Uncertainty aware machine-learning-based surrogate models for particle accelerators: Study at the Fermilab Booster Accelerator Complex

Malachi Schram[*] and Kishansingh Rajput

Thomas Jefferson National Accelerator Laboratory, Newport News, Virginia 23606, USA

Karthik Somayaji NS and Peng Li

*University of California Department of Electrical and Computer Engineering,
Santa Barbara, California 93106, USA*

Jason St. John

Fermi National Accelerator Laboratory, Batavia, Illinois 60510, USA

Himanshu Sharma

Pacific Northwest National Laboratory, Richland, Washington 99354, USA

Standard deep learning methods, such as Ensemble Models, Bayesian Neural Networks, and Quantile Regression Models provide estimates of prediction uncertainties for data-driven deep learning models. However, they can be limited in their applications due to their heavy memory, inference cost, and ability to properly capture out-of-distribution uncertainties. Additionally, some of these models require post-training calibration that limits their ability to be used for continuous learning applications. In this paper, we present a new approach to provide prediction with calibrated uncertainties that includes out-of-distribution contributions and compare it to standard methods on the Fermi National Accelerator Laboratory (FNAL) Booster accelerator complex.

I. INTRODUCTION

Particle accelerators are complex multisystem machines that include a large number of variables with nonlinear dynamics. To date, accelerator control systems are manually optimized by experts who are guided by physical principles whenever possible. Developing high-dimensional, physics-based models that account for multiple time scales is extremely challenging. Although, there are accelerator beam models with impressive and improving precision [1], building fully comprehensive Monte Carlo-based models of an entire facility is challenging, if not intractable. Data-driven methods, such as deep neural networks (DNNs), are well suited to capture the dynamics of these nonlinear complex systems. These surrogate models can then be used to develop new AI-based control systems provided they can inform the optimization algorithm on how reliable the predictions are.

The recent development of DNNs [2–4] has proven itself useful for complex control problems [5–8]. The use of machine learning for particle accelerator applications has grown in recent years to include, but is not limited to, diagnostics [9–16], anomaly detection/forecasting/classification [17–21], and controls [22–26]. Although these studies have shown some impressive results, the majority do not include any uncertainty quantification (UQ) to complement their predictions. Unfortunately, the use of DNNs for online safety-critical applications remains limited due to issues such as model explainability, in-domain and out-of-domain prediction and uncertainties, and uncertainty calibrations.

In recent years, there has been an increasing amount of effort in estimating uncertainties in DNNs. A prediction's uncertainty can be separated into the model's intrinsic uncertainty (model uncertainty) and the uncertainty caused by the data (data uncertainty). The model uncertainty is typically reducible, within limits, by improving the model architecture and hyperparameters, however, the data uncertainty is irreducible. Additionally, uncertainty estimation originating from out-of-distribution (ODD) samples is critical for a number of applications, such as using DNNs as a proxy to model a dynamical system used for system control and/or optimization. A deep learning method that

provides predictive uncertainty is not sufficient for safe decision making; a deep learning method with *properly calibrated* uncertainty is required.

Recent studies that include data-driven, machine learning (ML)–based surrogate models have started to include UQ in their models, such as developing a UQ-based surrogate model of cyclotron-based model [27], modeling the FNAL booster accelerator complex for a reinforcement learning application [23], and on uncertainty aware anomalies prediction [17]. Additionally, a recently published study compared the use of Bayesian neural networks (BNNs) and ensemble methods for particle accelerator applications [28]. In this paper, we compare three different methods to estimate data-related uncertainties for DNN models as it applies to modeling the FNAL booster accelerator complex. In Sec. II, we briefly describe the Fermilab booster accelerator complex and the data used for training the DNN models in the context of a control optimization problem. In Sec. III, we introduce three methods that estimate uncertainty quantification for DNNs. In Sec. IV, we present the performance of each method for in-distribution and out-of-distribution scenarios. Finally, we conclude with a summary of our results in Sec. V.

II. FERMILAB BOOSTER ACCELERATOR AND COMPLEX

At 15 Hz, the Fermilab Booster rapid-cycling synchrotron accelerates each injected batch of 400 MeV protons to 8 GeV and resets to receive the next injection. See Ref. [22] for a detailed discussion. A central component of the booster cycle is the gradient magnet power supply (GMPS), which provides the synchronously rising and falling electrical current to this circular synchrotron's main bending magnets, tracking the energy (and therefore the proton beam's magnetic stiffness) upward to extraction before returning to the injection state.

