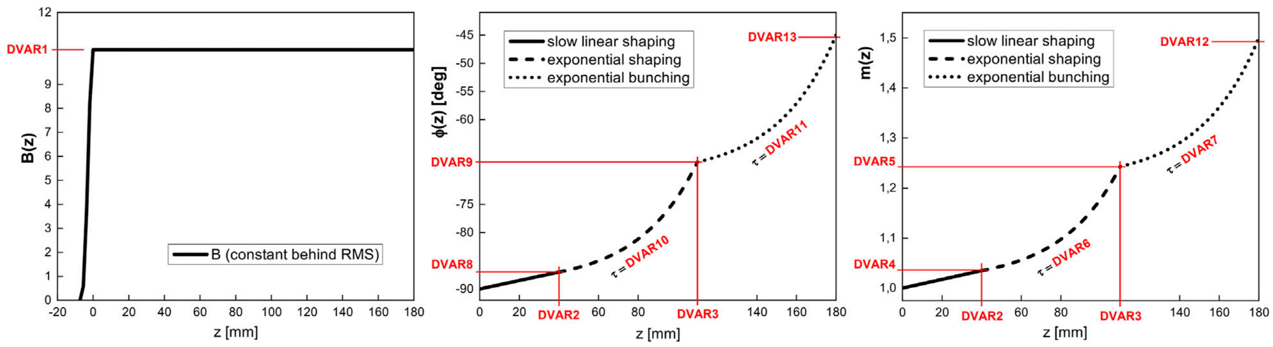


FIGURE 2 | Parametrization functions for the RFQ cell properties specified by design variables (DVARs): The transversal focusing parameter *B*(*z*) is kept constant behind the Radial Matching Section (RMS), with DVAR1 determining the absolute value. Regarding the synchronous phase *ϕ*(*z*) and the electrode modulation *m*(*z*), the RFQ is subdivided into three sections (slow linear shaping, exponential shaping and exponential bunching), the lengths of which are defined by DVARs 2 and 3. The total slope and the smoothness of the occurrence of the shaping/bunching effect are characterized by DVARs 4–13. Qualitatively, this overall design approach corresponds to a previously developed beam dynamics design using the PARMTEQM RFQ design tools and additionally applying manual changes to the design functions.

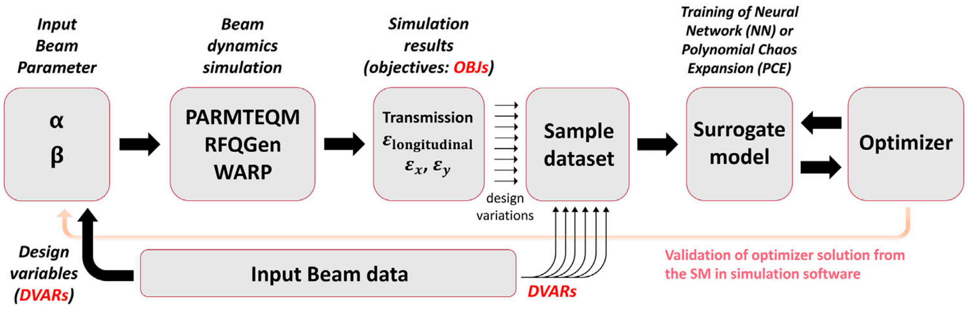


FIGURE 3 | General machine learning optimization scheme for RFQ beam dynamics.

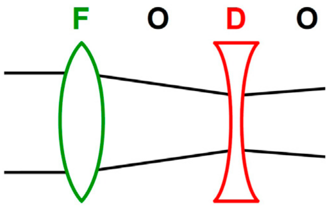


FIGURE 4 | Schematic depiction of a FODO cell, showing a transverse projection of the beam envelope undergoing focusing (F), drift (O), defocusing (D) and drift (O).

TABLE 2 | Hyperparameter boundaries for neural network hyperparameter scan and the best determined value for each case.

	Scan boundaries	Best value
Depth	2 to 40	6
Width	3 to 160	54
Learning rate	0.1 to 0.0001	0.0013
Batch size	8, 16, 32, 128 or 256	256
Activation function	Relu, Tanh or Sigmoid	Relu
L2 Regularization Penalty	0.001 to 0.05	0.018
Gaussian noise	0.001 to 0.1	0.008
Loss function	Mean square error	—
Epochs	Up to 10000	—

for the function classes given by a feed-forward neural network with a single hidden layer. In later works, authors studied the error estimates for such neural networks in terms of hyper-parameters such as number of neurons, layers of the network, and activation functions, a review can be found in [30] and [31].

2.2 Data Generation for Surrogate Modelling

The beam dynamics properties of an RFQ with a number of n modulation cells are fully described by the parameter sets $\mathbf{B} = (B_1, .., B_n)$, $\mathbf{m} = (m_1, .., m_n)$ and $\phi_s = (\phi_{s,1}, .., \phi_{s,n})$, quantifying the basic functions of an RFQ as explained in the sequel:

- The transversely defocusing effect of the space charge force has a $1/\gamma^2$-dependency (γ being the Lorentz factor) and hence at low beam velocities efficient and velocity-independent transverse focusing is required. As shown in **Figure 1**, the alternating electric quadrupole field between

TABLE 3 | Input beam design variables to the fixed FODO cell lattice generated using OPAL, and the range of their parameter space.

Parameter	Value
Corx	–0.5 to 0.5
Cory	–0.5 to 0.5
Beam current (mA)	2 to 10
RMS t (MeV deg)	0.0001 to 0.0005
RMS x (m)	0.001 to 0.005
RMS y (m)	0.001 to 0.005

TABLE 4 | Design variables and range of their parameter space for the FODO lattice system with varying beam and cell parameters.

Parameter	Value
Beam Current (mA)	–0.5 to 10
K1 (m^{-2})	4.2 to 4.8
K2 (m^{-2})	5.2 to 5.7
RMS t (MeV deg)	0.0001 to 0.0005
RMS x (m)	0.001 to 0.005
RMS y (m)	0.001 to 0.005

the RFQ electrodes leads to a focusing force along one of the transverse axes while defocusing occurs in the perpendicular direction, effectively constituting an alternating gradient focusing channel. The transverse focusing strength in an RFQ cell n is commonly characterized by the parameter B_n [32].

- By adding a sinusoidal modulation to the electrode shape, a longitudinal field component is generated which can be used to adiabatically bunch the DC input beam. This is a highly delicate procedure due to the high sensitivity of space-charge dominated beams to perturbations of the beam particle density. The consecutive modulation cells form a π-mode accelerator structure with a cell length of $\ell_c = \beta_c \lambda_{\mathrm{RF}}/2$. The extent of electrode modulation (corresponding to the magnitude of the longitudinal field component) of a cell n is parameterized by the modulation factor m_n.
- The synchronous phase $\phi_{\mathrm{s},n}$, which is set by the cell lengths, determines the ratio of longitudinal bunching to acceleration and hence the overall phase space stability. By increasing $\phi_{\mathrm{s},n}$ along the RFQ, beam acceleration is gradually introduced.

Ultimately, the beam output properties depend on the RFQ hardware specifications as well as on the given input beam parameters, which for a DC input beam are specified by the transverse emittances, the Twiss parameters and the beam current.

2.2.1 Simulated Data for a Fixed Radiofrequency Quadrupole Design

To investigate the capability of surrogate models to reproduce the RFQ beam output properties as a function of only the adjustable beam input parameters (in our case the Twiss parameters α and β), we used a fixed preliminary optimized RFQ design, through which we simulated the beam using the PARMTEQM code. A sample data set was obtained from the output of a number of PARMTEQM simulations with randomized values for the input Twiss parameters (corresponding to the design variables of the underlying optimization problem) within a predefined range of α = [1, 4] and β = [7, 25] (cm/mrad). The transverse and longitudinal output emittances as well as the transmission (constituting the optimization objectives) were evaluated directly at the end of the RFQ electrodes.

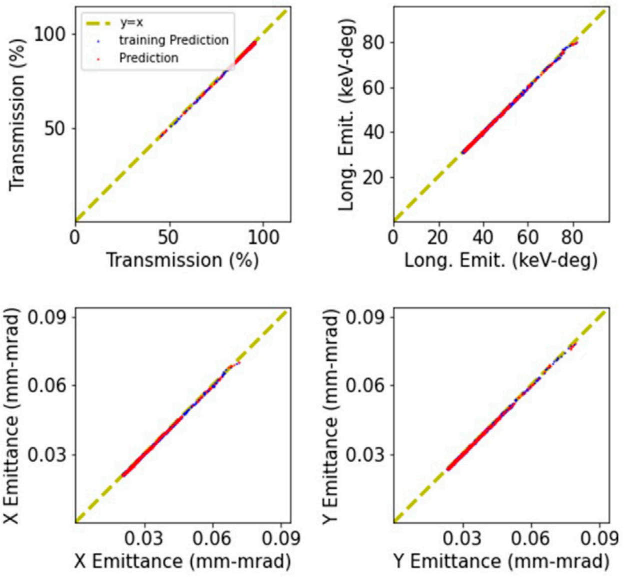


FIGURE 5 | Predictions by the neural network surrogate model as function of the actual data values for variation of only the beam input Twiss parameters to a fixed RFQ (MAEs being well below 1%). The red dots correspond to the test dataset whereas the blue dots are training data.

2.2.2 Simulations of Full Radiofrequency Quadrupole Design

To study the applicability of surrogate models for optimizing the RFQ design itself, we introduced a parameterization of the functions for transverse focusing $B(z)$, synchronous phase $\phi(z)$ and electrode modulation $m(z)$ according to **Figure 2**. This reduces the size of the RFQ design parameter space, corresponding to the number of design variables, from $3n + 1$ (B_n, $\phi_{\mathrm{s},n}$, m_n for each cell n, + 1 because the number of cells is a design variable itself) to a total number of 14.

The parameterization functions were chosen so that the crucial properties of the underlying baseline design remain variable for optimization; e.g. the constant value of $B(z)$ behind the Radial Matching Section (RMS) (corresponding to DVAR1), the lenghts of the linear and exponential shaping and bunching sections (DVAR2 and DVAR3) as well as the rate and smoothness of shaping and bunching (DVARs 5–13). The length of the RFQ is determined by DVAR14, being the cutoff energy after which PARMTEQM ends the electrode (always with a full RFQ cell).

We generated a sample data set from beam dynamics simulations using PARMTEQM for a number of random RFQ

TABLE 5 | Optimum set of Twiss parameters found by Bayesian optimizer based on the surrogate model output and corresponding predicted beam output parameters with comparison to PARMTEQM results.

Beam parameter	SM output	PARMTEQM output
Input α	2.55	2.55
Input β	16.60 cm/rad	16.60 cm/rad
Transmission (%)	95.5	95.3
ϵ_s (MeV Deg)	0.031	0.031
ϵ_x (mm mrad)	0.021	0.021
ϵ_y (mm mrad)	0.024	0.024

design variations (randomized DVAR values within a predefined range) with a fixed input beam (input Twiss parameters held constant).

2.3 Machine Learning Training and Use of Radiofrequency Quadrupole Surrogate Models

As being best practice for the training of ML models, we randomly split sample datasets into 70% training and 30% test data. A total of 1,000 samples was used for the input beam tuning studies, whereas for the full RFQ optimization with an increased number of design variables, up to 200,000 samples were used. The training data is then used to train either a PCE or NN based surrogate model. After training of the SM, the model predictions are evaluated on both the test and training data by comparison to the original simulation output values. The normalized Mean Absolute Error (MAE) is calculated and reported. To prevent overfitting, the PCE is run repeatedly with increased order to minimize the MAE until the difference between the test and training dataset are more than 5%. In our case, this was at 4th order. A general workflow scheme for surrogate model creation from simulation data is depicted in **Figure 3**.

To design and train neural networks we used the TensorFlow [33] machine learning framework and the hyperparameter optimization tools provided by Keras [34]. These support automated tuning of the neural network hyperparameters, the used boundary values of which are given in **Table 2**. We underwent a new hyperparameter scan for each case, and automatically selected the best hyperparameter configuration with minimized MAE for the training set. The choice of a Relu (Rectified Linear Unit) activation function was found to be the best option for the considered use cases. Eventually, the obtained surrogate model can be saved and used for beam dynamics sensitivity studies and optimization.

Based on the surrogate model, an optimization of the design variables with respect to the objectives using a generic optimizer algorithm can be performed, the result of which (SM output for the best found set of DVARs) can then be validated by the result of the corresponding PARMTEQM beam dynamics simulation output.

3 RESULTS

3.1 Basic FODO Cell Example

The effects of a quadrupole magnet on an ion beam causes focusing on one transverse spatial axis, while leading to defocusing in the perpendicular direction. However, using

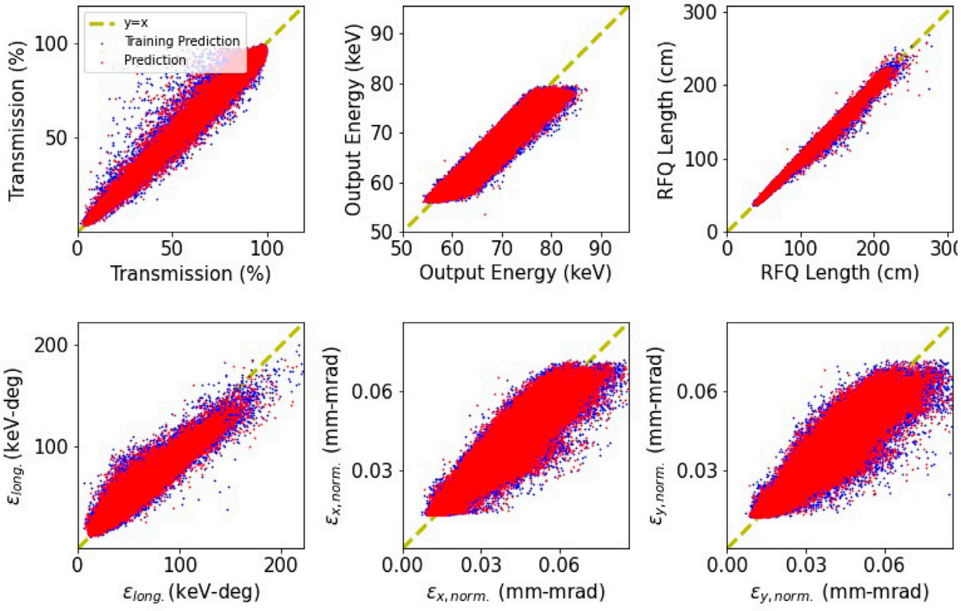


FIGURE 6 | Surrogate model predictions as function of the actual data values for full RFQ design variation by 14 DVARs and fixed beam input Twiss parameters. Again, the red dots correspond to the test dataset and the blue dots are training data.

TABLE 6 | Comparison between mean average errors (MAEs) for surrogate models based on polynomial chaos expansion (PCE) and neural networks (NN) for different optimization cases and objectives.

MAE's (%)	Input beam optimaization (2 DVARs)		Full RFQ optimaization (14 DVARs)		FODO cell		
					Variation of beam input only		Variation of quadrupole focusing
	PCE	NN	PCE	NN	PCE	NN	NN
Transmission	0.17	0.15	3.5	2.4	No variation		
$\varepsilon_{\text{longitudinal}}$	0.72	0.57	10.5	8.2	0.76	5.8	13.6
ε_x	1.85	0.55	13.2	12.8	0.19	1.8	5.1
ε_y	0.74	0.71	13.3	12.5	0.95	6.3	10.5
Output energy	No variation		1.8	1.9	No variation		
RFQ length			1.2	2.0	—		

alternating quadrupoles in series can lead to a net focusing effect for the beam. In accelerator physics, one of the most basic examples of this is called a FODO cell, thus named for focusing (F), drift (0), defocusing (D), and again drift (0). This is schematically depicted in **Figure 4**.

In order to demonstrate the feasibility of using machine learning techniques to replicate accelerators, we started by reproducing the beam dynamics of this focusing/defocusing FODO lattice. This is the simplest and most basic example that still features similar transverse beam behavior as in RFQs but with greatly reduced overall complexity, and was therefore decided to be a good case to prove the proposed modelling concept.We computed the FODO cell simulations in OPAL [33], using beam input parameters as summarized in **Table 3**. As shown in **Table 6**, the generated surrogate model of the FODO cell is capable of mapping the beam input parameters accurately to the values of the output emittances (both transversely and longitudinally) with MAEs of less than 1%, regarding the test data set.

3.2 FODO Lattice With Varying Cell Parameters

In addition to manipulating the beam input properties and simulating the beam through a fixed FODO cell, we also investigated the case of a variable hardware setup by using the focusing strengths K1 and K2 of the FODO cell quadrupole magnets as design variables. A summary of all design variables of the investigated system is given in **Table 4**.

This scenario resulted in significantly larger errors compared to the fixed cell example where variation was restricted to the input beam properties. A more detailed discussion of this issue is given later in the discussion section of this paper. The yielded MAE values can again be found in **Table 6**.

3.3 Creating a Beam Dynamics Tuning Tool for an Radiofrequency Quadrupole

Next, we created a surrogate model with the aim to reproduce the beam dynamics behavior through the RFQ, given a fixed RFQ and variable LEBT input parameters. As summarized in **Table 6**, a very high model accuracy could be achieved (using either PCE or NN) with values of the normalized MAEs typically being below 1%, regarding transmission and emittances. Corresponding accuracy plots are shown in **Figure 5**.

Because executing the surrogate models takes only about $7 \cdot 10^{-4}$ s, given the used computer hardware and software specification, this method can be used to rapidly model the RFQ output for different inputs from the LEBT, allowing to compare simulations and commissioning data in real time. We have thus been able to create a real time, accurate tool for use during the commissioning phase of our RFQ.

Furthermore, we were able to use the same surrogate model to optimize the input beam Twiss parameters (α and β) given a fixed RFQ setup.

Due to the high-fidelity of the achieved surrogate model, the intended optimization of the input beam Twiss parameters for RFQ injection could be performed using a Bayesian optimizer [35], with the SM as the test function and maximum output transmission and minimum output emittances as optimization objectives.

To cross check the optimization results based on the SM, the found optimum set of Twiss parameters was used to validate the predicted SM output by PARMTEQM simulations. The optimum Twiss parameters found for a preliminary revised design of the IsoDAR RFQ are given in **Table 5** together with the predicted beam output parameters by the SM and the corresponding PARMTEQM output. Deviations between the simulation and the SM prediction, i.e., optimization result, are less than 0.2% for both transmission and emittance values.

3.4 Optimization of the Entire Radiofrequency Quadrupole Beam Dynamics Design on the Basis of Surrogate Models

Ultimately, we used the 14-DVAR RFQ model sample data set to train PCE and NN based models. Corresponding accuracy plots can be seen in **Figure 6** and achieved MAEs are again summarized in **Table 6**.Similar to the previous case, the obtained surrogate models execute much faster than their simulation counterparts. Whereas the calculation of a SM prediction takes around 10^{-3} s, a corresponding physics beam dynamics simulation with PARMTEQM of a short IsoDAR type RFQ with an electrode length of around 1.3 m consumes up to around 40 s. With a sufficiently large design space, this

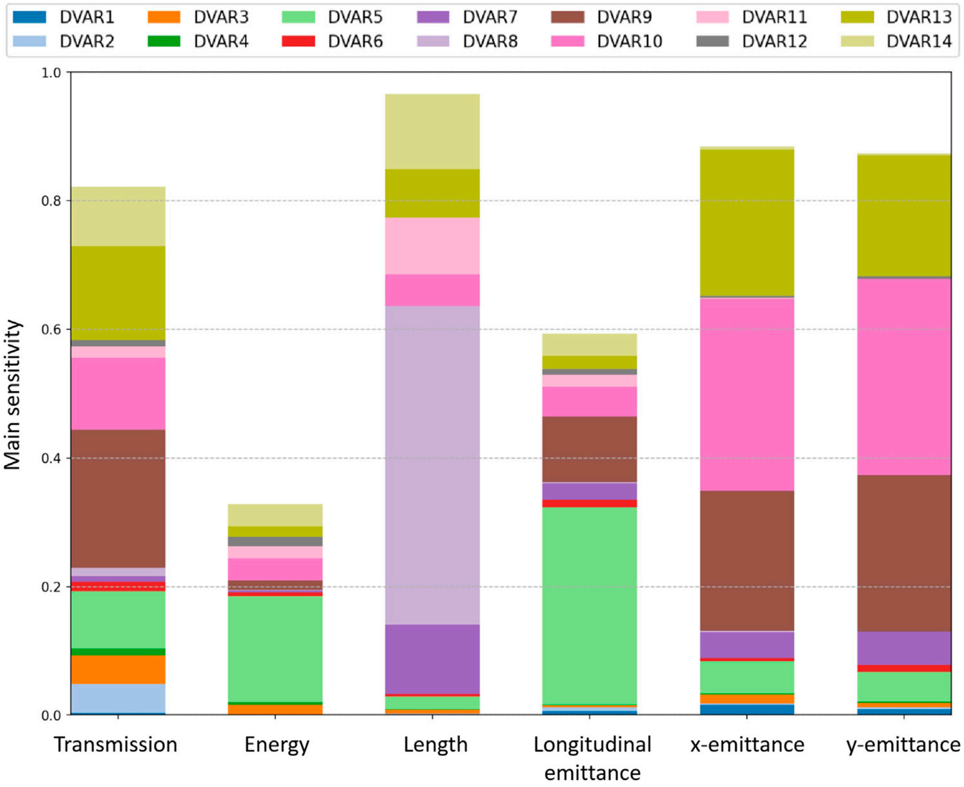


FIGURE 7 | Sensitivity plot for the full RFQ optimization with 14 design variables (DVARs).

significantly reduces the time to find an optimized RFQ beam dynamics design.

With MAEs of the predicted output emittances of up to 10% (the MAEs for the transmission however being noticeably smaller) we found the surrogate models currently do not provide decent enough accuracy in any of the considered cases to perform a full RFQ design optimization. However, these computationally inexpensive surrogate models can be used to perform a rough pre-optimization with respect to the beam output objectives, providing a starting point for fine tuning optimizations using beam dynamics simulation tools. Using these methods combined reduces the total computational need of RFQ optimization and allows to quickly explore different possible qualitative solution approaches.

4 DISCUSSION

The created surrogate models quickly proved to be a reliable rapid-use tool for observing the effects of input beam variations on the output beam properties of a given RFQ. This has been a useful tool in optimizing the LEBT design, and could be as much as useful during commissioning and tuning of the LEBT/RFQ system. Ultimately, we found that highly accurate ($< 1\%$ mean average error, MAE) RFQ surrogate models can be obtained for the optimization of only the input beam Twiss parameters (2 DVARs).This also matches our experience from previous studies on the simplistic test case of modelling the beam dynamics in a FODO lattice under variation of only the beam input parameters. For this highly simplified case, an optimization based on the surrogate model could also be performed with small deviations of the results to the beam dynamics simulation.In general, the use of neural networks (NN) seems to lead to more accurate surrogate models compared to polynomial chaos expansion (PCE).

On the other hand, however, the application of the developed techniques to the full RFQ beam dynamics design optimization proved problematic due to increased errors in predicted emittance whenever the space of design variables was expanded to include physical changes to the RFQ. This problem also already occurred in the case of the FODO cell. As shown in **Figure 6** and summarized in **Table 6**, models that include structural changes of the accelerator hardware system, such as variation of the FODO cell focusing strengths and the full RFQ optimization, suffer from errors in the emittances prediction $> 10\%$.In none of the problematic cases did the error values improve significantly by switching off space charge (beam dynamics simulation with zero-current). When comparing the FODO cell example with the full RFQ optimization, it seems that the higher errors result not from a larger number of design variables, but are only introduced in case that the design variables affect the structure of the accelerator itself.While the yielded errors are too high to do a full hardware optimization of the RFQ system, surrogate modelling still proved useful to eliminate large areas of the design parameter space. With a reduced design space, the accelerator can then be fine tuned using more accurate, computationally expensive models in the region of interest. Similar behaviour of the SM's are reported in [35]. For example, Figure 13 and Figure 14 (in [35]) show a comparable difference in accuracy.Future work will include the investigation of our systems with regards to hidden variables and the use of other neural network topologies that are not fully connected. It seems possible that the errors may be further reduced by altering the structure of the neural network, while maintaining the high computational speed.

As depicted in **Figure 7**, the surrogate model lends itself to perform sensitivity analyses investigating the impact of DVAR variation on the optimization objectives.

Eventually, this allows for an evaluation of the cell properties parameterization model and to reduce the number of DVARs by omitting design variables with little effect on the crucial optimization objectives.

In case of our specific RFQ, the sensitivity chart (**Figure 7**) reaveals that variation of DVARs 9, 10 and 13 (all relating to the function $\phi(z)$ of the synchronous phase) have the most significant influence on the transverse emittances, while the longitudinal emittance seems to be most sensitive to DVAR5 (value of the modulation factor $m(z)$ at the end of the exponential shaping section). Potential DVAR variations that might be omitted for the optimization procedure apparently relate to DVAR1 (value of the transverse focusing parameter $B(z)$ = const.) and DVARs 2 and 4 (properties of $m(z)$ in the slow linear shaping section) as well as DVARs 3 and 6 (properties of $m(z)$ in the exponential shaping section).

5 CONCLUSION

In this paper, we applied a recently developed surrogate modelling technique to the optimization of the beam output quality of RFQ linear accelerators for the first time. We tested our method on a simple FODO cell (having similar transverse focusing properties) first and on the IsoDAR RFQ thereafter. To create the surrogate models, we used polynomial chaos expansion and deep neural networks. We compared the results and found that we could very accurately predict the beam behaviour from varying input beam parameters as it goes through a fixed accelerator structure, which initially was our main goal. The trained model is intended to be used as an online feedback tool in the commissioning and tuning of the IsoDAR injector. Furthermore, we found that, when we train the surrogate model on sets of hardware parameters (i.e., many different design configurations of the investigated machine), we incur much higher training and validation errors. We are in the process of investigating the cause of this effect, and we can already say that, in a comparison between beams with and without space charge, we do not see a difference. Despite the large training errors (up to 10%), the surrogate models trained on hardware design variables can be used to perform preliminary optimization of the design, reducing the model space, followed by a second iteration using high-fidelity physics simulations. Furthermore, Sobol's indices can be used to elucidate the influence of single design variables on the objectives, allowing restricting design variations to the most crucial parameters.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

DK, DW, and LW contributed to conception and design of the study. DK conducted all RFQ beam dynamics studies. LW performed the machine learning studies. DW and JC supervised the project. AA and MF provided the Jupyter notebooks (PCE scripts and NN topology) for the machine learning studies and provided support on all machine learning problems, and also provided the OPAL simulations. DK, LW, and DW wrote the first draft of the manuscript and MF wrote the theory section. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This work was supported by NSF grants PHY-1505858 and PHY-1626069 and funding from the Bose Foundation and the Heising-Simons Foundation.

REFERENCES

1. Ren S, He K, Girshick R, Sun J. Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks. In: C Cortes, N Lawrence, D Lee, M Sugiyama, R Garnett, editors. *Advances in Neural Information Processing Systems*, 28. Curran Associates, Inc. (2015).
2. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016). 770–8. doi:10.1109/CVPR.2016.90
3. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al.*TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems* (2015). Software available from tensorflow.org [Dataset].
4. Chollet F. *Keras* (2015). [Dataset].
5. Ren X, Edelen A, Lutman A, Marcus G, Maxwell T, Ratner D, Temporal Power Reconstruction for an X-ray Free-Electron Laser Using Convolutional Neural Networks. *Phys Rev Acc Beams*. 23 (2020). American Physical Society. 040701. doi:10.1103/PhysRevAccelBeams.23.040701.Publisher
6. Kirschner J, Mutny M, Hiller N, Ischebeck R, Krause A. *Adaptive and Safe Bayesian Optimization in High Dimensions via One-Dimensional Subspaces* (2019). *arXiv:1902.03229 [cs, stat]* ArXiv: 1902.03229.
7. Kirschner J, Nonnenmacher M, Mutny M, Krause A, Hiller N, Ischebeck R, et al. *Bayesian Optimisation for Fast and Safe Parameter Tuning of SwissFEL*. JACoW Publishing (2019). p. 707–10. doi:10.3929/ethz-b-000385955
8. Duris J, Kennedy D, Hanuka A, Shtalenkova J, Edelen A, Baxevanis P, et al. Bayesian Optimization of a Free-Electron Laser. *Phys Rev Lett* (2020) 124:

124801. American Physical Society. doi:10.1103/PhysRevLett.124.124801. Publisher

9. Adelmann A. On Nonintrusive Uncertainty Quantification and Surrogate Model Construction in Particle Accelerator Modeling. *Siam/asa J Uncertainty Quantification* (2019) 7:383–416. doi:10.1137/16M1061928
10. Van Der Veken F, Azzopardi G, Blanc F, Coyle L, Fol E, Giovannozzi M, et al. Machine Learning in Accelerator Physics: Applications at the CERN Large Hadron Collider. *SISSA Medialab* (2020) 372:044. doi:10.22323/1.372.0044
11. Edelen AL, Biedron SG, Milton SV, Edelen JP. *First Steps toward Incorporating Image Based Diagnostics into Particle Accelerator Control Systems Using Convolutional Neural Networks* (2016). *arXiv:1612.05662 [physics]* ArXiv: 1612.05662.
12. Edelen A, Neveu N, Frey M, Huber Y, Mayes C, Adelmann A. Machine Learning for Orders of Magnitude Speedup in Multiobjective Optimization of Particle Accelerator Systems. *Phys Rev Accel Beams* (2020) 23:044601. doi:10.1103/PhysRevAccelBeams.23.044601
13. Bungau A, Adelmann A, Alonso JR, Barletta W, Barlow R, Bartoszek L, 109. American Physical Society (2012). p. 141802. doi:10.1103/PhysRevLett.109.141802.PublisherProposal for an Electron Antineutrino Disappearance Search Using High-Rate $\\{8\\} \mathrm\\{Li\\}$ Production and Decay*Phys Rev Lett*
14. Abs M, Adelmann A, Alonso JR, Axani S, Barletta WA, Barlow R, et al. *IsoDAR@KamLAND: A Conceptual Design Report for the Technical Facility* (2015). *arXiv:1511.05130 [hep-ex, physics:physics]* ArXiv: 1511.05130.
15. Winklehner D, Bahng J, Calabretta L, Calanna A, Chakrabarti A, Conrad J, et al. enHigh Intensity Cyclotrons for Neutrino Physics. *Nucl Instr Methods Phys Res Section A: Acc Spectrometers, Detectors Associated Equipment* (2018) 907:231–43. doi:10.1016/j.nima.2018.07.036
16. Diaz A, Argüelles CA, Collin GH, Conrad JM, Shaevitz MH. *Where Are We with Light Sterile Neutrinos* (2019). *arXiv:1906.00045 [hep-ex, physics:hep-ph]* ArXiv: 1906.00045.
17. Calanna A, Campo D, Yang JJ, Calabretta L, Rifuggiato D, Maggiore MM, et al. *A Compact High Intensity Cyclotron Injector for DAEdALUS Experiment*, C1205201 (2012). p. 424–6.
18. Winklehner D, Adelmann A, Conrad JM, Mayani S, Muralikrishnan S, Schoen D, et al. *Order of Magnitude Beam Current Improvement in Compact Cyclotrons* (2021). *arXiv:2103.09352 [physics]*.
19. Winklehner D, Hamm R, Alonso J, Conrad J. An RFQ Direct Injection Scheme for the IsoDAR High Intensity Cyclotron $\\mathrm{H}_{2}ˆ+$. In: 6th International Particle Accelerator Conference (2015). *IPAC2015)*.
20. Bergoz. Bergoz - ACCT - Precise Waveform Measurement of Long Pulses (2020). [Dataset] Available at: https://www.bergoz.com/products/acct/.
21. Reiser M. *Theory and Design of Charged Particle Beams*. 2 edn. Weinheim: Wiley VCH (2008).
22. Crandall KR, Wangler TP, 177. PARMTEQ Â€" A Beam-Dynamics Code Fo the RFQ Linear Accelerator. *Linear Accelerator Beam Opt Codes* (1988). AIP Publishing. 22–8.
23. Crandall K, Wangler T, Young L. *RFQ Design Codes*. Los Alamos National Laboratory (2005).
24. Neveu N, Spentzouris L, Adelmann A, Ineichen Y, Kolano A, Metzger-Kraus C, et al. Parallel General Purpose Multiobjective Optimization Framework with Application to Electron Beam Dynamics. *Phys Rev Accel Beams* (2019) 22: 054602. doi:10.1103/PhysRevAccelBeams.22.054602
25. Sudret B. Global Sensitivity Analysis Using Polynomial Chaos Expansions. *Reliability Eng Syst Saf* (2008) 93:964–79. doi:10.1016/j.ress.2007.04.002
26. Sobol' IM. Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates. *Mathematics Comput Simulation* (2001) 55: 271–80. doi:10.1016/S0378-4754(00)00270-6
27. Frey M, Adelmann A. Global Sensitivity Analysis on Numerical Solver Parameters of Particle-In-Cell Models in Particle Accelerator Systems. *Comp Phys Commun* (2021) 258:107577. doi:10.1016/j.cpc.2020.107577
28. Cybenko G. Approximation by Superpositions of a Sigmoidal Function. *Maths Control Signals Syst* (1989) 2:303–14. doi:10.1007/bf02551274
29. Hornik K, Stinchcombe M, White H. Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks* (1989) 2:359–66. doi:10.1016/0893-6080(89)90020-8
30. Ellacott SW. Techniques for the Mathematical Analysis of Neural Networks. *J Comput Appl Math* (1994) 50:283–97. doi:10.1016/0377-0427(94)90307-7
31. Pinkus A. Approximation Theory of the Mlp Model in Neural Networks. *Acta Numerica* (1999) 8:143–95. doi:10.1017/S096249290000291910.1017/s0962492900002919
32. Crandall K, Stokes R, Wangler T. RF Quadrupole Beam Dynamics Design Studies. In: Proceedings of LINAC1979 (1979).
33. Adelmann A, Gsell A, Kraus C, Ineichen Y, Russell S, Bi Y, et al. *The OPAL (Object Oriented Parallel Accelerator Library) Framework*. Tech. Rep. PSI-PR-08-02, Paul Scherrer Institut (2008). 2015.
34. Nogueira F. *Bayesian Optimization: Open Source Constrained Global Optimization Tool for Python*[Dataset] (2014).
35. Bellotti R, Boiger R, Adelmann A. Fast, Efficient and Flexible Particle Accelerator Optimisation Using Densely Connected and Invertible Neural Networks. *Information* (2021) 12:351. doi:10.3390/info12090351

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Physics

ORIGINAL RESEARCH
published: 22 July 2022
doi: 10.3389/fphy.2022.903559

Mixed Diagnostics for Longitudinal Properties of Electron Bunches in a Free-Electron Laser

J. Zhu, N. M. Lockmann, M. K. Czwalinna and H. Schlarb*

Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany

Longitudinal properties of electron bunches are critical for the performance of a wide range of scientific facilities. In a free-electron laser, for example, the existing diagnostics only provide very limited longitudinal information of the electron bunch during online tuning and optimization. We leverage the power of artificial intelligence to build a neural network model using experimental data, in order to bring the destructive longitudinal phase space (LPS) diagnostics online virtually and improve the existing current profile online diagnostics which uses a coherent transition radiation (CTR) spectrometer. The model can also serve as a digital twin of the real machine on which algorithms can be tested efficiently and effectively. We demonstrate at the FLASH facility that the encoder-decoder model with more than one decoder can make highly accurate predictions of megapixel LPS images and coherent transition radiation spectra concurrently for electron bunches in a bunch train with broad ranges of LPS shapes and peak currents, which are obtained by scanning all the major control knobs for LPS manipulation. Furthermore, we propose a way to significantly improve the CTR spectrometer online measurement by combining the predicted and measured spectra. Our work showcases how to combine virtual and real diagnostics in order to provide heterogeneous and reliable mixed diagnostics for scientific facilities.

Keywords: free-electron laser (FEL), longitudinal phase space, machine learning, encoder-decoder, mixed diagnostics

OPEN ACCESS

Edited by:
Alexander Scheinker,
Los Alamos National Laboratory (DOE), United States

Reviewed by:
Daniel Phillips,
Ohio University, United States
Zhi Zeng,
Hefei Institutes of Physical Science (CAS), China

****Correspondence:***
J. Zhu
jun.zhu@desy.de

Specialty section:
This article was submitted to Interdisciplinary Physics, a section of the journal Frontiers in Physics

Received: *24 March 2022*
Accepted: *08 June 2022*
Published: *22 July 2022*

Citation:
Zhu J, Lockmann NM, Czwalinna MK and Schlarb H (2022) Mixed Diagnostics for Longitudinal Properties of Electron Bunches in a Free-Electron Laser.
Front. Phys. 10:903559.
doi: 10.3389/fphy.2022.903559

1 INTRODUCTION

Tuning and optimization of the longitudinal phase space (LPS) of electron bunches are of vital importance for the performance of various scientific facilities such as free-electron lasers (FELs) [1], ultrafast electron diffractions (UEDs) [2], laser-plasma accelerators (LPAs) [3], plasma wakefield accelerators (PWFAs) [4], THz-driven accelerators [5] and so on. The prerequisite of quickly and accurately manipulating the LPS of an electron bunch is being able to measure and monitor it rapidly and reliably. The LPS of an electron bunch is usually measured directly in the time domain by combining a transverse deflecting structure (TDS) and a dipole spectrometer magnet in a dispersive section [6, 7]. An LPS image provides rich and important information such as the shape, the current profile and the slice energy spread of the electron bunch. However, this diagnostic method interferes with delivering photons to user experiments and thus cannot be employed online during machine tuning and optimization. The current profile of an electron bunch can also be reconstructed by measuring the coherent transition radiation (CTR) [8, 9] or the coherent diffraction radiation (CDR) [10] spectrum generated by an electron bunch. Although CTR generation is invasive to an electron bunch, it works as an online diagnostics at facilities such as FLASH, in which a single bunch from a bunch train can be selected for this purpose without interfering with user experiments [8]. On the

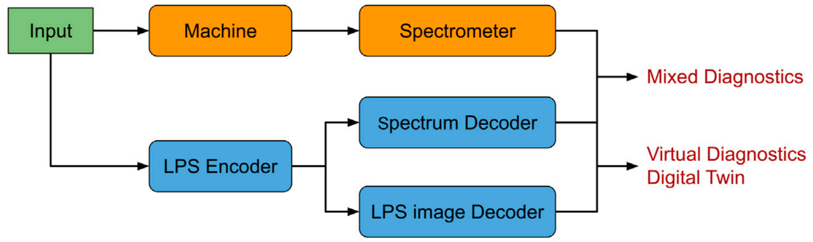


FIGURE 1 | Applications of an encoder-decoder neural network model with more than one decoder for heterogeneous predictions of electron bunch longitudinal properties.

other hand, physics-based beam dynamic simulation plays an important role in understanding the LPS of an electron bunch. However, high-resolution physics-based simulation is extremely time-consuming [11] and often does not agree with the measurement very well.

In recent years, machine learning has demonstrated to be able to learn relationships inside a complex system and produce accurate and fast predictions [12, 13]. Using artificial neural networks as a tool for electron bunch longitudinal property prediction has garnered more and more attention in recent years [14–18]. At the LCLS, early work has demonstrated prediction of LPS images and current profiles at the linac exit using two separated multi-layer perceptrons (MLPs) [14] and a single input parameter. Later, a spectral virtual diagnostics was proposed to improve the prediction accuracy and robustness by using the CDR spectrum of the electron bunch as input [17]. Simulated CDR spectra were used in this study. Indicated by tests with simulated LPS images for the LCLS-II, a neural network is also expected to predict the microbunching structure in the LPS on a shot-to-shot basis using the spectrum as input with a fixed machine set point [17]. The latest work at the European XFEL injector demonstrated that a deep encoder-decoder neural network can achieve extremely high accuracy in predicting megapixel LPS images using up to three RF phases as input. The current profile, energy spectrum and slice energy spread extracted from the predicted LPS image all show very good agreement with the measurement [18]. In addition, an innovative method was demonstrated to efficiently build large models with multiple distinctive working points [18].

In this paper, we experimentally demonstrate training an encoder-decoder neural network model with more than one decoder to predict the LPS image and the CTR spectrum of the electron bunches in a single bunch train concurrently at the end of the FLASH linac. Building a model using the data-driven approach heavily relies on the availability of the data. However, the main focus of a user facility is to deliver electron or photon beams to user experiments and thus cannot allocate a large amount of time exclusively for routine data collection. We demonstrate that only a reasonable amount of data is required to train a performant model around a user working point, with all major control knobs for LPS manipulation included. More importantly, we propose a method which can significantly improve the existing online current profile measurement using a CTR spectrometer by combining predicted and measured CTR spectra. Analogous to mixed reality, we call diagnostics which provide a blend of predicted and measured signals mixed diagnostics. This approach offers the possibility to provide heterogeneous and reliable LPS information in real-time for electron bunches with a broad range of parameters.

The applications of the above encoder-decoder model are summarized in **Figure 1**. The model can serve as not only virtual and mixed diagnostics for online machine tuning and optimization, but also a digital twin of the actual machine on which machine tuning and optimization algorithms [19–24] can be tested efficiently and effectively before applying them in the real world. These virtual experiments can even be performed in advance to find an optimal or near-optimal setup for the real machine. One of the major advantages of testing with neural network models is that it is orders of magnitude faster than real experiments or physics based simulations because the inference time is typically on the order of milliseconds. Compared with neural network models trained with simulated data [25–27], models trained with experimental data provide better testing environments by generating predictions which are almost identical to the real-world signals.

2 EXPERIMENTAL SETUP

FLASH is a soft x-ray free-electron laser (FEL) user facility which is capable of delivering MHz pulse trains to two user experiments, FLASH1 and FLASH2, in parallel with individually selected photon beam characteristics [28, 29]. The layout of the FLASH1 beamline is shown in **Figure 2**. The longitudinal properties of two electron bunches in a bunch train can be measured concurrently by streaking one of the electron bunches using the S-band TDS [30, 31] and picking up another one for the CRISP spectrometer using a fast kicker magnet [8]. It should be emphasized that the two diagnostics cannot measure the same bunch in a bunch train simultaneously. In this study, the electron bunch picked up by the CRISP spectrometer travels immediately after the one streaked by the TDS. A systematic comparison between these two diagnostics was conducted recently, which shows excellent agreement on the current profile down to the 10 fs level [9]. However, due to the non-optimized optics required by the parallel user experiment at the FLASH2 and a cap on the TDS power, the rms time resolution of the measurement using the TDS is larger than 70 fs

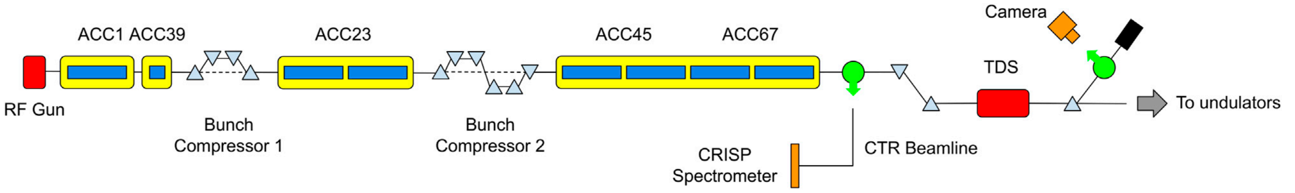


FIGURE 2 | Schematic of the FLASH1 beamline. Components are not to scale. ACC1, ACC23, ACC45 and ACC67 are 1.3 GHz cryomodules which boost the electron energy. ACC39 is a 3.9 GHz cryomodule which linearizes the LPS before the first bunch compression stage. The CTR radiation is generated by deflecting one electron bunch onto an off-axis screen using a fast magnetic kicker and measured by the coherent radiation intensity spectrometer (CRISP).

in our experiments. Moreover, the CRISP spectrometer has two sets of remotely interchangeable grating sets, which cover different frequency ranges. The low-frequency grating set covers the range from 0.7 to 6.6 THz, and the high-frequency grating set covers the range from 6.9 to 58.8 THz. In order to have an accurate reconstruction of the current profile, the full spectra range from 0.7 to 58.8 THz is required. For a meticulous characterization of the current profiles at a few set points, this can be achieved by recording data using the two grating sets consecutively [9]. However, this is not practical for a large number of machine set points. A detailed discussion about how to address this issue when building a neural network model is presented in **Section 4.2**.

Up to 6 control knobs, including the phases and amplitudes of ACC1, ACC39, and ACC23, are scanned during data collection around a user working point because of the following considerations. During machine operation, the RF phases and amplitudes of ACC1, ACC39, and ACC23 are commonly used to adjust the LPS of the electron bunch in order to optimize the FEL performance. The LPS is very sensitive to the phase and amplitude changes of these three RF stations because they affect the LPS of an electron bunch upstream the bunch compressors. ACC45 and ACC67 are operated on-crest downstream the 2nd bunch compressor and thus the LPS is very insensitive to the phase changes of these two RF stations. The amplitudes of ACC45 and ACC67 can be used to slightly tweak the beam energy which have nevertheless negligible impact on the LPS shape.

The scanned parameters are sampled uniformly within predefined ranges, which are mainly restricted by the OTR screen size for the TDS. For each combination, the new values are written into the control system *via* the *Python* interface of DOOCS [32] and the data readout has a delay of 0.5 s. Since it takes time to collect data which belong to the same bunch train but are sent out from different sources, the actual data collection speed is about 1 Hz.

3 METHODS

3.1 Data Wrangling

The data quality is essential to the performance of a neural network model. First of all, the data quality in this study is controlled during data taking. A single data point is recorded for each randomly selected machine set point. It prevents data with the same machine set point from appearing in both the train and test data, which leads to overfitting, as much as possible. Secondly, the data are further normalized and cleaned. The original size of the 12-bit camera image is 1360 × 1024 pixels. After background subtraction, the pixels are normalized by 4095. All the pixel values below 0.01 are set to 0 in order to remove negative pixel values and suppress background noise. Although the model is expected to learn and predict the position of the LPS on the screen [18], the horizontal position of the beam (along the streaking direction) depends on a lot of machine parameters, including the phase and amplitude of the TDS. We noticed that the beam sometimes moved significantly horizontally even when the machine set point remained unchanged. As a result, it is not possible to train a performant model even when only a single control knob (e.g., the phase of ACC23) is scanned. Since the horizontal position of the beam only provides the timing information, which can actually be measured using the bunch arrival monitors [33] installed at different locations of the machine, all the LPS images are cropped to 768 × 1024 and centered horizontally. Moreover, an image will be removed from the dataset if the electron bunch is completely or partially off the screen, or if the electron bunch is very close to the left or right edge.

3.2 Modeling

The detailed structure of the encoder-decoder neural network is shown in **Figure 3**. A MLP is used to build latent features from the input parameters. Two different decoders then translate the latent features to the LPS image and CTR spectrum (the raw signal measured by the CRISP spectrometer), respectively. Latent features are designed to be a compressed representation of the LPS, just as it can be interpreted by decoders for different longitudinal property diagnostics. It could also facilitate building a larger model efficiently by sharing decoders among multiple encoders which represent different machine working points [18]. The neural network model is implemented and trained using the machine learning framework TensorFlow [34] version 2.4.3. For training, we adopt the weight initialization in [35] and the Adam optimizer [36]. 80% of the data are used for training with a minibatch size of 32, and the rest are used for testing.

The loss function L_{total} for training is given by

$$L_{total} = L_{LPS} + wL_{spectrum}, \tag{1}$$

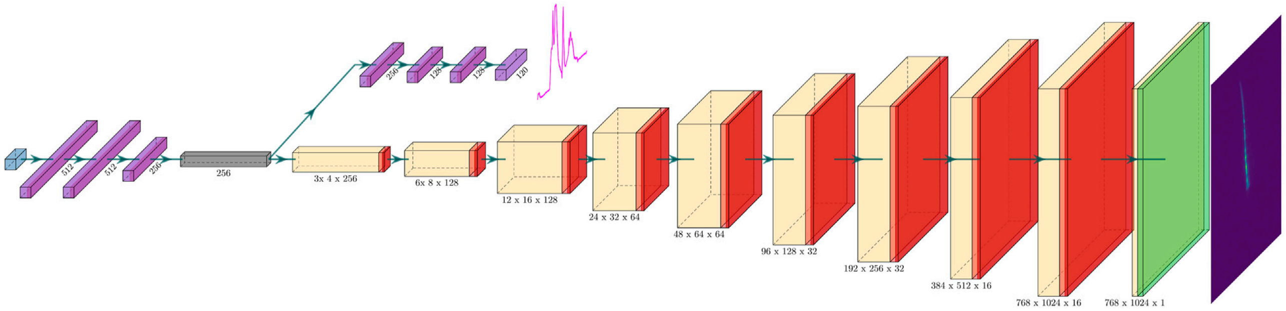


FIGURE 3 | Diagram of the encoder-decoder neural network. The leftmost blue box represents the input layer. It is followed by three fully-connected layers (encoder) in purple with each layer activated by the Leaky ReLU (Rectified Linear Unit) function. The latent space is depicted in grey. The latent space leads to two decoder branches. One consists of several fully-connected layers with each layer activated by the Leaky ReLU function. The other consists of ten transposed convolutional layers in yellow. Each transposed convolutional layer is followed by a batch normalization layer and activated by the leaky ReLU function except the last one, which is activated by the sigmoid function depicted in green. The kernel sizes of the first and second transposed convolutional layers are 3 × 4 and 3 × 3, respectively, and the kernel sizes of the other eight transposed convolutional layers are all 5 × 5. The total number of trainable parameters is 2,096,681 with 6 input parameters.

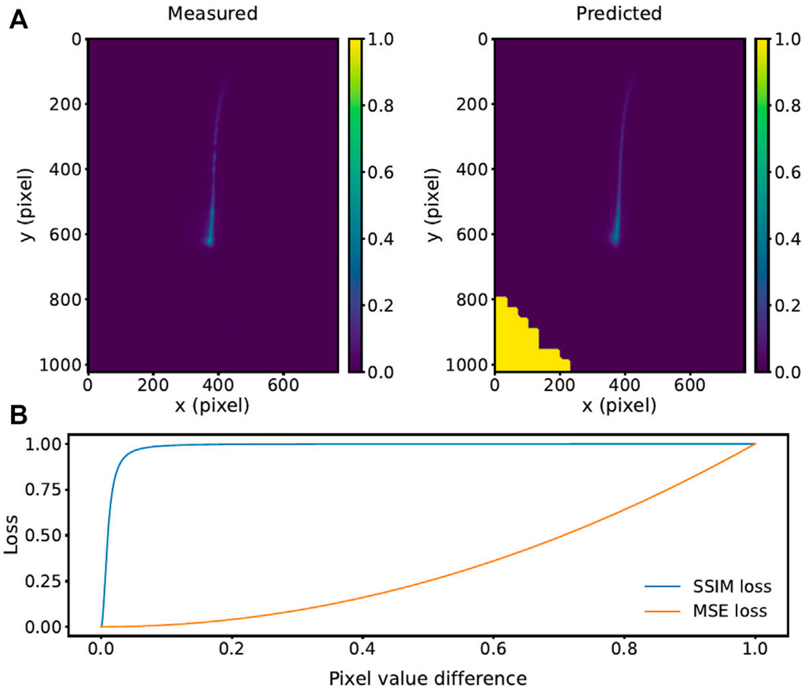


FIGURE 4 | (A) Example of a prediction from a model trained using only the multiscale SSIM loss for the LPS image decoder. The pixel values near the lower-left corner of the predicted image are trapped in values around 1. This phenomenon occurs in all the LPS images. **(B)** SSIM and MSE losses calculated between two unary images as a function of the pixel value difference. The pixel value of the first image is 0 while the pixel value of the second image ranges from 0 to 1.

where L_{LPS} and $L_{spectrum}$ are the loss functions for the LPS image decoder and the spectrum decoder, respectively, and w is the weight which balances the influences of the two decoders. The model is trained with a learning rate of 3×10^{-4} for 400 epochs, and then a learning rate of 1×10^{-4} for another 400 epochs. During the first 400 epochs, both L_{LPS} and $L_{spectrum}$ are the mean squared error (MSE) and w is set to 1. During the second 400 epochs, L_{LPS} is changed to the multiscale structural similarity index measure (SSIM) [37] with hyperparameters defined in [18] and w is set to 100 empirically in order to make the losses of the two decoders at the same order of magnitude at the end of the training. There are two advantages for using different loss functions for the LPS image decoder in different training phases. First, the multiscale SSIM loss is much more computationally expensive than the MSE loss. In our implementation, the training time for a single batch reduces

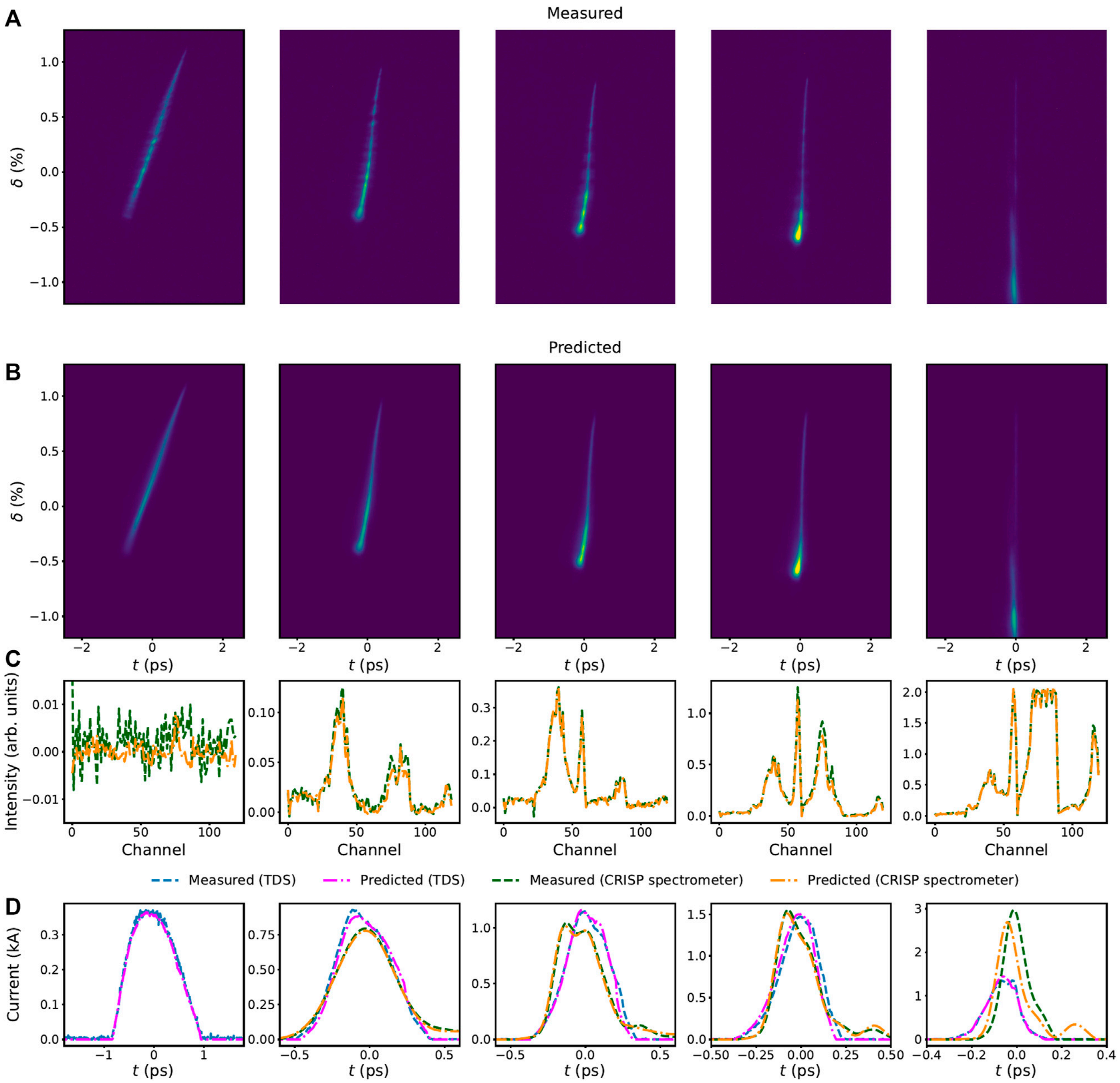


FIGURE 5 | Five typical results from the test data of a dataset of WP1. The compression strength increases monotonically from left to right. **(A)** Measured LPS images. δ denotes the fractional energy deviation. **(B)** Predicted LPS images. **(C)** Comparisons between the measured and predicted spectra from the low-frequency grating set of the CRISP spectrometer. **(D)** Comparisons of the current profiles calculated from the measured and predicted LPS images as well as reconstructed from the measured and predicted spectra.

by about 30% when the MSE loss is used. Therefore, the multiscale SSIM loss is only used to fine-tune the model in order to learn high-frequency features in the LPS images [18]. Second, it is found that sometimes the model does not learn correctly near a corner or an edge for all the images when only the multiscale SSIM loss is employed, as shown in **Figure 4A**. This can be explained by the gradients of both the loss functions, as shown in **Figure 4B**. The pixel values near the lower-left corner of the predicted LPS image shown in **Figure 4A** result in an extremely small gradient of the SSIM loss on the plateau, which prevents them from converging towards 0. However, how those pixels are trapped on the plateau is not clear.

The performance of our model is reported separately for the LPS image decoder and the spectrum decoder using the single-scale SSIM and MSE as metrics, respectively, over the test dataset. The single-scale SSIM value ranges between 0 and 1 with a value of 1 indicating two images are exactly the same.

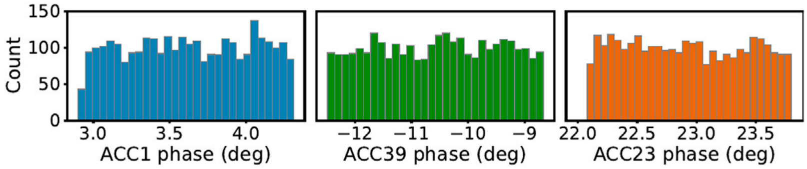


FIGURE 6 | Histograms of the scanned parameters for the dataset shown in **Figure 5**. The total number of data points is about 3000.

4 RESULTS

We record multiple datasets during two different beam times with different machine setups (working points). The beam energy is ~1124 MeV for the first working point (WP1) and ~1027 MeV for the second one (WP2). The bunch charge is ~400 pC for both the working points. As mentioned previously, the beam parameters are not optimized for the LPS measurement due to the parallel FLASH2 user experiment. Consequently, it is found that the LPSs depend on the streaking direction of the TDS in a non-trivial way for WP2 due to transverse-longitudinal correlations [6, 9], which makes it difficult to compare the current profiles measured by the TDS and the CRISP spectrometer. Nonetheless, this does not affect the conclusion of this study because the goal is to achieve an excellent agreement between the prediction and the measurement. Actually, the agreement between the current profiles measured by the TDS and the CRISP spectrometer can be verified using the dataset of WP1.

4.1 Prediction

Five typical prediction results from a dataset of WP1 are shown in **Figure 5**. The electron bunches have significantly different peak currents. The histograms of the scanned parameters are shown in **Figure 6**. The performance of the LPS image decoder is 0.9877 ± 0.00227 and the performance of the spectrum decoder is 7.9, ×, 10^{-5} ± 4.6 × 10^{-4}. As shown in **Figures 5A,B**, the predicted LPS images all agree with the measured ones excellently. There are also excellent agreements between the measured and predicted spectra, as shown in **Figure 5C**.

The current profiles calculated from the measured and predicted LPS images as well as reconstructed from the measured and predicted spectra using the combination of analytical (Kramers-Kronig) and iterative phase retrieval methods [9] are shown in **Figure 5D**. The spectra in the left-most case (weakly compressed) contain only noise. Therefore, no current profiles can be reconstructed from them. In the right-most case, due to considerable energy loss induced by the CSR effect, there are unknown portion of electrons off the screen. Therefore, the peak current calculated from the LPS image is much smaller than that reconstructed from the spectrum. For the other three cases, the current profiles calculated from the LPS images and reconstructed from the spectra agree reasonably well. Due to the lack of phase information in such spectral measurements [9], minor deviations in measured and predicted spectrum can lead to noticeable differences of the reconstructed current profiles. Nonetheless, key features (e.g., peak current) are barely affected by this.

