RELATING INITIAL DISTRIBUTION TO BEAM LOSS ON THE FRONT END OF A HEAVY-ION LINAC USING MACHINE LEARNING[*]

A. D. Tran[†], Y. Hao, Michigan State University, East Lansing, MI, USA
J. L. Martinez Marin, B. Mustapha, Argonne National Laboratory, Argonne, IL, USA

Abstract

This work demonstrates using a Neural Network and a Gaussian Process to model the ATLAS front-end. Various neural network architectures were created and trained on the machine settings and outputs to model the phase space projections. The model was then trained on a dataset, with non-linear distortion, to gauge the transferability of the model from simulation to machine.

INTRODUCTION

A challenging problem in obtaining high beam power in hadron linacs, such as ATLAS, SNS, and FRIB, is understanding and minimizing uncontrolled beam loss, a major unexpected loss of the beam within the beamline. [1] In the low energy beam transport lines (LEBT), the beam must be carefully controlled to minimize the beam loss downstream. The beam is generally a collection of particles that can be described in six-dimensional space; three positions, and three momentum coordinates. For the DC beam in the LEBT, the longitudinal coordinates may contribute if the dipole is not controlled, but this effect will be ignored in this paper. Therefore, each charged particle is described by its location in the four-dimensional (4D) transverse phase space (x, x', y, y'), where primed coordinates are derivatives with respect to the longitudinal direction.

In the LEBT, multiple beam measurement devices such as Alison Scanners [2], Pepper-Pot emittance meters [3], wire scanners [4], and viewers are used to capture one-dimensional (1-D) or two-dimensional (2D) profile measurements, which are projections of the four-dimensional (4D) transverse phase space. Inferring the 4-D distribution from these projected 1-D and/or 2-D information is referred to as 4D tomography. Mathematical and physical methods, such as the maximum entropy principle [5, 6] , has been successfully demonstrated to realize the 4-D tomography in accelerators.

In this paper, we tested a data-driven approach to predict the beam loss using 2D projections measurements. The data was generated from virtual diagnostic instruments simulated using the beam dynamics code TRACK. The simulation data was from a test lattice adopted from the LEBT of the ATLAS accelerator and were used to develop a convolutional autoencoder to encode the data into a meaningful lower-dimensional representation, which relates the phase-space information to the beam loss.

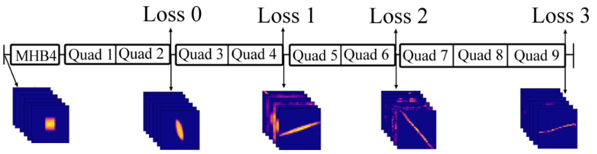

Figure 1: Cartoon of accelerator and beam measurements. The image shows where each beam measurement was collected from.

COLLECTING THE DATA

The presented study used data generated from the simulation of ATLAS's LEBT. The virtual diagnostic instruments capture the 4D phase space of the beam. The locations are shown in Fig. 1 where the phase-space projections of the 4D phase space and losses are measured and saved. This amount of information is currently hard to achieve in a real accelerator but it is used to study the feasibility of the method.

Generating Data Using TRACK

TRACK is a ray-tracing or particle-tracking code that can: (1) represent external fields accurately within the aperture. (2) calculate the particle coordinate at any point in the space. (3) determine beam loss in both the ideal case and in the presence of complex field errors and device misalignments [7].

TRACK simulations were used to gather data as machine data was unavailable. Over a million data point was generated on Michigan State University's high-performance computing cluster. This is needed since a significant amount of data will be required for training autoencoders to high fidelity. The parameters for these simulations were varied according to Table 1 and were chosen within and interpertable range. The data was filtered so that the initial beam distributions were contained within the beam aperture, resulting in a final data set of around 430,000 simulation points.

2D phase-space projections where taken by depositing the particles onto an $n \times n$ grid using pairs of the coordinates axes, (x, x', y, y'). This resulted in 6 independent projections.

Non-linear Field

A separate data set was generated to test the generalizability of the model which will be explained later. This was done with a perturbation to the initial distribution by putting a non-linear magnetic field, such as a sextupole, at the beginning of the simulation.

[*] This work was supported by a sub-reward from Argonne National Laboratory and by the U.S. Department of Energy, under Contract No. DE-AC02-06CH11357.

[†] tranant2@msu.edu

Table 1: Parameter range used to generate data set of the initial beam distributions and quadruple settings.

