Adaptive autoencoder latent space tuning for more robust machine learning beyond the training set for six-dimensional phase space diagnostics of a time-varying ultrafast electron-diffraction compact accelerator

Alexander Scheinker,[1,*] Frederick Cropp,[2,3] and Daniele Filippetto[2]
[1]*Applied Electrodynamics Group, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[2]*Lawrence Berkeley National Laboratory, One Cyclotron Road, Berkeley, California 94720, USA*
[3]*Department of Physics and Astronomy, University of California Los Angeles, Los Angeles, California 90095, USA*

(Received 9 December 2022; accepted 27 March 2023; published 19 April 2023)

We present a general adaptive latent space tuning approach for improving the robustness of machine learning tools with respect to time variation and distribution shift. We demonstrate our approach by developing an encoder-decoder convolutional neural network-based virtual 6D phase space diagnostic of charged particle beams in the HiRES ultrafast electron diffraction (UED) compact particle accelerator with uncertainty quantification. Our method utilizes model-independent adaptive feedback to tune a low-dimensional 2D latent space representation of ∼1 million dimensional objects which are the 15 unique 2D projections (x, y),...,(z, p_z) of the 6D phase space (x, y, z, p_x, p_y, p_z) of the charged particle beams. We demonstrate our method with numerical studies of short electron bunches utilizing experimentally measured UED input beam distributions.

DOI: 10.1103/PhysRevE.107.045302

I. INTRODUCTION

Machine learning (ML) tools such as deep neural networks are incredibly useful for a wide range of physics applications such as providing high-accuracy phase space diagnostics of particle accelerator beams [1], for exact representations of many-body interactions [2], for accelerating lattice quantum Monte Carlo simulations [3], for reconstructing quantum dynamics from physical observations [4], for 3D reconstructions of the electron density of crystals for coherent diffraction imaging [5], for quantum feedback [6], and even for determining the structures of unknown networks with time delays [7].

ML for nonstationary systems is an open problem and an active field of research. Recent studies include the use of recurrent neural networks for speech perception in nonstationary noise [8], neural networks for modeling time-varying audio processors [9], and complex-valued neural networks for ML on nonstationary physical data [10,11]. Many approaches for nonstationary systems rely on detecting significant changes after which the weights of neural networks are updated or retrained with new information or are continuously trained to keep up with continuous changes [12–14]. A powerful class of approaches has been developed for the case of covariate shift, where the input distribution $P(\mathbf{x})$ is different for training and test data, but the conditional distribution of output values $P(y|\mathbf{x})$ remains unchanged [15], based on importance-weighting (IW) techniques [16]. IW methods have also been developed using kernel mean matching methods [17] and by minimizing the Kullback-Leibler divergence between a test data density distribution and its estimate [18,19]. Methods have also been developed for extracting frequencies and amplitudes from time-series data [20], and Bayesian methods are being developed for periodic time-varying systems [21].

In this work we present a general adaptive latent space tuning approach, which does not rely on retraining, for increasing the robustness of encoder-decoder convolutional neural networks (CNNs) in the face of time-varying input distributions as well as time-varying systems for which the the conditional distribution of output values $P(y|\mathbf{x}, t)$ changes with time. Our method utilizes adaptive model-independent feedback directly in the learned low-dimensional latent space embedding of an encoder-decoder CNN as shown in Fig. 1 in which a subset $\hat{Y}_i$ of the predicted data $\hat{\mathbf{Y}}$ is compared to an online measurement Y_i to guide adaptive tuning of the low-dimensional latent space representation $\mathbf{y}_L$. We demonstrate that in 2 dimensions a latent representation ($\mathbf{y}_L \in \mathbb{R}^2$) can be adaptively tuned in an unsupervised approach to reconstruct million-dimensional sets of high-resolution images for unknown time-varying inputs $\mathbf{X}(t)$ beyond the span of the training data. Our approach has the potential to benefit a wide range of ML-based tools for complex time-varying systems for which retraining is not feasible.

II. TIME-VARYING SYSTEMS AND DISTRIBUTION SHIFT

In this work we consider two forms of time variation which are important for complex systems. The first is the most common input distribution shift, in which $P(\mathbf{x})$ is time varying, but for which the conditional distribution of output values remains unchanged. The second is a change of the system itself in which the the the conditional distribution of output values $P(y|\mathbf{x}, t)$ also changes with time. In practice we are interested in repeatable systems of the form which are reinitialized at

*ascheink@lanl.gov

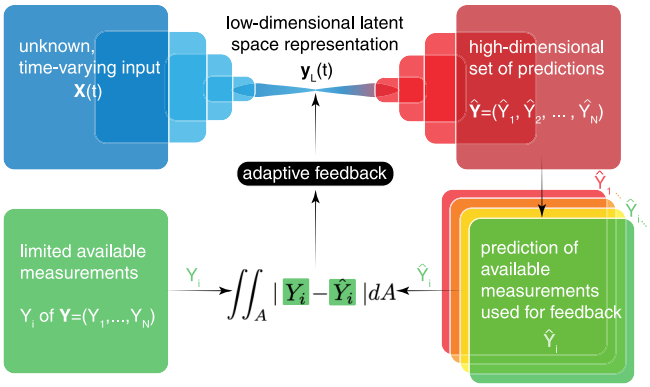

FIG. 1. Overview of the adaptive latent space tuning setup. A subset $\hat{Y}_i$ of the predicted data $\hat{\mathbf{Y}}$ is compared to an available measurement Y_i, and the difference is used to guide adaptive tuning of the low-dimensional latent space representation $\mathbf{y}_L$.

starting times t_i and evolve for a fixed amount of time $T > 0$ over the time interval $t \in [t_i, t_i + T]$.

Such systems can be described mathematically by nonlinear time-varying dynamics of the form

$$\frac{d\mathbf{X}}{dt} = \mathbf{F}(\mathbf{X}(t), t). \tag{1}$$

At any given time t if we allow the system to evolve over a fixed interval of time $[t, t + T]$ we get

$$\mathbf{X}(t + T) = \mathbf{X}(t) + \int_t^{t+T} \mathbf{F}(\mathbf{X}(\tau), \tau)\, d\tau, \tag{2}$$

and we assume that we can measure some function of this final state which depends on the time-varying initial condition $\mathbf{X}(t)$ which we denote by $\mathbf{Y}(t)$ as

$$\mathbf{Y}(t) \equiv \mathbf{G}[\mathbf{X}(t + T)], \tag{3}$$

for some output measurement function $\mathbf{G}(\mathbf{X})$.

Many controlled complex processes can be described by dynamics of the form (1) and (2) as illustrated by the following examples. Chemical reactions typically take place over a fixed time interval $[t_i, t_i + T]$ with time-varying initial conditions such as concentrations and time-varying dynamics such as environmental temperature changes. Charged particle beams in accelerators are generated repeatedly at times t_i, over a wide range of rates, from 1 Hz up to 1 MHz, with each initial beam distribution slightly different and with time variation of radio frequency and magnet components as the beam is accelerated over the length of the accelerator for some time $T > 0$. National power grid loads are diurnal and seasonal, and power grid component performance varies with time due to weather and damage. Finally, as a simple concrete example we consider the scalar dynamic system

$$\dot{x} = -x(x-3)^2 + f(t)x^3, \quad t \in [t_i, t_i + T], \tag{4}$$

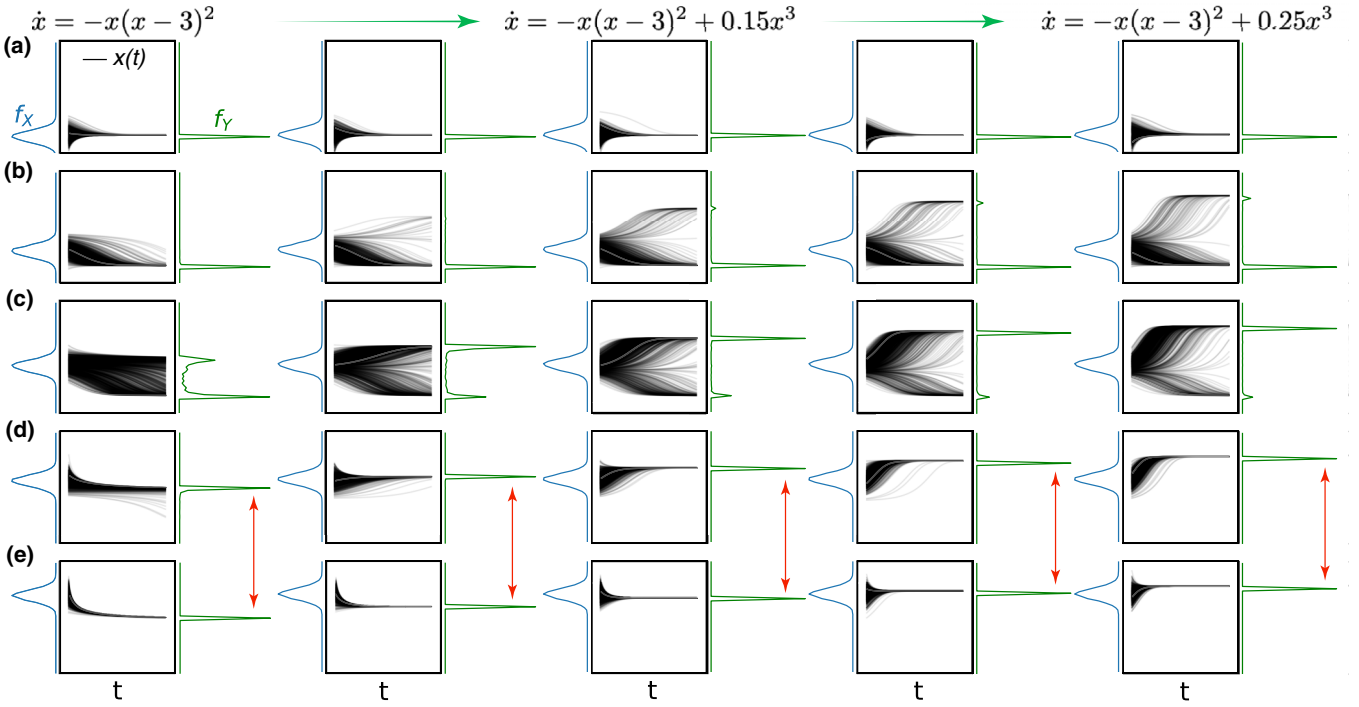

FIG. 2. Example of a system that is periodically restarted and evolves for a fixed time $T > 0$ while the initial conditions and system dynamics change with time. The trajectories of 1000 initial conditions sampled from a time-varying probability density function f_X are shown for $t \in [t_i, t_i + T]$ as well as the corresponding output distributions f_Y. Each row (a)–(e) fixes an initial condition distribution and shows the system's dynamics for various values of $f(t)$ for the system $\dot{x} = -x(x-3)^2 + f(t)x^3$ as $f(t)$ increases from 0 to 0.25. Each column shows the system trajectories for a fixed value of $f(t)$ for various sets of initial condition distributions. The red arrows show that in some cases different initial conditions result in identical final condition distributions, and so the input to output mapping is no longer one to one and unique reconstruction may be impossible.

where $f(t)$ is a slowly changing parameter relative to the dynamics $\dot{x} = dx/dt$. In Fig. 2 we show the evolution of $x(t)$ for various sets of initial conditions sampled from some probability density functions f_X as well as the distributions of final conditions $\mathbf{Y} = \mathbf{X}(t_i + 1)$. Moving from left to right in any row (a–e) of Fig. 2 the dynamics $x(t)$ are shown for a fixed distribution of 1000 initial conditions and various values of $f(t)$ as it is slowly increased from 0 to 0.25 thereby changing the equilibrium points of the system. Moving from top to bottom in any column of Fig. 2 the value of $f(t)$ is fixed, and the dynamics $x(t)$ are shown for 1000 initial conditions from various distributions.

System (4) illustrates several important properties and difficulties of time-varying systems. As seen in the top row (a) of Fig. 2, there is a set of initial conditions which are not sensitive to the time variation of the system dynamics because they remain within the region of attraction of the equilibrium at $x = 0$. For this situation, if the set of initial conditions is not broad enough, an ML model will never have a chance to explore the wider system dynamics and will fail for a large distribution shift.

As we move down the rows and the initial condition distribution's mean value increases we see that the output distribution begins to change as the system's dynamics evolve with time according to $f(t)$ (b, c). This is a rich set of initial conditions for which a wide range of the system dynamics can be more accurately explored.

In the bottom two rows (d, e) we again see a case in which the set of initial conditions is too limited, and therefore even though the final output distributions do change as a function of time with evolving $f(t)$, the equilibrium point at $x = 0$ is never explored. An additional difficulty in (d, e) is the fact that the output distributions for different input distributions begin to look almost identical as $f(t)$ increases (red arrows) because of the large region of attraction that is evolving with $f(t)$. This results in a $\mathbf{X} \to \mathbf{Y}$ map which is not one to one.

The general problem of time-varying distribution shift is not solvable if the changes in initial condition $\mathbf{X}(t)$ or dynamics $\mathbf{F}(\mathbf{X}, t)$ are arbitrarily large or fast or if the measurement function $\mathbf{G}(\mathbf{X})$ is not one to one, in which case it may be impossible to accurately predict $P(\mathbf{X}, \mathbf{Y}, t)$ based on a finite set of data. We start with some regularization assumptions which restrict the class of problems being considered.

We consider a system for which many pairs of inputs and outputs are sampled at times $\{t_1, \ldots, t_n\}$ to generate a data set D:

$$\begin{aligned} D &= \{(\mathbf{X}(t_1), \mathbf{G}[\mathbf{X}(t_1 + T)]), \ldots, \} \\ &= \{(\mathbf{X}(t_1), \mathbf{Y}(t_1)), \ldots, (\mathbf{X}(t_n), \mathbf{Y}(t_n))\}. \end{aligned} \tag{5}$$

We assume that the initial conditions of the state $\mathbf{X}(t)$ are bounded within a compact set $\mathcal{X}$, that $\mathbf{F}(\mathbf{X}, t)$ is piecewise continuous in t. For a fixed starting time t_i, we denote two solutions of (1) as $X_1(t_i) \neq X_2(t_i)$ for $t \in [t_i, t_i + T]$ and assume that $\mathbf{F}$ satisfies a Lipschitz condition for some $L > 0$ over $\mathcal{X}$,

$$\|\mathbf{F}(\mathbf{X}_1, t) - \mathbf{F}(\mathbf{X}_2, t)\| \leqslant L\|\mathbf{X}_1 - \mathbf{X}_2\| \quad \forall\, \mathbf{X}_1, \mathbf{X}_2, \tag{6}$$

and that the variation of the system dynamics is bounded so there exists $M > 0$ such that $\forall\, t_1, t_2 \in [t, t + T]$,

$$\|\mathbf{F}(\mathbf{X}_1, t_1) - \mathbf{F}(\mathbf{X}_2, t_2)\| < M, \quad \forall\, \mathbf{X}_1, \mathbf{X}_2 \in \mathcal{X}. \tag{7}$$

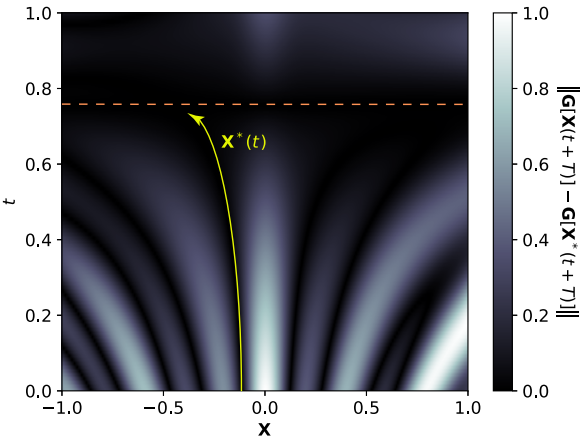

FIG. 3. A synthetic example is shown to illustrate cases where the local unique minimum condition is satisfied and where it fails as the dashed line is approached.