It is noticed that there are conspicuous density modulations in the measured LPSs while the predicted LPSs are rather smooth. The density modulation is indeed induced by the microbunching instability which is seeded by the shot noise of an electron bunch [11, 38–40]: an initial small density modulation inside the electron bunch can result in sufficient energy modulation due to the longitudinal space charge force, which in turn causes larger density modulation in a magnetic chicane bunch compressor. Because of the poor longitudinal resolution of the TDS measurement, only very weak density modulation can be observed in the current profile calculated from the measured LPS of the longest bunch in **Figure 5**. The reconstructed current profiles from the spectra also do not show any evidence of the density modulation due to the lack of high-frequency components in the spectrum. Because such density modulation is not deterministic, it cannot be predicted by the neural network model using only RF phases and amplitudes as input. However, it is worth mentioning that the non-deterministic density modulation does not degrade the performance of the model. It appears that the model tends to predict the averaged LPS in which the density modulation is smoothed out.

In order to demonstrate the scalability of the model, six parameters are scanned for a dataset of WP2. The histograms of the scanned parameters are shown in **Figure 7**. In total, more than 9,000 data points are taken. However, only about 5200 data points survive the data wrangling. Nonetheless, the performance of the LPS image decoder is 0.9801 ± 0.00978 and the performance of the spectrum decoder is 3.9, ×, 10^{-4} ± 3.2 × 10^{-3}. An example prediction is shown in **Figure 8**.

4.2 Mixed Diagnostics

As mentioned previously, it is essential to combine the spectra from both grating sets of the CRISP spectrometer in order to achieve an accurate reconstruction of the current profile. At European XFEL, the high beam energy and short bunch length make it possible to extrapolate the result measured by the high-frequency grating set to the low-frequency regime [41]. However, this is not feasible for the typical electron bunches at FLASH, especially when the electron bunches have a broad range of bunch lengths. Although another decoder could be trained to predict the spectrum measured by the high-frequency grating set, there are two concerns with this approach. Firstly, the spectra from the high-frequency grating set generally contain microbunching information, which fluctuates from bunch to bunch and train to train.

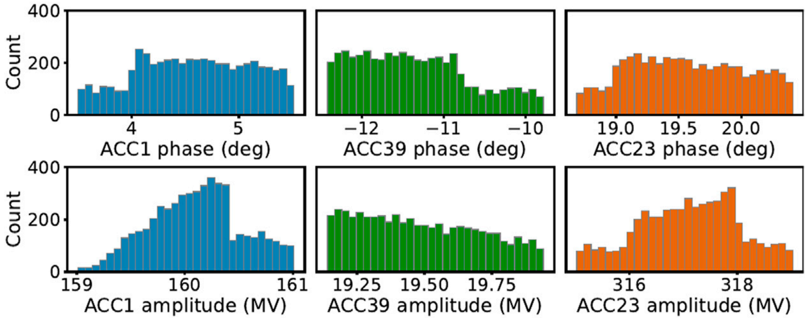


FIGURE 7 | Histograms of the scanned parameters for a dataset of WP2. The total number of data points is about 5200. The input data are not uniformly distributed because of two reasons. First, the data are combined from two datasets taken successively. The parameter ranges of the second dataset are smaller than the first one. Second, more than 40% of the data are dropped, as discussed in **Section 3.1**.

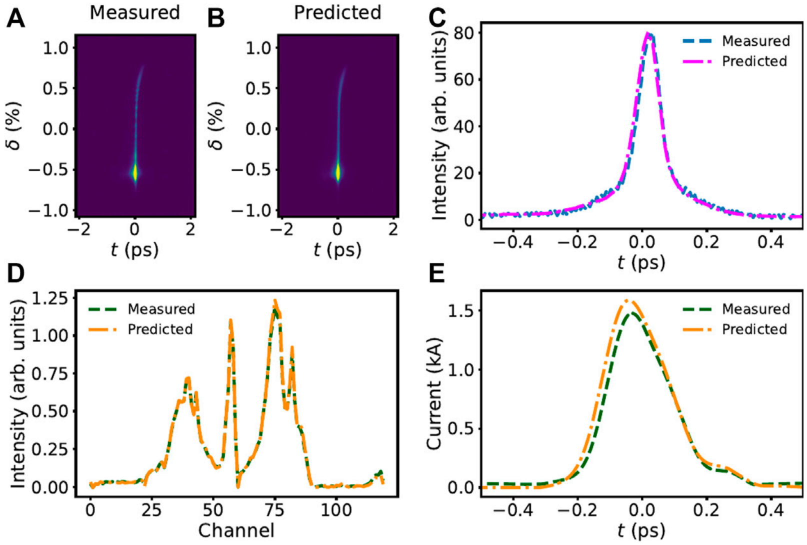


FIGURE 8 | An example result for the scan shown in **Figure 7**. **(A)** Measured LPS image. **(B)** Predicted LPS image. **(C)** Comparison of the current profiles calculated from the measured and predicted LPS images. Because the LPS depends on the streaking direction of the TDS in a non-trivial way due to transverse-longitudinal correlations, the current profile here is simply the projection of the LPS image in the vertical direction. **(D)** Comparison of the measured and predicted spectra from the low-frequency grating set of the CRISP spectrometer. **(E)** Comparison of the current profiles reconstructed from the measured and predicted spectra.

Secondly, this will double the data collection time, which could be unaffordable for user facilities with very limited beam time for non-user experiments.

In order to improve the reconstructed current profile for real-time measurement, a mixed diagnostics method is proposed. The predicted spectrum for the low-frequency grating set can be combined with the measured spectrum from the high-frequency grating set so as to achieve the full-range spectrum in real time. To this end, we recorded two datasets using the high-frequency and low-frequency grating sets, respectively. Each dataset contains about 3000 data points. Only three control knobs (the phases of ACC1, ACC39 and ACC23) are scanned within the same ranges. A neural network model is trained using the dataset recorded with the low-frequency grating set. The performance of the LPS image decoder is 0.9854 ± 0.00394 and the performance of the spectrum decoder is $1.8 \times 10^{-4} \pm 1.3 \times 10^{-3}$. The trained model is also tested against all the LPS images in the dataset recorded with the high-frequency grating set and the performance of the LPS image decoder is as high as 0.9814 ± 0.00525. It indicates that the machine drift during data recording is negligible. Three typical prediction results with significantly different peak currents are shown in **Figure 9**. They all show that it is essential to have the high-frequency components to reveal the exact shape and structures of the current profile although the low-frequency components are good enough to estimate the bunch length and the peak current.

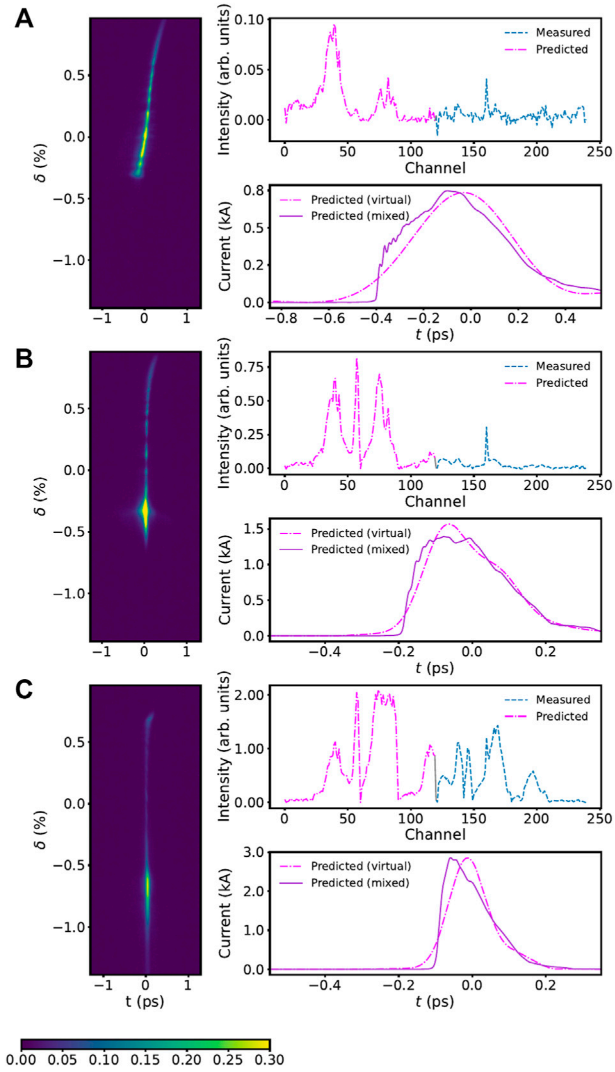


FIGURE 9 | Three typical results for mixed diagnostics with the CRISP spectrometer. The measured LPS image and the mixed spectrum are shown for each result. The current profiles reconstructed from the mixed spectra are compared with the current profiles reconstructed from only the predicted (virtual) spectra.

5 CONCLUSION

In summary, we have experimentally demonstrated highly accurate megapixel LPS images and CTR spectra predictions for electron bunches in a bunch train concurrently at the end of the FLASH linac. Up to six major control knobs for LPS manipulation are scanned in order to collect data for electron bunches with broad ranges of LPS shapes and peak currents. The model is capable of providing heterogeneous LPS information and ensures reliable online diagnostics because a single type of diagnostic cannot cover electron bunches with a broad range of parameters. LPS images measured in the time domain are essential for electron bunches longer than a few hundreds of fs while CTR spectra provide more accurate current profiles for strongly compressed electron bunches.

A mixed diagnostics method is proposed to significantly improve the online current profile measurement using the CRISP spectrometer. The predicted CTR spectrum for the low-frequency grating set can be combined with the spectrum measured by the high-frequency grating set, which enables reconstruction of the current profile with a much higher accuracy in real time. Although the current profile reconstructed from the spectrum measured by the low-frequency grating set is good enough to estimate the bunch lengths and peak currents for typical electron bunches at FLASH, the spectrum measured by the high-frequency grating set is indispensable to reveal the exact shape and structures of the current profile.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

JZ, MC, and HS designed the study; MC coordinated the experiments at FLASH; JZ, MC, and NL performed the measurements. JZ designed the neural network model, wrote the manuscript as well as the code for data collection and analysis; NL reconstructed the current profiles from the CTR spectra and provided critical insight for interpreting the data.

ACKNOWLEDGMENTS

We would like to thank V. Ayvazyan, J. Branlard, and S. Pfeiffer for advices on RF phase and amplitude scan, M. Hoffmann, J. Kral, J. Roensch-Schulenburg, C. Schmidt, and M. Vogt for assistance as well as useful discussions during the data collection campaigns. We are grateful to the FLASH team for support and to B. Steffen for his thoughtful comments.

REFERENCES

1. Decking W, Abeghyan S, Abramian P, Abramsky A, Aguirre A, Albrecht C, et al. A Mhz-Repetition-Rate Hard X-ray Free-Electron Laser Driven by a Superconducting Linear Accelerator. *Nat Photon* (2020) 14:391. doi:10.1038/s41566-020-0607-z
2. Qi F, Ma Z, Zhao L, Cheng Y, Jiang W, Lu C, et al. Breaking 50 Femtosecond Resolution Barrier in Mev Ultrafast Electron Diffraction with a Double bend Achromat Compressor. *Phys Rev Lett* (2020) 124:134803. doi:10.1103/PhysRevLett.124.134803
3. Jalas S, Kirchen M, Messner P, Winkler P, Hübner L, Dirkwinkel J, et al. Bayesian Optimization of a Laser-Plasma Accelerator. *Phys Rev Lett* (2021) 126:104801. doi:10.1103/PhysRevLett.126.104801

4. Lindstrøm CA, Garland JM, Schröder S, Boulton L, Boyle G, Chappell J, et al. Energy-spread Preservation and High Efficiency in a Plasma-wakefield Accelerator. *Phys Rev Lett* (2021) 126:014801. doi:10.1103/PhysRevLett.126.014801
5. Tang H, Zhao L, Zhu P, Zou X, Qi J, Cheng Y, et al. Stable and Scalable Multistage Terahertz-Driven Particle Accelerator. *Phys Rev Lett* (2021) 127: 074801. doi:10.1103/PhysRevLett.127.074801
6. Emma P, Frisch J, Krejcik P. *A Transverse Rf Deflecting Structure for bunch Length and Phase Space Diagnostics* (2000).
7. Akre R, Bentson L, Emma P, Krejcik P. A Transverse Rf Deflecting Structure for bunch Length and Phase Space Diagnostics. *Pacs2001 Proc 2001 Part Accelerator Conf (Cat. No.01CH37268)* (2001) 33:2353–5. doi:10.1109/PAC.2001.987379
8. Wesch S, Schmidt B, Behrens C, Delsim-Hashemi H, Schmüser P. A Multi-Channel Thz and Infrared Spectrometer for Femtosecond Electron bunch Diagnostics by Single-Shot Spectroscopy of Coherent Radiation. *Nucl Instr Methods Phys Res Section A: Acc Spectrometers, Detectors Associated Equipment* (2011) 665:40–7. doi:10.1016/j.nima.2011.11.037
9. Schmidt B, Lockmann NM, Schmüser P, Wesch S. Benchmarking Coherent Radiation Spectroscopy as a Tool for High-Resolution bunch Shape Reconstruction at Free-Electron Lasers. *Phys Rev Accel Beams* (2020) 23:062801. doi:10.1103/PhysRevAccelBeams.23.062801
10. Lockmann NM, Gerth C, Schmidt B, Wesch S. Noninvasive Thz Spectroscopy for bunch Current Profile Reconstructions at Mhz Repetition Rates. *Phys Rev Accel Beams* (2020) 23:112801. doi:10.1103/PhysRevAccelBeams.23.112801
11. Qiang J, Ding Y, Emma P, Huang Z, Ratner D, Raubenheimer TO, et al. Start-to-end Simulation of the Shot-Noise Driven Microbunching Instability experiment at the Linac Coherent Light Source. *Phys Rev Accel Beams* (2017) 20:054402. doi:10.1103/PhysRevAccelBeams.20.054402
12. Anirudh R, Thiagarajan JJ, Bremer P-T, Spears BK. Improved Surrogates in Inertial Confinement Fusion with Manifold and Cycle Consistencies. *Proc Natl Acad Sci U.S.A* (2020) 117:9741–6. doi:10.1073/pnas.1916634117
13. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly Accurate Protein Structure Prediction with Alphafold. *Nature* (2021) 596:583–9. doi:10.1038/s41586-021-03819-2
14. Emma C, Edelen A, Hogan MJ, O'Shea B, White G, Yakimenko V. Machine Learning-Based Longitudinal Phase Space Prediction of Particle Accelerators. *Phys Rev Accel Beams* (2018) 21:112802. doi:10.1103/PhysRevAccelBeams.21.112802
15. Edelen A, Neveu N, Emma C, Ratner D, Mayes C. *Machine Learning Models for Optimization and Control of X-ray Free Electron Lasers*. Vancouver: NeurIPS Machine Learning for the Physical Sciences Workshop (2019).
16. Emma C, Edelen A, Hanuka A, O'Shea B, Scheinker A. Virtual Diagnostic Suite for Electron Beam Prediction and Control at Facet-Ii. *Information* (2021) 12: 61. doi:10.3390/info12020061
17. Hanuka A, Emma C, Maxwell T, Fisher AS, Jacobson B, Hogan MJ, et al. Accurate and Confident Prediction of Electron Beam Longitudinal Properties Using Spectral Virtual Diagnostics. *Sci Rep* (2021) 11:2945. doi:10.1038/s41598-021-82473-0
18. Zhu J, Chen Y, Brinker F, Decking W, Tomin S, Schlarb H. High-fidelity Prediction of Megapixel Longitudinal Phase-Space Images of Electron Beams Using Encoder-Decoder Neural Networks. *Phys Rev Appl* (2021) 16:024005. doi:10.1103/PhysRevApplied.16.024005
19. Scheinker A, Edelen A, Bohler D, Emma C, Lutman A. 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* (2018) 121:044801. doi:10.1103/PhysRevLett.121.044801
20. Leemann SC, Liu S, Hexemer A, Marcus MA, Melton CN, Nishimura H, et al. Demonstration of Machine Learning-Based Model-independent Stabilization of Source Properties in Synchrotron Light Sources. *Phys Rev Lett* (2019) 123: 194801. doi:10.1103/PhysRevLett.123.194801
21. Duris J, Kennedy D, Hanuka A, Shtalenkova J, Edelen A, Baxevanis P, et al. Bayesian Optimization of a Free-Electron Laser. *Phys Rev Lett* (2020) 124: 124801. doi:10.1103/PhysRevLett.124.124801
22. Bruchon N, Fenu G, Gaio G, Lonza M, O'Shea FH, Pellegrino FA, et al. Basic Reinforcement Learning Techniques to Control the Intensity of a Seeded Free-Electron Laser. *Electronics* (2020) 9:781. doi:10.3390/electronics9050781
23. Kain V, Hirlander S, Goddard B, Velotti FM, Della Porta GZ, Bruchon N, et al. Sample-efficient Reinforcement Learning for Cern Accelerator Control. *Phys Rev Accel Beams* (2020) 23:124801. doi:10.1103/PhysRevAccelBeams.23.124801
24. Scheinker A, Cropp F, Paiagua S, Filippetto D. An Adaptive Approach to Machine Learning for Compact Particle Accelerators. *Sci Rep* (2021) 11:19187. doi:10.1038/s41598-021-98785-0
25. Edelen A, Neveu N, Frey M, Huber Y, Mayes C, Adelmann A. Machine Learning for Orders of Magnitude Speedup in Multiobjective Optimization of Particle Accelerator Systems. *Phys Rev Accel Beams* (2020) 23:044601. doi:10.1103/PhysRevAccelBeams.23.044601
26. Hanuka A, Huang X, Shtalenkova J, Kennedy D, Edelen A, Zhang Z, et al. Physics Model-Informed Gaussian Process for Online Optimization of Particle Accelerators. *Phys Rev Accel Beams* (2021) 24:072802. doi:10.1103/PhysRevAccelBeams.24.072802
27. Roussel R, Hanuka A, Edelen A. Multiobjective Bayesian Optimization for Online Accelerator Tuning. *Phys Rev Accel Beams* (2021) 24:062801. doi:10.1103/PhysRevAccelBeams.24.062801
28. Ackermann W, Asova G, Ayvazyan V, Azima A, Baboi N, Bähr J, et al. Operation of a Free-Electron Laser from the Extreme Ultraviolet to the Water Window. *Nat Photon* (2007) 1:336. doi:10.1038/nphoton.2007.76
29. Faatz B, Plönjes E, Ackermann S, Agababyan A, Asgekar V, Ayvazyan V, et al. Simultaneous Operation of Two Soft X-ray Free-Electron Lasers Driven by One Linear Accelerator. *New J Phys* (2016) 18:062002. doi:10.1088/1367-2630/18/6/062002
30. Röhrs M, Gerth C, Schlarb H, Schmidt B, Schmüser P. Time-resolved Electron Beam Phase Space Tomography at a Soft X-ray Free-Electron Laser. *Phys Rev ST Accel Beams* (2009) 12:050704. doi:10.1103/PhysRevSTAB.12.050704
31. Behrens C, Gerasimova N, Gerth C, Schmidt B, Schneidmiller EA, Serkez S, et al. Constraints on Photon Pulse Duration from Longitudinal Electron Beam Diagnostics at a Soft X-ray Free-Electron Laser. *Phys Rev ST Accel Beams* (2012) 15:030707. doi:10.1103/PhysRevSTAB.15.030707
32. Hensler GG, Goloborodko S, Grygiel G, Hensler O, Kocharyan V, Rehlich K, et al. *Doocs: An Object Oriented Control System as the Integrating Part for the Ttf Linac* (1998).
33. Viti M, Czwalinna M, Dinter H, Gerth C, Przygoda K, Rybaniec R, et al. The bunch Arrival Time Monitor at Flash and European Xfel. *Proc ICALEPCS2017* (2017) TUPHA125.
34. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems (2017).
35. He K, Zhang X, Ren S, Sun J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification. In: *2015 IEEE International Conference on Computer Vision* (2015). p. 1026–34. doi:10.1109/ICCV.2015.123
36. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization (2015).
37. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans Image Process* (2004) 13:600–12. doi:10.1109/TIP.2003.819861
38. Huang Z, Kim K-J. Formulas for Coherent Synchrotron Radiation Microbunching in a bunch Compressor Chicane. *Phys Rev ST Accel Beams* (2002) 5:074401. doi:10.1103/PhysRevSTAB.5.074401
39. Saldin EL, Schneidmiller EA, Yurkov MV. An Analytical Description of Longitudinal Phase Space Distortions in Magnetic bunch Compressors. *Nucl Instr Methods Phys Res Section A: Acc Spectrometers, Detectors*

Associated Equipment (2002) 483:516–20. doi:10.1016/S0168-9002(02)00372-8

40. Ratner D, Behrens C, Ding Y, Huang Z, Marinelli A, Maxwell T, et al. Time-resolved Imaging of the Microbunching Instability and Energy Spread at the Linac Coherent Light Source. *Phys Rev ST Accel Beams* (2015) 18:030704. doi:10.1103/PhysRevSTAB.18.030704
41. Lockmann NM. *Noninvasive Measurements of Electron Bunch Current Profiles with Few-Femtosecond Resolution at MHz Repetition Rates*. Universität Hamburg: Ph.D. thesis (2021).

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Physics

TYPE Technology and Code
PUBLISHED 07 September 2022
DOI 10.3389/fphy.2022.929064

Application of reinforcement learning in the LHC tune feedback

Leander Grech[1]*, Gianluca Valentino[1], Diogo Alves[2] and Simon Hirlaender[3]

[1]Department of Communication and Computer Engineering, University of Malta, Msida, Malta, [2]CERN, Geneva, Switzerland, [3]Department of Artificial Intelligence and Human Interfaces, University of Salzburg, Salzburg, Austria

OPEN ACCESS

EDITED BY
Robert Garnett,
Los Alamos National Laboratory (DOE),
United States

REVIEWED BY
Christine Darve,
European Spallation Source, Sweden
Baoxi Han,
Oak Ridge National Laboratory (DOE),
United States

*CORRESPONDENCE
Leander Grech,
leander.grech.14@um.edu.mt

SPECIALTY SECTION
This article was submitted to High-Energy and Astroparticle Physics, a section of the journal Frontiers in Physics

RECEIVED 26 April 2022
ACCEPTED 25 July 2022
PUBLISHED 07 September 2022

CITATION
Grech L, Valentino G, Alves D and Hirlaender S (2022), Application of reinforcement learning in the LHC tune feedback.
Front. Phys. 10:929064.
doi: 10.3389/fphy.2022.929064

The Beam-Based Feedback System (BBFS) was primarily responsible for correcting the beam energy, orbit and tune in the CERN Large Hadron Collider (LHC). A major code renovation of the BBFS was planned and carried out during the LHC Long Shutdown 2 (LS2). This work consists of an explorative study to solve a beam-based control problem, the tune feedback (QFB), utilising state-of-the-art Reinforcement Learning (RL). A simulation environment was created to mimic the operation of the QFB. A series of RL agents were trained, and the best-performing agents were then subjected to a set of well-designed tests. The original feedback controller used in the QFB was reimplemented to compare the performance of the classical approach to the performance of selected RL agents in the test scenarios. Results from the simulated environment show that the RL agent performance can exceed the controller-based paradigm.

KEYWORDS

LHC, beam-based controller, tune feedback, reinforcement learning, cern

1 Introduction

The LHC is the largest synchrotron built to date and its sheer scale meant that it was the first particle accelerator of its type to require automatic beam-based feedback systems to control key beam parameters [1]. The Beam-Based Feedback System (BBFS) implemented these feedback systems and was developed prior to the LHC start-up in 2008. Throughout the years, operator experience has dictated which functionality to keep, add and remove from the BBFS [2].

The BBFS comprised several subsystems, each responsible for controlling a specific beam parameter or machine parameter. One of the most critical parameters to control is the tune (Q). Q is defined as the number of transverse oscillations a particle performs in one revolution around the LHC. Ideally the value of the tune is an irrational number so that the location of the transverse oscillations do not occur in the same longitudinal locations in the LHC. The Tune Feedback (QFB) system was the BBFS subsystem responsible for controlling the tune.

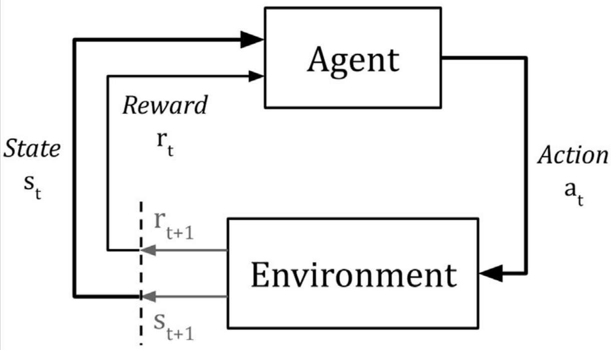


FIGURE 1
Top-level view of the interaction between the Agent and the Environment.

The QFB requires a constantly updated estimate of the value of the tunes in the horizontal and vertical planes of both beams in the LHC. The estimation of the tune was performed by the Base-Band Tune (BBQ) system [3]. Each plane in every beam was handled independently through a set of tuning quadrupoles of type MQT, to adjust the magnetic beam envelope. The QFB operated on the assumption that the effect of a changing quadrupolar magnetic field on the tune of the beam can be modelled by linear beam optics. These optics came in the form of a 2D matrix and were obtained by beam optics design programs such as Methodical Accelerator Design (MAD-X) developed by CERN [4].

The QFB on one beam relies on a Tune Response Matrix (Q_{RM}), which is a 16 × 2 matrix modelling the change in the vertical and horizontal tunes due to a change in the deflections of 16 (de)focusing tuning quadrupoles of type MQT. Therefore, a vector containing the delta quadrupole deflections, $\Delta\overrightarrow{\delta_Q}$ is multiplied by the Q_{RM}:

$$\Delta\overrightarrow{\delta_Q} \cdot Q_{RM} \quad = \quad \Delta\vec{Q}$$

where $\Delta\vec{Q}$ is the modelled change in tune due to $\Delta\overrightarrow{\delta_Q}$ being applied to the quadrupoles. Singular Value Decomposition (SVD) is used to calculate the pseudo-inverse of the Q_{RM} (Q_{PI}). A Proportional-Integral (PI) controller then uses Q_{PI} to calculate the corrections, $\Delta\overrightarrow{\delta_Q}$, for an optimal response:

$$\Delta\vec{Q} \cdot (Q_{PI})^\top \quad = \quad \Delta\overrightarrow{\delta_Q}$$

From the first operation of the QFB, it was observed that erroneous tune estimates from the BBQ system were causing unstable behaviour. To avoid an indeterministic response, a stability metric is used within the QFB to switch off the feedback controller in the presence of excess instability in the tune estimates.

This work is an explorative study on the application of Reinforcement Learning (RL) in the QFB. Since this work was carried out during LHC LS2, a simulation environment called QFBEnv was developed to mimic the operation of the QFB in the LHC. Several tests were designed to probe the robustness of the trained agents and the PI controller to external noise and non-stochastic environments. The results from these tests were used to evaluate the performance of the best trained RL agents.

This paper is organised as follows: Section 2 provides a mathematical formalism of RL and an overview of the RL algorithms used in this work. Section 3 describes the design of the RL environment, which mimics the QFB in operation in the LHC. Section 4 describes the training of each algorithm. Finally, Section 5 describes the evaluation of best-trained agents and compares their performance with the standard PI control used in the QFB.

2 Reinforcement Learning

Figure 1 shows a top-level view of the essential components and their interactions within an RL framework. The various signals between the agent and environment are labelled in notation used in RL nomenclature. s_t and a_t refer to the state of the environment and the action chosen at time t. r_t refers to the reward being given to the agent for the action taken at time $t - 1$, hence r_{t+1} is defined as the reward given to the agent for choosing action a_t when in state s_t.

The RL problem can be modelled by a finite Markov Decision Process (MDP) which is defined by a (s_t, a_t, r_t, s_{t+1}) tuple. A set of tuples of size H constitute an episode, where the initial state is drawn from the initial state distribution, $s_0 \sim p_0$. The reward is a scalar representing the *goodness* of the last action performed on the environment. The ultimate goal of any RL algorithm is to maximise the expected discounted cumulative future reward, or return (G), obtained by the agent:

$$G_t \quad \triangleq \quad \mathbb{E}_{a\sim\pi_\theta, s\sim p(s_{t+1}|s_t,a_t)} \sum_{i=t}^{H-t} \gamma^i r(s_i, a_i)$$

where π_θ is known as the policy, which maps the states to the actions and is parameterised by vector θ. p $(s_{t+1}|s_t, a_t)$ is the transition distribution of the environment. γ is called the discount factor ($\gamma \le 1$, e.g., 0.99) and its role is to control the importance of future rewards in the calculation of the value for a particular state. RL nomenclature defines the value function as a measure of the total expected future rewards the agent can expect when starting in some state, s [5].

Figure 2 shows a non-exhaustive taxonomy of the various RL algorithms found in the literature. RL can be split into two main classes: Model-Free (MF) and Model-Based (MB) algorithms. As the name implies, MF RL is the study of algorithms that do not require a model to be learned and solely depend on the relationship among the actions, states and rewards obtained on every interaction. On the other hand, MB RL is the study of algorithms that either has access to the full dynamics model, e.g., AlphaGo [6], or require that a model of the environment is

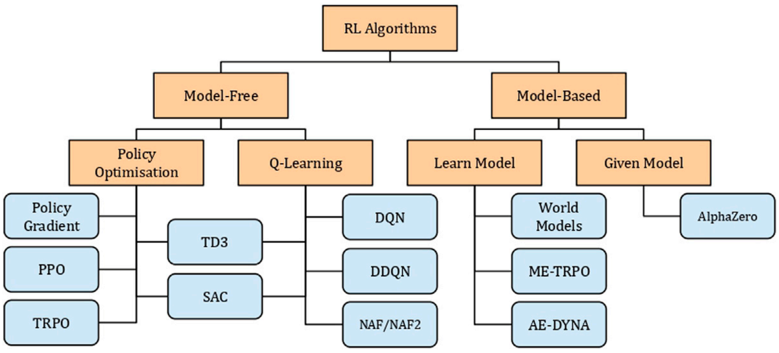


FIGURE 2
A non-exhaustive taxonomy of RL algorithms.

learned alongside the agent, e.g., Model-Ensemble Trust-Region Policy Optimization (ME-TRPO) [7].

The first RL algorithm considered in this work is Normalized Advantage Functions (NAF), which was introduced in [8] and is based on Deep Q Networks (DQN) and Duelling DQN (DDQN) to solve continuous tasks, e.g., robotic arm control. These types of algorithms update a policy that did not necessarily create the trajectory; hence they are called *off-policy* methods.

Unlike DQN, NAF contains a network with three output streams: 1) Value function estimate, $\hat{V}(s)$; 2) Advantage estimate $\hat{A}(s,a)$; 3) Policy $\mu(s)$, where:

$$Q(s,a;\theta) \quad \triangleq \quad V(s;\theta) \; + \; A(s,a;\theta) \tag{1}$$

By taking the argmax over the actions in the Advantage, the agent learns the optimal Q-value. NAF differs from standard Q-learning methods since A is explicitly parameterised as a quadratic function of non-linear features of the state:

$$\begin{aligned} A(s,a;\theta) \; &= \; -\frac{1}{2}(a-\mu(s;\theta))^{\top} P(s;\theta)(a-\mu(s;\theta)) \\ P(s,\theta) \; &\triangleq L(s;\theta)L(s;\theta)^{\top} \end{aligned} \tag{2}$$

where L is a lower-triangular matrix with an exponentiated diagonal, constructed from the second output stream of the network. By Eq. 2, p is a state-dependent, positive-definite square matrix. The third output stream of the network is $\mu(s;\theta)$, which is the action that maximises the Q-function in Eq. 1 when the following loss is minimised:

$$\mathcal{L} \; = \; \frac{1}{N}\sum_i (y_i - Q(s_i,a_i;\theta))^2$$

with:

$$y_i \; = \; r_i + \gamma V'(s_{i+1};\theta')$$

$V'(\cdot;\theta')$ denotes the target network which is a separate network, updated slower than the main network, $V(\cdot;\theta)$ by using Polyak averaging on their parameters:

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta'$$

where τ is set to a small number, e.g., 0.005. Hirlaender et al. introduced NAF2 in [9], which adds clipped smoothing noise to the actions, a technique also used in the Twin-Delayed Deep Deterministic Policy Gradient (TD3) algorithm to stabilise the policy training [10].

The second algorithm is Proximal Policy Optimization (PPO) introduced in [11], which is primarily based on the Policy Gradient (PG) method. PG methods use trajectory rollouts created by taking actions from the most recent policy trained by the agent on the environment; hence they are called *on-policy* methods. The most common form of the PG objective is written as:

$$\mathcal{J}^{PG}(\theta) \; = \; \hat{\mathbb{E}}_{t,a_t\sim\pi_\theta(s_t),s_{t+1}\sim\; p(\cdot|s_t,a_t)}\left[\log\pi_\theta(a_t|s_t)\hat{A}_t\right]$$

The expectation symbol $\hat{\mathbb{E}}$ denotes the empirical average over a batch of rollouts denoted by t, actions a_t sampled from $\pi_\theta(\cdot)$, which is a stochastic policy. $\hat{A}_t$ is an estimate of the Advantage, which is a measure of how good a_t is compared to the average action possible at time t. Differentiating $\mathcal{J}^{PG}$ with respect to the network parameters, θ, obtains the PG estimate, $\hat{g}$:

$$\hat{g} \; = \; \hat{\mathbb{E}}_t\left[\nabla_\theta \log\pi_\theta(a_t|s_t)\hat{A}_t\right]$$

The gradient estimate is used in gradient ascent to maximise the objective $\mathcal{J}^{PG}$. It was also shown in [11] that performing multiple optimisation steps using the same trajectory leads to destructively large policy updates that no longer converge to an optimal policy. Trust Region Policy Optimization (TRPO) introduced a solution by constraining the maximisation of $\mathcal{J}^{PG}$ by:

$$\hat{\mathbb{E}}_t\left[D_{KL}\left[\pi_{\theta_{old}}(\cdot\,|s_t),\pi_\theta(\cdot\,|s_t)\right]\right] \; \le \; \sigma \tag{3}$$

where $\sigma \in \mathbb{R}$, e.g., 0.01, θ_{old} is the policy parameters before the policy update and θ after the update. One drawback of TRPO is its implementation complexity. PPO uses a similar approach to TRPO, however, instead of constraining the policy to satisfy Eq. 3, PPO introduces a clipped policy objective with a penalty proportional to the KL-divergence between the old and new policy:

$$r_t \triangleq \frac{\pi_\theta}{\pi_{\theta_{old}}} \tag{4}$$

$$\mathcal{J}^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right] \tag{5}$$

$\mathcal{J}^{CLIP}$ is easier to implement and executes faster than TRPO. Experiments done in [11] also show that PPO and TRPO obtain a similar performance over many types of environments[1].

The most noticeable difference between off-policy and on-policy methods mentioned so far is the two types of policies used; deterministic and stochastic, respectively. NAF2 has a deterministic policy, which means that the agent will always choose the same action for one state. For each state, the stochastic policy of PPO provides an action distribution that can be sampled for the next action[2].

More RL algorithms were attempted, and their training and evaluation can be found in the Supplementary Material. The Soft Actor-Critic (SAC) algorithm combines the use of stochastic policies and the off-policy method by introducing the concept of entropy maximisation to RL [12]. Anchored-Ensemble DYNA-style (AE-DYNA) is an MBRL algorithm that internally relies on a SAC agent to optimise a policy in an uncertainty-aware world model [9].

3 Environment

The QFB uses two Q_{RM} per beam and each Q_{RM} is set up by default to contain six outputs: the horizontal and vertical tune, the horizontal and vertical chromaticity, and the real and imaginary components of the coupling coefficient. To separate the QFB from the coupling and chromaticity control, the Q_{RM} is truncated to have only two outputs; the tunes. The tune control sequence occurs at 12.5 Hz and a predefined sequence of steps is performed where: 1) the tune error, $\Delta\vec{Q}$, is obtained by subtracting the current tune estimate and the reference tune; 2) the tune error is clipped to $\approx 56\,\text{Hz}$[1]; 3) a velocity form Proportional-Integral (PI) controller is applied by using $\Delta\overrightarrow{Q_t}$ and $\Delta\overrightarrow{Q_{t-1}}$, where t denotes the time step; 4) the PI output is multiplied by the pseudo-inverse matrix of Q_{RM} (Tune Pseudo-Inverse (Q_{PI}) computed by Singular Value Decomposition) to obtain a set of residual quadrupole currents; 5) the currents are multiplied by -1; 6) the currents are globally scaled by a factor $k \leq 1$ to accommodate the slew rate of the quadrupoles; 7) finally the corrections are sent to the quadrupole power converters *via* UDP packets.

An OpenAI Gym environment [13] was set up to mimic the response of the LHC to a varying quadrupolar magnetic field. This environment will hereon be referred to as QFBEnv. QFBEnv has two continuous states as output and uses 16 bounded continuous actions as input. The states and actions were normalised to the range $[-1, 1]$. The normalised state-space represented a range of $[-25\ \text{Hz}, 25\ \text{Hz}]$ of tune error.

QFBEnv implicitly implemented current slew rate limiting since the actions were clipped to the range [-1,1]. A saturated action in QFBEnv is the maximum change in a quadrupolar field strength that the magnets can supply in the next time step. The normalised action space represented the fraction of the total allowable current rate in the magnets. Every step was also assumed to occur every 80 ms, which corresponds to the QFB controller frequency of 12.5 Hz. Therefore a normalised action of one on a magnet with a maximum current rate of 0.5 A s^{-1} is equivalent to a current change of $0.5 \times 0.08 = 40$ mA.

In addition, the PI controller used by the QFB was also re-implemented as a particular method within QFBEnv. This was done to provide a reference for the performance of a trained agent. The proportional, K_p, and integral, K_i, gains of the PI controller were set to low values by default as a conservative measure during initialisation of the QFB. To ensure a fair comparison, the PI controller was tuned using the Ziegler-Nichols method [14]: 1) K_i was set to zero; 2) K_p was increased until state oscillations were observed; 3) the latest value of K_p was halved; 4) K_i was increased until state oscillations were observed; 5) K_i was halved. The final gains for the PI controller implemented in QFBEnv were $K_p = 1{,}000$ and $K_i = 2000$.

The PI controller was implemented with the *global* slew rate limiting. As an example, consider a quadrupole magnet, M, with a slew rate of 0.5 A s^{-1}. If a current change at 1 A s^{-1} is requested for the next time step, all of the outputs of the PI controller are scaled by a factor $k = 0.5$ to accommodate M. The global factor k can be decreased if another magnet requires $k < 0.5$ to accommodate its respective slew rate. QFBEnv does not enforce the global scaling scheme by default, therefore any action within the $[-1, 1]$ bounds are applied to the environment within the next time step.

QFBEnv implemented two important functionalities: 1) the *reset* function and; 2) the *step* function. The reset function is the entry point to a new episode, whereby a new initial state, ($\Delta Q_{Hor.,0}$, $\Delta Q_{Ver.,0}$), is generated and returned to the agent. The step function is responsible for accepting an action, following the transition dynamics of the environment and then returning a tuple containing: 1) the next state; 2) the reward and; 3) a Boolean flag to indicate episode termination. The reward was chosen to be the negative average quadratic of the state, as shown in Eq. 6.

[1] As implemented in the QFB; $0.01 \times \frac{F_s}{2}$ where the sampling frequency, F_s, of the BBQ system is equal to the LHC revolution frequency, 11 245.55 Hz.

[2] Initialised randomly and set constant throughout one episode.

TABLE 1 The RL algorithms attempted in this work along with their type of policies and whether they train a world model: Model-Free (MF) or Model-Based (MB).

Algorithm	On-policy	Off-policy	MF	MB
NAF2		✓	✓	
PPO	✓		✓	
SAC		✓	✓	
TD3		✓	✓	
AE-DYNA	✓	✓		✓

$$r_{t+1} = -\frac{1}{n}\sum_{i}^{n}\left(s_{t+1}^{(i)}\right)^2 \tag{6}$$

Since the goal of any RL agent is to maximise the reward, a perfectly trained agent would thus have a policy which reduces the tune error to zero. It is also important to note that the gradient of quadratic reward increases, the farther the state is from the optimal point. This reflects the importance of controlling a larger tune error with respect to a smaller error in the QFB, e.g., if $\Delta Q_H \gg \Delta Q_V >$ **Goal**, the agent would put more importance on the correction of ΔQ_H. From trial and error, this shape of reward was observed to produce more stable training.

An episode is defined as starting from the first state initialisation until a terminal state is reached. A baseline optimal episode length was obtained by running 1,000 episodes using the PI controller. The measured average optimal episode length was $\approx 28 \pm 6$ steps. The maximum allowable episode length in QFBEnv was chosen to be 70 steps long, approximately double the maximum optimal episode length observed. Successful early termination in QFBEnv is defined as the latest five rewards being above a threshold. The threshold value of QFBEnv was chosen as a maximum tune error in both planes of 1 Hz. This is equivalent to $\frac{1}{25} = 0.04$ in normalised state space and the threshold reward can be obtained from Eq. 6:

$$r_{thresh} = -\frac{1}{2}\left(0.04^2 + 0.04^2\right) = -0.0016$$

Thus, a terminal state could be reached either by a successful early termination or after 70 steps were made without success. The successful early termination represents a real operational scenario, since the QFB is typically switched off manually when the measured tunes are close to their respective reference values. It also allows for more examples of the state below the threshold to be experienced by the RL agents, which leads to better learning close to the threshold boundary.

4 Training

Table 1 tabulates information about the types of RL algorithms which were trained on QFBEnv. The two algorithms which obtained the best performing policies were NAF2 and PPO. The training process of SAC, TD3 and AE-DYNA can be found in the Supplementary Material.

TABLE 2 Hyperparameters used for NAF2.

Name	Value
learning rate	0.001
γ	0.9999
batch size	100
buffer size	5,000
q_σ	0.02
q_{clip}	0.05
τ	0.001

During the training of the Model-Free (MF) agents, two callback functions were used: *Callback A* was called every 1,000 training steps to save the network parameters of the most recent agents to disk and; *Callback B* was called every 100 training steps to evaluate and log the performance of the most recent agent. The performance of the most recent agent was evaluated on a separate instance of QFBEnv in Callback B. Twenty episodes were played in sequence using the most recent agent to choose the actions. The training metrics were the average episode length, average undiscounted episode return and the average success rate. These values were logged with Tensorboard [15] and are used in the remaining part of this section to describe the training process of each algorithm.

SAC and TD3 required the most hyperparameter tuning to obtain a satisfactory result. NAF2 and PPO were less susceptible to hyperparameter tuning. AE-DYNA was more complex to set up correctly and also required some network adjustments in order for it to learn a successful policy on QFBEnv. The evolution of the agent throughout the training process was analysed off-line and the various policies trained by the different RL algorithms were re-loaded and compared.

All the agents used the same network architecture for their policies. The final architecture was chosen through a grid-search as an artificial Neural Network (NN) with two hidden layers having 50 nodes each and using the Rectified Linear Unit (ReLU) activation function. This network architecture was also used for the value function networks of the off-policy agents.

In on-line training on the QFB, the worst case episode length is 70 steps and every step is taken at a rate of 12.5 Hz. Therefore the maximum time of one episode on the QFB is:

$$\frac{70\text{ steps}}{12.5\text{ Hz}} = 5.6\text{ s}. \tag{7}$$

At episode termination, the actuators controlled by the action of the agent would, at worst, need to be re-adjusted to their initial settings at the start of the episode, e.g. set to reference current. The worst-case scenario occurs when an action is saturated

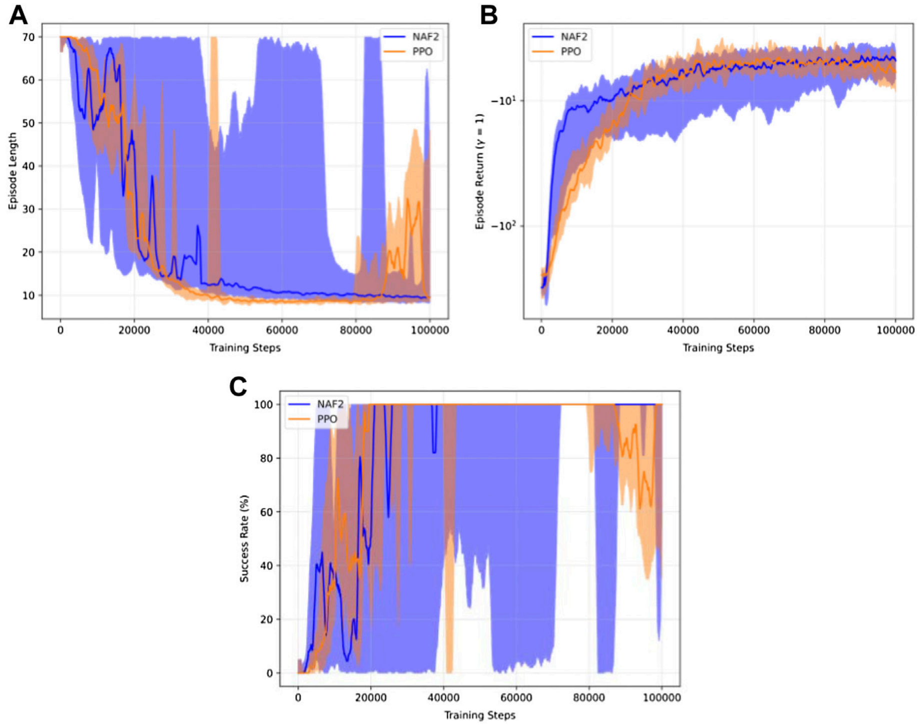


FIGURE 3
Performance statistics of NAF2 and PPO agents during training. The hyperparameters used are tabulated in Table 2 and Table 3, respectively. **(A)** Median episode length; **(B)** Median undiscounted episode return and; **(C)** Median success rate, of five agents initialised with different random seeds, per algorithm.

throughout the episodes. The worst-case re-adjustment time for one episode is thus equal to Eq. 7, 5.6s. Each RL algorithm trained in this work is attempted five times with different random seeds. An estimate of the real on-line training time on the LHC is also provided, which considers the worst case re-adjustment time for all the actuators. To achieve this, all training time estimates obtained from off-line environments were doubled to obtain the worst-case training time in an on-line environment.

Table 2 tabulates the hyperparameters that obtained the best NAF2 agent in this work. The discount factor, γ and the time constant for the Polyak averaging of the main and target networks, τ were set to the values listed in [9]. q_σ set up the standard deviation of the action smoothing noise applied at each step in QFBEnv when acquiring data. q_{clip} clipped the action smoothing noise to a range $[-q_{clip}, q_{clip}]$. q_σ and q_{clip} were chosen by trial and error. In addition to the smoothing noise, a decaying action noise was also applied during training. The following noise function was used:

$$a_i \quad \coloneqq \quad a_i \quad + \quad \mathcal{N}(0,1)\times\max\left(1-\frac{\mathrm{ep}_{idx}}{40},\ 0\right)$$

where a_i denotes the i^{th} action, $\mathcal{N}(0,1)$ is a standard Gaussian, and ep_{idx} is current episode number. Therefore, the action noise decays linearly to zero after 40 episodes. Note that the choice to decay the action noise relative to the number of episodes was arbitrary. The action noise can be made to decay relative to the number of steps taken so far during training.

Figure 3 shows the training performance statistics of the NAF2 agents. Figure 3A shows that the episode length decreases below 20 after 20000 steps. Figure 3C shows that the success rate goes to 100% after 20000 steps as well. From Figure 3B it can be seen that the average undiscounted return of the NAF2 agents was higher at the end of training. This shows a monotonic improvement in performance, regardless of the Min-Max bounds of NAF2 shown in Figures 3A,C. Partially solved episodes explain the large Min-Max boundaries. The policy manages to increase the reward in these episodes until a local minimum is reached without satisfying the successful early termination criterion. However, slight improvements in the policy push the states closer to the threshold, subsequently increasing the return.

Successful policies using NAF2 were relatively sample-efficient to train when compared to other algorithms attempted. The monotonic improvement shown in Figure 3B implies that a successful agent can be expected relatively early in terms of training steps taken on the environment. Some agents

TABLE 3 Hyperparameters used for PPO.

Name	Value
learning rate	0.00025
γ	0.99
ϵ	0.2

during the training performed well, with an average episode length of 20 steps after approximately 20,000 steps. In real LHC operation on the QFB, the worst-case training time of 20,000 steps is calculated by:

$$\frac{20,000\ \text{steps}}{12.5\,\text{Hz}} \times 2 \approx 53\,\text{min}$$

The hyperparameters shown in Table 3 are the salient parameters of the Proximal Policy Optimization (PPO) algorithm as implemented in Stable Baselines. Note that ϵ is the clipping parameter from Eq. 5. PPO proved to be the easiest algorithm to apply to QFBEnv in terms of hyperparameter tuning and usage. Figure 3A shows that PPO converges below an episode length of 20 after around 20000 training steps. A solution that drops the episode length to below ten steps is found after approximately 40000 steps. This remains stable until catastrophic forgetting of the policy occurs after 80000 steps. Figure 3C also shows that some agents expected a 100% success rate between 20000 and 80000 steps. Figure 3B illustrates that the episode return for PPO is not guaranteed to be monotonically improving. The best median performance of PPO was reached at approximately 86000 steps and obtained an undiscounted episode return of -4. All the metrics in Figure 3 shows that the PPO performance starts to degrade beyond this training step.

The catastrophic forgetting of the policy, however, did not occur quickly. At around 90000 steps, the expected episode length had increased to 20 steps again. A callback function can easily halt training if the policy starts forgetting and freezing the network parameters to obtain the best performing agent. This predictability is essential if the agent training occurs in the real LHC operation. Similarly to NAF2, approximately 20000 steps are required to learn a good policy which is equivalent to a worst-case on-line training time of 53 min.

5 Evaluation

This section evaluates the behaviour of the best policies obtained by each RL algorithm in corner cases of QFBEnv. These evaluations were performed by loading the network parameters of the agent with the best performance and recording its interactions with QFBEnv over multiple episodes. To ensure a fair comparison, a reference trajectory was created for each episode by using the actions from the PI controller and the same initial state. The PI controller was also subjected to the same tests, e.g., Gaussian noise was added to the action calculated by the PI controller in Figure 4B.

5.1 Effect of Gaussian noise

QFBEnv implements a deterministic model and actions passed through the *step* function are deterministic by default. However, by adding Gaussian noise to the action chosen by the policy, stochasticity can be introduced externally to QFBEnv. By subjecting each agent to a stochastic environment, the general robustness of each agent can be empirically verified. During this test, the initial state per episode was ensured to be sufficiently randomised to show more coverage of the state-action space.

The evolution of the episode trajectories are shown in sets of three evaluation episodes, e.g., Figure 4A. The state plots correspond to the evolution of ΔQ_H and ΔQ_V in time of the RL agents (top blue plots) and PI controller (top cyan plots), respectively. Green and red markers denote the start and end of each episode, respectively. A boundary (dashed green ellipse) is also drawn to indicate the success threshold state. The action plots correspond to the evolution of the 16 actions in time of the RL agents (bottom red plots) and PI controller (bottom magenta plots), respectively, until a terminal state is reached. Each set is obtained by applying Gaussian action noise with a zero mean and a standard deviation equal to 10%, 25%, and 50% of half the action range ([0,1]), respectively.

Figure 4A shows three episodes obtained with a deterministic NAF2 policy where it converges to an optimal state in approximately ten steps, while the PI controller takes approximately 25 steps until successful termination. However, it can be seen that the action chosen by the NAF2 policy at each terminal state is a non-zero vector. Ideally, the magnitudes of the actions are inversely proportional to the reward in Eq. 6, e.g., the PI actions of Figure 4A. This implies that NAF2 converged to a sub-optimal policy. Figures 4B–D show that the NAF2 policy satisfies the early termination criterion in each episode. The longest episode can be observed in Figure 4D to be approximately 20 steps long. Moreover, Episode #1 of Figure 4D shows that the PI controller failed to satisfy the successful early termination criterion. This indicates that the best agent trained by NAF2 performs better than the PI controller in a stochastic QFBEnv.

Figure 5A shows three episodes obtained by applying the actions sampled from the PPO stochastic policy, deterministically to QFBEnv, i.e., no noise applied. Similarly to NAF2, PPO converges to the optimal state. Furthermore, the actions of PPO start to converge back to zero at the end of the episode, which implies that PPO learned an optimal policy. Figures 5B–D show that the agent satisfies the early termination criterion in each scenario. PPO also shows a

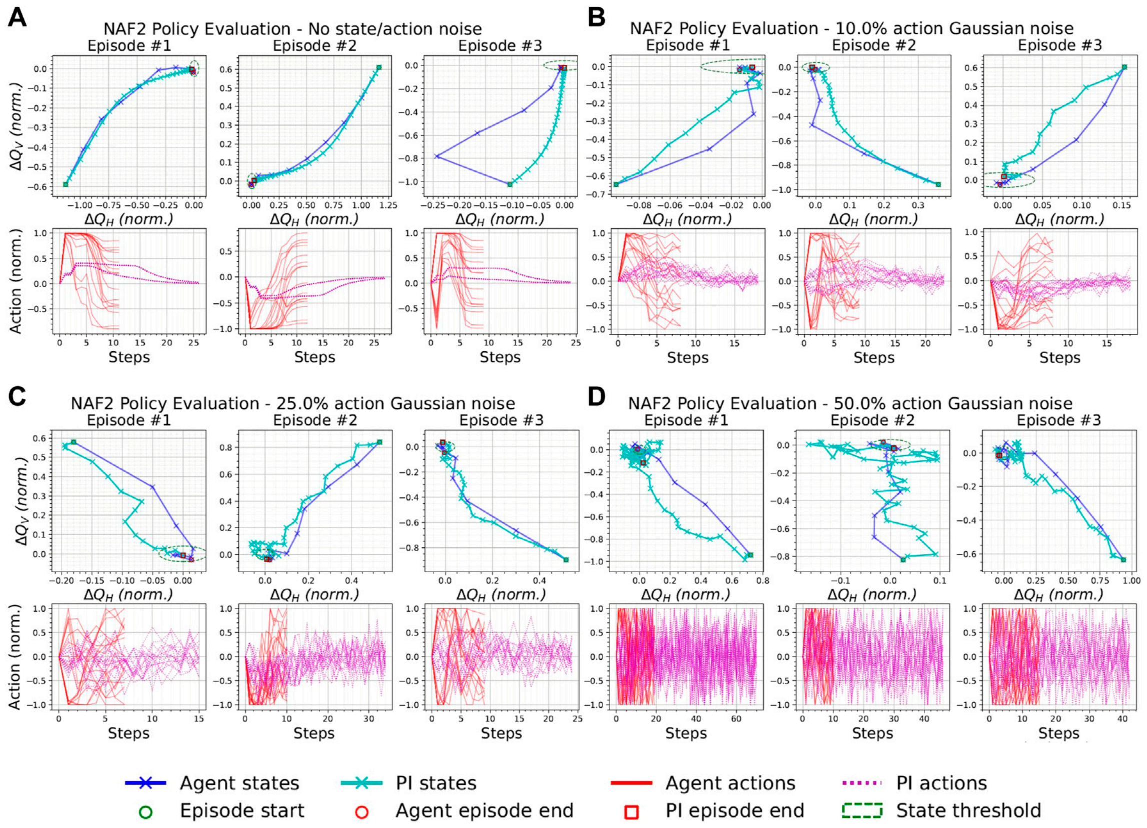


FIGURE 4
Episodes from the best NAF2 agent and the PI controller with the same initial states and with a varying additive Gaussian action noise with zero mean and standard deviation as a percentage of the half action space [0, 1]. **(A)** 0%, **(B)** 10%, **(C)** 25%, and **(D)** 50% Gaussian action noise.

wider range of episode lengths as the action noise is increased; the episode lengths in Figure 5D vary more than for NAF2 in Figure 4D. Therefore, this test indicates that the policy obtained by NAF2 is slightly more robust to action noise than PPO.

The action noise was also varied with a finer interval to aid with the analysis of the best agents under its effect and more statistics were taken on the performance of the agents and the PI controller e.g., Figure 6A. For each action noise value on the x-axis, 1,000 episodes were executed and used to obtain the statistics shown in the respective figures. These plots illustrate more information on the distribution of key measurements linked with the performance of the agents (in blue) and the PI controller (in red). In particular, the distributions of the following measurements are presented: 1) Episode length (one scalar per episode); 2) Distance of the terminal state from the optimal state of $[\Delta Q_H = 0, \Delta Q_V = 0]$, calculated by Eq. 8 and is referred to as Distance To Optimal (DTO) (one scalar per episode);

$$\mathrm{DTO} = \sqrt{\Delta Q_H^2 + \Delta Q_V^2} \tag{8}$$

and c) Concatenated values of the last actions applied in the episode before termination, be it successful or otherwise (array of size 16 per episode).

Table 4 tabulates the episode length statistics collected from 1,000 episodes, for the values of Gaussian action noise considered in the episode evaluation plots. As illustrated in Figure 6, the mean and the standard deviation of the episode lengths obtained by the NAF2 agent in Figure 6A and by the PPO agent in Figure 6B, outperformed those of the PI controller. Furthermore, the upper bound of the PI controller episode lengths reaches 70 steps at approximately 25% action noise; this indicates that the PI controller starts to fail to successfully terminate episodes at this point. This corresponds with the results shown in Figure 7 where at approximately 25% action noise, the upper bound of the DTO of the PI controller moves past the Goal threshold set within the QFBEnv (green dashed line). Both the NAF2 agent in Figure 7A and the PPO agent in Figure 7B maintain an upper bound DTO below the threshold until approximately 45% action noise.

Figure 8 illustrates the statistics of the last action chosen in each episode. Figure 8A exposes the weakness of the

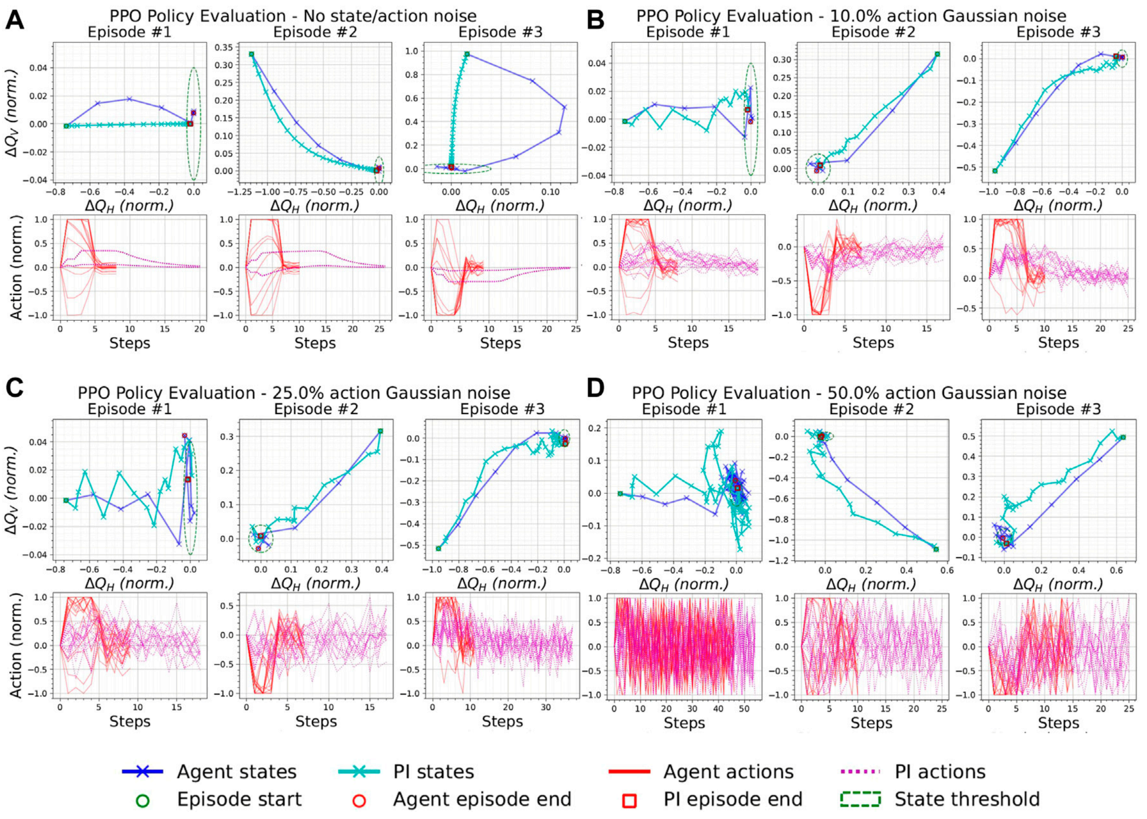


FIGURE 5
Episodes from the best PPO agent and the PI controller with the same initial states and with a varying additive Gaussian action noise with zero mean and standard deviation as a percentage of the half action space [0, 1]. **(A)** 0%, **(B)** 10%, **(C)** 25%, and **(D)** 50% Gaussian action noise.