The throughput efficiency of the synchrotron is sensitive to unwanted perturbations of the GMPS current, causing the beam's trajectory to deviate from the desired path and scrape on apertures. Such perturbations are understood to be induced by the power supplies of other nearby synchrotrons on their own cycles in the accelerator complex, temperature variations, 60-Hz power line frequency meanders, and other accelerator complex nuances. A proportional-integral-derivative (PID [29,30]) regulator circuit attempts to compensate for these perturbations with cycle-by-cycle adjustments to the minimum of the sinusoidal control signal. Figure 1 shows a schematic overview of the GMPS control environment. See Fig. 2 for a sample distribution of measured errors for the minimum value of the sinusoidally varying magnet current.

Machine learning techniques promise a new avenue for developing more sophisticated control agents with better overall regulation performance, allowing a predictive, anticipatory approach not encompassed by the reactive PID regulation paradigm. As the authors in Ref. [22] point
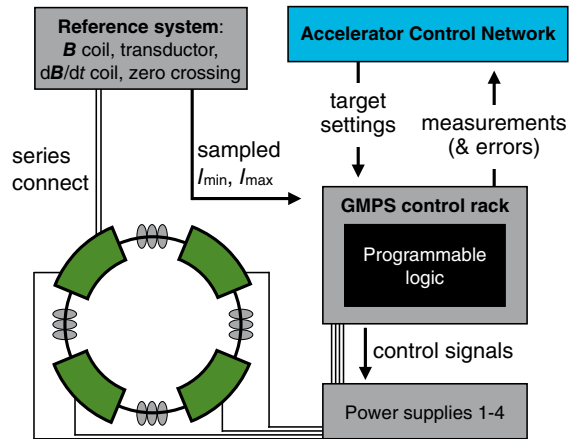

FIG. 1.   Schematic view of the GMPS control environment. The human operator specifies a target program via the Accelerator Control Network that is transmitted to the GMPS control board. The FPGA-based control logic utilizes these settings together with readings from a reference magnet to prescribe a driving signal to the GMPS. The effect of this prescribed signal on the bending magnets is measured by an in-series reference magnet, with sampled readings transmitted back to the GMPS control board. Reference measurements and prescribed signals may be logged and transmitted over the network for later analysis. From Ref. [22].

out, any ML-based GMPS regulator that replaces this PID circuit is required to deliver stable, fast inference times; the data intake, forward inference, and generation of the resulting control signal must always be complete in less than 66 ms.

Reinforcement learning (RL) was selected as the approach to train a ML-based agent to act as the GMPS regulator. RL learns an action policy by training a model using data describing a system's states, actions, and the resulting outcomes. This technique is a natural choice
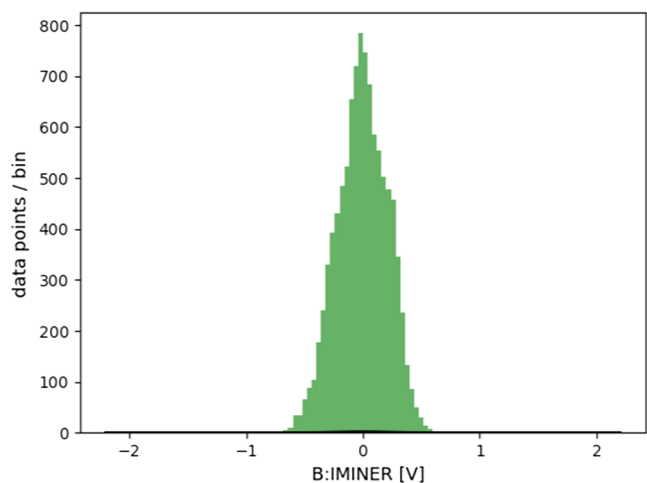

FIG. 2.   Distribution of fractional measured error in the GMPS current at the minimum value of the magnet current (prefix B: indicates booster, I current, MIN miniumum, and ER error), with the non-ML PID regulator discussed in the text. From Ref. [22].

because, once a competent agent is in operation, its real-world performance can be used to provide updated model parameters, tracking the slowly changing dynamics of the accelerator complex. At the outset of RL training, the control agent would be expected to make egregious mistakes and to learn from them. Thus a surrogate model is needed, one which captures the dynamics of the booster GMPS regulator's control environment, where the agent can learn from mistakes without risk to personnel, equipment, or the science program they support around the clock.

Surrogate model training and testing data were taken with the PID regulation circuit in operation. Additionally, a small amount of data were taken with the regulation circuit off or with changes to its coefficients. This choice maximized the available volume of training and testing data, sampling the changing response dynamics of the GMPS regulation while minimizing the impact on accelerator operations. The PID regulator circuit's residual error is typically only 0.1% of the injection current minimum. Without regulation, the fitted minimum of the magnetic field may vary from the set point by as much as a few percent.

