Machine learning based phase space tomography using kicked beam turn-by-turn centroid data in a storage ring

Kilean Hwang*

Facility for Rare Isotope Beams, Michigan State University, East Lansing, Michigan, USA

Chad Mitchell and Robert Ryne

Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, California 94720, USA

(Received 22 May 2023; accepted 2 October 2023; published 23 October 2023)

When a charged-particle bunch in a storage ring is kicked to a large transverse offset, the time series describing the dynamics of the bunch centroid is determined both by the lattice focusing and by the Fourier transform of the 1D density profile of the bunch projected along the angle of the kick. In the presence of nonlinear focusing, we show that this fact can be exploited to enable 2D phase space reconstruction of the bunch (computational tomography) based only on turn-by-turn beam position monitor data. We demonstrate various tomography methods based on this principle, including machine learning methods, and discuss their advantages and disadvantages, and measure of reliability. We also mention a possible extension to 4D phase space computational tomography.

DOI: 10.1103/PhysRevAccelBeams.26.104601

I. INTRODUCTION

Information regarding the particle beam phase-space density can be essential for understanding and solving various beam dynamics issues that an accelerator physicist may encounter during accelerator operation. In addition, detailed knowledge of the beam phase-space density is essential to the success of advanced phase space manipulation techniques in FEL light sources [1], to characterizing beam halo and predicting beam loss in high-intensity proton linacs and rings [2,3], and to implementing effective diagnostics in storage rings with complex nonlinear dynamics [4].

The cornerstone of our tomography method lies in the theoretical finding that is described in this paper and a previous work [5]. It shows that, in the limit of substantial kick strength, the temporal evolution of the beam centroid can be directly related to the Fourier transform of the 1D beam profile along the angle of the applied kick. Here, a kicked beam refers to single or multiple bunches of particles that have been given a one-time transverse momentum kick by an external force. Consequently, employing a multishot measurement involving multiple kicks at various angles within the phase-space domain yields two-dimensional (2D) phase-space density distribution of the beam. However, the theoretical prediction is limited to large kicks that can result in the loss of the beam hitting the beam pipe. In cases where the kicks are sufficiently small to avoid inducing beam loss, the connection between beam profile and the temporal evolution of the beam centroid becomes too complex for theoretical analysis. In such a complex data relationship, machine learning method is often suited.

Most of the existing transverse beam phase-space reconstruction methods rely on measurements of camera images of a beam on a phosphor screen [6,7], or on beam profiles obtained using a thin metallic [8] or laser [9] wire scanner. These methods often require multiple shots (i.e., measurement of an image on a screen or a profile orthogonal to the wire) at varying quadrupole settings to rotate the beam to various angles in the phase-space. Each shot can potentially disrupt the beam and consume several minutes of valuable beam time [10]. When considering the cumulative effect of multiple shots, the time expenditure can stretch to tens of minutes.

In addition to the beam time cost, conventional methods are susceptible to the accumulation of discrepancies between the physics model and the real machine. The efficacy of beam phase-space tomography techniques heavily relies on the physics model governing beam transport along the beam line. This is because the model is responsible for predicting the rotation angles in the beam phase-space. The modeling becomes more intricate and error-prone, particularly when accounting for non-negligible nonlinear optics. Any discrepancy between the physics model and the actual machine undermines the precision of phase-space reconstruction.

*hwang@frib.msu.edu

This discrepancy becomes larger when the measurement location of the beam image or profile is distant from the point of interest for tomographic reconstruction due to the accumulation of model errors (resulting from misalignment, field calibration, etc.) through the beam transportation. This implies that if there are multiple locations of interest situated far from a specific measurement device, such as a wire scanner, it might be necessary to remeasure using nearby instruments for each location of interest.

The tomography methods discussed in this paper involve analyzing the turn-by-turn (TBT) beam position monitors (BPMs) data of a kicked beam until the beam decoheres. The decoherence of the kicked beam refers to the decay of beam centroid due to nonlinear phase mixing which usually takes a few thousand turns over the storage ring or a few seconds assuming microseconds per turn. BPMs employ broadband capacitive pickups to capture beam centroid with a resolution of the order of 100 MHz, which is significantly faster than wire-scanners (although wire-scanner measurement contains much richer information), and in a nondestructive manner [11]. Due to their affordability, compact design, and continuous and nondestructive measurement capability, BPMs are significantly more densely distributed along the beamline compared to phosphor screens or wire scanners. This means that if there are multiple locations of interest for tomography, one can exploit the closest BPM data for each location to minimize the accumulation of model error. This allows us to measure the required data for tomography simultaneously at these locations of interest.

Because the TBT BPM measurement time is expected to be a few seconds, the primary limitation lies in the computational time required by the specific tomography method employed. If the computational time cost of the tomography method is less than or comparable to the measurement time-cost, the tomography methods based on BPM data could be an order of magnitude faster than methods relying on wire-scanner-based measurements. In this paper, we try various techniques including machine learning methods. We find that neural-network (NN) supervised learning with a model reliability metric is a promising method.

The layout of this paper is as follows. In Sec. II, we start with a common analytical method that may be used as a benchmark. In Sec. II A and in corresponding appendices, we show how the projected beam profile may be analytically reconstructed in terms of the kicked beam TBT BPM data and optics parameters, in the limit of large kick strength. This is extended to the 2D beam phase space using the inverse radon transformation in Sec. II B. In Sec. III, we try to solve the problems associated with the theoretical profile estimation by using a Gaussian mixture model. In Sec. IV, we try to solve the computational complexity problem of the Gaussian mixture model by using a simple differentiable physics model of the particles. In Sec. V, we train a neural network model which can not only predict the beam phase-space, but is also able to estimate the reliability of the prediction. In Sec. VI, we present a proof-of-concept test of methods we presented on a highly nonlinear Hamiltonian system. Finally, the conclusion follows in Sec. VII.

II. ANALYTICAL METHOD

A. Projected beam profile recovery

The evolution of the canonical variables (x, p) describing particle dynamics in a 2D phase space takes the following form for a regular orbit, when expressed in normal form:

$$x(t) - ip(t) = (x_0 - ip_0)e^{i\omega t} = \sqrt{2J_0}e^{i\omega t - i\theta}, \quad (1)$$

where $x_0 = x(t=0)$, $p_0 = p(t=0)$, are the initial phase-space coordinates in normal form, $J_0 = (x_0^2 + p_0^2)/2$ is the action, $\omega(J)$ is the action dependent frequency and θ is the initial phase. When a beam is kicked to an offset $x = x_0$ and $p = 0$ at the time $t = 0$, the evolution of the beam centroid follows:

$$\langle x - ip\rangle_t = \int (x - ip)e^{i\omega t}\rho(x - x_0, p)\, dx\, dp, \quad (2)$$

where the bracket denotes the ensemble average over the particles of the beam, and $\rho(x, p)$ is the initial density of the beam phase-space. Now, let us assume that the oscillation frequency ω varies slowly over the beam phase space area. In this case, we can expand ω about the action value at the initial (offset) location of the beam centroid as:

$$\omega(\Delta J) = \mu_0 + \mu_1\Delta J\ldots = \mu_0 + \mu_1 x_0\Delta x\ldots, \quad (3)$$

where $\mu_0 \equiv \omega|_{J=J_0}$ is the angular frequency at the kick action $J_0 = x_0^2/2$, $\mu_1 \equiv \partial_J\omega|_{J=J_0}$ is the nonlinear detuning parameter at the kick action, $\Delta J \equiv J - J_0$, and $\Delta x \equiv x - x_0$. We also assume that the kick action is large compared to the beam emittance ϵ:

$$x_0/\epsilon \gg 1. \quad (4)$$

(Throughout this paper, we work in normalized units such that the beam emittance and betatron functions are $\epsilon = 1$ and $\beta = 1$.) Then, it can be shown (see Appendices A and B) that the discrete Fourier transform of periodically sampled beam centroid data reads:

$$\sum_{t=0}^{T} e^{-ikt}\langle x - ip\rangle_t = \frac{x_0}{2} + \frac{\pi x_0}{|\mu_1 x_0|}\lambda\left(\frac{k-\mu_0}{\mu_1 x_0}\right) + i\mathcal{P}\int \frac{x_0\lambda(\Delta x)}{\mu_0 + \mu_1 x_0\Delta x - k}\, d\Delta x, \quad (5)$$

where $\mathcal{P}$ represents the Cauchy's principal value and

$$\lambda(x) = \int \rho(x,p)\,dp \tag{6}$$

is the projected beam profile in the direction of the kicked beam offset. Therefore, one can reconstruct the projected beam profile using TBT beam centroid data from the BPM by inverting (5),

$$\lambda(x) = \frac{|\mu_1 x_0|}{\pi}\left[\Re\sum_{t=0}^{T} e^{-i(\mu_1 x_0 x+\mu_0)t}\frac{\langle x-ip\rangle_t}{x_0} - \frac{1}{2}\right]. \tag{7}$$

To be more general, when the kick angle θ is arbitrary, and when the beam momentum centroid data $\langle p\rangle_t$ is not available, then under some additional but general assumptions, the projected beam profile at the kick angle θ can be written, by (see Appendix B):

$$\lambda_\theta(s) = 2\frac{|\mu_1|}{\pi}\Re\sum_{t=0}^{T} e^{i\theta}e^{-i(\mu_1\sqrt{2J_0}s+\mu_0)t}\langle x\rangle_t - \frac{|\mu_1|}{\pi}\sqrt{2J_0}\cos^2\theta, \tag{8}$$

where λ_θ is the projected beam profile along the kick angle, s is the function argument of λ_θ, thereby it is the rotated coordinate along the kick angle, and $\langle x\rangle_t$ is the BPM data in normal coordinates.

This allows us to estimate the beam profile along the kick offset direction, provided that Eq. (3) and Eq. (4) are satisfied, optics parameters μ_0, μ_1 are known, and the kick strength J_0 and angle θ are also known.

If the beam phase-space density is same for all the fresh beam (that is not yet kicked for the profile estimation), we can also estimate the beam profiles along various angles $\{\theta\}$ in phase-space by kicking the fresh beam to the corresponding angles. It is important to emphasize that this requires a multishot measurement of the TBT BPM data. See Appendix C for illustration of the nonlinear decoherence of the kicked beam and application of Eq. (8) for beam profile prediction on a toy-model.