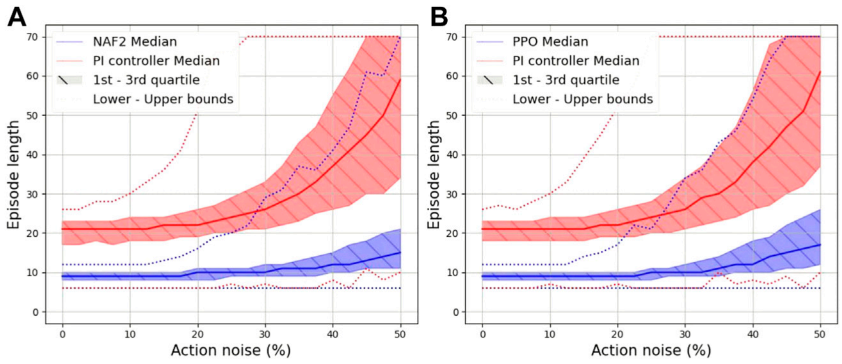


FIGURE 6
Effect of action noise on the episode length due to varying action noise on **(A)** the best NAF2 agent, **(B)** the best PPO agent and the PI controller.

NAF2 agent, where it can be observed that the values of the last actions are unpredictable even without action noise, i.e., at 0% action noise, the PI controller actions are below ±0.05 while the NAF2 action value distribution populates most of the action range. These results signify that NAF2 has trained a policy which outperforms the PI controller in a noisy environment, albeit the policy is sub-optimal.

TABLE 4 The statistics (mean ± std.) for the episode length obtained by the best RL agents trained in Section 4 and PI controller with respect to the amplitude of Gaussian action noise.

Action noise	0%	10%	25%	50%
NAF2	9.06 ± 1.35	9.16 ± 1.41	9.82 ± 1.91	17.89 ± 9.24
PPO	8.80 ± 1.28	8.89 ± 1.32	9.77 ± 2.11	20.69 ± 12.60
PI controller	20.05 ± 3.78	20.41 ± 3.99	24.81 ± 7.47	53.32 ± 18.37

Figure 8B illustrates that the PPO policy chooses last actions which behave similarly to the PI controller in the presence of Gaussian action noise. At 0% action noise, the distribution of the PPO last action values are slightly larger than those of the PI controller. However, as the action noise is increased the distribution widths of both PPO and the PI controller increase at the same rate. This consolidates what was observed in Figure 5, where the last action values are dispersed with respect to the amplitude of the action noise applied. This concludes that PPO successfully trained a policy on QFBEnv which is also the closest to the optimal policy. LATEX.

5.2 Effect of actuator failure

In this test, the performance of the best policies trained in this work was analysed in the presence of magnet failures. For each episode shown in this section, an action was chosen at random at a predetermined step in the episode. For the remaining steps until a terminal state, the action chosen was set to -1 to simulate a cool-down of the magnet after a circuit failure. The corresponding action obtained by the PI controller was set to the same value. While this test is not a perfect representation of magnet failures in the LHC, it is a worst-case scenario that tests the performance of the policies and PI controller in unseen and unideal conditions.

For each policy, two scenarios with three episodes each are shown. In the first scenario, one actuator fails on step 1; in the

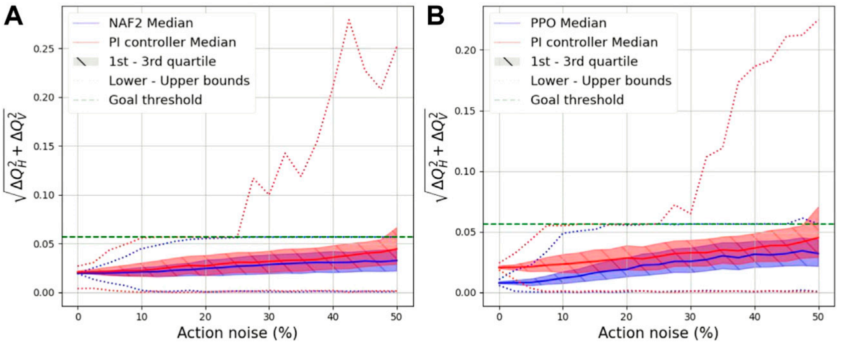


FIGURE 7
Effect of action noise on the distance to the optimal point at the end of the episode due to varying action noise on **(A)** the best NAF2 agent, **(A)** the best PPO agent and the PI controller.

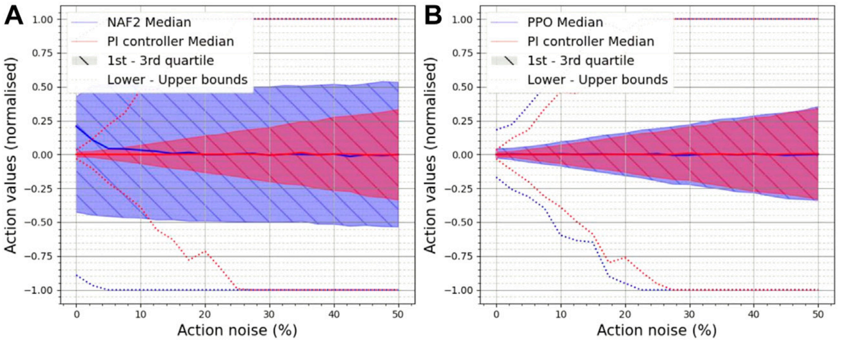


FIGURE 8
Effect of action noise on the last action used in the episode due to varying action noise on **(A)** the best NAF2 agent, **(B)** the best PPO agent and the PI controller.

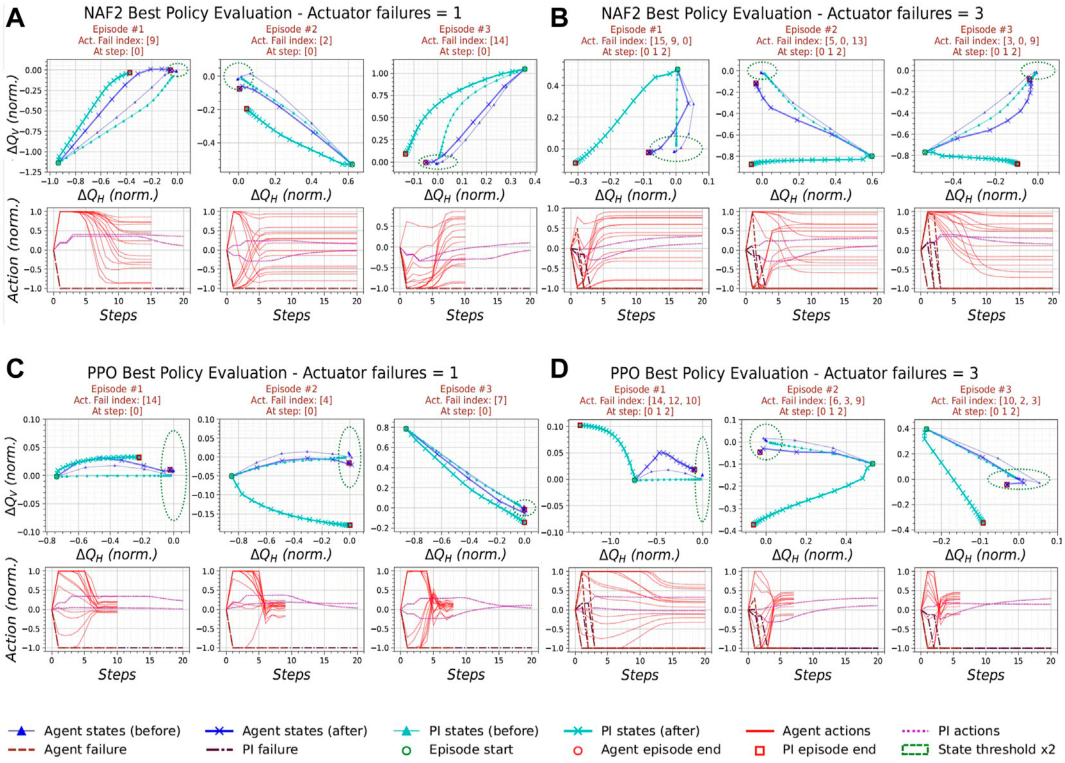


FIGURE 9
Episodes from the best RL agents and the PI controller under the effect of different number of actuator failures.**(A)** NAF2 agent with one actuator failure, **(B)** NAF2 with three actuator failures, **(C)** PPO agent with one actuator failure, **(D)** PPO agent with three actuator failures.

second scenario, three actuators fail on steps 1, 2, and 3, respectively. All the plots shown in this test also show an episode trajectory obtained by repeating the episode with the same initial conditions and all actions functioning. The episode trajectories affected by the actuator failures use × as a marker, while the episodes using all actions use ▲ as a marker. It was also found that a maximum episode length of 70 obtained large state deviations for certain scenarios. To aid the analysis of the results, the maximum episode length was set to 20.

The evaluation episodes from the best NAF2 agent with one actuator failure are shown in Figure 9A. It can be observed that the PI controller already diverges from the threshold boundary and fails on all episodes. NAF2 succeeds in terminating $\frac{1}{3}$ of the episodes in under 20 steps. At three actuator failures in Figure 9B, the terminal state achieved by NAF2 is within four times the value of the threshold, while the PI controller terminal state shifts farther from the optimal point.[3]

Similarly to NAF2, PPO performs better than the PI controller in all scenarios of actuator failures in Figure 9C and Figure 9D. The effect of an increasing number of actuator failures on the best PPO policy is also evident in the action plots of the PPO actuator failure tests. These figures, along with bottom plots of Figure 5A, show that when all actions are used, the actions decay the closest to zero at the end of the episode. They also show that the actions still decay during actuator failures. However, the actions remain separated by a range proportional to the number of actuators that failed during the episode. This observation suggests that the PPO policy has successfully generalised the optimal policy trained on one environment to another environment with slightly different model dynamics.

5.3 Effect of incorrect tune estimation

This test subjects the best agents to the effects of 50 Hz noise harmonics on the BBQ system. A similar procedure to the previous test is followed, where the best agent trained by each respective RL algorithm is loaded and is used to produce evaluation episodes. The only difference in this test is that after each step in the environment, the state is intercepted and a perturbation is added, which simulates the effect of

[3] $\mathcal{U}$ is a uniform distribution.

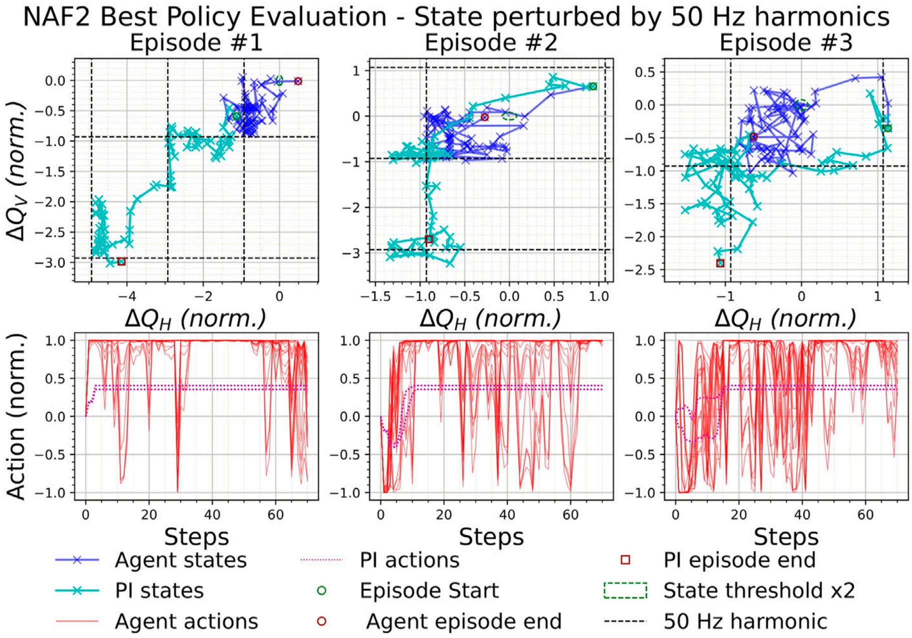


FIGURE 10
Effect of inaccurate tune estimation on the best NAF2 agent.

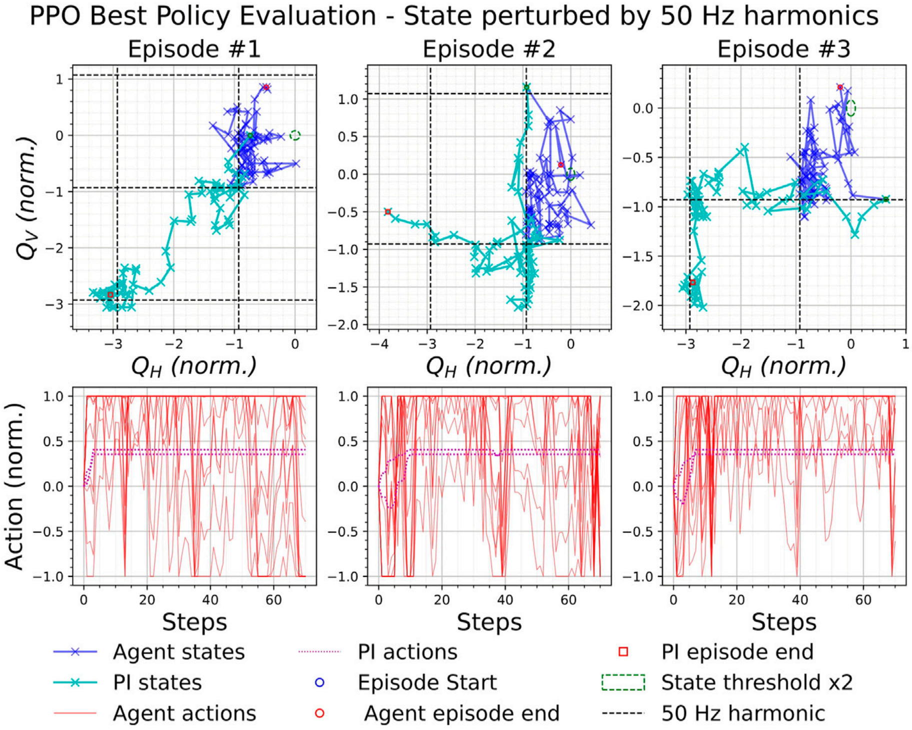


FIGURE 11
Effect of inaccurate tune estimation on the best PPO agent.

50 Hz noise harmonic-induced perturbations within the state.

These perturbations were obtained by the following steps: 1) The state, ΔQ, was added to a random frequency, f_r[2]; 2) The realistic second-order system spectrum simulation procedure in [16] was performed for a spectrum with a resonance frequency, $f_{res}^{true} = f_r + \Delta Q$ and damping factor $\zeta \sim 10^{\mathcal{U}(-2.5,-1.8)}$ [3]; 3) The BQ algorithm is used to obtain perturbed tune estimates [16]. All the plots shown in this section also show black dashed lines which mark the locations of the horizontal and vertical 50 Hz harmonics in normalised state space.

Figure 10 and Figure 11 show three episodes obtained by the best NAF2 and PPO policies, respectively, when the state is perturbed by 50 Hz noise harmonics. It can be observed that the states of the policies (dark blue) appear to be concentrated around an intersection of a horizontal and vertical 50 Hz noise harmonics, which is closest to the optimal point of the state space. On the other hand, the PI controller states sometimes extend up to three 50 Hz harmonics from the optimal point. This observation suggests that even without the tune estimation renovation discussed in [16, 17], it is possible to train NAF2 and PPO agents to maintain the tune error as close as possible to the optimal point.

5.4 Summary

When taking into consideration the algorithms shown in the Supplementary Material, NAF2 and PPO trained the best two policies. However, AE-DYNA-SAC was the most sample efficient and also obtained a policy that is stable in low action noise. The policies trained by TD3 and SAC were sometimes successful. However, their performance was significantly worse than NAF2 and PPO. On the other hand, the best SAC-TFL agent trained an adequate policy that works well on QFBEnv.

6 Conclusion and future work

This work explored the potential use of RL on one of the LHC beam-based feedback controller sub-systems, the QFB. An RL environment called QFBEnv was designed to mimic the QFB in real operation in the LHC. The original implementation of the QFB PI controller was re-implemented to serve as a reference agent to the trained RL agents.

A total of five RL algorithms were selected from literature and trained on QFBEnv. A series of evaluation tests were performed to assess the performance of the best two agents against the standard controller paradigm. These tests were designed to capture the performance of the agents during corner cases. PPO and NAF2 obtained a high performance in each test. Slightly better generalisation was also observed during the actuator failure tests. The training and evaluation of the other RL algorithms attempted in this work are in the Supplementary Material. It was not easy to tune the hyperparameters of TD3 even in the most straightforward deterministic cases, while depending on the implementation, the SAC algorithm could learn a good policy. Finally, AE-DYNA-SAC was the most sample efficient agent attempted, and the performance of the best policy trained was comparable to that of the PI controller.

Our studies showed that RL agents could generalise the environment dynamics and outperform the standard control paradigm in specific situations which commonly occur during accelerator operation.

Future work will concentrate on more sample efficient RL algorithms, e.g., Model-Based Policy Optimization (MBPO) [18] since the real operation is restricted by the beam time. By addressing robustness and sample efficiency when training on simulations, it will be possible to design an RL agent that can be feasibly trained on the QFB during the LHC Run 3. As was shown in this work, this would allow for more reliable tune control even in situations where the standard controller is not applicable.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

LG and GV contributed to the conception and design of the study. LG contributed to the implementation, logging and plotting of results. All authors contributed to the manuscript revision.

Funding

Project DeepREL financed by the Malta Council for Science and Technology, for and on behalf of the Foundation for Science and Technology, through the FUSION: R&I Research Excellence Programme.

[2] Initialised randomly and set constant throughout one episode.

[3] $\mathcal{U}$ is a uniform distribution.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphy.2022.929064/full#supplementary-material

References

1. Steinhagen RJ. *LHC beam stability and feedback control-orbit and energy*. Ph.D. thesis. Aachen, Germany: RWTH Aachen U (2007).

2. Grech L, Valentino G, Alves D, Calia A, Hostettler M, Wenninger J, Jackson S. *Proceedings of the 18th international conference on accelerator and large experimental Physics control systems (ICALEPCS 2021)* (2021).

3. Gasior M, Jones R. *Proceedings of 7th European workshop on beam diagnostics and instrumentation for particle accelerators (DIPAC 2005)* (2005). p. 4.

4. *MAD - methodical accelerator design* (2019). p. 2019–626. Accessed.

5. Sutton RS, Barto AG. *Reinforcement learning - an introduction, adaptive computation and machine learning*. Cambridge, MA, USA: The MIT Press (2018).

6. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, et al. Mastering the game of Go with deep neural networks and tree search. *Nature* (2016) 529:484–9. doi:10.1038/nature16961

7. Kurutach T, Clavera I, Duan Y, Tamar A, Abbeel P. *Model-Ensemble Trust-Region policy optimization* (2018). arXiv:1802.10592 [cs.LG].

8. Gu S, Lillicrap T, Sutskever I, Levine S. *International conference on machine learning* (2016). p. 2829–38.

9. Hirlaender S, Bruchon N. *Model-free and bayesian ensembling model-based deep reinforcement learning for particle accelerator control demonstrated on the FERMI FEL* (2020). arXiv:2012.09737 [cs.LG].

10. Fujimoto S, van Hoof H, Meger D. *Addressing function approximation error in Actor-Critic methods* (2018). arXiv:1802.09477 [cs.AI].

11. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. *Proximal policy optimization algorithms* (2017). arXiv:1707.06347 [cs.LG].

12. Haarnoja T, Zhou A, Abbeel P, Levine S. *Soft Actor-Critic: Off-Policy maximum entropy deep reinforcement learning with a stochastic actor* (2018). arXiv:1801.01290 [cs.LG].

13. Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, et al. *Openai gym* (2016) 1606:01540.

14. Ziegler JG, Nichols NB. *J Dyn Syst Meas Control* (1942) 115:220.

15. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. Savannah, GA: USENIX Association (2016). p. 265–83.

16. Grech L, Valentino G, Alves D, Gasior M, Jackson S, Jones R, et al. *Proceedings of the 9th international beam instrumentation conference (IBIC 2020* (2020).

17. Grech L, Valentino G, Alves D. A machine learning approach for the tune estimation in the LHC. *Information* (2021) 12:197. doi:10.3390/info12050197

18. Janner M, Fu J, Zhang M, Levine S. In: H Wallach, H Larochelle, A Beygelzimer, F d Alché-Buc, E Fox, R Garnett, editors. *Advances in neural information processing systems*, Vol. 32. Red Hook, NY, USA: Curran Associates, Inc. (2019).

19. Achiam J. *Spinning up in deep reinforcement learning* (2018).

20. Dhariwal P, Hesse C, Klimov O, Nichol A, Plappert M, Radford A, et al. *Openai baselines* (2017).

21. Hill A, Raffin A, Ernestus M, Gleave A, Kanervisto A, Traore R, et al. *Stable baselines* (2018). Available from: https://github.com/hill-a/stable-baselines (Accessed Septermber, 2021).

22. Hirländer S. *MathPhysSim/FERMI_RL_Paper*. Preprint release (2020).

23. *Addressing function approximation error in actor-critic methods* (2018). arXiv:1802.09477 [cs.AI].

24. Pearce T, Anastassacos N, Zaki M, Neely A. *Bayesian inference with anchored ensembles of neural networks, and application to exploration in reinforcement learning* (2018). arXiv:1805.11324 [stat.ML].

ORIGINAL RESEARCH
published: 08 September 2022
doi: 10.3389/fphy.2022.875464

SPIRAL2 Cryomodule Models: A Gateway to Process Control and Machine Learning

Adrien Vassal[1,2], *Adnan Ghribi*[2,3]*, *François Millet*[1], *François Bonne*[1], *Patrick Bonnay*[1] *and Pierre-Emmanuel Bernaudin*[1,2]

[1]*Commissariat à l'Energie Atomique et aux énergies appropriées (CEA), Paris, France,* [2]*Grand Accélélérateur National d'Ions Lourds (GANIL), Caen, France,* [3]*Centre National de la Recherche Scientifique (CNRS), Paris, France*

From simple physical systems to full production lines, numerical models could be used to minimize downtime and to optimize performances. In this article, the system of interest is the SPIRAL2 heavy ion accelerator cryogenic system. This article illustrates three different applications based on a SPIRAL2 cryostat model: optimal controller synthesis, virtual sensor synthesis, and anomaly detection. The two first applications have been deployed on the system. Experimental results are used to illustrate the benefits of such applications. The third application is a case study based on data generated from a thermodynamic twin model.

Keywords: machine learning, cryogenics, modeling, accelerators, thermodynamics, control

Edited by:
Alexander Scheinker,
Los Alamos National Laboratory (DOE), United States

Reviewed by:
Benjamin Bradu,
European Organization for Nuclear Research (CERN), Switzerland
Christine Darve,
European Spallation Source, Sweden

Correspondence:
Adnan Ghribi
adnan.ghribi@ganil.fr

Specialty section:
This article was submitted to Interdisciplinary Physics, a section of the journal Frontiers in Physics

Received: *14 February 2022*
Accepted: *10 May 2022*
Published: *08 September 2022*

Citation:
Vassal A, Ghribi A, Millet F, Bonne F, Bonnay P and Bernaudin P-E (2022) SPIRAL2 Cryomodule Models: A Gateway to Process Control and Machine Learning.
Front. Phys. 10:875464.
doi: 10.3389/fphy.2022.875464

1 INTRODUCTION

SPIRAL2[1] is a heavy ion accelerator located in Caen, France. Its main part is a linear superconducting accelerator (LINAC) [1] composed of 26 bulk niobium radio-frequency (RF) resonators that accelerate charged particles by the mean of electromagnetic fields [2]. To be operated, those resonators need to be maintained in their superconducting state. As the niobium superconducting transition temperature at atmospheric pressure reaches 9.2 K, a cryogenic system is required. The resonators, also called SRF[2] cavities, are coupled to the RF system, as well as vacuum and cryogenic components. The assembly of these subsystems forms a cryomodule.

The cooling power is provided by a cryoplant with a maximal capacity of 1300 W at 4.2 K. A cold box coupled with a 5000 L Dewar provides the necessary liquid helium to all the cryomodules through a cryodistribution. Inside the cryomodules, the liquid helium evaporates to extract heat from the resonator, and cold gaseous helium is returned to the cold box. More details on the cryogenic system can be found in [3,4]. As a perturbation in the cooling system might lead to a shutdown of the accelerator, it is mandatory to develop a highly reliable operation and control system. To achieve this, modeling tools are developed to improve the control robustness, predict valuable information, and detect faults or anomalies.

The present article mainly focuses on the cryomodules and not on the overall cryogenic system. The first section is dedicated to the modeling of the cryomodules. In the second section, an optimized control law is proposed. The third section details the synthesis of a virtual sensor used to predict unmeasured parameters. The last section is dedicated to fault detection using machine learning techniques.

[1] Système de Production d'Ions RAdioactifs en Ligne de 2e génération.
[2] Superconducting Radio-Frequency.

TABLE 1 | Main thermal differences between type-A and type-B cryomodules.

Characteristic	Type-A	Type-B
Helium bath volume [L]	20.5	91.5
Static heat load [W]	3.5 ± 1.4	12.5 ± 1.8
RF heat load [W]	5.8 ± 2.2	12.1 ± 2.6

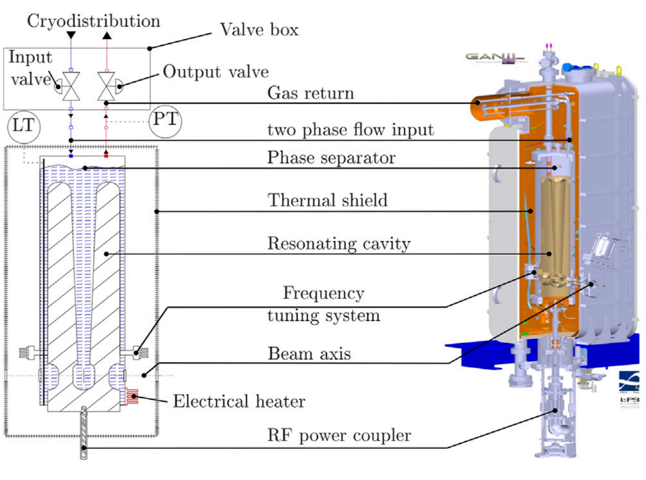


FIGURE 1 | Representation of a type-A cryomodule. On the left is a simplified scheme. On the right is a 3D cut view. LT and PT are, respectively, level transmitter and pressure transmitter.

2 MODELING OF THE CRYOMODULES

2.1 Description

There are two types of cryomodules, namely, type-A and type-B. The main difference between them lies in the fact that type-A contains one resonating cavity, whereas type-B contains two of them. For more details concerning the cryomodules design and performance, please refer to [5,6].

In terms of thermodynamics, both types of cryomodules undergo different heat loads. First is the static heat load induced by the heat transfer between the cold and their surrounding parts. Second is the dynamic heat load due to the RF resistive losses in the resonating cavity. Heat load amplitudes are different for the two types of cryomodules. For the 4.4°*K* bath, those characteristics as well as the volumes are given in **Table 1**.

The cryogenic system is in charge of keeping the superconducting RF cavities under their superconducting critical temperature at all times. This requirement is achieved by keeping the cavities submerged in a liquid helium bath. As the cavities are fed with RF power, a liquid helium bath ensures the resistive heat loads extraction at the cavity's surface walls. As a consequence, the temperatures of the cavities are kept uniform and stable at 4.4°K far beyond their superconducting critical temperature of 9.2°*K*. Would the cavity be partially exposed out of the liquid helium bath, it would undergo a quench[3]. Three main elements, shown in **Figure 1**, insure this constraint: a phase separator filled with liquid helium at 4.4°K and 1,200 mbar, a thermal shield that surrounds the phase separator and is kept at 60°K, and finally a valves box containing all the valves used to control cryogenics operation. As the phase separator is the most critical element of the cryogenic system, we will only focus on that element and its associated valves. The **Figure 1** presents a simplified scheme with the subsystems of interest.

The phase separator is fed with liquid helium through the input valve, which is used to regulate the level of liquid. Due to thermal heat load, liquid helium evaporates and is returned to the cold box. In that process, gas goes through the output valve, which is used to regulate the pressure within the phase separator.

Both the valves and the phase separator have been modeled. The equations governing the operation of the valve are the ones given in the standard ISA [7], whereas the phase separator dynamics are described through energy and mass balance. The equations have been implemented in the Simcryogenics library [8] of MATLAB®, which is a modeling tool used to simulate and optimize cryogenic systems. Helium properties are extracted from tabulated data using the HEPAK® package. As those equations have been extensively described in [9–11], they will not be discussed in this article. Rather, the comparison between experimental and simulation results will be emphasized.

2.2 Model vs. Data

The simulation results for both cryomodule types have been compared to experimental data. For each of the cryomodules, an operating scenario has been performed starting from stable operating conditions[4]. This scenario is a series of steps applied to the input and output valve opening command. The same values have been applied to the model and to the real process in an open-loop manner. The comparison obtained for cryomodule 1 (the first one on the line considering the beam direction) is shown in **Figure 2**. The comparison shows a good agreement between experimental and simulated data for both level and pressure dynamics. It is worth mentioning that the uncertainty of the modeled liquid helium level increases with time as the level is an integrator system. Furthermore, the high uncertainty on the pressure at time t = 1,500 s is mostly due to the valve position uncertainty: an error of ±1% on valve position could lead to a pressure uncertainty up to 10 *mbar*. Finally, the pressure peak occurring at t = 1,200 s is due to a pressure oscillation in the cryodistribution (i.e. the inlet boundary of the model).

Similar results have been observed on all other cryomodules. The following criteria have been calculated for each comparison:

$$Cr = \frac{100 \cdot \int_{t_{init}}^{t_{final}} \left(V_{mes}(t) - V_{sim}(t)\right)^2 dt}{V_{moy}^{init} \cdot t_{total}}, \tag{1}$$

where Cr is the criteria representing the integral of the error between measured and simulated data. t_{init}, t_{final} , and t_{total} are, respectively, the initial time, the final time and the overall duration of the scenario, whereas $V_{mes}(t)$ and $V_{sim}(t)$ are,

[3] Loss of the superconducting state.

[4] Operating conditions mean the internal thermal conditions (heat load), the external hydraulic and pneumatic conditions (set by the cryoplant), and the operation set-points (typically liquid helium bath pressure and level).

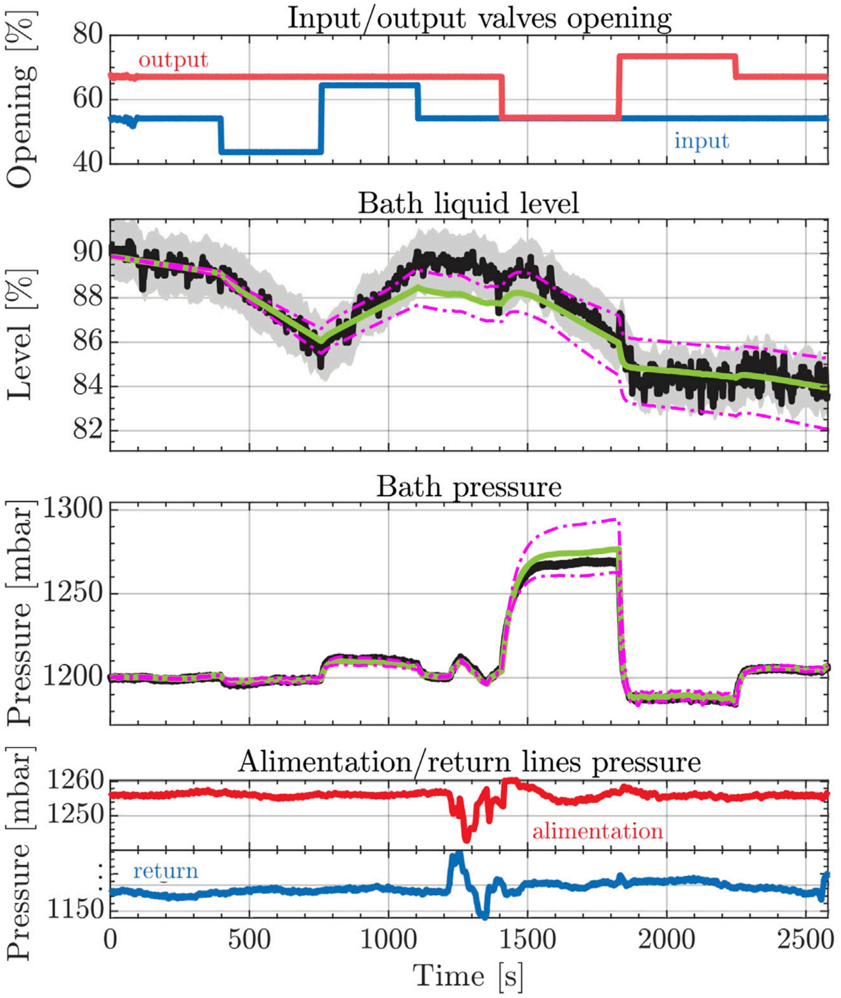


FIGURE 2 | Model vs. measurement for the first type-A cryomodule. Measurement and associated uncertainty are, respectively, represented by black line and gray background. Simulation value and uncertainty are, respectively, represented by green lines and magenta dash-dotted lines.

respectively, the measured and simulated values. Finally, V_{moy}^{init} designates the mean value at the beginning of the scenario. Normalizing by the overall duration and the mean value makes it possible to compare multiple scenarios with different durations and operating conditions. The criteria values obtained for the scenario shown in **Figure 2** and applied to all cryomodules are plotted in **Figure 3**. The latter plot gives an important insight into the usability of such a model in a generic way for all cryomodules across the LINAC. In fact, obtained *Cr* values give a deviation sufficiently small to be considered for applications such as control and fault detection. The next sections will investigate this feasibility.

3 OPTIMAL CONTROLLER SYNTHESIS

3.1 Problem Overview

Cryogenic system control loops are critical items that can affect the overall accelerator. Two requirements are to be met in order to allow the nominal operation of the RF cavity. The first is to ensure that the temperature of the cavity remains below its critical value. Otherwise, the cavity could quench[5]. To do so, the cavity is submerged in a liquid helium bath, and the level of liquid helium is regulated through a PID (Proportional–Integrator Derivative) controller acting on the input valve (see **Figure 1**). The goal is to maintain a level at 90% ± 5% which is high enough to maintain the overall cavity fully submerged with a comfortable operating margin. The second is to ensure that the shape of the cavity does not change as the performances of the resonator are intrinsically linked to the cavity shape. This could be seen in the expression of its unloaded quality factor:

$$Q_{f0} = \frac{G}{Rs}, \tag{2}$$

[5] Fast transitions between superconducting state and normal conducting state that can lead, in the worst case, to irreversible mechanical damages.

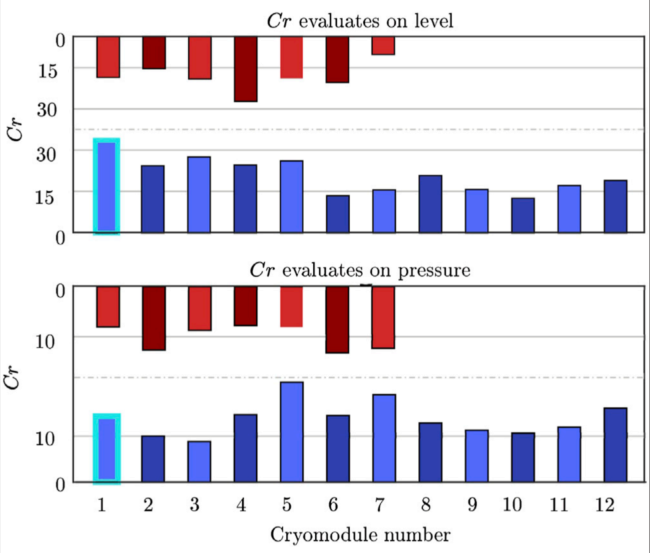


FIGURE 3 | Evaluation of the criteria *Cr* on each cryomodule. The type-A and type-B cryomodule are, respectively, plotted in blue and red. The blue highlighted element corresponds to the cryomodule investigated in **Figure 2**.

where Q_{f0} is the unloaded quality factor, Rs is the surface resistance, and G is the geometric factor that depends on the surface and the volume of the cavity. As the cavity is submerged in liquid a helium bath, any pressure variation (ΔP) in the separator will induce a mechanical force on the cavity walls that slightly deforms the cavity. This results in a variation of the geometric factor that can lead to a drop in the cavity quality factor, hence significantly reducing the nominal cavity accelerating gradient.

Considering the bandwidth of the cavity and its associated RF system, a pressure variation limit of $\Delta P = \pm 5\ mbar$ has been set up for the SPIRAL2 cryomodules. It is worth mentioning that the nominal pressure of the helium bath is 1,200 $mbar$, which means that a ΔP of $\pm 5\ mbar$ represents a tolerance of ±0.41%. Both the level and the pressure are regulated through PID controllers. Although the PID performance is enough to achieve the level requirement, it is not the case for the pressure requirement. Even using a state-of-the-art [12] PID (Proportional, Integral Derivative) tuning tool, we were not able to maintain the pressure variation within a range of $\pm 5\ mbar$ for long periods of time (i.e., more than a few hours) without having significant overshoots. This is probably due to the fact that the two regulation loops are coupled: an action on the input valve influences the level and the pressure. A similar statement is also true for the output valve: an action on the output valve has an impact on both on the pressure and the level. As PID controllers are more suitable in the case of linear SISO[6] system, another control algorithm is necessary to achieve the project requirement.

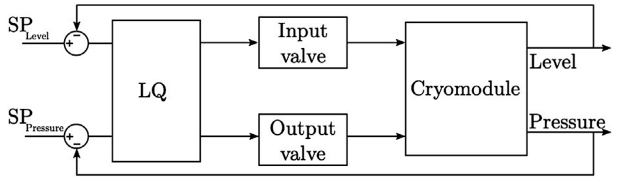


FIGURE 4 | Block diagram of the synthesized LQ regulator. SP designates the setpoint.

3.2 Synthesis of a LQ Regulator

Few parameters have to be considered while choosing the most suitable solution for the cryomodules control loops. First, the cryomodule cryogenic system is a multiple inputs multiple outputs (MIMO) system with two valves as inputs and the level and pressure as outputs. As there is internal coupling between all inputs and outputs, a controller that can handle this coupling is mandatory. Second, as the accelerator will be used for many years, it must be a solution proven on multiple systems with full documentation. Third, the controller has to be implemented in a dedicated PLC (Programmable Logical Controller) with a limited amount of calculation capacity. Considering those parameters, an LQ (Linear Quadratic) controller seems to be a good candidate. The block diagram of such controller applied to our system is given in **Figure 4**. The mathematical development of this controller has already been described in [9,11]. In this section, we will only recall the main equations of the discrete LQ controller and focus on experimental results.

The principle of a LQ controller is to synthesize a state feedback gain such that the command input is given by:

$$u(k) = -K \cdot x(k), \tag{3}$$

where K is the state feedback gain and x the state of the system. K is calculated so that it minimizes the following quadratic cost:

$$J = \sum_{i=k}^{\infty} x(i)^T \cdot Q \cdot x(i) + u(i)^T \cdot R \cdot u(i), \tag{4}$$

where J is the cost, and Q and R are respectively state and input weights. As for gain and integral time for a PI controller, the goal is to tune the matrices Q and R to fulfill the process specifications. Details about the way to tune those gains are given in [9].

The calculation of the state feedback gain K requires the state-space model of the system which could be directly generated with the previously described model and a linearization algorithm such that the one described in [13]. To allow a comparison between the existing PID and the proposed regulation law, the LQ controller has been implemented on the existing PLC of each cryomodule. Even if they have a limited calculation capacity (a work memory of 192 Ko), it is more than enough for the proposed LQ controller which only requires around 30 multiplications/additions per sampling time. This is due to the fact that only the control law described in **Eq. 3** and its associated Luenberger observer [14] have been implemented. The calculation of the state feedback gain K that minimizes the cost **Eq. 4** could be carried out offline using dedicated optimizers.

[6] Single Input Single Output.

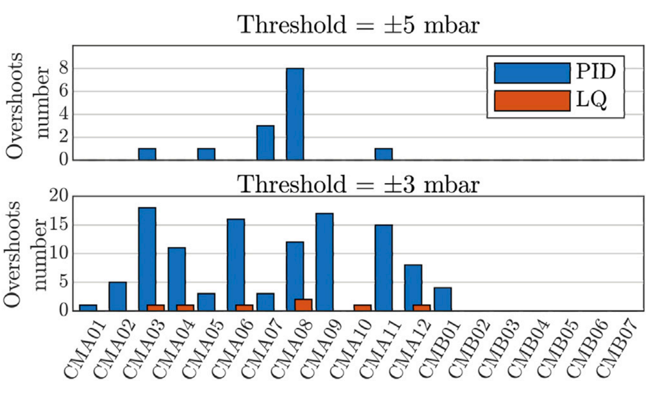


FIGURE 5 | LQ vs. PID pressure overshoot number for one night.

3.3 Experimental Results

Classical tests like set point variation and output disturbance test (using electrical heaters as disturbance sources) have been performed on both PID and LQ controllers. The results of those tests show that the LQ regulator is slightly better in terms of robustness and speed than the PI controller. Nevertheless, our main interest is to see which controller is able to respect the level and pressure requirements during a long period of operation. To check this, the following experiment has been realized for two nights[7]. During the first night (12 h), all cryomodules are regulated by PID controllers. During the second night, they are regulated by LQ controllers. For both cases, the level requirement has been respected, but not the pressure requirement. To illustrate this, the number of times the pressure overshoots the threshold of ±5 *mbar* has been used as a metric to compare the two controllers. The result of the comparison is given in **Figure 5**. The LQ controller shows no pressure overshoot at all, whereas the PID controller shows multiple ones. However, the LQ controller could not perfectly dump pressure oscillations. To illustrate this, a counter of overshoot with a tightened pressure threshold of ±3 *mbar* (in comparison with the specification of ±5 *mbar*) has also been plotted in **Figure 5**. Nevertheless, the results obtained with the LQ regulator are satisfying considering process requirements. As the algorithm has been deployed in the PLCs for the purpose of the test, it is already available for the current operation. This new control strategy is an important improvement that could reduce the accelerator downtime as one pressure overshoot may arise safety chains that shuts down the accelerator beam. One drawback of such method is the knowledge of the thermal set point of operation. This setpoint depends on static load, RF losses, and other effects such as beam loading. A deviation from the setpoint due to isolation vacuum leaks or field emitters in the cavity might make the LQ control worse than a simple PID[8]. In this matter, having state observers able to monitor the thermal behavior of a cryomodule is vital. Such an observer could drive the change of the thermal operation set point and LQ inputs to automatically adapt to the real state of the system. The synthesis of precision state observers using supervised learning will be the subject of future studies. The next section is the first study of a thermal load observer based on a twin model synthesis as a starting point for future planned studies.

4 VIRTUAL SENSOR

4.1 Problem Overview

As the RF signal injected in the resonator is sinusoidal, it generates energy dissipation in the cavity walls called AC losses [15]. Those losses are considered an indicator of the cavity state: an abrupt raise of those losses can indicate that a part of the cavity is no more in a superconducting state. This could be the premise of a global quench of the cavity. On another timescale, a slow increase of the dissipated AC losses can indicate a pollution of the cavity with non-superconducting elements. In the case of SPIRAL2, there is no continuous measurement of these AC losses. Measurements can only be performed when the cavity is not in operation as the measurement method is intrusive [16]. There is no operating solution in the case of SPIRAL2 that would allow us to perform such measurements online and without disturbing the process.

To solve this problem, we proposed a method to estimate these losses based on the phase separator model and an extended Kalman filter (EKF) [17].

4.2 Synthesis of an Extended Kalman Filter

From the phase separator point of view, the AC losses represent an external thermal heat load. The more AC losses, the more heat has to be extracted through the evaporation of liquid helium. Equations that link the AC losses to the thermal heat load are given in [18]. Knowing this, measuring the AC losses is equivalent to measuring the heat load extracted by the liquid helium bath. Nevertheless, as for the AC losses, there is no continuous measurement of the heat load dissipated in each cavity in the SPIRAL2 cryogenic system. Discontinuous measurement can be made by measuring the evaporating rate of the liquid helium [19], but once again it is an intrusive method that could not be realized during operation. This is where the cryomodule model becomes very useful: using the model and process measurements such as phase separator level and pressure as well as valve opening, it is possible to predict the current heat loads. Therefore, the idea is to synthesize an observer (also called a virtual sensor in that case) of the heat load.

An extended Kalman filter seems to be the best choice as it is designed to work with nonlinear processes and has been successfully used in many applications [20,21]. The process diagram of such an observer applied to our process is described in **Figure 6** where it is decomposed into elementary steps represented as a number in green circles:

- 1: calculate phase separator internal energy (e) and density (ρ) through property interpolation using bath pressure and liquid level.

[7] Night is chosen to avoid daily operations that could induce comparison bias.
[8] See [10] for thermal set points for both type A and type B cryomodules.

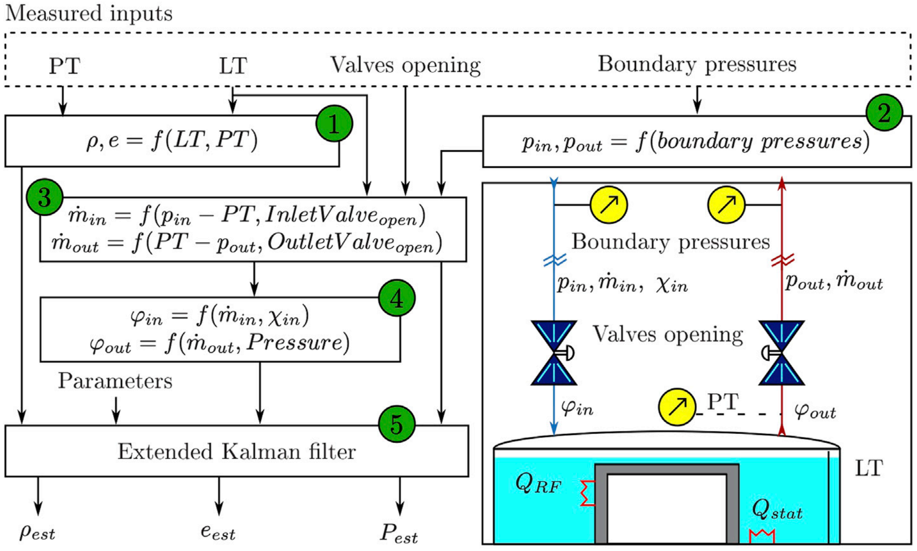

Measured inputs
PT
LT
Valves opening
Boundary pressures
$\rho, e = f(LT, PT)$
1
$p_{in}, p_{out} = f(boundary\ pressures)$
2
3
$\dot{m}_{in} = f(p_{in} - PT, InletValve_{open})$
$\dot{m}_{out} = f(PT - p_{out}, OutletValve_{open})$
4
$\varphi_{in} = f(\dot{m}_{in}, \chi_{in})$
$\varphi_{out} = f(\dot{m}_{out}, Pressure)$
Parameters
5
Extended Kalman filter
ρ_{est}
e_{est}
P_{est}
Boundary pressures
$p_{in}, \dot{m}_{in}, \chi_{in}$
$p_{out}, \dot{m}_{out}$
Valves opening
φ_{in}
PT
φ_{out}
LT
Q_{RF}
Q_{stat}

FIGURE 6 | Block diagram of the virtual sensor and its associated model schematic view. *f* designates different functions depending on the associated bloc number. *LT* and *PT* are, respectively, level and pressure transmitters. Numbers within green circles are explained in the text.

- 2: define model boundary pressures (p_{in} and p_{out}) using the closest available pressure transmitters. This is equivalent to calculating a pressure drop between the closest sensors and the model boundaries based on the current mass flows and temperatures.
- 3: calculate input ($\dot{m}_{in}$) and output ($\dot{m}_{out}$) mass flows through valves considering valve pressure drop, valves opening, and valves input quality (χ_{in}).
- 4: define phase separator input (φ_{in}) and output (φ_{out}) energy flux.
- 5: apply the extended Kalman filter algorithm using model parameters (i.e., valves coefficients, bath volume, and bath static heat loads) to estimate the heat loads dissipated in the phase separator.

In a nutshell, we use the difference between estimated values based on the model (i.e., e_{est} and ρ_{est}) and values (e and ρ) directly calculated from measurement (tabulated data in HEPACK), to correct the estimated heat load based on the model equations. The complete algorithm is being deployed in cryomodule PLCs. The following results were obtained using an external computer connected to the data acquisition system of the process. In that way, it was possible to directly get sensor process values but with a delay of few seconds.

4.3 Experimental Results

To evaluate the estimation capacity of the extended Kalman filter, a reference was needed. A controllable resistive heater thermally linked to the liquid helium phase separator was used for this purpose. Actual power dissipated in the helium bath[9] showed good agreement with the electrical power of the heater (within 1 Watt). Stepped variations of the heater power were used to test the synthesized virtual sensor. The results of **Figure 7** showed an exceptionally good prediction of the heater power. The actual dynamics precision was better than 2 Watts for heat loads lower than 20 Watts. The absolute average estimation error was equal to 0.7 *W* which represents about 3% of the maximal tested heat load. This means that the estimation precision could reach 1 *W* if the

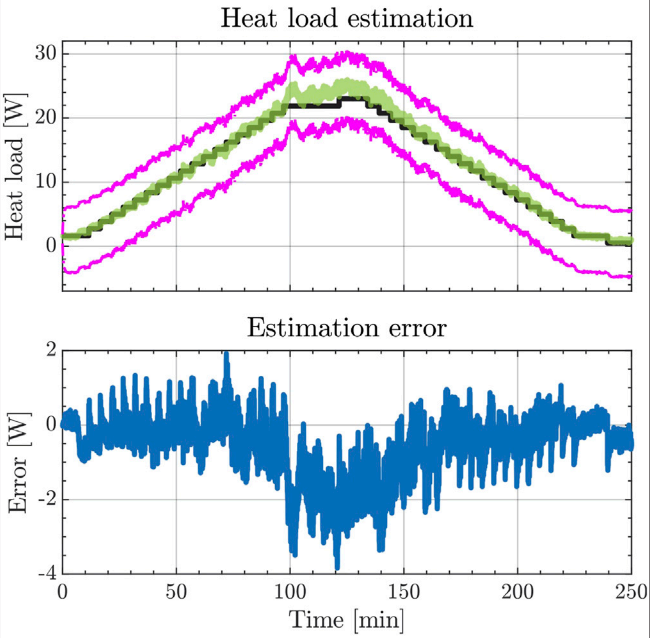


FIGURE 7 | Estimated heat loads (green) and uncertainty (magenta) vs. electrical heater setpoint (black).

[9] Measured by liquid helium level decay while the inlet valve is closed.

estimation is averaged on a sufficiently long timescale. Above 20 Watts, the method used to measure the actual heat load dissipation of the heater introduces a bias higher than 2 Watts which limits the interpretability of the results. However, the presented virtual sensor[10] shows its capability to predict heat load in real-time with a precision of few Watts. This is enough to detect anomalies during operation.

5 ANOMALY DETECTION

5.1 Problem Overview

"Anomaly detection" is used to designate algorithms capable of identifying events or items differing from the majority of the events/items. For the case of plant monitoring these algorithms could be used to address the problem of continuous fault detection on process actuators or transmitters. These kinds of algorithms are particularly suitable for large processes which contain thousands of actuators and transmitters because it is almost impossible for a single operator to continuously check the functioning of each element within the process. For example, in the SPIRAL2 cryogenic system, there are more than 70 control valves and 300 transmitters.

In such a case, we demonstrate the possibility to use the cryomodule model to perform actuator malfunction detection. To be more specific, we try to predict if the output value of one cryomodule (see **Figure 1**) is undergoing a deadband[11] problem. We use machine learning (ML) algorithms to predict the malfunction.

This section is decomposed into two parts. First, the generation of the dataset used to train ML algorithms is explained. Then, the ML algorithms themselves are described as well as their performances.

5.2 Dataset Creation

Nowadays, the main concern when working with ML algorithms is the generation of a clean dataset rather than the algorithm itself. Why so? Because it exists many libraries that already contain codes for all the commonly used ML algorithms. In Python, some of the most popular libraries are TensorFlow [22], PyTorch [23], and Skitlearn [24].

In the present work, the MATLAB statistics and ML toolbox [25] are used. All the data used for the anomaly detection problem are simulated data. Nevertheless, white noise has been added to each input and output of the model. The amplitude of this noise has been defined such that the simulated data look like the real measurements. Furthermore, slow fluctuations have been added to the input and output boundaries pressures in order to mimic the real operating conditions. To a naked eye, it is almost impossible to differentiate simulated data from measured data.

Before generating a dataset for valve anomaly detection, it is required to model the deadband problem on the output valve. In our case, the deadband has been set to random values between 1% and 4% to generate different test cases. The following signals have been recorded:

- phase separator pressure
- phase separator liquid level
- input and output valves command

Only the valve command (and not the real position) is considered. It mimics the case where valves are not equipped with a position indicator. In total, 500 time series of 60 s have been simulated. The dataset has been perfectly balanced: in half of the cases, the valve was subject to deadband, and in the other half it was not. For the two ML algorithms described in the next sub-sections, we used a standard cross-validation method. So, the overall dataset has been decomposed into a training set (60% of the data), a validation set (20% of the data), and a testing set (20% of the data). Thus we are able to perform hyper-parameters[12] tuning for each tested ML algorithm.

5.3 Solution 1: Classification Learner

The first solution is to use a classification learner to determine if a valve is faulty or not. This kind of algorithms require features as input and not time series. So, features were extracted from each time series of the dataset. As we do not know which features would be most suitable to identify a deadband problem, we calculate all the most common ones (i.e., variance, peak to peak, skewness, and kurtosis, etc.). In our case, we define 36 features which are few enough not to be concerned with limitations due to computer performances. But if it was the case, it would still be possible to use the same brute force approach and apply a principal component analysis [26,27] to reduce the number of features. Consequently, for all the time series of the dataset, each of the 4 measured signals has been transformed into a list of 9 features that could be used as input for a classification learner[13].

Once again, as there is no methodology to choose the best classification algorithm, we trained multiple ones and selected the one with the highest accuracy. Thanks to parallel computing it takes less than a few minutes to train multiple algorithms including decision trees, support vector machine (SVM), logistic regression, and nearest neighbors.

It appears that SVM with Gaussian kernel [28,29] gets the best performance among the other algorithms. SVM with Gaussian kernel is particularly suitable for our problem as we have a small number of features (less than 1,000) and not too much data to get concerned with the computation time issue.

The final results obtained with the SVM are given in **Table 2**. They are compared to the results obtained with another method: a deep neural network presented in the next section.

[10] The overall synthesis of this virtual sensor has been patented [18].

[11] A deadband is a range of input control that does not result in any output on the valve position.

[12] A hyper-parameter is a parameter whose value is set before the learning process begins. By contrast, the values of other parameters are derived *via* training.

[13] Weighted average of the precision and recall.

TABLE 2 | Performance index comparison between SVN and LSTM.

Metric	SVM	LSTM
Accuracy	0.98	0.93
Precision	0.97	0.88
Recall	0.99	0.98
F_1 score	0.98	0.93

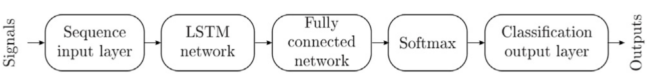


FIGURE 8 | Architecture used for the LSTM network.

5.4 Solution 2: Deep Network

The second idea while developing the valve anomaly detection consists of a Long Short Term Memory (LSTM) network [30]. The main advantage of this deep learning algorithm is the fact that time series signals could be directly used as network inputs. It means that there is no need to calculate features in that case. Nevertheless, it generally requires more data to train this kind of network than for an SVM.

The architecture of the LSTM network is given in **Figure 8**. As one can see, the network is decomposed into five layers:

- The sequence input layer used to sequence data to the network.
- The LSTM layer that learns long-term dependencies between time steps in sequence data.
- The fully connected layer that applies weight and bias to the LSTM output in order to predict the right label.
- The softmax layer that applies a SoftMax function to calculate the probability associated to each case (in our case normal operation of the deadband problem).
- The classification output layer that provides the final prediction depending on the probability calculated in the previous layer.

In total, it took 200 training epochs with a constant learning rate of 0.001 to train the network. This took less than 1 min of computation time.

5.5 Prediction Results

In this section, we compare the performances of the synthesized SVM and the LSTM algorithm. The comparison is based upon usual ML metrics: accuracy, precision, recall, and F1 score. More details about those metrics are available in [31]. Comparison is performed on a test set of 100 time series used only for this purpose (and not for training). Results are given in **Table 2**.

As one can see, both SVM and LSTM algorithms show good results in terms of error predictions. Nevertheless, the SVM results are slightly better. As the implementation complexity of those two algorithms is quite similar, the best option would be to deploy an SVM algorithm on the system to get an online anomaly detector. It is worth mentioning that anomaly detection has been tested on the cryomodules only to remain consistent with the rest of the article. Nevertheless, it would be more interesting to generate an anomaly detector for process-critical elements such as rotating machines of the cryogenic system: the turbines and the compressors.