Five-time series were used to produce the surrogate model, and we use their names as they are logged by the accelerator control system. (In the accelerator control system's data-logging nomenclature, device parameters with the `B:` prefix are related to the booster, whereas device parameters beginning with `I:` are related to the main injector. "MDAT" denotes the accelerator ("machine") data communication broadcast.) `B:VIMIN` is the compensating recommendation for the minimum value of the offset-sinusoidal GMPS current, issued by the GMPS regulator in order to reduce the magnitude of `B:IMINER`, which is a measure of the residual error. `B:LINFRQ` is the measured offset from the expected 60-Hz line frequency powering the GMPS, in mHz. `I:IB` and `I:MDAT40` provide measurements of the Main Injector bending dipole current at different points in the circuit and through different communication channels. Among all candidate device time series analyzed, main injector's power supplies were shown in [22] to have the highest Granger [31] causality with respect to `B:IMINER`, the minimization control objective of this application, whose typical value distribution is shown in Fig. 2.

Table I briefly summarizes the parameters of interest used in surrogate modeling.

Data were collected nominally at 15 Hz, and due to clock drift among the front-ends taking those samples, the data as logged were then time aligned to a periodic reference signal. Period 0 (June 3, 2019, to July 11, 2019) ended with the annual Summer Shutdown and Maintenance. Period 1 (December 3, 2019, to April 13, 2020) ended when the accelerator operations were suspended in response to the COVID-19 pandemic. More detail on the collection and preparation of the data can be found in the Data Descriptor article [32].

TABLE I. Description of dataset parameters chosen by experts and later validated with a causality study. Here, "MI" means main injector, "MDAT" means accelerator (machine) data communication, and device parameters that begin with B are related to the booster, whereas device parameters that begin with I are related to the main injector (reused with permission from the authors of [22]).

Parameter	Details (Units)
`B:IMINER`	Setting-error discrepancy at injection (A)
`B:LINFRQ`	60-Hz line frequency deviation (mHz)
`B:VIMIN`	Compensated minimum GMPS current (A)
`I:IB`	MI lower bend current (A)
`I:MDAT40`	MDAT measured MI current (A)

III. MACHINE LEARNING METHODS

There has been a lot of research in uncertainty quantification for deep learning models which includes, but is not limited to, BNN [33], deep quantile regression (DQR) [34], and deep Gaussian process approximation (DGPA) models [35–38]. In this paper, we do not consider ensemble methods because these methods require the training of multiple models and multiple inferences to provide an uncertainty estimation, making it computationally expensive, slow, and memory intensive. For this paper, we implemented models for BNN, DQR, and DGPA to better understand their performance for in-distribution and out-of-distribution uncertainty estimation and report the results in the next section. Our specific effort to develop a new DGPA model is most closely related to the recent paper on spectral-normalized neural Gaussian process models [37] for classification. In the following subsections, we discuss the working mechanism of these methods.

A. Bayesian neural network (BNN) model

Monte Carlo (MC) dropout [33] is a commonly used approach to estimate the prediction uncertainty for deep neural network models. The MC-dropout approach has been shown [33] to overcome the computational challenges of estimating uncertainty using Bayesian models. The review by Abdar *et al.* [39] provides comprehensive details on estimating uncertainty in deep learning. The initial application of MC dropout was to overcome overfitting associated with deep neural networks (DNN) while training.

The MC-dropout approach relies on introducing a tunable uncertainty into a network training process by adding a dropout layer that randomly removes nodes to the following layer with a set probability (p) at each forward pass in the training process. While training the DNN with dropout, the units in a layer are randomly dropped and are typically used to avoid overfitting, it can also be used during inference to estimate the uncertainties. In order to combing the aleatoric and epistemic uncertainty into the

BNN model, we implemented the loss function in [40] as shown in Eq. (1)

$$\mathcal{L}_{\text{dropout}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2\sigma(x_i)^2}||\mathbf{y}_i - \mathbf{y}_i^p||^2 + \frac{1}{2}\log\sigma(x_i^2). \quad (1)$$

With this loss function, we can account for both sources of uncertainties and provide a calibrated model. Here y_i^p and y_i are the prediction and measured values, and σ is the model's predicted noise which is determined during training by optimizing the dropout level to minimize Eq. (1).

B. Deep quantile regression (DQR) model

DQR is a method used to estimate the conditional quantiles of a response variable distribution that is more robust against outliers in the response measurements [34]. We define the conditional quantile function for a nonlinear relationship in Eq. (2).

$$Q_y(\tau|x_t) = G_\tau(x_t, w). \quad (2)$$

Here $G_\tau(x_t, w)$ is a nonlinear function that is approximated by a DNN, x is the input feature vector at time t, and τth is the conditional quantile. We develop the DNN model to simultaneously learn predictions based on a set of defined quantiles. For each defined quantile, the prediction y_i^p, and outcome y_i, the regression loss for a τth quantile is given in Eq. (3):

$$\mathcal{L}(\mathbf{y_i^p}, \mathbf{y_i}) = \max[\tau(\mathbf{y}_i - \mathbf{y}_i^p), (\tau - 1)(\mathbf{y}_i - \mathbf{y}_i^p)]. \quad (3)$$

As such, DQR provides a comprehensive statistical model that captures nonlinear relationships by providing conditional quantiles along with the median, in contrast to traditional regression methods. In this paper, we implemented a deep learning model with convolutional layers, similar to the model architecture used for the other methods discussed in this paper, however, the output layers mapping to a dedicated quantile value.