Input	
Voltages on Quadruples 1, 3, 5	uniform random number from [0,8] V
Voltages on Quadruples 2, 4, 6	uniform random number from [-8,0] V
Initial Distribution	random distribution from 9 built in distribution
$\epsilon_{x,y}$	$0.12 + Normal(\mu = 0, \sigma = 0.012)$ cm*mrad
$\alpha_{x,y}$	$Normal(\mu = 0, \sigma = 1)$ unitless
$\beta_{x,y}$	$100 + Normal(\mu = 0, \sigma = 10)$ cm/rad
Output	
Number of particles left	[0,10000] particles. Taken at 4 different points
Position of all particles	Taken at 5 different points

CREATING THE MODEL

Autoencoder

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

In the model, a convolutional autoencoder was implemented in PyTorch [9] to reduce the input dimension. A convolutional autoencoder uses a convolutional neural network as the encoder and decoder. A convolutional neural network is a type of neural network used to analyze visual information [10]. This has the advantage over principal component analysis [11], another data reduction algorithm, in that it includes spacial information, and can account for non-linear effects by using non-linear activation functions in the network. Activation functions are functions that map the input onto a set range. It was found that the reLu activation function and eLu activation function were the best activation functions to use [12], which in this case, helps the model to train fast and be less likely to fail during training.

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

Modeling

A neural network was used to create a surrogate model of the ATLAS front-end as shown in Fig. 2. The architecture is composed of first an encoder-decoder block to reduce, separately, each of the six phase-space projections into lower latent dimensions and then concatenated together. The quad settings were also concatenated onto this vector. This vector is inserted into a fully connected layer which does a phase-space transformation on the latent dimension. The output from this is inserted into a decoder block to reconstruct the 2D phase-space projections at that location, another fully connected layer to predict the number of particles left, and into another fully connected layer to repeat the same process until the end.

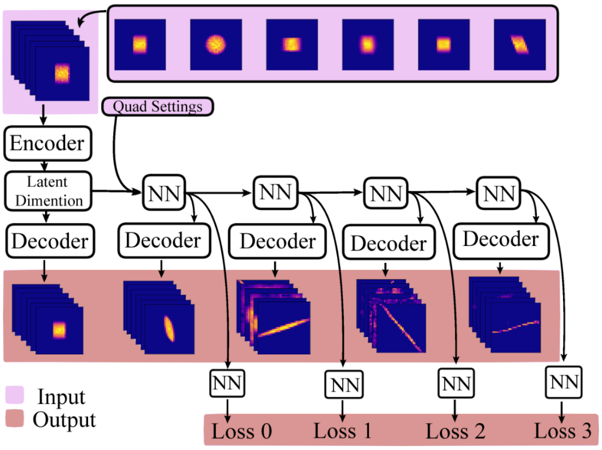

Figure 2: Cartoon of architecture. During training, the model takes all the 2D projections and loss value as input into the training. During testing, only the initial 2D projections were given and the model predicts the loss values and 2D projections in addition.

The encoder-decoder block uses a convolutional autoencoder as described in the previous section. A decoder was not trained for every location, but was combined for each projection. This saves limited GPU memory and produces a more generalized decoder.

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

RESULTS

The model was tested on a separately generated dataset using the same parameters for the original and non-linear

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-MOPA90

dataset. Only the initial distributions were given, but the model would still predict the 2D projections and beam transmission at the other locations downstream. Then, to test the generalization of the model, a nonlinear field in the form of a sextupole [13] was added to the beginning of the simulation to generate a dissimilar subset of inputs.

For reference, an error of less than 1%, or 100 particles, for losses within 2 standard deviations from the mean would be sufficiently good for the prediction of the loss on ATLAS since it is a relatively low power machine. For the rest of the paper, the percentage refers to the maximum bound within 2 standard deviation. The error is defined as the absolute difference between the ground truth and the predicted values divided by the total number of particles. The obtained values were plotted in Fig. 3A as a correlation graph. If there were no error, then there would be a perfectly straight line. Given the total number of particles is 10^4, and a maximum 2 standard deviation error for the original data set using six projections is 263 particles, the error would be around 3%.

This was then tested on the nonlinear sextupole distribution with fair results, an error of around 5.5% as shown in Fig. 3C. The model was able to generalize fairly well, however, it is still far from the ideal case.

Since random setting on accelerators usually results in high loss, most of the dataset would be skewed towards high loss, resulting in higher accuracy in those cases for the model since there is more data in those cases. To analyze this effect, the dataset was split into bins and as expected, the bin of particle loss between 9000 – 10000 has an error around 2.5% and for the bin of particle loss between 0 – 1000, the error was as high as 5%.

Testing on a Smaller Data Set

The same model was tested again, but with the (x, y'), (x', y'), and (y, x') projections removed. In Fig. 3B, the error predictions from the original data set show an improvement in the accuracy for "Loss: 0" while it has around the same error for the other losses. This is likely due to overfitting as the predictions from the non-linear data set show a loss of accuracy overall as seen in Fig. 3D; however, the model was shown to work with half the image data used, making this model more practical.

CONCLUSION

A proof-of-principle machine learning based model has been reported to test a ML-based 4D tomography using its 2D projections and its capability to predict the beam transmission. The result shows that if given only three projections of the 4D phase space, the projections can be reduced into a smaller latent dimension that contains the core information, which can then be used to predict the beam transmission downstream. The latent dimension was verified to have contained the core information through a decoder which correctly reconstructed the encoded images. This method generalizes fairly well to initial beam distributions with non-linear perturbations, showing robustness and the potential to model the real machine.