6 CONCLUSION

Advanced operation and diagnostics tools are slowly becoming a vital part of the operation of large infrastructures such as particle accelerators. While sub-systems such as cryogenics are not usually studied and documented in that perspective, they can be critical in maintaining a high beam availability. Working on a twin model of the cryomodules opens a gate that allows a control and operation method that otherwise would be difficult. It also introduces the possibility to use machine learning techniques for synthesizing monitoring proxies and smart fault detection observers. The premises of such observers have been studied and is in the process of being implemented in the SPIRAL2 control system. Developed modeling allows us to generate training data sets for machine learning algorithms. Future work will include the extension of the thermodynamic models to its radio-frequency counter-part and the application of SVM-based algorithms on actual machine data.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

REFERENCES

1. Ferdinand R, Junquera T, Bosland P, Bernaudin P, Saugnac H, Olry G, et al. The Spiral2 Superconducting Linac. In: Proceedings of LINAC 2008; Victoria, France (2008).
2. Padamsee H. *RF Superconductivity: Science, Technology, and Applications*. Hoboken, NJ, USA: Wiley VCH (2009). doi:10.1002/9783527627172
3. Ghribi A, Bernaudin P-E, Vassal A, Bonne F. Status of the SPIRAL 2 LINAC Cryogenic System. *Cryogenics* (2017) 85(Suppl. C):44–50. doi:10.1016/j.cryogenics.2017.05.003
4. Ghribi A, Bernaudin P-E, Bert Y, Commeaux C, Houeto M, Lescalié G. Spiral 2 Cryogenic System for the Superconducting LINAC. *IOP Conf Ser Mater Sci Eng* (2017) 171:012115. doi:10.1088/1757-899X/171/1/012115

5. Bernaudin P, Bosland P, Chel S, de Girolamo P, Devanz G, Hardy P, et al. Design of the Low-Beta, Quarter-Wave Resonator and its Cryomodule for the Spiral2 Project. *EPAC* (2004) 6:1276.
6. Olry G, Biarrotte J-L, Blivet S, Bousson S, Commeaux C, Joly C, et al. Development of a Beta 0.12, 88MHz, Quarter-Wave Resonator and its Cryomodule for the SPIRAL2 Project. *Physica C: Superconductivity* (2006) 441(1-2):197–200. doi:10.1016/j.physc.2006.03.030
7. ISA. *ANSI/ISA-75.01.01 Industrial-Process Control Valves*. Gurugram, India: ISA (2012).
8. Bonne F, Varin S, Vassal A, Bonnay P, Hoa C, Millet F, et al. Simcryogenics: a Library to Simulate and Optimize Cryoplant and Cryodistribution Dynamics. *IOP Conf Ser Mater Sci Eng* (2020) 755:012076. doi:10.1088/1757-899x/755/1/012076
9. Vassal A. *Etude D'un Réseau Cryogénique Multi-Clients Pour SPIRAL2*. PhD thesis (2019). Available from: https://tel.archives-ouvertes.fr/tel-02390264v2.
10. Vassal A, Bernaudin P-E, Bonnay P, Bonne F, Ghribi A, Millet F, et al. Spiral2 Cryogenic System Thermodynamic Behavior Prediction through Dynamic Modeling. In: Proc. 29th Linear Accelerator Conference (LINAC'18); 16-21 September 2018; Beijing, China (2018). p. 366–9.
11. Vassal A, Bonne F, Ghribi A, Millet F, Bonnay P, Bernaudin P-E. Dynamic Modeling and Control of the Spiral2 Cryomodules. *IOP Conf Ser Mater Sci Eng* (2019) 502:012111. doi:10.1088/1757-899x/502/1/012111
12. Matworks. *Control System Toolbox: User's Guide (R2019a)* (2019).
13. Bonne F. *Modélisation et contrôle des grands réfrigérateurs cryogéniques*. PhD thesis. Grenoble: HAL open science (2014). Available from: http://www.theses.fr/2014GRENT094/document.
14. Luenberger D. An Introduction to Observers. *IEEE Trans Automat Contr* (1971) 16(6):596–602. doi:10.1109/tac.1971.1099826
15. Wipf SL. AC Losses in Superconductors. *J Appl Phys* (1968) 39(6):2538. doi:10.1063/1.1656612
16. Ding X, Boucher S, Technologies R. A Method for Establishing Q-Factors of Rf Cavities ∗. In: IPAC; Kyoto (2010). p. 3789–91.
17. Kalman RE. A New Approach to Linear Filtering and Prediction Problems. *Trans ASME* (1960) 82(D):35–45. doi:10.1115/1.3662552
18. Vassal A, Bonne F, Bonnay P, Ghribi A. *Procédé de détermination d'un facteur quailté d'une cavité accélératrice d'un accélérateur de partricules* (2018).
19. Ghribi A, Aburas M, Baumont Y, Bernaudin P-E, Bonneau S, Duteil G, et al. First Full Cool Down of the Spiral 2 Superconducting Linac. *Cryogenics* (2020) 110:103126. doi:10.1016/j.cryogenics.2020.103126
20. McGee LA, Schmidt SF, Mcgee LA, Sc SF. "Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry." in *Industry," National Aeronautics and Space Administration, Ames Research*. Citeseer (1985).
21. Torres L, Jiménez-Cabas J, González O, Molina L, López-Estrada F-R. Kalman Filters for Leak Diagnosis in Pipelines: Brief History and Future Research. *J Mar Sci Eng* (2020) 8(3):173. doi:10.3390/jmse8030173
22. Martin A, Ashish A, Paul B, Eugene B, Zhifeng C, Craig C. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems* (2015). Available from: https://www.tensorflow.org/.
23. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. Pytorch: An Imperative Style, High-Performance Deep Learning Library. *Adv Neural Inf Process Syst* (2019) 32.
24. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in \{P\}ython. *J Mach Learn Res* (2011) 12:2825–30.
25. The MathWorks I. *MATLAB Statistics and Machine Learning Toolbox Release 2018b* (2018). Available at: https://fr.mathworks.com/help/stats/.
26. Pearson K. LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *Lond Edinb Dublin Philos Mag J Sci* (1901) 2(11):559–72. doi:10.1080/14786440109462720
27. Hotelling H. Analysis of a Complex of Statistical Variables into Principal Components. *J Educ Psychol* (1933) 24(6):417–41. doi:10.1037/h0071325
28. Cortes C, Vapnik V. Support-vector Networks. *Mach Learn* (1995) 20(3):273–97. doi:10.1007/bf00994018
29. Cristianini N, Shawe-Taylor J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge: Cambridge University Press (2000). doi:10.1017/CBO9780511801389
30. Hochreiter S, Schmidhuber J. Long Short-Term Memory. *Neural Comput* (1997) 9(8):1735–80. doi:10.1162/neco.1997.9.8.1735
31. Shung KP. *Accuracy, Precision, Recall, or F1*. Canada: Towards data science (2018).

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 20 September 2022
DOI 10.3389/fphy.2022.957128

Check for updates

OPEN ACCESS

EDITED BY
Alexander Scheinker,
Los Alamos National Laboratory (DOE),
United States

REVIEWED BY
Nhan Tran,
Fermi National Accelerator Laboratory (DOE), United States
Saurabh Kulkarni,
Graphcore, United States

*CORRESPONDENCE
Berthié Gouin-Ferland,
berthie.gouin-ferland@usherbrooke.ca

SPECIALTY SECTION
This article was submitted to Interdisciplinary Physics, a section of the journal Frontiers in Physics

RECEIVED 30 May 2022
ACCEPTED 29 August 2022
PUBLISHED 20 September 2022

CITATION
Gouin-Ferland B, Coffee R and Therrien AC (2022), Data reduction through optimized scalar quantization for more compact neural networks.
Front. Phys. 10:957128.
doi: 10.3389/fphy.2022.957128

Data reduction through optimized scalar quantization for more compact neural networks

Berthié Gouin-Ferland[1]*, Ryan Coffee[2] and Audrey C. Therrien[1]

[1]Interdisciplinary Institute for Technological Innovation - 3IT, Sherbrooke, QC, Canada, [2]SLAC National Accelerator Laboratory, Menlo Park, CA, United States

Raw data generation for several existing and planned large physics experiments now exceeds TB/s rates, generating untenable data sets in very little time. Those data often demonstrate high dimensionality while containing limited information. Meanwhile, Machine Learning algorithms are now becoming an essential part of data processing and data analysis. Those algorithms can be used offline for post processing and post data analysis, or they can be used online for real time processing providing ultra low latency experiment monitoring. Both use cases would benefit from data throughput reduction while preserving relevant information: one by reducing the offline storage requirements by several orders of magnitude and the other by allowing ultra fast online inferencing with low complexity Machine Learning models. Moreover, reducing the data source throughput also reduces material cost, power and data management requirements. In this work we demonstrate optimized nonuniform scalar quantization for data source reduction. This data reduction allows lower dimensional representations while preserving the relevant information of the data, thus enabling high accuracy Tiny Machine Learning classifier models for online fast inferences. We demonstrate this approach with an initial proof of concept targeting the CookieBox, an array of electron spectrometers used for angular streaking, that was developed for LCLS-II as an online beam diagnostic tool. We used the Lloyd-Max algorithm with the CookieBox dataset to design an optimized nonuniform scalar quantizer. Optimized quantization lets us reduce input data volume by 69% with no significant impact on inference accuracy. When we tolerate a 2% loss on inference accuracy, we achieved 81% of input data reduction. Finally, the change from a 7-bit to a 3-bit input data quantization reduces our neural network size by 38%.

KEYWORDS

machine learning - ML, neural network, quantization, classification, free electron lasers, data acquisition, ultra high rate detectors

1 Introduction

Detectors for large physics and light source experiments now produce data much faster than acquisition systems can collect, triage and store it [1, 2]. The current approach of saving all raw data requires a large amount of cabling, power and downstream storage, beyond what the architecture or budget can allow [1]. Thus, several current and planned experiments would benefit from data reduction at the source. Furthermore, the initial data preparation steps before analysis tend to be very similar over time - deletion of invalid events, baseline corrections and initial information extraction, such as calculating timestamps or energy. Moving these steps at the edge–near the detector–would reduce data at the source and thus lightening the load of the high speed communication system and high-speed storage. Even so, several of this initial analysis requires complex mathematical operations which require many sequential steps, an iterative approach, and significant computational resources. This limits the capacity for true real-time data reduction [3, 4].

One strategy exploits ultra low latency Edge Machine Learning (edgeML); the deployment of inference models near the detector capable of real-time analysis, veto and compression of incoming data. Machine Learning (ML) models like neural networks (NN) can be trained to emulate arbitrary mathematical operations while using simpler addition and multiplication operations that can be greatly accelerated using appropriate hardware [5]. This strategy of moving much of the data preparation steps at the source enables to reduce both data velocity and data volume, resulting in resource savings in term of data transfer, processing and storage.

The LCLS-II built at SLAC National Accelerator Laboratory is capable of generating coherent x-ray shots at a 1 MHz rate [1, 6]. The experimental hutches host several dozen different instruments to capture the maximum information about each event. However, the system must be run at a lower rate to collect the data from all these instruments and send it to disk [1]. To achieve continuous full rate experiments, a first proof of concept targeting the Cookiebox detector demonstrated that deploying ML inference models on FPGA can reduce data velocity in real-time [7].

The Cookiebox is a diagnostic detector which non-destructively samples each x-ray shot to reconstruct the single shot time–energy profile via the method of attosecond angular streaking [3]. The reconstructions are to be used to select which x-ray shots fit particular experimental objectives, rejecting invalid shots, aggregating simple reference shots, or reserving complicated shots for deeper covariance-based analysis. Such a streaming shot evaluation system significantly reduces the raw data rate from other instruments before it is written to persistent storage. However, to achieve this, each x-ray shot must be analyzed with very low latency, within about 100 μs, to avoid overly large raw data ring buffers. Such low-latency capability of edgeML has been demonstrated [7] and further work is ongoing to provide a fully working system. The Cookiebox detector produces a large volume of data, on the order of 100's of GB/s, which itself becomes a challenge when designing low complexity ML algorithms suited for limited capacity edgeML accelerators. For that reason, the compression and analysis must be distributed all along the data path, including prior to the ML algorithms. In this work, we suggest to optimally quantize the Cookiebox data before feeding it to our NN inference model. This compression strategy reduces throughput while preserving relevant information which enabled for leaner and more accurate NNs.

The Materials and Methods section begins with an overview of the CookieBox diagnostic detector, followed by a description of the quantization algorithm and the NN developed for the CookieBox. We then present the results for both the optimal quantizer model and for the ML model. Finally, we conclude with a discussion of how the quantization impacts the data and the ML model.

2 Materials and methods

2.1 Cookiebox

The diagnostic detector that we take as our demonstration use case is an attosecond angular streaking instrument composed of an array of 16 electron time-of-flight (ToF) spectrometers, illustrated in Figure 1 [8]. The spectrometers are placed on a plane perpendicular to the x-ray propagation. A micrometer wavelength infra-red laser with a circular polarization modulates the central electrical field with a period of 10–30 fs [3]. A low pressure gas is present in the center chamber. When the atoms or molecules are hit by x-rays, their electrons are ejected and collected by the electron spectrometers. This instrument can measure the polarization and the time-energy spectrum of each individual x-ray shot produced by LCLS-II (diagnostic mode), or be used to measure numerous features associated with a given target atom or molecule (experimental mode).

Each spectrometer signal is fed to a 12 bit, 6.4 GS/s analog-to-digital converter. Thus, the total instrument generates data at a rate of 1.229 Tb/s. The first data reduction step consists of identifying the time at which electrons hit the spectrometer using a peak finding algorithm. The digitized signal is thus converted into timestamps corresponding to each electron "hit". For each x-ray shot, each spectrometer collects approximately 100 hits, which are converted into 16 bit timestamps, which results in a data rate of 26 Gb/s. In experimental mode, this data is collected, however in diagnostic mode, the data must be analyzed within 100 μs to select the correct processing for each shot while avoiding large rapid memory buffers.

We choose this detector since it has recently been shown compatible with both the signal rates and the energy resolution

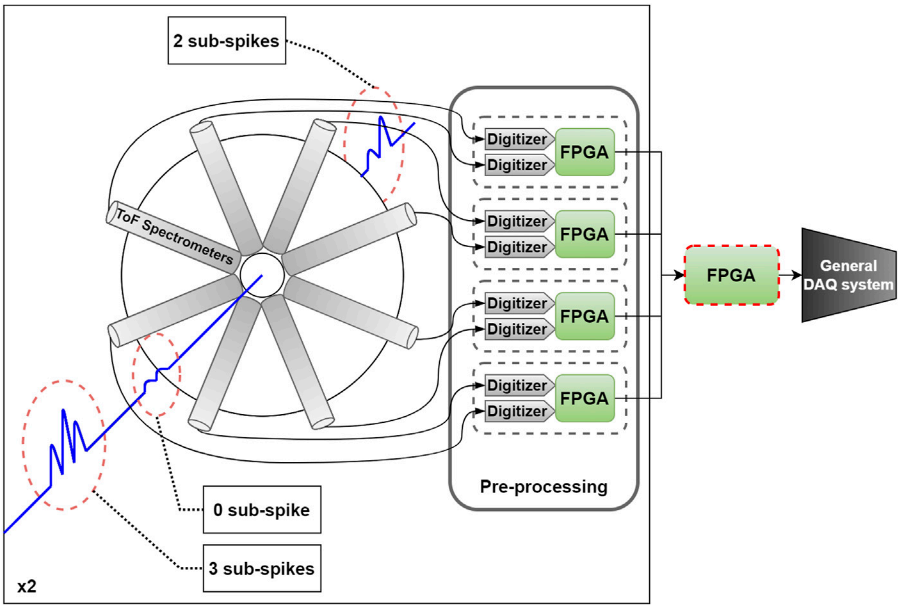


FIGURE 1
Diagram of the data flow of the CookieBox. The ToF spectrometers signals are processed by analog electronics before being digitized. Each 2 digitizers feed into an FPGA that will be hosting the neural network discussed in this article. The central FPGA, circled with a red dotted line, collects the extracted features from all 8 peripheral FPGA and forwards the inference results to the rest of the DAQ system.

required for the x-ray pulse reconstruction algorithms. The facility wide interest at LCLS-II has further motivated that such an instrument provide continuous streaming diagnoses of x-ray pulses, both time-energy distribution as well as polarization, at the full MHz repetition rate of the facility. As such, this instrument is now capable of working with the full data rate as soon as the LCLS-II ramps up to the highest quasi-continuous rate to provide offline diagnostic. The present work aims to move the diagnostic capability on the edge and in real-time.

2.2 Dataset (CookieSimSlim)

To create the initial training datasets, we use a simple simulation [9] to generate data via Monte-Carlo simulation of attosecond angular streaking. The simulation begins with a Poissonian choice of so-called self-amplified spontaneous emission (SASE) sub-spikes, forming the x-ray shot, each with an energy consistent with the few % SASE bandwidth of the FEL process and a relative temporal delay that is chosen as an even random choice across the 2π period of the angular optical cycle.

The period of the optical cycle is chosen experimentally by the choice of dressing optical laser field and is typically in the regime of few femtoseconds total period for addressing SASE structures as targeted here [3]. The data generator also allows for sub-spike polarization variation such that our model is fully compatible with the recent developments in time-dependent polarization shaping in SASE FELs [10].

The resulting dataset from the CookieSimSlim generator is an HDF5 formatted tree of events, or “x-ray shots”, each with a list of electron hit energies (Xhits) for each detector angle. This list of hit energies itself is a sampling from a smooth probability distribution Y_{pdf} that is the sum of the Gaussian energy distributions for each of the sub-spikes (offset *via* $\kappa \sin(\phi)$ where κ is the streaking kick strength and $\phi \in [0, 2\pi)$ is the random phase associated with the sub-spike relative arrival timing. The shot-dependent parameters such as kick strength, phase, dark-count rate, SASE width and so on are all produced as attributes of the particular shot in the HDF5 file. For convenience the output file also includes an “image” representation of the energy hit histogram X_{img}.

The hit energies in X_{hits} are represented as 32 bit floats from the generator. This bit depth is considered a “conventional” representation since in the experiment, the data will be represented as an energy conversion of an integer arrival time that is typically only of 16 bit resolution. Allowing 32 bit floating precision for the energy mapping result is therefore considered a convenient precision for sake of the arithmetic in producing that calibrated energy for each hit.

2.3 Quantization

All of the information on the x-ray shot is contained in the timing of the electron hits; that is the 16 bit timestamps obtained after the pulse finding step. With CookieSimSlim, this data is provided in a 32 bit float format. We encode this large set of data to a small set of optimized values modelled on the probability distribution function (PDF) of the source detector with a nonuniform scalar quantizer [11, 12].

The mean-square error (MSE) is used to judge of the quality of the quantization. The MSE is obtained with Eq. 1:

$$\mathrm{MSE}(Y, Q(Y)) = \frac{1}{N}\sum_{i=0}^{N-1}\left(y_i - Q(y_i)\right)^2 \tag{1}$$

where Y is the original discrete dataset and $Q(Y)$ is the quantized dataset.

The first order entropy of the datasets is used to mesure the amount of information it contains. The first order entropy is obtained with Eq. 2:

$$\mathrm{H}(Y) = -\sum_{i=0}^{N-1} P(y_i)\log_2 P(y_i) \tag{2}$$

where Y a discrete variable, that represent our dataset or quantized dataset, with possible outcomes $y_0 \ldots y_{N-1}$ which occur with probability $P(y_0) \ldots P(y_{N-1})$ [13]. The base two of the logarithm function is to gives the entropy in bits.

2.4 The neural networks

The data quantization allows for a much reduced input size for the convolutional neural network (CNN) which contributes to its size reduction. The CNN type of architecture is used to reduce the impact of the input data dimensionality on the model dimensionality itself. We also design our model to be the smallest as possible. For that, we use strategies inspired from the SqueezeNet CNN architecture [14]. However, since our CNN only has three convolution layers, our strategies boil down to using as few and as small as possible filters. We only use 3 × 3 filters which is the smallest kernel size to capture the notion of relative dependencies in all direction within a 2D space. We gradually double the numbers of filters between each convolution layer from the beginning to the end of the network like in the VGG model [15]. We use 10 filters in the first convolution layer allowing for below or close to 10,000 network parameters while potentiating the accuracy. The two last layers of the CNN are fully connected layer with 5 neurons each. For all layers, except the last one, the activation function is the rectified linear unit (ReLU) activation function. For the last layer, the Softmax activation function is used for classification.

A specific CNN is dedicated for each corresponding bit depth because of the model input size changes according to the number of bit used for the quantization. Except the input size, all the other configurations are the same for all CNNs. Figure 4 shows examples of the CNN input heatmap images in regards of the number of quantization levels. In this example, each input heatmap images (Figures 4A–C) requires a specific CNN.

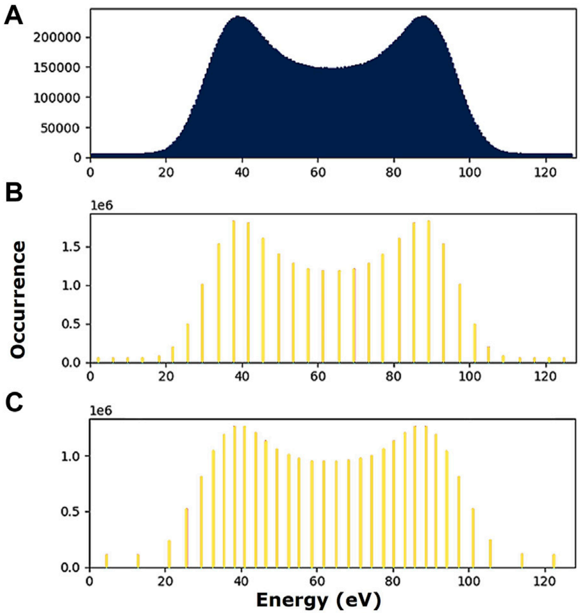


FIGURE 2
5-bit uniform quantization **(B)** and 5-bit optimized nonuniform quantization **(C)** effect on the original data distribution **(A)**. The Lloyd-Max algorithm gives more granularity to the ranges where the data are more occurring within the distribution which maximizes the entropy and minimizes the mean-square error when quantizing.

3 Results

3.1 Quantization effect on dataset

Uniform quantization allows for a quick and simple design. However, optimized nonuniform quantization minimizes the MSE, but also requires to train the quantizer beforehand. For comparison, the quantization is done using a uniform quantizer and a PDF-optimized nonuniform quantizer. In both cases, we quantized to obtain 5 strategic and realistic bit depths from 3 to 7 bits. This yields M quantization levels with $M = 2^n$ and n the bit depth.

The quantization levels for the uniform quantizer are uniformly placed within the distribution. For training the PDF-optimized nonuniform quantizer, we used the Lloyd-Max (LM) algorithm [11, 12]. The LM algorithm uses an iterative k-means clustering approach to determine which quantization level locations minimize the MSE. The initial estimate for the quantization levels are uniformly placed within the distribution. The tolerance for change in MSE after which the LM algorithm converged is set to $1e-5$. For both quantizer designs, the actual quantization is done by mapping the input value to its nearest quantization levels. Figure 2A shows the original data distribution while Figures 2B,C respectively show the distribution with uniform quantization and optimized nonuniform quantization.

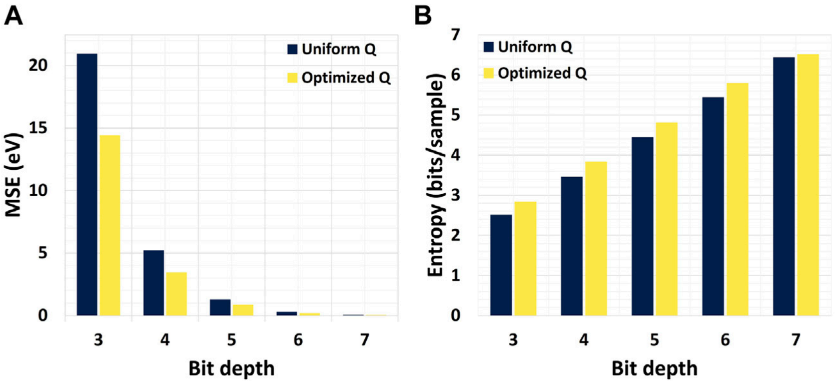


FIGURE 3
Mean-square error **(A)** and entropy **(B)** as a function of the quantization. We interpret low mean-square error with high entropy as a better information representation within data.

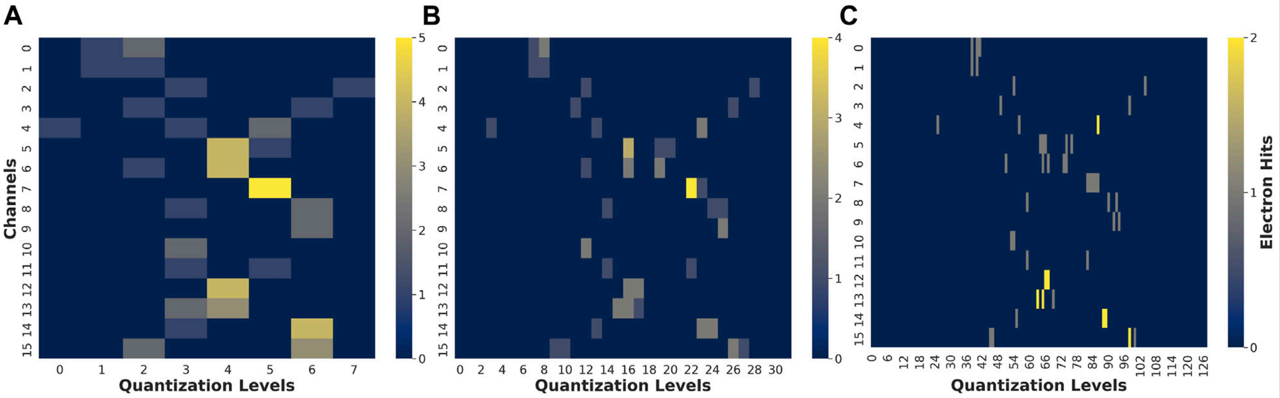


FIGURE 4
3, 5 and 7 bits quantized data heatmaps for a 2 pulses event [respectively **(A–C)**]. For a human eye, the 5 bits quantized data heatmaps **(B)** makes the two pulses distinction less ambiguous over 3 bits quantized data heatmaps **(A)**. However, the 7 bits quantized data heatmaps does not improve the distinction for a human eye.

Figure 3A shows how nonuniform quantization minimized the MSE compared to a uniform quantization. In addition, Figure 3B shows how nonuniform quantization also maximized the entropy. The data are quantized using the uniform and nonuniform quantizer and then converted as an heatmap with the bin intervals being the quantization levels. This result is an input heatmap image of size 16 ×M and it is the input of the CNN. Each pulse within an x-ray shot will create a vertical sinusoidal wave where the relative phase between waves reflects the time interval between the pulses. Figure 4 shows how quantizing with 5 bits over 3 bits makes the pulse count less ambiguous, but also how quantizing with 7 bits does not drastically simplify the pulse counting task (for a human eye).

3.2 Classification accuracy and model size

We trained the CNNs to classify the pulse count in every x-ray shot event (i.e. heatmap image). A unique and dedicated CNN is trained for each bit depth and corresponding quantized heatmap image size. This is because the quantized heatmap image size determine the input size of the CNN which then impact the overall CNN dimensionality. However, all the model parameters (kernel size, number of filters. . .) and initial weights are kept steady for all CNNs.

The desired pulse count per x-ray shot may change between LCLS-II experiments. For that reason, we trained the CNNs on a local GPU (RTX3090) to classify 0, 1, 2, 3 and “many” pulses for every shots. The “many” class correspond to all events with

TABLE 1 CNN training Configurations.

Loss Function	Sparse Categorical Crossentropy
Optimizer	Adam
Learning Rate	0.001
Batch Size	2042
Validation Split	0.2

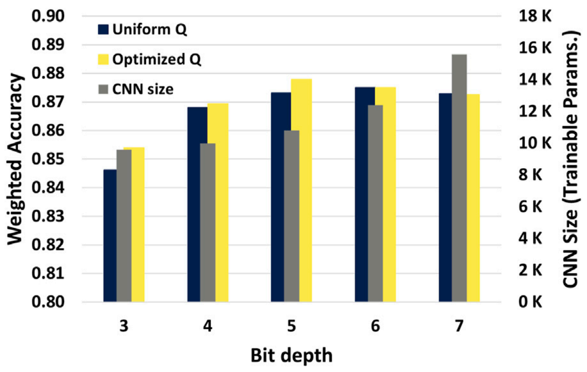


FIGURE 5
CNN weighted accuracy and size. Grey bars read on the right axis and the blue and yellow bars on the left axis. The model shows higher accuracy with 5-bit depth over 7-bit depth while requiring fewer model parameters.

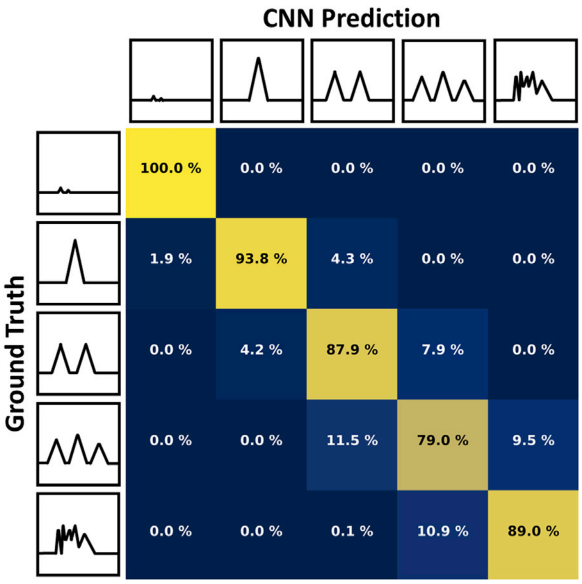


FIGURE 6
Confusion matrix showing the performance of the 5-bit dedicated CNN in predicting the number of pulses (0, 1, 2, 3 or “Many”) in a single x-ray shot with data optimally quantized using 5 bits. A perfect CNN would have 100% (yellow) in every box on the diagonal and 0% (dark blue) everywhere else.

4 pulses or more, which have little value in most experiments and are generally rejected. The training setup is described in Table 1. The training set includes 400,000 events and the test set 100,000 events. All the data are generated using the CookieSimSlim generator. Figure 5 shows the relation between the CNN weighted prediction accuracy as well as the CNN size when applying the method in simulation with the test set. We see that this optimized quantization scheme allows for data reduction on 5 bits while allowing for more accurate and leaner inference models then when using 7 bits. Figure 6 shows the confusion matrix of the 5-bit dedicated CNN in predicting the number of pulses.

4 Discussion

4.1 Quantization

Our goal for using optimized nonuniform quantization over uniform quantization was to maximize the information representation of the Cookiebox source on a lower bit budget. This is what Figure 3 suggest. We saw that while a nonuniform quantization minimized the MSE compare to a uniform quantization, it also maximized the entropy which represent the amount of information carried out by the data. Nonetheless, we also saw that the gap in performance tends to shrink for a larger bit depth. This is because the optimized quantization levels converge towards a uniform distribution as more levels are created within the same limited interval. We noticed that this nonuniform quantization is sensitive to changes in the data distribution; if the source statistic changes overtime a mismatch effect could occur and change the quantizer performance. We recommend training the nonuniform quantizer on a dataset that includes those variations or to include a calibration step to train the quantizer before the data acquisition to ensure the quantizer representativity. Note that the same mismatch effect would occur to a standalone NN (i.e. without the prior optimized nonuniform quantization).

Nevertheless, quantization allows to pass from a 16-bit scalar data representation to a 7, 5 and even 3-bit representation. This yields a data reduction of 56%, 69 % and 81%. Even if quantization is a lossy coding, let’s recall that optimized nonuniform quantization allows for data reduction while maximizing the information retention. This method avoids the computational load of lossless coding, which reduces the acquisition system latency and overall resources usage.

4.2 Convolutional neural network inference model

The goal of limiting the Cookiebox data dimensionality was also to reduce the input dimensionality for our inference model. Fully connected NNs are really sensitive to their input size and tend to become very large if not contained. We used a simple CNN architecture to minimize this effect, but Figure 5 still shows the benefits of a small input size in term of model dimensionality. For instance, with a straightforward 16-bit representation and the same architecture, this CNN would require approximately 2.6 million parameters. By contrast, our 7, 5 and 3-bit representations shrunk the CNN size to 15,595, 10,795 and 9595 parameters respectively, a reduction of two orders of magnitude compared to the 16 bit input model. Within these smaller models, the change from a 7-bit to a 5-bit input data quantization reduces the CNN size by 31%, with no significant impact on inference accuracy. When we tolerate a 2% loss, the change from a 7-bit to a 3-bit input data quantization reduces the CNN size by 38%. Note that the CNN size reduction is only due to the input dimensionality reduction and that no optimization (weight pruning, weight quantization. . .) is done on the model itself.

We saw small improvements in accuracy between uniform and nonuniform quantization in Figure 5, but as for the MSE and entropy, the difference tend to plateau beyond 6 bits. Our simulation data exhibit bimodality within the distribution for all 16 channels dimensions and we expect a better gain of nonuniform over uniform quantization for more complex multimodal distributions. If the dimensions exhibit different distributions, we recommend to train and quantized in respect to each dimensions. With that said, our model still demonstrates better performances then the first iteration of NN that tackled the Cookiebox problem while being almost two orders of magnitude smaller [7].

The drop in accuracy from 6 to 7 bits correlates with a significant input size growth. Because the number of filters and kernel size are constant for all bit depth, it limits the learning potential of the CNN when having larger and more complex input. A solution to that is to use a bigger CNN. However, we would also need a bigger dataset to maintain the model generalization ability. Because our goal was to designed a small NN and to reduce data generation, we do not consider going bigger and deeper a viable, sustainable and elegant solution for edgeML.

Finally, we used scalar quantization as a proof of concept, but the next step is to use vector quantization with the Linde-Buzo-Gray algorithm to compress even more multidimensional data while conserving relevant information [16]. This could be even more promising for data source reduction and for smaller edgeML models.

5 Conclusion

Large physics experiments now produce more and more data at an ever-increasing throughput. Simultaneously, ML is becoming more popular among the community for its ability to model complex systems and the growing ML hardware accessibility. In addition to that, edgeML is a promising tool for large science experiment online data reduction. However, edgeML applications face challenges in terms of power efficiency and for hardware implementation. Moreover, some applications like the Cookiebox diagnostic detector require ultra low-latency inference.

In this work, we combined optimized data quantization with the generalization capacities of NN to reduce data source throughput while preserving relevant information and thus reducing material cost, power and data management requirements. This approach also enables smaller NN for fast real-time–on the edge–inferencing. The real-time diagnostic function would be a huge boon for upcoming LCLS-II experiments.

Beyond energy efficiency for data management system, it is worth mentioning that the cheapest data is the data which is never generated. The Jevons paradox showed us that in many technological area, increasing a process efficiency only tend to rise its absolute usage. That is already something addressed by the communication technology community [17]. The scientific community now have a real opportunity to save in development, infrastructure and energy cost by using previously developed models directly at the source to generate as much useful information and less data.

With the development larger and faster detectors planned over the next decades in several disciplines such a medical imaging, particle physics and quantum computing, the data velocity problem will not go away. There is no universal approach; each application presents a different set of challenges. Yet, edgeML is a powerful tool that can take advantage of the inherent structure of many data types which makes it a perfect candidate for real-time data reduction in many fields.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

BGF, RC, and AT contributed to conception and design of the study. RC designed the simulation and BGF adapted it for manuscript. BGF performed the results analysis supervised by AT. BGF wrote the first draft of the manuscript. RC wrote sections of the manuscript about the CookieBox detector. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding

This research was undertaken, in part, thanks to funding from the Canada Research Chairs Program. AT hold the Canada Research Chair in Real-Time Intelligence Embedded for High-Speed Sensors, which funded the current work, materials and personnel, including BGF. RC thanks the Department of Energy, Office of Science, Office of Basic Energy Sciences for funding the development of the detector array itself under Grant Number FWP 100498 “Enabling long wavelength Streaking for Attosecond X-ray Science” and for funding Field Work Proposal 100643 “Actionable Information from Sensor to Data Center” for the development of CookieNetAE as well as the associated algorithmic methods, EdgeML computing hardware, and personnel. RC also acknowledges synergistic funding for the computational method development by the Office of Fusion Energy Science under Field Work Proposal 100636 “Machine Learning for Real-time Fusion Plasma Behavior Prediction and Manipulation”.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. Thayer JB, Carini G, Kroeger W, O’Grady C, Perazzo A, Shankar M, et al. Building a data system for lcls-ii. In: 2017 IEEE Nuclear Science Symposium and Medical Imaging Conference, NSS/MIC 2017 - Conference Proceedings. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers Inc. (2018). p. 1–4. doi:10.1109/NSSMIC.2017.8533033

2. Guglielmo GD, Fahim F, Herwig C, Valentin MB, Duarte J, Gingu C, et al. A reconfigurable neural network ASIC for detector front-end data compression at the HL-LHC. *IEEE Trans Nucl Sci* (2021) 68:2179–86. doi:10.1109/TNS.2021.3087100

3. Hartmann N, Hartmann G, Heider R, Wagner MS, Ilchen M, Buck J, et al. Attosecond time–energy structure of x-ray free-electron laser pulses. *Nat Photon* (2018) 12:215–20. doi:10.1038/s41566-018-0107-6

4. Li S, Guo Z, Coffee RN, Hegazy K, Huang Z, Natan A, et al. Characterizing isolated attosecond pulses with angular streaking. *Opt Express* (2018) 26:4531. doi:10.1364/OE.26.004531

5. Duarte J, Han S, Harris P, Jindariani S, Kreinar E, Kreis B, et al. Fast inference of deep neural networks in FPGAs for particle physics. *J Instrum* (2018) 13:P07027. doi:10.1088/1748-0221/13/07/P07027

6. Schoenlein R. *New science opportunities enabled by lcls-ii x-ray lasers*. SLAC Report SLAC-R-1053 (2015). p. 1–189.

7. Therrien AC, Herbst R, Quijano O, Gatton A, Coffee R. Machine learning at the edge for ultra high rate detectors. In: 2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC) (2019). p. 1–4. doi:10.1109/NSS/MIC42101.2019.9059671

8. Walter P, Kamalov A, Gatton A, Driver T, Bhogadi D, Castagna JC, et al. Multi-resolution electron spectrometer array for future free-electron laser experiments. *J Synchrotron Radiat* (2021) 28:1364–76. doi:10.1107/S1600577521007700

9. [Dataset] Coffee RN. *Cookiesimslim: A simple simulation and data generator that approximates attosecond x-ray angular streaking results for lcsl-ii algorithm development* (2022).

10. Sudar N, Coffee R, Hemsing E. Coherent x rays with tunable time-dependent polarization. *Phys Rev Accel Beams* (2020) 23:120701. doi:10.1103/PhysRevAccelBeams.23.120701

11. Lloyd S. Least squares quantization in pcm. *IEEE Trans Inf Theor* (1982) 28: 129–37. doi:10.1109/TIT.1982.1056489

12. Max J. Quantizing for minimum distortion. *IEEE Trans Inf Theor* (1960) 6: 7–12. doi:10.1109/TIT.1960.1057548

13. Sayood K. *Introduction to data compression, fourth edition*. 4th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. (2012).

14. Gholami A, Kwon K, Wu B, Tai Z, Yue X, Jin P, et al. SqueezeNext: Hardware-Aware neural network design. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2018). p. 1719–171909. doi:10.1109/CVPRW.2018.00215

15. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings (2015). p. 1–14. Cited By 15336.

16. Linde Y, Buzo A, Gray R. An algorithm for vector quantizer design. *IEEE Trans Commun* (1980) 28:84–95. doi:10.1109/TCOM.1980.1094577

17. Liao HT, Chen KS. *Mapping the landscape of green communications and green computing: A review based on bibliometric analysis*. In: 2021 IEEE 21st International Conference on Communication Technology (ICCT) (2021). p. 565–9. doi:10.1109/ICCT52962.2021.9658091

TYPE Original Research
PUBLISHED 03 October 2022
DOI 10.3389/fphy.2022.967645

OPEN ACCESS

EDITED BY
Alexander Scheinker,
Los Alamos National Laboratory (DOE),
United States

REVIEWED BY
Kehui Sun,
Central South University, China
Alexander Vladimirovich Bogdanov,
Saint Petersburg State University, Russia

*CORRESPONDENCE
Kazuhiro Fujita,
kfujita@sit.ac.jp

SPECIALTY SECTION
This article was submitted to
Interdisciplinary Physics,
a section of the journal
Frontiers in Physics

RECEIVED 13 June 2022
ACCEPTED 24 August 2022
PUBLISHED 03 October 2022

CITATION
Fujita K (2022), Electromagnetic field computation of multilayer vacuum chambers with physics-informed neural networks.
Front. Phys. 10:967645.
doi: 10.3389/fphy.2022.967645

Electromagnetic field computation of multilayer vacuum chambers with physics-informed neural networks

Kazuhiro Fujita*

Department of Information Systems, Saitama Institute of Technology, Saitama, Japan

The electromagnetic interaction of a charged particle beam with multilayer vacuum chambers is of particular interest in accelerator physics. This paper presents a deep learning-based approach for calculating electromagnetic fields generated by the beam in infinitely long multilayer vacuum chambers with arbitrary cross section. The presented approach is based on physics-informed neural networks and the surface impedance boundary condition of a multilayer structure derived from the transmission line theory. Deep neural networks (DNNs) are utilized to approximate the solution of partial differential equations (PDEs) describing the physics of electromagnetic fields self-generated by a charged particle beam traveling in a particle accelerator. A residual network is constructed from the output of DNNs, the PDEs and boundary conditions are embedded into the loss function and differential operators are calculated using the automatic differentiation. As a result, the presented approach is regarded to be mesh-free. The approach is applied to circular and elliptical vacuum chambers with a three-layer structure. It is verified in comparison with the recently proposed boundary element method. The effects of chamber geometries and multilayer structure on the beam coupling impedance are demonstrated.

KEYWORDS

deep learning, machine learning, neural network, electromagnetic field, resistive-wall wakefield, charged particle

1 Introduction

Charged particles moving in an accelerator are exposed to electromagnetic fields of components specified by the accelerator design. These external fields are used to store and accelerate beams of charged particles. However, the charged particles themselves are sources of electromagnetic fields [1]. Due to the movement of the particles, they behave like a current in an accelerator vacuum chamber. The field accompanied by the beam (called as self-field) is scattered on the wall surface of its chamber due to finite conductivity of the wall material and/or cross section variation. As a result of this electromagnetic interaction, unwanted electromagnetic fields are excited. Such fields, referred as to wakefields due to the fact that they are left behind the traveling particle, act back on the beam itself and the following beams, and subsequently influence the energy and motion of the charged particles [2, 3]. As the beam intensity increases, the wakefields become significantly strong, and also will perturb the prescribed external fields. The wakefield effects can limit the performance of an accelerator in terms of beam quality and beam current. Therefore, accurate knowledge of wakefields is required in the design of vacuum chamber components in an accelerator.

The integrated effects of the wakefields on the beam can be estimated by the beam coupling impedance in the frequency domain [4, 5]. To calculate the impedances of various accelerator vacuum chambers, both analytical and numerical approaches have been developed complementarily. Analytical (or semi-analytical) approaches [6–14] are important for understanding the characteristic of wakefields although their applicabilities are usually limited to simple geometries. A practical way of calculating impedances of realistic vacuum chambers with complex geometries and materials is to use purely numerical methods such as the finite integration technique (FIT) [15] and finite element method (FEM) [16]. To obtain a numerical solution, the FIT and FEM require the discretization of domains for a vacuum chamber of interest. The standard FIT with structured grids suffers from the staircasing error of curved boundaries. The FEM with unstructured grids allows accurate modeling of boundary surfaces but may need the generation of dense meshes due to a large variation of the fields in the vicinity of the source domain. When calculating the impedance due to finite conductivity of the chamber wall, generally called resistive wall impedance, one needs to address the skin effect in the chamber wall properly. The resistive wall impedance of multilayer vacuum chamber is of particular interest for modern high energy accelerators and x-ray free electron laser projects. Many efforts have been made to analytically investigate the wakefields and impedance in multilayer vacuum chambers, see e.g., [17–23]. However, it is still challenging to calculate the impedance of multilayer vacuum chambers with arbitrary cross section.

This paper presents a new approach for calculating the beam coupling impedance of infinitely long multilayer vacuum chambers with arbitrary cross section. Unlike traditional numerical methods, our approach is based on deep learning in the form of neural networks (NN), termed as the physics-informed neural networks (PINN) [24, 25]. A deep neural network is utilized as a solution surrogate to approximate the solution of governing equations of wakefields generated in a multilayer vacuum chamber. The differential operators in the residual network of the governing equations are evaluated with automatic differentiation. Therefore, no numerical mesh exists inside the domain surrounded by the chamber wall surface and also on its surface. For these reasons, the presented approach can be regarded to be mesh-free. To avoid calculating fields penetrated in the wall and model a multilayer structure, the transmission line (TL) theory [18] and the surface impedance concept [26] are used in the present study.

It should be pointed out that PINN was first introduced into beam coupling impedance modeling in Ref. [27], where the space charge impedance of a relativistic beam with transversally Gaussian charge density in an infinitely long vacuum chamber with walls of infinite conductivity is simulated for various cross sections. Very recently, PINN was extended to the case when a vacuum chamber has finite wall conductivity in Ref. [28], where the surface impedance concept was combined with PINN. A single-layer circular vacuum chamber with a finite wall thickness and a small conductivity was analyzed in [28], where it is assumed that the magnetic field on the resistive wall is the same as that on the perfectly electric conductor (PEC) wall; the effect of finite wall conductivity on the magnetic field is enough small. This is referred to as the perturbative treatment of the magnetic field. It has been also used to simplify a problem in the analytical impedance studies; see e.g., [5, 11]. However, the perturbation is valid only for a limited frequency range. A nonperturbative treatment of the magnetic field is required to compute the coupling impedance at high frequencies.

To the best of the author's knowledge, the modeling of multilayer vacuum chambers with more than two layers in particle accelerators has never been studied in the framework of PINN, although its possibility has been mentioned in Ref. [28]. Nonperturbative cases are also not yet discussed in this context. The purpose of this paper is therefore to extend the previous study [28] to the nonperturbative modeling of coupling impedances and wakefields in multilayer vacuum chambers. This study will be the first application of PINN to this subject. The key idea of the presented approach is to use PINN [24, 25] and the surface impedance boundary condition (SIBC) of a multilayer structure derived from TL theory [18] in the nonperturbative model. To model vacuum chamber geometries other than the circular one, the Swish activation function is chosen for the NN architecture. The goal of this paper is to clarify that the presented approach can be applied to the nonperturbative impedance modeling of multilayer vacuum chambers. The discussion of this paper is limited to this point.

This paper is organized as follows. In Section 2, we state the problem solved. In Section 3, we present a mesh-free numerical method based on PINN for calculating electromagnetic wakefields in frequency domain. In Section 4, we show numerical results of circular and elliptical multilayer vacuum chambers with PINN. This paper is concluded in Section 5.

2 Problem statement

2.1 Partial differential equations

We consider a relativistic charged particle beam with a transversally Gaussian charge density distribution (total charge

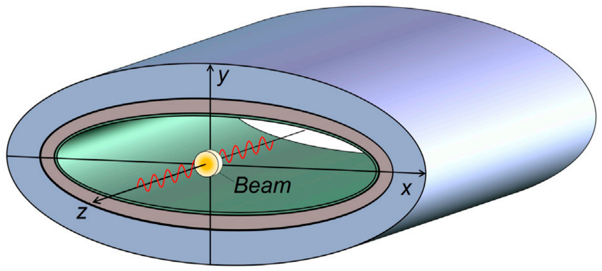


FIGURE 1
A relativistic charged particle beam with transversally Gaussian charge density moving at constant velocity on the axis of an infinitely long multilayer vacuum chamber with elliptical cross section.

Q) moving in an infinitely long vacuum chamber with an arbitrary but constant cross section and walls of finite conductivity, as shown in Figure 1. Throughout this paper, we assume that the particle beam has a rigid charge distribution ρ and a constant velocity $\mathbf{v}$, and the beam current density $\mathbf{J} = \rho\mathbf{v}$, where $\mathbf{v} = v\mathbf{e}_z = \beta c\mathbf{e}_z$, $\beta = v/c$, c is the speed of light in vacuum, and $\mathbf{e}_z$ is the unit vector along the z-direction. The influence of the field on the beam motion is neglected in the field calculation. This rigid beam picture is not self-consistent, but it is an excellent approximation for relativistic beams [2–4]. Our interest is to solve electromagnetic boundary value problems for a given charge distribution in the context of PINN.

In general, the electric and magnetic fields $(\mathbf{E}, \mathbf{H})$ in the presence of the particle beam in vacuum obey the Maxwell equations [1, 5]:

$$\nabla\times \mathbf{E} = -\mu_0\frac{\partial \mathbf{H}}{\partial t}, \tag{1}$$

$$\nabla\times \mathbf{H} = \mathbf{J} + \varepsilon_0\frac{\partial \mathbf{E}}{\partial t}, \tag{2}$$

$$\nabla\cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}, \tag{3}$$

$$\nabla\cdot \mathbf{H} = 0. \tag{4}$$

where ε_0 and μ_0 denote the permittivity and the permeability of vacuum, respectively. In the frequency domain, the Maxwell Eqs. 1–4 can be transformed into

$$\nabla\times \mathbf{E} = -j\omega\mu_0\mathbf{H}, \tag{5}$$

$$\nabla\times \mathbf{H} = \mathbf{J} + j\omega\varepsilon_0\mathbf{E}, \tag{6}$$

$$\nabla\cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}, \tag{7}$$

$$\nabla\cdot \mathbf{H} = 0. \tag{8}$$

Here we assume the time dependence $e^{j\omega t}$ with an angular frequency $\omega = 2\pi f$. When one consider only a particular harmonic component, the charge and current densities are written as

$$\rho(x, y, z, t) = \rho_\perp(x, y)e^{j(\omega t - kz)}, \tag{9}$$

$$\mathbf{J}(x, y, z, t) = J_z(x, y)e^{j(\omega t - kz)}\mathbf{e}_z, \tag{10}$$

respectively, where $k = \omega/v$ is the wave number, $J_z = \rho_\perp\beta c$, $\rho_\perp$ is the transverse charge density distribution function. The electric and magnetic fields are also written as:

$$\mathbf{E}(x, y, z, t) = \mathbf{E}(x, y)e^{j(\omega t - kz)}, \tag{11}$$

$$\mathbf{H}(x, y, z, t) = \mathbf{H}(x, y)e^{j(\omega t - kz)}, \tag{12}$$

respectively. From the frequency domain Maxwell Eqs. 5–8, we can obtain the following inhomogeneous wave equations of the electric field $\mathbf{E} = (E_x, E_y, E_z)$ and magnetic field $\mathbf{H} = (H_x, H_y, H_z)$ [1]:

$$\nabla^2\mathbf{E} + \omega^2\varepsilon_0\mu_0\mathbf{E} = \frac{\nabla\rho}{\varepsilon_0} + j\omega\mu_0\mathbf{J}, \tag{13}$$

$$\nabla^2\mathbf{H} + \omega^2\varepsilon_0\mu_0\mathbf{H} = -\nabla\times \mathbf{J}. \tag{14}$$

Substituting Eqs. 9–12 into Eqs. 13, 14 leads to

$$\frac{\partial^2 E_x}{\partial x^2} + \frac{\partial^2 E_x}{\partial y^2} - \frac{k^2}{\gamma^2}E_x = \frac{1}{\varepsilon_0}\left(\frac{\partial\rho_\perp}{\partial x}\right), \tag{15}$$

$$\frac{\partial^2 E_y}{\partial x^2} + \frac{\partial^2 E_y}{\partial y^2} - \frac{k^2}{\gamma^2}E_y = \frac{1}{\varepsilon_0}\left(\frac{\partial\rho_\perp}{\partial y}\right), \tag{16}$$

$$\frac{\partial^2 E_z}{\partial x^2} + \frac{\partial^2 E_z}{\partial y^2} - \frac{k^2}{\gamma^2}E_z = -\frac{jk}{\varepsilon_0\gamma^2}\rho_\perp, \tag{17}$$

$$\frac{\partial^2 H_x}{\partial x^2} + \frac{\partial^2 H_x}{\partial y^2} - \frac{k^2}{\gamma^2}H_x = -\frac{\partial J_z}{\partial y}, \tag{18}$$

$$\frac{\partial^2 H_y}{\partial x^2} + \frac{\partial^2 H_y}{\partial y^2} - \frac{k^2}{\gamma^2}H_y = \frac{\partial J_z}{\partial x}, \tag{19}$$

$$\frac{\partial^2 H_z}{\partial x^2} + \frac{\partial^2 H_z}{\partial y^2} - \frac{k^2}{\gamma^2}H_z = 0, \tag{20}$$

where $\gamma = (1 - \beta^2)^{-1/2}$ is the Lorentz factor. As described later, we are interested in Eq. 17 to calculate the coupling impedance.

2.2 Boundary conditions

To model the walls of a vacuum chamber, one needs to enforce the boundary condition. In the case of PEC walls, we use the PEC boundary condition (BC) as follows [1, 5]:

$$\mathbf{n} \times \mathbf{E} = 0. \tag{21}$$

For a smooth wall of finite conductivity, we replace the PEC-BC by the Leontovich boundary condition or SIBC [5, 26]:

$$\mathbf{n} \times \mathbf{E} = Z_s(\omega)\mathbf{n} \times \mathbf{n} \times \mathbf{H}, \tag{22}$$

where $\mathbf{n}$ is the outward unit vector normal to the wall surface and $Z_s(\omega)$ is called as the surface impedance, which is complex. When $Z_s = 0$, Eq. 22 can be reduced to the PEC-BC Eq. 21. Note that Eq. 22 is enforced only on the innermost wall of the chamber.

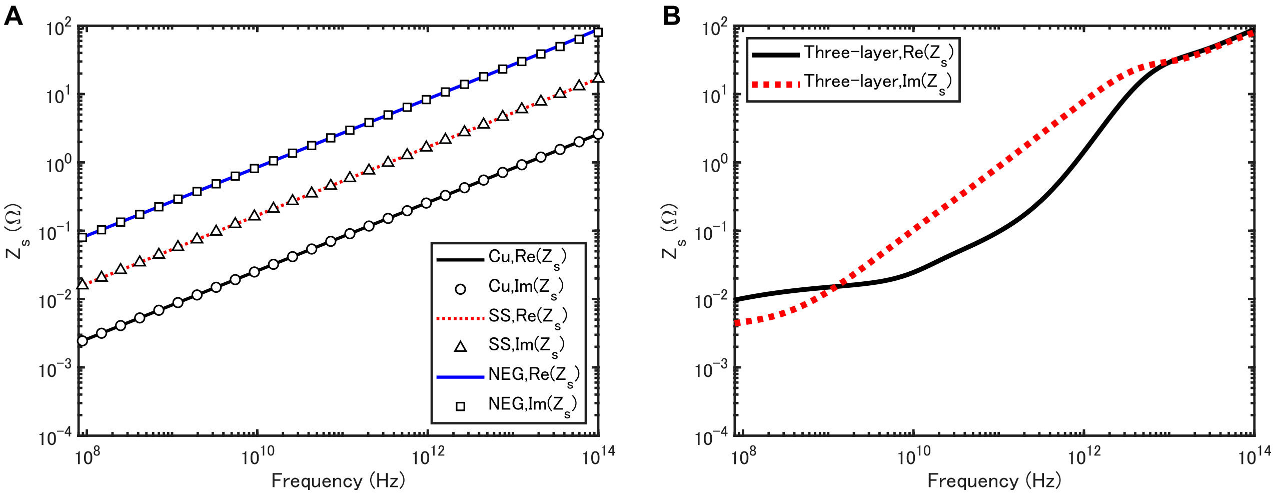


FIGURE 2
Surface impedances for the smooth wall of SS, Cu, NEG **(A)** and for the inner wall of the three-layer structure **(B)**.

In the field solution, all the domain outside the innermost wall is assumed to be filled by PEC. This can be also regarded as the assumption of infinitely thick PEC wall. Therefore, the field is zero outside the innermost chamber wall. This surface impedance concept can be widely used for the impedance modeling in particle accelerators. See e.g. [29].

For later reference, we summarize Z_s in some special cases. As a well-known example, the normal skin effect in a chamber wall with infinite thickness can be modeled with [5, 18, 26].

$$Z_s(\omega) = \sqrt{\frac{j\omega\mu_0}{\sigma_c + j\omega\varepsilon_0}}. \tag{23}$$

where σ_c is the static (dc) conductivity. For a multilayer structure, the transmission line theory [18] can be used to obtain its surface impedance in Eq. 22. The same approach as Ref. [18] is adapted in this work. As a result, for a three-layer structure with dc conductivities σ_i and thicknesses d_i ($i = 1, 2, 3$, respectively), Z_s on the inner wall is given by

$$Z_s = Z_1 \cdot \frac{Z_{i2} + Z_1 \tanh(\gamma_1 d_1)}{Z_1 + Z_{i2} \tanh(\gamma_1 d_1)}, \tag{24}$$

$$Z_{i2} = Z_2 \cdot \frac{Z_{i3} + Z_2 \tanh(\gamma_2 d_2)}{Z_2 + Z_{i3} \tanh(\gamma_2 d_2)}, \tag{25}$$

$$Z_{i3} = Z_3 \cdot \frac{Z_0 + Z_3 \tanh(\gamma_3 d_3)}{Z_3 + Z_0 \tanh(\gamma_3 d_3)}, \tag{26}$$

where Z_0 is the free space impedance and Z_i and γ_i are the intrinsic impedance and the propagation constant of ith layer material, respectively, given by

$$Z_i = \sqrt{\frac{j\omega\mu_0}{\sigma_i + j\omega\varepsilon_0}}, \tag{27}$$

$$\gamma_i = \sqrt{j\omega\mu_0(\sigma_i + j\omega\varepsilon_0)}. \tag{28}$$

Note that the vacuum regions are assumed to be inside the innermost wall surface and also outside the outermost wall surface.

Figure 2 shows surface impedances obtained with Eq. 23 for stainless steel (SS), copper (Cu), and nonevaporable getter (NEG) and a surface impedance calculated with Eq. 24 for a three-layer structure with the same conductivities. The materials used from external to inner layer are SS, Cu and NEG with the corresponding dc conductivities $\sigma_3 = 0.14 \times 10^7$ S/m, $\sigma_2 = 5.88 \times 10^7$ S/m, $\sigma_1 = 0.55 \times 10^5$ S/m. Layer thicknesses are $d_3 = 1$ mm for SS, $d_2 = 1$ μm for Cu and $d_1 = 1$ μm for NEG, respectively. As well known, for good conductors, Eq. 23 can be simplified as

$$Z_s(\omega) = (1 + j)\sqrt{\frac{Z_0\omega}{2\sigma_c c}}. \tag{29}$$

Therefore, the curve of the real part of the surface impedance agree with that of the imaginary one for Cu, SS and NEG, respectively. By contrast, for the three-layer structure, the frequency dependency of the real part is quite different from that of the imaginary one.

2.3 Kirchhoff's boundary integral representation of electromagnetic field

The electromagnetic field in an infinitely long vacuum chamber can be also expressed in Kirchhoff's boundary integral representation as [30].

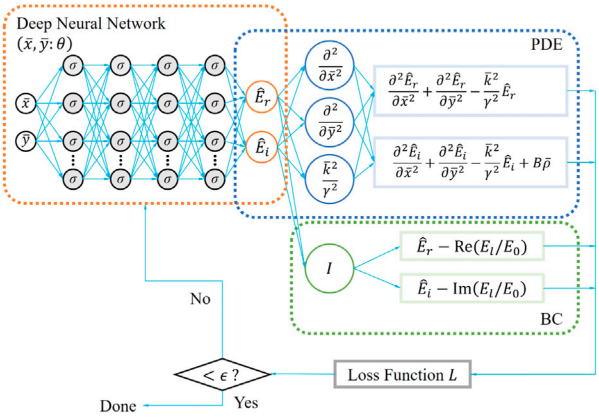

Deep Neural Network
$(\bar{x}, \bar{y}: \theta)$
σ
$\bar{x}$
$\bar{y}$
$\hat{E}_r$
$\hat{E}_i$
PDE
$\frac{\partial^2}{\partial \bar{x}^2}$
$\frac{\partial^2}{\partial \bar{y}^2}$
$\frac{\bar{k}^2}{\gamma^2}$
$\frac{\partial^2 \hat{E}_r}{\partial \bar{x}^2} + \frac{\partial^2 \hat{E}_r}{\partial \bar{y}^2} - \frac{\bar{k}^2}{\gamma^2} \hat{E}_r$
$\frac{\partial^2 \hat{E}_i}{\partial \bar{x}^2} + \frac{\partial^2 \hat{E}_i}{\partial \bar{y}^2} - \frac{\bar{k}^2}{\gamma^2} \hat{E}_i + B\bar{\rho}$
I
$\hat{E}_r - \mathrm{Re}(E_l/E_0)$
$\hat{E}_i - \mathrm{Im}(E_l/E_0)$
BC
No
$< \epsilon$?
Yes
Done
Loss Function L

FIGURE 3
Physics-informed neural network.

$$\mathbf{E} = \mathbf{E}_b - \nabla_\perp \phi - j\omega \mathbf{A} - \frac{1}{\varepsilon_0} \nabla_\perp \times \mathbf{F}, \tag{30}$$

$$\mathbf{H} = \mathbf{H}_b - \nabla_\perp \phi_m - j\omega \mathbf{F} + \frac{1}{\mu_0} \nabla_\perp \times \mathbf{A} \tag{31}$$

with the vector and scalar potentials

$$\mathbf{A}(\mathbf{r}) = \mu_0 \int_C G(\mathbf{r}, \mathbf{r}') \mathbf{K}(\mathbf{r}') d\mathbf{r}', \tag{32}$$

$$\mathbf{F}(\mathbf{r}) = \varepsilon_0 \int_C G(\mathbf{r}, \mathbf{r}') \mathbf{M}(\mathbf{r}') d\mathbf{r}', \tag{33}$$

$$\phi(\mathbf{r}) = \frac{1}{\varepsilon_0} \int_C G(\mathbf{r}, \mathbf{r}') \sigma(\mathbf{r}') d\mathbf{r}', \tag{34}$$

$$\phi_m(\mathbf{r}) = \frac{1}{\mu_0} \int_C G(\mathbf{r}, \mathbf{r}') \eta(\mathbf{r}') d\mathbf{r}', \tag{35}$$

and the boundary conditions

$$\mathbf{n} \times \mathbf{E} = -\mathbf{M}, \tag{36}$$

$$\mathbf{n} \times \mathbf{H} = \mathbf{K}, \tag{37}$$

$$\mathbf{n} \cdot \mathbf{E} = \sigma / \varepsilon_0, \tag{38}$$

$$\mathbf{n} \cdot \mathbf{H} = \eta / \mu_0, \tag{39}$$

and Green's function [1].

$$G(\mathbf{r}, \mathbf{r}') = \frac{1}{2\pi} K_0(k_r R), R = |\mathbf{r} - \mathbf{r}'|, k_r = k/\gamma, \tag{40}$$

where ($\mathbf{E}_b$, $\mathbf{H}_b$) are the beam self-fields, $\nabla_\perp^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the two-dimensional (2D) Laplacian operator, $\mathbf{r} = (x, y)$ is the observation point in the bounded vacuum region, $\mathbf{r}'$ is the point on the boundary surface, $\mathbf{n}$ is the inner unit vector normal to the inner wall surface C, K_0 is the modified Bessel function of the second kind of order zero, ($\mathbf{K}$, σ) are the electric surface current and charge densities and ($\mathbf{M}$, η) are the magnetic surface current and charge densities.