C. Deep Gaussian process approximate (DGPA) model

Gaussian process (GP) models use a kernel function to transform the input data into some higher dimensional representation. The function utilizes the point-to-point distance between samples in the new representation to produce predictions. With this property, it is intrinsically distance aware and can detect OOD samples based on the distance from training distribution. Unfortunately, traditional Gaussian process models (GPs) [41] do not scale well with large datasets because the calculation grows as $\mathcal{O}(N)$, where N is the number of samples. As such, using GP on high dimensional data usually requires dimension reduction, feature extraction, or some other form of approximation. In contrast, DNNs can be very expressive

and readily applied to problems with large datasets and high dimensional feature space. Unfortunately, deterministic DNNs can make predictions on samples that are outside their training dataset which are not guaranteed to be accurate [42] and are unable to identify these predictions as being OOD. As such, we incorporated the desired qualities of the GP into the DNN by adding a fixed size lower rank approximation of the GP with an RBF kernel, $K = \Phi\Phi^T$, at the final layer using random Fourier features (RFF) as defined in [43]. Although this provides an uncertainty estimation that is distance aware, it is not distance preserving because there is no guarantee that distance between the input data is preserved at the hidden layer where the GP approximation is applied. In order to make the DNN distance preserving, we used the *bi-Lipschitz* constraint as part of the training loss function, as shown in Eq. (4).

$$L_1 \times ||x_1 - x_2|| \le ||h_{x_1} - h_{x_2}|| \le L_2 \times ||x_1 - x_2||. \quad (4)$$

Here x is the input feature vector and h_x is the last hidden layer output. To summarize, we implemented the same model architecture used for the other methods in this paper, however, we introduced a GP RBF kernel approximation with 256 RFF and modified the loss function to ensure the distance between input and hidden layers is preserved using soft *bi-Lipschitz* constraint using $L_1 = 0.75$ and $L_2 = 1.25$.

IV. RESULTS

Traditional deep learning models are deterministic and provide a prediction for each input with no measure of confidence associated with the prediction. Providing methods with reliable predictive uncertainty for ML models is critical for real-world applications. As discussed in the previous section, there are number of methods being proposed in the literature to make the DL models uncertainty aware. In this section, we compare the performance of the methods presented in Sec. III for in-distribution and out-of-distribution samples as it applies to the prediction for the FNAL booster accelerator complex. The input samples are independent and identically distributed (IID) as training data (in-distribution), if the underlying system/data produces noisy labels, a DL model will learn to produce the mean. We expect the uncertainty values for such predictions to reliably represent the variance in the underlying data labels. The input samples that are dissimilar to the training samples, called OOD samples, are the most difficult for a DL model to provide the prediction accurately; most of the time, the model will produce inaccurate predictions leading to unreliable results. A prediction without the associated calibrated, distance-aware uncertainty quantification results in a system without contextual information required to select a safe response for a prediction. We require the uncertainty values for OOD predictions to be high indicating low confidence in the prediction.

To compare the results, we trained all the models with similar architecture and datasets. The models consist of three convolutional blocks where each block contained a one-dimensional convolutional layer with 32 filters of size 3 followed by a batch normalization, a max pooling layer, and a dropout layer with a probability of 0.1. The output of the third convolutional block is then flattened to process through a dense layer containing 256 nodes leading to the output layer. The differences between the three models are in how they quantify uncertainty. The DGPA model has a Gaussian approximation layer as the output layer, BNN has a vanilla dense layer as the output layer but the dropouts are kept on during inference, and DQR has multiple dense layers to produce output for different quantiles.

For this study, the raw data were processed using MinMax scaling and restructured so that 15 previous timesteps of the input variables were fed into the models to predict the next timestep forward in the output variables. We divided the data into orthogonal samples, 80% for training and 20% for testing. The samples were further filtered by explicitly excluding contributions when the main injector lower bound current (I:IB) had a value that exceeded 0.995. This filter was used to create a in distribution only training samples that would prevent the models to see the cyclic high amplitude in the predicted variable (B:VIMIN). The relationship between the filtered variable (I:IB) and the predicted variable (B:VIMIN) is shown in Fig. 3.