B. Inverse radon transformation

Recall that we can measure the beam profile along the kicked beam offset direction under some assumptions. This motivates us to reconstruct the 2D phase-space by applying the inverse radon transform (IRT) [12–14] to multiple beam profiles measured along various angles $\{\theta\}$. The IRT is an analytical method often used in medical computerized tomography (CT) to reconstruct (higher dimensional) images [15] from projected (lower-dimensional) images. Figure 1 illustrates a phase-space density reconstruction using the inverse radon transformation. We start with a complex initial beam phase-space density and use simulated BPM data using the toy-model in Appendix C. The original beam phase-space density, in normal coordinates,

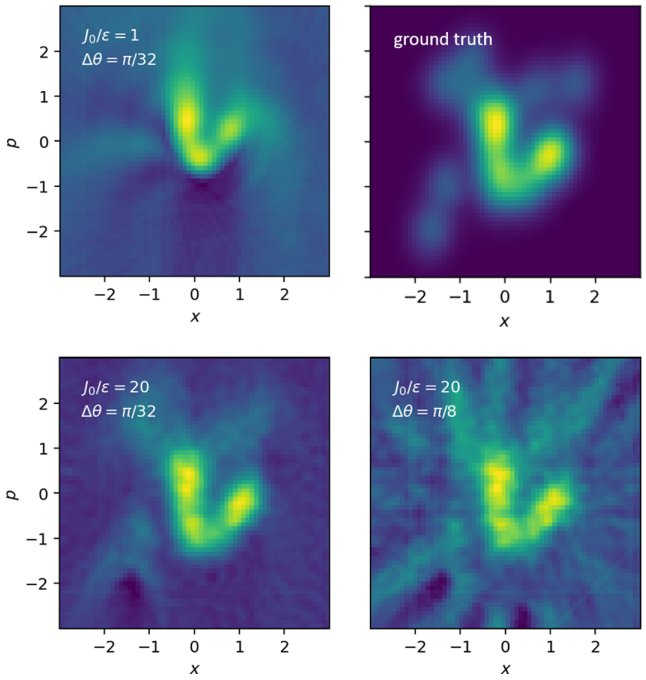

FIG. 1. Inverse Radon transform using estimated beam profiles over equally spaced angles. Top left: small kick strength such that kick action is equal to the beam emittance $J_0 = \epsilon$, Bottom row: large kick strength $J_0 = 20\epsilon$, Left column: 32 kicks at equally spaced distinct angles, Right column: 8 kicks at equally spaced distinct angles. The case of 32 large kicks agrees well with the ground truth that is on top-right and also shown in Fig. 11.

is shown on the top-right subplot and also in Fig. 11. Note that the image is relatively well constructed for the case of a large number of angular slices and a large initial kick (bottom-left plot of Fig. 1). This illustration also points out a few things: (1) There is limited angular resolution, as seen in the images on the right side of Fig. 1 when the number angular slides are limited, (2) the requirement of large initial kick (top left of Fig. 1), and (3) the requirement of knowledge of the optics μ_0, and μ_1 and kick action $\{J_0\}$, and angle $\{\theta\}$ to calculate the projected beam profiles from Eq. (8).

III. GAUSSIAN MIXTURE MODEL

Recall that the kicked beam turn-by-turn BPM data depends on the beam phase-space density as in Eq. (2). When the initial beam density is an isotropic Gaussian, the integration in Eq. (2) can be analytically carried out [16] without the large initial offset assumption $J_0/\epsilon \gg 1$:

$$\Re\int (x-ip)e^{i\omega t}\mathcal{N}(\boldsymbol{x}-\boldsymbol{x}_0,\sqrt{\epsilon}\boldsymbol{I})\,dxdp = \frac{x_0(1-\tau^2)+2p_0\tau}{(1+\tau^2)^2}\exp\left(-\frac{J_0}{\epsilon}\frac{\tau^2}{1+\tau^2}\right)\cos\Psi - \frac{2x_0\tau-p_0(1-\tau^2)}{(1+\tau^2)^2}\exp\left(-\frac{J_0}{\epsilon}\frac{\tau^2}{1+\tau^2}\right)\sin\Psi, \tag{9}$$

where $\mathcal{N}(\boldsymbol{x}, \sigma \boldsymbol{I})$ is the Gaussian kernel of mean $\boldsymbol{x} \equiv \{x, p\}$ and covariance $\sqrt{\epsilon}\boldsymbol{I}$, with $\boldsymbol{I}$ being the 2-by-2 identity matrix, $\tau \equiv \epsilon\mu_1 t$ and $\Psi \equiv \mu_0 t - \frac{(J_0/\epsilon)\tau^3}{1+\tau^2}$. This motivates us to build the initial beam phase-space model ρ_{GMM} by using a linear mixture of isotropic Gaussian kernels, referred to as a Gaussian mixture model (GMM) [17]. That is:

$$\rho_{\mathrm{GMM}}(x, p) = \frac{1}{G}\sum_{g=1}^{G} \mathcal{N}(\boldsymbol{x} - \boldsymbol{m}_g, \sigma_g \boldsymbol{I}). \tag{10}$$

Our goal is to fit the model so that ρ_{GMM} agrees with the true initial beam phase-space density ρ in the sense that:

$$\langle x \rangle_{GMM,t} = \langle x \rangle_t, \tag{11}$$

where

$$\langle x \rangle_{\mathrm{GMM},t} = \Re \int (x - ip) e^{i\omega t} \rho_{\mathrm{GMM}}(x - x_0, p - p_0)\, dx dp. \tag{12}$$

The parameters we want to fit for Eq. (11) are the optics parameter β, μ_0, and μ_1, the kick information $\{J_0\}$, and $\{\theta\}$ and the GMM parameters $\boldsymbol{m}_g$, and σ_g. Note that this method overcomes the weaknesses of the IRT because the optics and the initial offset parameters are not required to be known, and the presence of an analytic solution for Eq. (2) eliminates the requirement of a large initial kick. However, the high numerical complexity resulting from the large number of parameters is the main disadvantage. In order to mitigate this problem, we tried various tricks that we defer to Appendix D. Figure 2 illustrates a phase-space density reconstruction using a GMM whose parameters are fit to satisfy Eq. (11) using maximum *a posteriori* (MAP) estimation. Note that the isotropic Gaussian kernels are visible in the low density areas. The limited resolution, resulting from the finite number of Gaussian kernels, is in compromise with the numerical complexity. We used 100 Gaussian kernels, and the result took about 12 hours of computation time using a single CPU core. (The computation time may be reduced by many factors by using an efficient GPU implementation and gradient descent fitting with a numerical automatic differentiation technique.) To be consistent with the illustration of the IRT in Fig. 1, the simulated BPM data used 8 different kicked beams based on the toy-model in Appendix C. The same tune ω_0 and nonlinear detuning parameters ω_1 and ω_2 are used. The same kick angles $\{\theta\}$ are used. The difference is the kick strengths $\{J_0\}$. Here, we used eight equally spaced kicks $J_{0,k}/\epsilon = 2 + 2(k/7)$ (with $\epsilon = 1$ in normalized units). The different values of kick strength were needed to find the nonlinear detuning parameter μ_1. This is because the action variable can be estimated from the first few turns of BPM data assuming known optics parameter β, and the frequency of each kick from TBT BPM data. Note also that these choices of the kick strength do not satisfy $J_0/\epsilon \gg 1$, so as to illustrate one of the advantages of GMM compared to the IRT.

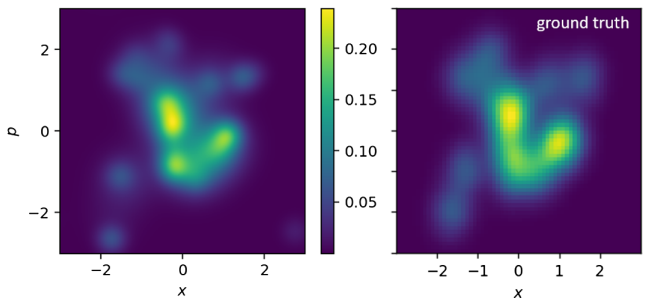

FIG. 2. Tomography using a Gaussian mixture model (GMM). The ground truth phase-space density is shown on the right and in Fig 11. Eight equally spaced kicks $J_{0,k}/\epsilon = 2 + 2(k/7)$ (with $\epsilon = 1$ in normalized units) and $\theta_k = \pi(k/8)$ with $k \in \{0, 1, \ldots, 7\}$ are applied. The corresponding (ground truth) 8 BPM data are simulated using the toy-model described in Appendix C. We added white noise of $\sigma = 0.1$ to the BPM data to mimic virtualization of a real measurement.

IV. PARTICLE MODEL

The main issue with the GMM from Sec. III is the high computational complexity associated with the parameter fitting. Recently, Ref. [18] illustrated 5D phase-space tomography using a numerically differentiable particle tracking simulator for beam transport in a quadrupole channel. The differentiable simulation enabled a gradient decent optimization to be used for fast parameter fitting. In the same spirit, we used a simple differentiable particle tracking simulator that is based on Eq. (1) with a simple frequency model:

$$\omega = \mu_0 + \mu_1 \Delta J, \tag{13}$$

where $\mu_0 = \omega|_{J=J_0}$, and $\Delta J = J - J_0$. Then the particle locations, optics parameters μ_0 and μ_1, and initial offsets $\{J_0\}$ and $\{\theta\}$ are optimized. For simplicity, we will refer this model as the particle model (PM). We use the same virtual BPM data from 8 different kicks as was used in the GMM case (Sec. III). With the aid of the differentiable simulator and the gradient descent method, the optimization was about 10 times faster (on a single CPU core) than the GMM case. This also allowed us look into the model uncertainty using an approximate Bayesian ensemble method. Details of these techniques are elaborated in Appendix E. We used 8 ensemble models which cost about 10 hours on a single CPU core, that is about 1 hour for each model. This can be further reduced with a GPU implementation. Figure 3 shows the tomography result. Note that the resolution issue of the GMM is resolved. In addition, the evolution of the fitted nonlinear detuning parameter μ_1 over the course of the model training is shown in Fig 4. Note that we obtained better accuracy for μ_1

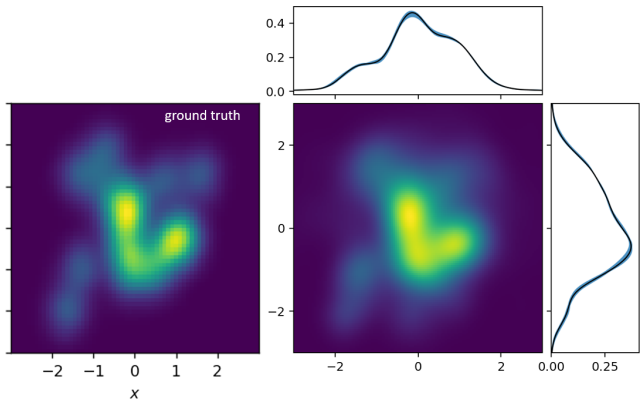

FIG. 3. Tomography using a differentiable particle model (PM). The ground truth phase-space density is on the left and shown in Fig 11. The mean prediction is shown in the colored density plot. The model uncertainty is visualized in the projected beam profile plots on top and right by shade (the black line is the mean prediction).