The above integral representation clearly shows that the total electromagnetic fields ($\mathbf{E}$, $\mathbf{H}$) can be expressed as a superposition of the beam self-fields and the boundary integrals over the surface charges and currents on the chamber walls. See Ref. [30] for more detailed discussions on Eqs. 30, 31.

Here we should mention the perturbative treatment of magnetic field; the magnetic field on resistive walls is assumed to be the same as that on PEC walls. In this case, $\mathbf{M} = 0$ and $\eta = 0$. Therefore, to compute the magnetic field in the perturbative model, one uses

$$\mathbf{H} = \mathbf{H}_b + \frac{1}{\mu_0} \nabla_\perp \times \mathbf{A} \tag{41}$$

together with Eq. 37. In the previous study [28], the boundary data for PINN was generated with Eq. 22 and Eq. 41

Unlike Ref. [28], in this paper, boundary data for PINN are generated with the nonperturbative model based on Eqs. 30, 31 including Eq. 22 with Eq. 23 or 24.

3 Physics-informed neural network method

3.1 Data and equation scaling

For calculating the electromagnetic field in particle accelerators and the beam coupling impedance, we will deal with different magnitudes of input and output. For example, the field magnitude may vary largely within a vacuum chamber under consideration or in a frequency range of interest. The real part of the field can be also different from the imaginary one due to the frequency dependence of a surface impedance. In such cases, we have to deal with a dataset carefully. Using a raw dataset can lead to slow convergence of a gradient-based optimizer. To avoid this, we need to scale the input, output and PDE in an appropriate way. It should be mentioned that this basic idea was first described in Ref. [27]. Although the final scaled form of the PDE Eq. 17 was given in Ref. [28], its derivation was omitted. In the following, we present a detailed formulation for the data and equation scaling.

Let us consider modeling a vacuum chamber geometry in Cartesian coordinates. The x-axis and y-axis are scaled with typical chamber dimensions (s_x, s_y) such as radius, height and width as

$$\bar{x} = \frac{x}{s_x}, \bar{y} = \frac{y}{s_y}. \tag{42}$$

We then scale the sampling points (input) with Eq. 42 and transform the PDE Eq. 17 into

$$\frac{1}{s_x^2} \frac{\partial^2 E_z}{\partial \bar{x}^2} + \frac{1}{s_y^2} \frac{\partial^2 E_z}{\partial \bar{y}^2} - \frac{k^2}{\gamma^2} E_z = -\frac{jk}{\varepsilon_0 \gamma^2} \rho_\perp. \tag{43}$$

Next, by scaling the field E_z (output) as

$$\bar{E}_z = \frac{E_z}{E_0}, \tag{44}$$

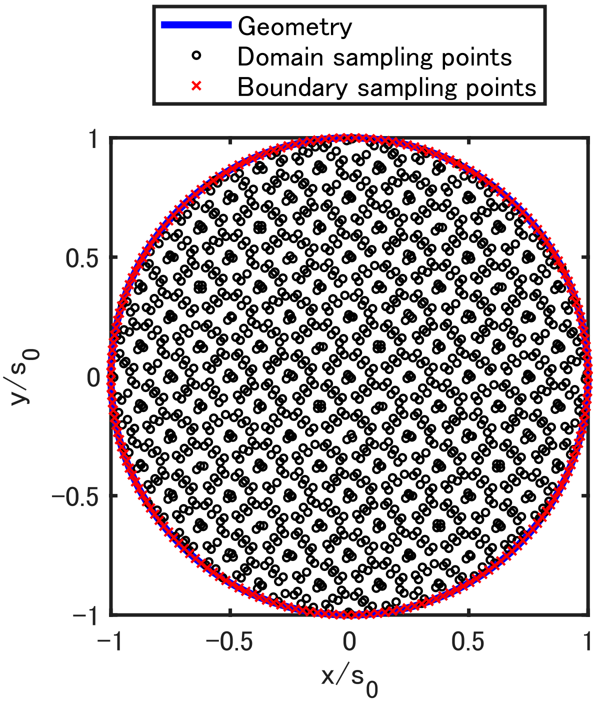


FIGURE 4
Generation of sampling points in a domain surrounded by the innermost wall of a circular vacuum chamber.

we obtain

$$\frac{1}{s_x^2}\frac{\partial^2 \bar{E}_z}{\partial \bar{x}^2} + \frac{1}{s_y^2}\frac{\partial^2 \bar{E}_z}{\partial \bar{y}^2} - \frac{k^2}{\gamma^2}\bar{E}_z = -\frac{jk}{\varepsilon_0\gamma^2 E_0}\rho_\perp, \tag{45}$$

where $\rho_\perp$ is given by

$$\rho_\perp = \frac{Q}{2\pi\sigma_x\sigma_y} e^{-\frac{(x-x_c)^2}{2\sigma_x^2} - \frac{(y-y_c)^2}{2\sigma_y^2}}, \tag{46}$$

where (σ_x, σ_y) is the Gaussian rms size in the x- and y-direction and (x_c, y_c) is the center position in the transverse plane. Finally, by setting $s_x = s_y = s_0$, introducing a new parameter B and choosing E_0 as

$$E_0 = q_n s_0{}^2 \frac{k}{B\varepsilon_0\gamma^2}, q_n = \frac{Q}{2\pi\sigma_x\sigma_y}, \tag{47}$$

we can derive a scaled PDE as follows:

$$\frac{\partial^2 \bar{E}_z}{\partial \bar{x}^2} + \frac{\partial^2 \bar{E}_z}{\partial \bar{y}^2} - \frac{\bar{k}^2}{\gamma^2}\bar{E}_z = -jB\bar{\rho}, \tag{48}$$

$$\bar{\rho} = \exp\left[-\frac{(\bar{x}s_0 - x_c)^2}{2\sigma_x^2} - \frac{(\bar{y}s_0 - y_c)^2}{2\sigma_y^2}\right], \tag{49}$$

where $\bar{k} = ks_0$ and $\bar{\rho} = \rho_\perp/q_n$ denotes the scaled charge density distribution. Throughout this paper, $B = 1$ is empirically chosen. Note that γ and $\bar{\rho}$ are included even in the scaled PDE Eq. 48, which has the same form as the original PDE Eq. 17.

As the scaled boundary condition for $\bar{E}_z$, we adapt

$$\bar{E}_z = E_l \Big/ E_0, \tag{50}$$

where E_l is the longitudinal electric field on the wall surface obtained with the recently proposed boundary element method (BEM) [30]. Note that, as already mentioned in Section 2.3, the nonperturbative model is used to obtain E_l in Eq. 50.

We highlight the effect of the above data and equation scaling by comparing Eq. 48 with Eq. 17. The right-hand side (RHS) of Eq. 17 will be relatively large due to the existence of $\varepsilon_0 \approx 8.854 \times 10^{-12}$ [F/m]. In addition, the RHS depends on the wavenumber k or the frequency f. These are unpreferable for using a gradient-based optimizer. By contrast, in the presented formulation Eqs. 42–50, the RHS of (48) is characterized by $B\bar{\rho}$ and it remains unchanged over a frequency range of interest. This feature is preferable for calculating the beam coupling impedance in the frequency domain and for maintaining the accuracy of trained NNs over a frequency range. In fact, this scaling has been successfully applied to all benchmark examples in this study. The presented scaling scheme is performed as pre-processing and post-processing.

Note that the physics of the electromagnetic field self-generated by the beam in a vacuum chamber can be described by Eqs 48, 50.

3.2 Deep learning

Let us consider solving the derived PDE *via* a new class of deep learning, termed as PINN [24, 25]. A deep neural netowrk (DNN) is used to approximate the solution of Eq. 48. This is often called as a solution surrogate. In addition, to train this DNN, we take the output of a DNN, define a network associated to the residual of Eqs 48, 50 and calculate the residual value (called a loss function). Note that differential operators in this residual network are calculated using the automatic differentiation. Therefore, unlike traditional numerical methods, our approach does not need to define (and generate) meshes inside the chamber. Physically, the space charge field has only a purely imaginary part, the resistive wall wake field has both real and imaginary parts. Therefore, the DNN also has two outputs $(\hat{E}_r, \hat{E}_i)$ corresponding to the real (r) and imaginary 1) parts of $\bar{E}_z$. A schematic picture of constructed PINN including the DNN and the residual network is illustrated in Figure 3. The inputs of the DNN are sampling points $(\bar{x}_p, \bar{y}_p)$ in the scaled xy plane. We assume that the beam traverses inside the chamber and the field is zero outside the chamber. The approach described here works well especially for smooth transverse charge density as in (49).

Our algorithm is summarized in the following list.

1) Set up a computational domain, the boundary condition, physical constants, beam parameters, source domain, and the scaling parameters.

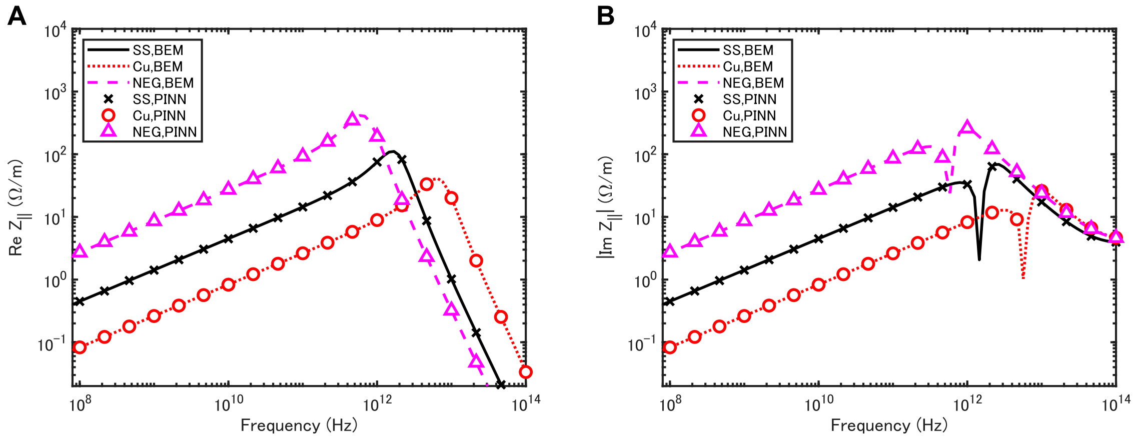


FIGURE 5
Coupling impedances of a round Gaussian beam with σ_r = 0.5 mm moving on the axis of a circular vacuum chamber with infinite thickness for three different wall materials: copper (Cu), stainless steel (SS), nonevaporable getter (NEG). The copper and NEG chambers have radius of 5 mm and the SS chamber has radius of 6 mm. Real part **(A)** and imaginary part **(B)**.

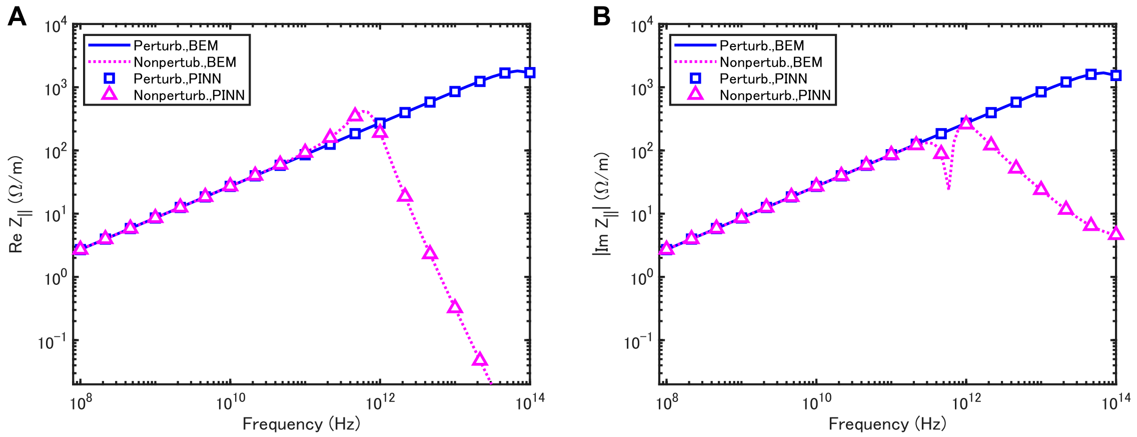


FIGURE 6
Comparison of the perturbative and nonperturbative models in the coupling impedances of a round Gaussian beam with σ_r = 0.5 mm moving on the axis of a circular vacuum chamber with infinite thickness for NEG. The NEG chamber has radius of 5 mm. Real part **(A)** and imaginary part **(B)**.

2) Generate randomly sampled points (or regular or irregular grid points) within the computational domain surrounded by the innermost wall. No sampling point is generated outside the domain.
3) Construct a DNN with two outputs $\hat{E}_r(\bar{x},\bar{y};\theta), \hat{E}_i(\bar{x},\bar{y};\theta)$ as a solution surrogate of the scaled PDE $\bar{E}_z(\bar{x},\bar{y}) = \bar{E}_r(\bar{x},\bar{y}) + j\bar{E}_i(\bar{x},\bar{y})$, where θ is a vector containing all weights w and bias b in the neural network to be trained, σ denotes an activation function.
4) Define the loss function L including Eqs 48, 50
5) Train the constructed DNN to find the best parameters θ by minimizing L via the L-BFGS algorithm [31] as a gradient-based optimizer, until L is smaller than a threshold ϵ.
6) Calculate the original field (unscaled) from $\hat{E}_z(x,y;\theta) = E_0\hat{E}_r(x/s_0, y/s_0;\theta) + jE_0\hat{E}_i(x/s_0, y/s_0;\theta)$ using the trained DNN.

In this method, the loss function L is defined by

$$L = w_r^d L_r^d + w_r^b L_r^b + w_i^d L_i^d + w_i^b L_i^b, \tag{51}$$

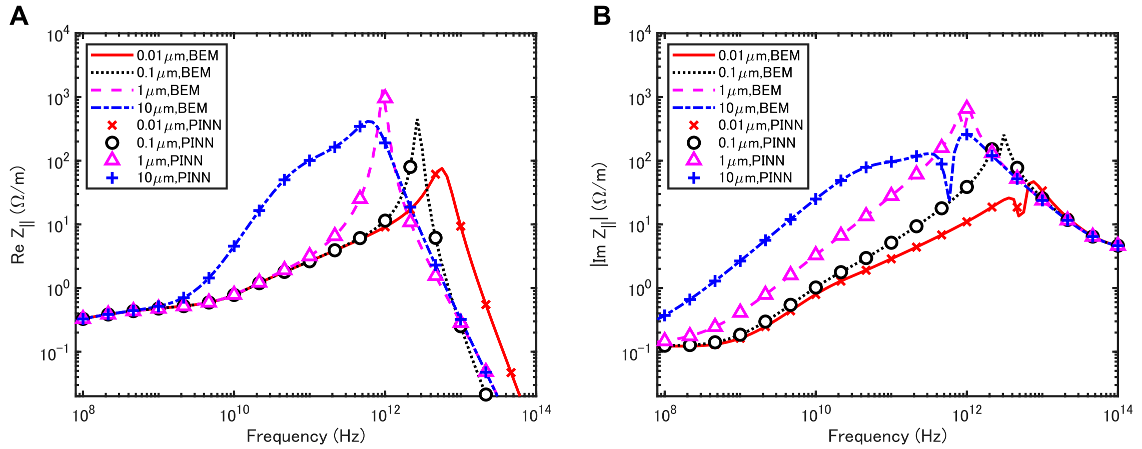


FIGURE 7
Coupling impedances of a round Gaussian beam with σ_r = 0.5 mm moving on the axis of the circular vacuum chamber with the three-layer structure of copper (Cu), stainless steel (SS), nonevaporable getter (NEG). Real part **(A)** and imaginary part **(B)**.

$$L_r^d = \frac{1}{N_d}\sum_{p=1}^{N_d}\left|f_r\left(x_p, y_p; \theta\right)\right|^2, \quad L_i^d = \frac{1}{N_d}\sum_{p=1}^{N_d}\left|f_i\left(x_p, y_p; \theta\right)\right|^2, \tag{52}$$

$$L_r^b = \frac{1}{N_b}\sum_{p=1}^{N_b}\left|g_r\left(x_p, y_p; \theta\right)\right|^2, \quad L_i^b = \frac{1}{N_b}\sum_{p=1}^{N_b}\left|g_i\left(x_p, y_p; \theta\right)\right|^2, \tag{53}$$

$$f_r = \frac{\partial^2 \hat{E}_r}{\partial \bar{x}^2} + \frac{\partial^2 \hat{E}_r}{\partial \bar{y}^2} - \frac{\bar{k}^2}{\gamma^2}\hat{E}_r, \quad f_i = \frac{\partial^2 \hat{E}_i}{\partial \bar{x}^2} + \frac{\partial^2 \hat{E}_i}{\partial \bar{y}^2} - \frac{\bar{k}^2}{\gamma^2}\hat{E}_i + B\bar{\rho}, \tag{54}$$

$$g_r = \hat{E}_r - \mathrm{Re}\left(E_l/E_0\right), \quad g_i = \hat{E}_i - \mathrm{Im}\left(E_l/E_0\right), \tag{55}$$

where p denotes the sampling point, N_d and N_b are the numbers of sampling points in the computational domain and on the boundary surface, respectively. w_r^d, w_i^d, w_r^b and w_i^b are the weights of the loss function. $L_{r/i}^d$ is the loss function related to the scaled PDE, and its minimization ($L_{r/i}^d \to 0$) enforces Eq. 48 at a set of finite sampling points in the computational domain. $L_{r/i}^b$ is the loss function related to the SIBC, and its minimization ($L_{r/i}^b \to 0$) enforces Eq. 50 at a set of finite sampling points on the boundary surface.

Throughout this study, a fully connected feedforward neural network was adapted, and four hidden layers and 30 neurons per layer were used. In deep learning, tanh, sigmoid, Rectified Linear Unit (ReLU) and Swish functions are commonly used as nonlinear activation functions. See e.g., Refs. [32, 33]. Smooth activation functions are required in PINN. Since ReLU is not differentiable at the origin, we do not choose it here. For this study, the Swish function [32] is used, because it tends to work well for various chamber geometries, compared to the tanh and sigmoid functions. This activation function was not used in the previous studies [27, 28]. $B = 1$ and $(w_r^d, w_i^d, w_r^b, w_i^b)$=(10,1,10,1) were chosen.

3.3 Impedance computation

After training the neural network, one can predict the field $\hat{E}_z(x, y; \theta)$ for any position (x, y) in a vacuum chamber. In order to estimate integrated effect of the field on the beam in the frequency domain, the concept of coupling impedance is commonly used in accelerator physics. Here the coupling impedance is defined as [4].

$$Z_{\|}(x, y) = -\frac{\hat{E}_z(x, y; \theta)}{I}, \tag{56}$$

where $I = Q\beta c$ is the total beam current. The coupling impedance (per unit length) at any position inside the source domain of (46) is obtained from (56). Although one can define the impedance averaged over the source domain, its evalution is out of the present paper. This paper focuses on the local impedance defined in (56).

4 Results and discussion

To show the feasibility of the presented approach, we apply it to the impedance analysis of multilayer vacuum chambers with circular and elliptical cross section. Here, we assume the vertical dimension (height) $h = 2b = 10$ mm as in [20] and $\gamma = 10{,}000$, which is corresponding to the ultrarelativistic case. The PINN-predicted results are verified in comparison with simulation results of the recently developed BEM.

First, the coupling impedance of a circular vacuum chamber of infinite thickness is simulated for different wall materials (N_d, N_b)=(1,578, 200) was used for a circular chamber. The domain and boundary sampling points were generated as in Figure 4, where the coordinates are scaled with $s_0 = 5$ mm.

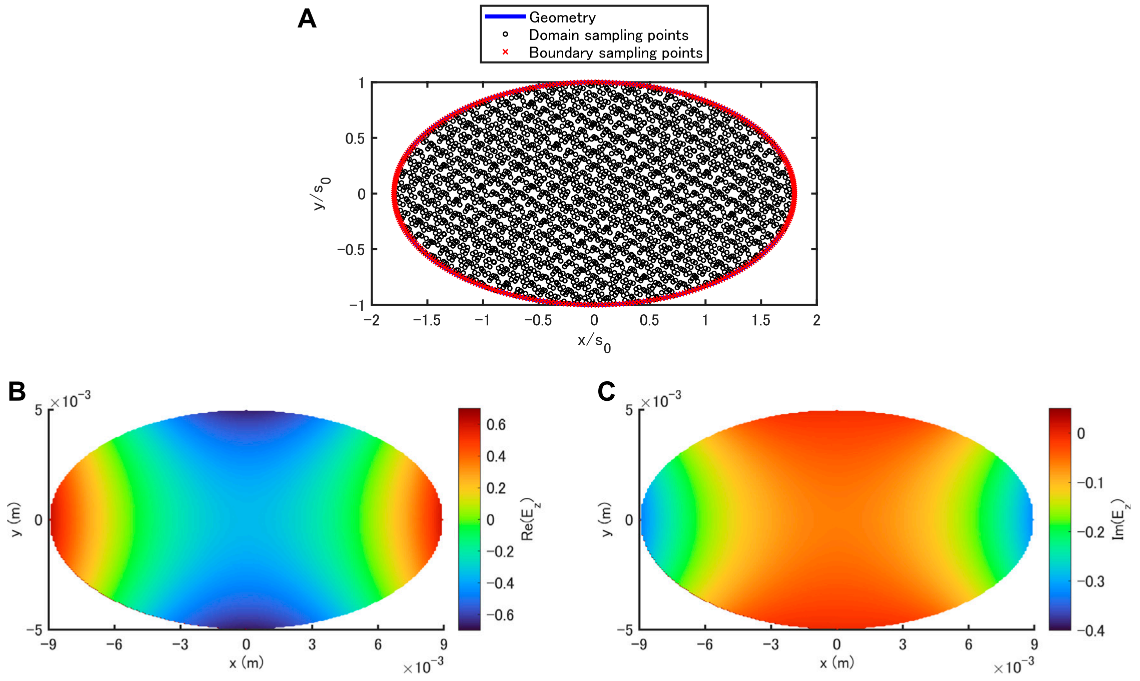


FIGURE 8
PINN-simulated field distribution E_z over a domain surrounded by the innermost wall of a three-layer elliptical vacuum chamber **(A)** sampling points **(B)** real part **(C)** imaginary part.

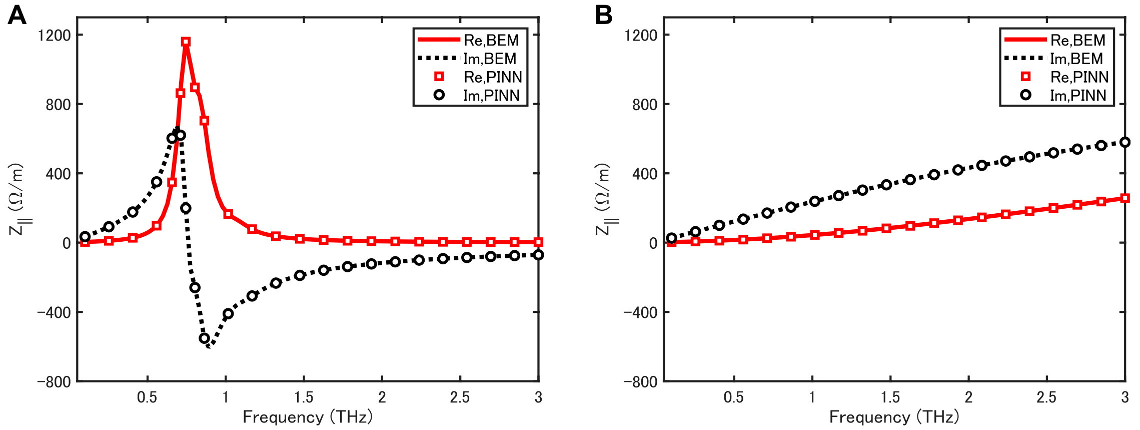


FIGURE 9
Coupling impedance of a round Gaussian beam with σ_r = 0.5 mm for an elliptical vacuum chamber with the three-layer structure of copper (Cu), stainless steel (SS), nonevaporable getter (NEG) **(A)** nonperturbative model **(B)** perturbative model.

Figure 5 shows the coupling impedances of a round Gaussian beam with $\sigma_x = \sigma_y = \sigma_r = 0.5\,\text{mm}$ on the axis of the circular chamber for copper (Cu), stainless steel (SS) and nonevaporable getter (NEG). As expected, both the real and imaginary parts of the coupling impedance decrease at each frequency point as σ_c increases. For example, the impedance of

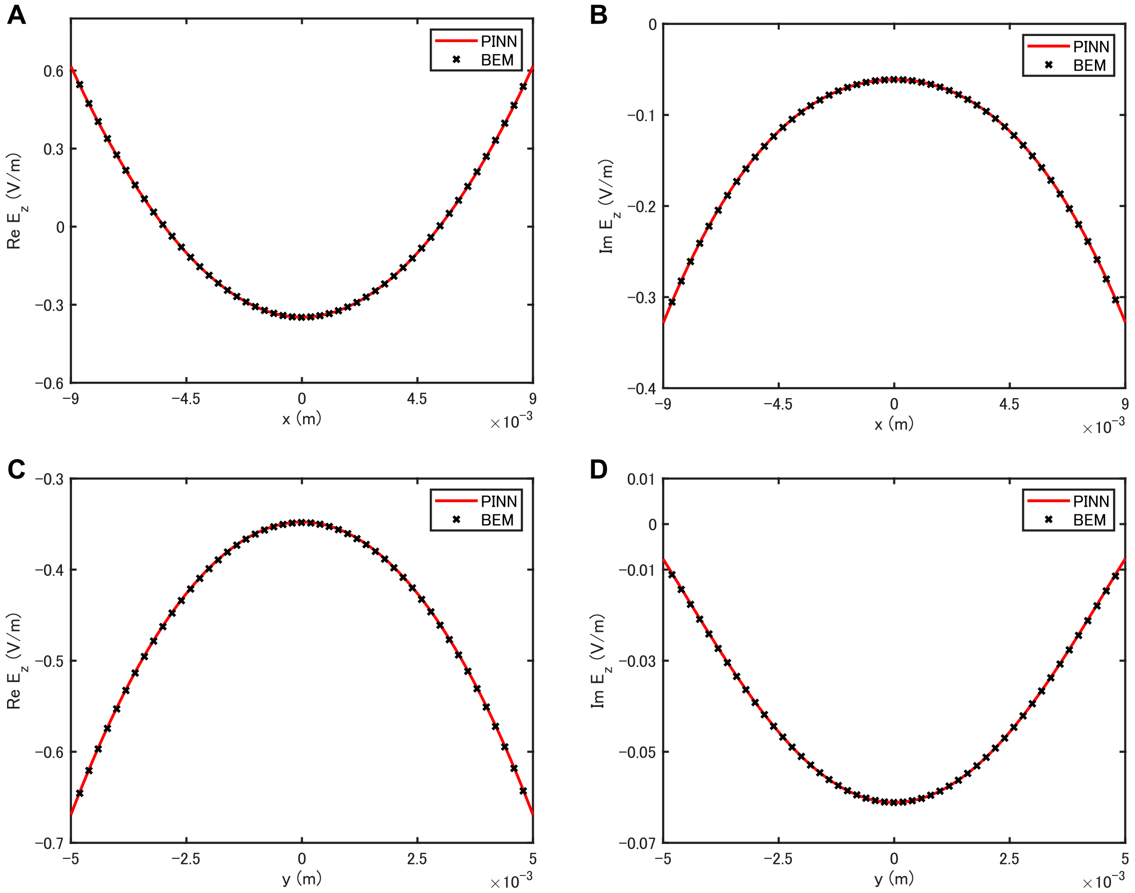


FIGURE 10
Comparison of electric field in a three-layer elliptical vacuum chamber calculated with PINN and BEM **(A)** Re (E_z) along x-axis at y = 0 **(B)** Im (E_z) along x-axis at y = 0, and **(C)** Re (E_z) along y-axis at x = 0 and **(D)** Im (E_z) along y-axis at x = 0.

the copper vacuum chamber is smaller than those of the other two. For each σ_c, at low frequencies, the frequency dependence of the real part is very similar to that of the imaginary part. This feature originates from the frequency dependence of Eq. 23 and also the fact that the magnetic field on the resistive wall is very similar to that on the PEC wall in a limited frequency range. However, at high frequencies, the real part is different from the imaginary one. This is not surprising. From the impedance theory [5], it is known that the above low frequency characteristic is valid for $(|Z_s|\omega b)/(2cZ_0) \ll 1$. In this example, it reads $f \ll 7.2$ THz for Cu and b = 5mm, $f \ll 1.8$ THz for SS and b = 6mm, $f \ll 0.71$ THz for NEG and b = 5mm, respectively. This feature is confirmed in Figure 5. Note that the geometry parameter b and the dc conductivity σ_c are also included in this condition. We also find good agreement between PINN and BEM results in both the real and imaginary parts. This means that the wideband behavior of coupling impedance due to both the geometric effect and the skin effect characterized by Eq. 23 is reproduced in the PINN simulations.

Figure 6 demonstrates a direct comparison of perturbative and nonperturbative models in the coupling impedance for the circular vacuum chamber with the NEG wall as shown in Figure 5. As expected, the perturbative impedance agrees with the nonperturbative one for $f \ll 0.71$ THz while the perturbative impedance is different from the nonperturbative one at higher frequencies. This result means that the magnetic field on the resistive wall is not same as that on the PEC walls at higher frequencies. In other words, the effect of finite conductivity on the magnetic field cannot be negligible as the frequency increases. This demonstrates that the nonperturbative effect is reproduced in the PINN framework.

Next, the coupling impedance of a three-layer circular vacuum chamber is simulated for different inner coating thicknesses. The same domain and boundary sampling points as in Figure 4 and (N_d, N_b)=(1,578, 200) were used here.

Figure 7 shows the coupling impedances of a round Gaussian beam with σ_r = 0.5 mm on the axis of a circular chamber of inner radius 5 mm and outer radius 6 mm with the three-layer structure of SS, Cu and NEG. For each of different NEG thicknesses of 0.01, 0.1, 1 and 10μm, the frequency dependence of the real part of the coupling impedance is quite different from that of the imaginary part. For all NEG layer thicknesses, we find good agreement between PINN and BEM results in both the real and imaginary parts. Note that a characteristic high frequency peak in the real part is observed for the inner layer thickness of 1 μm. This peak was originally found in the impedance theory of the circular multilayer tube [20]. Our result shows the applicability of PINN to the impedance modeling of a circular vacuum chamber with three-layer structure.

Finally, the coupling impedance of an elliptical multilayer vacuum chamber is simulated. The chamber is assumed to have the major axis $a = 9$ mm and the minor axis $b = 5$ mm at the inner surface. This application is not shown in our previous study [28]. For this elliptical chamber, the domain and boundary sampling points shown in Figure 8 are generated with (N_d, N_b) =(2,845, 360).

Figure 9 demonstrates a direct comparison of nonperturbative and perturbative models in the coupling impedance of a Gaussian round beam with σ_r = 0.5 mm for the elliptical vacuum chamber with the same three-layer structure as shown in Figure 2. As expected, the perturbative results (right) are quite different from the nonperturbative ones (right); no resonant peak is observed in the perturbative result. This comparison clearly show even for the three-layer elliptical vacuum chamber the nonperturbative effect is reproduced in the PINN framework.

In the left side of Figure 9, the frequency dependence of the real part of the coupling impedance is different from that of the imaginary part. The highest peak in the real part is at $f \approx$ 0.74 THz. In the bottom of Figure 8, we display the PINN-predicted field distribution E_z over a domain surrounded by the innermost wall at its peak frequency. Unlike the case of a circular chamber in Ref. [28], the field distribution shown here is non-uniform over the elliptical cross section. This feature is also confirmed from the field curves along x- and y-axes in Figure 10. The effect of elliptical cross section on the field is demonstrated in these results.

We find good agreement between PINN and BEM results for both the real and imaginary parts of the impedance in Figure 9 and also for the electric field distributions along the x- and y-axes in Figure 10. These results clearly show the applicability of PINN to multilayer vacuum chambers other than the circular one.

5 Conclusion

This paper has presented a deep learning-based approach for calculating the wakefields of a relativistic charged particle beam with transversally Gaussian charge density distribution in an infinitely long multilayer vacuum chamber with arbitrary cross section. This approach is based on PINN and SIBC of a multilayer structure derived from TL theory. It has been successfully applied to circular and elliptical vacuum chambers with the three-layer structure of SS, Cu and NEG in the nonperturbative model. The PINN results have been cross-checked with the recently developed BEM ones. The effects of chamber geometries and three-layer structure on the coupling impedance have been demonstrated in the framework of PINN.

In this study, a transversally Gaussian charge density distribution is assumed. Therefore, the presented method cannot model a uniform charge density with hard edges, point and ring charges as often used in accelerator physics. A solution to this limitation will be presented in near future. On the other hand, the method has been formulated for two-dimensional problems in the frequency domain. As future works, its extensions to three-dimensional problems and the time domain are under consideration.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Author contributions

KF confirms being the solo contributor of this work and has approved it for publication.

Conflict of interest

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. Jackson JD. *Classical electrodynamics*. New York: Wiley (1998).

2. Chao AW. *Physics of collective beam instabilities in high energy accelerators*. New York: Wiley (1993).

3. Ng KY. *Physics of intensity dependent beam instabilities*. Singapore: World Scientific (2006).

4. Zotter BW, Kheifets SA. *Impedance and wakes in high-energy accelerators*. Singapore: World Scientific (1998).

5. Stupakov G, Penn G. *Classical mechanics and electromagnetism in accelerator physics*. Cham: Springer International Publishing AG (2018).

6. Morton PL, Neil VK, Sessler AM. Wake fields of a pulse of charge moving in a highly conducting pipe of circular cross section. *J Appl Phys* (1966) 37:3875–83. doi:10.1063/1.1707941

7. Zotter B. Longitudinal instabilities of charged particle beams inside cylindrical walls of finite thickness. *Part Accel* (1970) 1:311–26.

8. Henke H, Napoly O. Wake fields between two parallel resistive plates. In: Proc. 2nd European Particle Accelerator Conference; Nice, France (1990). p. 1046–8.

9. Gluckstern R, van Zeijts J, Zotter B. Coupling impedance of beam pipes of general cross section. *Phys Rev E* (1993) 47:656–63. doi:10.1103/physreve.47.656

10. Gluckstern RL. *Analytic methods for calculating coupling impedances*. CERN Report No. 2000-011 (2000).

11. Al-khateeb AM, Boine-Frankenheim O, Hasse RW, Hofmann I. Analytical calculation of the longitudinal space charge and resistive wall impedances in a smooth cylindrical pipe. *Phys Rev E* (2001) 63:026503. doi:10.1103/physreve.63.026503

12. Zimmermann F, Oide K. Resistive-wall wake and impedance for nonultrarelativistic beams. *Phys Rev ST Accel Beams* (2004) 7:044201. doi:10.1103/physrevstab.7.044201

13. Bane KLF, Stupakov G. Resistive wall wakefield in the LCLS undulator beam pipe. In: Proc. 2005 Particle Accelerator Conference. Knoxville, Tennessee: IEEE (2005). p. 3390–2. See also SLAC-PUB-10707.

14. Stupakov G. Resistive-wall wake for nonrelativistic beams revisited. *Phys Rev Accel Beams* (2020) 23:094401. doi:10.1103/physrevaccelbeams.23.094401

15. Doliwa B, Arevalo E, Weiland T. Numerical calculation of transverse coupling impedances: Comparison to spallation neutron source extraction kicker measurements. *Phys Rev ST Accel Beams* (2007) 10:102001. doi:10.1103/physrevstab.10.102001

16. Niedermayer U, Boine-Frankenheim O, Gersem HD. Space charge and resistive wall impedance computation in the frequency domain using the finite element method. *Phys Rev ST Accel Beams* (2015) 18:032001. doi:10.1103/physrevstab.18.032001

17. Burov A, Lebedev V. Transverse resistive wall impedance for multi-layer round chamber. In: Proc. 2002 European Particle Accel. Conf.; Paris, France (2002). p. 1452–4.

18. Vos L. *The impedance of multi-layer vacuum chambers*. CERN-AB 2003-093 (2003).

19. Zotter B. *New results on the impedance of resistive metal walls of finite thickness*. CERN-AB 2005-043 (2005).

20. Ivanyan M, Laziev E, Tsakanov V, Vardanyan A, Heifets S, Tsakanian A. Multilayer tube impedance and external radiation. *Phys Rev ST Accel Beams* (2008) 11:084001. doi:10.1103/physrevstab.11.084001

21. Mounet N, Métral E. Impedance of two dimensional multilayer cylindrical and flat chambers in the non-ultrarelativistic case. In: Proc. HB2010; Morschach, Switzerland (2010). p. 353–7.

22. Macridin A, Spentzouris P, Amundson J, Spentzouris L, McCarron D. Coupling impedance and wake functions for laminated structures with an application to the fermilab booster. *Phys Rev ST Accel Beams* (2011) 14:061003. doi:10.1103/physrevstab.14.061003

23. Migliorati M, Palumbo L, Zannini C, Biancacci N, Vaccaro VG. Resistive wall impedance in elliptical multilayer vacuum chambers. *Phys Rev Accel Beams* (2019) 22:121001. doi:10.1103/physrevaccelbeams.22.121001

24. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys* (2019) 378:686–707. doi:10.1016/j.jcp.2018.10.045

25. Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. *Nat Rev Phys* (2021) 3:422–40. doi:10.1038/s42254-021-00314-5

26. Senior TBA, Volakis JL. *Approximate boundary conditions in electromagnetics*. London: IEE Press (1995).

27. Fujita K. Physics-informed neural network method for space charge effect in particle accelerators. *IEEE Access* (2021) 9:164017–25. doi:10.1109/access.2021.3132942

28. Fujita K. Physics-informed neural network method for modelling beam-wall interactions. *Electron Lett* (2022) 58:390–2. doi:10.1049/ell2.12469

29. Fujita K. Modeling of beam-wall interaction in a finite-length metallic pipe with multiple surface perturbations. *IEEE J Multiscale Multiphys Comput Tech* (2017) 2:237–42. doi:10.1109/jmmct.2017.2786864

30. Fujita K. Impedance computation of cryogenic vacuum chambers using boundary element method. *Phys Rev Accel Beams* (2022) 25:064601. doi:10.1103/physrevaccelbeams.25.064601

31. Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. *Math Program* (1989) 45:503–28. doi:10.1007/bf01589116

32. Ramachandran P, Zoph B, Le Q. Searching for activation functions. In: Proc. ICLR (2018).

33. Aggarwal CC. *Neural networks and deep learning*. Cham: Springer International Publishing AG (2018).

TYPE Original Research
PUBLISHED 17 October 2022
DOI 10.3389/fphy.2022.958120

OPEN ACCESS

EDITED BY
Robert Garnett,
Los Alamos National Laboratory,
United States

REVIEWED BY
Markus Diefenthaler,
Jefferson Lab (DOE), United States
Saravana Prakash
Thirumuruganandham,
Universidad technologica de
Indoamerica, Ecuador

*CORRESPONDENCE
Ryan Coffee,
coffee@slac.stanford.edu

SPECIALTY SECTION
This article was submitted to
Interdisciplinary Physics,
a section of the journal
Frontiers in Physics

RECEIVED 31 May 2022
ACCEPTED 20 September 2022
PUBLISHED 17 October 2022

CITATION
Milan PJ, Rong H, Michaud C, Layad N, Liu Z and Coffee R (2022), Enabling real-time adaptation of machine learning models at x-ray Free Electron Laser facilities with high-speed training optimized computational hardware. *Front. Phys.* 10:958120.
doi: 10.3389/fphy.2022.958120

Enabling real-time adaptation of machine learning models at x-ray Free Electron Laser facilities with high-speed training optimized computational hardware

Petro Junior Milan[1], Hongqian Rong[1], Craig Michaud[1], Naoufal Layad[2], Zhengchun Liu[3] and Ryan Coffee[2,4]*

[1]SambaNova Systems, Inc, Palo Alto, CA, United States, [2]LCLS, SLAC National Accelerator Laboratory, Menlo Park, CA, United States, [3]DSL, Argonne National Laboratory, Lemont, IL, United States, [4]PULSE Institute, SLAC National Accelerator Laboratory, Menlo Park, CA, United States

The emergence of novel computational hardware is enabling a new paradigm for rapid machine learning model training. For the Department of Energy's major research facilities, this developing technology will enable a highly adaptive approach to experimental sciences. In this manuscript we present the per-epoch and end-to-end training times for an example of a streaming diagnostic that is planned for the upcoming high-repetition rate x-ray Free Electron Laser, the Linac Coherent Light Source-II. We explore the parameter space of batch size and data parallel training across multiple Graphics Processing Units and Reconfigurable Dataflow Units. We show the landscape of training times with a goal of full model retraining in under 15 min. Although a full from scratch retraining of a model may not be required in all cases, we nevertheless present an example of the application of emerging computational hardware for adapting machine learning models to changing environments in real-time, during streaming data acquisition, at the rates expected for the data fire hoses of accelerator-based user facilities.

KEYWORDS

AI acceleration, x-ray free electron laser, machine learning, training, AI hardware, gpu, sambanova

1 Introduction

In this manuscript we motivate using machine learning (ML) acceleration engines that enable continuous data acquisition streams to accommodate rapid intermittent self-calibration in order to accommodate both sensor and source variation. Extendable to a wide variety of scientific use cases, there is an explosive trajectory for autonomous and semi-autonomous control systems across the spectrum of accelerator applications [1–3]. Many of these use cases require either real-time model update in a reinforcement learning approach in support of active control systems or they require predictive planning for optimised experimental parameter exploration, e.g. "next shot" planning for DIII-D tokamak.

We concentrate on a particular detector currently under development for the Department of Energy's premiere ultra-high data-rate x-ray Free Electron Laser (xFEL), the Linac Coherent Light Source II (LCLS-II) [4], an angle resolving array of charged particle Time-of-Flight (ToF) spectrometers [5]. For this demonstration we simulate 128 sensor channels of an angular array. Although the current physical system as described in Ref. [5] is comprised of only 20 total sensor angles, our motivation for the oversampling lies in the potential for generalization across multiple domains that share a common tomographic-like image reconstruction/classification pattern.

There is an impending need to develop advanced computational hardware accelerators to process, analyze, and act upon—in real-time—the ultra-high rate data that will stream from the detectors at next generation particle accelerator facilities [6,7]. We foresee a coming broad adoption of transformer models [8–10], originally engineered for language interpretation, that will accommodate situations where unlabeled data is abundant but task-specific labeled data is rare. An inspiring example comes from the extension of GPT-2 to the very different task of music interpretation [11]. This exemplifies the pattern of using a deep encoding network as a structure preserving "featurizer." The resulting latent representation can then feed many downstream models that manage experiment-specific tasks with significantly smaller labeled data sets.

The featurizing, or embedding, model poses a unique challenge for real experimental systems. Although we concentrate here on simulation based results, our motivation is the ultimate physical detector system of electron spectrometers [5]. In such spectrometers, there is a series of electron focusing and retarding electrostatic lenses. Physically, these are copper rings and meshes, each having different high voltage static electric potentials that energetic electrons must pass through on their way to the detection sensor. This forms an electrostatic potential hill for the electrons to climb, shedding the majority of their kinetic energy, such that the drift time in a flat field region more favorably scales with energy. This hill is a compound electrostatic lens with a comparable length to the post-hill drift length with up to 0.25 eV resolution on a 500 eV electron (~2000 resolving power). As such, a simple deterministic equation is quite challenging and would nevertheless exclude the physical particulars of a given detector channel such as high voltage supply variations. The typical for the environmental conditions inside the experimental end-station as well as the changing accelerator operating conditions and user experimental plan changes on the 20–30 min to hourly time-scale. Furthermore, raw data will stream unaltered through the acquisition stream at 0.1% of the full 1 MHz frame rate, at 1 kHz. At this rate, a freshly updated 900k sample-size training set will accumulate every 15 min, thus allowing a fresh update of the embedding model to accommodate the environment, end-station, and accelerator variation. For the sake of such convergence of variation timescales, we target a 15 min model retraining cadence in this manuscript.

In this study, we compare the model training time for a noise eliminating encoder-decoder network, CookieNetAE [12]. We demonstrate the power of parallel training in batches for moving quickly through a large, 900k images, training set. We evaluate a new AI optimized hardware accelerator, the SambaNova Reconfigurable Dataflow Unit$^{\text{TM}}$ (RDU), and compare it to a benchmark DGX node available at the Argonne Leadership Computing Facility (ALCF) with 8 A100 Graphics Processing Units (GPUs). We additionally compare to a more commonly available training engine of up to 8 V100 GPUs also hosted in a single node. We investigate the performance of RDUs *versus* GPUs for scientific ML training applications and discuss our results in the context of high data rate accelerator-based scientific facilities where diagnostics and detectors provide continuous streams of data to keep 1M sample size training sets continuously refreshed.

The remainder of the paper is organized as follows. In Subsection 2.1, we present an overview of the RDU AI accelerator involved in this study and give a description of the distributed training on RDUs in Subsection 2.2. We do not review the GPU accelerator owing to community familiarity with GPUs for ML tasks. In Subsection 2.3, we discuss briefly the data generation process and we give the structure of the CookieNetAE model [12] in Subsection 2.4. We briefly describe the experimental setup in Subsection 2.5. Obtained accuracy and performance results are provided in Subsections 3.1 and 3.2 for single and multi-accelerator cases, respectively, with comparisons with the two generations of GPU. Finally, concluding remarks and thoughts on future work are given in Section 4.

2 Materials and methods

Foreseeing a future of transformer models for distilling information from streaming scientific sensors, we have targeted an encoder-decoder network as a demonstrating case for training acceleration. To relieve concerns about discovery information being lost, we have chosen the downstream task of reconstructing Y', the noise free probability distribution function (PDF) used to produce the under-sampled and grainy X, the input. This reconstruction can only perform well if all of the physically relevant information is contained in the latent representation, otherwise the variation in stochastic sampling of the PDF would dominate the mean-squared-error (MSE) loss between Y and the predicted Y'.

The structure of the encoding side of our CookieNetAE network [12] closely matches that of the embedding side of transformer models, and therefore we take this as a first stage in ML-enabled data featurization for streaming acquisition at so-called data fire hose facilities like the next generation of the Linac Coherent Light Source (LCLS-II) [4,6] and the Upgraded Advanced Photon Source (APS-U) [13]. Although downstream models will be experiment specific, changing on the daily or weekly time scale, the upstream embedding (encoder) layers will be closely tied to the shared detectors. Nevertheless, they will need to accommodate a minutes scale "breathing" of experimental and accelerator conditions. Embedding model retraining must therefore be accelerated to handle such frequent—likely continuous—retraining, thus motivating our exploration of uniquely engineered training accelerators like the SambaNova RDU in comparison to the familiar family of GPU accelerators, Nvidia V100 and A100.

2.1 Sambanova reconfigurable dataflow architecture

We investigated the SambaNova solution for its flexible, dataflow-oriented execution model that enables pipeline operations and programmable data access patterns as will be required of our high velocity data pipelines from streaming scientific detectors. For multi-user facilities, reconfigurability is essential; thus our interest in an architecture that can be programmed specifically for any model application but nevertheless results in an application-specific optimized accelerator. The core of the SambaNova Reconfigurable Dataflow Architecture™ (RDA) [14,15] is a dataflow-optimized processor, the Reconfigurable Dataflow Unit™ (RDU). It has a tiled architecture that is made up of a network of programmable compute (PCUs), memory (PMUs) and communication units. There are 640 PCUs and 640 PMUs connected to one another and the external world *via* the communication units. The PCUs yield a peak performance of over 300 TFLOPs per RDU. The PMUs provide over 300 MB of on-chip memory and 150 TB/s of on-chip bandwidth. These units are programmed with the structure of the dataflow graph that instantiates the ML application, allowing the RDU to use its own parallelism to natively leverage the parallel patterns that are inherent to dataflow graphs.

SambaFlow™ is the framework used to leverage RDUs. As a complete software stack, it takes computational graphs as input from common ML frameworks such as PyTorch [16] and automatically extracts, optimizes, and maps the dataflow graph onto one or more RDUs. SambaFlow achieves performance without the need for low-level kernel tuning.

2.2 Distributed training

The RDA is a scalable solution that not only leverages highly-parallel on-chip computation but also enables parallel computation across multiple RDUs. The SambaFlow framework automatically handles the parallelization used here for data parallel training with the DataScale® platform, a rack-level, datacenter accelerated computing platform. The platform consists of one or more DataScale SN10-8 nodes with integrated networking and management infrastructure in a standards-compliant data center rack—the DataScale SN10-8R [15]. We used up to 4 SN10-8 nodes for the results presented here, each consisting of a host module and 8 RDUs. The RDUs on a node are interconnected *via* the RDU-Connect™ fabric while the multiple SN10-8 nodes communicate *via* Remote direct memory access over Converged Ethernet (RoCE). Beyond the more traditional model parallelism—large models spread across multiple devices—we use the node interconnects to enable data parallelism across all RDUs in the system.

Data parallelism spreads the training workload across multiple accelerator devices, each with its own copy of the full model to be trained. Each device uses the same model with different training samples. For every iteration, each device runs the forward and backward passes to compute gradients on a batch of its respective data. The gradients from all the devices are then aggregated to compute the averaged gradients which are in turn transmitted back to each device to update the local model weights and proceed with the next iteration.

2.3 Data generation

We concentrate on x-ray pulse time-energy reconstruction for two reasons. First, it is one of the more compute-intensive examples of attosecond angular streaking [17], and it is associated with a detector suite that is fully capable of the highest data acquisition rates in the early stages of the LCLS-II [5]. In angular streaking, x-ray induced photoelectron spectra are modulated by the dressing laser field. This modulation can be crudely simulated in the energy domain simply by adding a sinusoidal excess energy to photo-electrons depending on the angle of emission and phase of the dressing laser field. This simplified simulation [18] begins with an electron emission probability distribution,

$$Y_{\text{pdf}} = \sum_{i=0}^{N} p_i\left(\nu_i, \theta, \phi_i\right). \tag{1}$$

The sum over p_i in Eq. 1 runs for each of N sub-spikes where p_i itself is a function of the photon energy (ν_i), the emission angle (θ), and a random phase (ϕ_i) that represents the sub-spike arrival time relative to the optical carrier field of the dressing laser. From this we draw random samples from Y_{pdf}, plus a uniform dark-count likelihood, for every measurement angle θ and sub-spike energy ν_i. These draws form a list of energies X_{hits} for each measurement angle (row in Figure 1A) which are in turn used to update the image X_{img}. Therefore, the goal of CookieNetAE is an optimal inverse mapping from X_{img} (Figure 1A) to Y_{pdf}

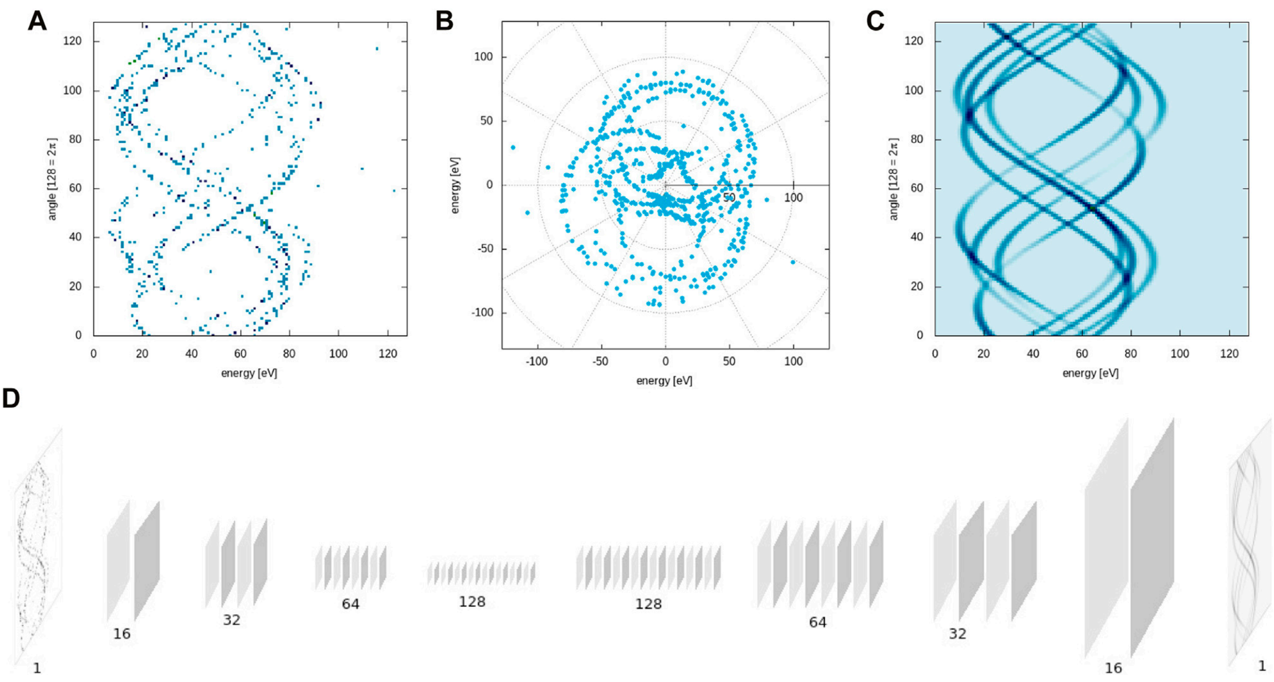


FIGURE 1
(A). Cartesian "image" representation of the in energy-angle space (our input *X*). **(B)**. Polar representation of an example streaking shot. **(C)**. Cartesian "image" representation of the energy-angle emission probability distribution (our output *Y*). **(D)**. Schematic of the CookieNetAE as described in Table 1.

(Figure 1C). In so doing, we have confidence that the latent representation of CookieNetAE (Figure 1D) holds all information that could be used as an input feature vector for any relevant classification or regression task. We note that noise is included insofar as we have dark counts at such a level in physical measurements, but the more insidious difference between actual measurements of $X_{\rm img}$ and that simulated here is that the conversion to energy domain, here a pre-supposition from upstream featurizing algorithm, loses calibration, then the rows of $X_{\rm img}$ would not align, suffering arbitrary relative shifts. Such an artifiact, interestingly, would be effectively trained out by just the sort of adaptation of CookieNetAE by applying the corrective shift to recover smooth sinusoidal curves of $Y_{\rm pdf}$; re-training would therefore accommodate the expected effects of power supply drift and resister failure in the electronics of the individual spectrometers.

Algorithmic xFEL pulse reconstruction requires inverting the angle-resolved photo-electron spectra as per Refs. [17,19] and is much the same as tomographic image reconstruction. A principle challenge here is the high frame rate of the x-ray source. In preparation, we create a simulated dataset of one million example images. The full simulated dataset spans nearly 80 GB and yet represents only one second of acquisition of the LCLS-II. These examples are used with 90% for training and 10% are held out for validation and testing of the CookieNetAE model.

2.4 Encoder-decoder model

CookieNetAE [12] is a convolutional encoder-decoder network designed to infer the angle-energy probability density function of photoionized noble gas electrons. Since we are free in simulation to explore any number of angular samples, we increase the number of evenly distributed angles to 128 around the plane perpendicular to x-ray pulse propagation. In Figure 1A we show an unwrapped (Cartesian) example such that each row in the image corresponds to an energy histogram with 128 energy bins of 1 eV width; panel B shows the polar representation of the electrons emitted. We also show the to-be-recovered $Y_{\rm pdf}$ in panel C with a schematic of the encoder-decoder in panel D (see Ref. [12]). From the input at left in Figure 1D to the output probability distribution at right, we indicate schematically the halving of spatial dimensions and doubling of channels as the filter-depth of layer denoted; this is detailed in Table 1. The encoder contains three convolution layers with the corresponding max pooling layers followed by a single convolution layer to get the latent space representation of the input image. The decoder contains four transposed convolution layers followed by a single convolution layer to get the output. The model is trained by using a standard MSE loss with the following hyperparameters: a maximum number of epochs of 51, a learning rate of 3×10^{-4} for the Adam optimizer, and the ReLU activation function at relevant layers.

TABLE 1 Network structure of the CookieNetAE [12].

Layer	Output shape	Activation
Input	(1,128,128)	–
1st Conv2d	(16,128,128)	ReLU
1st MaxPool2d	(16,64,64)	–
2nd Conv2d	(32,64,64)	ReLU
2nd MaxPool2d	(32,32,32)	–
3rd Conv2d	(64,32,32)	ReLU
3rd MaxPool2d	(64,16,16)	–
4th Conv2d	(128,16,16)	ReLU
1st ConvTranspose2d	(128,32,32)	ReLU
2nd ConvTranspose2d	(64,64,64)	ReLU
3rd ConvTranspose2d	(32,128,128)	ReLU
4th ConvTranspose2d	(16,128,128)	ReLU
5th Conv2d	(1,128,128)	ReLU

The model architecture was chosen for reconstruction fidelity and to serve as a somewhat generic model form representing auto-encoders like a “U-net” but without skip-connects. Avoiding the skip-connections holds closer to a scheme that is consistent with compression at sensor and decompression at acquisition when the model is used in inference at the recording node and was held fixed since the scope of the manuscript is taken to be a survey of data-parallel training across the different training accelerators rather than model architectures. Of course different model forms are likely to instantiate with better or worse performance for different acceleration hardware architectures, and we do indeed plan to investigate this in future research across multiple scientific domains.

2.5 Experimental setup

We compare accelerators by measuring the training time for the SambaNova DataScale SN10-8R in comparison with an Nvidia DGX node of A100 GPUs and another Nvidia node of V100 GPUs. For both the A100 and V100 GPU tests, the CookieNetAE model is run with the PyTorch API v1.9 in data parallel training with Horovod [20]. For the DataScale SN10-8R, the SambaFlow software stack v1.11.2 compiles the model from the same PyTorch reference. For all hardware, we measure the average training time per epoch.