A. In-distribution results

For in-distribution input samples, deep learning models are expected to have accurate predictions with an uncertainty estimation that is consistent with the training data. To compare the predictive performance along with the uncertainty quantification for these models we use a set of standard metrics including R-square, root mean square error
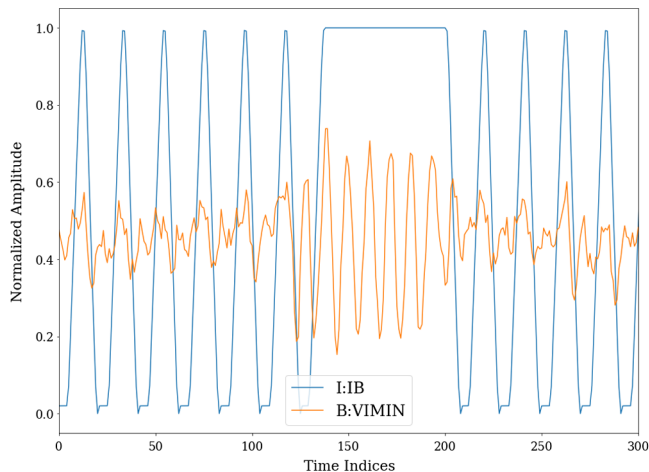
TABLE II.   In-distribution prediction performance for BNN, DQR, and DGPA models.

Model	R^2	RMSE	MACE	RMSCE
BNN	0.835	0.032	0.020	0.022
DQR	0.864	0.029	0.015	0.017
DGPA	0.853	0.030	0.014	0.016

(RMSE) between the ground truth labels and the predictions, mean absolute calibration error (MACE), and root mean square calibration error (RMSCE) from the uncertainty toolkit [44]. Table II shows the values of these metrics for each of the three models before performing any calibration.

All three models have very similar predictive performance in terms of R^2 and RMSE and their uncertainty estimations as shown in Fig. 4 and Table II.

B. Out-of-distribution results

The majority of ML applications assume that the test samples for a model are IID and similar to the training data. Unfortunately, in practice, this assumption does not always hold. When test data draw from an out-of-training distribution sample, the trained model is not guaranteed to produce accurate predictions [42]. Providing an uncertainty estimation consistent with the distance between in-distribution and out-of-distribution data is desirable. We expect the model to produce high uncertainty for inaccurate predictions and lower uncertainty when the predictions are accurate. We expect this relationship between uncertainty
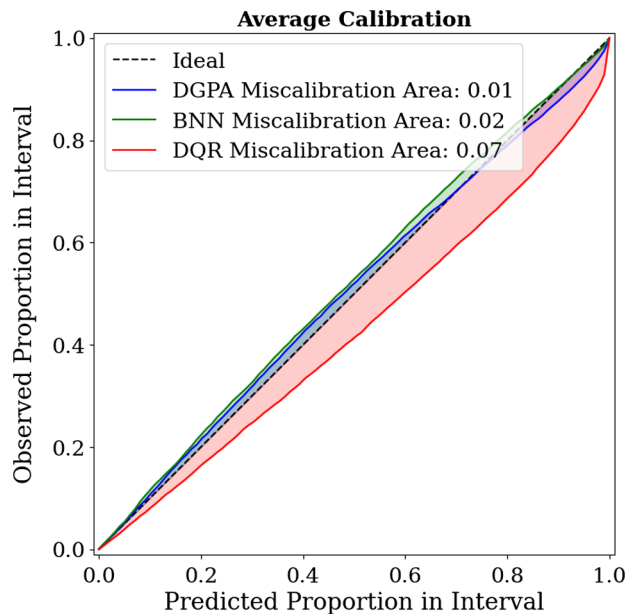

FIG. 4.   Comparison of miscalibration between DGPA, BNN, and DQR for in-distribution data. The shaded area represents the amount of miscalibration in the uncertainty estimation with respect to the true labels.

FIG. 3.   Comparison between the main injector lower bend current and the compensated minimum GMPS current.
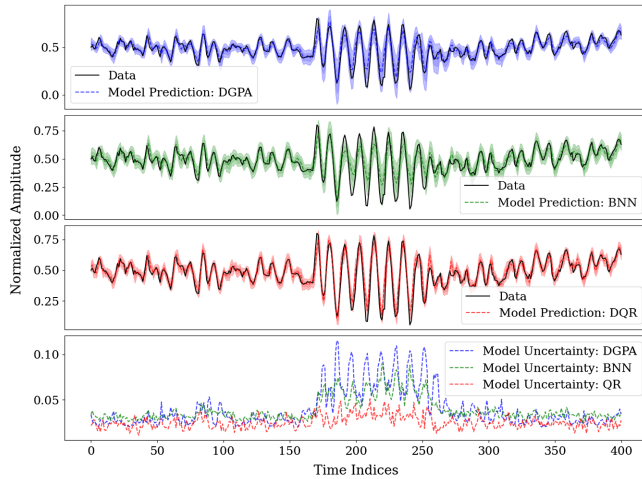
FIG. 5. Side-by-side comparison of the predictive performance as well as uncertainty quantification results for in-distribution and out-of-distribution samples for DGPA, BNN, and DQR, respectively. The middle region with the high frequency component on the time series represents OOD samples while the initial and tail-end regions represent in-distribution data samples.

and error even for the OOD samples. To evaluate the ability of each model's technique to estimate the OOD uncertainty, we created two scenarios detailed below.

1. Scenario 1

For the first scenario, we trained the models with the in-distribution samples and evaluated their performance using the full test sample. The top three plots in Fig. 5 show each model's predictions and uncertainties using data that include the OOD samples. As can be seen, the predictions from all three models degrades in the OOD region, which is expected. The respective uncertainty values are expected to correlate to the deviation from the ground truth labels. We used the same metrics from the in-distribution study to quantify how close each model is to the ground truth and present the results in Table III. These results show how close the models are to the ground truth and are not used as a calibration study. The R^2 and RMSE is similar for all three models, however, the uncertainty estimation varies. As shown in Fig. 6, all models underestimate the uncertainty, however, both the uncertainty predictions from the BNN and DQR are significantly lower than those from the DGPA.

The DPGA has approximately 3× smaller MACE and RMSCE than the DQR and BNN which indicated that the

TABLE III. Out-of-distribution prediction performance for BNN, DQR, and DGPA models.

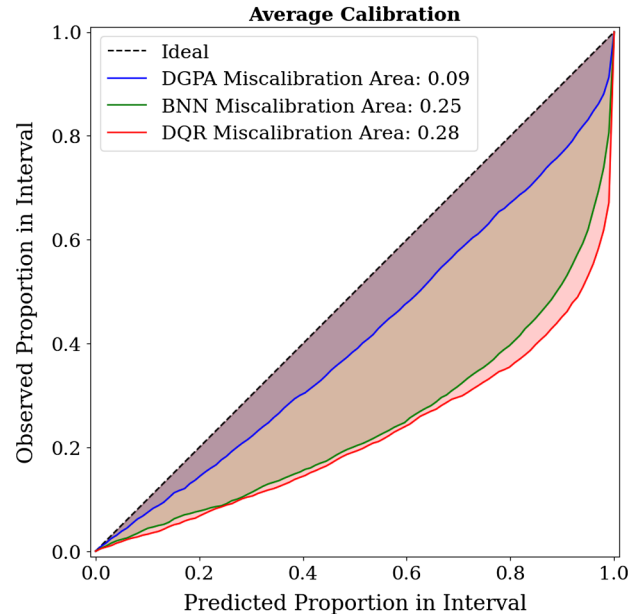| Model	R^2	RMSE	MACE	RMSCE
BNN	0.694	0.104	0.252	0.281
DQR	0.836	0.076	0.275	0.309
DGPA	0.784	0.088	0.091	0.100

FIG. 6. Comparison of miscalibration between DGPA, BNN, and DQR for OOD data. The shaded area represents the amount of miscalibration in the uncertainty estimation with respect to the true labels.

overall uncertainty estimation of the predictions are closer to the ground truth.

2. Scenario 2

Figure 7 shows the second scenario where we monotonically increase one of the key input variables VIMIN until the data samples enter into a region of feature space that is out-of-training distribution. VIMIN is chosen because it is one of the key variables affecting our target.
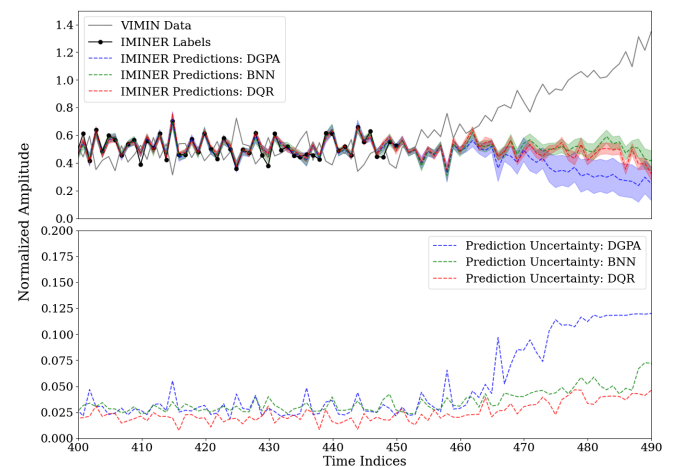

FIG. 7. Comparison of the predictive performance with uncertainty quantification between DGPA, BNN, and DQR, respectively for a manually induced OOD. We manually increased the VIMIN input variable at time index 450 to induce the OOD scenario.

These data samples are then fed into the models for inference and uncertainty quantification. Since these data samples are not within the training distribution they are seen as OOD by the models. Similar to the above-discussed OOD results, the uncertainty values from the BNN and DQR are underestimated as compared to DGPA when the data samples go into OOD region. From Figs. 5 and 6 as well as the Fig. 7, it is clear that for OOD samples, DGPA produces more accurate uncertainty estimates as compared to BNN and DQR, both of which underestimate the uncertainty values.

V. SUMMARY AND CONCLUSIONS

In this paper, we compared three different DNN techniques that estimate the prediction uncertainty. We present the DGPA technique as a new approach to estimating prediction uncertainties; DGPA is self-calibrated and includes an awareness of out-of-distribution uncertainty. The results show that all models provide similar performance for the predictions for in-distribution, however, the DGPA provides uncertainty estimations that are closer to the ground truth for the out-of-distribution Scenario 1. Although we cannot quantify the explicit expected uncertainty for Scenario 2, we can see that the uncertainty estimation from the DPGA is larger than the other two methods where the DQR provides a very small uncertainty estimation. In conclusion, the results from this study on the FNAL booster accelerator complex data suggest that the DGPA model provides the best single inference calibrated data-driven ML-based model for in- and out-of-distribution uncertainty estimation for all scenarios. This was achieved by using a fixed size GP RBF kernel approximation and applying the *bi-Lipschitz* constraint in the loss function. Additional research should be conducted to better understand the trade-off from kernel approximation size and how this can be applied to real-time systems where the hardware could be a constraint.

ACKNOWLEDGMENTS

[1] J. Amundson, A. Macridin, and P. Spentzouris, High performance computing modeling advances accelerator science for high energy physics, Comput. Sci. Eng. **16,** 32 (2014).

[2] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521,** 436 (2015).

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016), http://www.deeplearningbook.org.

[4] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91,** 045002 (2019).

[5] DeepMind, Safety-first AI for autonomous data centre cooling and industrial control, https://deepmind.com/blog/safety-first-ai-autonomous-data-centre-cooling-and-industrial-control/ (2018).

[6] A. Davies, The WIRED guide to self-driving cars, https://www.wired.com/story/guide-self-driving-cars/ (2018).

[7] Jennifer Langston, How AI is building better gas stations and transforming Shell's global energy business, https://blogs.microsoft.com/ai/shell-iot-ai-safety-intelligent-tools/ (2018).

[8] Will Knight, This factory robot learns a new job overnight, https://www.technologyreview.com/s/601045/this-factory-robot-learns-a-new-job-overnight/ (2018).

[9] 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).

[10] A. Sanchez-Gonzalez *et al.*, Machine learning applied to single-shot x-ray diagnostics in an XFEL, Nat. Commun. **8,** 15461 (2017).

[11] M. Wielgosz, A. Skoczeń, and M. Mertik, Using LSTM recurrent neural networks for monitoring the LHC superconducting magnets, Nucl. Instrum. Methods Phys. Res., Sect. A **867,** 40 (2017).

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

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

[14] A. Scheinker, C. Emma, A. L. Edelen, and S. Gessner, Advanced control methods for particle accelerators (ACM4PA) 2019 workshop report, Technical Report No. LA-UR-19-32526, 2020, arXiv:2001.05461.

[15] 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).

[16] S. Li, M. Zacharias, J. Snuverink, J. Coello de Portugal, F. Perez-Cruz, D. Reggiani, and A. Adelmann, A novel approach for classification and forecasting of time series in particle accelerators, Information **12,** 121 (2021).

[17] W. Blokland, P. Ramuhalli, C. Peters, Y. Yucesan, A. Zhukov, M. Schram, K. Rajput, and T. Jeske, Uncertainty aware anomaly detection to predict errant beam pulses in the SNS accelerator, arXiv:2110.12006.

[18] M. Rescic, R. Seviour, and W. Blokland, Predicting particle accelerator failures using binary classifiers, Nucl. Instrum. Methods Phys. Res., Sect. A **955,** 163240 (2020).

[19] M. Reščič, R. Seviour, and W. Blokland, Improvements of pre-emptive identification of particle accelerator failures using binary classifiers and dimensionality reduction, Nucl. Instrum. Methods Phys. Res., Sect. A **1025,** 166064 (2022).

[20] C. Tennant, A. Carpenter, T. Powers, A. Shabalina Solopova, L. Vidyaratne, and K. Iftekharuddin, Super-conducting radio-frequency cavity fault classification using machine learning at Jefferson laboratory, Phys. Rev. Accel. Beams **23,** 114601 (2020).

[21] T. Powers and A. Solopova, CEBAF C100 fault classification based on time domain RF signals, in *Proceedings of 19th International Conference on RF Superconductivity, SRF 2019, Dresden, Germany* (JACoW, Geneva, Switzerland, 2019), WETEB3.

[22] J. St. John, C. Herwig, D. Kafkes, J. Mitrevski, W. A. Pellico, G. N. Perdue, A. Quintero-Parra, B. A. Schupbach, K. Seiya, N. Tran, M. Schram, J. M. Duarte, Y. Huang, and R. Keller, Real-time artificial intelligence for accelerator control: A study at the Fermilab booster, Phys. Rev. Accel. Beams **24,** 104601 (2021).

[23] D. Kafkes and M. Schram, Developing robust digital twins and reinforcement learning for accelerator control systems at the Fermilab Booster, in *Proceedings of the 12th International Particle Accelerator Conference, IPAC-2021, Campinas, SP, Brazil* (JACoW, Geneva, Switzerland, 2021), arXiv:2105.12847.

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

[25] S. Hirlaender and N. Bruchon, Model-free and Bayesian ensembling model-based deep reinforcement learning for particle accelerator control demonstrated on the Fermi FEL, arXiv:2012.09737.

[26] A. Scheinker, D. Filippetto, and F. Cropp, 6D phase space diagnostics based on adaptively tuned physics-informed generative convolutional neural networks, J. Phys. Conf. Ser. **2420,** 012068 (2023).

[27] A. Adelmann, On nonintrusive uncertainty quantification and surrogate model construction in particle accelerator modeling, SIAM/ASA J. Uncertainty Quantif. **7,** 383 (2019).

[28] A. A. Mishra, A. Edelen, A. Hanuka, and C. Mayes, Uncertainty quantification for deep learning in particle accelerator applications, Phys. Rev. Accel. Beams **24,** 114601 (2021).

[29] N. Minorsky., Directional stability of automatically steered bodies, J. Am. Soc. Nav. Eng. **34,** 280 (1922).

[30] J. G. Ziegler and N. B. Nichols, Optimum settings for automatic controllers, J. Dyn. Syst., Meas., Control **115,** 220 (1993).

[31] C. W. J. Granger, Investigating causal relations by econometric models and cross-spectral methods, Econometrica **37,** 424 (1969).

[32] D. Kafkes and J. St. John, BOOSTR: A dataset for accelerator control systems, Data **6,** 42 (2021).

[33] Y. Gal and Z. Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, in *Proceedings of the 33rd International Conference on Machine learning, New York, NY* (2016), Vol. 48, pp. 1050–1059.

[34] R. Koenker, *Quantile Regression*, Econometric Society Monographs (Cambridge University Press, Cambridge, England, 2005).

[35] A. Rahimi and B. Recht, Random features for large-scale kernel machines, in *Advances in Neural Information Processing Systems*, edited by J. Platt, D. Koller, Y. Singer, and S. Roweis (Curran Associates, Inc., Red Hook, NY, 2007), Vol. 20.

[36] C. E. Rasmussen, Gaussian processes in machine learning, in *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2003, Tübingen, Germany, August 2003, Revised Lectures*, edited by O. Bousquet, U. von Luxburg, and G. Rätsch (Springer, Berlin, Heidelberg, 2004), pp. 63–71.

[37] J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan, Simple and principled uncertainty estimation with deterministic deep learning via distance awareness, in *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Curran Associates, Inc., Red Hook, NY, 2020), Vol. 33, pp. 7498–7512.

[38] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, Deep kernel learning, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain* (PMLR, 2015), Vol. 51.

[39] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya *et al.*, A review of uncertainty quantification in deep learning: Techniques applications and challenges, Inf. Fusion **76,** 243 (2021).

[40] A. Kendall and Y. Gal, What uncertainties do we need in Bayesian deep learning for computer vision?, in *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Long Beach, CA* (Curran Associates Inc., Red Hook, NY, 2017), pp. 5580–5590.

[41] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, Adaptive Computation and Machine Learning (MIT Press, Cambridge, MA, 2006), pp. I–XVIII, 1–248.

[42] C. Cortes, M. Mohri, and A. Rostamizadeh, Learning non-linear combinations of kernels, in *Advances in Neural Information Processing Systems*, edited by Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (Curran Associates, Inc., Red Hook, NY, 2009), Vol. 22.

[43] J. Z. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax-Weiss, and B. Lakshminarayanan, Simple and principled uncertainty estimation with deterministic deep learning via distance awareness (2020).

[44] Y. Chung, I. Char, H. Guo, J. Schneider, and W. Neiswanger, Uncertainty toolbox: an open-source library for assessing, visualizing, and improving uncertainty quantification, *Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning, Hawaii, USA* (2021), arXiv:2109.10254.