Condition (6) guarantees that each initial condition $\mathbf{X}(t)$ has unique solution $\{\mathbf{X}(\tau),\ \tau \in [t, t + T]\}$. Condition (7) guarantees that any two trajectories that start at the same time, t_i, but with different initial conditions satisfy the following bound for all $t \in [t_i, t_i + T]$ relative to their initial difference

$$\|\mathbf{X}_1(t) - \mathbf{X}_2(t)\| \leqslant \|\mathbf{X}_1(t_i) - \mathbf{X}_2(t_i)\|e^{LT} + (M/L)e^{LT-1}. \tag{8}$$

The bound (8) also implies that all trajectories remain within a compact set [22]. In particular, a bound on the distance between any trajectory $X(t)$ for $t \in [t_i, t_i + T]$ and the compact set $\mathcal{X}$ is given by

$$d_{\mathcal{X}}(\mathbf{X}) \leqslant r(L, M, T) = (M/L)e^{LT-1}. \tag{9}$$

If we take the union over all balls of radius $r(L, M, T)$ centered at all $\mathbf{X} \in \mathcal{X}$,

$$K = \bigcup_{\mathbf{X} \in \mathcal{X}} B(\mathbf{X}, r(L, M, T)), \tag{10}$$

then its closure, $\bar{K}$, is a compact set containing all possible values of $\mathbf{X}(t_i + T)$. If we then assume that $G(\mathbf{X})$ is bounded, we guarantee that any value of $\mathbf{Y}(t_i)$ is contained within the compact set $G(\bar{K})$.

For unique tracking in the case of a time-varying system, considering the mapping

$$\mathbf{X}(t) \to \mathbf{Y}(t) = \mathbf{G}[\mathbf{X}(t + T)], \tag{11}$$

we assume that at any time $t \in [t, t + T]$ the function $\mathbf{G}$ has nonzero derivative so that for any particular $\mathbf{X}^*(t)$, $\mathbf{Y}^*(t)$ there is some $\epsilon > 0$ such that

$$\mathbf{Y}(t) = \mathbf{Y}^*(t) \Longleftrightarrow \mathbf{X}(t) = \mathbf{X}^*(t),\ \forall \|\mathbf{Y}(t) - \mathbf{Y}^*(t)\| < \epsilon. \tag{12}$$

Figure 3 shows a synthetic example which satisfies the above property for a portion of time before it fails at the dashed orange line where the derivative of $\mathbf{G}$ is zero.

III. ADAPTIVE LATENT SPACE TUNING

In our approach we assume that we are able to perform initial training based on measurements of both system inputs

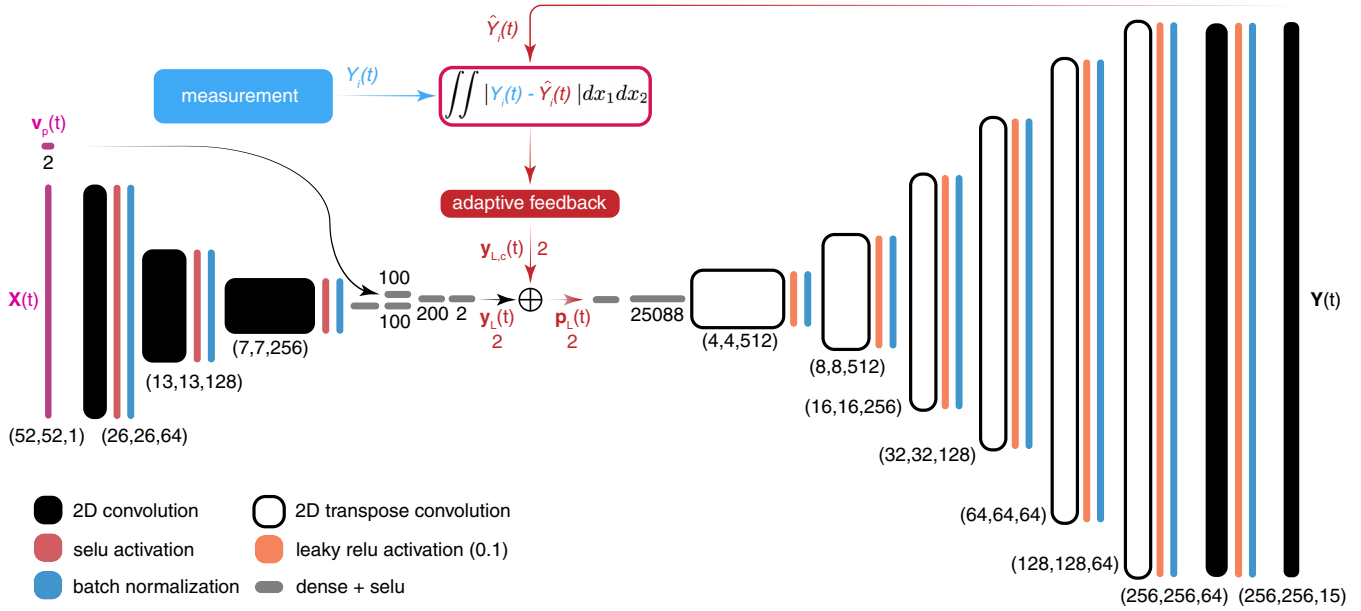

FIG. 4. A CNN-based adaptive latent space tuning setup in which a measurement of a single projection Y_i is compared to the distribution's prediction $\hat{Y}_i$ to guide adaptive feedback in the latent space to track the other projections which are not measured. The relu and selu activation functions are $\text{relu}(x) = \max\{0, x\}$, $\text{selu}(x) = \{0.1x \text{ for } x < 0 \text{ and } x \text{ for } x > 0\}$.

$\mathbf{X}(t)$ and outputs $\mathbf{Y}(t)$, but that afterwards we must rely only on limited diagnostics with loss of input data and only partial output data availability, so that retraining is impossible. Our approach is to adaptively tune an ML model in real time to keep up with the time variation of the system and of its initial conditions based only on limited output measurements. In what follows, we consider an encoder-decoder CNN architecture whose inputs are $\mathbf{X}(t)$ which is an image and $\mathbf{v}_p(t)$ which is a vector of parameters. The output $\mathbf{Y}(t)$ a stack of images, as shown in Fig. 4. Training was carried out using the Adam optimizer built into Tensorflow to minimize a mean-squared error loss with the standard learning rate of 10^{-3} and batch sizes of 32 on a GV100 GPU. To maintain the physical relationship between all of the images used as inputs and generated as outputs of the encoder-decoder network we performed a simple normalization in which each image type was replaced by $I/\max\{D_I\}$, where the maximum is calculated over each sample of that image from the entire data set. For example, the (x, y) projections were each divided by the maximum value of all of the (x, y) projections within the training data set so that each pixel lived within the global bounds [0,1].

We represent an $N\times N$ input image $\mathbf{X}(t)$ as a matrix

$$I^0 = \{I^0(i_0, j_0), i_0, j_0 \in \{1, 2, \ldots, N\}\}, \tag{13}$$

which passes through a stride 2 convolutional layer such that the output image size is reduced by a factor of 4 resulting in a $N/2\times N/2$ image I^1. A collection of $N_f > 1$ filters is used giving an output image I^1 whose (i_1, j_1) pixel has value

$$b^1 + \sum_{n=1}^{N_f} w_n \times f_n \left(b_n^0 + \sum_{i=-1}^{1} \sum_{j=-1}^{1} F_{0,ij,n} \times I^0_{i_0+i, j_0+j} \right). \tag{14}$$

For a deep encoder-decoder CNN the total number of adjustable parameters easily grows to millions, and retraining requires large collections of new data sets. After several layers of convolutions we significantly reduce the size of a relatively large image ($52\times52 \to 7\times7$) using multiple filters at each stage, resulting in a tensor of shape $N_i\times N_i\times N_f$ where $N_i\times N_i$ is the final image size and N_f is the number of filters in the last convolution layer of the encoder. We flatten the image and apply dense fully connected layers which are concatenated with dense layers acting on the vector input $\mathbf{v}_p(t)$ resulting in a final low-dimensional latent space representation, a vector $\mathbf{y}_L$ of length $N_L \ll N_{i,p}$. We then add an N_L-dimensional control input vector $\mathbf{y}_{L,c}$ so that the latent space parameters are given by

$$\begin{aligned} \mathbf{p}_L &= (p_1, \ldots, p_{N_L}) = (y_1 + v_{c,1}, \ldots, y_{N_L} + v_{c,N_L}) \\ &= \mathbf{y}_L + \mathbf{y}_{L,c}, \end{aligned} \tag{15}$$

where the vector $\mathbf{y}_L$ is the output of the trained encoder and $\mathbf{y}_{L,c}$ the controlled parameters used for adaptively tuning the latent space ($\mathbf{y}_{L,c} = \mathbf{0}$ when training). The vector $\mathbf{p}_L$ is passed through additional fully connected dense layers before being reshaped into a small image ($\sim 8\times8$) which then passes through a series of 2D transpose convolution layers until a collection of N_o output images of size $N_{im}\times N_{im}$ are generated in a final layer with N_c channels with size $\hat{\mathbf{Y}}(i, j, d) = N_{im}\times N_{im}\times N_c$. Once the network is trained this collection of output images is a general nonlinear function (the generative branch of the network) of the parameters $\mathbf{p}_L$ of the form

$$\begin{aligned} \hat{\mathbf{Y}}(i, j, d) &= \mathbf{F}(\mathbf{p}_L, \mathbf{w}, \mathbf{b}, \{A\}), \\ & i, j \in \{1, 2, \ldots, N_o\}, \quad d \in \{1, 2, \ldots, N_c\}, \end{aligned} \tag{16}$$

where $\mathbf{w}$ and $\mathbf{b}$ are the weights and biases and $\{A\}$ are the set of activation functions of the generative layers. Our prediction $\hat{\mathbf{Y}}$ is our estimate of some unknown physical quantity $\mathbf{Y}$ which we assume we cannot easily directly measure fully. In order to enable the adaptive feedback part of this procedure we must assume that we have some form of noninvasive online

measurement of $M(\mathbf{Y}(t))$ that can be compared to a simulated measurement of our generated prediction $\hat{M}(\hat{\mathbf{Y}})$, which we know how to approximate. For example, we may be interested in a 6D density distribution, but we have direct measurements of only one or several 2D projections in which case both M and $\hat{M}$ are simply projection operators. A detailed example and simulation study of such a problem for particle accelerator applications is presented in Sec. IV. We set up a dynamic feedback loop for minimization of a cost function of the form

$$C(\mathbf{p}_L(t), t) = \mu[M(\mathbf{Y}(t)), \hat{M}(\hat{\mathbf{Y}}(\mathbf{p}_L(t)))], \tag{17}$$

where μ is a metric quantifying error, such as the L^1 norm of the difference between a pair of 2D projections:

$$\mu[M(\mathbf{Y}), \hat{M}(\hat{\mathbf{Y}})] = \iint |Y_i - \hat{Y}_i| dx_1\, dx_2, \tag{18}$$

with adaptive latent space dynamics

$$\frac{\partial p_i}{\partial t} = \frac{\partial v_{c,i}}{\partial t} = \sqrt{\alpha\omega_i}\cos\left[\omega_i t + kC(\mathbf{p}_L(t), t)\right], \tag{19}$$

which are chosen based on the results in [23–27].

In (19) α represents a dithering amplitude which controls the size of the dynamic perturbations, k a feedback gain, and the product $k\alpha$ is as a learning rate. The dithering frequencies ω_i are chosen relative to a base frequency ω such that they are distinct, of the form $\omega_i = r_i\omega \neq r_j\omega = \omega_j$ for $i \neq j$ such that no two frequencies are integer multiple of each other (such as distinct $r_i \in [1, 1.75]$) because nonlinearity typically introduces harmonics into the system dynamics. The convergence results for this feedback algorithm depend on the parameters being orthogonal in Hilbert space such that for any $t > 0$ and any measurable $f(t) \in L^2[0, t]$, the $L^2[0, t]$ inner products in the limit of large frequency ω are

$$\lim_{\omega\to\infty} \int_0^t \cos(\omega_i\tau) f(\tau) \cos(\omega_j\tau)\, d\tau = 0, \tag{20}$$

$$\lim_{\omega\to\infty} \int_0^t \cos^2(\omega_i\tau) f(\tau)\, d\tau = \frac{1}{2}\int_0^t f(\tau)\, d\tau. \tag{21}$$

In fact, any orthogonal functions can be used, including non-differentiable and discontinuous square waves, as described in more detail in [23,25,27]. The resulting on average dynamics of the evolution of the cost function with the latent space variables evolving under feedback (19) are then given by

$$\begin{aligned}\frac{dC}{dt} &= \frac{\partial C}{\partial t} + (\nabla_{\mathbf{p}_{L,c}} C)^T \frac{\partial \mathbf{p}_L}{\partial t} \\ &= \frac{\partial C}{\partial t} - \frac{k\alpha}{2}(\nabla_{\mathbf{p}_{L,c}} C)^T (\nabla_{\mathbf{p}_{L,c}} C).\end{aligned} \tag{22}$$

For a metric of the form (18) which is convex and positive semidefinite the gradient $\nabla_{\mathbf{p}_{L,c}} C$ satisfies the condition

$$\|\nabla_{\mathbf{p}_{L,c}} C\| > 0, \quad \forall \hat{Y}_i \neq Y_i, \tag{23}$$

$$\|\nabla_{\mathbf{p}_{L,c}} C\| = 0, \quad \text{iff} \quad \hat{Y}_i = Y_i. \tag{24}$$

Therefore, for any desired accuracy $\delta > 0$ there exists some lower bound $\delta_C > 0$ on $\|\nabla_{\mathbf{p}_{L,c}} C\|$ such that

$$\|Y_i(t) - \hat{Y}_i(t)\| > \delta \Longrightarrow \|\nabla_{\mathbf{p}_{L,c}} C\| > \delta_C, \tag{25}$$

which ensures that over the set $\|Y_i(t) - \hat{Y}_i(t)\| > \delta$ we can choose $k\alpha > 0$ sufficiently large to ensure that $dC/dt < 0$ according to

$$\begin{aligned}\frac{dC}{dt} &= \frac{\partial C}{\partial t} - \frac{k\alpha}{2}(\nabla_{\mathbf{p}_{L,c}} C)^T (\nabla_{\mathbf{p}_{L,c}} C) \\ &< \frac{\partial C}{\partial t} - \frac{k\alpha}{2}\delta_C^2 < 0, \quad \forall k\alpha > \frac{2}{\delta_C^2}\frac{\partial C}{\partial t}.\end{aligned} \tag{26}$$

Therefore, over an annulus of any radius $\delta > 0$ surrounding $Y_i(t)$ a positive definite function C can be considered as a Lyapunov function for the overall system dynamics (17) and (19), which is negative definite for sufficiently large $k\alpha > 0$ ensuring the asymptotic convergence of $\hat{Y}_i(t)$ to within a δ ball of $Y_i(t)$:

$$\lim_{t\to\infty} \hat{Y}_i(t) \in B(Y_i(t), \delta). \tag{27}$$

Finally, based on the assumption (12) made in Sec. II, there exists an $\epsilon > 0$ neighborhood of $\mathbf{Y}(t)$ such that the convergence of $\hat{Y}_i(t)$ to $Y_i(t)$ guarantees the convergence of $\hat{\mathbf{Y}}(t)$ to $\mathbf{Y}(t)$. Therefore, under the above conditions, it is possible for adaptively tuned latent space parameters to track a unique global minimum based only on limited measurements when the feedback gain $k\alpha > 0$ is sufficiently large relative to the time variation of the system $\frac{\partial C}{\partial t}$, such that $dC/dt < 0$. Furthermore, by the same arguments, even in the case that the cost function is not convex we may still maintain a globally optimal set of latent space parameters as long as we start close enough to a global minimum and continuously track it with time as demonstrated in Sec. VII below.

Our approach attempts to combine the complementary strengths of ML and model-independent feedback to provide the best of both worlds: an ability to learn directly from large complex data, while maintaining robustness to time variation and distribution shift.

IV. PARTICLE ACCELERATOR APPLICATION

Charged particle dynamics evolve in a six-dimensional phase space (x, y, z, p_x, p_y, p_z) where (x, y, z) are particle positions and (p_x, p_y, p_z) are momentum components. In accelerator physics, p_z is many orders of magnitude larger than p_x and p_y, and we usually consider the density function $\rho(x, y, z, x', y', E)$, where $(x', y') = (p_x/p_z, p_y/p_z)$ are angles of motion relative to the acceleration z axis, and E is total particle energy. All six dimensions are coupled through collective effects such as space charge forces and coherent synchrotron radiation in which accelerating charged particles release light which impacts other particles in the bunch thereby changing their energy [28]. The influence of collective effects grows as accelerators generate shorter more intense bunches such as 30 fs bunches at the SwissFEL x-ray free-electron laser (FEL) [29], sub-100 fs bunches for ultrafast electron diffraction (UED) [30], and picosecond bunch trains for UEDs and multicolor XFELs [31]. Laser and beam-driven plasma wakefield particle accelerators (PWAs) are especially complex and could greatly benefit from real-time advanced 6D phase space diagnostics to aid in control and optimization [32–35].

Typically the initial conditions of charged particle beams, the 6D phase space distributions at the beam source, drift

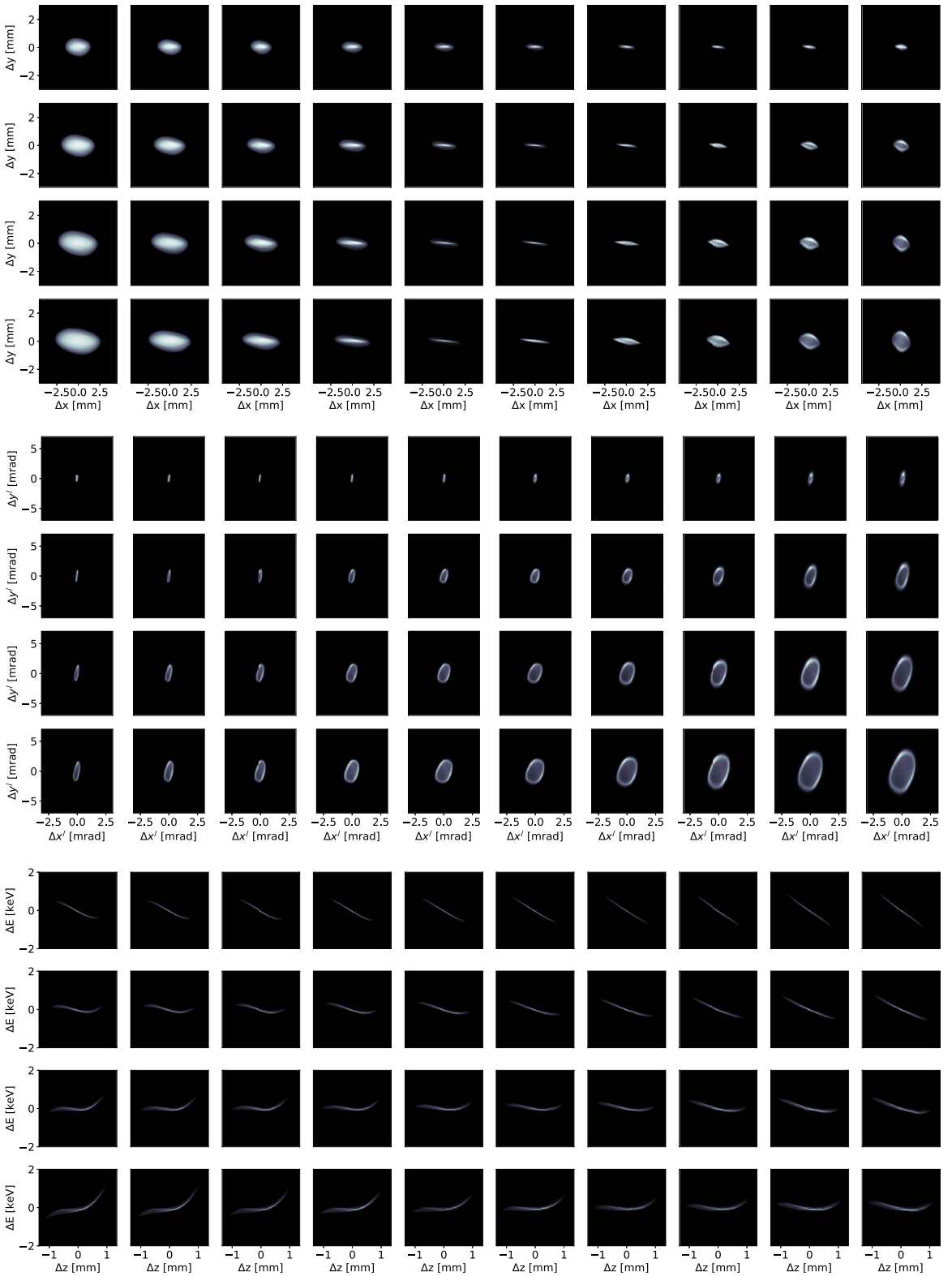

FIG. 5. 2D projections of the first simulated beam with 40 different charge and solenoid settings are shown. For each 2D phase space projection, the bunch charge is increasing for each row from top to bottom, and the solenoid current is increasing for each column from left to right within the range shown in Fig. 7(a).

unpredictably with time and only can be measured destructively during lengthy dedicated studies [36]. Furthermore the beams are accelerated and focused by magnets and resonant electromagnetic field structures whose characteristics are uncertain due to hysteresis and misalignments and drift with time due to disturbances such as temperature drifts and vibrations. In [37] the parameter drift was even shown to be large enough to enable parasitic measurements. Once

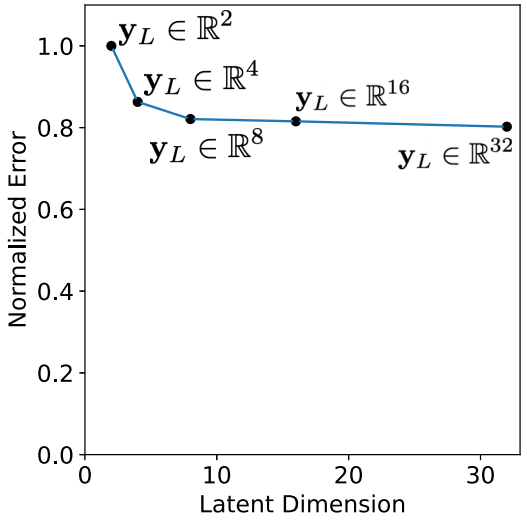

FIG. 6. The mean absolute error for predictions is shown for several latent space dimensions normalized by the error of the lowest dimension choice. For the largest latent space size of $\mathbf{y}_L \in \mathbb{R}^{32}$ the error was approximately 80% of $\mathbf{y}_L \in \mathbb{R}^2$.

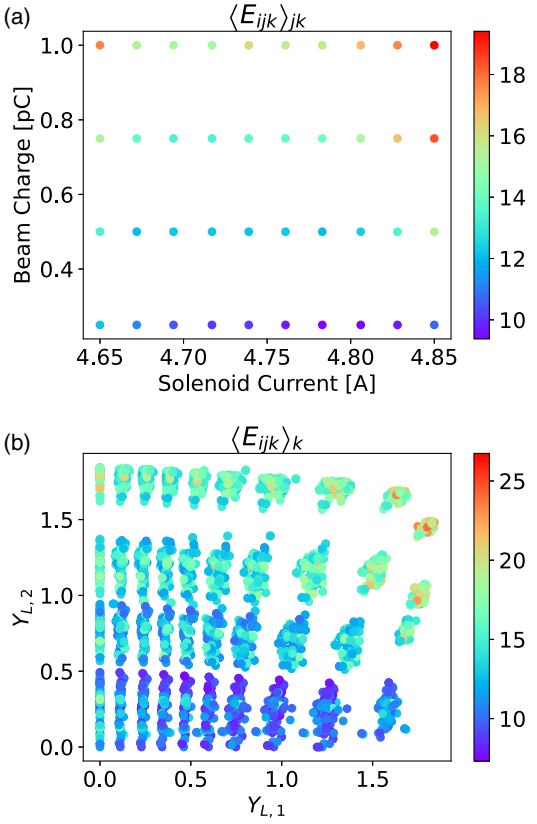

FIG. 7. (a) Averaged percent error as defined in (34) is shown as a function of beam charge and solenoid current. (b) Averaged percent error as defined in (33) is shown relative to position in the latent space. Errors shown for training data.

a machine is running to perform experiments new detailed beam measurements are typically very limited, especially for the initial beam conditions, which can rely on destructive methods such as wire scans or lengthy quadrupole magnet scan-based measurements which interrupt all downstream operations. What is sometimes possible is to record a few 2D projections of a beam's 6D phase space. For example, it is possible to measure the longitudinal phase space (LPS) of a charged particle bunch by using a transverse deflecting radio frequency resonant cavity (TCAV) which measures (z, E), and at many accelerators it is also possible to measure the transverse beam image (x, y) in higher energy sections by using scintillating screens which do not have much impact on the beam dynamics. In the field of particle accelerators, ML methods are becoming popular for the control and diagnostics of charged particle beams [38–40]. Neural networks are being used for uncertainty aware anomaly detection to predict errant beam pulses [41], as virtual diagnostics for 4D tomographic phase space reconstructions [42], for predicting the transverse emittance of space charge-dominated beams [43], for electron ghost imaging [44], and for control of accelerator magnets [45]. At CERN, supervised learning techniques are being applied for the reconstruction of magnet errors in the incredibly large (thousands of magnets) Large Hadron Collider lattice [46], for detecting faulty beam position monitors [47], and for beam dynamics studies [40]. At SLAC, Bayesian methods are being developed for online accelerator tuning [48], for optimization [49], and as surrogate models for beam diagnostics [50,51]. Bayesian methods with safety constraints are being developed at the SwissFEL and the High-Intensity Proton Accelerator at PSI [52]. At the EuXFEL, CNNs have been used to generate incredibly high resolution virtual diagnostics [1]. A laser PWA has also been optimized by utilizing Gaussian processes at the Central Laser Facility [53].

Although ML tools such as deep neural networks can learn complex relationships in large systems directly from data, a major challenge faced by any model-based methods (physics or data-based) is that of time-varying systems or systems with distribution shift, which require extensive retraining and adaptive retuning [54]. At Los Alamos National Laboratory, preliminary adaptive ML methods have been developed, such as the use of neural networks together with extremum seeking (ES) for automatic femtosecond-level control of the time-varying longitudinal phase space distribution of the electron beams in free electron lasers [55], and for mapping downstream measurements to unknown time-varying input beam distributions [56,57]. Relative to the results presented here, the two most closely related published results are those in [42] and [57]. In [42] a very nice approach to generating 4D tomographic transverse phase space reconstructions of the (x, y, p_x, p_y) projections of a beam's 6D phase space are demonstrated utilizing a densely connected neural network unlike our 2D convolutional neural network approach and without adaptive feedback. The work in [57] is a collection of preliminary proof-of-principal results on using a 2D CNN to generate the 2D projections of a 6D latent space, and in the work presented here we significantly generalize those preliminary adaptive ML results, study a much wider range of beam conditions, provide analytic as well as numerical robustness studies for going beyond the training data set, and develop methods for uncertainty quantification.

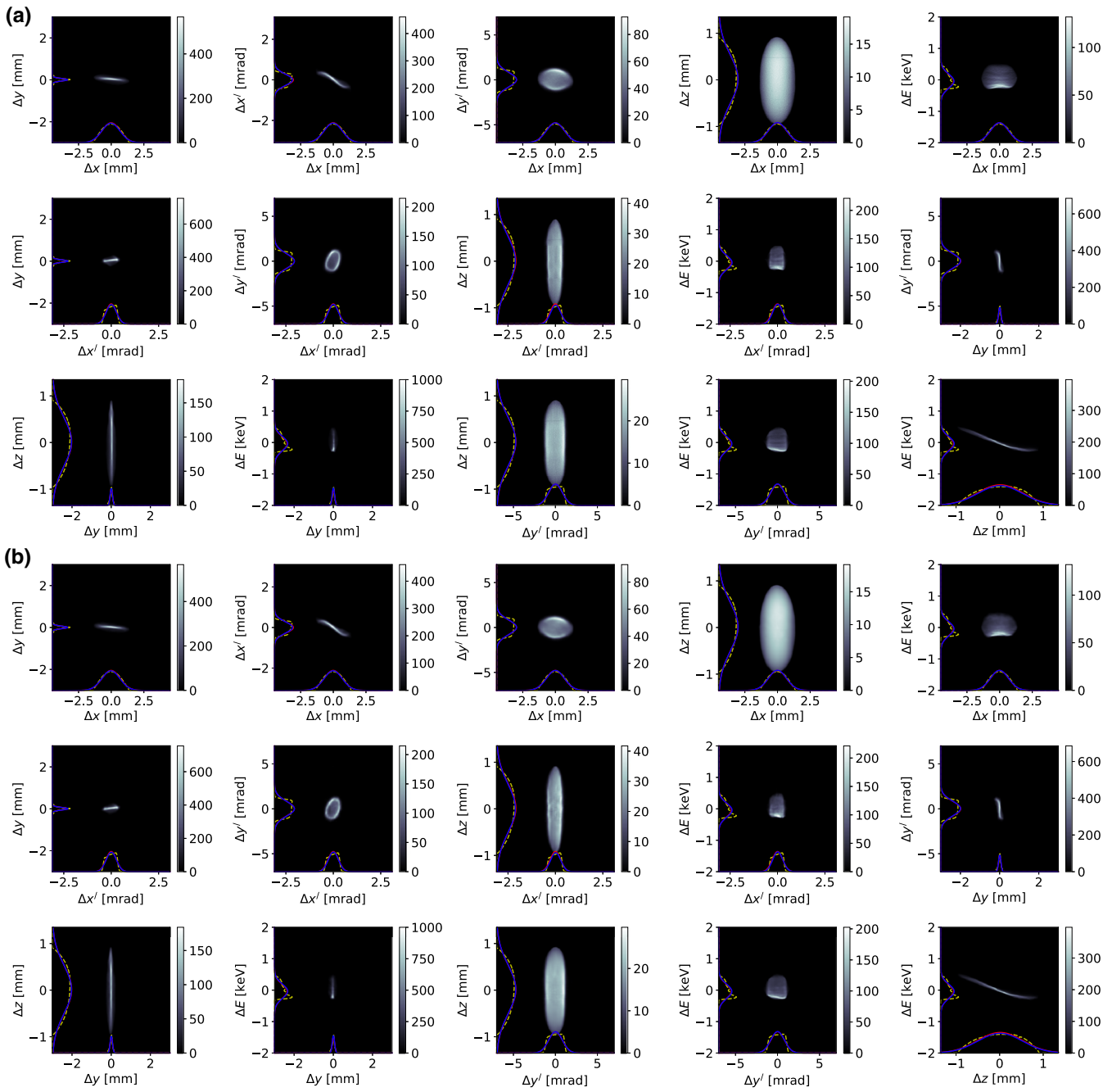

FIG. 8. True and CNN-based phase space predictions compared for a 0.5 pC bunch with a 4.76 A solenoid current. (a) The top 15 images are the true 2D projections of the phase space. (b) The bottom 15 images are the same projections created by the generative half of the CNN from a 2D latent representation. In each image the dashed yellow curves are the various 1D projections of the 2D phase space image shown, while the blue and red curves are overlayed Gaussian fits to the dashed yellow projections of the true and CNN-generated 2D phase space images, respectively, for comparison.

For our accelerator application, we focus on the High Repetition-rate Electron Scattering apparatus (HiRES) at Lawrence Berkeley National Laboratory (LBNL), which accelerates pC-class, subpicosecond-long electron bunches up to one million times a second (MHz), providing some of the most dense 6D phase space among accelerators at unique repetition rates, making it an ideal test bed for advanced algorithm development [58,59].

We design an encoder-decoder style generative network with a 2706-dimensional input consisting of a 52×52 input beam (x, y) distribution image as well as a 2D vector of bunch charge and solenoid current. We utilize five measured input beam distributions together with 95 synthetic input beam distributions generated from random combinations of principal components extracted from the measured data [60,61]. For each of the 100 distributions we run 40 simulations of beam

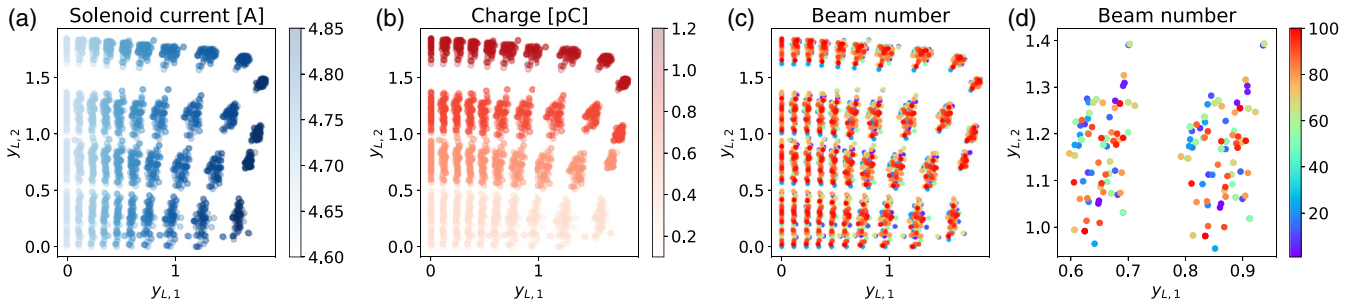

FIG. 9. A view of the 2D latent space embedding locations of 4000 different inputs (100 beam distributions each with ten solenoid strengths and four bunch charges). (a) Latent space locations colored by solenoid strength (T). (b) Latent space locations colored by charge (pC). (c) Latent space locations colored by input beam distribution number. (d) Zoom in on two islands of the locations of 100 input beam distributions each for a 1 pC charge at two different solenoid settings with markers colored by input beam number showing that island to island the network has placed various input beams consistently.

dynamics with 3D space charge for ten solenoid currents ranging from 4.65 to 4.85 A and four bunch charges of 0.25, 0.5, 0.75, and 1 pC. The 40 combinations of generated (x, y), (x', y'), and (z, E) projections for a single input distribution over the entire range of solenoid current and bunch charge are shown in Fig. 5.

The output of our network is a 256×256×15 pixel object which is an image with 15 channels. Each of the 15 channels represents a 2D projection of the 6D phase space: (x, y), (x, z), (x, x'), (x, y'), (x, E), (x', y), (x', z), (x', y'), (x', E), (y, z), (y, y'), (y, E), (y', z), (y', E), and (z, E) downstream from the HiRES injector. By forcing the CNN to simultaneously generate all 15 projections of the 6D phase space we introduced observational biases directly through data embodying the underlying physics, allowing the CNN to learn functions that reflect the physical structure of the data [62]. We demonstrate that the network has learned the correlations in the system and use only measurements of the 2D (x, y) or (z, E) projections, which are typically available online, to predict all other 2D phase space projection distributions, which are not easily measured in accelerators in real time.

By squeezing our 2706-dimensional input space down to a general nonlinear representation in a much smaller latent space (two in this case), we show that we can quickly adaptively tune our system by utilizing the encoder-decoder representation learned by the CNN. This flexibility allows us to quickly respond to unknown disturbances and changes, such as unknown changes of the input images and the input parameters, which create a difference between a function of the CNN's generated predictions and some related measurement. We demonstrate the method with three different studies as described below.

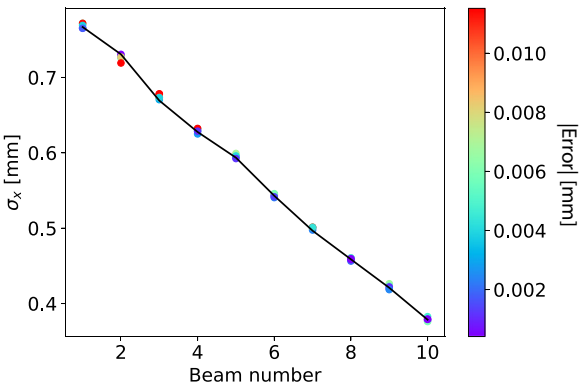

FIG. 10. All five of the estimated values of σ_x based on CNN predictions are shown relative to their true value for ten beams. The markers are colored by error in millimeters relative to the colorbar.

We experimented with higher dimensional latent spaces, doubling the latent dimension several times with a slight improvement in prediction error. The tradeoff is that a higher dimensional latent space gives small accuracy improvements, but slows down the adaptive feedback, which must search over a higher dimensional space. Because the 2D latent space gave accurate predictions we chose to maintain optimal adaptive speed at the cost of a negligible performance drop, as shown in Fig. 6. Another benefit of a low-dimensional latent space is that the encoder-decoder is forced to try and find a compact low-dimensional manifold that captures the underlying structure inherent to the data, whereas a much higher dimensional latent space would not force any such representation and would be much more prone to overfitting by brute-force memorization

A. Physically interpretable latent space embedding

We start by confirming that we can compress our input data down to an incredibly low-dimensional $\mathbf{y}_L \in \mathbb{R}^2$ representation from which we then accurately generate high-quality detailed phase space projection images. Figure 7(a) shows the percent error of our training data averaged over all 15 2D phase space predictions and all 100 input beam distributions as a function of beam charge and solenoid current. Figure 7(b) shows the percent error averaged over all 15 2D phase space predictions for every one of the 100 input beam distributions with the positions of the markers corresponding to their location in the 2D latent space as mapped by the encoder half of the CNN. In Fig. 8 we show all 15 true and reconstructed projections of a single input beam at a single charge and solenoid current value to illustrate the ability of the encoder-decoder CNN to generate high-quality, high-resolution (256×256 pixels) images with 15 such images representing a ∼1 million-dimensional object which is being generated from a 2D latent embedding.

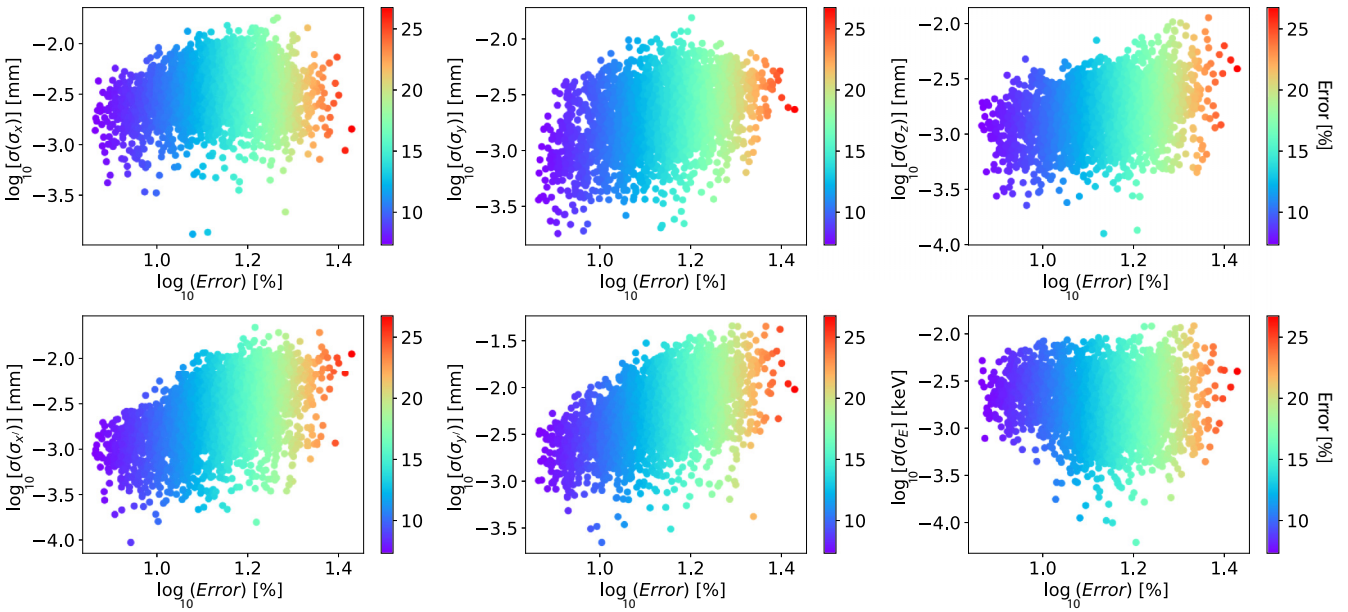

FIG. 11. log - log plots of fit $\sigma(\sigma_\bullet)$, $\bullet \in \{x, y, z, x', y', E\}$ values vs percent error are shown for all 4000 training data sets.

Some interesting features of the learned latent space embedding can be seen in Fig. 9 where the positions of all 4000 input beams (100 beam distributions, each at ten different solenoid strengths and four different bunch charge values) that were used for training are shown. Because the network was trained with regularization which penalized the norm of network weights, we see that a natural compact, continuous, and dense latent embedding has been learned so that one can continuously dynamically move throughout the latent space. It is worth mentioning that a compact and continuous latent embedding can also be forced by more complicated approaches such as variational autoencoders in cases where the input data have no natural continuous relationship; however, in our approach we simply penalize the norm of the latent embedding according to the following addition to the cost function:

$$C_L = w_L \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{y}_{L,i}\|^2, \tag{28}$$

with a small weight of $w_L = 10^{-7}$, which gently nudges the latent embedding towards a smaller and more compact continuous representation, but allows the encoder the freedom of arranging the latent space without any specific assumptions. This is in contrast to variational autoencoders which force the latent space to a very particular normal distribution. It is also worth mentioning that for this problem, if we had made a variational autoencoder version of our encoder-decoder, we could have easily achieved a continuous latent space as we

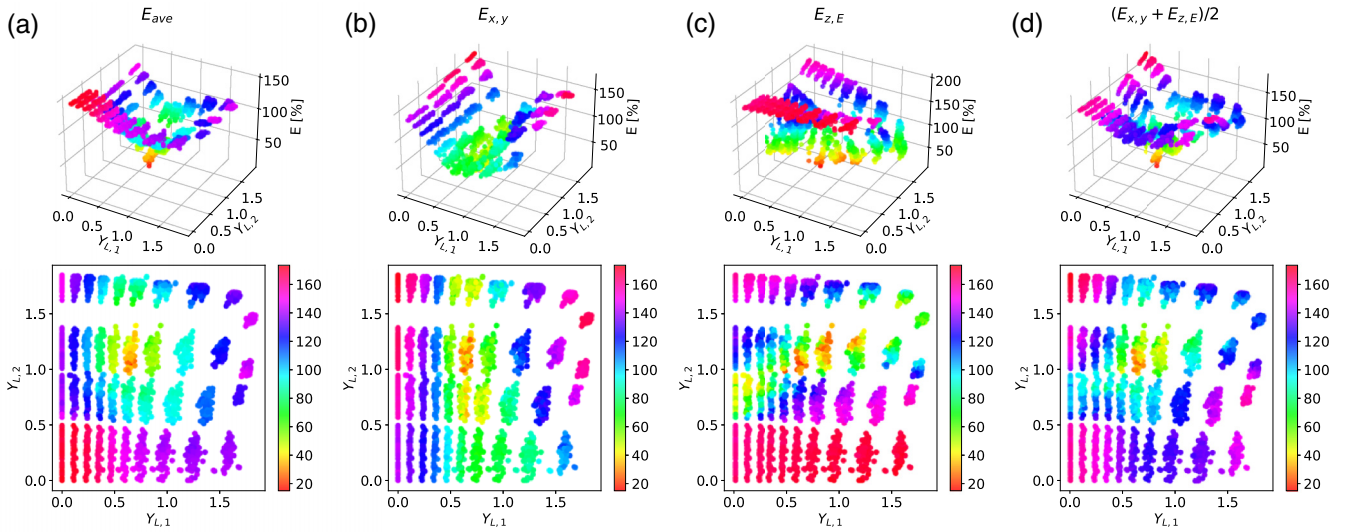

FIG. 12. Errors relative to various metrics.

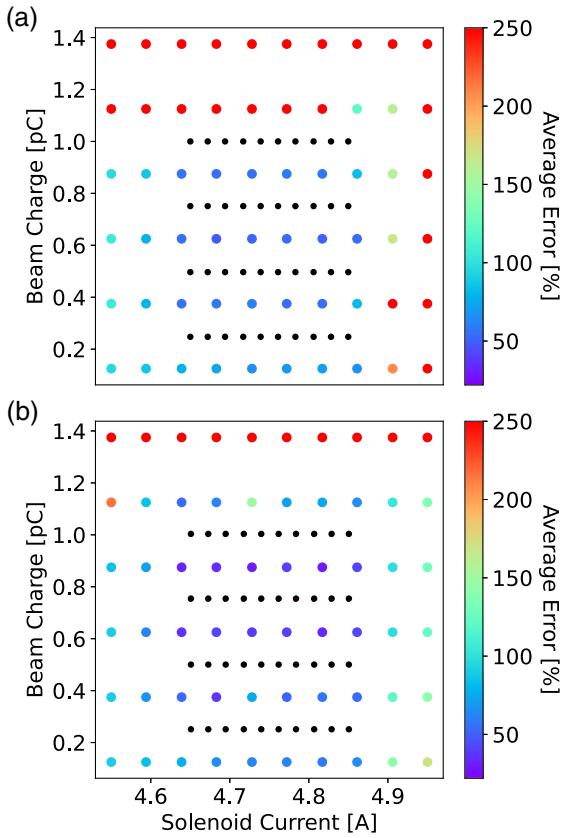

FIG. 13. Values of beam charge and solenoid current used to generate the training data are shown in black. CNN prediction errors are shown for a new grid distinct from and beyond the span of the training data (a). All prediction errors are improved by utilizing the adaptive latent space tuning approach (b).

have here, but we would have destroyed all physical interpretability by forcing all of the latent space dimensions to normal distributions centered at 0.

It is also apparent that the encoder-decoder has learned some of the underlying physics of the problem and has naturally clustered and sorted beams by solenoid current, charge, and input beam distribution. In Fig. 9(a) the markers are colored by solenoid current and can be seen to form ten columns with each column corresponding to one of the ten solenoid current values. In Fig. 9(b) the markers are colored by bunch charge and can be seen to form four rows, corresponding to the four bunch charge values. In Figs. 9(c) and 9(d) the markers are colored by input beam distribution number, and zooming in on two islands in 9(d) shows that the locations of individual input beams for a given solenoid current and bunch charge are consistent from island to island.

B. Uncertainty quantification

The 6D phase space dynamics of charged particle beams are coupled and evolve under physics constraints unique to a given accelerator lattice, as described by the the relativistic Vlasov equation

$$\frac{\partial \rho}{\partial t} + \mathbf{v}\cdot\nabla_{\mathbf{x}}\rho + \frac{\partial \mathbf{p}}{\partial t}\cdot\nabla_{\mathbf{p}}\rho = 0, \quad \frac{\partial \mathbf{p}}{\partial t} = q(\mathbf{E} + \mathbf{v}\times\mathbf{B}),$$

$$\mathbf{p} = \gamma m\mathbf{v}, \quad \gamma = 1/\sqrt{1 - \frac{v^2}{c^2}}, \quad v = |\mathbf{v}|,$$

$$\mathbf{v} = (\dot{x}, \dot{y}, \dot{z}), \quad \mathbf{p} = (p_x, p_y, p_z),$$

where the electric and magnetic fields include contributions from charges within the beam as well the external electromagnetic fields of accelerator components such as radio frequency resonant accelerating cavities, solenoids, dipoles, and quadrupole magnets. Therefore, it may be possible to uniquely recover or track various 2D slices of the 6D phase space based on a single or several 2D measurements.

Furthermore, all 15 unique 2D distributions are projections of the same 6D density function; for example,

$$\rho(x,y) = \int_z\int_{x'}\int_{y'}\int_E \rho(x,y,z,x',y',E)\,dz\,dx'\,dy'\,dE,$$

$$\rho(x',y') = \int_x\int_y\int_z\int_E \rho(x,y,z,x',y',E)\,dx\,dy\,dz\,dE,$$

$$\rho(z,E) = \int_x\int_{x'}\int_y\int_{y'} \rho(x,y,z,x',y',E)\,dx\,dx'\,dy\,dy'.$$

Considering all 15 projections, each component of the phase space shows up in five different images. Therefore, if we project any of the five images containing x onto the x axis, the generated distributions should be identical. This provides a natural method for uncertainty quantification as we can fit Gaussian distributions to each projection of every one of the phase space dimensions:

$$\rho(x,y),\ \rho(x,z),\ \rho(x,x'),\ \rho(x,y'),\ \rho(x,E) \longrightarrow \sigma_x,$$
$$\rho(x,y),\ \rho(y,z),\ \rho(x',y),\ \rho(y,y'),\ \rho(y,E) \longrightarrow \sigma_y,$$
$$\rho(x,z),\ \rho(y,z),\ \rho(x',z),\ \rho(y',z),\ \rho(z,E) \longrightarrow \sigma_z,$$
$$\rho(x,x'),\ \rho(y,x'),\ \rho(z,x'),\ \rho(y',x'),\ \rho(x',E) \longrightarrow \sigma_{x'},$$
$$\rho(x,y'),\ \rho(y,y'),\ \rho(z,y'),\ \rho(y',x'),\ \rho(y',E) \longrightarrow \sigma_{y'},$$
$$\rho(x,E),\ \rho(y,E),\ \rho(z,E),\ \rho(x',E),\ \rho(y',E) \longrightarrow \sigma_E.$$

In Fig. 10 we show all five versions of the predicted σ_x fit for ten different training data samples. Despite all predictions being very accurate some variation is seen.

Considering the five different fits of each dimension's width we can then consider the standard deviation of any projection's predictions. If we denote by $\{\sigma_{xi}\}$, $i = 1,\ldots,5$ the five different versions of σ_x, we can define

$$\mu(\sigma_x) = \frac{1}{5}\sum_{i=1}^{5}\sigma_{xi}, \quad \sigma(\sigma_x) = \sqrt{\frac{1}{5}\sum_{i=1}^{5}[\sigma_{xi} - \mu(\sigma_x)]^2}. \tag{29}$$

The value of $\sigma(\sigma_x)$ is then a natural unsupervised way to check whether the generated distributions are physically consistent. To check this relationship we plot the standard deviations as defined above for all six variables for each of the 4000 samples in the training data set versus the percent prediction error averaged over all 15 projections, as shown in Fig. 11. Because the training data have been learning very accurately

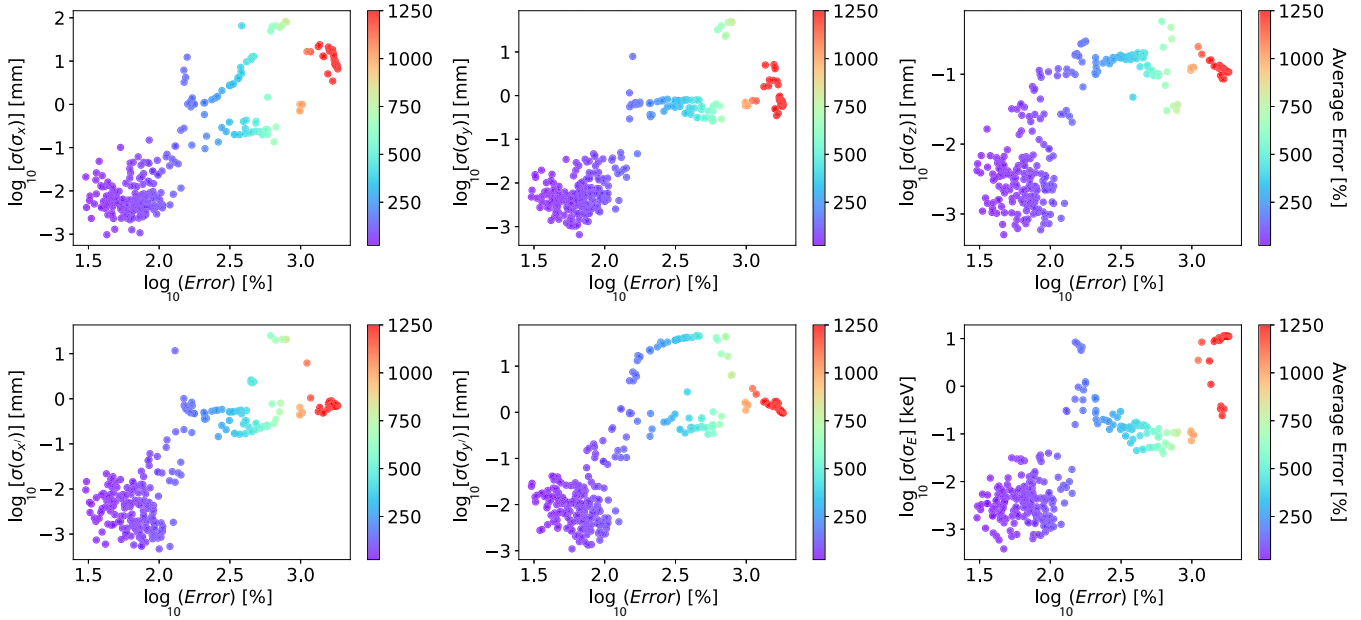

FIG. 14. Error vs σ of the 6D phase space variables is shown with a clear correlation between large prediction errors and the standard deviation of the predicted Gaussian fits based on five different projections for each phase space dimension.

the errors are very small (note the log scales), but nevertheless a correlation is seen. This relationship will be much more pronounced in the following sections when new setups with large prediction errors are studied.

V. IMPROVED ROBUSTNESS BEYOND THE TRAINING SET

For notational convenience, in what follows we will sometimes rewrite our 6D phase space variables as

$$(x, y, z, x', y', E) = (x_1, x_2, x_3, x_4, x_5, x_6), \tag{30}$$

and denote the 100 input beam distributions as $\rho_i^{in}(x, y)$, $i \in \{1, \ldots, 100\}$; for each one we consider 40 different combinations of the two-parameter bunch charge [pC] and solenoid current [A] denoted as $\mathbf{p}_j = (p_{j1}, p_{j2})$, $j \in \{1, \ldots, 40\}$. For each such (i, j) input beam–parameter combination we denote the 15 unique 2D phase space projections of the beam's 6D phase space as

$$\hat{\rho}_{kl}^{ij}(x_k, x_l), \quad k, l \in \{1, \ldots, 5\}, \quad k \neq l, \tag{31}$$

which are estimates of the true 2D projections ρ_{kl}^{ij}. For each 2D phase space projection (k, l) of input beam distribution i and parameter setting j we quantify the percent absolute phase space prediction error as

$$E_{ijkl} = 100 \times \frac{\sum_{x_k} \sum_{x_l} \left| \hat{\rho}_{kl}^{ij}(x_k, x_l) - \rho_{kl}^{ij}(x_k, x_l) \right|}{\sum_{x_k} \sum_{x_l} \rho_{kl}^{ij}(x_k, x_l)}. \tag{32}$$

We also average over all 15 projections to get an overall average phase space prediction error for a beam

$$\langle E \rangle_{ij} = \frac{1}{15} \sum_{k \neq l} E_{ijkl}. \tag{33}$$

Finally we also calculate the error averaged over all 100 of the different input beams, for each charge and solenoid parameter setting j, according to

$$\langle E \rangle_j = \frac{1}{100} \sum_i \left[\frac{1}{15} \sum_k \sum_l E_{ijkl} \right]. \tag{34}$$

In Fig. 7(a) we show the value of $\langle E \rangle_j$ as defined in Eq. (34) at each of the 40 values of bunch charge and solenoid settings. In Fig. 7(b) we show the value of $\langle E \rangle_{ij}$ for all 4000 input beam, bunch charge, and solenoid setting combinations as well as their locations in the 2D latent space embedding.

In Fig. 12(a) we plot the average difference E_{ave} as defined in (33) for all 4000 beams in the training data relative to one beam located at the center of the latent space, which we see is a relatively convex function. To test the feasibility of this limited projection-based latent space tuning approach in Fig. 12(b) for each beam we plot the (x, y) difference $E_{ij12} = E_{ijxy}$ as defined in (32), and we see that although very close to the center the function is convex, there is a long valley along the second latent dimension which is the beam charge because in this case the final (x, y) distribution is more sensitive to solenoid current than bunch charge. This implies that adaptive feedback based on (x, y) projections alone may be relatively slow depending on the initial condition. In Fig. 12(c) for each beam we plot the (z, E) difference $E_{ij36} = E_{ijzE}$ as defined in (32) and again see a relatively convex function with a diagonal valley which again implies that convergence might be slow depending on the initial condition. Finally in Fig. 12(d) for each beam we plot the sum of the errors $(E_{ij12} + E_{ij36})/2 = (E_{ijxy} + E_{ijzE})/2$ and see that as expected this combination gives a better more convex overall cost function than either projection alone, implying that adaptive tuning relative to this metric will have the best results.

We test the ability of adaptive latent space feedback to improve the robustness of the encoder-decoder CNN by utilizing

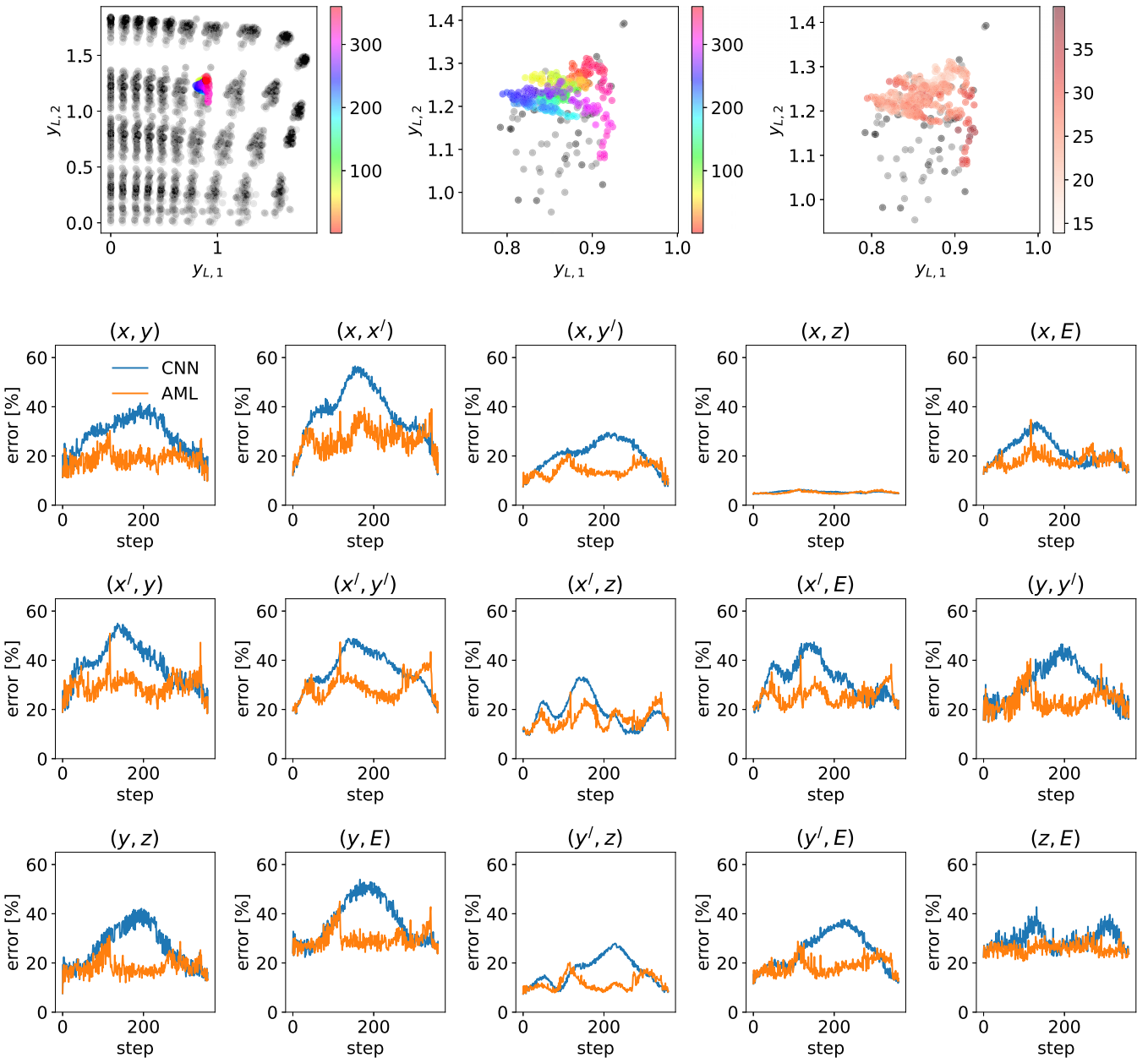

FIG. 15. The top part shows the location within the 2D latent space to which the CNN maps the input beam distribution as it is rotated with the colorbar showing rotation angle (0° to 360°) for the first two images and the colorbar showing the average error for the third image. The bottom part shows the prediction percent error for each of the phase space projections with and without adaptive ML-based tuning.

the cost function

$$C(t) = \iint |\rho_{z,E}(t) - \hat{\rho}_{z,E}(t)| dE\, dz + \iint \left|\rho_{x,y}(t) - \hat{\rho}_{x,y}(t)\right| dx\, dy \tag{35}$$

such that the CNN's LPS prediction $\hat{\rho}_{z,E}$ and transverse beam profile prediction $\hat{\rho}_{x,y}$ are compared to their measurements as provided by a TCAV and a scintillating screen. No other projections of the beam's phase space are assumed to be available for measurement.

As described above, the ES-based perturbation of the latent space parameters takes place according to the ES dynamics:

$$\frac{dp_{L1}}{dt} = \sqrt{\alpha\omega}\cos\left[\omega t + kC(\mathbf{p}_L, t)\right],$$
$$\frac{dp_{L2}}{dt} = \sqrt{\alpha\omega}\sin\left[\omega t + kC(\mathbf{p}_L, t)\right], \tag{36}$$

which tracks the time-varying minimum of the analytically unknown cost function $C(t)$ as defined in (35).

To demonstrate the robustness of this approach beyond the span of the training data set, we utilize an input beam distribution that was measured six months later than all of the data that has been used for the neural network training, and for that distribution we compare the average difference E_{ave}

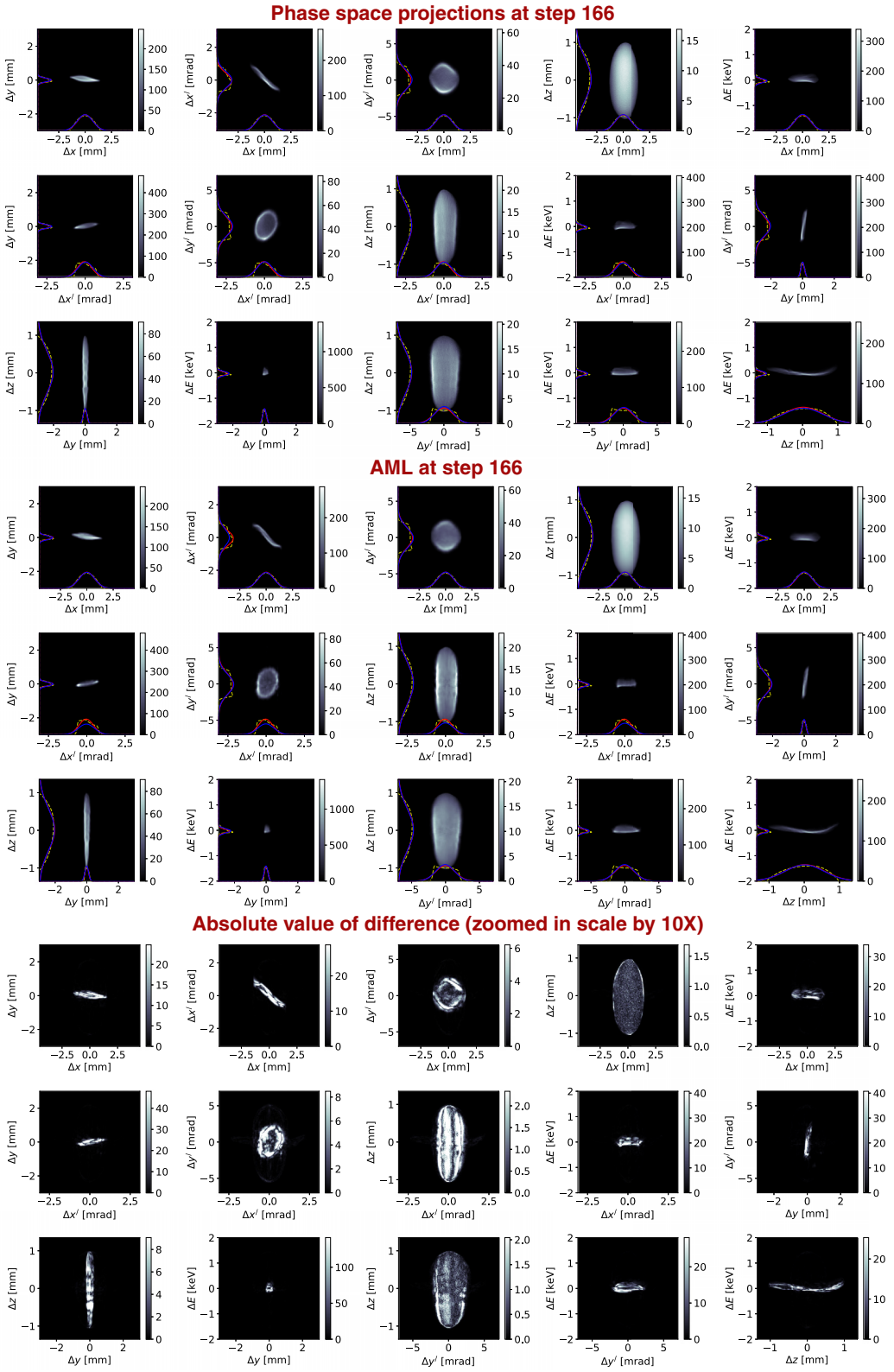

FIG. 16. AML predictions and errors are shown for step 166 of tracking with a time-varying input beam distribution. This is the point where the CNN-based method has the largest error as shown in Fig. 15. The absolute value of the difference between true and predicted projections is shown on a zoomed in color scale which is 10× that of the projection images showing that the errors are mostly very fine textural details. In each image the dashed yellow curves are the various 1D projections of the 2D phase space image shown, while the blue and red curves are overlayed Gaussian fits to the dashed yellow projections of the true and CNN-generated 2D phase space images, respectively, for comparison.

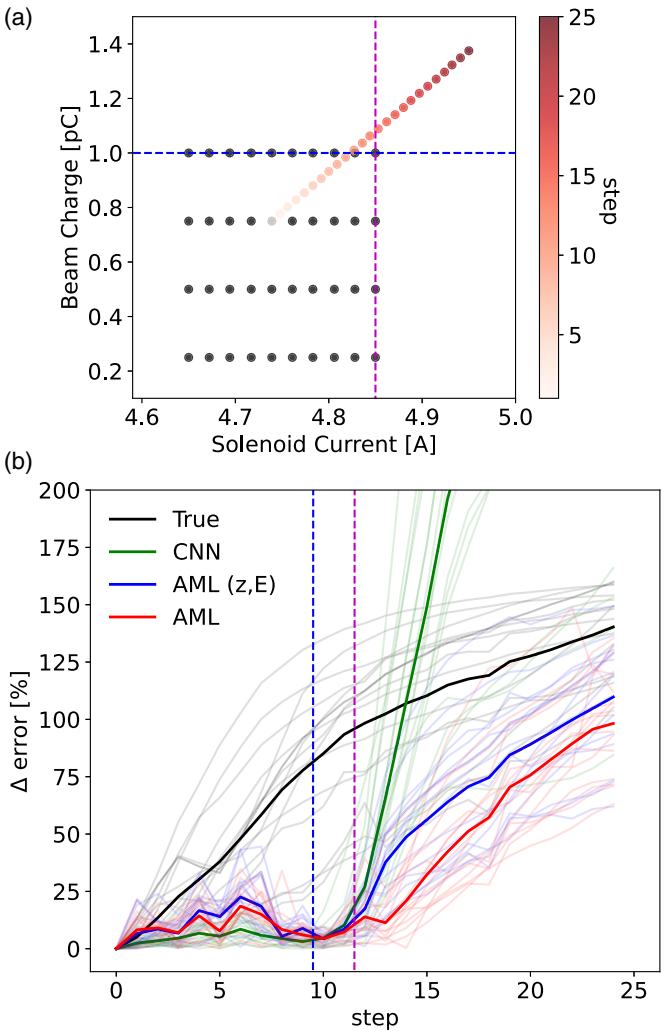

FIG. 17. (a) View of the input parameter values used as we leave the span of the training data. (b) The difference between the initial distribution and new distributions along the path shown in the top part are shown in black. In green we see the CNN's predictions, which catastrophically fail beyond the training set. The blue curve shows the accuracy achieved by latent space tuning based only on (z, E), and the red curve shows the same results when based on (z, E) and (x, y) simultaneously.

as defined in (33) over a grid of bunch charge and solenoid current values that is distinct from and beyond the range of the training data, which is shown in black. Figure 13(a) shows the average error E_{ave} of the trained CNN, and Fig. 13(b) shows the average error achieved after 100 steps of ES-based tuning were taken in the latent space according to (36). The adaptive latent space tuning approach decreases errors for higher solenoid current and higher beam charge beyond the span of the training data. When the distance from the training data is made to be very large, for bunch charge of 1.4 pC, the adaptive method's performance also drops off.

In Fig. 14 the average error vs σ of the 6D phase space variables is shown with a clear correlation between large prediction errors and the standard deviation of the predicted Gaussian fits based on five different projections for each phase space dimension. The correlations are shown of all of the predicted beams including both the predictions that were adaptively tuned via the latent space method and those generated by the CNN without any adaptive tuning.

VI. TRACKING WITH UNKNOWN TIME-VARYING INPUT BEAMS

Next, we demonstrate that this method can be used to make the CNN-based diagnostic much more robust to unknown time variations. We vary the input beam distribution by rotating it over 360 steps while keeping the CNN's input distribution fixed at its initial condition. This simulates a case in which we may have initially performed an invasive and slow input beam distribution measurement, but then the beam begins to change and we have no way of measuring that during operations as it would require intercepting the low-energy beam. In the top part of Fig. 15 we show where the input beam's representation would be mapped to within the latent space if the time-varying input beam was available for measurement. In the bottom part of Fig. 15 we show percent error for each of the 15 2D phase space projections with and without adaptive latent space tuning.

The adaptive latent space tuning method is able to accurately track all 15 of the 2D phase space projections based only on feedback which uses the downstream (x, y) and (z, E) beam measurements. In Fig. 16 we show the adaptively tuned network's prediction of all 15 phase space projections relative to their true values at step 166 of the tracking process, which is when a network which is not tuned has the maximum prediction error. Note that the error plots in Fig. 16 are purposely exaggerated with colorbar scales zoomed in on by a factor of 10 to emphasize that the small differences between true and predicted values differ mostly in terms of minor fine details and textures.

VII. TRACKING WITH UNKNOWN TIME-VARYING INPUT BEAMS AND UNKNOWN ACCELERATOR AND BEAM PARAMETERS BEYOND THE TRAINING SET

The final and most challenging demonstration of the robustness of this AML method utilizes a combination of time-varying input distributions and beam and accelerator parameters beyond the span of the training set. For this test we measured an additional input beam distribution at the HiRES injector six months after the initial training data was collected. We then generated a series of input beams in which we performed linear interpolation from one input beam distribution $\rho_0(x, y)$ which was seen during training to the new unseen distribution $\rho_u(x, y)$ over 25 steps $(n = 1, \ldots, 25)$:

$$\rho(x, y, n) = \rho_0(x, y)\frac{25 - n}{24} + \rho_u(x, y)\frac{n - 1}{24}. \tag{37}$$

During this interpolation we also chose a new unseen bunch charge Q_u and solenoid strength S_u far outside of the span of the training data and interpolated their values as well, starting

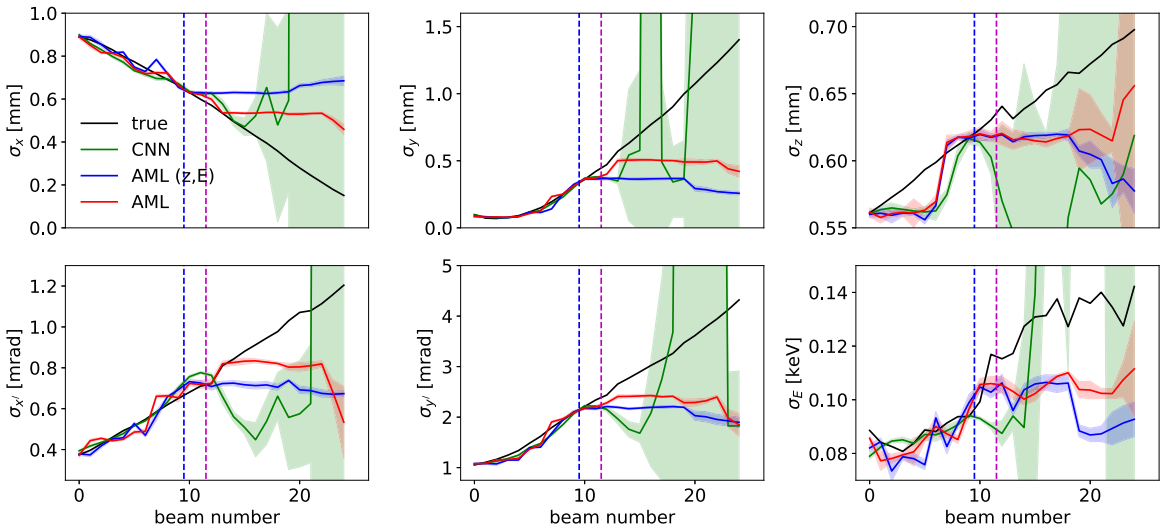

FIG. 18. Mean values surrounded by envelopes of $\pm\sigma$ for the predicted values of σ_x, σ_y, σ_z, $\sigma_{x/}$, $\sigma_{y/}$, and σ_E are shown relative to their true values (black). It is clearly seen that the predictions based on the CNN alone without feedback quickly bifurcate and diverge beyond the range of the plot, while the feedback-based networks remain physically consistent much further beyond the training set. As expected, using both (x, y) and (z, E) measurements has advantages in terms of reconstruction accuracy.

with initial values Q_0, S_0 within the span of the training data:

$$Q(n) = Q_0\frac{25-n}{24} + Q_u\frac{n-1}{24}, \tag{38}$$

$$S(n) = S_0\frac{25-n}{24} + S_u\frac{n-1}{24}, \tag{39}$$

as shown in Fig. 17(a). Figure 17(b) shows the growing difference between the beam's downstream phase space projections and their initial values, with the thin lines showing errors of each of the 15 individual phase space projections, and the solid line showing the average error. The green curves show the predictions of the trained CNN with an assumed known knowledge of the input beam distribution and accelerator parameters. The trained CNN at first performs very well as the input data are still not very far from the training set but quickly catastrophically fails as soon as the edge of the training data is passed.

We also compare two latent space tuning approaches, one that utilizes only the (z, E) measurement for feedback as shown in blue, and another that utilizes both (z, E) and (x, y) measurements for feedback resulting in more accurate tracking beyond the training set as shown in red. One important feature to notice is that beyond the training set the CNN alone has a catastrophic failure resulting in predictions which are further from the truth than what would have been achieved by just assuming that the initial state was always the correct one.

In Fig. 18 we show the mean values of the predicted pσ_x, σ_y, σ_z, $\sigma_{x/}$, $\sigma_{y/}$, and σ_E relative to their true values (black), surrounded by envelopes of $\pm\sigma$. The vertical dashed lines correspond to the same color lines in Fig. 17 so that it is clear when and how far beyond the training data predictions are being made. It is clear that the adaptive ML methods both remain physically consistent and give much more accurate predictions much further beyond the span of the training data than the ML method alone. Furthermore, in both the adaptive and nonadaptive case, the σ envelope provides a useful unsupervised measure of uncertainty.

Initially, while still comfortably within the span of the training data, both the AML- and the CNN-based predictions are very accurate and almost identical. We show the AML predictions and their differences from the true values in Fig. 19. In Fig. 20 we show the true, AML-, and CNN-based predictions for step 16 where it is clear that the CNN-based predictions have catastrophically failed while the adaptive method's predictions are still physically consistent approximations of the true distributions. Finally, in Fig. 21 we have gone so far beyond the training data that even the adaptive method has failed, which was predicted by the extremely wide σ band of the unsupervised prediction as shown at step 25 of the $\sigma_{x/}$, σ_z and the σ_E predictions in Fig. 18.

VIII. SUMMARY OF RESULTS

In Sec. II we carried out a general analytic treatment for a general class of dynamic systems, proving under what conditions our adaptive feedback method can be guaranteed to track time-varying systems and to provide unique results. This in itself is a general mathematical result which can be useful for many time-varying systems and how to couple adaptive feedback with virtual diagnostics or other types of models.

In Sec. III we showed how the general adaptive approach can be applied by using encoder-decoder CNNs, which are powerful ML tools for working directly with high-dimensional images. Again, this is a general result that can be applied to any type of image-based data that describes the evolution of dynamic systems.

In Sec. IV we further specialized the approach to the special case of a particle accelerator application in which the measured and generated images are the 2D projections of a charged particle beam's 6D phase space. This is an important application, especially for the particle accelerator community, for which having such a diagnostic would enable finer control over beam properties.

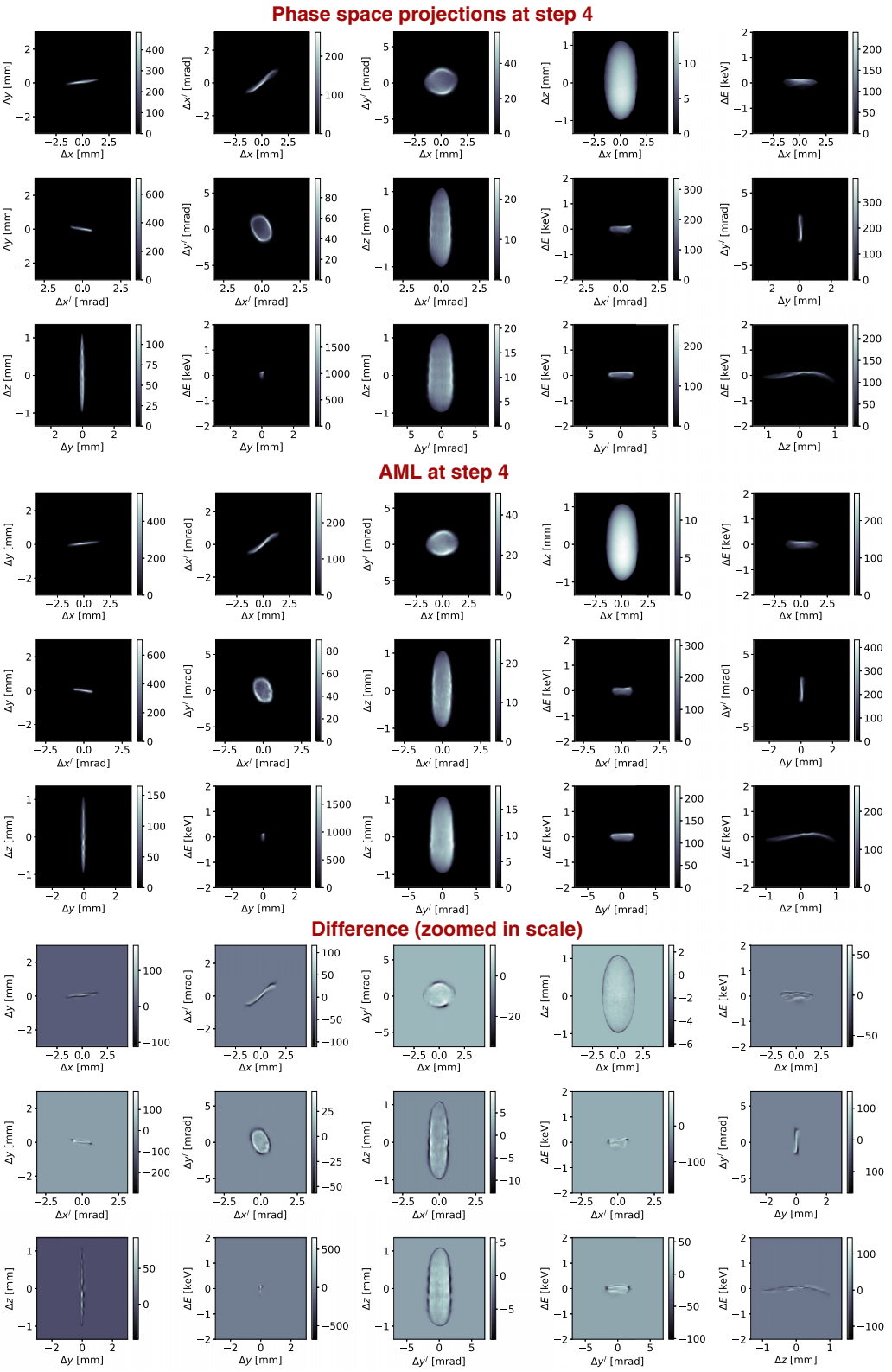

FIG. 19. AML predictions for step 4. At this point both the AML- and the CNN-based predictions are very accurate and look almost identical, so we show just the AML predictions and the difference from the true values.

In Sec. IV A we have demonstrated that our encoder-decoder CNN naturally learned a physically interpretable latent space representation of the high-dimensional data.

In Sec. IV B we demonstrate how this method can be utilized to generate a physics-based uncertainty quantification (UQ) of the predicted phase space distributions. This

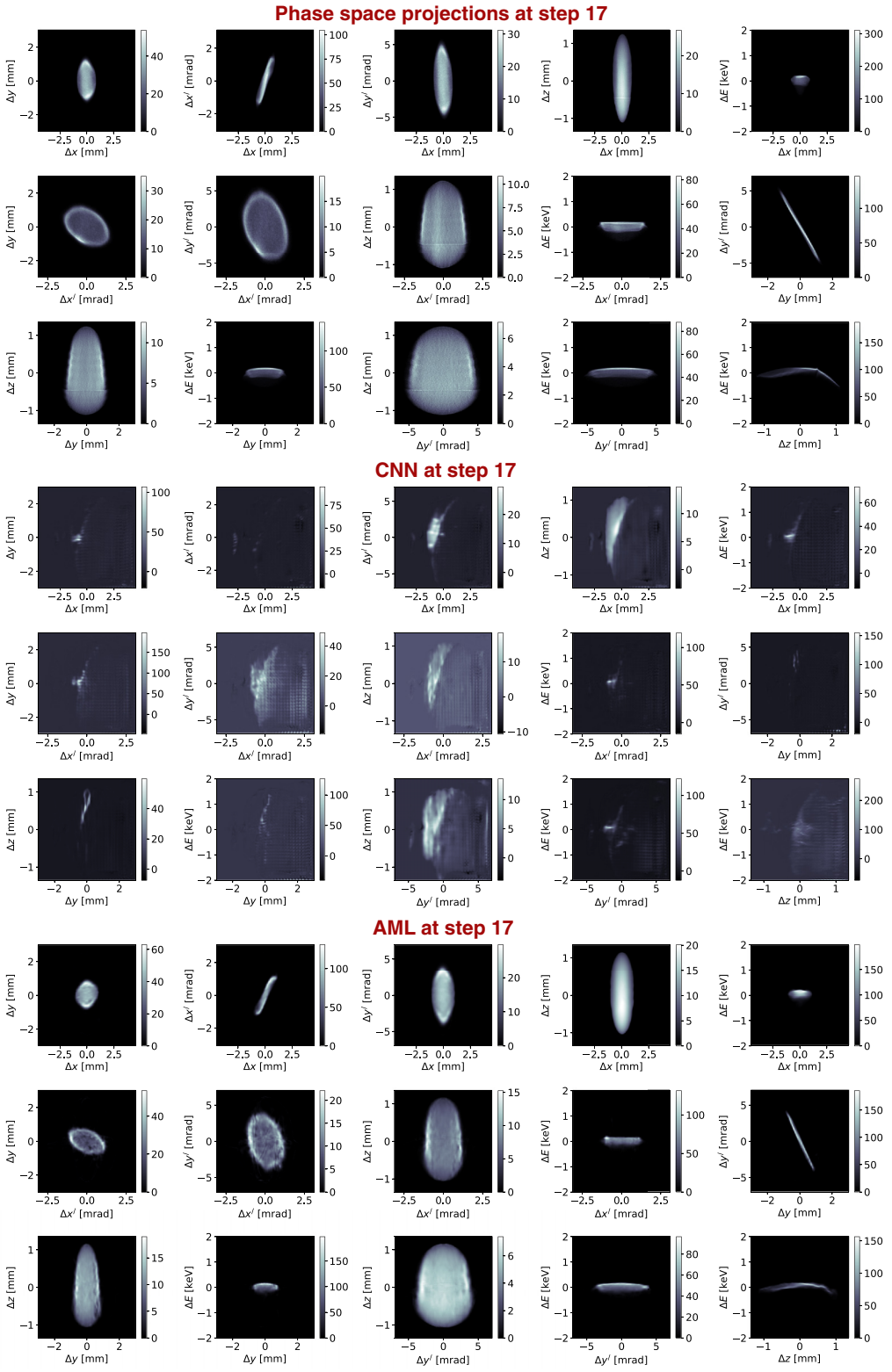

FIG. 20. AML and CNN predictions for step 16. At this point the CNN-based predictions have broken down, but the AML method is still providing relatively accurate and physically consistent estimates of the 6D phase space predictions

is an especially useful result as such high-dimensional deep neural network-based methods usually lack any kind of UQ.

In Sec. V we demonstrate the improved robustness provided by incorporating feedback within the latent space as opposed to simply using a trained CNN without such feedback.

In Secs. VI and VII we demonstrate that this method can handle truly time-varying systems whose variation takes them far beyond the span of the training data set. This is a result

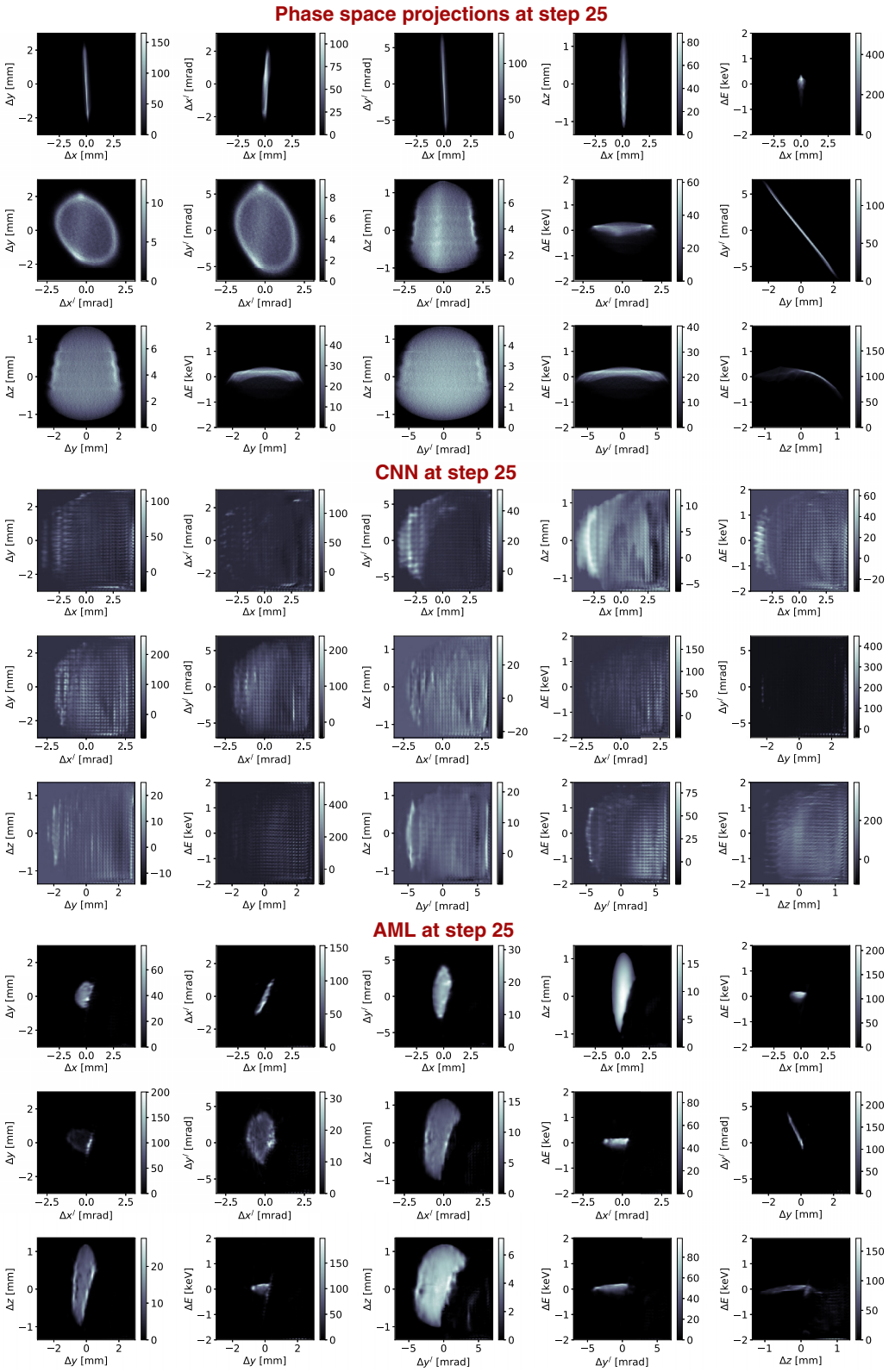

FIG. 21. AML and CNN predictions for step 25 where both the CNN and AML-based predictions have started to break down.

which goes far beyond what is possible with existing ML methods which catastrophically fail much sooner than our approach when simply utilizing a trained ML model without such adaptive feedback.

IX. CONCLUSIONS

The overall process that we have proposed for 6D charged particle beam phase space diagnostics can be broken down into the following algorithmic steps.

Data are experimentally collected or supplemented by synthetic data generated using a physics model, and an encoder-decoder CNN is then trained to learn the underlying physics implicit within the data by generating all phase space projections simultaneously as various channels of a single output, thereby learning correlations between the various phase space projections.

In application, the trained CNN is then applied in an adaptive and unsupervised manner in which we no longer have access to the correct input beam distributions and accelerator or beam parameters, and we assume that they are time-varying and cannot be measured noninvasively.

Because the network has learned the physics and correlations inherent in the data, by utilizing the adaptive feedback we are able to track the correct phase space projections by comparing a subset of the CNN's predictions to online measurements which guide fast adaptive feedback within the low-dimensional latent space.

As the network begins to make predictions for unknown beams it also provides a natural form of UQ.

We have demonstrated a general adaptive latent space tuning method for increasing the robustness of neural networks relative to time variation or distribution shift and farther beyond the span of the training data set. This approach can be useful for a large class of ML-based models for complex systems in cases where retraining is too time-consuming or is impossible without invasive measurements that interrupt regular operations. Although the approach presented here is applicable to any type of neural network, our demonstration of the approach focused on convolutional encoder-decoder architectures as they are incredibly powerful for working with very high dimensional data such as 2D (images) and 3D (volumes) distributions directly. We have demonstrated preliminary studies of the method, and for a complex particle accelerator application we have shown that it is more robust than traditional encoder-decoder CNNs for tracking all 15 projections of a charged particle beam with unknown and time-varying initial distribution, charge, and solenoid strength, and we have also provided a physics-informed method for uncertainty quantification.

ACKNOWLEDGMENTS

This work was funded by the U.S. Department of Energy (DOE), Office of Science, Office of High Energy Physics under Contract No. 89233218CNA000001 and the Los Alamos National Laboratory LDRD Program Directed Research (DR) Project No. 20220074DR. F.C. acknowledges support from the DOE, Office of Science, Office of Workforce Development for Teachers and Scientists, Office of Science Graduate Student Research (SCGSR) program. The SCGSR program is administered by the Oak Ridge Institute for Science and Education for the DOE under Contract No. DE-SC0014664. F.C. also acknowledges support from NSF PHY-1549132, Center for Bright Beams. D.F. acknowledges support by the DOE Office of Science, Office of Basic Energy Sciences under Contract No. DE-AC02-05CH11231.

[1] J. Zhu, Y. Chen, F. Brinker, W. Decking, S. Tomin, and H. Schlarb, High-Fidelity Prediction of Megapixel Longitudinal Phase-Space Images of Electron Beams Using Encoder-Decoder Neural Networks, Phys. Rev. Appl. **16**, 024005 (2021).

[2] E. Rrapaj and A. Roggero, Exact representations of many-body interactions with restricted-Boltzmann-machine neural networks, Phys. Rev. E **103**, 013302 (2021).

[3] S. Li, P. M. Dee, E. Khatami, and S. Johnston, Accelerating lattice quantum Monte Carlo simulations using artificial neural networks: Application to the Holstein model, Phys. Rev. B **100**, 020302(R) (2019).

[4] E. Flurin, L. S. Martin, S. Hacohen-Gourgy, and I. Siddiqi, Using a Recurrent Neural Network to Reconstruct Quantum Dynamics of a Superconducting Qubit from Physical Observations, Phys. Rev. X **10**, 011006 (2020).

[5] A. Scheinker and R. Pokharel, Adaptive 3D convolutional neural network-based reconstruction method for 3D coherent diffraction imaging, J. Appl. Phys. **128**, 184901 (2020).

[6] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, Reinforcement Learning with Neural Networks for Quantum Feedback, Phys. Rev. X **8**, 031084 (2018).

[7] A. Banerjee, J. D. Hart, R. Roy, and E. Ott, Machine Learning Link Inference of Noisy Delay-Coupled Networks with Optoelectronic Experimental Tests, Phys. Rev. X **11**, 031014 (2021).

[8] T. Goehring, M. Keshavarzi, R. P. Carlyon, and B. C. Moore, Using recurrent neural networks to improve the perception of speech in nonstationary noise by people with cochlear implants, J. Acoust. Soc. Am. **146**, 705 (2019).

[9] M. A. M. Ramírez, E. Benetos, and J. D. Reiss, A general-purpose deep learning approach to model time-varying audio effects, arXiv:1905.06148 (2019).

[10] G. Javadi, M. N. N. To, S. Samadi, S. Bayat, S. Sojoudi, A. Hurtado, S. Chang, P. Black, P. Mousavi, and P. Abolmaesumi, Complex cancer detector: Complex neural networks on non-stationary time series for guiding systematic prostate biopsy, in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020 23rd International Conference, Lima, Peru, Proceedings, Part III 23* (Springer, 2020), pp. 524–533.

[11] J. S. Dramsch, M. Lüthje, and A. N. Christensen, Complex-valued neural networks for machine learning on non-stationary physical data, Comput. Geosci. **146**, 104643 (2021).

[12] R. Calandra, T. Raiko, M. P. Deisenroth, and F. M. Pouzols, Learning deep belief networks from non-stationary streams, in *Artificial Neural Networks and Machine Learning–ICANN 2012 22nd International Conference on Artificial Neural Networks, Lausanne, Switzerland, Proceedings, Part II 22* (Springer, Berlin, 2012), pp. 379–386.

[13] A. Koesdwiady, S. Bedawi, C. Ou, and F. Karray, Non-stationary traffic flow prediction using deep learning, in *Proceedings of the 2018 IEEE 88th Vehicular Technology Conf. (VTC-Fall)* (IEEE, Piscataway, NJ, 2018), pp. 1–5.

[14] R. Kurle, B. Cseke, A. Klushyn, P. van der Smagt, and S. Günnemann, Continual learning with Bayesian neural networks for non-stationary data, in *International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia* (OpenReview.net, 2020).

[15] H. Shimodaira, Improving predictive inference under covariate shift by weighting the log-likelihood function, J. Stat. Plan. Infer. **90**, 227 (2000).

[16] G. Fishman, in *Monte Carlo: Concepts, Algorithms, and Applications*, edited by G. Peter (Springer-Verlag, New York, 2013).

[17] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. Smola, Correcting sample selection Bias by unlabeled data, in *Advances in Neural Information Processing Systems*, edited by B. Schölkopf, J. Platt, and T. Hoffman (MIT Press, Cambridge, 2006), Vol. 19.

[18] M. Sugiyama, S. Nakajima, H. Kashima, P. Von Buenau, and M. Kawanabe, Direct importance estimation with model selection and its application to covariate shift adaptation, in *Neural Information Processing Systems (NIPS)* (Citeseer, 2007), Vol. 7, pp. 1433–1440.

[19] M. Sugiyama and M. Kawanabe, *Machine Learning in Non-stationary Environments: Introduction to Covariate Shift Adaptation* (MIT Press, Cambridge, MA, 2012).

[20] D. E. Shea, R. Giridharagopal, D. S. Ginger, S. L. Brunton, and J. N. Kutz, Extraction of instantaneous frequencies and amplitudes in nonstationary time-series data, IEEE Access **9**, 83453 (2021).

[21] N. Kuklev, Y. Sun, H. Shang, M. Borland, and G. Fystro, Time-aware Bayesian optimization for adaptive particle accelerator tuning, in *Machine Learning and Physical Sciences workshop at the Advances in Neural Information Processing Systems Conference* (New Orleans, USA, 2022).

[22] H. K. Khalil and J. W. Grizzle, *Nonlinear Systems* (Prentice Hall, Upper Saddle River, NJ, 2002), Vol. 3.

[23] A. Scheinker and M. Krstić, Minimum-seeking for CLFs: Universal semiglobally stabilizing feedback under unknown control directions, IEEE Trans. Auto. Control **58**, 1107 (2012).

[24] A. Scheinker, Simultaneous stabilization and optimization of unknown, time-varying systems, in *Proceedings of the 2013 American Control Conference, Washington, DC* (IEEE, Piscataway, NJ, 2013), pp. 2637–2642.

[25] A. Scheinker and D. Scheinker, Bounded extremum seeking with discontinuous dithers, Automatica **69**, 250 (2016).

[26] A. Scheinker, Application of extremum seeking for time-varying systems to resonance control of RF cavities, IEEE Trans. Control Syst. Tech. **25**, 1521 (2016).

[27] A. Scheinker and D. Scheinker, Extremum seeking for optimal control problems with unknown time-varying systems and unknown objective functions, Intl. J. Adapt. Control Signal Proc. **35**, 1143 (2021).

[28] L. D. Landau, *The Classical Theory of Fields* (Elsevier, Moscow, Russia, 2013), Vol. 2.

[29] A. Malyzhenkov, Y. P. Arbelo, P. Craievich, P. Dijkstal, E. Ferrari, S. Reiche, T. Schietinger, P. Juranić, and E. Prat, Single- and two-color attosecond hard x-ray free-electron laser pulses with nonlinear compression, Phys. Rev. Res. **2**, 042018(R) (2020).

[30] T. van Oudheusden, P. L. E. M. Pasmans, S. B. Van Der Geer, M. J. de Loos, M. J. van der Wiel, and O. J. Luiten, Compression of Subrelativistic Space-Charge-Dominated Electron Bunches for Single-Shot Femtosecond Electron Diffraction, Phys. Rev. Lett. **105**, 264801 (2010).

[31] F. Lemery, P. Piot, G. Amatuni, P. Boonpornprasert, Y. Chen, J. Good, B. Grigoryan, M. Gross, M. Krasilinikov, O. Lishilin *et al.*, Passive Ballistic Microbunching of Nonultrarelativistic Electron Bunches Using Electromagnetic Wakefields in Dielectric-Lined Waveguides, Phys. Rev. Lett. **122**, 044801 (2019).

[32] M. Turner, E. Adli, A. Ahuja, O. Apsimon, R. Apsimon, A.-M. Bachmann, M. B. Marin, D. Barrientos, F. Batsch, J. Batkiewicz *et al.*, Experimental Observation of Plasma Wakefield Growth Driven by the Seeded Self-Modulation of a Proton Bunch, Phys. Rev. Lett. **122**, 054801 (2019).

[33] J. Faure, Y. Glinec, A. Pukhov, S. Kiselev, S. Gordienko, E. Lefebvre, J.-P. Rousseau, F. Burgy, and V. Malka, A laser–plasma accelerator producing monoenergetic electron beams, Nature (London) **431**, 541 (2004).

[34] J. Faure, C. Rechatin, A. Norlin, A. Lifschitz, Y. Glinec, and V. Malka, Controlled injection and acceleration of electrons in plasma wakefields by colliding laser pulses, Nature (London) **444**, 737 (2006).

[35] M. Labat, J. C. Cabadağ, A. Ghaith, A. Irman, A. Berlioux, P. Berteaud, F. Blache, S. Bock, F. Bouvet, F. Briquez *et al.*, Seeded free-electron laser driven by a compact laser plasma accelerator, Nat. Photon., **17**, 150 (2023).

[36] B. Cathey, S. Cousineau, A. Aleksandrov, and A. Zhukov, First Six-Dimensional Phase Space Measurement of an Accelerator Beam, Phys. Rev. Lett. **121**, 064804 (2018).

[37] K. Kabra, S. Li, F. Cropp, T. J. Lane, P. Musumeci, and D. Ratner, Mapping photocathode quantum efficiency with ghost imaging, Phys. Rev. Accel. Beams **23**, 022803 (2020).

[38] A. L. Edelen, S. Biedron, B. Chase, D. Edstrom, S. Milton, and P. Stabile, Neural networks for modeling and control of particle accelerators, IEEE Trans. Nucl. Sci. **63**, 878 (2016).

[39] A. Scheinker, C. Emma, A. L. Edelen, and S. Gessner, Advanced control methods for particle accelerators (ACM4PA) 2019 workshop report, arXiv:2001.05461.

[40] P. Arpaia, G. Azzopardi, F. Blanc, G. Bregliozzi, X. Buffat, L. Coyle, E. Fol, F. Giordano, M. Giovannozzi, T. Pieloni *et al.*, Machine learning for beam dynamics studies at the CERN Large Hadron Collider, Nucl. Instrum. Methods Phys. Res. A **985**, 164652 (2021).

[41] W. Blokland, K. Rajput, M. Schram, T. Jeske, P. Ramuhalli, C. Peters, Y. Yucesan, and A. Zhukov, Uncertainty aware anomaly detection to predict errant beam pulses in the Oak Ridge spallation neutron source accelerator, Phys. Rev. Accel. Beams **25**, 122802 (2022).

[42] A. Wolski, M. A. Johnson, M. King, B. L. Militsyn, and P. H. Williams, Transverse phase space tomography in an accelerator test facility using image compression and machine learning, Phys. Rev. Accel. Beams **25**, 122803 (2022).

[43] F. Mayet, M. Hachmann, K. Floettmann, F. Burkart, H. Dinter, W. Kuropka, T. Vinatier, and R. Assmann, Predicting the transverse emittance of space charge dominated beams using the phase advance scan technique and a fully connected neural network, Phys. Rev. Accel. Beams **25**, 094601 (2022).

[44] S. Li, F. Cropp, K. Kabra, T. J. Lane, G. Wetzstein, P. Musumeci, and D. Ratner, Electron Ghost Imaging, Phys. Rev. Lett. **121**, 114801 (2018).

[45] J. S. John, C. Herwig, D. Kafkes, J. Mitrevski, W. A. Pellico, G. N. Perdue, A. Quintero-Parra, B. A. Schupbach, K. Seiya, N. Tran *et al.*, Real-time artificial intelligence for accelerator control: A study at the Fermilab booster, Phys. Rev. Accel. Beams **24**, 104601 (2021).

[46] E. Fol, R. Tomás, and G. Franchetti, Supervised learning-based reconstruction of magnet errors in circular accelerators, Eur. Phys. J. Plus **136**, 365 (2021).

[47] E. Fol, R. Tomás, J. Coello de Portugal, and G. Franchetti, Detection of faulty beam position monitors using unsupervised learning, Phys. Rev. Accel. Beams **23**, 102805 (2020).

[48] J. Duris, D. Kennedy, and D. Ratner, Bayesian optimization at LCLS using Gaussian processes, in *Proceedings of the ICFA Adv. Beam Dyn. Workshop High-Intensity High-Brightness Hadron Beams (HB)* (Daejeon, Korea, 2018).

[49] R. Roussel, J. P. Gonzalez-Aguilera, Y.-K. Kim, E. Wisniewski, W. Liu, P. Piot, J. Power, A. Hanuka, and A. Edelen, Turn-key constrained parameter space exploration for particle accelerators using Bayesian active learning, Nat. Commun. **12**, 5612 (2021).

[50] C. Emma, A. Edelen, M. J. Hogan, B. O'Shea, G. White, and V. Yakimenko, Machine learning-based longitudinal phase space prediction of particle accelerators, Phys. Rev. Accel. Beams **21**, 112802 (2018).

[51] L. Gupta, A. Edelen, N. Neveu, A. Mishra, C. Mayes, and Y.-K. Kim, Improving surrogate model accuracy for the LCLS-II injector frontend using convolutional neural networks and transfer learning, Mach. Learn.: Sci. Tech. **2**, 045025 (2021).

[52] J. Kirschner, M. Mutný, A. Krause, J. C. de Portugal, N. Hiller, and J. Snuverink, Tuning particle accelerators with safety constraints using Bayesian optimization, Phys. Rev. Accel. Beams **25**, 062802 (2022).

[53] R. Shalloo, S. Dann, J.-N. Gruse, C. Underwood, A. Antoine, C. Arran, M. Backhouse, C. Baird, M. Balcazar, N. Bourgeois *et al.*, Automation and control of laser wakefield accelerators using Bayesian optimization, Nat. Commun. **11**, 6355 (2020).

[54] A. Scheinker and S. Gessner, Adaptive method for electron bunch profile prediction, Phys. Rev. ST Accel. Beams **18**, 102801 (2015).

[55] A. Scheinker, A. Edelen, D. Bohler, C. Emma, and A. Lutman, Demonstration of Model-Independent Control of the Longitudinal Phase Space of Electron Beams in the Linac-Coherent Light Source with Femtosecond Resolution, Phys. Rev. Lett. **121**, 044801 (2018).

[56] A. Scheinker, F. Cropp, S. Paiagua, and D. Filippetto, An adaptive approach to machine learning for compact particle accelerators, Sci. Rep. **11**, 19187 (2021).

[57] A. Scheinker, Adaptive machine learning for time-varying systems: Low dimensional latent space tuning, J. Instrum. **16**, P10008 (2021).

[58] D. Filippetto and H. Qian, Design of a high-flux instrument for ultrafast electron diffraction and microscopy, J. Phys. B: At. Mol. Opt. Phys. **49**, 104003 (2016).

[59] F. Ji, D. B. Durham, A. M. Minor, P. Musumeci, J. G. Navarro, and D. Filippetto, Ultrafast relativistic electron nanoprobes, Commun. Phys. **2**, 1 (2019).

[60] K. Pearson, LIII. on lines and planes of closest fit to systems of points in space, London Edinburgh Dublin Philos. Mag. J. Sci. **2**, 559 (1901).

[61] H. Abdi and L. J. Williams, Principal component analysis, WIREs Comput. Stat. **2**, 433 (2010).

[62] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, Physics-informed machine learning, Nat. Rev. Phys. **3**, 422 (2021).