We perform these measurements by examining the dependence of model training time on variation of batch sizes for single accelerator as well as data-parallel training across multiple accelerators. Because the batch size is an important hyperparameter in deep neural networks, we investigate the hardware performance for a wide variety of local and global batch sizes in order to exercise the available design space defined by the number of accelerators and batch size. The local batch size (LBS) is the batch size per device, while the global batch size (GBS) is the batch size across all devices. This design space clearly impacts hardware utilization and convergence characteristics of the learning algorithm [21]. Results from Ref. [22] show that training with batch sizes of 32 samples or smaller can help improve training stability and model generalization. On the other hand, larger batch sizes expose more computational workload per weight update and therefore often result in better hardware utilization. For these reasons, we span a broad range of relevant batch/parallelization parameters. In particular, given the relative novelty of the SambaNova hardware, a more extensive survey of its performance landscape is conducted compared to the GPU landscape.

3 Results

3.1 Single-processor runs

The CookieNetAE model is validated against a held out set of samples not used by the model for parameter updating. A low MSE value of 2.763×10^{-4} is observed for this validation dataset for the case with the reference batch size of 128, indicating that the model performs in the desired manner and without overfitting. Example predictions shown in Figure 2 indicate excellent agreement with the ground truth data.

The variation of the training time per epoch *versus* the batch size is shown in Figure 3A for a single RDU as filled black circles, a single A100 as open black circles, and a single V100 as open black squares (color denotes the number of accelerators used). Numerical values are listed in Tables 3 and 4. For the single RDU case, results are included for batch sizes as few as 8 and as large as 1024, while for the single GPU cases only batch sizes 64–1024 are tested. The training time per epoch is defined as the time taken to run over all the batches in a given epoch. Both the A100 and RDU are roughly twice as fast as V100 in terms of training time, and the RDU is comparable or faster than A100 for all but one batch size, i.e., batch size of 64. The RDU shows a speedup of up to 1.82 times over V100, and up to 1.09 times over A100. Additional results for the RDU with very small batch sizes (i.e., 8, 16 and 32) are also included in Figure 3A and Table 3. We note Ref. [23] demonstrated that FPGA has a performance advantage over GPU on small batch size ResNet-50 inference workloads. Given that RDU and FPGA are both instances of reconfigurable architectures, it remains as an interesting future work to compare very small batch size CookieNetAE training performance between RDU and GPU.

3.2 Multi-processor runs

For fixed LBS of 128, Figure 4 shows the variation of the end-to-end training time *versus* the number of RDUs. This time includes data loading, model initialization, training and validation for 51 epochs along with printing selected output variables to file. As one can see, the end-to-end training time is reduced as the number of RDUs is increased. For example, it is

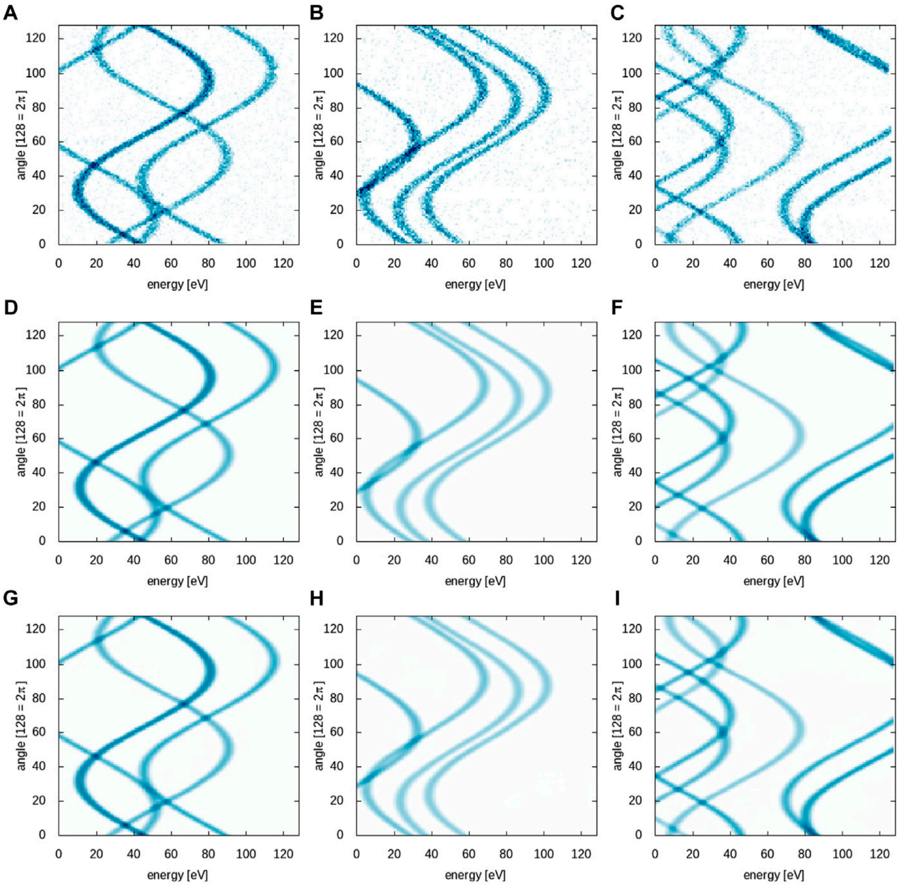


FIGURE 2
Comparison of model input **(A–C)**, ground truth **(D–F)** and model prediction **(G–I)** for three example hold-out test samples from the single-RDU run with a batch size of 128. The MSE values are 6.255×10^{-4}, 3.036×10^{-4}, and 9.162×10^{-4}, respectively.

162 min for the case with one RDU while only 8.3 min for the case with 32 RDUs. This corresponds to a speedup factor of more than 19 times. The parallel efficiency, defined as the ratio of the speedup factor to the number of RDUs, is around 78% of linear scaling for the case of one SN10-8 node with 8 RDUs and only decreases gradually as the number of nodes is increased, reaching 61% for the case of 32 RDUs (4 nodes). From this weak scaling analysis, we observe that two or more SN10-8 nodes (16 or more RDUs) are capable of achieving end-to-end training times in under 15 min. The MSEs for validation data are shown in Table 2, indicating that the accuracy is not significantly affected with the increase in the number of RDUs, and consequently the GBS; the error remains below 5×10^{-4}.

The variation of the training time per epoch *versus* the GBS for multiple RDUs, A100s and V100s is shown in Tables 3 and 4 and visualized in Figure 3. For GPUs, results are shown using up to 8 devices with GBSs up to 1024, while for RDUs results are shown using up to 32 devices with GBSs up to 4096. Both RDUs and A100s outperform V100s in terms of training time for all tested conditions. For the comparison between RDUs and A100s, a mixed picture is observed, where the result depends on the batch size and number of devices. For given GBS and number of devices, when the resulting LBS is small, RDUs outperform A100s, for example, GBS of 64 with 2, 4 and 8 devices, which correspond to LBS of 32, 16 and 8, respectively. For GBS of 128, results are closely comparable between the two architectures, with RDUs slightly faster by a factor of 1.13 for the case with 8 devices. For GBS of 1024, A100s appear to be slightly faster than RDUs by a factor of up to 1.17 for the cases with 2, 4 and 8 devices. Additional performance data for RDUs with 16, 24 and 32 devices are included in Table 3, which correspond to cases with end-to-end training times below 15 min (cf. Figure 4).

The thick blue line in Figure 3A indicates the power law of $x^{-0.89}$ scaling for constant LBS of 128 for RDUs, quite close to inverse scaling, even when the training workload is distributed across multiple SN10-8 nodes. The general trends in Figures 3B–D indicate that RDUs tend to give a performance advantage for smaller LBSs, for example at LBS = 8 (GBS =

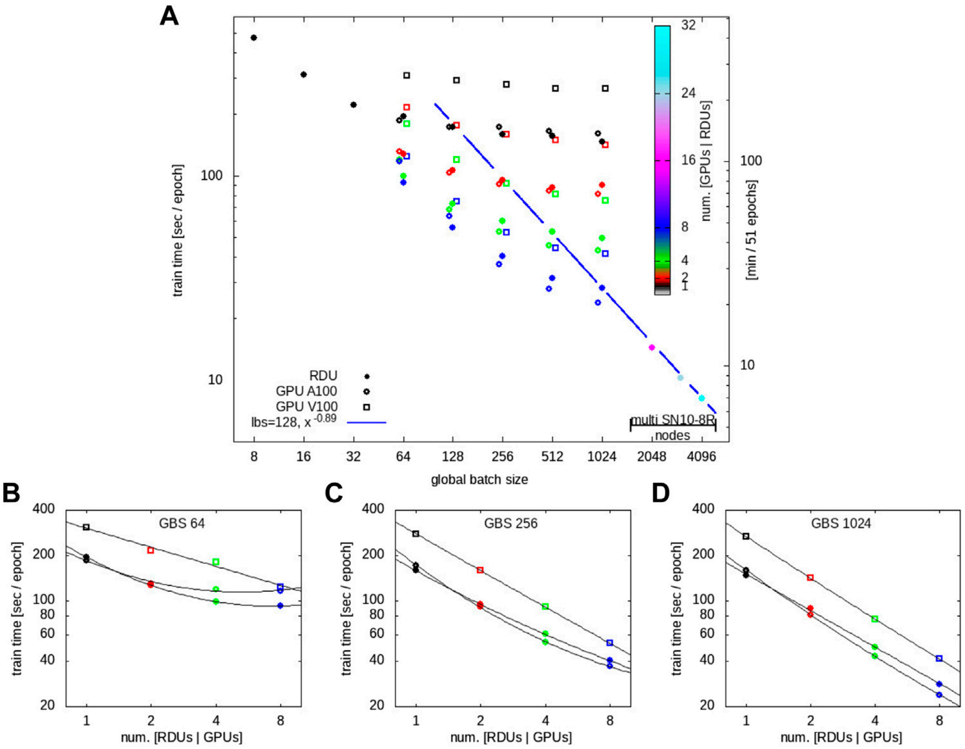


FIGURE 3
(A). Per epoch training time *versus* the global batch size (GBS). The number of accelerators used for data-parallel training is represented by symbol color as indicated in panel A: 1 = black, 2 = red . . . 32 = cyan. Note that we have added a slight "jitter" shift around the *x*-axis value for ease of data point visibility—all GBSs and number of devices are indeed integer powers of 2. The thick blue line indicates that, for a constant LBS of 128, increasing the number of RDUs scales the training time as $x^{-0.89}$. That the V100 (open squares) line up with A100 (open circles) with a color offset of one (a doubling) indicates that for this workload the A100 generally performs twice as fast as the equivalent V100 condition. **(B–D)**. Training time *versus* number of accelerators for GBSs of 64, 256, and 1024 as indicated with symbol color remaining consistent with panel A convention.

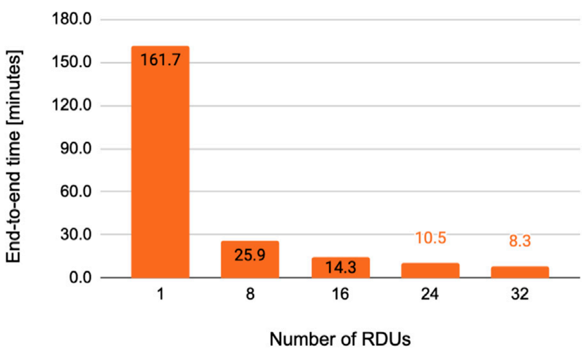


FIGURE 4
Variation of the end-to-end training time with the number of RDUs. This end-to-end time begins with the loading of data from system memory, includes 51 epochs of training into convergence (no early stopping), and return of trained model back to system memory. Results are shown using a fixed LBS of 128 and up to 32 RDUs.

64, num. devs = 8) the RDU has a 1.26x and 1.34x advantage over the A100 and V100, respectively. Overall, we find that the CookieNetAE model can be efficiently trained at scale on multiple RDUs with relative performance *versus* A100 between a 1.26 advantage to a 0.85 disadvantage for GBS of 64 and 1024, respectively. The RDUs therefore comparable to the A100 with a slight disadvantage for GBS above 128 and a performance advantage for GBS at or below 128. These results are promising, point to the benefits for alternative architectures in regions of hyperparameter space, and motivate further research using other scientific ML models.

4 Discussion

The rapid adoption of ML for scientific data processing is triggering an explosion in researchers' appetites for data [24]. An

TABLE 2 MSE values on validation data for multi-RDU runs.

Number of RDUs	GBS	MSE valid
1	128	2.763e-4
8	1024	3.672e-4
16	2047	4.186e-4
24	3072	4.594e-4
32	4096	4.992e-4

often overlooked challenge for such appetites, mining that voluminous data becomes its own challenge for which researchers quickly lose a taste. There has been a growing effort at accelerator-based user facilities to distill sensor data into physics-informed representations automatically in real-time [7]. When physics information is available as a stream of results, one can consider adaptive experimental campaigns that can rapidly explore parameters for faster scientific discovery. In the context of tokamak plasma fusion, this experimental redirection would occur between shots, e.g. one would incorporate the previous shot into an adaptive "shot plan" for the upcoming shots. This shot interval at the DIII-D reactor is 10 min which sets the timescale for our conservative expectation of 10–15 min training time in order to incorporate the last shot results into the adaptive sensor interpretation model for between-shot adaptation. A similar paradigm is an active pursuit for upcoming United States facilities as the LCLS-II [25] and APS-U [13] and existing foreign facilities like the EuroXFEL, each of which faces data acquisition rates that are pushing the limits of what can be transferred continuously over a network. Addressing this challenge, researchers are exploring reduced representations that retain as much of the relevant information contained in the original raw data while mapping to a more information-dense representation for storage and downstream use. This is a particular challenge for scientific cases whereby representation bias could poison data production pipelines. This negative aspect reduces trust of supervised ML approaches both for the fear that it will implicitly exclude novel discovery results or propagate errors undetected.

The attractiveness of recent developments in transformers [8–10] lies in their tendency to be forgiving in situations where task-specific labeled data is rare but unlabeled data, upon which deep embedding models can be trained, is abundant. Scientific use cases, though they rarely involve natural language processing, can in many situations treat multi-sensor data streams as if they were multi-channel audio streams as in analogy with MuseNET of Ref. [11]. The flexibility of transformer architectures to encode general structures into the feature embedding will allow researchers to leverage volumes of unlabeled results at our user facilities, leaving them with significantly less parameters to train with the highly valuable, but scarce, labeled datasets.

This manuscript was driven by our interest in retraining such embedding models frequently in order to accommodate sensor, light-source, and experimental variation. Our embedding models will continuously evolve with a running experiment, at the human and thermal timescale of minutes to hours, while the downstream task specific layers will be constant throughout an experimental campaign—days to weeks. In this case, the task-specific layers are significantly more static than the deep embedding layers. This is very much opposite to transformer use in language tasks. By analogy, imagine the spellings of the words in your vocabulary vary appreciably every 15 min during an extended conversation. This is exactly the case for the deep embedding model in scientific use cases, as the experimental environment varies, so too does the encoded representation of the incoming data stream. Our aspirations to keep on top of these variations, accommodating the experimental variations, is why we have chosen the encoder-decoder CookieNetAE example [12] for our benchmark.

Our measured results demonstrate rapid retraining of the network that is sufficiently deep to capture all relevant information needed for a broad range of domain specific tasks. We show that the expected 15-min scale of experimental evolution can be accommodated with the use of as few as 16 RDUs (two SN10-8 nodes) in data parallel training. Our work finds that the RDU architecture provides for an attractive system, one that is unique from GPUs, that accelerates ML workloads for scientific applications. The general trends seem to point that the RDU represents an

TABLE 3 Strong scaling, training time in seconds for one epoch with different GBSs using RDUs. Cases not carried out are indicated with dashes.

RDUs	GBS = 8	16	32	64	128	256	512	1024	2048	3072	4096
1	471.1	312.5	222.0	195.2	173.7	158.8	156.2	147.1	–	–	–
2	–	–	–	127.2	106.2	94.9	87.4	89.7	–	–	–
4	–	–	–	99.3	73.0	60.2	53.3	49.6	–	–	–
8	–	–	–	92.9	56.0	40.3	31.6	28.1	–	–	–
16	–	–	–	–	–	–	–	–	14.4	–	–
24	–	–	–	–	–	–	–	–	–	10.3	–
32	–	–	–	–	–	–	–	–	–	–	8.2

TABLE 4 Strong scaling, training time in seconds for one epoch with different GBSs using GPUs. The A100s were made available as a single DGX node with NVlink while the V100s were available across the PCI bus of a single node multi-GPU server.

GPU	Num. Devices	GBS = 64	128	256	512	1024
A100	1	186.5	173.3	173.0	165.7	160.2
	2	131.0	103.5	91.3	84.5	81.1
	4	119.9	68.4	53.5	45.5	43.0
	8	117.2	63.5	36.9	28.0	24.0
V100	1	308.4	291.7	278.9	267.7	267.1
	2	215.7	176.2	158.9	149.2	141.8
	4	180.3	119.4	91.4	81.4	75.9
	8	124.7	74.7	52.6	44.3	41.4

advantage to the A100 GPU for training data that is broadly distributed among devices and shallow in each batch while the A100 favors deeper and less broadly distributed training data.

We add to the potential advantage to a shallow and broadly distributed training data, e.g. smaller LBSs and more number of devices [22]. In particular, for ensembles of models, a gradual increase in the variance of outputs could indicate the onset of concept drift. In such a case, having the ability to quickly add new individuals to a larger ensemble of smaller training batches, potentially enlisting additional RDUs for the growing ensemble, would allow for rapid adaptation to this drift. Since accelerator systems like free-electron lasers are typically in a state of fluctuation—much less calm than their synchrotron brethren—they are just such a case for wide, shallow, and dynamic training data sets.

Given the impending TB/s scale of data ingest at the LCLS-II [6], it is imperative that we leverage the trickle of raw—pre-scaled—data for adapting running inference models. Because this new machine [4] and others of its ilk [13] are quickly ramping the data velocity, we expect that an increasing number of users will explore ways to move as much of the pre-processing into the various operations that can be accelerated by dataflow architectures. These accelerators and the downstream user beamlines are dynamic environments where experimental configurations change on the 15 min timescale. The rapid retraining of the associated models could opportunistically leverage intermittent pauses in acquisition, typically every 20–30 min, but only for model retraining that consumes a small fraction of that cadence, e.g. one to few minutes. This cadence is set by the human driven environment at the xFEL where human interpretation of interactive data visualization consumes of order 15 min of collaborator discussion before deciding how best to drive the next steps in experimental campaign. In the fusion case, as noted above, the 10 min of between-shot time likewise sets a natural few shots and then discuss cadence to experimental campaigns. To date, this cadence represents a loose constraint as the results of such diagnostic interpretation models are not yet incorporated into accelerator feedback control systems dynamically, but such plans to feed dynamic model output into machine controls are actively being pursued by this and other groups at FELs and tokamaks. This study serves as a timely impetus for benchmarking short training time with emerging new computational hardware.

In conclusion, the advent of new architectures and the continual improvement in data parallel training will be a win for advancing our accelerator-based scientific user facilities. It will enable the kind of dynamic autonomous control that is required of these large accelerator facilities as is already being incorporated into fields as far reaching as neutron diffraction [1] and magnetic confinement fusion [2]. As the scientific data velocity accelerates in the coming years, and as control systems move to include low-latency high throughput inference, we will find an ever increasing need to match ML acceleration architectures to the scientific facilities that best take advantage of them.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/supplementary material.

Author contributions

PM and HR performed and collected all RDU results and ZL performed and collected all GPU results. NL and RC designed the initial CookieNetAE model architecture while ZL adapted for use in RDU and ThetaGPU. RC built the data simulation code and performed the analysis and presentation of all results. PM and HR contributed equally to the optimization of the RDU results, and CM described the SambaNova system in the manuscript. PM and RC contributed equally to the analysis and presentation of results and composition of the manuscript. HR, CM, and ZL also provided significant contribution to the preparation of the manuscript.

Funding

This work was funded by the Department of Energy, Office of Science, Office of Basic Energy Sciences under Field Work Proposal 100643 “Actionable Information from Sensor to Data Center.” Simulation of the detector was funded by Field Work Proposal 100498 “Enabling long wavelength Streaking for

Attosecond X-ray Science." The Argonne Leadership Computing Facility is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. The Linac Coherent Light Source (LCLS), SLAC National Accelerator Laboratory, is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Contract No. DE-AC02-76SF00515.

Acknowledgments

RNC acknowledges support from the Department of Energy, Office of Science, Office of Basic Energy Sciences for funding the development of the detector array itself under Grant Number FWP 100498 "Enabling long wavelength Streaking for Attosecond X-ray Science." He also acknowledges synergistic support for computational method development by the Office of Fusion Energy Science under Field Work Proposal 100636 "Machine Learning for Real-time Fusion Plasma Behavior Prediction and Manipulation." This research used resources of the Argonne Leadership Computing Facility, a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357, for collecting GPU related benchmarks.

Conflict of interest

Authors Petro Junior Milan, Hongqian Rong and Craig Michaud were employed by SambaNova Systems, Inc.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. McDannald A, Frontzek M, Savici AT, Doucet M, Rodriguez EE, Meuse K, et al. On-the-fly autonomous control of neutron diffraction via physics-informed bayesian active learning. *Appl Phys Rev* (2022) 9:021408. doi:10.1063/5.0082956

2. Degrave J, Felici F, Buchli J, Neunert M, Tracey B, Carpanese F, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature* (2022) 602:414–9. doi:10.1038/s41586-021-04301-9

3. Zhang C, Shao YT, Baraissov Z, Duncan CJ, Hanuka A, Edelen AL, et al. Bayesian optimization for multi-dimensional alignment: Tuning aberration correctors and ptychographic reconstructions. *Microsc Microanal* (2022) 28: 3146–8. doi:10.1017/S1431927622011692

4. Schoenlein R. *New science opportunities enabled by lcls-ii x-ray lasers*. SLAC Report SLAC-R-1053 (2015). p. 1–189.

5. Walter P, Kamalov A, Gatton A, Driver T, Bhogadi D, Castagna JC, et al. Multi-resolution electron spectrometer array for future free-electron laser experiments. *J Synchrotron Radiat* (2021) 28:1364–76. doi:10.1107/S1600577521007700

6. Thayer JB, Carini G, Kroeger W, O'Grady C, Perazzo A, Shankar M, et al. *Building a data system for lcls-ii*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers Inc. (2018). doi:10.1109/NSSMIC.2017.8533033

7. Liu Z, Ali A, Kenesei P, Miceli A, Sharma H, Schwarz N, et al. Bridging data center ai systems with edge computing for actionable information retrieval. In: 2021 3rd Annual Workshop on Extreme-scale Experiment-in-the-Loop Computing (XLOOP). Lemont, IL, USA: IEEE (2021). p. 15–23.

8. Devlin J, Chang M, Lee K, Toutanova K. *BERT: Pre-training of deep bidirectional transformers for language understanding* (2018). *CoRR* abs/1810.04805.

9. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. *Attention is all you need* (2017). doi:10.48550/ARXIV.1706.03762

10. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. *Language models are unsupervised multitask learners* (2018).

11. Payne C. *Musenet* (2019). Available from: openai.com/blog/musenet.

12. [Dataset] Layad N, Liu Z, Coffee RN. *Open source implementation of the cookienetae model* (2022). Available from: https://github.com/AISDC/CookieNetAE.

13. Hansard B. *Advanced photon source upgrade will transform the world of scientific research* (2020). Available from: https://www.anl.gov/article/advanced-photon-source-upgrade-will-transform-the-world-of-scientific-research.

14. SambaNova Systems. *Accelerated computing with a reconfigurable dataflow architecture* (2021). Available from: https://sambanova.ai/wp-content/uploads/2021/06/SambaNova_RDA_Whitepaper_English.pdf.

15. Emani M, Vishwanath V, Adams C, Papka ME, Stevens R, Florescu L, et al. Accelerating scientific applications with sambanova reconfigurable dataflow architecture. *Comput Sci Eng* (2021) 23:114–9. doi:10.1109/MCSE.2021.3057203

16. Paszke AT. Pytorch: An imperative style, high-performance deep learning library. In: H Wallach, H Larochelle, A Beygelzimer, F d'Alché Buc, E Fox, R Garnett, editors. *Advances in neural information processing systems 32*. Red Hook, NY, USA: Curran Associates, Inc. (2019). p. 8024–35.

17. Hartmann N, Hartmann G, Heider R, Wagner MS, Ilchen M, Buck J, et al. Attosecond time–energy structure of x-ray free-electron laser pulses. *Nat Photon* (2018) 12:215–20. doi:10.1038/s41566-018-0107-6

18. [Dataset] Coffee RN. *Cookiesimslim: Slim simulator for lcls-slac cookiebox detector* (2022). Available from: https://github.com/ryancoffee/CookieSimSlim.

19. Li S, Guo Z, Coffee RN, Hegazy K, Huang Z, Natan A, et al. Characterizing isolated attosecond pulses with angular streaking. *Opt Express* (2018) 26:4531–47. doi:10.1364/OE.26.004531

20. Uber I. *Horovod* (2022). Available from: https://github.com/horovod/horovod.git.

21. Goodfellow I, Bengio Y, Courville A. *Deep learning*. Cambridge, MA, USA: MIT Press (2016). Available from: http://www.deeplearningbook.org.

22. Masters D, Luschi C. *Revisiting small batch training for deep neural networks* (2018). doi:10.48550/ARXIV.1804.07612

23. Duarte J, Harris P, Hauck S, Holzman B, Hsu SC, Jindariani S, et al. Fpga-accelerated machine learning inference as a service for particle physics computing. *Comput Softw Big Sci* (2019) 3:13–5. doi:10.1007/s41781-019-0027-2

24. Sanchez-Gonzalez A, Micaelli P, Olivier C, Barillot TR, Ilchen M, Lutman AA, et al. Accurate prediction of x-ray pulse properties from a free-electron laser using machine learning. *Nat Commun* (2017) 8:15461. doi:10.1038/ncomms15461

25. Corbeil Therrien A, Herbst R, Quijano O, Gatton A, Coffee R. Machine learning at the edge for ultra high rate detectors. In: 2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC). Manchester, UK: IEEE (2019). p. 1–4. doi:10.1109/NSS/MIC42101.2019.9059671

frontiers | Frontiers in Physics

TYPE Brief Research Report
PUBLISHED 18 October 2022
DOI 10.3389/fphy.2022.955555

Predicting beam transmission using 2-dimensional phase space projections of hadron accelerators

Anthony Tran[1], Yue Hao[1]*, Brahim Mustapha[2] and Jose L. Martinez Marin[2]

[1]Facility for Rare Isotope Beams, Michigan State University, East Lansing, MI, United States, [2]Argonne National Laboratory, Argonne, IL, United States

Check for updates

OPEN ACCESS

EDITED BY
Robert Garnett,
Los Alamos National Laboratory (DOE), United States

REVIEWED BY
Yuri Batygin,
Los Alamos National Laboratory (DOE), United States
Gianlica Valentino,
University of Malta, Malta

*CORRESPONDENCE
Yue Hao,
haoyue@msu.edu

SPECIALTY SECTION
This article was submitted to Interdisciplinary Physics, a section of the journal Frontiers in Physics

RECEIVED 28 May 2022
ACCEPTED 30 September 2022
PUBLISHED 18 October 2022

CITATION
Tran A, Hao Y, Mustapha B and Martinez Marin JL (2022), Predicting beam transmission using 2-dimensional phase space projections of hadron accelerators.
Front. Phys. 10:955555.
doi: 10.3389/fphy.2022.955555

We present a method to compress the 2D transverse phase space projections from a hadron accelerator and use that information to predict the beam transmission. This method assumes that obtaining at least three projections of the 4D transverse phase space is possible and that an accurate simulation model is available for the beamline. Using a simulated model, we show that—a computer can train a convolutional autoencoder to reduce phase-space information which can later be used to predict the beam transmission. Finally, we argue that although using projections from a realistic nonlinear distribution produces less accurate results, the method still generalizes well.

KEYWORDS

virtual accelerator, convolutional autoencoder, neural network, autoencoder, beam transmission, hadron accelerator, phase space projections, nonlinear field

1 Introduction

A challenging problem in obtaining high beam power in hadron linacs, such as ATLAS, SNS, and FRIB, is understanding and minimizing uncontrolled beam loss—a significant unexpected loss of the beam within the beamline. [1]. In the low energy beam transport lines (LEBT), the machine must carefully control the beam to minimize downstream beam loss. The beam describes a collection of particles in six-dimensional space; three positions and three momentum coordinates. For the DC beam in the LEBT, the longitudinal coordinates are not involved in the dynamics but appear as parameters. Therefore, each charged particle is described by its location in the four-dimensional (4D) transverse phase space (x, x', y, y'), where primed coordinates are derivatives with respect to the longitudinal direction.

In the LEBT, multiple beam measurement devices such as Alison Scanners [2], Pepper-Pot emittance meters [3], wire scanners [4], and viewers, are used to capture one-dimensional (1D) or two-dimensional (2D) profile measurements. These are projections of the four-dimensional (4D) transverse phase space. Inferring the 4D distribution from this projected 1D and 2D information is referred to as 4D tomography. Mathematical and physical methods, such as the maximum entropy

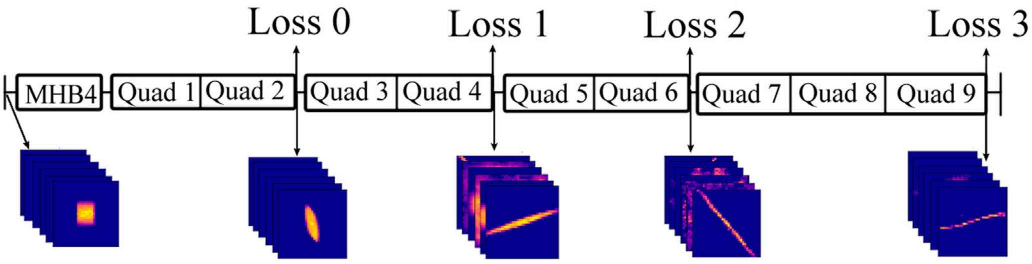


FIGURE 1
Cartoon of accelerator and beam measurements. The image shows where the model collected each beam measurement.

principle [5, 6], have been successfully demonstrated to realize the 4D tomography in accelerators. However, there are still challenges in combining 1D or 2D information from different locations. The optics deviation, when the machine deviates from the model, for example, will affect the accuracy of the 4D tomography.

In this paper, we tested a data-driven approach to predict the beam loss using 4D phase space distribution information encoded in a low-dimensional vector from 2D projection measurements. The data was generated from virtual diagnostic instruments simulated using the beam dynamics code TRACK. The simulation is of a test lattice adopted from the LEBT of the ATLAS accelerator, which consists of six quadrupoles and five virtual diagnostic instruments. The simulation results were used to develop a convolutional autoencoder to encode the data into a meaningful lower-dimensional representation, which the model will then use to relate the phase-space information to the beam loss.

2 Methods

2.1 ATLAS lattice

The presented study used data generated from the simulation of ATLAS's LEBT. The lattice consists mainly of a multi-harmonic buncher, nine quadrupole magnets—six of which are being used to tune the lattice—and five virtual diagnostic instruments. The virtual diagnostic instruments capture the 4D phase space of the beam, providing information on the 2D projections and beam transmission for use later. This amount of data is currently hard to achieve in an actual accelerator, but it is used to study the method's feasibility, giving the initial model a higher chance to succeed.

Figure 1 shows the location of the virtual diagnostic instruments. The measurements were recorded at five locations: beginning, end, and after quadrupoles 2, 4, and 6.

2.2 Generating data using TRACK

TRACK is a ray-tracing or particle-tracking code that can:

1) represent external fields accurately within the aperture.
2) calculate the particle coordinate at any point in the space.
3) determine to calculate beam loss in both the ideal case and in the presence of complex field errors and device misalignments [7].

Since machine time is costly, the TRACK simulation was used to gather data. It was generated on Michigan State University's high-performance computing cluster for a week, producing over a million data points. As will be covered in a later section, a significant amount of data will be required for training autoencoders to high fidelity. The model varied the parameters for these simulations according to Table 1. The model chose these parameters within a small range to interpolate well. Once the model collected the data, it was filtered so that the initial beam distributions were contained within the beam aperture, resulting in a final data set of around 430,000 simulation points.

2D phase-space projections from the 4D phase space were created by having all the particles deposited onto an $n \times n$ grid using pairs of the coordinates axes, (x, x', y, y'). This method resulted in 6 independent projections.

2.2.1 Non-linear field

Another data set was created to test the model's generalizability, which will be explained later. This was done with a perturbation to the initial distribution. The distribution was created at the beginning of the simulation by using a nonlinear magnetic field, such as a sextupole. All other generative settings were kept the same. This simulates the realistic case of ECR beams which experience a sextupole field inside the source.

TABLE 1 Inputs: Parameter range used to generate a dataset of the initial beam distributions and quadrupole settings. Architecture: Description of some blocks used in the architecture. Each projection was given a separate encoder and decoder block where the latent dimension differ for the (x, x') and (y, y') projections.

Dataset

Inputs	
Voltages on Quadrupoles 1, 3, 5	uniform random number from [0,8] V
Voltages on Quadrupoles 2, 4, 6	uniform random number from [-8,0] V
Initial Distribution	random distribution from 9 built-in distributions
$\epsilon_{x,y}$	0.12 + *Normal* ($\mu = 0$, $\sigma = 0.012$) cm*mrad
$\alpha_{x,y}$	*Normal* ($\mu = 0$, $\sigma = 1$) unitless
$\beta_{x,y}$	100 + *Normal* ($\mu = 0$, $\sigma = 10$) cm/rad
Outputs	
Number of particles left	0-10,000 particles. Taken at 4 different points
Position of all particles	Taken at 5 different points
Architecture	
Encoder	
Conv 1	channel-out: 1, channel-in: 64, kernel: 3, stride: 3
Conv 2	channel-out: 64, channel-in: 128, kernel: 3, stride: 3
Linear	in: 6,272, out: (x, x'), $(y, y') \rightarrow 32$, (x, y) (x, y') (y, x') $(x', y') \rightarrow 16$
Decoder	
Linear	in: (x, x'), $(y, y') \rightarrow 32$, (x, y) (x, y') (y, x') $(x', y') \rightarrow 16$, out: 6,272
ConvTranspose 1	channel-in: 128, channel-out: 128, kernel: 3, stride: 2
ConvTranspose 2	channel-in: 128, channel-out: 64, kernel: 3, stride: 2
ConvTranspose 3	channel-in: 64, channel-out: 1, kernel: 3, stride: 1
Latent Dimension	
Linear	in: 134, out: 134
NN 1–4	
Linear	in & out: (x, x'), $(y, y') \rightarrow 32$, (x, y) (x, y') (y, x') $(x', y') \rightarrow 16$
LossP	
Linear 1	in: 134, out: 134 * 4
Linear 2	in: 134 * 4, out: 134 * 4
Linear 3	in: 134 * 4, out: 1

2.3 Autoencoder

An autoencoder is a nonlinear data reduction algorithm used in machine learning. It is composed of two parts, an encoder and a decoder. The encoder takes a significant input and reduces it into a lower dimension, known as a latent dimension, while the decoder attempts to reconstruct the latent dimension back into the original input. The error, which is the difference between the original and reconstructed data, quantifies how well the latent dimension explains the initial input. The advantage of compressing the data into a meaningful representation [8] makes it more efficient to train a neural network model on the reduced data.

In the model, a convolutional autoencoder was implemented in PyTorch [9] to reduce the dimension of the 2D projections of the phase space. A convolutional autoencoder uses a convolutional neural network as the encoder and decoder. A convolutional neural network is a type of neural network used to analyze visual information [10, 11]. This network has the

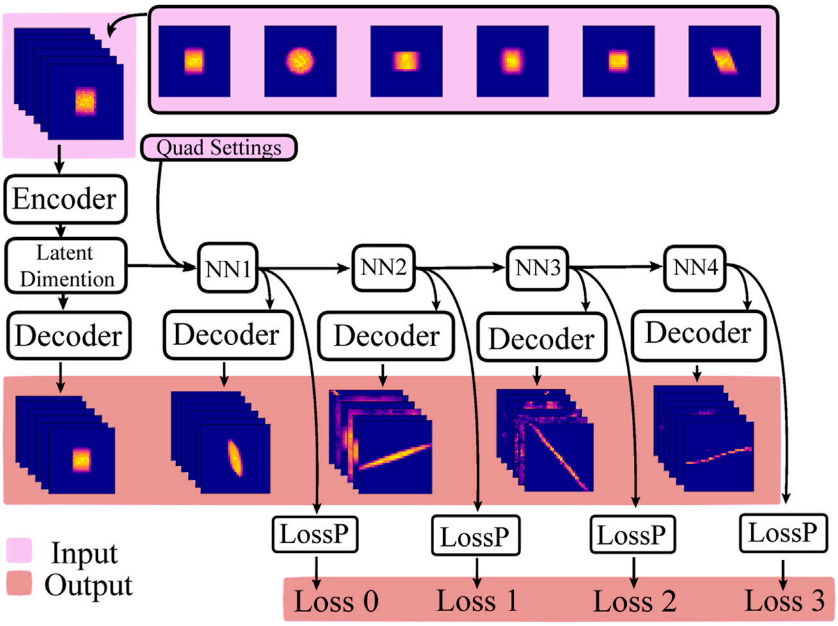


FIGURE 2
Cartoon of architecture. During training, the model takes all the 2D projections and loss value as input into the training. Only the initial 2D projections were given during testing, and the model predicts the loss values and 2D projections in addition.

advantage over principal component analysis [12], another data reduction algorithm, in that it includes spatial information and can account for nonlinear effects by using nonlinear activation functions in the network. Activation functions take the summed weighted inputs of a layer inputs of a layer and map it onto a set range. It was found that the ReLU activation function and the ELU activation function were the best activation functions to use [13] in this case, it helps the model to train fast and be less likely to fail during training.

Each of the six 2D projections was given its own autoencoder. The decoder reproduces all the original projections reasonably, verifying that the model effectively encoded the projections into a latent dimension. The latent dimensions sizes used for this paper were 32 for the (x, x'), and (y, y') projection, and 16 for the rest. Given that the code made the original images to be 33 × 33, the model significantly reduced the inputs.

2.4 Modeling

A neural network was used to create a surrogate model of the ATLAS front-end, as shown in Figure 2. A more detailed description of each block is described in Table 1, and the full detail can be seen in the code[1]. The architecture is composed of an encoder-decoder block to reduce each of the six phase-space projections into lower latent dimensions and then concatenate them together. The quad settings were also joined onto this vector and processed through fully connected layers. Each fully connected layer attempts to model the phase space changes by transforming the latent variables into a different set of latent variables describing the new 4D phase space. This new vector would be the input into the next fully connected layer to model the following transformation, a decoder block to reconstruct the distribution as 2D phase-space projections at that location, and another fully connected layer to predict the beam transmission represented by the number of particles left.

The encoder-decoder block uses a convolutional autoencoder, as described in the previous section. The decoder was built similarly to the encoder, the only difference being that some adjustments were made to match the original dimensions. A decoder was not trained for every location but was combined for each projection. This method saves limited GPU memory and produces a more generalized decoder.

The model used a two-layer, fully connected network to calculate the number of particles left. Again, the model did not train the network at every location, but it was combined to make a generalized particle loss predictor for the same reasons stated above.

2.5 Training

Overall, the model encodes each initial phase-space projection into separate latent dimensions, then attempts to recreate the phase-space forecast and predict the beam transmissions at the other

[1] The code is available upon request.

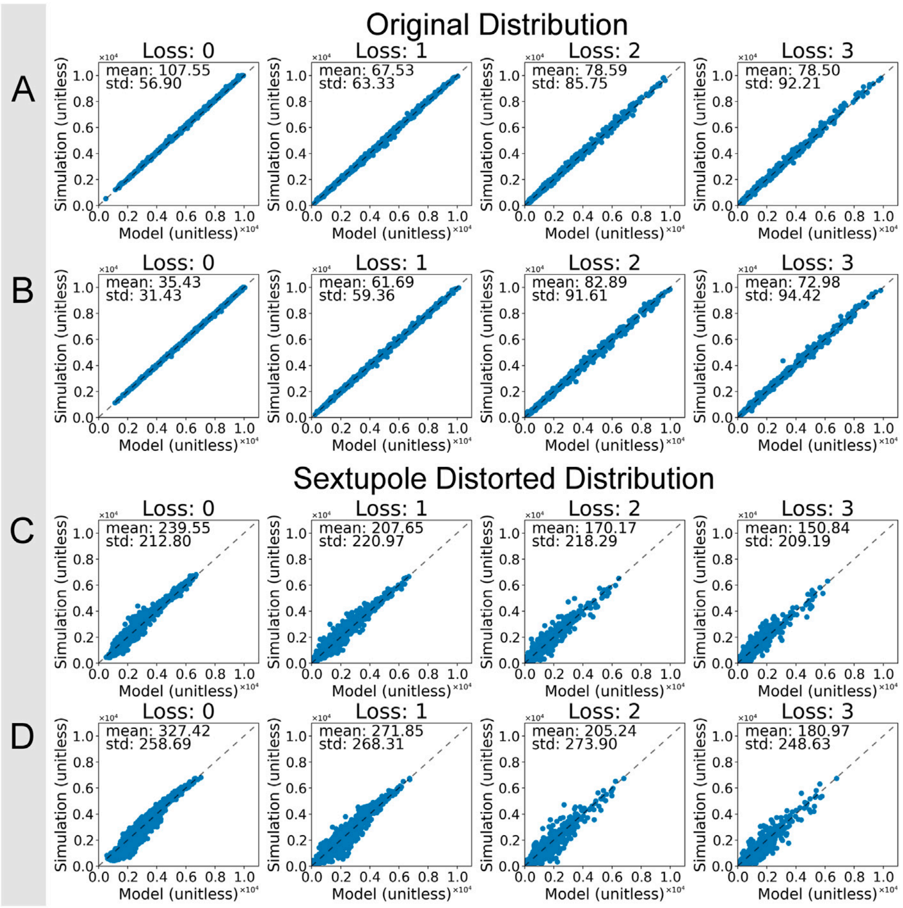


FIGURE 3
(A) Histogram of original data set using six projections **(B)** and same model but using three projections. **(C)** Histogram of original data set using six projections, **(D)** and same model but using three projections.

4 locations, and then compares them to the ground truth from the dataset. Using both results in a training loop, the model will update itself using gradient descent [10] to recreate the image in the next iteration better. Gradient descent is performed using backpropagation on PyTorch computational map. The basic idea is that parameters in the network will be adjusted using the gradient information in the direction that will result in the final output closer to the actual values. The size of the step taken is related to the learning rate.

The model used a loss function to quantify the difference between the predicted and ground truth results. Commonly, mean square error loss is used because it punishes large deviations; however, this results in an overflow—an error that occurs when a computer produces a number more significant than it can represent, in the gradient calculation during training. This could have be resolved by normalizing the inputs and outputs, but a non-normalized dataset was used since that gave a better convergence. The model used absolute loss (L1 Loss) in these cases. This could also have been resolved by using a double or a long float, but this would use up valuable memory on the GPU.

To also aid in training, frozen layers were implemented to decrease training time and prevent gradient overflow. Frozen layers are layers where the gradient information was disconnected, thus preventing changes to that layer during training. This method was implemented in the following three-step training procedure:

1. LossP blocks were frozen and trained with a learning rate of 0.01. Everything else was trained for 20 epochs.
2. LossP blocks were unfrozen, and everything else was frozen; thus, only LossP was trained for 10 epochs. The learning rate was also 0.01
3. Everything was unfrozen and trained for 5 epochs at a lower learning rate of 0.001.

A problem that arises from using simulation is overfitting, which is a state where the model memorizes the training data rather than generalizing it. Because simulation generally differs from the actual machine due to installation and operation errors, if the model is not generalized or adjusted enough, it will perform poorly on the actual device.

The model used the last step of the training procedure to better merge the LossP block with the whole model. This also helps to prevent overfitting. A lower learning rate than the previous two steps discourages radical changes that may destroy the learned model in the first two steps.

2.6 Results

The model was tested on a newly generated dataset using the original parameters and a nonlinear dataset generated using a sextupole. Only the initial distributions were given, but the model would still predict the 2D projections and beam transmission at the other locations downstream. Then, to test the model's generalization, a nonlinear field in the form of a sextupole [14] was added to the beginning of the simulation to generate a distinct subset of inputs.

An error of less than 1% for results within two standard deviations from the mean would be sufficiently good for predicting the loss on ATLAS since it is a low-power machine. For the rest of the paper, the percentage refers to a two standard deviation bound. The error is defined as the absolute difference between the ground truth and the predicted values divided by the total number of particles. The obtained values were plotted as a correlation graph in Figure 3A. If there were no errors, there would be a straight line. Given that we have 10^4 particles, the error for the original data set using six projections would be 3%.

This was then tested on the nonlinear sextupole distribution with fair results, an error of 5.5% as shown in Figure 3C. The model was able to generalize reasonably well; however, it is still far from the ideal case. In this case, a machine learning model mainly interpolates the results, so the accuracy of a model depends on how much training data it is given. The more data points a model has, the better the interpolation.

Due to the nature of hadron accelerators, many quadrupole configurations would produce a high particle loss because only a few configurations would allow most particles to pass. Thus, most of the dataset would be skewed towards high loss, resulting in higher accuracy since there is more data in those cases. The dataset was split into bins, and as expected, the bin of particle loss between 9,000–10,000 has an error around 2.5%, and for the bin of particle loss between 0–1,000, the error was as high as 5%.

2.6.1 Testing on a smaller data set

The same model was tested again, but with the (x, y') (x', y'), and (y, x') projections removed. In Figure 3B, the error predictions from the original data set show an improvement in the accuracy for "Loss: 0", while it has around the same error for the other losses. This difference in error is likely due to overfitting as the predictions from the nonlinear data set show a loss of accuracy overall, as seen in Figure 3D; however, the model was shown to work with half the image data used, making this model more practical.

3 Discussion

A proof-of-principle machine learning-based model has been reported to test an ML-based 4D tomography using its 2D projections and the capability to predict beam transmission. The result shows that if given only three projections of the 4D phase space, the model can reduce the projections into a smaller latent dimension that contains the core information, which the model can then use to predict the beam transmission downstream. If the model used fewer projections, the model would not have enough information to describe the entire 4D phase space. The latent dimension was verified to have contained the core information through a decoder that correctly reconstructed the encoded images. This method generalizes reasonably well to initial beam distributions with nonlinear perturbations, showing robustness and the potential for modeling the real machine.

Before bringing this to a real machine, it should be noted that this is a simplified model of an actual accelerator, with considerable differences to consider. This model assumes that a single parameter can model the accelerator elements; therefore, more complicated effects, such as misalignment of the magnets and the longitudinal overlapping of transverse magnets, are not considered. The model also assumes no space charge, neglects the profile's beam-to-beam fluctuation, and overlooks the longitudinal components of the beam distribution.

It requires at least three projections to make the prediction, but this is not always available. In addition, it assumes perfect accuracy in the measurements. The prediction accuracy will also depend on the accuracy of the projections and beam loss monitors, but one method to reduce measurement error at various locations can be to use multiple measurements with different optics. The current model cannot predict fractional beam loss over the entire length of the accelerator at the level of 1e-06 to 1e-08 per meter, which would be desirable for future works, but methods such as adaptive tuning and physics-informed learning may prove helpful to make the model more robust and accurate and eventually reach that goal.

The beam loss value is represented by the percentage of the total beam intensity. Accuracy of a few percent beam loss has been demonstrated with this simplified model; however, characterizing the beam loss with a certain percentage of the total power results in a very different threshold for accelerators with another goal of beam power. Therefore, the absolute power loss should be used in later applications for real accelerators.

The traditional approach to evaluating beam loss would be to use simulations or beam loss monitors. The simulation requires an accurate lattice model with boundary data information where beam halo information is usually absent. Other methods, such as beam loss monitors, are physical devices used to measure beam loss, whereas some standard devices are ion chambers and photomultipliers. In choosing the suitable machine, there are various considerations such as sensitivity, cost, size, dynamic range, and radiation hardness [15].

The model could be trained using data from a real machine, but collecting a large amount of data is expensive and time-consuming. Thus, models will have to be introduced on realistic simulation models and then transferred to the actual machine. Methods known as "transfer learning" allow knowledge learned from the source dataset to be transferred to a target dataset [10]. This could be done, for example, by freezing the model, adding another layer to the encoder and decoder, and training that layer on the distribution from the machine to adapt the model to the machine. Then, the rest of the model can be unfrozen and trained with a much lower learning rate to fine-tune the model. Further studies will allow models to be trained first with simulations and then transferred to machines.

Machine learning has various advantages and disadvantages. Machine learning has an advantage since it can make a data-driven model with no lattice information at the expense of a large data set. Two disadvantages are that the model will need high-fidelity data to reach high accuracy and that the model will constantly need to be calibrated to the current machine. Future research will study ways to combine machine learning with physics to make a more sample-efficient and robust model. One way this could be done is by encoding constraints in the loss function during model training or by incorporating domain knowledge by including the transfer matrices in the calculation. The positive results of this work give hope that incorporating this knowledge may increase sample efficiency and further reduce the beam transmission error.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

AT did the data generation, the analysis, and the writing of this paper. YH advised AT along the way and revised the paper. BM helped revise the paper for better understanding. JM created the initial wrapper to run TRACK in python.

Funding

This work was supported by the U.S. Department of Energy, under Contract No. DE-AC02-06CH11357. This research used the ATLAS facility, which is a DOE Office of Nuclear Physics User Facility. Funded through a "Data, Artificial Intelligence, and Machine Learning at DOE Scientific User Facilities" Grant from the DOE's Office of Nuclear Physics.

Acknowledgments

I acknowledge Jaturong Kongmanee for multiple discussions related to machine learning.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. Aleksandrov A, Shishlo A. Path to beam loss reduction in the sns linac using measurements, simulation and collimation. In: 57th ICFA Advanced Beam Dynamics Workshop on High-Intensity, High Brightness and High Power Hadron Beams; Malmö, Sweden (2016).

2. Allison PW, Sherman JD, Holtkamp DB. An emittance scanner for intense low-energy ion beams. *IEEE Trans Nucl Sci* (1983) 30:2204–6. doi:10.1109/tns.1983.4332762

3. Pikin A, Kponou A, Ritter J, Zajic V. Pepper pot emittance meter. In: *Tech. rep.* Upton, NY, United States: Brookhaven National Lab. (BNL) (2006). Relativistic Heavy.

4. Koziol H. Beam diagnostics for accelerators. In: *Tech. rep.* Meyrin, Switzerland: CERN (2001).

5. Reggiani D, Seidel M, Allen C. Transverse phase-space beam tomography at psi and sns proton accelerators. In: Proceedings of IPAC10; Kyoto, Japan (2010).

6. Wong JC, Shishlo A, Aleksandrov A, Liu Y, Long C. 4d transverse phase space tomography of an operational hydrogen ion beam via noninvasive 2d measurements using laser wires. *Phys Rev Accel Beams* (2022) 25:042801. doi:10.1103/physrevaccelbeams.25.042801

7. Aseev V, Ostroumov P, Lessner E, Mustapha B. Track: The new beam dynamics code. In: Proceedings of the 2005 Particle Accelerator Conference (2005). p. 2053–5. doi:10.1109/PAC.2005.1591006

8. Bank D, Koenigstein N, Giryes R. *Autoencoders* (2020). *arXiv preprint arXiv:2003.05991*.

9. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. Pytorch: An imperative style, high-performance deep learning library. *Adv Neural Inf Process Syst* (2019) 32.

10. Zhang A, Lipton ZC, Li M, Smola AJ. *Dive into deep learning* (2021). *arXiv preprint arXiv:2106.11342*.

11. Scheinker A. Adaptive machine learning for time-varying systems: Low dimensional latent space tuning. *J Instrum* (2021) 16:P10008. doi:10.1088/1748-0221/16/10/p10008

12. Abdi H, Williams LJ. Principal component analysis. *Wires Comp Stat* (2010) 2:433–59. doi:10.1002/wics.101

13. Ding B, Qian H, Zhou J. Activation functions and their characteristics in deep neural networks. In: 2018 Chinese control and decision conference (CCDC). Shenyang, China: IEEE (2018). p. 1836–41.

14. Lee SY. *Accelerator physics*. Singapore: World Scientific Publishing Company (2018).

15. Wittenburg K. Beam loss monitoring and control. In: Proceedings of EPAC (2002). p. 109–13.

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 13 December 2022
DOI 10.3389/fphy.2022.957509

EdgeAI: Machine learning *via* direct attached accelerator for streaming data processing at high shot rate x-ray free-electron lasers

Mike Kraus[1], Naoufal Layad[2], Zhengchun Liu[3] and Ryan Coffee[2,4]*

[1]Graphcore, Inc., Palo Alto, CA, United States, [2]LCLS, SLAC National Accelerator Laboratory, Menlo Park, CA, United States, [3]DSL, Argonne National Laboratory, Lemont, IL, United States, [4]PULSE Institute, SLAC National Accelerator Laboratory, Menlo Park, CA, United States

OPEN ACCESS

EDITED BY
Robert Garnett,
Los Alamos National Laboratory (DOE),
United States

REVIEWED BY
Christine Darve,
European Spallation Source, Sweden
Saravana Prakash
Thirumuruganandham,
Universidad technologica de
Indoamerica, Ecuador
Markus Diefenthaler,
Jefferson Lab (DOE), United States

*CORRESPONDENCE
Ryan Coffee,
coffee@slac.stanford.edu

SPECIALTY SECTION
This article was submitted to Interdisciplinary Physics, a section of the journal Frontiers in Physics

RECEIVED 31 May 2022
ACCEPTED 16 November 2022
PUBLISHED 13 December 2022

CITATION
Kraus M, Layad N, Liu Z and Coffee R (2022), EdgeAI: Machine learning *via* direct attached accelerator for streaming data processing at high shot rate x-ray free-electron lasers.
Front. Phys. 10:957509.
doi: 10.3389/fphy.2022.957509

We present a case for low batch-size inference with the potential for adaptive training of a lean encoder model. We do so in the context of a paradigmatic example of machine learning as applied in data acquisition at high data velocity scientific user facilities such as the Linac Coherent Light Source-II x-ray Free-Electron Laser. We discuss how a low-latency inference model operating at the data acquisition edge can capitalize on the naturally stochastic nature of such sources. We simulate the method of attosecond angular streaking to produce representative results whereby simulated input data reproduce high-resolution ground truth probability distributions. By minimizing the mean-squared error between the decoded output of the latent representation and the ground truth distributions, we ensure that the encoding layers and resulting latent representation maintains full fidelity for any downstream task, be it classification or regression. We present throughput results for data-parallel inference of various batch sizes, some with throughput exceeding 100 k images per second. We also show *in situ* training below 10 s per epoch for the full encoder–decoder model as would be relevant for streaming and adaptive real-time data production at our nation's scientific light sources.

KEYWORDS

machine learning, edge computing, AI hardware, low latency, x-ray, free-electron laser, GPU, Graphcore

1 Introduction

Among the leading major scientific facilities in the US Department of Energy's portfolio is the Linac Coherent Light Source (LCLS). As the world's first hard x-ray free-electron laser (xFEL), its ultra-short x-ray pulses, shorter than typical periods for most molecular vibrations, allow its international research users to peer into the inner workings of some of nature's key chemical reactions [1]. The very high peak brightness of these x-ray pulses also allow for imaging of the interior structure of material in extreme

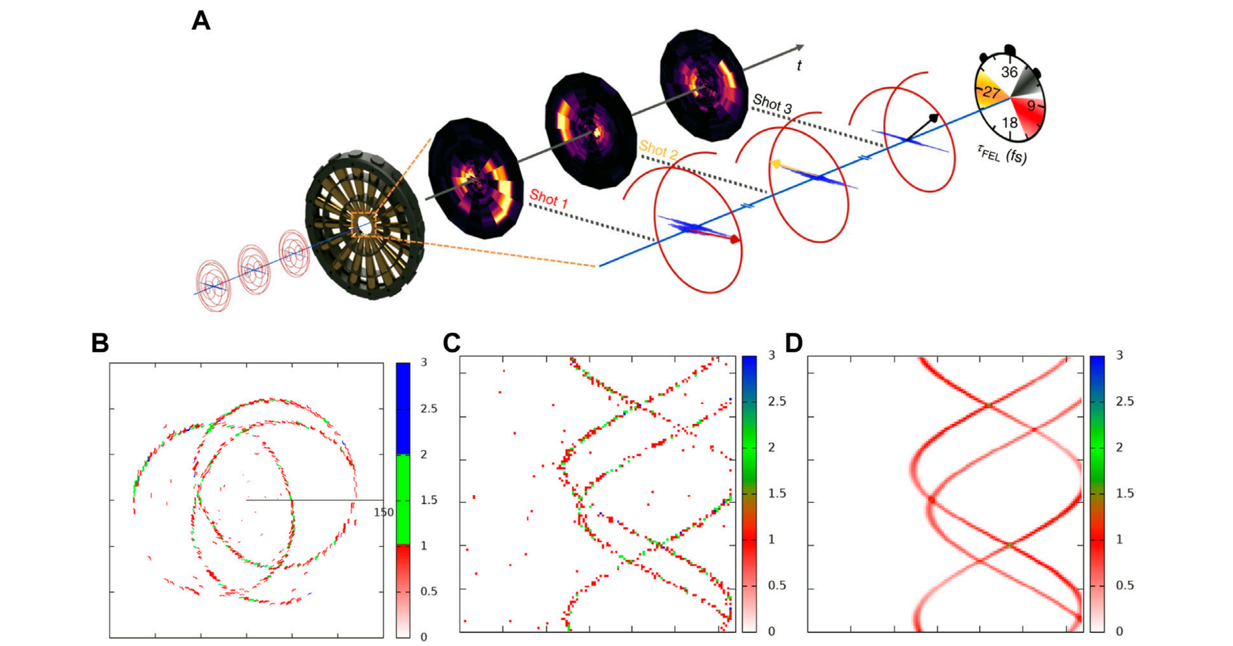


FIGURE 1
(A) Schematic of attosecond angular streaking reproduced from Ref. 9, one of the many use cases for angular array spectrometers for LCLS-II single-shot diagnostics. Sample image of angular streaking of SASE pulses in polar **(B)** and Cartesian **(C)** representations. Panel **(D)** shows the emission probability that is our desired output of the inference model.

conditions of heat and density to elucidate the inner working of planets [2] and are helping expose the workings of fusion energy [3]. The breadth of scientific discovery opened by such a facility impacts fields from novel material designs to biological function, even helping drug design for SARS-CoV-2 [4].

There is an upgrade imminent for this xFEL, the so-called LCLS-II [5], which will further accelerate the rapidly broadening range of scientific-use cases. The overwhelming data-ingest rates [6] will also be coupled with automated schemes for experimental execution. Such automation will accommodate intermittent updates that track the variations that are inherent to dynamic experimental conditions. We use the example of ultra-high throughput continuous data acquisition, analysis, and decision streams to motivate inference acceleration directly at the data acquisition node *via* direct attached co-processors. We demonstrate the high throughput inference and training for one of the early-streaming high data velocity detectors at the LCLS-II such as the array of electron spectrometers described in Ref. 7.

The xFEL pulses arise from the self-amplification of spontaneous emission (SASE) [8], and so the time and spectral structure of SASE pulses are typically quite complicated, comprising many so-called SASE sub-spikes in the phase space of time–energy joint distributions. Characterizing complicated x-ray pulses, as are commonly produced by xFELs, drives our pursuit of stream-processing angular arrays of electron time-of-flight (eToF) spectrometers. The method of attosecond angular streaking was first applied to the xFEL in Ref. 9. It is based on the measuring angle-resolved photoelectron spectra for noble gas atoms that are so-called "dressed" by a circularly polarized long-wavelength optical field. The circular polarization of the dressing laser field gives a directional push to the x-ray photo-ionized electrons at the instance of release from the atom. This directional push sweeps out the full 2π revolution in one optical cycle period, 33 1/3 fs, in the case of 10 μm wavelength. We discuss the process in depth in Refs. 9 and 10 and outline our more simplified simulation.