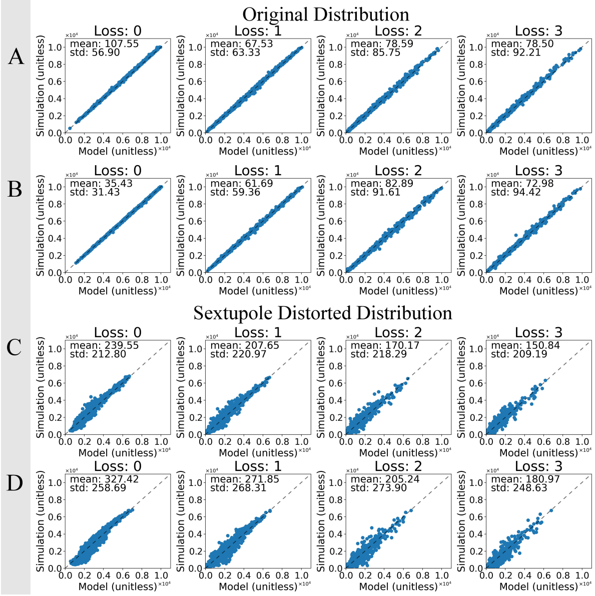

Figure 3: Histogram of original data set using six projections (A), and same model but using three projections (B). Histogram of original data set using six projections (C), and same model but using three projections (D).

Before applying this method to a real machine, it should be noted that this is a simplified model of an actual accelerator. First, this model assumes that the accelerator elements can be modeled by a single parameter. Therefore, more complicated effects, such as misalignment of the magnets and the longitudinal overlapping of transverse magnets, are not considered. Second, the model assumes the 2D projections can be precisely measured with no errors. Measurement errors exist, but they can be reduced by taking multiple measurements with different optics settings, which was not done in our simplified model.

To use this method in experiments methods known as "transfer learning" will have to be tested. This allow knowledge learned from the source dataset to be transferred to a target dataset [10]. This is done by freezing the model, adding an extra layer, and training that layer with the frozen model on the real machine. After that, the whole model can then be unfrozen and trained with a much smaller learning rate in order to fine-tune the model.

Finally, this work assumes no accelerator knowledge, but further research will involve incorporating physics into the model. Some ways this could be done is by encoding constraints in the loss function during model training or by incorporating domain knowledge by including the transfer matrices in the calculation as a prior information. The positive results of this work give hope that incorporating this knowledge may save time, increase sample efficiency, and further reduce the beam transmission error.

REFERENCES

[1] A. V. Aleksandrov and A. P. Shishlo, "Path to Beam Loss Reduction in the SNS Linac Using Measurements, Simulation and Collimation," in *Proc. HB'16*, Malmö, Sweden, Jul. 2016, pp. 548–552. doi:10.18429/JACoW-HB2016-THAM5Y01

[2] P. W. Allison, J. D. Sherman and D. B. Holtkamp, "An Emittance Scanner for Intense Low-Energy Ion Beams," in *IEEE Trans. Nucl. Sci.*, vol. 30, no. 4, pp. 2204–2206, Aug. 1983. doi:10.1109/TNS.1983.4332762

[3] A. Pikin, A. Kponou, J. Ritter, and V. Zajic, "Pepper pot emittance meter," in *Tech. Rep.*, Brookhaven National Lab, Upton, NY, USA, 2006.

[4] H. Koziol, "Beam diagnostics for accelerators," in *Tech. Rep.*, CERN, Geneva, Switzerland, 2001.

[5] D. Reggiani, M. Seidel, and C. Allen, "Transverse phase-space beam tomography at PSI and SNS proton accelerators," in *Proc. IPAC'10*, Kyoto, Japan, 2010.

[6] 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*, vol. 25 p. 042801, 2022. doi:10.1103/PhysRevAccelBeams.25.042801

[7] V. Aseev, P. Ostroumov, E. Lessner, and B. Mustapha, "Track: The new beam dynamics code," in *Proc. PAC'05*, pp. 2053–2055. doi:10.1109/PAC.2005.1591006

[8] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders." doi:10.48550/arXiv.2003.05991

[9] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library." doi:10.48550/arXiv.1912.01703

[10] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning." doi:10.48550/arXiv:2106.11342

[11] H. Abdi and L. J. Williams, "Principal component analysis," *WIREs Comput. Stat.*, vol. 2, pp. 433–459, 2010. doi:10.1002/wics.101

[12] B. Ding, H. Qian, and J. Zhou, "Activation functions and their characteristics in deep neural networks," in *Proc. CCDC'18*, pp. 1836–1841, 2018. doi:CCDC.2018.8407425

[13] S. Y. Lee, *Accelerator physics*, World Scientific Publishing Company, 2018. doi:10.1142/8335