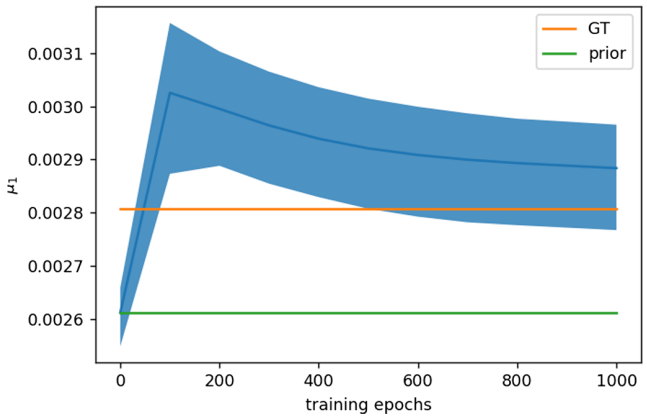

FIG. 4. Evolution of the fitted nonlinear detuning parameter μ_1 over the course of the model training. GT is the ground truth. The prior is based on estimation under a (single) Gaussian beam assumption. The blue shaded area represents the model uncertainty, while the blue line represent the mean prediction (taken over the model ensemble).

estimation after training, when compared to the prior estimate, which is based on a (single) Gaussian beam assumption. The better accuracy is reasonable, since the beam distribution that we used here is not Gaussian.

V. NEURAL NETWORK MODEL

The tomography methods using GMM or PM do not require large training datasets, using only a single input dataset (that is composed of the multishot kicked beam TBT BPM data with multiple different kicks). These methods train the model on the fly by fitting physical model parameters to the data. However, computational complexity can still be an issue. For this reason, we also try

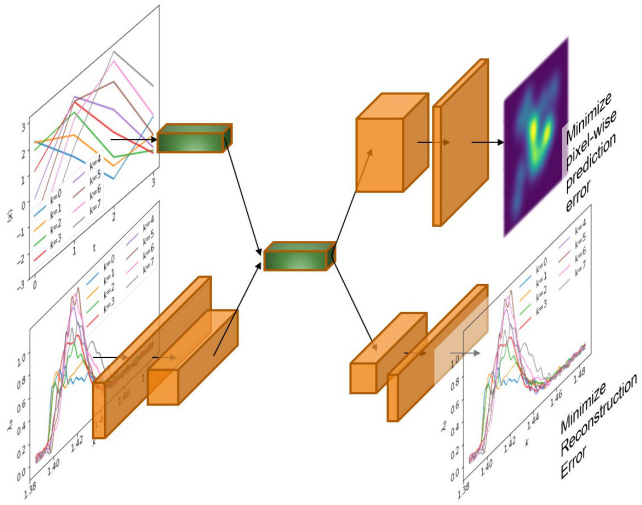

FIG. 5. Sketch of the input and output data flow of the NN used. Orange blocks represent the convolutional layers, green blocks represent dense layers.

supervised learning with a neural network (NN) model. Figure 5 illustrates the NN data flow. The input consists of the theoretically estimated projected beam profiles in Eq. (8), together with the first 4 turns of the beam centroid data. The theoretically estimated projected beam profiles may not correctly predict the true projected beam profiles because: (i) the large kick strength assumption can be violated, and (ii) the optics parameters and the kick parameters used in Eq. (8) may not be precisely known. Nevertheless, the NN may make corrections to predict the true beam phase-space, in spite of the incorrect projected beam profiles provided as input. The first few turns of BPM data are added to the input of the NN, because these contain useful information about the kick strength and the angles. Further details regarding the data preparation and training of the NN are provided in Appendix F.

The outputs are the phase-space density plot and the reconstructed projected beam profiles. An AutoEncoder-like structure for the reconstructed output is added for model reliability quantification. It may also allow the latent space to be trained in a more meaningful way, so that the model may better generalize. More precisely, the model may extrapolate better to data out of (the training data) distribution (OOD) [19]. The argument for using the reconstruction loss as an indication of model reliability follows: The ensemble method that is often used for uncertainty quantification is based on the expectation that the variance of the model prediction is larger when the input data is farther from the training data. In the same way, the reconstruction loss should be larger when the input data is farther from the training data. This fact is also often utilized for anomaly detection [20].

Once the model is trained, we checked the model performance on several sets of test data. Figure 6 shows a few random samples of NN predictions from test data,

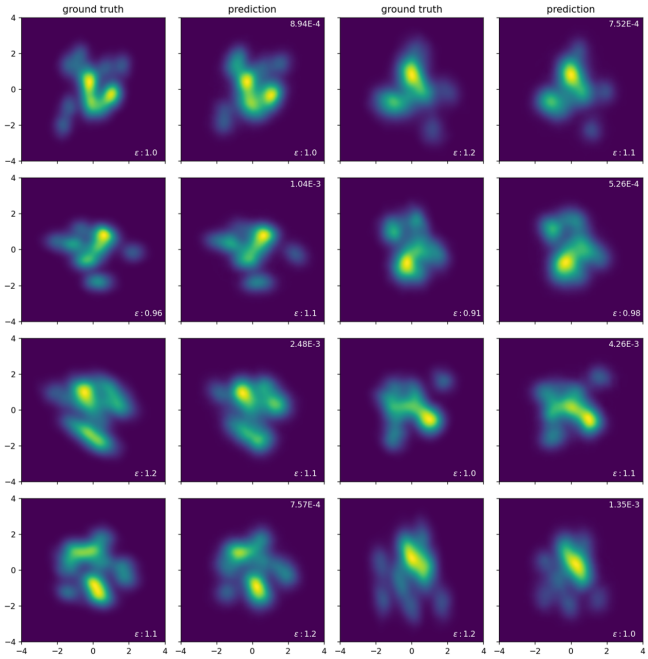

FIG. 6. Tomography using a neural network (NN) model. Prediction samples are shown in the second and fourth columns, and the corresponding ground truth in the first and third columns. The number shown on the top-right of each prediction is the MSE of the profile reconstruction. The unit of the MSE is the variance of the data values, because we normalized all data by the standard deviation of the dataset. The beam emittance (in normalized units) is calculated based on the image data, and values are denoted on the bottom right of each image.

with the corresponding ground truth shown for comparison. All the samples shown appear to agree well with the ground truth. We validated many other prediction samples using test data, including a few of the worst (out of all the test data) predictions in terms of the averaged pixel-by-pixel mean-squared-error (MSE). In most cases, the qualitative agreement of the predicted density image with the corresponding ground truth image was visually similar to that shown in Fig 6. The statistical beam moments, including the beam emittance, can also be calculated from the image data. Figure 6 shows that the predicted emittance values agree with the corresponding ground truth to within 10%–20%.

We also investigated the correlation between the loss of the density plot image and the loss of the reconstructed projected beam profile, each computed as the mean-squared error (MSE). Figure 7 shows these two loss values for each test data point, together with the fit obtained by linear regression. Note that they are reasonably correlated (Pearson's correlation coefficient was 0.53). This means that the loss in the reconstructed projected beam profile can serve as a reliability metric for the NN model prediction of the image. In order to refine the criteria for acceptance of the model prediction, it is good to check the retention curve [21]. Figure 8 shows the statistics of the image and profile MSE for retained test data based on the profile MSE. It looks desirable to choose retention criteria based on the profile MSE value that corresponds to 90% of the test data retention, as the curve shows a "kink" around this point. However, since a NN tends to be overconfident [22] of its predictions, it would be better to be more conservative. Therefore, one may like to choose retention criteria corresponding to 80%, 70% or lower of the test data retention. This corresponds to about 0.002, 0.0015 or lower profile MSE, respectively, for this example.

For further validation of the retention criteria, we applied the NN to data that is well out of the training data distribution (OOD). We generated OOD by using a larger variance for randomization of the simulation, while fixing the ground truth distribution used for comparison. More details are provided in Appendix F. Figure 9 shows predictions of the phase-space density images for several samples from OOD. Samples meeting the retention criterion (having profile MSE smaller than 0.0015) are marked by red frames. Note that the red framed samples that are retained (based on the profile reconstruction loss)

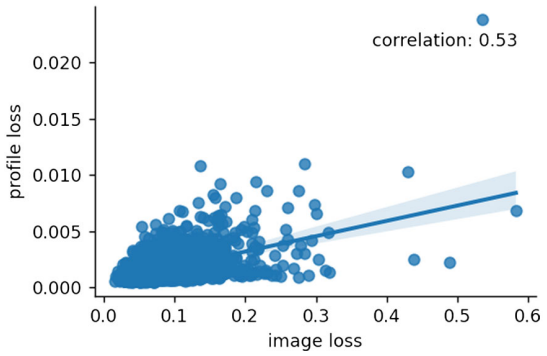

FIG. 7. Loss (MSE) of the predicted density plot image and loss (MSE) of the reconstructed projected beam profile, shown together with a linear fit. The Pearson's correlation coefficient is labeled. The unit of the image and profile MSE values are the variances of the image and profile data, respectively, because we normalized all data by the standard deviation of the data values.

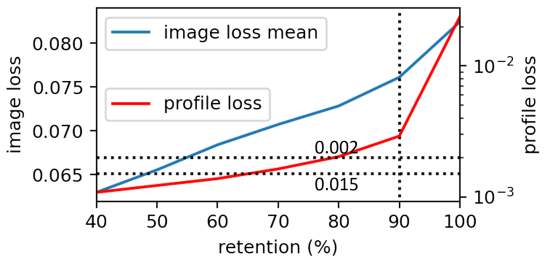

FIG. 8. Retention curve based on test data for the NN model described in this section. The horizontal axis denotes the retention percentage of the test data. The two horizontal dashed lines mark the profile MSE values at 0.0015 and 0.002. The vertical dashed line denotes 90% retention.

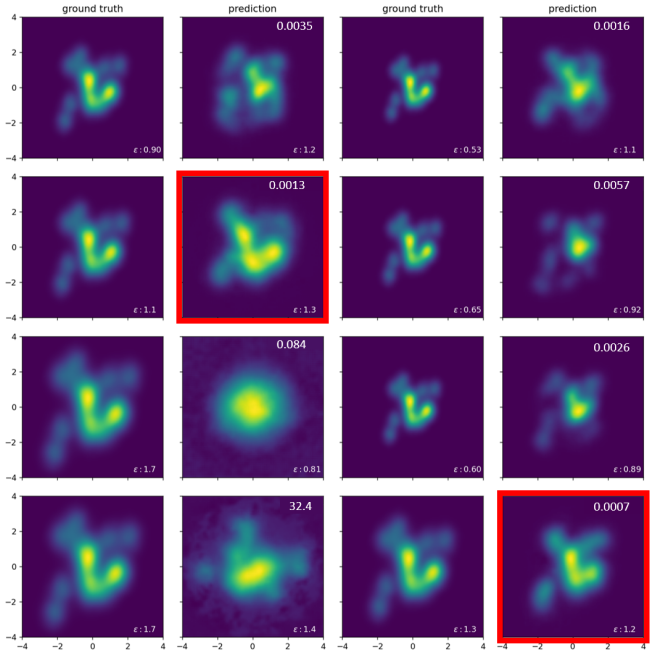

FIG. 9. Image prediction (second and fourth columns) and the corresponding ground truth (first and third column), for samples from OOD. The samples with beam profile losses smaller than 0.0015 are marked by red rectangular frames. The ground truth sample beam sizes vary because we randomized the emittance. The purpose was to follow the domain randomization technique used to adapt simulated data to a real machine. See Appendix F for details.

show density plot images that visually agree well with the ground truth.

VI. TEST ON NONLINEAR INTEGRABLE OPTICS

In this section, we demonstrate application of these techniques to a virtual machine. We consider the following 4D Hamiltonian [4,23,24]:

$$H=\frac{1}{2}(P_X^2+P_Y^2+|Z|^2)-\tau_{dn}\Re\left(\frac{Z}{\sqrt{1-Z^2}}\sin^{-1}(Z)\right), \quad (14)$$

where $Z=X-iY$ and τ_{dn} is a dimensionless parameter (representing a nonlinear magnet strength). Neglecting chromatic effects, this Hamiltonian is a model of the nonlinear integrable optics experiment at Fermilab's IOTA storage ring [25]. It is designed, in part, to study the effects of large intrinsic betatron tune spreads on beam halo formation and collective instabilities through Landau damping or decoherence [23]. We choose this Hamiltonian for its strong nonlinearity, which may represent a worst possible scenario for the applicability of the tomography methods we presented (neglecting chromatic aberrations, the beam kicking device's field variation over the bunch length, large BPM noise, bunch-by-bunch jitter, etc.).

The independent dynamical (time) variable for the Hamiltonian (14) is the phase advance associated with an underlying "bare" linear lattice. We refer to the bare phase advance over a single turn as μ_{dn}, so this parameter represents the frequency at $\tau_{dn}=0$ for both horizontal and vertical betatron oscillations. As we will mention later, all our 2D tomography methods failed on this test as stated, due to the strong nonlinear transverse coupling in some regions of the phase space. However, with a simple tweak of the horizontal detuning, our tomography methods performed reasonably well.

We use vertical kicks because the horizontal aperture in (14) is more limited due to the singular points of the potential at $X=\pm 1$. In order to find the transformation to normal form, at least to leading order, we expand Eq. (14) to second order in Y, considering the vertical dynamics only:

$$H|_{X=P_X=0}=\frac{1}{2}(P_Y^2+Y^2)+\tau_{dn}Y^2+O(Y^4). \quad (15)$$

Using the following type II generating function to define a canonical transformation [26],

$$G_2=(1+2\tau_{dn})^{1/4}Yp_y \quad (16)$$

the transformed Hamiltonian becomes:

$$H=\frac{\sqrt{(1+2\tau_{dn})}}{2}(p_y^2+y^2)+O(y^4,p_y^4). \quad (17)$$

Therefore, we have the following linear part of the normal form transformation:

$$y=(1+2\tau_{dn})^{1/4}Y, \qquad p_y=(1+2\tau_{dn})^{-1/4}P_Y, \quad (18)$$

where y, p_y are normal coordinates. For simplicity, we will call X, P_X, Y, P_Y the bare coordinates.

We used 8 vertical kicks given by $J_{0,k}/\epsilon=2+2(k/7)$ and $\theta_k=\pi(k/8)$ with $k\in\{0,1,\ldots,7\}$ in normal coordinates, as we have done throughout this paper. We manually sampled the simulation parameters μ_{dn}, τ_{dn}, and ϵ such that the betatron frequencies and detuning parameters are not very far from the NN training data distribution (see Sec. V and Appendix F). The betatron frequencies and the detuning parameters are calculated by tracking a single particle at each designed kick action. This made the nonlinear magnet strength τ_{dn} about 4–10 times smaller than the nominal design value in the IOTA storage ring, which is $\tau_{dn}=-0.4$ [27]. Once a kicked beam is prepared in the normal coordinates, we (linearly) transform the beam into bare coordinates to track the particles using the Hamiltonian Eq. (14) and record the virtual BPM data (at every bare phase advance μ_{dn}) for each kick. Finally, we (linearly) transform the BPM data back to normal coordinates. Once the data is prepared, we apply the PM and NN tomography methods to reconstruct the 2D beam phase space.

However, both the PM and NN methods failed to predict a beam phase-space density that agrees reasonably well with the ground truth. In the case of the NN method, large reconstruction losses are observed for all the samples with different simulation parameter settings: μ_{dn}, τ_{dn}, and ϵ. This is mainly because of the nonlinear coupling between x and y, which we did not include in the model Eq. (13). In order to address this problem, one approach is to include the coupling effect on the vertical frequency ω_y as follows, and thus to extend the problem from 2D to 4D tomography:

$$\omega_y = \mu_{y,0} + \mu_{y,1}\Delta J_y + \mu_{x,1} J_x. \tag{19}$$

This is reasonable unless the detuning term $\mu_{y,1}\Delta J_y$ dominates over the coupling term (which may happen for very large kick $J_0/\epsilon \gg 1$). Although such an extension can be done, it requires significant effort, and it is outside the scope of this paper.

In order to avoid the significant effects of nonlinear coupling, we added an artificial offset to the horizontal tune by advancing the betatron phase by $\Delta\mu_{x,0}/2\pi = 0.11$ each turn. Although the nonlinear coupling is still present, its effect may be less significant since the horizontal and vertical motions are well-separated in the frequency domain. This is not an unreasonable assumption, because most storage rings are designed to have different horizontal and vertical betatron frequencies to avoid resonances. With the addition of the artificial detuning, we prepared a few samples of the simulation parameters: μ_{dn}, τ_{dn}, and ϵ, and ran the NN model that we trained with a simple toy model Eq. (C1). Figure 10 shows the ground truth and mean NN prediction of the tomography images for these samples. Note, again, we see good agreement for samples having small reconstruction loss. The samples with large reconstruction loss likely correspond to OOD samples, which may result from nonlinear transverse coupling, the unknown nonlinear transformation to normal form, or from optics parameters $\{\mu_0\}$ or μ_1 that are far from the training data.

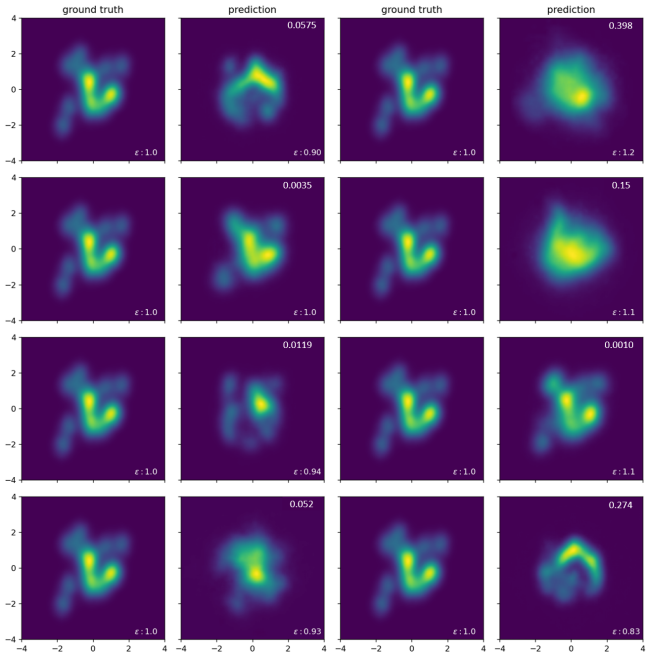

FIG. 10. Tomography images for DN Hamiltonian system with added horizontal detune. The predictions on the second and fourth columns and the ground truths on the first and third columns. The number on top right of the prediction plots are the profile reconstruction MSE.

VII. CONCLUSION

Phase-space tomography methods in 2D using kicked beam turn-by-turn BPM data were developed and tested on simulated environments. This was motivated by our work on theoretical reconstruction of the beam profile along the kick angle in the limit of large kick offset (compared to the beam emittance). This theoretical profile reconstruction requires prior knowledge of the kick action and angle, the betatron frequency, and the first-order nonlinear detuning parameter at the value of the kick action. We presented the theoretical formula and its derivation. A CT method is then presented, using IRT to combine the estimated profiles along various angles in the phase-space. However, this method is again limited by the large kick assumption and the requirement of prior knowledge. Such limitations can be alleviated with the methods presented in this paper. First, we used parameter fitting on simple machine learning and physics models. We used two models to represent the beam: The Gaussian mixture model (GMM) and the particle model (PM), using a finite number of macroparticles. For both representations of the beam, the dynamics are modeled using a simple but general amplitude dependent phase rotation [28]. By fitting the model parameters on the simulated kicked beam TBT BPM data (based on the amplitude dependent phase rotation model) we could reconstruct the 2D beam phase-space. However, the computational speed was shown to be not fast enough for online applications. Second, we used a supervised learning method to train a NN model that can predict the 2D beam phase-space together with a model reliability metric, based on an Auto-Encoder like input reconstruction. The theoretically estimated profiles at chosen angles in phase-space are used as input to the NN. This method could reconstruct the 2D beam phase-space quickly and successfully on the simulated kicked beam TBT BPM data (based on the amplitude dependent phase rotation model). For a more general test, we used the highly nonlinear Danilov-Nagaitsev Hamiltonian system [23] which is a model of one operating mode for the IOTA storage ring. In this test, we found that 2D tomography is insufficient when the horizontal and vertical betatron frequencies are close to each other and the kick action is not very large compared to the beam emittance. Therefore, 4D tomography that assumes a 4D amplitude dependent

frequency model would be needed. The extension to 4D would be natural. However, instead of extending the current work to 4D, we added an artificial horizontal detuning which can separate the nonlinear transverse coupling effect in the frequency domain. (Most storage rings are designed to have different horizontal and vertical betatron frequencies.) With this artificial detuning, we could reconstruct the phase-space density successfully using the tomography methods based on physics model fitting (both GMM and PM) and the NN (trained in a supervised way using the data from simple amplitude dependent phase rotation model).

ACKNOWLEDGMENTS

The author Kilean Hwang was supported by the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 and the U.S. Department of Energy under Cooperative Agreement No. DE-SC0000661. The authors Chad Mitchell and Robert Ryne were supported by the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This work used computer resources at the National Energy Research Scientific Computing Center.

APPENDIX A: DISCRETE FOURIER TRANSFORMATION KERNEL

Using a regularization trick, the kernel of the discrete Fourier transformation becomes:

$$\lim_{\epsilon\to 0_+}\sum_{t=0}^{\infty} e^{-ikt-\epsilon t} = \frac{1}{2} - \lim_{\epsilon\to 0_+}\frac{i}{2}\cot\left(\frac{k-i\epsilon}{2}\right). \tag{A1}$$

Using

$$\frac{1}{2}\cot\left(\frac{k}{2}\right) = \frac{1}{k} - O(k) \tag{A2}$$

and Sokhotsky's formula [29],

$$\lim_{\epsilon\to 0}\frac{1}{k\pm i\epsilon} = \mp i\pi\delta(k) + \mathcal{P}\frac{1}{k}, \tag{A3}$$

where $\mathcal{P}$ represents the Cauchy principal value, we have, for $|k| \ll 1$,

$$\lim_{\epsilon\to 0_+}\frac{i}{2}\cot\left(\frac{k-i\epsilon}{2}\right) = -\pi\delta(k) + i\mathcal{P}\frac{1}{k} + O(k). \tag{A4}$$

Therefore, in the sense of distributions:

$$\sum_{t=0}^{\infty} e^{-ikt} = \frac{1}{2} + \pi\delta(k) - i\mathcal{P}\frac{1}{k} \quad \text{for } |k| \ll 1. \tag{A5}$$

APPENDIX B: PROJECTED BEAM PROFILE AT AN ARBITRARY KICK ANGLE

When the beam centroid is kicked to an offset $x = x_0$ and $p = p_0$ (in normal coordinates) at $t = 0$, the evolution of the beam centroid is

$$\langle x - ip\rangle_t = \int (x-ip)e^{i\omega t}\rho(x-x_0, p-p_0)\,dxdp. \tag{B1}$$

Let $\rho_\theta(x,p)$ be the phase-space distribution function viewed from a rotated angle θ such that,

$$\rho_\theta(x,p) = \rho(x\,\cos\theta - p\,\sin\theta, p\,\cos\theta + x\,\sin\theta). \tag{B2}$$

Also, let the rotation angle be equal to the initial beam offset angle such that,

$$x_0 - ip_0 = \sqrt{2J_0}e^{-i\theta}. \tag{B3}$$

Plugging Eqs. (B2), (B3) into Eq. (B1), the beam centroid reads

$$\langle x - ip\rangle_t = \int (x-ip)e^{-i\theta}e^{i\omega t}\rho_\theta(\Delta x, p)dxdp, \tag{B4}$$

where $\Delta x \equiv x - \sqrt{2J_0}$. Now, using Eq. (A5) and plugging in the following,

$$\omega(\Delta x, p) = \mu_0 + \mu_1\sqrt{2J_0}\Delta x + \cdots \tag{B5}$$

the beam centroid motion in the frequency domain becomes:

$$\begin{aligned}
&\sum_{t=0}^{T} e^{-ikt}\langle x-ip\rangle_t \\
&= \frac{\sqrt{2J_0}}{2}e^{-i\theta} \\
&\quad + \frac{\pi e^{-i\theta}}{\sqrt{2J_0}|\mu_1|}\left(\sqrt{2J_0} + \frac{k-\mu_0}{\mu_1\sqrt{2J_0}}\right)\lambda_\theta\left(\frac{k-\mu_0}{\mu_1\sqrt{2J_0}}\right) \\
&\quad + ie^{-i\theta}\mathcal{P}\int \frac{(\sqrt{2J_0}+\Delta x)\lambda_\theta(\Delta x)}{\mu_0 + \mu_1\sqrt{2J_0}\Delta x - k}d\Delta x,
\end{aligned} \tag{B6}$$

where

$$\lambda_\theta(x) \equiv \int \rho_\theta(x,p)\,dp \tag{B7}$$

is the projected beam profile along the initial offset angle. Therefore, in the limit of large initial offset $|\sqrt{2J_0}/\epsilon| \gg 1$, the projected beam profile can be written as:

$$\lambda_\theta(x) = \frac{|\mu_1|}{\pi}\Re\sum_{t=0}^{T} e^{i\theta}e^{-i(\mu_0+\mu_1\sqrt{2J_0}x)t}\langle x - ip\rangle_t - \frac{|\mu_1|}{\pi}\frac{\sqrt{2J_0}}{2}, \tag{B8}$$

where $\Re$ is the real part operator.

On the other hand, when the beam centroid momentum $\langle p\rangle$ is not available, we find the following:

$$\begin{aligned}\sum_{t=0}^{T} e^{-ikt}\langle x\rangle_t = {} & \frac{\sqrt{2J_0}}{4}e^{-i\theta} + \frac{\pi e^{-i\theta}}{2\sqrt{2J_0}|\mu_1|}\left(\sqrt{2J_0} + \frac{k-\mu_0}{\mu_1\sqrt{2J_0}}\right)\lambda_\theta\left(\frac{k-\mu_0}{\mu_1\sqrt{2J_0}}\right) + i\frac{e^{-i\theta}}{2}\mathcal{P}\int\frac{(\sqrt{2J_0}+\Delta x)\lambda_\theta(\Delta x)}{\mu_0+\mu_1\sqrt{2J_0}\Delta x - k}d\Delta x \\ & + \frac{\sqrt{2J_0}}{4}e^{i\theta} + \frac{\pi e^{i\theta}}{2\sqrt{2J_0}|\mu_1|}\left(\sqrt{2J_0} - \frac{k+\mu_0}{\mu_1\sqrt{2J_0}}\right)\lambda_\theta\left(-\frac{k+\mu_0}{\mu_1\sqrt{2J_0}}\right) - i\frac{e^{i\theta}}{2}\mathcal{P}\int\frac{(\sqrt{2J_0}+\Delta x)\lambda_\theta(\Delta x)}{\mu_0+\mu_1\sqrt{2J_0}\Delta x + k}d\Delta x.\end{aligned} \tag{B9}$$

Therefore, in the limit of $|\sqrt{2J_0}/\epsilon| \gg 1$, and $\lambda_\theta(-x - \frac{2\mu_0}{\mu_1\sqrt{2J_0}}) \to 0$, the projected beam profile can also be written in terms of $\langle x\rangle_t$ as,

$$\lambda_\theta(x) = 2\frac{|\mu_1|}{\pi}\Re\sum_{t=0}^{T} e^{i\theta}e^{-i(\mu_1\sqrt{2J_0}x+\mu_0)t}\langle x\rangle_t - \frac{|\mu_1|}{\pi}\sqrt{2J_0}\cos^2\theta. \tag{B10}$$

which is re-written in Eq. (8). Note that the variable x is measured along the kick angle, while $\langle x\rangle_t$ is the BPM data in normal form.

APPENDIX C: ILLUSTRATION OF KICKED BEAM NONLINEAR DECOHERENCE AND PROFILE MEASUREMENT ON A TOY MODEL

Here we illustrate the application of Eq. (8) to a toy model which assumes a polynomial action dependent frequency:

$$\omega = \omega_0 + \omega_1 J + \omega_2 J^2/2, \tag{C1}$$

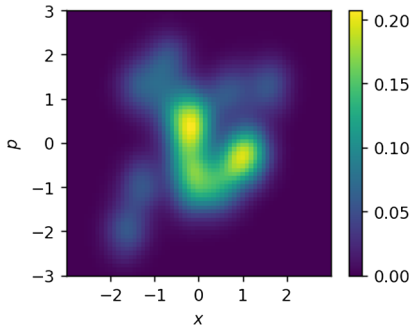

FIG. 11. A randomly generated phase-space density using multiple Gaussian kernels of randomly chosen means and covariance matrices. The emittance is $\epsilon = 1$, and dimensionless (all other length scale variables are normalized accordingly). The color bar represents the density.

where $\omega_0 = 2\pi \times 0.2222$, $\omega_1 = \omega_0/500$, and $\omega_2 = \omega_1/500$ with J being dimensionless (normalizing all lengths by the rms beam size).

Given the initial phase-space distribution shown in Fig. 11, we applied a kick offset and tracked each particle using the assumed frequency model. Due to phase-mixing, the beam centroid decays over time as shown in Fig. 12. We

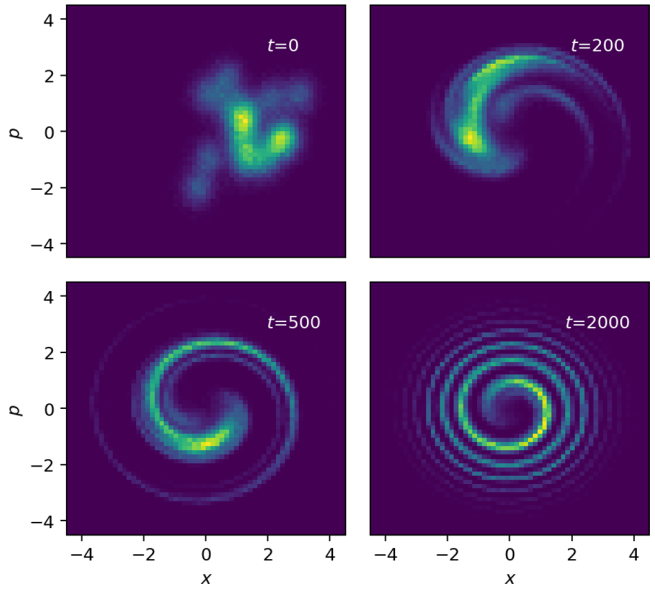

FIG. 12. Illustration of the phase-mixing of the kicked beam. The kick offset $x_0 = \epsilon$ is applied, and then each particle is tracked using the polynomial action dependent frequency $\omega = \omega_0 + \omega_1 J + \omega_2 J^2/2$. On top right of the each plot, t represents the turn number.

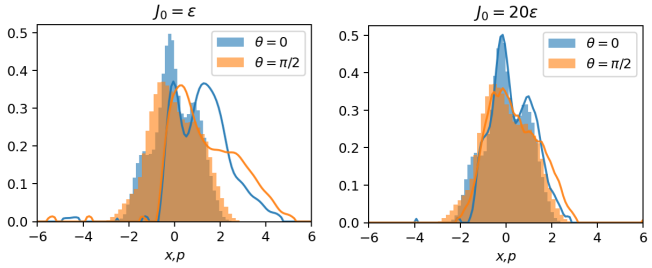

FIG. 13. Beam profile prediction. The shaded histogram represents the ground truth beam profile, while the lines represent the predicted beam profile using Eq. (8). The kick angles $\theta = 0$ and $\theta = \pi/2$ correspond to the x and p directions, respectively. Left: small kick strength $J_0/\epsilon = 1$. Right: large kick strength $J_0/\epsilon = 20$.

also added Gaussian noise of the form $\mathcal{N}(0, 0.1\epsilon)$ to the centroid data.

Figure 13 shows the beam profile prediction using kick angles $\theta = 0$ and $\theta = \pi/2$. We tried both large $J_0/\epsilon = 20$ and small $J_0/\epsilon = 1$ kick strengths. Here, we assumed the kick strengths, kick angles, frequency and detuning parameters are known upon application of Eq. (8). Note that the prediction is not reliable when the kick strength is not large compared to the initial beam emittance.

APPENDIX D: DETAILED PROCEDURE FOR FITTING GMM

Although a large number of Gaussian kernels are good for better expressibility of the GMM, too many kernels may not only overfit but also increase numerical complexity. The parameters we fit are the optics β (the betatron amplitude), μ_0, and μ_1, the initial offsets $\{J_0\}$ and $\{\theta\}$, and the GMM parameters $\boldsymbol{m}_g$. (We fix σ_g to reduce the problem dimension.) We use 100 Gaussian kernels and 8 different kicks that are equally spaced in $\theta \in [0, \pi]$ and $J_0 \in [2\epsilon, 4\epsilon]$. The procedure we used to realize the high-dimensional fitting follows.

1. Single Gaussian kernel

First, we used a single isotropic Gaussian kernel to estimate the optics parameters β, $\{\mu_0\}$ and μ_1, and the initial offsets $\{J_0\}$ and $\{\theta\}$. The nonlinear detuning parameter μ_1 is estimated by using a linear fit over (J_0, θ) pairs of 8 kicks. The following objective is minimized using a global optimizer called differential evolution [30]:

$$\sum_{k=1}^{8}\sum_{t=0}^{T}(\sqrt{\beta}\langle x\rangle_{1,k,t} - \langle X\rangle_{k,t})^2. \tag{D1}$$

Here the capital X represents the physical coordinate (in contrast to the normal coordinate x), the index 1 represents a single Gaussian kernel, k represents the index of the kick, and

$$\langle x\rangle_{1,k,t} \equiv \Re\int (x - ip)e^{i\omega t}\mathcal{N}(\boldsymbol{x} - \boldsymbol{x}_k, \boldsymbol{I})dxdp \tag{D2}$$

is the analytically expressible centroid with $\boldsymbol{x}_k$ representing the initial offset of the kth kick. The number of turns T for this fit is chosen when the envelope of the BPM data has decayed by half due to the nonlinear decoherence. For tomography based on the GMM model, we use the resulting estimate (that is obtained under the single Gaussian kernel assumption) as the starting point of the following procedure for fast fitting through a local minimization algorithm (specifically, the Nelder-Mead algorithm [31]). In addition, we used this initial guess to construct prior belief for MAP estimation.

2. Matching to theoretically estimated profiles

Second, we used 100 Gaussian kernels to build the GMM model. In order to reduce the number of parameters, we fixed the covariance σ_g for half of the Gaussian kernels to the estimated beam size obtained from the procedure in the previous subsection. The covariance for the remaining Gaussian kernels was set to one-quarter of this value. This is because we expected the large covariance kernels to capture the overall shape of the beam density, while the small covariance kernels may resolve finer details in the density. We start with the estimated parameters β, $\{\mu_0\}$ and μ_1, and the initial offsets $\{J_0\}$ and $\{\theta\}$ fixed, and we fit only the GMM parameters: $\{\boldsymbol{m}_g\}$ on the following objective:

$$\mathbb{E}_k(\lambda_{\theta_k} - \lambda_{\mathrm{GMM},\theta_k})^2, \tag{D3}$$

where $\mathbb{E}_k$ represents the average over the kicks, λ_{θ_k} is the theoretically estimated beam profile along the kick angle θ_k using Eq. (8), and $\lambda_{\mathrm{GMM},\theta_k}$ is the beam profile obtained for the GMM by adding all the kernel's projections. Although the kick strengths are not large enough to satisfy Eq. (4), the resulting estimation of GMM parameters $\boldsymbol{m}_g$ can be used for the starting point of the next procedure for fast fitting using a local minimization algorithm (specifically, the Nelder-Mead algorithm).

3. Matching to TBT BPM data

Third, we further tune the mean $\boldsymbol{m}_g$ of each Gaussian kernel while fixing the rest of the parameters, on the following objective:

$$\mathbb{E}_{k,t}(\sqrt{\beta}\langle x\rangle_{\mathrm{GMM}\,k,t} - \langle X\rangle_{k,t})^2. \tag{D4}$$

Finally, we tune all the parameters further using MAP estimation. We constructed the log of the prior in the following way,

$$\log \mathcal{P}_{\text{prior}} = -\mathbb{E}_k \frac{(\mu_0 - \tilde{\mu}_0)^2}{2\sigma_\mu^2} - \frac{(\beta - \tilde{\beta})^2}{2\sigma_\beta^2} - \mathbb{E}_k \frac{(J_{0,k} - \tilde{J}_{0,k})^2}{2\sigma_J^2} - \mathbb{E}_k \frac{(\theta_k - \tilde{\theta}_k)^2}{2\sigma_\theta^2}, \quad \text{(D5)}$$

where the tilde represents the prior mean that is estimated from the procedure of subsection D 1. The normalization factors σ_μ^2, σ_β^2, σ_J^2 and σ_θ^2 can be chosen based on one's fidelity to this prior mean. Here these values are chosen based on the decoherence time (since the number of turns affects the accuracy of the frequency measurement) and the estimated BPM noise, based on the tail part of the BPM data. The log of the likelihood is constructed as follows:

$$\log \mathcal{P}_{\text{like}} = -\mathbb{E}_{k,t} \frac{(\sqrt{\beta}\langle x\rangle_{\text{GMM}k,t} - \langle X\rangle_{k,t})^2}{2\sigma_X^2} - \frac{\log \sigma_X^2}{2}, \quad \text{(D6)}$$

where σ_X represents the size of the BPM noise, which we also regard as a model parameter. Equations (D5) and (D6) constitute the posterior, and by maximizing it, we finally estimate the phase-space density as shown in Fig 2.

APPENDIX E: DETAILED PROCEDURE FOR FITTING PM

In the case of tomography using GMM, we used a point estimation (through MAP) due to the high computational complexity required. In the particle model, with the differentiable simulator and gradient decent, we could reduce the training time by 10 times (for a single model) compared to the GMM case. Therefore we could train multiple (specifically 8) models for uncertainty quantification (with 8 times the computational cost). In order to estimate the prediction mean and model uncertainty in a Bayesian way [32], we performed the following procedure.

1. Prior samples based on the single Gaussian kernel assumption

First, as have done in the case of GMM, we use a single Gaussian kernel assumption for initial model parameter estimation. For each model, we prepared 1024 initial particles by random sampling from a normal distribution, thereby constructing a Gaussian beam. Then we fit the optics parameters μ_0 and μ_1 and kick information $\{J_0\}$ and $\{\theta\}$ for each model using a stochastic global optimizer (specifically, we used the differential evolution algorithm [30]) while fixing the particles' locations. These model parameters are regarded as a sample from prior belief that is based on the single Gaussian kernel assumption. For simplicity, we write a prior sample of the model parameters as ψ_{sample}, which includes particle locations, optics parameters and kick information.

2. Model parameter anchoring around the prior samples

Now we train each model while regularizing the model parameters through anchoring [33] as follows:

$$\text{loss}_{\text{anchor}} = \lambda_{\text{anchor}} \sum_\psi (\psi - \psi_{\text{sample}})^2/\sigma_\psi, \quad \text{(E1)}$$

where we used the variance of the prior samples for σ_ψ^2, and λ_{anchor} is a weight for the anchoring loss that we decided experimentally. In addition, we also added the following consistency loss:

$$\text{loss}_{\text{consistency}} = \lambda_{\text{consistency}} (\mu_1 - \text{slope}[\{\mu_0\}, \{J_0\}])^2, \quad \text{(E2)}$$

where the 'slope' represents the linear regression coefficient, so that the detuning parameter is consistent with the slope of the frequencies $\{\mu_0\}$ against the kick actions $\{J_0\}$, and $\lambda_{\text{consistency}}$ is a weight for the consistency loss. These regularization losses are added to the following loss that is the mean squared difference between the (virtually) measured and model predicted BPM data.

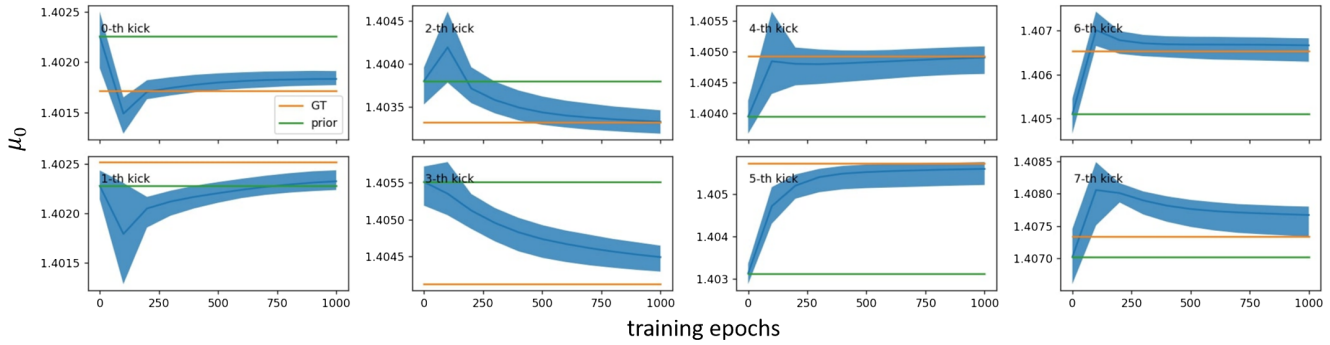

FIG. 14. Evolution of the fitted frequencies $\{\mu_0\}$ at each kick action $\{J_0\}$ over the course of the NN model training. GT is the ground truth. The prior is the estimate based on the Gaussian beam assumption. The shaded area represents the model uncertainty.

$$\text{loss}_{\text{BPM}} = \mathbb{E}_{k,t}(\langle x\rangle_{\text{PM},k,t} - \langle x\rangle_{\text{BPM},k,t})^2 \tag{E3}$$

where $\langle x\rangle_{\text{PM},k,t}$ the PM predicted BPM data, k represent the kick number, and t is the turn number. Once all the model trained separately, we aggregate them to get mean prediction of optics parameters and phase-space density from particle locations. The model uncertainty is measured by looking at the quantiles of predictions. Figures 14 and 4 show the evolution of optics parameters over the course of the model training. Note that as the training progresses so that the simulation particles locations move to form near ground-true phase-space density, the predicted optics parameters also move close to the ground truth values starting from the prior samples (that were estimated under single Gaussian kernel beam assumption).

APPENDIX F: DETAILED PROCEDURE FOR TRAINING NN

1. Data generation

We train the NN model in a supervised way. The model performance will strongly depend on the data used for training. We simulate virtual BPM data using the second order polynomial frequency toy model given in Eq. (C1). In order to adapt the NN model, trained using simulation data, to the real machine measurement data (*sim-to-real*), we use the so-called *domain randomization* technique. For each dataset, we sample an initial particle distribution (using multiple Gaussian kernels of randomly-chosen size and location), and we randomly sample an emittance value. We then scale the particles' locations so as to match the sampled emittance value. We then track the particles to generate TBT BPM data using the model in Eq. (C1), where the coefficients ω_0, ω_1, and ω_2 are also randomly sampled. The kick strengths and angles are also randomly sampled around the desired value. In summary, the simulation parameters to generate the training data are randomly sampled in the following way:

$$\epsilon \sim \mathcal{N}(1, \sigma_\epsilon), \tag{F1}$$

$$\omega_i \sim \mathcal{N}(\bar{\omega}_i, \sigma_{\omega_i}), \tag{F2}$$

$$J_{0,k} \sim \mathcal{N}(\bar{J}_{0,k}, \sigma_{J_{0,k}}), \tag{F3}$$

$$\theta_k \sim \mathcal{N}(\bar{\theta}_k, \sigma_{\theta_k}), \tag{F4}$$

where $\mathcal{N}(m, \sigma)$ represents the normal distribution of mean m and standard deviation σ. The mean values should be the expected parameter values for the experiment as designed. (Here, we choose arbitrary values.) The standard deviations should be chosen large enough to cover the uncertainty of one's belief in the expected parameter values of the experiment. For example, one may design an experiment to get input data for the NN model using one of the kick actions $\bar{J}_{0,k}$ at a kick angle $\bar{\theta}_k$. However, the beam kicking device's calibration, the beam energy, or the Twiss parameters at the BPM and kick locations may not be correctly known or available by measurement. Such uncertainty must be covered by the size of $\sigma_{J_{0,k}}$ or σ_{θ_k}, so that the training data contains the possible machine status and experimental design. In this proof-of-concept test, we used the following values:

$$\begin{aligned}
\bar{\epsilon} &= 1 & \sigma_\epsilon &= 0.2 \\
\bar{\omega}_0 &= 2\pi \times 0.222 & \sigma_{\omega_0} &= 0.002 \\
\bar{\omega}_1 &= \bar{\omega}_0/500 & \sigma_{\omega_1} &= \bar{\omega}_1 \\
\bar{\omega}_2 &= \bar{\omega}_1/500 & \sigma_{\omega_2} &= 10\bar{\omega}_2 \\
\bar{J}_{0,k} &= 2 + 2(k/7) & \sigma_{J_{0,k}} &= 0.02 \\
\bar{\theta}_k &= \pi(k/8) & \sigma_{\theta_k} &= 2 \text{ (degree)}.
\end{aligned} \tag{F5}$$

(The same mean values are used to generate the assumed ground truth BPM data in Secs. III and IV.) For each sample, we generate virtual BPM data (with added white

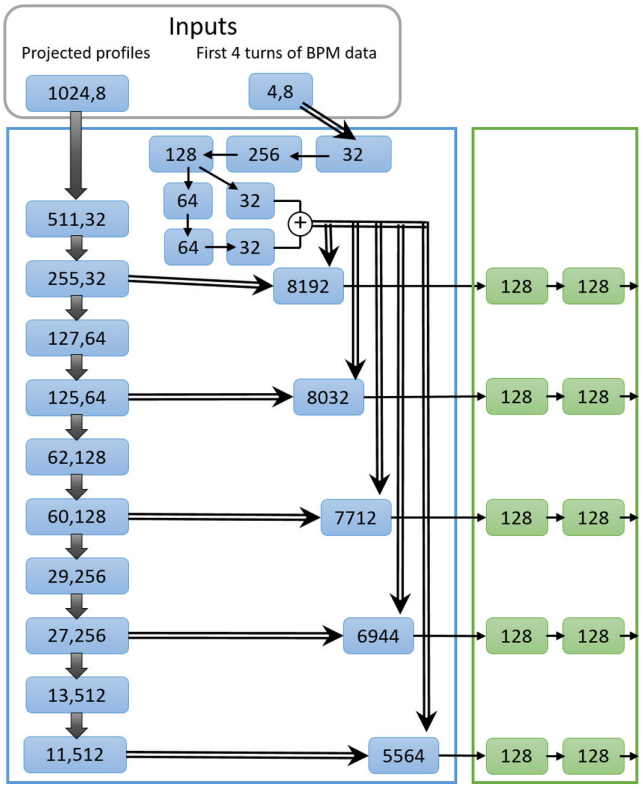

FIG. 15. Encoder. Numbers represent the array shape of the layer. Slim arrows denote the links connecting the fully connected layers. Bold arrows indicate the connections between convolution layers. We omit the convolution filter, strides, padding size, and number of filters for simplicity but can be roughly inferred from the array shape changes. Double-lined arrows represent reshaping and concatenation.

noise of size $\sigma = 0.1$), and calculate the theoretical estimate of the projected beam profile. The projected beam profile is calculated by plugging the nominal parameters (that is, from Eq. (F5), the expected machine status $\bar{\omega}_i$ and experimental design of the kick strengths $\bar{J}_{0,k}$ and angles $\bar{\theta}_k$) into Eq. (8). Note that this estimate does not represent the true projected beam profile. In addition, the kick strength is not large enough for Eq. (8) to be reliable. Nevertheless, this way of preprocessing the variable length of BPM data in the time domain into a fixed length of projected beam profile data in the frequency domain makes the NN input data consistent, regardless of the length of the TBT BPM data, the exact machine status, and the error of the kick strengths and angles. Also, for each sample, we calculate the phase-space density from the particle data to obtain a density plot for the NN output data. These data (the estimated profiles for each kick, the first few turns of BPM data for each kick, and the density plot) constitute the input and output of NN model training. The total number of datasets we prepared is 65536. The preparation of such a large amount of data (which took about a week) was possible due to the simplicity of the simulation model Eq. (1).

2. NN structure

We tried a few different NN structures, but only the U-Net [34] like structure worked. The schematic drawing in Fig. 5 illustrates the shallowest branches of the U-Net. Figure 15 shows the encoding part of NN. The projected profiles are processed through the deep convolution layers and then flattened and concatenated with the first 4 turns of the BPM data which are also processed through deep fully connected layers. Note it has 5 different latent layers represented by green boxes. The latent layers branched out from different levels of network depth signifying the U-Net like structure. Figures 16 and 17 show the decoding part for the phase-space density plot image and projected profile reconstruction, respectively. The single data pass over the NN took 0.5 sec with an Intel Xeon 2.2 GHz CPU.

3. NN training

As we have a large amount of data and a complex NN, we exploited a tensor processing unit (TPU) [35] for a majority of training and GPU for the last few epochs through the Kaggle [36] environment. Out of the 65536

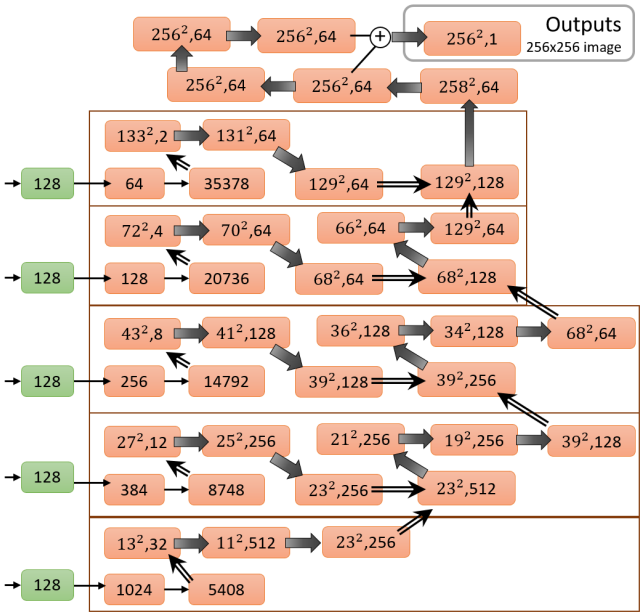

FIG. 16. Decoder for the phase-space density plot image. The green layers on the leftmost column are the same as the rightmost column shown in Fig. 15. Numbers represent the array shape of the layer. Slim arrows denote the links connecting the fully connected layers. Bold arrows indicate the connections between convolution or transpose convolution layers. We omit the convolution filter, strides, padding size, and number of filters for simplicity but can be roughly inferred from the array shape changes. Double-lined arrows represent reshaping and concatenation.

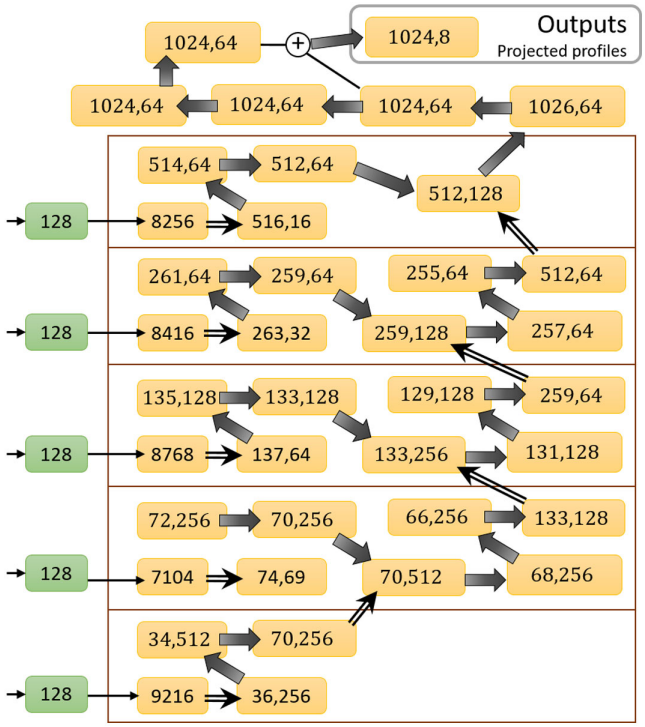

FIG. 17. Decoder for projected profile reconstruction. The green layers on the leftmost column are the same as the rightmost column shown in Fig. 15. Numbers represent the array shape of the layer. Slim arrows denote the links connecting the fully connected layers. Bold arrows indicate the connections between convolution or transpose convolution layers. We omit the convolution filter, strides, padding size, and number of filters for simplicity but can be roughly inferred from the array shape changes. Double-lined arrows represent reshaping and concatenation.

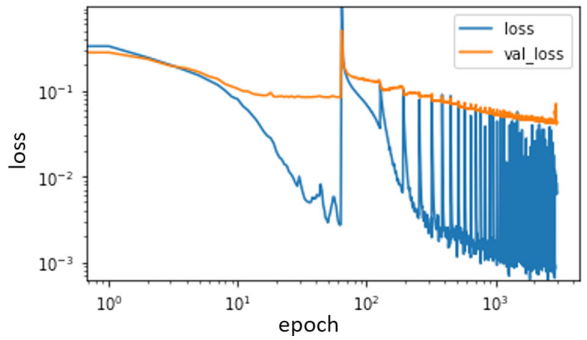

FIG. 18. Training history.

datasets, we used 2048 datasets for testing and another 2048 datasets for validation. We choose a small number of validation datasets in order to minimize the computational cost required for validation loss evaluation and implementation simplicity (due to the limited memory) of training. Due to implementation difficulties with TPU and limited memory issues, we used only randomly selected 6144 data out of 61440 training data for every 64 epochs when training with TPU. This caused large training loss fluctuation every 64 epochs. However, this could expedite the training procedure compared to GPU while minimizing implementation effort. In addition, we used batch sizes of 128 and 16 while using TPU and GPU respectively due to memory limitations. A larger batch size is desirable to reduce batch-to-batch distribution shift while a small batch size can reduce over-fitting through stochasticity of the gradient. We used ADAM optimization [37] and a learning rate scheduler for the first 64 epochs which reduced the learning rate from 4×10^{-5} to 5×10^{-5} within 64 epochs. This was an implementation mistake as opposed to reducing the learning rate slowly over the whole training epoch. After 64 epochs, the learning rate was fixed to 5×10^{-5}. The mean squared loss over normalized data is used for training criteria. Figure 18 shows training history. The spike around epoch 64 may be from the training data change (random selection of 6144 data out of 61440 training data) and implementation mistake regarding the learning rate scheduler. The last 240 epochs are trained using GPU using the whole training data, and each epoch in the last 240 epochs sees 61440/6144 times more data than the previous epochs. The training and validation loss increase at the last 240 epochs is not well understood. Maybe the smaller batch size of 16 (compared to 128 when using TPU) is leading to training instability.

[1] P. Piot, D. R. Douglas, and G. A. Krafft, Longitudinal phase space manipulation in energy recovering linac-driven free-electron lasers, Phys. Rev. ST Accel. Beams **6**, 030702 (2003).

[2] J. O'Connell, T. Wangler, R. Mills, and K. Crandall, Beam halo formation from space-charge dominated beams in uniform focusing channels, in *Proceedings of International Conference on Particle Accelerators, PAC'93, Washington, DC* (1993), Vol. 5, pp. 3657–3659, https://www.osti.gov/biblio/10157983.

[3] C. K. Allen and T. P. Wangler, Beam halo definitions based upon moments of the particle distribution, Phys. Rev. ST Accel. Beams **5**, 124202 (2002).

[4] C. Mitchell, R. Ryne, and K. Hwang, Bifurcation analysis of nonlinear Hamiltonian dynamics in the fermilab integrable optics test accelerator, Phys. Rev. Accel. Beams **23**, 064002 (2020).

[5] K. Hwang, C. Mitchell, and R. Ryne, Transverse 2D phase-space tomography using beam position monitor data of kicked beams, in *Proceedings of the 12th International Particle Accelerator Conference (IPAC2021)* (JACoW, Geneva, Switzerland, 2021), MOPAB235.

[6] K. Hock and A. Wolski, Tomographic reconstruction of the full 4D transverse phase space, Nucl. Instrum. Methods Phys. Res., Sect. A **726**, 8 (2013).

[7] A. Romanov, Beam phase space tomography at fast electron linac at fermilab, in *Proceedings of the 9th International Particle Accelerator Conference (IPAC'18), Vancouver, BC, Canada, 2018* (JACoW Publishing, Geneva, Switzerland, 2018), pp. 3146–3149, 10.18429/JACoW-IPAC2018-THPAF073.

[8] B. Hermann, V. A. Guzenko, O. R. Hürzeler, A. Kirchner, G. L. Orlandi, E. Prat, and R. Ischebeck, Electron beam transverse phase space tomography using nanofabricated wire scanners with submicrometer resolution, Phys. Rev. Accel. Beams **24**, 022802 (2021).

[9] J. C. Wong, A. Shishlo, A. Aleksandrov, Y. Liu, and C. Long, 4D transverse phase space tomography of an operational hydrogen ion beam via noninvasive 2D measurements using laser wires, Phys. Rev. Accel. Beams **25**, 042801 (2022).

[10] Thin wire scanners typically operate at low beam currents to prevent wire damage necessitating extended averaging time or slow wire movement.

[11] M. Wendt, Bpm systems: A brief introduction to beam position monitoring, arXiv:2005.14081.

[12] K. Hock and A. Wolski, Tomographic reconstruction of the full 4D transverse phase space, Nucl. Instrum. Methods Phys. Res., Sect. A **726**, 8 (2013).

[13] A. Watts, C. Johnstone, and J. Johnstone, Computed tomography of transverse phase space, in *Proceedings of the 2nd North American Particle Accelerator Conference, NAPAC2016, Chicago, IL, USA* (2017), TUPOA36, 10.18429/JACoW-NAPAC2016-TUPOA36.

[14] A. Romanov, Beam phase space tomography at fast electron linac at fermilab, arXiv:1811.04114.

[15] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, scikit-image: Image processing in Python, PeerJ **2**, e453 (2014).

[16] R. E. Meller, A. W. Chao, J. M. Peterson, S. G. Peggs, and M. Furman, Decoherence of kicked beams, Report No. SSC-N-360, 1987.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss,

V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. **12**, 2825 (2011), https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf.

[18] R. Roussel, A. Edelen, C. Mayes, D. Ratner, J. P. Gonzalez-Aguilera, S. Kim, E. Wisniewski, and J. Power, Phase space reconstruction from accelerator beam measurements using neural networks and differentiable simulations, Phys. Rev. Lett. **130**, 145001 (2023).

[19] A. Scheinker, Adaptive machine learning for robust diagnostics and control of time-varying particle accelerator components and beams, Information **12**, 161 (2021).

[20] C. Zhou and R. C. Paffenroth, Anomaly detection with robust deep autoencoders, in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'17* (Association for Computing Machinery, New York, NY, USA, 2017), pp. 665–674.

[21] O. Convery, L. Smith, Y. Gal, and A. Hanuka, Uncertainty quantification for virtual diagnostic of particle accelerators, Phys. Rev. Accel. Beams **24**, 074602 (2021).

[22] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. Hinton, Regularizing neural networks by penalizing confident output distributions, arXiv:1701.06548.

[23] V. Danilov and S. Nagaitsev, Nonlinear accelerator lattices with one and two analytic invariants, Phys. Rev. ST Accel. Beams **13**, 084002 (2010).

[24] C. Mitchell, Complex representation of potentials and fields for the nonlinear magnetic insert of the integrable optics test accelerator, LBNL Report No. LBNL-1007217, 2019, https://arxiv.org/abs/1908.00036.

[25] S. Antipov, D. Broemmelsiek, D. Bruhwiler, D. Edstrom, E. Harms, V. Lebedev, J. Leibfritz, S. Nagaitsev, C. Park, H. Piekarz, P. Piot, E. Prebys, A. Romanov, J. Ruan, T. Sen, G. Stancari, C. Thangaraj, R. Thurman-Keup, A. Valishev, and V. Shiltsev, Iota (integrable optics test accelerator): Facility and experimental beam physics program, J. Instrum. **12**, T03002 (2017).

[26] H. Goldstein, C. Poole, and J. Safko, Classical mechanics, 3rd ed., Am. J. Phys. **70**, 782 (2002), .

[27] S. Antipov, D. Broemmelsiek, D. Bruhwiler, D. Edstrom, E. Harms, V. Lebedev, J. Leibfritz, S. Nagaitsev, C. Park, H. Piekarz, P. Piot, E. Prebys, A. Romanov, J. Ruan, T. Sen, G. Stancari, C. Thangaraj, R. Thurman-Keup, A. Valishev, and V. Shiltsev, Iota (integrable optics test accelerator): Facility and experimental beam physics program, J. Instrum. **12**, T03002 (2017).

[28] H. S. Dumas, *The KAM Story* (World Scientific, Singapore, 2014), https://www.worldscientific.com/doi/pdf/10.1142/8955.

[29] J. J. Sakurai, *Advanced Quantum Mechanics* (Addison-Wesley Publishing Company, Boston, 1967).

[30] P. K. Storn R., Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. **11**, 341 (1997).

[31] F. Gao and L. Han, Implementing the nelder-mead simplex algorithm with adaptive parameters, Comput. Optim. Appl. **51**, 259 (2012).

[32] I. Osband, J. Aslanides, and A. Cassirer, Randomized prior functions for deep reinforcement learning, in *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (2018), pp. 8626–8638, https://proceedings.neurips.cc/paper/2018/hash/5a7b238ba0f6502e5d6be14424b20ded-Abstract.html.

[33] T. Pearce, N. Anastassacos, M. Zaki, and A. Neely, Bayesian inference with anchored ensembles of neural networks, and application to exploration in reinforcement learning, arXiv:1805.11324.

[34] O. Ronneberger, P. Fischer, and T. Brox, U-net: Convolutional networks for biomedical image segmentation, in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*, edited by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi (Springer International Publishing, Cham, 2015), pp. 234–241.

[35] N. P. Jouppi *et al.*, In-datacenter performance analysis of a tensor processing unit, SIGARCH Comput. Archit. News **45**, 1 (2017), https://arxiv.org/abs/1704.04760.

[36] Kaggle, https://www.kaggle.com.

[37] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.