We are inspired by recent successes in computational ghost-imaging that highlight the value in using complex varied structures as illumination sources [11, 12]. The method achieves better results than the conventional spatial resolution by treating the measured results statistically for signal co-variance with shot-to-shot illumination variations, thus leveraging the natural fluctuations of xFEL time–energy and even polarization [13] distributions to preferentially enhance the sensitivity to nonlinear x-ray interactions [14, 15]; inference-based pulse reconstructions approaching 1 million frames per second would unlock the natural advantages of our new scientific data fire hoses, the LCLS-II and the APS-U.

Given the dynamic nature of experiments at both xFEL and synchrotron sources, we foresee a need for model adaptation and ultra-high throughput inference. Model adaptation or full re-

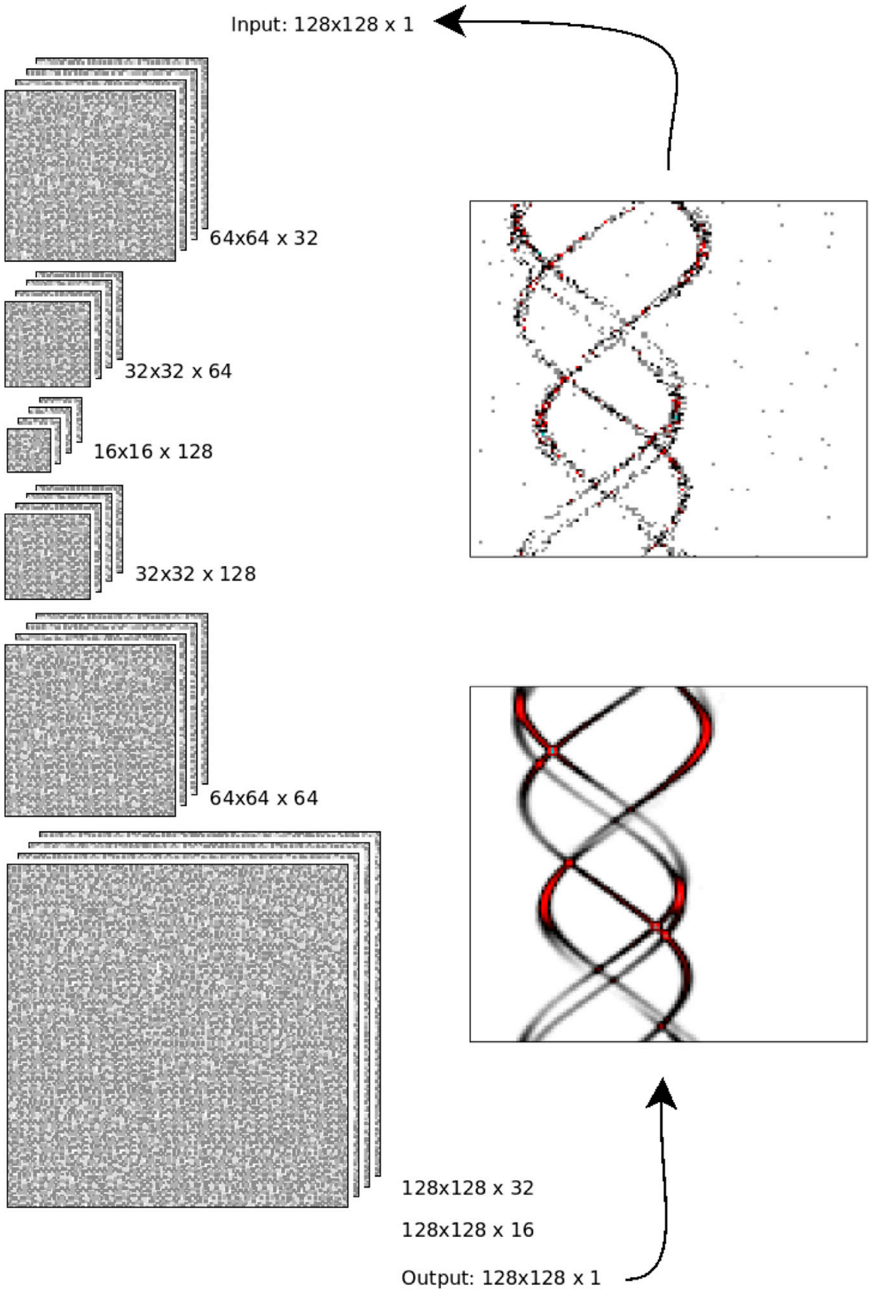


FIGURE 2
Schematic of the encoder–decoder architecture.[18].

training can be expected to occur with the cycle of so-called "runs" at these facilities. Owing to their exquisite stability, synchrotrons can be expected to pass hour-long experimental runs where environmental conditions change negligibly and the x-ray source parameters change immeasurably other than the potentially periodic "re-filling" of the ring which is easily accommodated with an intensity normalization. The xFEL is a beast of a very different color. Each shot, coming at up to 1 million per second, grows from a stochastic process and in a dynamic environment of variability in steering magnet currents, undulator settings, injector laser mode variation, and thermal motion of the experimental halls. Furthermore, variations in the experimental plan for the short 5 × 12 h campaigns typically redirect the scientific detectors parameters

TABLE 1 Network structure of CookieNetAE.[18]. All convolution layers use ReLU activation.

Layer	lYPe	Kernel	Channels	Image shape
1	Input	—	1	128 × 128
2	Convolution	3 × 3	16	128 × 128
3	Max Pooling	2 × 2	16	64 × 64
4	Convolution	3 × 3	32	64 × 64
5	Max Pool	2 × 2	32	32 × 32
6	Convolution	3 × 3	64	32 × 32
7	Max Pooling	2 × 2	64	16 × 16
8	Convolution	3 × 3	128	16 × 16
9	Convolution Transpose	2 × 2	128	32 × 32
10	Convolution Transpose	2 × 2	64	64 × 64
11	Convolution Transpose	2 × 2	32	128 × 128
12	Convolution Transpose	2 × 2	16	128 × 128
13	Convolution	1 × 1 × 16	1	128 × 128

at the 15–20 min scale. For this reason, we target very rapid re-training of our example model in the 10-min scale rather than the hours scale.

On the inference side, at the xFEL, we aim to treat each of the unique x-ray pulses individually. Since the imminent pulse rate of the LCLS-II will quickly ramp up to 100,000 shots per second over the course of its first year or so, we pursue inference acceleration on a scale that can keep abreast of such an inference rate. It should be noted that the facility is run continuously, not in burst mode, and so small batch sizes for inference, more typically batch size = 1, are expected to be the norm during operation.

In this manuscript, we focus on low batch-size streaming inference with processing rates commensurate with the expected LCLS-II repetition rates. We also discuss data-parallel training on a Graphcore POD16 and an NVidia DGX node, pointing to some of the considerations that impact computing hardware considerations when tackling high-rate training and inference. We conclude with a discussion of high-speed accelerated diagnostics for low-latency real-time adaptive control for a running xFEL experiment.

2 Materials and methods

The physics behind the x-ray pulse reconstruction is based on an external electric field with circular polarization as described in Ref. 9. Reproduced in Figure 1A, photo-electrons are emitted from a target gas with an energy that gets boosted toward the instantaneous direction of the optical laser vector potential $\vec{A}(t)$, with itself rotating in the detector reference frame with an angular frequency $\omega = 2\pi c/\lambda$. We call this laser field the “dressing” field. We typically choose an infrared wavelength of 10 μm such that the time is encoded into an angle with a calibration of about 0.19 radians/femtoseconds, e.g., the full period of our clock hand turns one revolution in 33 fs which is about three times the duration of a typically desired SASE pulse. The resulting “images” (see Figures 1B–D) of these boosted electrons encode SASE spikes as offset circles in polar coordinates (B) or sinusoidal features in the un-wrapped Cartesian representation as they are more familiar as sinograms (C) in medical radiography imaging. We simulate images in this manuscript based on the smooth emission distribution (D) in order to have full knowledge of the SASE sub-spikes to be recovered. We use this to demonstrate the principle of high throughput inference and short training cycles in preparation for experimental data when it becomes available in the very near future of LCLS-II.

We simulate results for an 8-fold over-sampled angular dimension as compared to the planned instrument described in Ref. 7 to help draw a broader generalization of our approach across time-series analyses such as in magnetic fusion applications [16]. The external “dressing” field pushes photo-ionized electrons as described previously. Our focus on simulated sinograms [17] of representative few-spike SASE pulses provides a known ground truth for training the inference model. We perform a random sampling of the electron emission probability distribution, Y (ν, θ), where ν is the photon energy and θ is the emission angle. The sampling is not strictly micro-canonical Monte-Carlo in that we do not require a statistical agreement with the process of SASE growth from the vacuum field fluctuation as in proper SASE pulse simulations. Rather, we prefer an even statistical distribution, not Bolzmann distribution, of the modes in the field such that the training examples for our machine-learned algorithm are not biased to see only that which is statistically more likely. Electron counts are sampled from this distribution to yield the simulated measurement X (ν, θ). Thus, we are attempting to train the

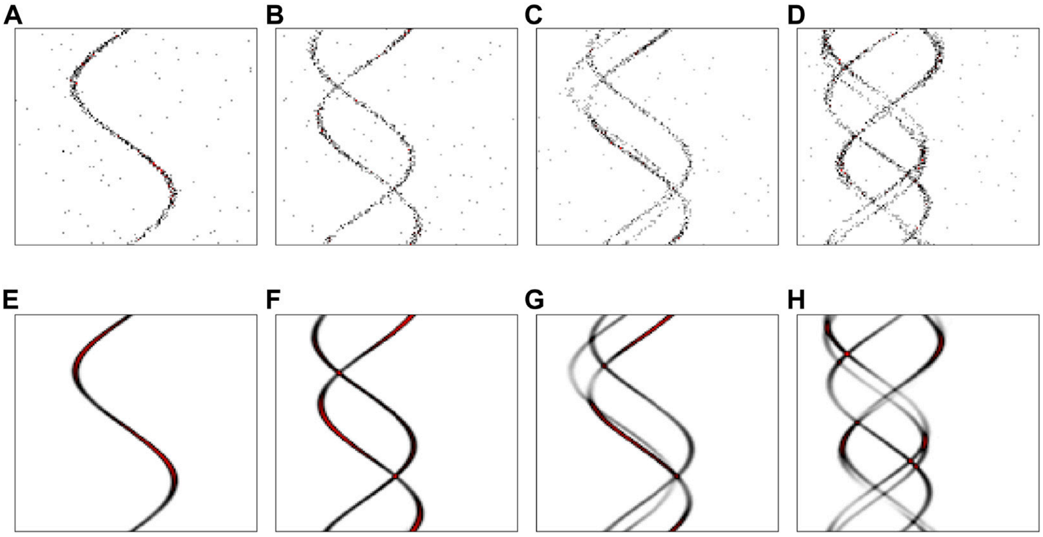


FIGURE 3
(A–D) Input images and **(E–H)** inference output images as described in Section 1. Examples of 1-4 SASE sub-spikes are shown, respectively, although the training set includes higher numbers of sub-spikes.

model, CookieNetAE [18], to input X (ν, θ) and reproduce the original Y (ν, θ) free of the grainy Poisson statistics of the sampling. Our encoder–decoder network—CookieNetAE [18]—demonstrates our ability to transform the high data rate into a latent representation at the rate of the streaming data pipeline. The loss is computed as the mean-squared error (MSE) between the input image and the original smooth Y (ν, θ) probability distribution. Physically relevant parameters are thus fully constrained by this Y, its recovery from X indicates that the latent representation, at the interface between the encoder and decoder sides, contains sufficient information for any downstream task, be it regression or classification. In effect, one could freeze the encoder-side weights and use the latent representation as a feature vector input for any conceivable downstream inference task. For the purposes of this manuscript, we presuppose that sensor-specific calibrations (time-to-energy) will be implemented in the signal acquisition electronics since the time-to-energy calibration is free to be adjusted independently for each angle of detection in the detector system [7]. To validate the live calibrations, a fraction of the events will be routed as raw data that bypass the upstream pre-processing chain. Since this expected 0.1%, or 1 kHz, rate of raw data could feed adaptive re-training, we, therefore, also use our model to benchmark the acceleration of *in situ* training.

2.1 Simulation

Specifically, we use forward simulations to build from a ground truth Y to the example X, from which we train the inference model to generate a predicted Y' that minimizes the MSE (Y, Y'). The probability density function at a given detection angle is a sum of Gaussian distributions, each associated with a single SASE sub-spike j. This emission energy is modified by the so-called dressing laser field according to a sinusoidal variation discussed previously. The angular registration, e.g., the phase ϕ_j, is determined by the relative delay between jth sub-spike and the carrier field of the dressing optical laser. The mechanism behind this attosecond resolution in x-ray photoelectron angular streaking is detailed in Ref. 9 and 10; and for our purposes, we crudely, but sufficiently, simplify by writing the probability density function for electron emission as:

$$P(E,\theta) = \sum_{j=1}^{n} a_j \mathcal{N}\left(\nu_j, \sigma_j, \theta, \phi_j\right), \qquad (1)$$

$$\mathcal{N}\left(\nu_j, \sigma_j, \phi_j, \theta\right) = \nu_j + A_j \cos\left(\theta + \phi_j\right) + C, \qquad (2)$$

where n is the number of SASE sub-spikes in a given shot, A_j is the maximum streaking amplitude, E is the energy of detection (horizontal in Figure 1C), and θ is the angle of detection (vertical in Figure 1C). The number of sub-spikes is chosen *via* a Poisson distribution with a peak at four sub-spikes but many shots include higher numbers of SASE spikes. We build the ground truth probability density function and use it as the output Y' for training the inference model. As input, we randomly sample that distribution P (E, θ) as shown in Figure 1D, and use the results to fill in a 2-dimensional histogram X image as shown in Figures 1B, C.

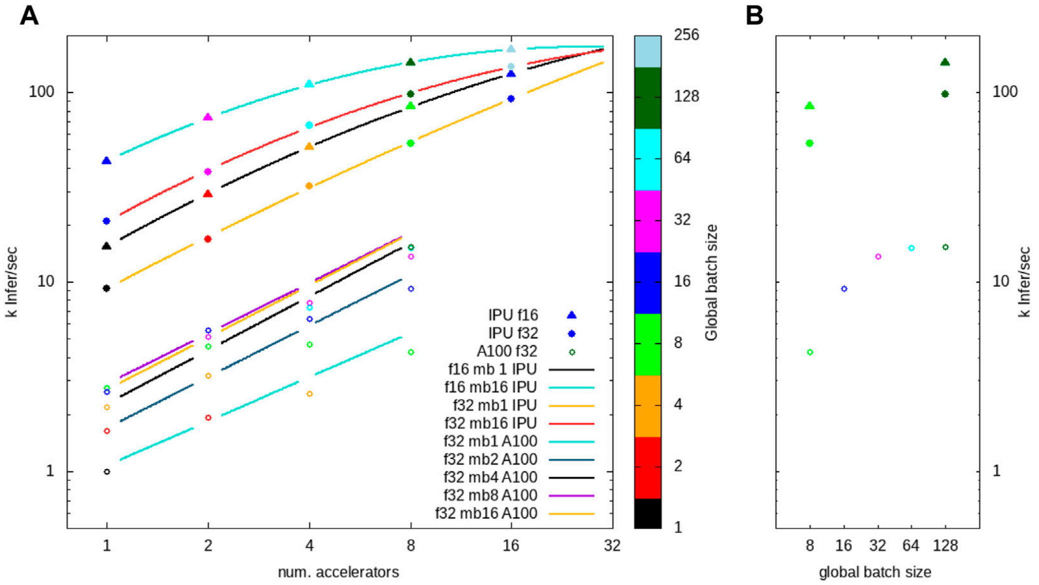


FIGURE 4
(A). Inference results *versus* number of accelerators used in the data-parallel operation using 1–16 IPUs and 1–8 A100 GPUs. (B). Results *versus* global batch size for the case of eight accelerators, and the symbol definitions are identical with panel (A). The fitting results are given in Table 2.

TABLE 2 Coefficients for the following general formula described in Eqs. 3 and 7. The curves follow along the constant mini batch (mb) such that the global batch size = mb × η. Note that only four or less evaluation points for the A100 case require we fit only coefficients *a* and *b*.

Case	Device	Precision	Mini batch	a	b	c
infer [infer/sec]	IPU	f16	1	13.9	0.988	−0.0575
		f16	16	15.4	0.845	−0.0889
		f32	1	13.2	0.933	−0.0254
		f32	16	14.3	0.974	−0.0739
infer [infer/sec]	A100	f32	1	10.1	0.775	–
			2	10.7	0.896	–
			4	11.1	0.941	–
			8	11.5	0.884	–
			16	11.4	0.913	–
train [sec/epoch]	IPU	f32	16	5.40	−0.969	0.0734
			16	6.53	−0.984	0.0433
train [sec/epoch]	A100	f32	16	8.74	−0.917	–
			32	7.94	−0.914	–
			64	7.57	−0.916	–
			128	7.44	−0.956	–

2.2 Model and data

For the inference task, we have built a convolutional neural network (CNN) based encoder–decoder model for predicting the smooth probability density function Y'. We have chosen to work with a representative CNN model architecture since the topic at hand is not optimization of the model design, but rather the epochs per second in training and inferences per second in

TABLE 3 Inferences per second for various conditions measured.

Device	η	Mini batch	Global batch	γ [kInfer/sec]	γ [kInfer/sec]
	Devices	Size	Size	(f16)	(f32)
IPU	1	1	1	15.4	9.25
	2		2	28.8	16.8
	4		4	51.6	32.1
	8		8	84.0	53.8
	16		16	125	92.3
IPU	1	16	16	43.1	20.8
	2		32	73.2	37.9
	4		64	110	66.6
	8		128	143	97.8
	16		256	169	136
A100	1	1	1	–	0.996
	2		2	–	1.92
	4		4	–	3.51
	8		8	–	4.87
A100	1	16	16	–	2.64
	2		32	–	5.13
	4		64	–	9.75
	8		128	–	17.6

application. Our encoder–decoder model has 343,937 trainable parameters. We use the rectified linear unit (ReLU) activation in all layers; mean-squared error (MSE) as a loss function; and the Adam optimizer with $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$. We use a convolution 2D-transpose method with a stride of (2,2) for up-sampling in the decode layers. The model takes a (128, 128, 1) small integer image as input and gives (128, 128, 1) float image of the originating 2D probability function as output. We note the asymmetry in our encoder–decoder as can be seen in Figure 2. This asymmetry is partly used for future developments of an image segmentation model that is beyond the scope of the present manuscript. The details of the model architecture are given in Table 1.

The model exercised here only serves to provide a sample architecture. Since the aim of this work is to demonstrate the performance of the hardware across a range of potential hyper-parameters, we have not explicitly performed so-called "hyper-parameter tuning" in order to find rapid convergence; the training duration and inference throughput as functions of batch size and number of accelerators are our aim, not the accuracy of the model. We do show an example of the performance for the hold-out set in Figure 3, simply to show that the model is not pathological. We run a full 50 epochs of training regardless of the fact that convergence is achieved as early as epoch 10 for some of the match-size configurations.

The dataset was generated using the open source simulation tool [17]. One million samples were generated for training, validation, and inference using a 90/5/5 split. The training dataset had 900,000 pairs of 128 × 128 images, a grainy (Poisson starved) input image, and a smooth target image. The inference dataset consisted of 50,000 unique 128 × 128 images and was repeated 10 times in the case of the Graphcore Intelligence Processing Units (IPUs) to provide a sufficient workload. Training and inference results for the IPU case were collected on a direct attached Graphcore POD16—four interconnected M2000s, each with 4 IPUs.

For the A100 graphics processing units (GPUs), only single precision, 32-bit floating points (FP32) were tested, while on the IPU, Accumulating Matrix Product (AMP) and Slim Convolution (SLIC) instructions were used for high-performance multiply–accumulate sequences for both 32-bit (FP32) and 16-bit (FP16) precision formats [19]. The two different mixed-precision arithmetic schemes for the IPU case are as follows:

- FP32.32: AMP operation with FP32 input multiplicands as well as FP32 partial sums of products

TABLE 4 IPU train time considered in seconds per epoch based on the average results for 50 epochs of 900 k training images per epoch.

Device	Precision	η	Mini batch	Global batch	Sec/epoch	Min/50 epochs
IPU	f16	1	16	16	42.13	35.10
		2		32	22.56	18.80
		4		64	13.51	11.25
		8		128	8.86	7.38
		16		256	6.47	5.39
IPU	f32	1	16	16	92.56	77.13
		2		32	48.01	40.00
		4		64	26.93	22.44
		8		128	15.52	12.93
		16		256	9.80	8.16
A100	f32	1	64	64	186.5	155.41
			128	128	173.3	144.41
			256	256	173.0	144.16
			512	512	165.7	138.08
			1,024	1,024	160.2	133.50
A100	f32	2	32	64	131.0	109.16
			64	128	103.5	86.25
			128	256	91.3	76.08
			256	512	84.5	70.41
			512	1,024	81.1	67.58
A100	f32	4	16	64	119.9	99.91
			32	128	68.4	57.00
			64	256	53.5	44.58
			128	512	45.5	37.91
			256	1,024	43.0	35.83
A100	f32	8	8	64	117.2	97.66
			16	128	63.5	52.91
			32	256	36.9	30.75
			64	512	28.0	23.33
			128	1,024	24.0	20.00

- FP16.16: AMP operation with FP16 input multiplicands and FP16 partial sums of products.

In the case of the IPU, the command line utility PopRun was used to create multiple instances and launch them as distributed data-parallel applications on the Graphcore POD16 compute system, either on a single server or multiple servers within the same POD [20].

In our setup, a single host server with SDK 2.5 was used with the number of instances being set equal to the number of model replicas for the inference case. For training, the number of instances was set to half the number of model replicas. Horovod was used to distribute the inference and training across 1-8 A100 GPUs in a single Nvidia-DGX node in Argonne's ThetaGPU cluster. Samples of the input images and output predictions are shown in Figure 3.

3 Results

3.1 Inference

Our primary focus in this manuscript is the demonstration of inference throughput that approaches compatibility with the

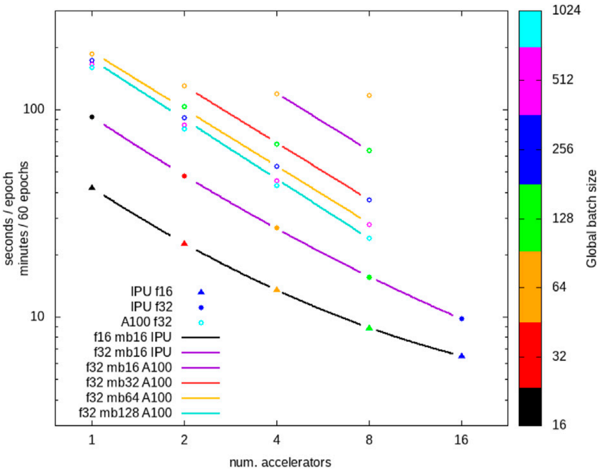


FIGURE 5
Training time per epoch (900 k images in set) *versus* number of accelerators for various global batch sizes. We indicate FP16/FP32 for the IPU case and note that all A100 results are based on the FP32 tests. The trend curves follow the coefficients given in Table 2.

eventual 1 million shots per second rate of the LCLS-II xFEL. We used CookieNetAE [18] trained on the simulation as described previously to process a stream of images across a range of available instances of GPU and IPU inference accelerators. Our results are presented graphically in Figure 4A as the inference rate y *versus* number of accelerators η used in the data-parallel mode. Figure 4B gives an example of the inference throughput *versus* the global batch size for the case of one DGX node of eight GPUs and two interconnected POD4 nodes with four IPUs each. We note that the GPU case is not leveraging TensorRT for inference so as to maintain negligible code changes for compiling models to IPU and GPU.

Table 2 presents fitted coefficients based on the pseudo-inverse method (Eqs 3–5) for Taylor expansion fitting the log-of-rate $y = \log_2 \gamma$ *versus* the log-of-number $x = \log_2 \eta$. In this logarithmic representation, 2^a is the single accelerator, $\eta = 1$ is the rate, and the coefficient b represents the scaling power law with an increasing number of accelerators, e.g., the "slope" in the Taylor expansion of the data around $\eta = 1, x = 0$ in log-space; Θ is the vector of polynomial expansion coefficients (Eq. 7).

$$\log_2 \gamma = y = \left(a + bx + cx^2\right), \tag{3}$$

$$y = \Theta \cdot X, \tag{4}$$

$$y \cdot X^{-1} = \Theta \cdot \left[XX^{-1}\right] = \Theta, \tag{5}$$

where

$$y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}, \; X = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_n \\ x_0^2 & x_1^2 & \cdots & x_n^2 \end{bmatrix}, \tag{6}$$

$$\Theta = \left[\, a \;\; b \;\; c \,\right]. \tag{7}$$

We can see from Figure 4A that the fits deviate from linear as more than two or four IPUs are used. This deviation is seen both from coefficient $b \lessapprox 1$ and from the quadratic term $c \lessapprox 0$. We attribute this to bandwidth limitation in the IPU interconnect fabric as indicated by the extrapolated trend—to be always taken with a grain of salt—that would extend the lines toward a common saturation at about 200 k inferences per second. Of course, this particular saturation limit is highly dependent on the incoming raw data geometry, even for very similar models. Since we are targeting a rate 5 times higher than that achieved here, we are actively pursuing a reduced representation that alleviates this data ingestion limitation while still preserving the image integrity and information. Our GPU results were all run with a single DGX machine using the eight A100 GPUs in the same data-parallel inference mode. Inference results for all tested configurations are presented in Table 3.

3.2 Training

Although not the principle goal for this study, the ability to train models directly at the source for both IPUs and GPUs motivated our investigation of the time to train these devices in the data-parallel mode as well. From Table 4, we find a training time for 50 epochs that accelerate from nearly an hour with 1 IPU at batch size 16 (FP16) to only 6 min with 16 IPUs. We see in Figure 5 that the general trend of inverse scaling ($1/\eta$) dominates until about eight accelerators for a constant mini-batch size, e.g., the global batch size scales linearly with the number of accelerators. One can see from the result for constant global batch size (symbol color) for A100 GPUs that splitting global batches across increasing accelerators quickly suffers diminishing returns for global batch sizes below about 128 (green, open circles). This coincides with the rule of thumb suggesting 64 or 128 local batch sizes given the DGX configuration of eight accelerators. With up to 16 IPUs, hosted as four interconnected M2000 nodes, each with four IPUs, we find a very nearly ideal expected inverse scaling $1/\eta$ as we hold constant the mini batch size of 16 and use PopRun to spread the workload across multiple IPUs and multiple M2000 nodes.

Table 2 shows our log-space Taylor coefficients for training. The coefficient $b \gtrapprox -1$ is quite close to an ideal inverse power law, and there is only a very small quadratic term $c \gtrapprox 0$. Even given the communications bandwidth limitation, for 16 IPUs at FP16 precision, we find from Table 4 that we can achieve full-model training with 50 epochs of 900 k images in under 6 min.

4 Discussion

The potential impact of scientific machine learning [21] for incorporating data analysis as a streaming processing pipeline from source to data center [22] cannot be overstated for next generation accelerator-based user facilities such as the LCLS-II [23] and the APS-U [24]. The growing adoption of transformer models [25-26] even outside the domain of natural language processing [27-29] is sure to extend to the scientific data interpretation domain. We, therefore, expect that embedding models will work their way upstream, eventually into the sensor electronics themselves. For this reason, we have chosen an encoder–decoder network, in particular, one that has no skip connections, as our example inference workflow for the upcoming single-shot 1 million frames per second LCLS-II attosecond streaking x-ray diagnostic.

We demonstrate the ability to produce a latent feature vector that fully captures the information contained in the simulated experimental results (Figures 3A–D) while effectively removing any impact of noise. We do so by recovering the original smooth probability distribution (Figures 3E, F) used to create the simulated experimental results. Our encoder–decoder model, CookieNetAE [18], is therefore a stand-in for the upstream encoding and embedding layers for transformer architectures.

Larger models pose challenges for streaming data processing, particularly so for real-time control decisions. Although CookieNetAE was used as a surrogate for transformer models, the high fidelity in image reconstruction of Figure 3 demonstrates that the number of composite sinusoids is fully encoded in the latent-space feature vector. As such, one of the potential use cases of the encoding layers of CookieNetAE could be the rapid prediction of a particular shot's SASE complexity from, e.g., Figure 3: (A) triggers simple-binned single-spike reference accumulator, (B) triggers 2D histogram accumulation based on double-pulse relative delay and energy separation, and (C) and (D) trigger the full-feature vector to be stored along with downstream detector results for offline statistical treatments. Since the LCLS-II will quickly ramp the shot rate from few tens of kF/sec to a million frames/sec, these data-routing decisions must keep abreast of the rate; they must inform and direct the path of the streaming analysis for each shot as it is acquired [30]. Our A100 inference results are consistent with the early expectation of 10 kF/s and the Graphcore POD16 can carry us sufficiently beyond the 100 kF/s rate needed for the rapid increase in the repetition rate expected. We must however continue to develop leaner models, bandwidth efficient data ingestion, and faster inference environments to enable the full million frames per second rate expected in the coming few years.

In pursuing accelerated inference at the sensor edge, we also demonstrated that models can be re-trained *in situ* with the very same hardware for both GPU and IPU. Although not a requirement for our particular case, it does however raise a significant opportunity given the fact that 0.1% raw data could be leveraged locally for model re-training. Combined with software-defined memory provisioning [31], incoming anomaly events could be held locally in system memory for inclusion in updated training sets and used in rapid re-training of the embedding model. This small fraction of raw data nevertheless accounts for up to 1 GB/s of the continuous data stream. The prospect of dynamically provisioned TB-scale local memory directly at the acquisition node that supports accelerated local training as the experimental conditions vary throughout an experimental shift would truly enable a continuously adaptive autonomous experimental ecosystem.

Data availability statement

The datasets generated and used for this study can be accessed with the code in CookieSimSlim v1.1.0 https://github.com/ryancoffee/CookieSimSlim/archive/refs/tags/v1.1.0.tar.gz. The analysis results are compiled in the Tables.

Author contributions

MK performed and collected all IPU results and ZL performed and collected all GPU results. NL and RC designed the initial CookieBoxAE model architecture, while MK adapted it for use in IPU and ZL adapted for ThetaGPU. RC built the data simulation code and performed the analysis and presentation of all results. MK and RC contributed equally to the composition of the manuscript with additional contributions from ZL.

Funding

This work was predominantly funded by the Department of Energy, Office of Science, Office of Basic Energy Sciences under Field Work Proposal 100643 "Actionable Information from Sensor to Data Center." The development of the algorithm for simulation was funded by the Department of Energy, Office of Science, Office of Basic Energy Sciences under Field Work Proposal 100498 "Enabling long wavelength Streaking for Attosecond X-ray Science."

Acknowledgments

RC acknowledges support from the Department of Energy, Office of Science, Office of Basic Energy Sciences, for funding the development of the detector array itself under Grant Number FWP 100498 "Enabling long wavelength Streaking for Attosecond X-ray Science." He also acknowledges the synergistic support for computational method development by the Office of Fusion Energy Science under Field Work Proposal 100636 "Machine Learning for Real-time Fusion Plasma

Behavior Prediction and Manipulation." This research used resources of the Argonne Leadership Computing Facility, a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357, for collecting GPU-related benchmarks.

Conflict of interest

MK was employed by Graphcore, Inc.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. Wernet P, Kunnus K, Josefsson I, Rajkovic I, Quevedo W, Beye M, et al. Orbital-specific mapping of the ligand exchange dynamics of fe(co)5 in solution. *Nature* (2015) 520:78–81. doi:10.1038/nature14296

2. Kraus D, Vorberger J, Pak A, Hartley NJ, Fletcher LB, Frydrych S, et al. Formation of diamonds in laser-compressed hydrocarbons at planetary interior conditions. *Nat Astron* (2017) 1:606–11. doi:10.1038/s41550-017-0219-9

3. Bekx JJ, Lindsey ML, Glenzer SH, Schlesinger K-G. Applicability of semiclassical methods for modeling laser-enhanced fusion rates in a realistic setting. *Phys Rev C* (2022) 105:054001. doi:10.1103/PhysRevC.105.054001

4. Durdagi S, Dag C, Dogan B, Yigin M, Avsar T, Buyukdag C, et al. *Near-physiological-temperature serial femtosecond x-ray crystallography reveals novel conformations of sars-cov-2 main protease active site for improved drug repurposing.* bioRxiv (2020).

5. LCLS. Lcls-ii: A world-class discovery machine lcls (2022). Available at: slac.stanford.edu/lcls-ii.

6. Thayer JB, Carini G, Kroeger W, O'Grady C, Perazzo A, Shankar M, et al. *Building a data system for lcls-ii*. New York, NY: Institute of Electrical and Electronics Engineers Inc. (2018). doi:10.1109/NSSMIC.2017.8533033

7. Walter P, Kamalov A, Gatton A, Driver T, Bhogadi D, Castagna J-C, et al. Multi-resolution electron spectrometer array for future free-electron laser experiments. *J Synchrotron Radiat* (2021) 28:1364–76. doi:10.1107/S1600577521007700

8. Pellegrini C, Marinelli A, Reiche S. The physics of x-ray free-electron lasers. *Rev Mod Phys* (2016) 88:015006. doi:10.1103/RevModPhys.88.015006

9. Hartmann N, Hartmann G, Heider R, Wagner MS, Ilchen M, Buck J, et al. Attosecond time–energy structure of x-ray free-electron laser pulses. *Nat Photon* (2018) 12:215–20. doi:10.1038/s41566-018-0107-6

10. Li S, Guo Z, Coffee RN, Hegazy K, Huang Z, Natan A, et al. Characterizing isolated attosecond pulses with angular streaking. *Opt Express* (2018) 26:4531–47. doi:10.1364/OE.26.004531

11. Shapiro JH. Computational ghost imaging. *Phys Rev A (Coll Park)* (2008) 78: 061802. doi:10.1103/PhysRevA.78.061802

12. Padgett MJ, Boyd RW. An introduction to ghost imaging: Quantum and classical. *Phil Trans R Soc A* (2017) 375:20160233. doi:10.1098/rsta.2016.0233

13. Sudar N, Coffee R, Hemsing E. Coherent x rays with tunable time-dependent polarization. *Phys Rev Accel Beams* (2020) 23:120701. doi:10.1103/PhysRevAccelBeams.23.120701

14. Giri SK, Alonso L, Saalmann U, Rost JM. Perspectives for analyzing non-linear photo-ionization spectra with deep neural networks trained with synthetic Hamilton matrices. *Faraday Discuss* (2021) 228:502–18. doi:10.1039/D0FD00117A

15. Kumar Giri S, Saalmann U, Rost JM. Purifying electron spectra from noisy pulses with machine learning using synthetic Hamilton matrices. *Phys Rev Lett* (2020) 124:113201. doi:10.1103/PhysRevLett.124.113201

16. Jalalvand A, Kaptanoglu AA, Garcia AV, Nelson AO, Abbate J, Austin ME, et al. Alfvén eigenmode classification based on ECE diagnostics at DIII-d using deep recurrent neural networks. *Nucl Fusion* (2021) 62:026007. doi:10.1088/1741-4326/ac3be7

17. Coffee RN. *Cookiesimslim: Slim simulator for lcls-slac cookiebox detector* (2022). Available at: https://github.com/ryancoffee/CookieSimSlim.

18. Layad N, Liu Z, Coffee R. Open source implementation of the cookienetae model (2022). Available at: https://github.com/AISDC/CookieNetAE.

19. Graphcore. Mixed-precision arithmetic for ai: A hardware perspective docs (2022). Available at: graphcore.ai/projects/ai-float-white-paper/en/latest/index.html.

20. Graphcore. *Popdist and poprun: User guide docs* (2022). Available at: graphcore.ai/projects/poprun-user-guide/en/latest/index.html.

21. Sanchez-Gonzalez A, Micaelli P, Olivier C, Barillot TR, Ilchen M, Lutman AA, et al. Accurate prediction of x-ray pulse properties from a free-electron laser using machine learning. *Nat Commun* (2017) 8:15461. doi:10.1038/ncomms15461

22. Liu Z, Ali A, Kenesei P, Miceli A, Sharma H, Schwarz N, et al. *Bridging data center ai systems with edge computing for actionable information retrieval* (2022).

23. Schoenlein R. *New science opportunities enabled by lcls-ii x-ray lasers*. SLAC Report SLAC-R-1053 (2015). p. 1–189.

24. Hansard B. *Advanced photon source upgrade will transform the world of scientific research* (2020). Available at: www.anl.gov/article/advanced-photon-source-upgrade-will-transform-the-world-of-scientific-research.

25. Devlin J, Chang M, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. (2018). CoRR abs/1810.04805.

26. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. *Language models are unsupervised multitask learners* (2018).

27. Payne C. *Musenet* (2019). Available at: openai.com/blog/musenet.

28. Naseer MM, Ranasinghe K, Khan SH, Hayat M, Shahbaz Khan F, Yang M-H. Intriguing properties of vision transformers. In: *Advances in neural information processing systems*. Ranzato M, Beygelzimer A, Dauphin Y, Liang P, Vaughan JW, editors, 34. Curran Associates, Inc. (2021). p. 23296–308.

29. Khan S, Naseer M, Hayat M, Zamir SW, Khan FS, Shah M. *Transformers in vision: A survey* (2022). Available at: https://arxiv.org/pdf/2101.01169.pdf.

30. Corbeil Therrien A, Herbst R, Quijano O, Gatton A, Coffee R. Machine learning at the edge for ultra high rate detectors. (2019). 1–4. doi:10.1109/NSS/MIC42101.2019.9059671

31. Kove I. Kove machine learning white paper (2022). Available at: kove.net/downloads?file=machine-learning.pdf.

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 05 January 2023
DOI 10.3389/fphy.2022.960963

Check for updates

Data-driven modeling of beam loss in the LHC

Ekaterina Krymova[1]*, Guillaume Obozinski[1], Michael Schenk[2], Loic Coyle[2,3] and Tatiana Pieloni[2]

[1]Swiss Data Science Center, EPFL and ETH Zürich, Zürich, Switzerland, [2]Particle Accelerator Physics Laboratory, Institute of Physics, EPFL, Lausanne, Switzerland, [3]CERN, Geneva, Switzerland

OPEN ACCESS

EDITED BY
Alexander Scheinker,
Los Alamos National Laboratory (DOE),
United States

REVIEWED BY
Elena Fol,
European Organization for Nuclear Research (CERN), Switzerland
Egle Tomasi,
CEA Saclay, France
Pietro Chimenti,
State University of Londrina, Brazil

*CORRESPONDENCE
Ekaterina Krymova,
✉ ekaterina.krymova@sdsc.ethz.ch

SPECIALTY SECTION
This article was submitted to High-Energy and Astroparticle Physics,
a section of the journal
Frontiers in Physics

RECEIVED 03 June 2022
ACCEPTED 06 December 2022
PUBLISHED 05 January 2023

CITATION
Krymova E, Obozinski G, Schenk M, Coyle L and Pieloni T (2023), Data-driven modeling of beam loss in the LHC.
Front. Phys. 10:960963.
doi: 10.3389/fphy.2022.960963

In the Large Hadron Collider, the beam losses are continuously measured for machine protection. By design, most of the particle losses occur in the collimation system, where the particles with high oscilla-tion amplitudes or large momentum error are scraped from the beams. The particle loss level is typically optimized manually by changing control parameters, among which are currents in the focusing and defocusing magnets. It is generally challenging to model and predict losses based only on the control parameters, due to the presence of various (non-linear) effects in the system, such as electron clouds, resonance effects, etc., and multiple sources of uncertainty. At the same time understanding the influence of control parameters on the losses is extremely important in order to improve the operation and performance, and future design of accelerators. Prior work [1] showed that modeling the losses as an instantaneous function of the control parameters does not generalize well to data from a different year, which is an indication that the leveraged statistical associations are not capturing the actual mechanisms which should be invariant from 1 year to the next. Given that this is most likely due to lagged effects, we propose to model the losses as a function of not only instantaneous but also previously observed control parameters as well as previous loss values. Using a standard reparameterization, we reformulate the model as a Kalman Filter (KF) which allows for a flexible and efficient estimation procedure. We consider two main variants: one with a scalar loss output, and a second one with a 4D output with loss, horizontal and vertical emittances, and aggregated heatload as components. The two models once learned can be run for a number of steps in the future, and the second model can forecast the evolution of quantities that are relevant to predicting the loss itself. Our results show that the proposed models trained on the beam loss data from 2017 are able to predict the losses on a time horizon of several minutes for the data of 2018 as well and successfully identify both local and global trends in the losses.

KEYWORDS

beam losses, accelerator control, predictive model, ARMAX, Kalman filter

1 Introduction

Excessively high beam losses in the Large Hadron Collider (LHC) [2, 3] can cause a quench in the superconducting magnets which will trigger a beam dump and a long recovery period to restore the nominal temperature. As a result, valuable time and hence integrated luminosity is lost for the physics experiments while the LHC needs to be refilled with a new beam. On the other hand, better control of losses guarantees more efficient operations, higher luminosity, and thus greater discovery potential. During the LHC run, the machine operators may change several parameters of the system, such as currents in the quadrupole, sextupole, and octupole magnets, in order to maximize the beam intensity and thus minimize the particle loss. Machine learning and statistical methods have been extensively used to analyze the data from accelerators and to improve operations [4–9]. It is possible to construct predictive models of the losses from the control parameters using standard ML techniques [1] based on LHC beam loss data within the same year, but the generalization of such approaches to the data of the next year was found to be challenging. A better understanding of the instantaneous and longer-term effects of control parameters on beam losses can help decrease the losses and improve the operations and performance output of the LHC. Furthermore, it can possibly lead to mitigation techniques for the particle losses for other existing machines as well as provide valuable input already during the design phase of potential future colliders, such as the Future Circular Collider (FCC) [10, 11]. At present, there is no accurate physics model for particle beam losses as a function of machine settings and control parameters. A statistical or machine learning model that would cover the possible scenarios of control parameters evolution could eventually be employed to find the optimal control policy. Using an optimized plan for the control parameters could significantly improve the performance of the accelerators. In what follows we make a step in this direction: the goal of this work is to improve the understanding of the effect of input parameters on losses and to propose an interpretable model which would be general and robust enough to generalize to beam loss data acquired in different years.

We assume that the two LHC beams evolve similarly under the change of control parameters disregarding possible beam coupling effects, and we concentrate on modeling the beam 1 losses. The majority of the losses occur in the collimation systems located in the beam cleaning areas at the Insertion Region (IR) 3 and IR7 of the accelerator, where the losses are recorded by beam loss monitors (BLM) [12]. The collimators at IR7 remove particles with large transverse oscillation amplitudes, whereas those at IR3 are responsible for removing the particles with a momentum error beyond a chosen threshold. Among the two collimator systems, the most active cleaning happens in IR7, therefore we concentrate on modeling of losses of beam 1 recorded by the BLM at IR7 and further refer to it as “the loss.”

Several additional important characteristics can be measured during the machine run, which cannot be directly controlled, but that contain information about hidden non-linear processes affecting the beam. Among such quantities are heatloads, that are a proxy to electron cloud effects [13], as well as horizontal and vertical emittances. The emittances describe the spread of the particles in phase space and are related to the mean physical dimension of the beam in the machine [14]. The emittance measurements are carried out in time along the beam and specific post-processing is used to estimate the average emittance of the whole beam. During the operation of the LHC, electron cloud can appear due to the acceleration of electrons in the beam pipe by the proton bunches, causing an avalanche process which leads to the heating of the beam pipe and magnets, to increased emittance and potentially to beam instabilities [15].

Tune variables are related to the frequency of betatronic oscillations in the machine. Tunes are corrected through a dedicated feedback system and mainly depend on the strength of the quadrupole magnets, although they can be also affected by the quadrupole component in the main dipoles, by the sextupole component from the main dipoles and sextupole corrector magnets [16].

As most of the non-linear effects in the beam physics are indirectly related to the change of input parameters it appears natural to rely on the information contained in the past observations. We therefore consider Vector AutoRegressive Moving Average models with eXogenous input variables (VARMAX) and compare them with models that relate the input variables directly to the losses. We train a common model on the data from multiple time series from one given year corresponding each to a different LHC fills with diverse filling schemes[1].

Available data: To construct and evaluate the predictive model of losses the observations of the losses along with other quantities measured during the LHC fills of the years 2017 and 2018 are available [17]. Observations are recorded with a frequency of 1 Hz. The fills start with the injection of the beam according to a selected filling scheme. We have selected the filling schemes which are occurring most frequently among all the encountered schemes in the data as we expected that the properties of the injected beam could vary depending on the type of injection. Altogether, for the selected schemes, the data of 105 fills are available in 2017, and 144 fills are recorded for 2018. We focus on the data for beam 1 during the “PRERAMP” beam mode of the machine fill. The “PRERAMP” beam mode occurs after the full beam has been injected and before starting the

[1] “Fill” refers to the time period starting from injection of new beam into the LHC until the beam gets dumped. The filling scheme, on the other hand, defines which of the radio-frequency buckets are filled with particles and which ones are left empty.

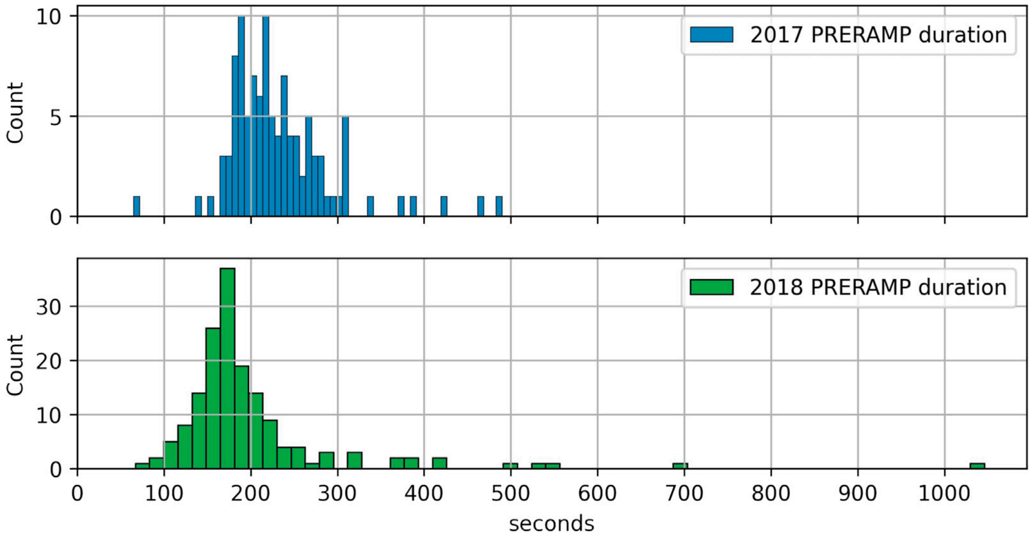


FIGURE 1
Histogram of PRERAMP time series lengths for the years 2017 and 2018.

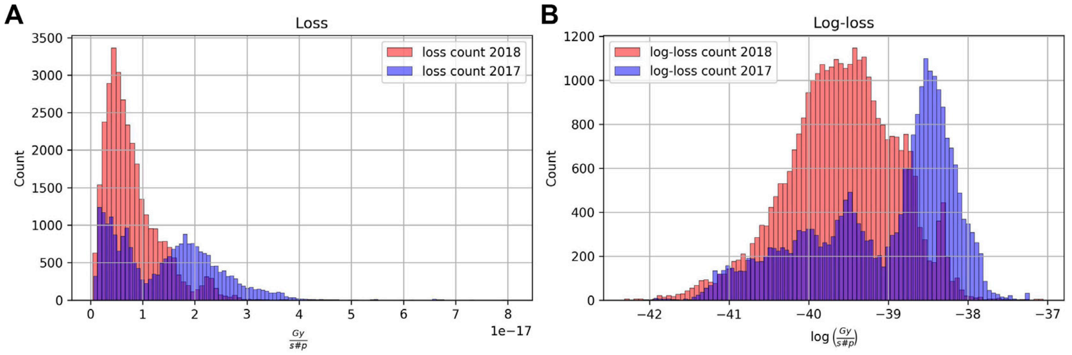


FIGURE 2
(A) original losses (normalized by intensity), **(B)** log-losses (normalized by intensity), for the years 2017 and 2018. The log-transform helps to reduce the skewness. Note that several modes on the pictures are due to changes of the losses level because of changes in control parameters, see e.g. Figure 3.

energy ramp process. During "PRERAMP," measures are taken to prepare for the energy ramp, such as retracting the injection collimators, adjusting the feedback reference and loading other machine settings. During the "PRERAMP" mode machine is in stable conditions, and thus it should be supposedly the easiest beam mode to model losses. Nevertheless, on practice, it occurred to be challenging [1]. For each fill during the years 2017 and 2018 the time series of "PRERAMP" mode vary in length (see Figure 1).

Throughout the paper, we assume that a logarithmic transformation is applied to the losses normalized by intensity. Further, we omit "log" and "normalized by intensity" while mentioning losses. The logarithmic transform is generally applied to reduce the skewness of the distribution (see the loss in original scale in Figure 2A and the log-transformed loss in Figure 2B). For the losses, the log-transformation is partially motivated by the fact that losses normalized by intensity are produced from the particle count data. For the convenience of further analysis, we assume that the log losses are Gaussian. Alternatively, one could follow the Generalized Linear Model approach, assuming the Poisson distribution of the count data, e.g. as in [18].

Several quantities are recorded during the experiment, among which we will use the following variables as the input/controlled parameters for the loss models.

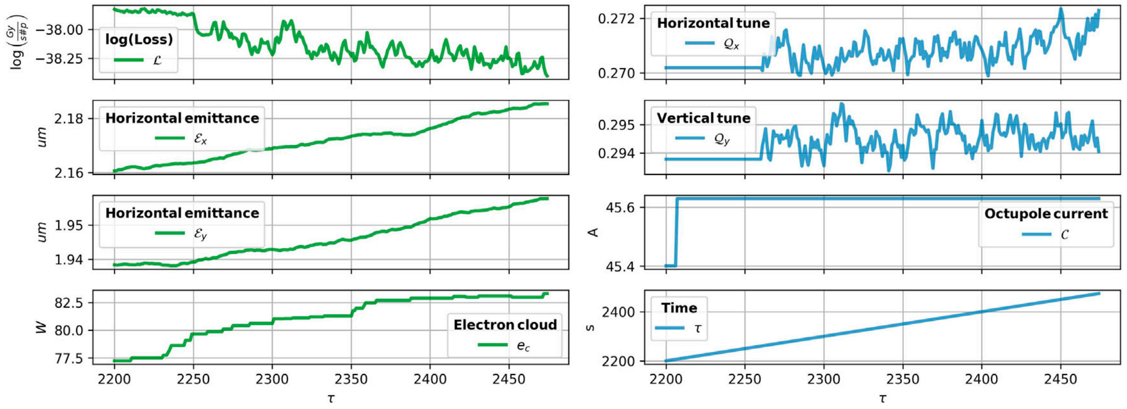


FIGURE 3
An example of PRERAMP mode time series for the fill 6,243 in 2017. The time series of vertical and horizontal tunes and octupole currents are shown in blue, observations of the loss, vertical and horizontal emittance and the sum of heatload measurements are shown in green.

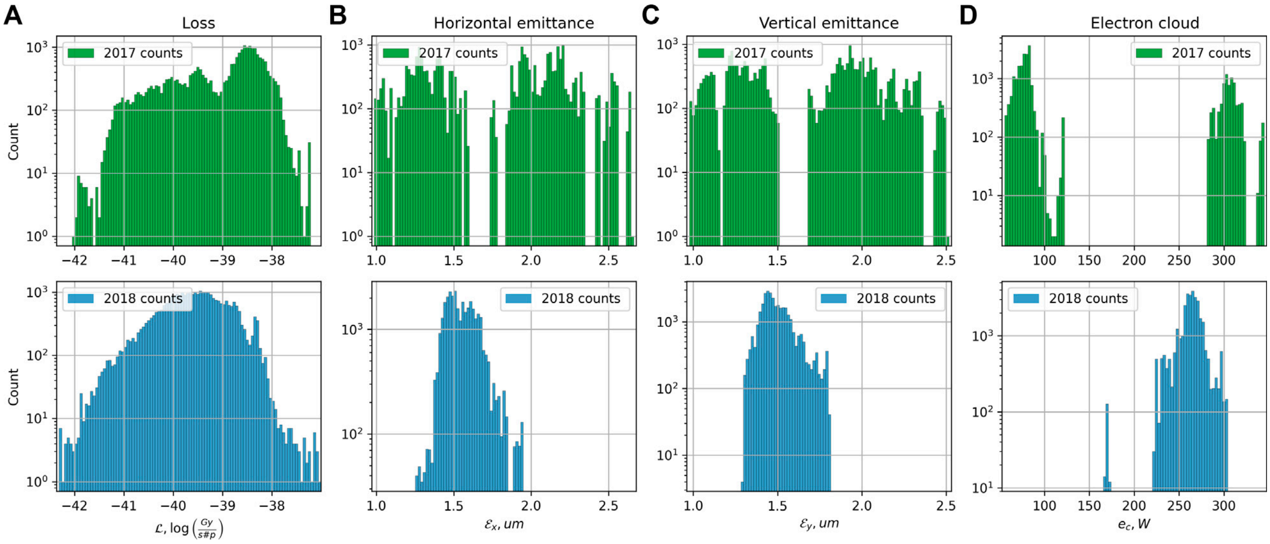


FIGURE 4
Histograms of the observations in 2017 and 2018. **(A)** Logarithm of loss normalized by intensity, **(B)** horizontal emittance, **(C)** vertical emittance, **(D)** heatload induced by electron cloud. Almost all the variables demonstrate different ranges of values for both years, e.g. see the sum of the heatloads.

- $\mathcal{Q}_x$ and $\mathcal{Q}_y$ — vertical and horizontal tunes,
- $\mathcal{C}$ — current in octupole magnets.

Non-controlled observed variables are

- $\mathcal{L}$ — logarithm of loss divided by intensity at BLM 6L7 for the beam 1,
- $\mathcal{E}_x$, $\mathcal{E}_y$ — horizontal and vertical emittance,
- e_c—sum of the heatload measurements over the 8 sectors of the LHC,
- τ—time since the beam injection.

Remark: Note that the tunes are actually controlled by quadrupole magnet currents, the field decays due to persistent currents and a feedback system that keeps the tune values measured with a Base-Band Tune (BBQ) measurement system at a constant value [19]. Although the control that we have over the tunes is an indirect one, we nonetheless treat the tunes (measured online in our dataset) as one of the control variables of the system. For the octupoles, we use the currents

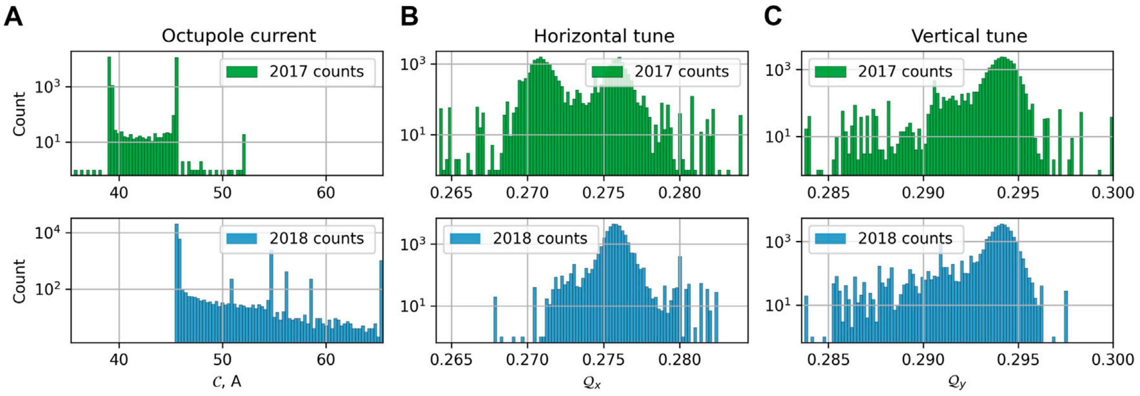


FIGURE 5
Histograms of the observations in 2017 and 2018. **(A)** Octupole current, **(B)** horizontal tune, **(C)** vertical tune. Almost all the variables demonstrate different ranges of values for both years, e.g. see the octupole current $\mathcal{C}$.

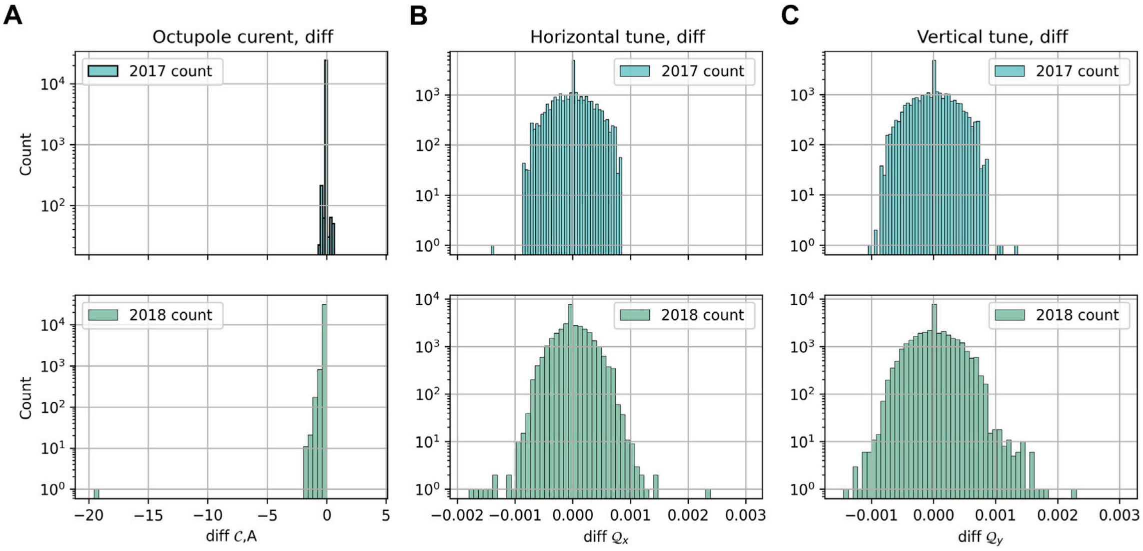


FIGURE 6
Histogram of increments of **(A)** octupole current, **(B)**horizontal tune and **(C)** vertical tune in 2017 and 2018. The fill 7,236 is a big outlier to the left in the histogram of octupole current in 2018.

because there is no measurement of the tune spread associated with these elements. Also, time since the beam injection, though not controlled, is used as an input parameter.

For an example of evolution of the loss and other variables during the "PRERAMP" mode within one fill in the year 2017, see Figure 3. Typically, one significant modification of current and tunes occurs during "PRERAMP."

The data for the years 2017 and 2018 differ in the ranges of input parameters used and in the level of losses: in Figure 4, non-controlled variables have different distributions, e.g. losses (Figure 4A) in 2017 were overall higher than in 2018; input variables, shown in Figure 5, had different ranges, e.g. see octupole current in Figure 5A, as well as increments of input variables in Figure 6, e.g. octupole current was only decreasing in 2018, both increasing and decreasing in 2017. Some of the control parameters, such as octupole current, change quite rarely, e.g. in the data of 2017, it changes from one level to another in "PRERAMP" time series only in half of the fills. This already

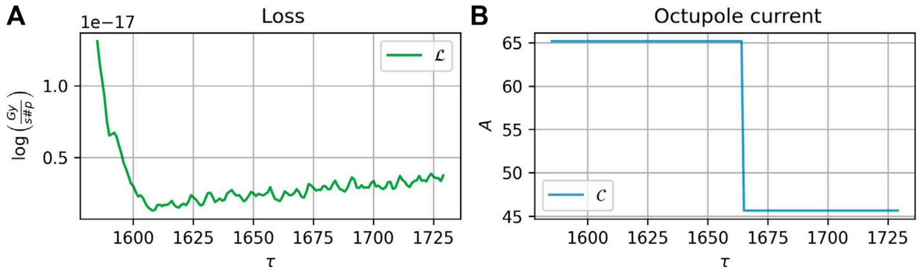


FIGURE 7
Fill 7,236 from 2018, “PRERAMP” mode, unexpected behavior: no change in losses **(A)** after large decrease in octupole current **(B)**.

suggests that applying the model trained on the data from 2017 to the data of 2018 require to extrapolate which is known to be quite challenging. We therefore privilege the use of simple (and thus robust) modeling techniques.

Starting from the injection of the beam, multiple effects occur in the system, which cannot be fully described analytically, and have been so far observed in separate experiments, e.g. changes in the distribution of particle tunes in time due either to drift or to changes of the control parameters and for example associated with crossing resonance lines. Given the limited information about such events and about long-term dependence between control variables and losses, we consider a simple, but robust linear approach to model the dependence of losses on its prior history and on controlled and possibly uncontrolled variables. An example of such model in the 1D case which models losses as a function of tunes and octupole current would be of the form

$$\begin{aligned}\mathcal{L}_t = {} & \alpha_1\mathcal{L}_{t-1} + \alpha_2\mathcal{L}_{t-2} + \ldots \\ & + \beta_1\mathcal{Q}_{x,t-1} + \beta_2\mathcal{Q}_{x,t-2} + \ldots \\ & + \gamma_1\mathcal{Q}_{y,t-1} + \gamma_2\mathcal{Q}_{y,t-2} + \ldots \\ & + \zeta_1\mathcal{C}_{t-1} + \zeta_2\mathcal{C}_{t-2} + \ldots \\ & + \text{noise.}\end{aligned} \tag{1}$$

with a general form of correlated noise process this is an instance of an autoregressive model with moving average and exogeneous variables, a so-called ARMAX model [20].

In a model of the form above, given that losses are closely related to emittances and will be affected by electron cloud, and in spite of the fact that these variables cannot be controlled, we chose to include them among the exogenous variables.

Finally, we will consider a multivariate time series model for the losses together with the emittances and electron cloud induced heatloads to try and predict their evolution jointly from controlled variables. This is motivated by the fact that we wish to obtain a model that captures the effect of the control variables so that if we rely on the values of emittances and electron cloud at some time *t*–*l* it should itself be predicted from the control variables at prior time points. This kind of multivariate extension is known as a vector ARMAX–or VARMAX–model.

To estimate the parameters of this type of model, we will exploit the relationship between VARMAX models and the Kalman Filter (KF).

The paper is organized as follows. In Section 2, after introducing more precisely the form of the different models we discuss VARMAX models and their relation to so-called state space models and in particular to the Kalman filter. Different parameterizations will lead us to consider a general KF with time-varying parameters, including a KF model with parameters depending on the input variables. We will consider different ways of regularizing the coefficients, and discuss a general Expectation-Maximization (EM) procedure for the estimation of parameters. Section 3 is devoted to the results of numerical experiments and to comparisons of the models.

2 Models

In this paper we consider several variants of the model described in Eq. 1.

KF1: First, we verify whether it is possible to construct a predictive model of the current losses as a function of the recent histories of the losses and of the control variables, as in KF1. More precisely, we consider a 1D linear model of losses.

$$\begin{aligned}\mathcal{L}_t = {} & \alpha_1\mathcal{L}_{t-1} + \alpha_2\mathcal{L}_{t-2} + \ldots + \alpha_p\mathcal{L}_{t-p} \\ & + \beta_0\mathcal{Q}_{x,t} + \beta_1\delta\mathcal{Q}_{x,t} + \beta_2\delta\mathcal{Q}_{x,t-1} + \cdots + \beta_L\delta\mathcal{Q}_{x,t-L+1} \\ & + \gamma_0\mathcal{Q}_{y,t} + \gamma_1\delta\mathcal{Q}_{y,t} + \ldots + \gamma_L\delta\mathcal{Q}_{y,t-L+1} \\ & + \zeta_0\mathcal{C}_t + \zeta_1\delta\mathcal{C}_t + \cdots + \zeta_L\delta\mathcal{C}_{t-L+1} + \text{noise,}\end{aligned} \tag{KF1}$$

where δ stands for taking the first order differences, i.e. $\delta\mathcal{Q}_{x,t} = \mathcal{Q}_{x,t} - \mathcal{Q}_{x,t-1}$, and p and L are the depths of the histories of observations of the outputs and inputs correspondingly, which we include in the model.

KF1*: Since we are given additional observations of emittances and electron cloud (heatload sum), we could include them into the input variables to see whether their presence help to model the losses better, thus we will also consider a model.

$$
\begin{aligned}
\mathcal{L}_t = {} & \alpha_1\mathcal{L}_{t-1} + \alpha_2\mathcal{L}_{t-2} + \ldots + \alpha_p\mathcal{L}_{t-p} \\
& + \beta_0\mathcal{Q}_{x,t} + \beta_1\delta\mathcal{Q}_{x,t} + \beta_2\delta\mathcal{Q}_{x,t-1} + \cdots + \beta_L\delta\mathcal{Q}_{x,t-L+1} \\
& + \gamma_0\mathcal{Q}_{y,t} + \gamma_1\delta\mathcal{Q}_{y,t} + \ldots + \gamma_L\delta\mathcal{Q}_{y,t-L+1} \\
& + \zeta_0\mathcal{C}_t + \zeta_1\delta\mathcal{C}_t + \cdots + \zeta_L\delta\mathcal{C}_{t-L+1} \qquad \text{(KF1*)} \\
& + \eta_0\mathcal{E}_{x,t} + \eta_1\delta\mathcal{E}_{x,t} + \eta_2\delta\mathcal{E}_{x,t-1} + \cdots + \eta_L\delta\mathcal{E}_{x,t-L+1} \\
& + \theta_0\mathcal{E}_{y,t} + \theta_1\delta\mathcal{E}_{y,t} + \ldots + \theta_L\delta\mathcal{E}_{y,t-L+1} \\
& + \kappa_0 e_{c,t} + \kappa_1\delta e_{c,t} + \cdots + \kappa_L\delta e_{c,t-L+1} + \text{noise}\,.
\end{aligned}
$$

KF4: Next, we could add the additional variables into the output together with the losses.

$$
\begin{aligned}
\begin{pmatrix}\mathcal{L}_t\\ \mathcal{E}_{x,t}\\ \mathcal{E}_{y,t}\\ e_{c,t}\end{pmatrix} = {} & A_1^\circ\begin{pmatrix}\mathcal{L}_{t-1}\\ \mathcal{E}_{x,t-1}\\ \mathcal{E}_{y,t-1}\\ e_{c,t}\end{pmatrix} + \cdots + A_p^\circ\begin{pmatrix}\mathcal{L}_{t-p}\\ \mathcal{E}_{x,t-p}\\ \mathcal{E}_{y,t-p}\\ e_{c,t}\end{pmatrix} \qquad \text{(KF4)} \\
& + B_0^\circ\begin{pmatrix}\mathcal{Q}_{x,t}\\ \mathcal{Q}_{y,t}\\ \mathcal{C}_{y,t}\\ \tau_t\end{pmatrix} + B_1^\circ\begin{pmatrix}\delta\mathcal{Q}_{x,t}\\ \delta\mathcal{Q}_{y,t}\\ \delta\mathcal{C}_{y,t}\end{pmatrix} \\
& + \ldots + B_L^\circ\begin{pmatrix}\delta\mathcal{Q}_{x,t-L+1}\\ \delta\mathcal{Q}_{y,t-L+1}\\ \delta\mathcal{C}_{y,t-L+1}\end{pmatrix} + \text{noise}.
\end{aligned}
$$

KF4-quad: is an extension of KF4, where the matrices A_i° and B_i° depend on control parameters. In order to fit the model parameters under additional structural assumptions we first consider equivalent formulation of the class of models, which makes it possible to efficiently estimate the parameters. The model is discussed in more detail in Section 2.2.2.

2.1 VARMAX and state space modeling

Formally, the models introduced above are all particular instances of a Vector AutoRegressive Moving Average model with eXogenous variables (VARMAX). VARMAX models can be written as follows:

$$
\begin{aligned}
y_t = {} & \sum_{i=1}^{p} A_i^\circ y_{t-i} + \left[B_0^\circ u_t + \sum_{i=0}^{L} B_{i+1}^\circ \delta u_{t-i}\right] \\
& + \sum_{i=0}^{m} C_i^\circ \xi_{t-i}, \quad t = 1,\ldots,T. \qquad (2)
\end{aligned}
$$

the first Vector AutoRegression part represents the belief that the past observations could be predictive of future losses. The second sum, “X”-part in VARMAX, assumes linear dependence on control variables u_t and their retrospective changes. The last term is a stationary Moving Average process which is a sum of independent random (standard Gaussian) variables (shocks) ξ_t in the past.

A response vector (variable) $y_t \in \mathbb{R}^n$ in VARMAX corresponds to:

- a scalar $\mathcal{L}_t$ in the case (KF1) and (KF1*) and $n = 1$,
- a vector $[\mathcal{L}_t, \mathcal{E}_{x,t}, \mathcal{E}_{y,t}, e_{c,t}]^\top$ with the loss, horizontal and vertical emittances and electron cloud ($n = 4$) for the case of (KF4) and KF4-quad.

We will further assume that y_t is a vector with a 1D case as a sub-case. The control variable u_t contains different sets of variables depending on the considered model:

- $u_t \in \mathbb{R}^q = [\mathcal{Q}_{x,t}, \mathcal{Q}_{y,t}, \mathcal{C}_{y,t}, \tau_t]^\top$ a vector with horizontal and vertical tunes, currents in octupole magnets, and time passed since the end of injection observed at time t.
- Vectors δu_{t-l} contain l-lagged differences of u_{t-l}, i.e. $\delta u_{t-l} = [\delta\mathcal{Q}_{x,t-l}, \delta\mathcal{Q}_{y,t-l}, \delta\mathcal{C}_{y,t-l}]^\top$.

For estimation we further denote stacked matrices in exogenous term as $B^\circ = [B_0^\circ, B_1^\circ, \ldots, B_L^\circ]$ and stacked vector of all exogenous components as $\nu_{L,t} = [u_t^\top, \delta u_t^\top, \ldots u_{t-L}^\top]^\top$. In these notations (2) reads as

$$
y_t = \sum_{i=1}^{p} A_i^\circ y_{t-i} + \sum_{i=1}^{p} B_i^\circ \nu_{L,t-i} + \sum_{i=0}^{m} C_i^\circ \xi_{t-i}, \quad t = 1,\ldots,T. \qquad (3)
$$

motivated by VARMAX, we will further consider more general state space models, where the dimension of hidden state could be different from dimension of observations.

2.2 State space models

State space models represent the state of a dynamical system by a latent variable, which varies in time and is different from the input and output variables. The most well known model in this family is the Kalman Filter model. State space models are relevant to model time series with rich structure, and there is in particular a well known connection between (V)ARMAX models and Kalman filters that we will exploit in this work.

Consider the 1D autoregressive moving average ARMA (3,2) model (with lag parameters $p = 3$, $m = 2$):

$$
\begin{aligned}
y_t = {} & a_1^\circ y_{t-1} + a_2^\circ y_{t-2} + a_3^\circ y_{t-3} \\
& + c_0^\circ \xi_t + c_1^\circ \xi_{t-1} + c_2^\circ \xi_{t-2}, \quad t = 1,\ldots,T, \qquad (4)
\end{aligned}
$$

where ξ_t are independent standard Gaussian random variables. The ARMA model can be viewed as a special case of the state space model [21] with hidden vector $x_t = (x_{1,t}\, x_{2,t}\, x_{3,t})^\top$:

$$
\begin{aligned}
y_t &= \begin{pmatrix}1 & 0 & 0\end{pmatrix}\begin{pmatrix}x_{1,t}\\ x_{2,t}\\ x_{3,t}\end{pmatrix}, \\
\begin{pmatrix}x_{1,t}\\ x_{2,t}\\ x_{3,t}\end{pmatrix} &= \begin{pmatrix}a_1 & 1 & 0\\ a_2 & 0 & 1\\ a_3 & 0 & 0\end{pmatrix}\begin{pmatrix}x_{1,t-1}\\ x_{2,t-1}\\ x_{3,t-1}\end{pmatrix} + \begin{pmatrix}c_0\\ c_1\\ c_2\end{pmatrix}\xi_t.
\end{aligned} \qquad (5)
$$

the first measurement equation describes the relation between observations and hidden state x_t of the system, and the second transition equation describes hidden evolution of the state x_t. The equivalence between (4) and (5) with $a_i = a_i^\circ$ and $c_i = c_i^\circ$ can be easily seen, if one substitutes $x_{2,t-1}$ and then $x_{3,t-2}$ in the first equation of transition equations using the rest of equations. Thus, in such a representation the hidden state components equal lagged output. The state space representation (5) is not unique, e.g. consider

$$
\begin{aligned}
y_t &= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \end{pmatrix} + c_0 \xi_{t-1} \\
\begin{pmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \end{pmatrix} &= \begin{pmatrix} a_1 & 1 & 0 \\ a_2 & 0 & 1 \\ a_3 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_{1,t-1} \\ x_{2,t-1} \\ x_{3,t-1} \end{pmatrix} + \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \xi_t.
\end{aligned} \tag{6}
$$

in this state space representation equivalence to (4) is slightly less straightforward. One can check that $a_i = a_i^\circ$, $c_0^\circ = c_0$, $c_1^\circ = c_1 - a_1 c_0$, $c_2^\circ = c_2 - a_2 c_0$, $c_3 {-} a_3 c_0 = 0$.

One can see that, for ARMA and corresponding state space representations, each component of the hidden state vector is related to the lagged output, i.e. the first component represents the relation to lagged-1 output, and so forth.

In the same way it is possible to write the VARMAX model (3) in an similar to state space form:

$$
\begin{aligned}
y_t &= D x_t + C_0^\circ \xi_t, \\
x_t &= \tilde{A} x_{t-1} + \tilde{B} \nu_{L,t-i} + \tilde{C} \xi_{t-1},
\end{aligned} \tag{7}
$$

where $x_t \in \mathbb{R}^h$ with $h = \max(p, m)$, $\xi_t \in \mathcal{N}(0, I_n)$.

The matrices in Eq. 7 can be defined as follows

$$
D = \begin{pmatrix} I_h \\ 0_{n-h \times h} \end{pmatrix}, \text{ if } n > h, \text{ otherwise } D = \begin{pmatrix} I_n & 0_{n \times h-n} \end{pmatrix},
$$

where I_h is a squared identity matrix with h columns and rows, $0_{m \times n}$ is a matrix with zeros of the noted size;

$$
\begin{aligned}
\tilde{A} &= \begin{pmatrix} A_1^\circ & I_n & 0 & \dots & 0 \\ A_2^\circ & 0 & I_n & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{h-1}^\circ & 0 & 0 & \dots & I_n \\ A_h^\circ & 0 & 0 & \dots & 0 \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} B_1^\circ \\ B_2^\circ \\ \vdots \\ B_{h-1}^\circ \\ B_h^\circ \end{pmatrix}, \\
\tilde{C} &= \begin{pmatrix} C_1^\circ \\ C_2^\circ \\ \vdots \\ C_{r-1}^\circ \\ C_r^\circ \end{pmatrix} + \begin{pmatrix} A_1^\circ C_0^\circ \\ A_2^\circ C_0^\circ \\ \vdots \\ A_{h-1}^\circ C_0^\circ \\ A_h^\circ C_0^\circ \end{pmatrix}.
\end{aligned} \tag{8}
$$

The state-space representation allows for the use of the efficient inference procedures of the Kalman Filter in the case of Gaussian noise for the known parameters. When the transition and observation matrices, as well as the noise matrices, are not known, one can use the classical Expectation-Maximization algorithm for their estimation. We discuss briefly their application to the inference and estimation of the parameters in our models.

2.2.1 State space model for KF1, KF1* and KF4

We will be interested in the estimation of the a state space model for the (KF1), (KF1*) and (KF4) models, which will be done in the classical form of the Kalman Filter model:

$$
\begin{aligned}
y_t &= D x_t + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, R), \\
x_t &= A x_{t-1} + B \nu_{L,t} + \eta_t, \quad \eta_t \sim \mathcal{N}(0, V),
\end{aligned} \tag{9}
$$

where ν_t contains control parameters and their lagged difference up till lag L as in Eq. 2. We assume $x_t \in \mathbb{R}^h$, where h is a multiple of n.

We use the standard form of the Kalman filter here, instead of Eq. 8 which has a single noise term, as these representations are generally equivalent (see [22, 23]). See the remark in the Supplementary Material about the conversion between two state space forms (9) and (8).

2.2.2 State space model KF4-quad with control dependent transitions

In the models we considered so far, and which are motivated initially by a VARMAX model, the exogeneous variables induce linear shifts in state space *via* the term $B\nu_{t,L}$. However, another fairly natural way that the control variable (or control parameters) can enter the model is *via* the autoregressive coefficients of the VARMAX model or *via* the transition matrices of the state-space model itself. This motivated us to consider a model which combines both effects: we keep a model of the previous general form, but make the matrices A and B now linearly dependent on u_t. We however limit ourselves to an instantaneous effect.

We thus consider a Kalman filter model of the form:

$$
\begin{aligned}
y_t &= D\, x_t + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, R), \\
x_t &= A(u_t)\, x_{t-1} + B(u_t)\, \nu_t + \eta_t, \quad \eta_t \sim \mathcal{N}(0, V).
\end{aligned} \tag{10}
$$

with matrices $A\,(u_t)$ and $B\,(u_t)$ now being linear functions of the control variables

$$
A(u_t) = A_0 + \sum_{j=1}^{q} A_j u_{tj} \quad \text{and} \quad B(u_t) = B_0 + \sum_{j=1}^{q} B_j u_{tj}. \tag{11}
$$

This model is now non-linear, and in particular it includes cross-terms of the form $u_t u_{t-j,i}$ and $u_t x_{t-j,i}$.

This formulation has q+1 times more parameters for the state transitions, and regularization becomes necessary to prevent overfitting. Several regularizations would be possible, but given that our model is parameterized by several matrices, we propose to use a matrix regularizer that encourages these matrices to be low-rank (or equal to 0 which is rank 0). More precisely, we propose to use trace norm regularization [24, 25]. The trace norm of a matrix (aka nuclear norm), is a matrix norm which is defined as the ℓ_1-norm of the singular values of the matrix. The trace norm $\|A\|^*$ of a matrix A can be equivalently defined by

TABLE 1 Hyperparameters estimates from 10-fold cross validation for the models (KF1), (KF1*), (KF4).

	L, 2017	h, 2017	L, 2018	h, 2018
KF1	90	1	55	1
KF1*	85	1	80	1
KF4	80	16	80	16

$$\|A\|_* = \mathrm{tr}\Big(\big(A^\top A\big)^{1/2}\Big),$$

where tr denotes the trace of a matrix. Note that the more classical Tikhonov regularization is here $\mathrm{tr}(A^\top A)$, which is equal to the squared ℓ_2 norm of the singular values of A and that the ℓ_0 pseudo-norm of the singular values of a matrix A is the rank of the matrix A. So the trace norm is to rank as the ℓ_1 norm to ℓ_0. The trace norm is a convex regularizer but induces sparsity in the spectrum of the matrix, in a similar way as the ℓ_1 norm induces sparsity which means that it becomes low-rank.

In the end, we wish for $A(u)$ and $B(u)$ to be low rank but regularizing directly these matrices with the trace norm leads to an optimization problem which is not so easy to optimize. So, instead, we apply the regularization to all the individual matrices A_j and B_j. We denote

$$\Omega\Big(\big(A_j, B_j\big)_{j=0..q}\Big) = \gamma_0\|A_0\|_* + \delta_0\|B_0\|_* + \gamma\sum_{j=1}^{q}\|A_j\|_* + \delta\sum_{j=1}^{q}\|B_j\|_*. \tag{12}$$

The details on parameter estimation with the Expectation Maximization algorithm for KF1, KF1*, KF4 and KF4-quad can be found in the Supplementary Material.

3 Evaluation

3.1 Datasets

The parameters of the model were estimated using "PRERAMP" observations from 1 year and then tested on the data of another year. The data from 2017 contains 105 time series corresponding each to an LHC fill; in 2018, 144 fills are available. The duration of the "PRERAMP" phase varies in 2017 from 65 to 490 s, whereas in 2018 it varies from 67 to 1046 s, with a typical duration which is slightly larger in 2017, see Figure 1.

First, we take the data from 2017 as the training dataset and the data of 2018 as the testing dataset. After the selection of the hyperparameters (cf Section 3.1.1), the model parameters are computed from the full training dataset. Then, we validate the trained model on the data from 2018. Next, we repeat the validation for 2018 data as a training set and 2017 as testing to check whether we can also predict the loss of 2017 from 2018.

Remark: We excluded fill 7236 from the dataset of 2018 for the second validation. The main reason for exclusion is that during that fill, an abnormally high jump in octupole current occurred (Figure 7B), which unexpectedly did not lead to a noticeable change in the loss (Figure 7A). There was no other evidence of such events in the datasets and our analysis showed that other variables present in the dataset do not explain such a behavior of the loss. Extending the models for the case of fill 7,236 would require additional understanding of the reasons for such loss behavior or more data on similar events. This can be also seen from Figure 6A, where the upper histogram is for the increments of control parameters in 2017 and the lower one is for increments in 2018. The change of the octupole current during fill 7,236 is close to the value −20 A and is distinctly very far from

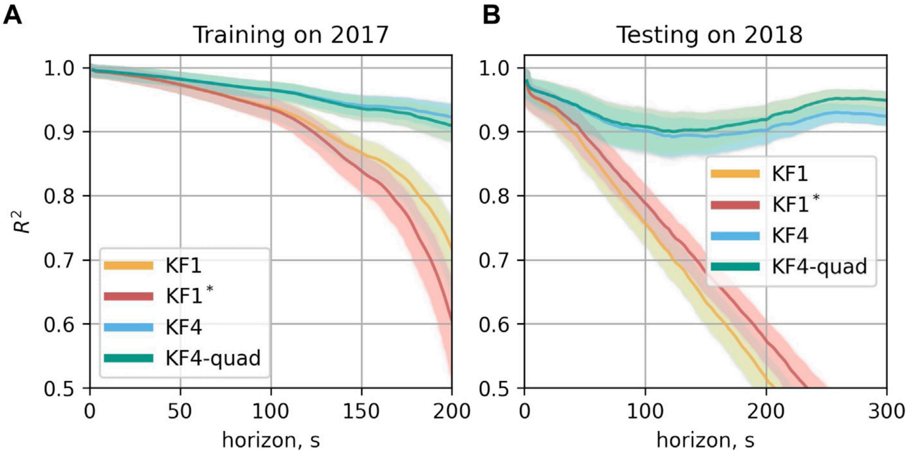


FIGURE 8
R^2-score as a function of the forecast horizon. **(A)** On the training dataset, here from 2017 **(B)** on the test data from 2018. The mean value of R^2 is shown for each horizon in the darker color. The mean value was computed from 1,000 bootstrapped estimates of R^2, which are shown in the light color.

the main range of values. Additionally, we can note that apart from the fill 7,236, the changes in octupole current in 2018 are mostly negative, whereas, for the year 2017, the octupole current was both increasing and decreasing. Thus one can anticipate that in terms of octupole input, forecasting the losses of 2018 from the model built from the data from 2017 might be an easier task than when swapping the datasets, since it involves extrapolation to the larger range of impulse values in the input.

Data Transformations: For the losses normalized by intensity, we apply a log transformation. Next, the input variables of the training dataset are scaled to be in the interval (−1, 1). The output variables of the training dataset are centered and normalized. For validation, both input and output are centered and scaled with the parameters obtained for the training dataset.

3.1.1 Hyperparameters estimation

For the KF models, we have two hyperparameters to estimate: the number of lags L in ν_t in Eq. 9 and the dimension of the hidden space h. To find their estimates we use a 10-fold cross-validation procedure on the training dataset to minimize the mean absolute error (MAE) of the prediction over the parameters in the grid. The MAE for the prediction $\hat{y}_t$ of the ground truth (1D) values y_t is defined as MAE $= \frac{1}{T}\sum_t |y_t - \hat{y}_t|$. We estimate the quality of the predictions of the models built from 9/10 of the fill time series on the rest of the data. Namely, on each 9/10 of fills, we make an EM estimation of the KF parameters. We set $T_0 = 10$. Next on the rest of 1/10 of fills, for each fill, we use the KF equations and smoother applied to the first T_0-th observation of the time series to get an initial estimate of the hidden process. Starting from T_0+1-th observation, we run the KF model state

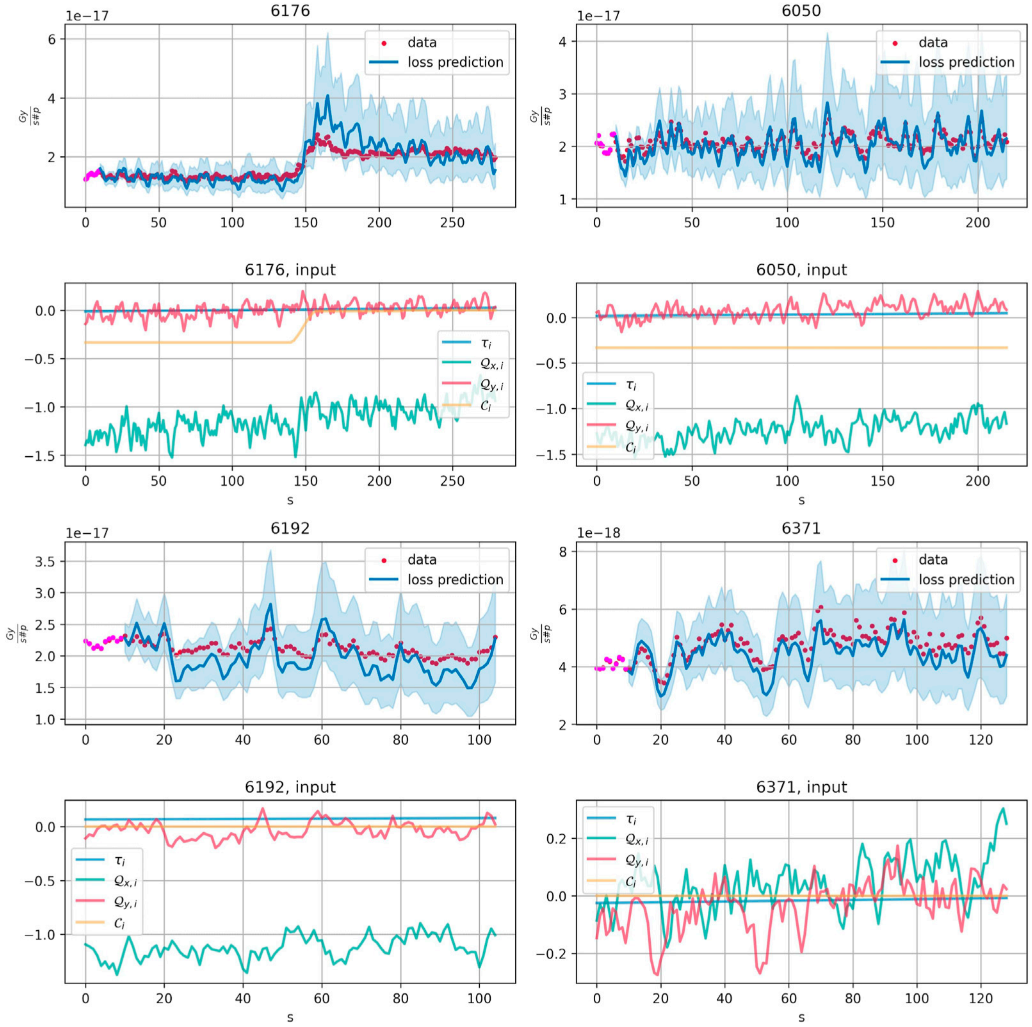


FIGURE 9
KF4 trained on the data of 2017, prediction on the fills 6176, 6050, 6192, and 6371 of 2018 and corresponding input control parameters. Pink points correspond to T_0 observations which the model uses to get initial KF smoother results. Further model propagates without seeing the loss and other output values, with control parameters given as the input. Two standard deviation confidence bands are shown in light blue.

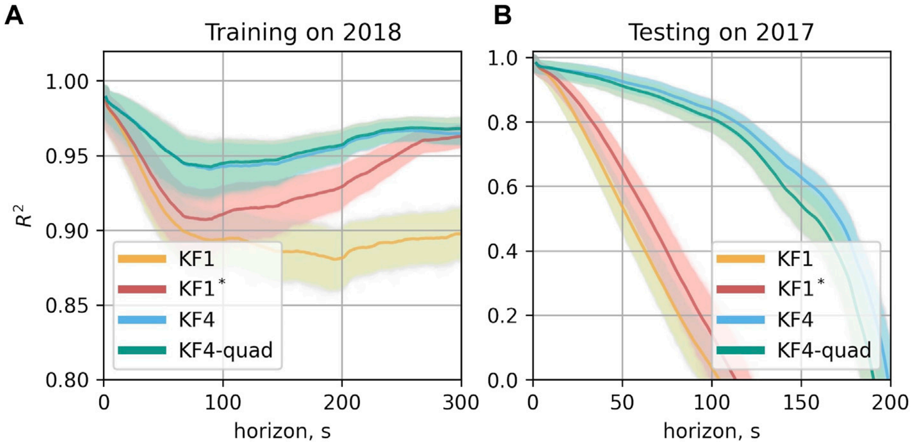


FIGURE 10
R^2-score as a function of the forecast horizon. **(A)** On the training dataset, here from 2018 **(B)** on the test data in 2017.The mean value of R^2 is shown for each horizon in the darker color. The mean value was computed from 1,000 bootstrapped estimates of R^2, which are shown in the light color.

evolution dynamic forward in time to propagate the prediction, giving the control observations as input. This way, the model "sees" only the first T_0 data points of the output from the fill and the input variable at each new prediction time point. We stack all the predictions for each of the 10 folds to compare them with the corresponding true values, i.e. in each fold we compute MAE: $\frac{1}{\sum_j T_j}\sum_j \sum_{t=T_0+1}^{T_j} |\hat{y}_t^{(j)} - y_t^{(j)}|$ where T_j is the length of fill j, and j runs over the fills in the fold. Finally, the hyperparameters are selected *via* minimization of the mean MAE across folds. The hyperparameters selection was carried out on the same intervals of prediction, that is, we fixed the largest history L_{max}, and the first data point to predict in the fill of the other year was L_{max} for all the models. We fixed $L_{max} = 90$ to have sufficiently long forecasting horizons and to have enough data to train the models.

The results can be found in Table 1. The hyperparameter selection procedure favored deep histories of the input parameters and their changes, thus the changes in control parameters might have a relatively long-term effect on the loss evolution. For KF4-quad, h and L were set to be the same as the ones found for KF4, and optimization of regularization hyperparameters was carried out in the same way by optimizing the MAE on the grid. See Supplementary Figure S1 in the Supplementary Material for illustration of 10-fold cross-validated MAE behaviour for different versions of Eq. 9 for a range of parameters h and L based on the data of 2017 or 2018 and Supplementary Figure S2 for selection of γ and δ in Eq. 12 for KF4-quad.

Remark: We compute MAE over different forecast horizons, as opposed to instantaneous one-step ahead forecasting for hyperparameter selection. This is motivated by the fact that minimization of one-step-ahead prediction error tends to select models which better follow local trends. For example, a simple forecast which is just repeating the previous loss observation would often have quite a low one-step ahead forecasting error, whereas for long-term forecasting this is not the case.

3.2 Evaluation of predictive ability for different time horizons

We compare the variants of the Kalman Filters: (KF1), (KF1*), (KF4), and KF4-quad. As evaluation metric of losses prediction for different time horizons we compute R^2-score, which is defined as

$$R^2 = 1 - \frac{\sum_t (y_t - \hat{y}_t)^2}{\sum_t (y_t - \bar{y}_t)^2},$$

where $\hat{y}_t$ is the predicted value of y_t and $\bar{y}_t = \frac{1}{T}\sum_t y_t$ for the models trained on one of the datasets either of 2017 or 2018. First, we fit hyperparameters and parameters of the models on the training dataset. For each of the training and testing datasets, for each fill, we fix the horizon H. Next, for each time point t of the fill where $t \in \{T_{0+1}, \ldots, T - H\}$, where T is the duration of the fill, , we use KF equations and smoother to obtain an estimate of the hidden state at t, from the preceding T_0 observations. Starting from t, we propagate the model to predict the evolution till $t + H$. Thus we get a collection of predictions at horizon H based on the data of different fills. From all the predictions at horizon H we compute a bootstrap estimate of the mean R^2 [26] obtained from 10^3 subsamples of 10^3 predictions and corresponding observations.

We limited the predictions to the time horizon of 200 s for the dataset of 2017 and to the horizon of 300 s for the dataset

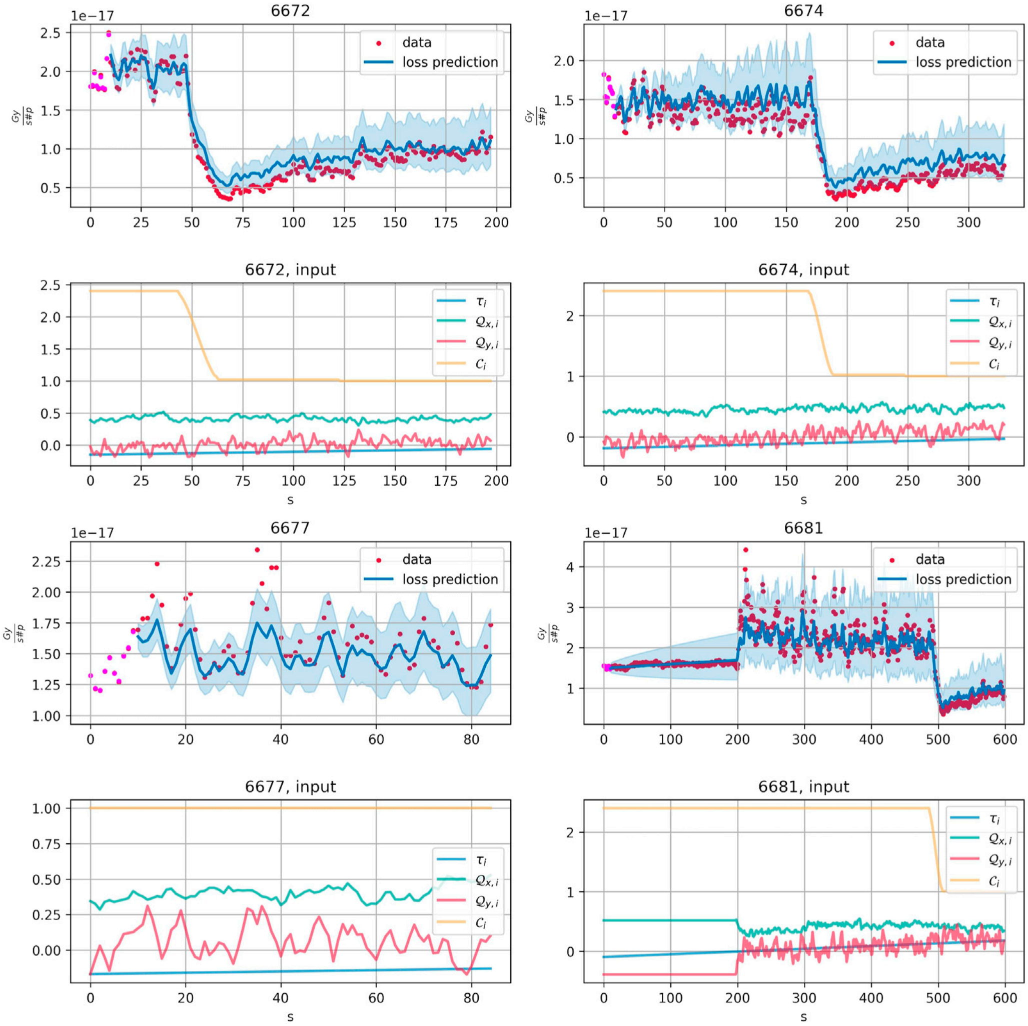
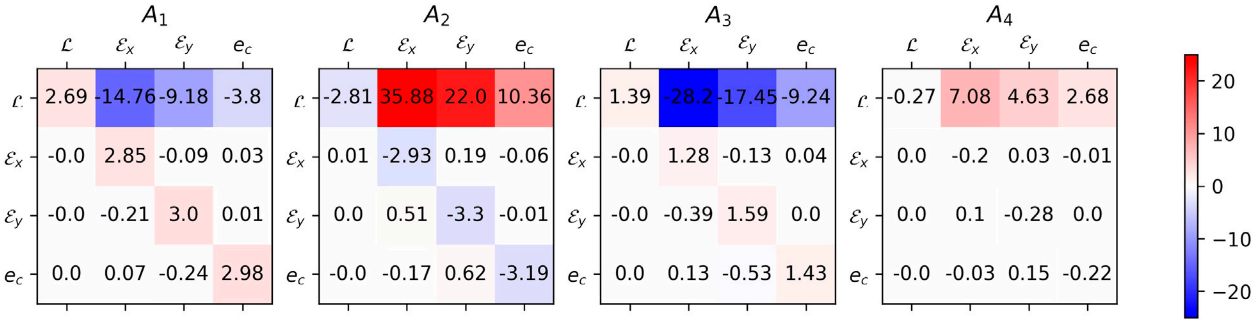


FIGURE 11
KF4 trained on the data of 2018, prediction on the fills 6726, 6674, 6677, and 6681 of 2018 and corresponding input control parameters. Pink points correspond to T_0 observations which the model uses to get initial KF smoother results. Further model propagates without seeing the loss and other output values, with control parameters given as the input. Two standard deviation confidence bands are shown in light blue.



A_1

	$\mathcal{L}$	$\mathcal{E}_x$	$\mathcal{E}_y$	e_c
$\mathcal{L}$	2.69	-14.76	-9.18	-3.8
$\mathcal{E}_x$	-0.0	2.85	-0.09	0.03
$\mathcal{E}_y$	-0.0	-0.21	3.0	0.01
e_c	0.0	0.07	-0.24	2.98

A_2

	$\mathcal{L}$	$\mathcal{E}_x$	$\mathcal{E}_y$	e_c
$\mathcal{L}$	-2.81	35.88	22.0	10.36
$\mathcal{E}_x$	0.01	-2.93	0.19	-0.06
$\mathcal{E}_y$	0.0	0.51	-3.3	-0.01
e_c	-0.0	-0.17	0.62	-3.19

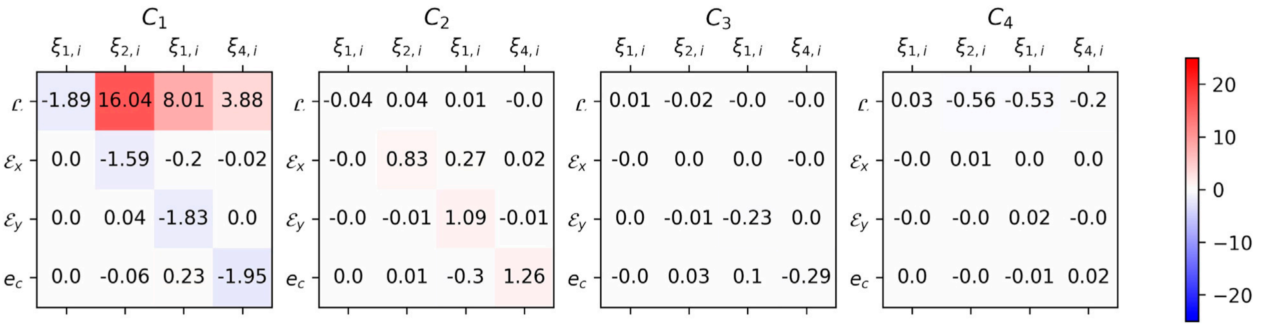
A_3

	$\mathcal{L}$	$\mathcal{E}_x$	$\mathcal{E}_y$	e_c
$\mathcal{L}$	1.39	-28.2	-17.45	-9.24
$\mathcal{E}_x$	-0.0	1.28	-0.13	0.04
$\mathcal{E}_y$	-0.0	-0.39	1.59	0.0
e_c	0.0	0.13	-0.53	1.43

A_4

	$\mathcal{L}$	$\mathcal{E}_x$	$\mathcal{E}_y$	e_c
$\mathcal{L}$	-0.27	7.08	4.63	2.68
$\mathcal{E}_x$	0.0	-0.2	0.03	-0.01
$\mathcal{E}_y$	0.0	0.1	-0.28	0.0
e_c	-0.0	-0.03	0.15	-0.22

FIGURE 12
Estimated autoregression matrices in KF4.



C_1	$\xi_{1,i}$	$\xi_{2,i}$	$\xi_{1,i}$	$\xi_{4,i}$
$\mathcal{L}$	-1.89	16.04	8.01	3.88
$\mathcal{E}_x$	0.0	-1.59	-0.2	-0.02
$\mathcal{E}_y$	0.0	0.04	-1.83	0.0
e_c	0.0	-0.06	0.23	-1.95

C_2	$\xi_{1,i}$	$\xi_{2,i}$	$\xi_{1,i}$	$\xi_{4,i}$
$\mathcal{L}$	-0.04	0.04	0.01	-0.0
$\mathcal{E}_x$	-0.0	0.83	0.27	0.02
$\mathcal{E}_y$	-0.0	-0.01	1.09	-0.01
e_c	0.0	0.01	-0.3	1.26

C_3	$\xi_{1,i}$	$\xi_{2,i}$	$\xi_{1,i}$	$\xi_{4,i}$
$\mathcal{L}$	0.01	-0.02	-0.0	-0.0
$\mathcal{E}_x$	-0.0	0.0	0.0	-0.0
$\mathcal{E}_y$	0.0	-0.01	-0.23	0.0
e_c	-0.0	0.03	0.1	-0.29

C_4	$\xi_{1,i}$	$\xi_{2,i}$	$\xi_{1,i}$	$\xi_{4,i}$
$\mathcal{L}$	0.03	-0.56	-0.53	-0.2
$\mathcal{E}_x$	-0.0	0.01	0.0	0.0
$\mathcal{E}_y$	-0.0	-0.0	0.02	-0.0
e_c	0.0	-0.0	-0.01	0.02

FIGURE 13
Estimated moving average part in KF4.

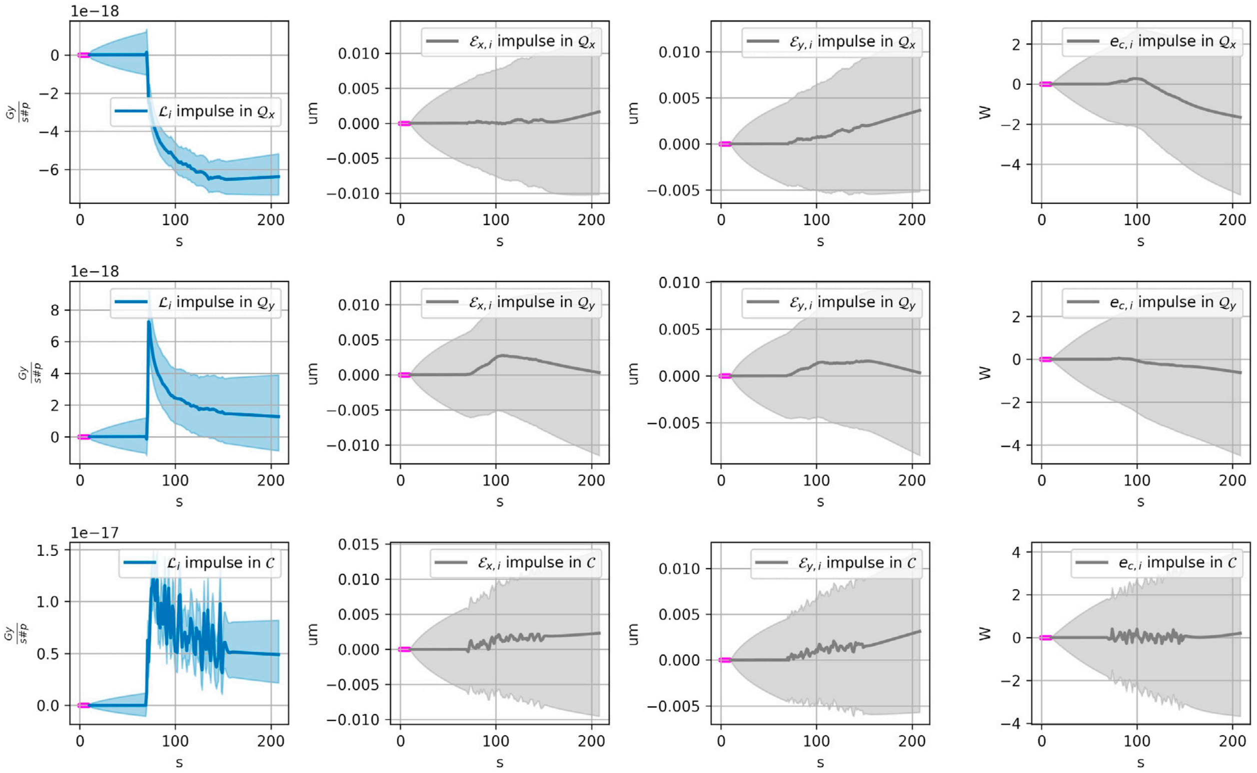


FIGURE 14
Impulse response plots for (KF4) model, impulses in the input parameters: in horizontal and vertical tunes and in the octupole current. The changes in losses and (2σ-)confidence intervals produced by (KF4) are shown in blue colors. The changes in the rest of the output components (horizontal and vertical emittances and electron cloud induced heatloads) and corresponding (2σ-)confidence intervals are shown in gray.

2018, so that 1) for each horizon, the prediction of at least 10 different fills should contribute to computation of R^2 and 2) a number of aggregated prediction was not less than 10^3.

Training on 2017: Figure 8 shows how the R^2 varies for different horizons H for the models trained on the data from 2017, where 8A outlines the quality of forecasts on the training dataset, and 8B shows the quality of forecasts on the testing dataset of 2018. For the models (KF1), (KF1*) with 1D outputs, the results show that they were capable of predicting losses of the other year only on short horizons. Inclusion of additional noncontrolling observations, such as emittances and electron cloud in (KF1*) helps to slightly improve the predictive ability in the 1D output case on the testing dataset with a certain drop in quality on the training dataset. Models with additional output components KF4 and KF4-quad demonstrate significantly improved performance compared to 1D output models. For quite long horizons of the forecast, for both KF4 and KF4-quad R^2 remains high. It is worth noting that the hyperparameters were learned from the dataset of 2017, whereas in 2018 several fills had much longer in time “PRERAMP” intervals than in 2017 (see Figure 1). Nevertheless, the propagated KF4 model kept on predicting well on longer horizons. This suggests that overall the model to some extent captures the global trend and its dependence on the input. The bump in R^2 for the higher horizons in Figure 8B probably could be explained by, first of all, too few fills participating in the estimate, and secondly, most of the changes of input parameters occur on the time horizons smaller than 200 s. The model KF4-quad demonstrates slightly better performance than KF4 on longer horizons. From Figure 8A the additional regularization helped to reduce the overfitting on the training dataset and improve the R^2 on the testing dataset.

Predictions for selected fills 6,672, 6,674, 6,677, and 6,681 of the testing dataset of the year 2018 are shown in Figure 9 for the model (KF4). The values of input parameters in the training dataset lie in the interval (−1, 1). In Figure 9 one can see that the values of some of the input parameters for the testing dataset of 2018 which were standardized to the scale of the training dataset fall outside of interval (−1, 1). For the fills 6,672, 6,674, and 6,681 the scaled octupole current decreases from almost 2.5 to 1. It is visible that for these fills the model captured the dependence of the loss on the current correctly, even outside of the range of the values given during training.

Training on 2018: For the models estimated from the data of 2018, the results are shown in Figure 10. Remind, that the control parameters and their increments in the data of 2 years have different ranges. The results show that the case of modeling of the loss in 2017 based on the data of the year 2018 is more challenging for the proposed approach. Nevertheless KF4 and KF4-quad show significantly better predictive performance than (KF1), (KF1*). Selection of hyperparameters by optimizing MAE in cross-validation for KF4-quad did not lead to improvements compared to KF4. Predictions for the fills 6,176, 6,050, 6,192, and 6,371 of the year 2017 are shown in Figure 11 for KF4 model that was trained on the data of 2018.

3.2.1 Fitted models

KF4: Hyperparameter selection procedure from the data of 2017 led to the KF4 with the dimension of hidden process equal 16, which corresponds to the lag order 4 in the autoregressive part and MA parts of the VARMAX model. The selected input parameters increments history length was 80.

After checking that observability condition and condition on the initial value distribution (see remark in Section 2.2.1) for the estimated KF4 were satisfied, we could transform KF4 in the form of (9) to (8) to obtain the coefficients of equivalent VARMAX model formulation (3). Matrices with autoregressive coefficients are shown in Figure 12. One can see that for the losses $\mathcal{L}$, all output variables, including emittances and electron cloud induced heatloads, participate in AR terms. The opposite is not true, in the trained model, all the rest of output values have small coefficients corresponding to the lagged loss variables. Moving average coefficients in Figure 13 show that the first lag shocks have the most impact on the losses. The loss component of the output shares the shock of “0” lag with the other output variables. Further, instead of presenting coefficients for input variables and 80-lagged increments of input variables, we consider the impulse response function of the model.

Impulse response functions: To analyze how the model (KF4) responds to shocks in one of the input variables, it is convenient to compute an Impulse Response Function (IRF) [20, 27]. Figure 14 demonstrates IRF for (KF4) trained on the data of 2017: the plot shows the change in output parameters after we modify the input parameters within the ranges in the training dataset. All the values of control and output values were set to median values based on the data from 2017, such that after standardizing them, their values equal zero. The impulse for each of the variables was taken as .5 of the range of the observations in 2017 at 50 s in Figure 14: for the horizontal tune $\mathcal{Q}_x$ the impulse was .005 from the level .272, for the vertical tune $\mathcal{Q}_y$ the impulse was .004 from .294, for octupole current the increase was 3.26A from 39.11A. After the value of the control variable increases, the model (KF4) continues to propagate until the outputs stabilize at a certain level. For emittances and electron cloud, IRFs demonstrate that the increase in the octupole current and tunes brings a more steady growth or decrease in values.

4 Conclusion

In this work, we proposed a VARMAX model to predict the evolution of the beam losses in the “PRERAMP” mode of the LHC on horizons up to 5 min based on control and context variables. Given the relationship with state-space models, the model is estimated under an equivalent Kalman Filter form. We considered a VARMAX models on the 1D loss time series and a VARMAX model with a vectorial output composed of the loss, the horizontal and vertical emittances and the aggregated heatload due to the electrom cloud as components, which induce the learning of a hidden state representation that helps predicting the evolution of the losses on a longer horizon. In addition, we proposed an extension of the linear KF for the transition matrix and exogeneous coefficients dependent on the input variables.

The hyperparameter selection procedure on lags needed in the exogeneous and in the autoregressive terms revealed that the control variables have lagged effects on the losses with lags of up to 80 s while a shorter history of 4 s is needed for the autoregressive term, to obtain good predictions on a horizon of 5 min. The loss model with additional output components fitted on the data from 2017 performed well in predicting the loss measured at IR7 in 2018 for a horizon of up to 5 min.

The inclusion of additional output variables in the model, such as heatload and emittance, helped significantly to improve the long-term prediction of the loss. Finally, in terms of interpretability of the model, the proposed impulse response analysis of the estimated model can help investigate different scenarios of the changes in the control parameters to understand their effect on the loss.

A possible extension of this work could be to model jointly both beam losses, which might account for beam coupling. Further, the development of a tool that could guide the operators in the control room could be anticipated, which would propose optimal changes in the available parameters space for a given set of initial setting (i.e. emittances, intensities, *etc.*) while commissioning and re-optimizing the collider at every physics fill. Besides that, the new data obtained from the operation should be useful to re-train and revise the model. Our model based on machine data is a valuable addition to numerical models of particle losses, that can boost and improve the understanding of particle losses and help in the design of future colliders.

Data availability statement

The datasets analyzed in this study are publicly available. This data can be found here: https://doi.org/10.5281/zenodo.7305102.

Author contributions

EK, GO, MS, LC, and TP contributed to the conception and design of the approach. EK performed the statistical analysis and wrote the first draft of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding

This work is funded by the Swiss Data Science Center project grant C18-07. This work was performed under the auspices and with the support from the Swiss Accelerator Research and Technology (CHART) program. Open access funding provided by ETH Zurich.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The reviewer EF declared a shared affiliation with the author MS to the handling editor at the time of review.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphy.2022.960963/full#supplementary-material

References

1. Coyle LTD. *Machine learning applications for hadron colliders: LHC lifetime optimization*. Switzerland: Grenoble INP, France and EPFL (2018).

2. Evans L. The large hadron collider. *New J Phys* (2007) 9:335. doi:10.1088/1367-2630/9/9/335

3. Brüning O, Burkhardt H, Myers S. The large hadron collider. *Prog Part Nucl Phys* (2012) 67:705–34. doi:10.1016/j.ppnp.2012.03.001

4. Arpaia P, Azzopardi G, Blanc F, Bregliozzi G, Buffat X, Coyle L, et al. Machine learning for beam dynamics studies at the cern large hadron collider. *Nucl Instr Methods Phys Res Section A: Acc Spectrometers, Detectors Associated Equipment* (2021) 985:164652. doi:10.1016/j.nima.2020.164652

5. Li S, Zacharias M, Snuverink J, Coello de Portugal J, Perez-Cruz F, Reggiani D, et al. A novel approach for classification and forecasting of time series in particle accelerators. *Information* (2021) 12:121. doi:10.3390/info12030121

6. Koser D, Waites L, Winklehner D, Frey M, Adelmann A, Conrad J. Input beam matching and beam dynamics design optimizations of the IsoDAR RFQ using statistical and machine learning techniques. *Front Phys* (2022) 10:302. doi:10.3389/fphy.2022.875889

7. Schenk M, Coyle L, Pieloni T, Obozinski G, Giovannozzi M, Mereghetti A, et al. Modeling particle stability plots for accelerator optimization using adaptive sampling *JACoW IPAC* (2021), 1923–1926. doi:10.18429/JACoW-IPAC2021-TUPAB216

8. Edelen A, Neveu N, Frey M, Huber Y, Mayes C, Adelmann A. Machine learning for orders of magnitude speedup in multiobjective optimization of particle accelerator systems. *Phys Rev Accel Beams* (2020) 23:044601. doi:10.1103/physrevaccelbeams.23.044601

9. Coyle L, Blanc F, Pieloni T, Schenk M, Buffat X, Camillocci M, et al. Detection and classification of collective beam behaviour in the LHC. *JACoW IPAC* (2021), 4318–21. doi:10.18429/JACoW-IPAC2021-THPAB260

10. Abada A, Abbrescia M, AbdusSalam SS, Abdyukhanov I, Abelleira Fernandez J, Abramov A, et al. FCC-hh: The hadron collider. *Phys J Spec Top* (2019) 228:755. doi:10.1140/epjst/e2019-900087-0

11. Abada A, Abbrescia M, AbdusSalam SS, Abdyukhanov I, Abelleira Fernandez J, Abramov A, et al. HE-LHC: The high-energy large hadron collider. *Phys J Spec Top* (2019) 228:1109. doi:10.1140/epjst/e2019-900088-6

12. Hermes PD, Redaelli S, Jowett J, Bruce R. Betatron cleaning for heavy ion beams with IR7 dispersion suppressor collimators. In: 6th International Particle Accelerator Conference; Richmond, VA (2015). No. CERN-ACC-2015-0193.

13. Zimmermann F. *Electron cloud effects in the LHC*. Geneva: Proceedings ECLOUD (2002). doi:10.5170/CERN-2002-001.47

14. Edwards DA, Syphers MJ. *An introduction to the physics of high energy accelerators*. John Wiley & Sons (2008).

15. Zimmermann F. A simulation study of electron-cloud instability and beam-induced multipacting in the LHC. LHC Project Report 95, CERN (1997), 18.

16. Aquilina N, Giovannozzi M, Lamont M, Sammut N, Steinhagen R, Todesco E, et al. Tune variations in the large hadron collider. *Nucl Instr Methods Phys Res Section A: Acc Spectrometers, Detectors Associated Equipment* (2015) 778:6–13. doi:10.1016/j.nima.2014.12.081

17. Krymova E, Obozinski G, Schenk M, Coyle LT, Pieloni T. *Data-driven modeling of beam loss in the lhc* (2022). doi:10.5281/zenodo.7305102

18. Fokianos K, Rahbek A, Tjøstheim D. Poisson autoregression. *J Am Stat Assoc* (2009) 104:1430–9. doi:10.1198/jasa.2009.tm08270

19. Solfaroli Camillocci M, Lamont M, Juchno M, Schaumann M, Wenninger J, Todesco E. Feed-forward corrections for tune and chromaticity injection decay during 2015 lhc operation. In: Proceedings of IPAC2016; Busan, Korea (2016), 1489–1492.

20. Hamilton JD. *Time series analysis*. Princeton, NJ: Princeton university press (2020).

21. de Jong P, Penzer J. The ARMA model in state space form. *Stat Probab Lett* (2004) 70:119–25. doi:10.1016/j.spl.2004.08.006

22. Casals J, Sotoca S, Jerez M. A fast and stable method to compute the likelihood of time invariant state-space models. *Econ Lett* (1999) 65:329–37. doi:10.1016/s0165-1765(99)00165-2

23. Casals J, García-Hiernaux A, Jerez M. From general state-space to VARMAX models. *Mathematics Comput Simulation* (2012) 82:924–36. doi:10.1016/j.matcom.2012.01.001

24. Abernethy J, Bach F, Evgeniou T, Vert JP. A new approach to collaborative filtering: Operator estimation with spectral regularization. *J Machine Learn Res* (2009) 10: 803–826.

25. Hou K, Zhou Z, So AMC, Luo ZQ. On the linear convergence of the proximal gradient method for trace norm regularization. *Adv Neural Inf Process Syst* (2013) 26: 1–9.

26. Ohtani K. Bootstrapping r2 and adjusted r2 in regression analysis. *Econ Model* (2000) 17:473–83. doi:10.1016/s0264-9993(99)00034-6

27. Belomestny D, Krymova E, Polbin A. Bayesian tvp-varx models with time invariant long-run multipliers. *Econ Model* (2021) 101:105531. doi:10.1016/j.econmod.2021.105531

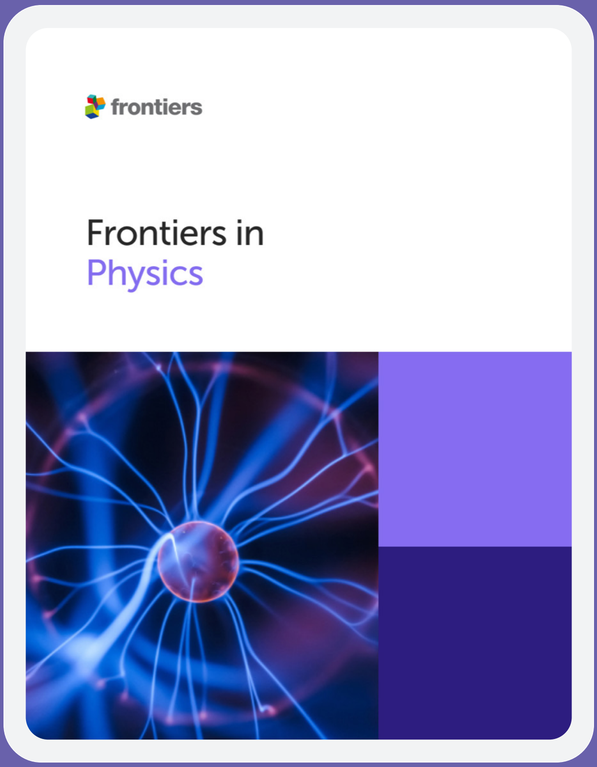
Frontiers in Physics

Investigates complex questions in physics to understand the nature of the physical world

Addresses the biggest questions in physics, from macro to micro, and from theoretical to experimental and applied physics.

Discover the latest Research Topics

See more →



frontiers

Frontiers in Physics

Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact