BNL-225437-2024-TECH
C-A/AP/705

EBIS BEAM INTENSITY ONLINE OPTIMIZATION WITH GPTUNE AND OFFLINE ANALYSIS WITH XGBOOST

X. Gu

February 2024

Collider Accelerator Department
Brookhaven National Laboratory

U.S. Department of Energy
USDOE Office of Science (SC), Nuclear Physics (NP)

DISCLAIMER

EBIS BEAM INTENSITY ONLINE OPTIMIZATION WITH GPTUNE AND OFFLINE ANALYSIS WITH XGBOOST

Xiaofeng Gu[*], Takeshi Kanesue, Masahiro Okamura,

Collider Accelerator Department

Brookhaven National Lab

UPTON, NY 11973

xgu@bnl.gov

Yue Hao

Michigan State University

East Lansing, MI 48824

Ji Qiang, Xiaoye S. Li, Yang Liu

Lawrence Berkeley National Laboratory

Berkeley, CA 94720

February 28, 2024

ABSTRACT

The application of machine learning techniques to accelerator research has led to significant breakthroughs in optimization strategies. This paper presents a pioneering study using a novel machine learning algorithm, GPTune, to optimize beam intensity by adjusting parameters in the EBIS injection and extraction beam lines. Our research demonstrates substantial improvements, achieving a remarkable 22% and 70% increase in beam intensity at two separate measurement locations.

Furthermore, the XGBoost package is employed for offline data analysis to evaluate the individual impact of each parameter on beam intensity. This analysis provides valuable insights to guide us towards optimal parameter settings, paving the way for further beam intensity enhancements.

***K*eywords**

[*] xgu@bnl.gov

EBIS beam line, Machine Learning, GPTune, XGBoost, Partical Dependence Plot

1 Introduction

Brookhaven National Lab has successfully developed the Electron Beam Ion Source (EBIS), a compact and versatile heavy ion accelerator. EBIS serves as the pre-injector system for both the Relativistic Heavy Ion Collider (RHIC) and NASA Space Radiation Laboratory (NSRL). It utilizes an electron beam ionization source followed by a radiofrequency quadrupole linac and an interdigital-H linac.

One of EBIS's key advantages is its ability to produce short, high-intensity pulses of ions. These pulses are ideally suited for single or few-turn injection into synchrotrons like RHIC, where ions need to be injected quickly and efficiently. Additionally, EBIS offers significant operational benefits compared to traditional injector systems. Its lower energy consumption translates to reduced operating costs, while its ability to quickly switch between different ion beams within one second enhances operational flexibility. This, in turn, allows for the simultaneous feeding of beams of different ions to RHIC and NSRL, enabling quick transitions between various ion species for diverse research programs.

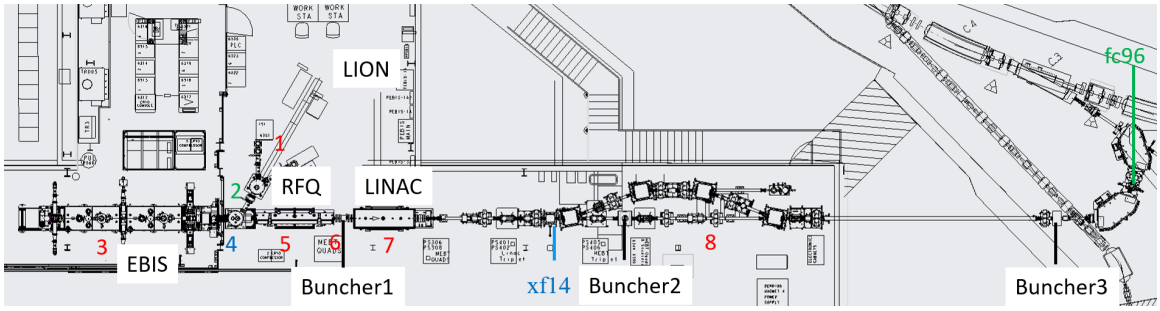

Fig. 1: The Layout of EBIS system

Fig. 1 shows the layout of the EBIS system. The system comprises several beamline sections, as listed in Table 1 and further detailed in Fig. 1. These sections are: LION (Laser Ion Source), EBIS Injection Line, EBIS, EBIS Extraction Line, RFQ, MEBT, Linac and HEBT. Each beamline section has numerous parameters that can influence beam performance. Table 1 shows a selection of parameters which could have more effects on beam properties. These operational parameters can affect beam performance simultaneously, making it challenging to isolate their individual effects.

Furthermore, as is evident in Fig. 2, the beam intensity signal exhibits significant noise at xf14 (current transformer). Optimizing these parameters individually based solely on the intensity signal would be a time-consuming task. This is due to the instability of the beam intensity when the system is not optimized, necessitating the collection of multiple data cycles to obtain representative intensity values. This is particularly true after certain parameters have already reached their optimal values.

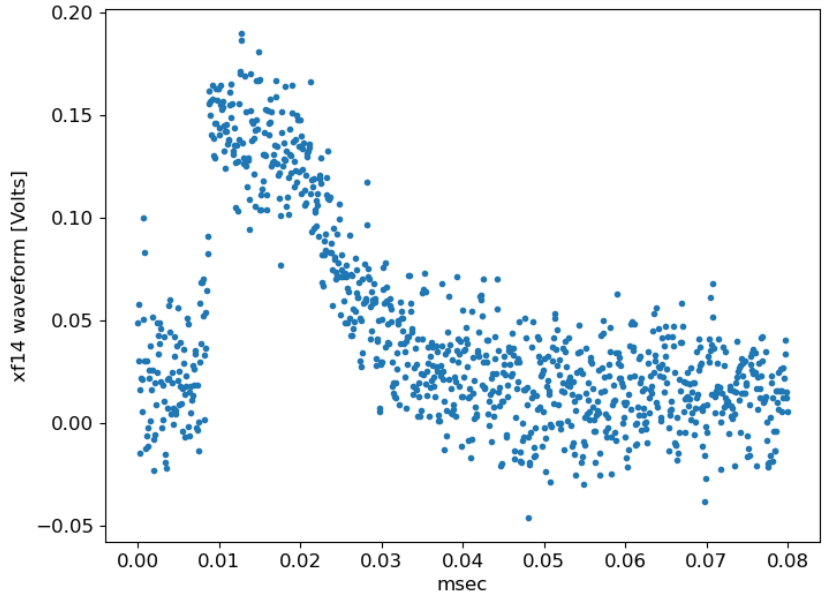

Fig. 2: The xf14 waveform for beam intensity measurement.

To address the aforementioned issues and optimize EBIS beam intensity online, a machine learning algorithm, GPTune, was implemented on the EBIS beam injection and extraction lines at the conclusion of the RHIC 2023 run.

Concurrently, following the online optimization with GPTune, we applied a machine-learning algorithm, XGBoost [1, 2], to the same operational data for offline analysis. Upon acquiring the data for these parameters and constructing a model using XGBoost, the beam intensity as a function of individual parameters can be plotted separately for distinct parameters. This enables the identification of optimized operational parameters.

Table 1: A selection of parameters which could have more effects on beam properties

sections	**Item**	**Description**	Parameters Number
1	LION	Laser Ion Source	4
2	Injection Line	Electron Beam Ion Source Injection Line	9
3	EBIS	Electron Beam Ion Source	$30-40$
4	EBIS Extraction Line	Electron Beam Ion Source Extraction Line	10
5	RFQ	Radiofrequency Quadrupole Linac	2
6	MEBT	Medium Energy Beam Transport	4
7	Linac	Linear Accelerator	4
8	HEBT	High Energy Beam Transport	17
9	Buncher	High Energy Beam Transport	6

2 GPTune and XGBoost

GPTune is a Python optimization package developed by Lawrence Berkeley National Laboratory (LBNL) for online beam intensity optimization in particle accelerators. It constructs a multi-task Gaussian Process (GP) model for the objective function by optimizing its hyperparameters. Then, it identifies the optimum of an acquisition function applied to the model and evaluates it. GPTune allows for both exploitation and exploration for model evaluation: conducting a local search within promising regions for exploitation or undertaking a global search for new promising regions for exploration.

GPTune has several noteworthy features, namely:

- Dynamic Process Management: Executes applications with varying core counts and GPUs efficiently.
- Coarse Performance Models: Enhances the surrogate model by incorporating prior knowledge of the system.
- Multi-objective Tuning: Enables optimization of a hybrid combination of computation, memory, and communication metrics.
- Multi-fidelity Tuning: Optimizes the utilization of a limited resource budget by incorporating data of different fidelities.
- Checkpoint and Historical Data Support: Supports checkpoints and reuse of historical performance databases for efficient optimization.
- Signal Minimization for Maximum Output: Optimizes the objective function by minimizing the signal to achieve maximum output.

Extreme Gradient Boosting (XGBoost), a machine learning algorithm developed in Python, was utilized for regression problems in this study. XGBoost constructs a "black-box" model relating beam intensity to all machine parameters. It is a supervised learning algorithm that employs gradient descent to optimize its loss function for both classification and regression tasks.

XGBoost attempts to establish the most predictive relationship or model using historical logged data. The algorithm evaluates its performance by minimizing a loss function, which quantifies the discrepancy between predicted and actual outputs. Mean squared error (MSE) is a common loss function for regression algorithms.

XGBoost stands out as a versatile and efficient tool for tackling non-linear regression tasks. Its strengths lie in its robustness, enabled by features like regularization, tree pruning, and the ability to handle missing data. XGBoost's popularity hinges on its high performance and effectiveness across diverse machine learning applications. Its strength lies in its ability to excel in scenarios where the relationship between input features and the target variable is complex and non-linear.

Its success is demonstrated by its use in numerous winning teams of machine learning competitions, including the CERN LHCb experiment Flavour of Physics [3]. XGBoost gained significant recognition after its success in the Higgs Machine Learning Challenge [4], resulting in the "HEP meets ML award" being bestowed upon its creators [5].

Understanding the inner workings of complex models like XGBoost is crucial for drawing meaningful insights. Here, we explore two techniques for interpreting XGBoost predictions, they are Partial Dependence Plots (PDPs) [7] [8] and Shapley Additive exPlanations (SHAP) [9]

After training the model, PDPs visualize the marginal effects of one or two input parameters on the predicted outcome. To address potential correlation issues inherent in PDPs, Shapley values offer an alternative interpretation method. Similar to PDPs, SHAP values measure the marginal contribution of an individual input parameter to the model prediction.

By employing these techniques, we gain valuable insights into the relationships captured by XGBoost models, enabling us to draw informed conclusions, make better predictions and find better operation parameters setting.

3 GPTune Optimization

3.1 Experimental Setup

During the optimization, the ion beam species was Si^{+11}. The beam injection system (Booster-AGS) has a supercycle time of 6.6 seconds. Although within a single supercycle, up to 12 pulses of EBIS beam could be injected into the Booster-AGS ring, while only one pulse was injected into the Booster-NSRL target room. Only one pulse of ion beam was then used for subsequent processes.

Meanwhile, some power supplies require two supercycles for their outputs to settle. After the power supplies stabilize, the script takes four measurements for averaging, each separated by the supercycle time, to obtain more robust statistics.

Table 2: Parameters for GPTune Optimization

	Parameters	**Injection [fc96]**	**Extraction [xf14]**
1	IonLens20-40kV	Yes	Yes
2	DeflPlatBias	Yes	Yes
3	16PoleX	Yes	Yes
4	16PoleY	Yes	Yes
5	Gridded_Lens	Yes	Yes
6	Horiz_Bend_Defl	Yes	Yes
7	Inter_Vert_Defl	Yes	Yes
8	Inter_Vert_Defl_Lower	Yes	Yes
9	Horiz_Sphere_Bend	Yes	No
10	RFQ_Horiz_Bend	No	Yes
11	LEBT_Solenoid	No	Yes
Total Variables		9	10

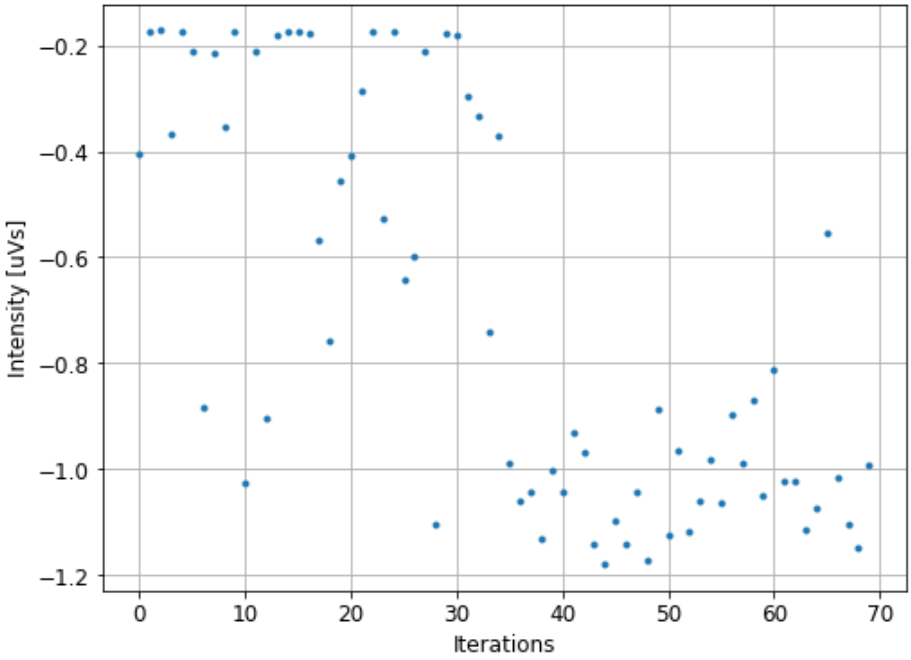

Fig. 3: Injection Line Optimization using GPTune

The Farady-cup FC96 measurement was used for injection optimization, employing 9 control parameters with 70 iterations. Similarly, the current transformer XF14 measurement was used for extraction optimization, utilizing 10 control parameters with 60 iterations. The conversion factors from raw integral to beam charge vary depending on the location. At xf14, $1\mu Vs$ corresponds to $1.43nC$, while at fc96, the same signal translates to $0.109nC$.

3.2 Injection Line Optimization

During the optimization, the injection and extraction beam lines were initially optimized separately. When optimizing the injection line, the current transformer fc96 was used to measure the ion beam intensity. The corresponding control parameters are listed in Table 2.

Figure 3 shows the progress of the injection line optimization using GPTune. The horizontal axis represents the iteration number of the GPTune script, while the vertical axis represents the normalized and averaged signal of fc96. As mentioned earlier, GPTune aims to minimize the signal to achieve maximum output; therefore, a more negative value indicates a better optimization result. As Figure 3 demonstrates, GPTune finds a significantly improved result after approximately 45 iterations for the 9 variable parameters.

Figure 4 displays the fc96 beam intensity signal during the optimization process (between the two vertical green lines). Following the optimization, the average beam intensity increased from 11.5 uVs to 14.0 uVs, representing an improvement of approximately 22%.

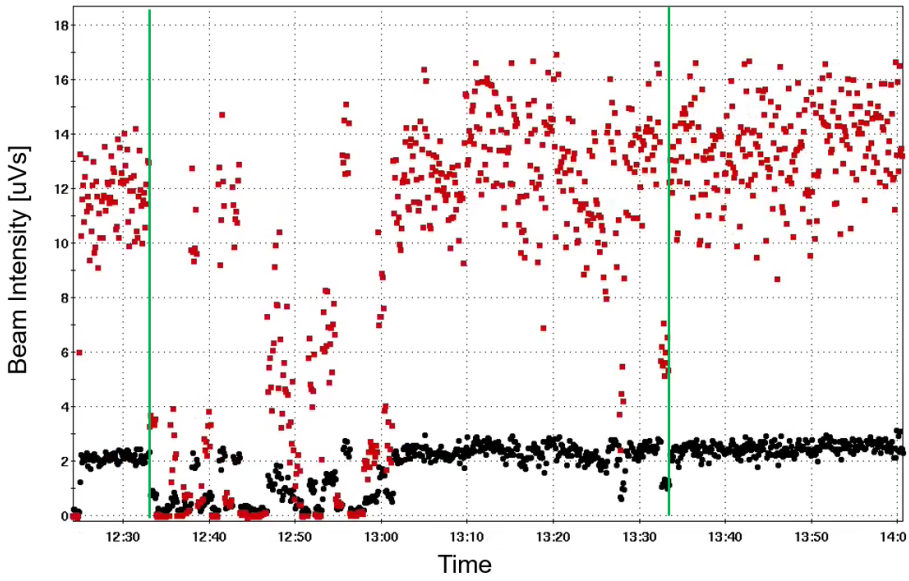

Fig. 4: Injection Line Optimization Result. Red dots represent the fc96 signals, and black dots represent the xf14 signal.

3.3 Extraction Line Optimization

Fig. 5 depicts the progression of injection line optimization using GPTune. It's evident that GPTune identifies a superior outcome after approximately 45 iterations with 10 control variable parameters.

Fig. 6 illustrates the xf14 beam intensity signal during the optimization process (delineated by the two vertical green lines). Post-optimization, the average beam intensity exhibited a substantial increase from 1.4 uVs to 2.0 uVs, translating to a remarkable 43% improvement.

It's crucial to note that, while xf14 and fc96 share the same unit, direct comparison of their results is not feasible due to the utilization of different normalization factors during post-measurement data processing.

3.4 Injection and Extraction Combined Optimization Settings

In the previous sections, we optimized the injection and extraction lines separately. Their power supply settings were also saved individually. To evaluate their combined contribution to beam intensity, we compared the intensity under three settings:

- Inj + Ext: Power supplies optimized for both injection and extraction lines
- Ext: Power supplies optimized only for extraction with original injection settings
- Original: Original settings without any optimization

The optimization results for above three different settings are shown in Fig. 7 and Table 3. in Fig. 7, the red dots and the black dots are the measurement from fc96 and xf14 respetively. The green and cyan number are the average beam intensity with their deivation for fc96 and xf14 measurement.

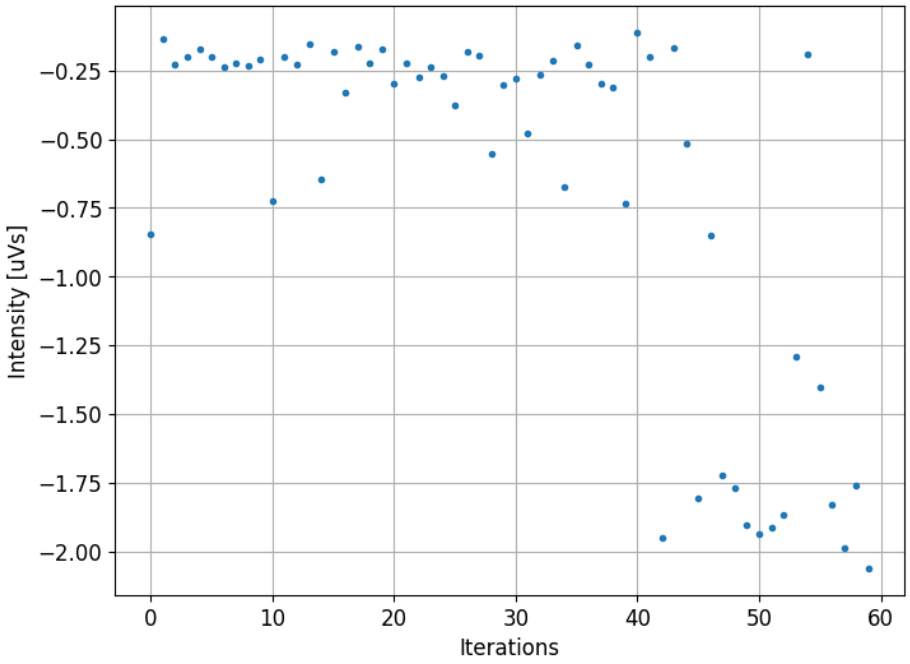

Fig. 5: Extraction Line Optimization using GPTune

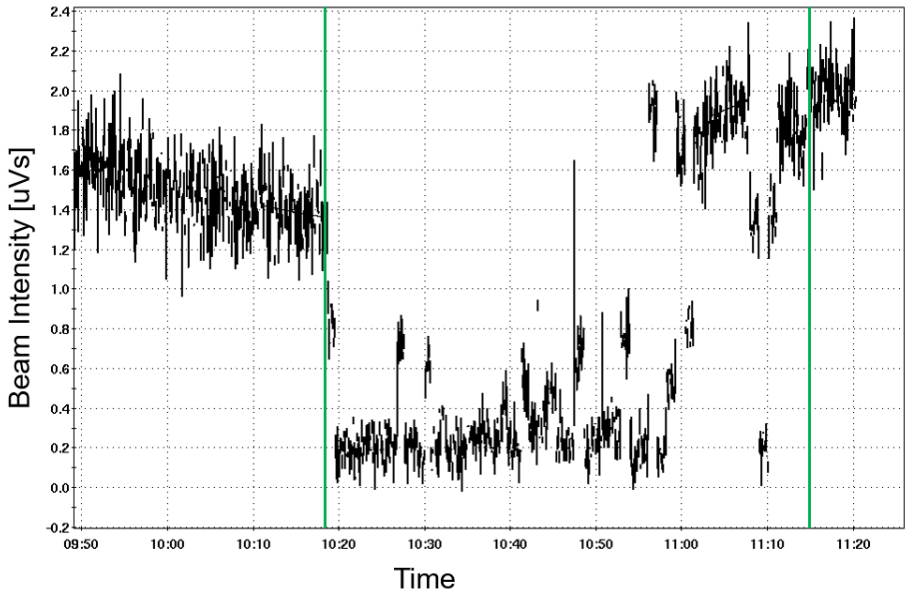

Fig. 6: Extraction Line Optimization result.

The optimization results for the three different settings are shown in Figure 7 and Table 3. In Figure 7, the red and black dots represent the measurements from fc96 and xf14, respectively. The green and cyan numbers represent the average beam intensity with their deviations for the fc96 and xf14 measurements.

From the table, we observe significant intensity gains:

- xf14 measurement: 42% for extraction-only optimization and $68 - 71\%$ for combined optimization.

- fc96 measurement: 8.4% for extraction-only optimization and $22-24\%$ for combined optimization.

Fig. 7 and Table 3 reveal a characteristic of the beam intensity signal: substantial noise. The standard deviation is 10%, and the peak-to-peak deviation is 15%. This demonstrates GPTune's outstanding capability to handle noisy signals, a valuable feature for many experimental settings.

Fig. 7 and Table 3 also show that the total beam intensity gain at xf14 ($68-71\%$) is about three times that at fc96 ($22-24\%$). One reason for this result is that the xf14 measures all charge states of Si, while fc96 only measured the charge state of Si^{+14}. Another possible reason is that the beam line after the EBIS extraction lines does not have an optimal operation setting. This suggests significant potential for further gain at fc96 through other beam lines optimization.

Further improvements to beam intensity are likely achievable through optimizing other beamline components, including the RFQ, MEBT, Linac, and HEBT sections.

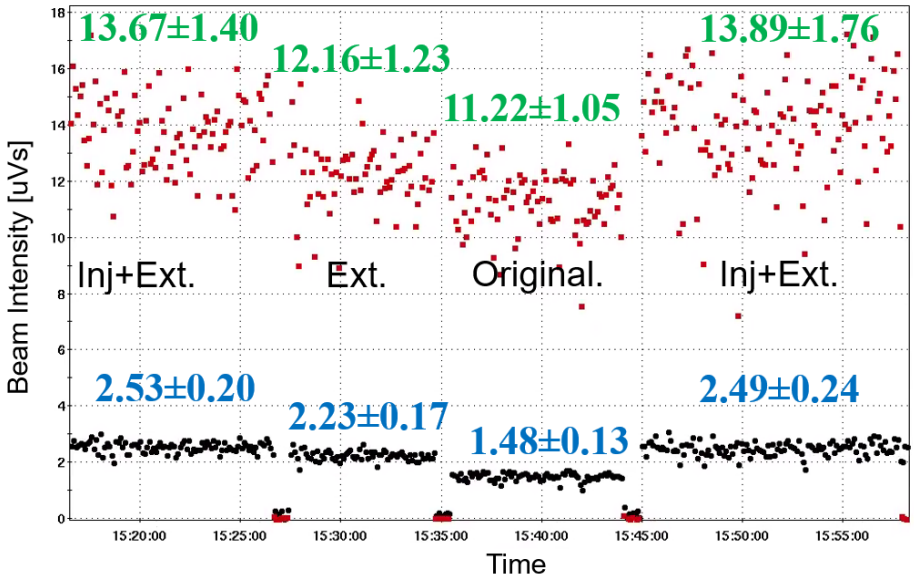

Fig. 7: Optimization results with different settings: Inj + Ext, Ext only, and Original.

Table 3: Beam Intensity Optimization using different settings

Device	Orignal	**Ext**	**Gain from Ext**	**Ext+Inj**	**Gain from Ext+Inj**
xf14 [uVs]	1.48±0.13	2.23±0.17	42 %	2.53±0.20/2.49±0.24	68-71 %
fc96[uVs]	11.22±1.05	12.16±1.23	8.4 %	13.67±1.40/13.89±1.76	22-24 %

3.5 Injection and Extraction Line Optimization Simultaneously

In the previous three sections, the beam intensity has been optimized either with the EBIS injection line (10 parameters) and extraction line (9 parameters) separately, or using their combined settings. This section presents the optimization results with 19 parameters obtained by simultaneously optimizing both the injection

and extraction beam line power supplies. This optimization builds upon the previous two optimizations by retaining all previously optimized power supply settings.

Figure 8 shows the GPTune optimization progress for the injection line. The optimization consisted of two stages: an initial run with 80 iterations followed by an additional 10 iterations enabled by the "Checkpoint and Historical Data Support" feature of GPTune, resulting in a total of 90 iterations. The fc96 intensity signal was used as the objective function during the optimization process.

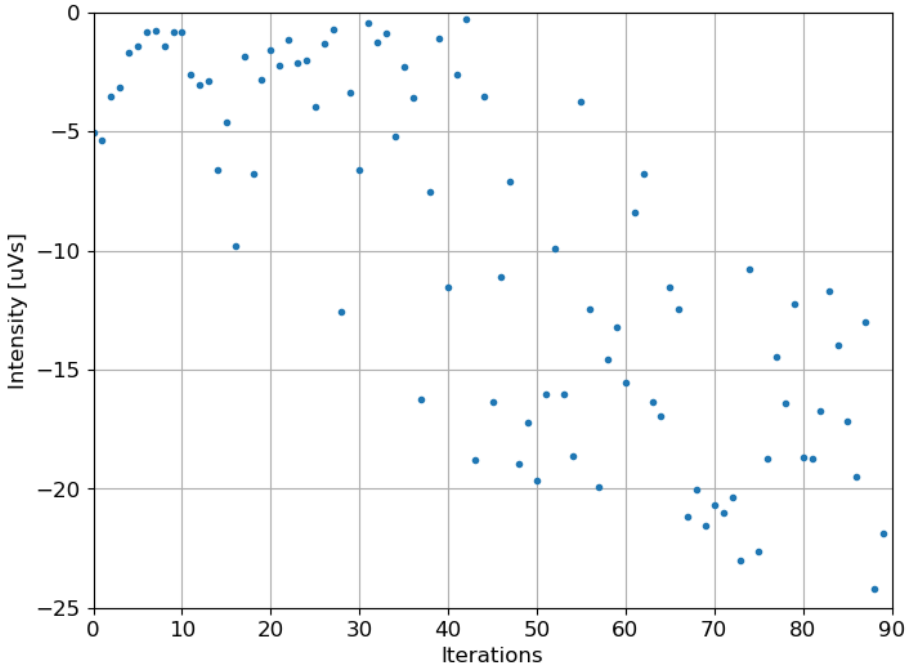

Fig. 8: Optimization with both injection and extraction line simultaneously (19 optimization parameters)

Fig. 9 shows the result with both injection and extraction lines at the same time. The original average signal without any further optimization is about 22 uV; after 80 iterations, the averaged signal changed to 23 uV; after 90 iterations, it improved to 23.56 uV. Therefore, there is an additional 7.1% beam intensity improvement.

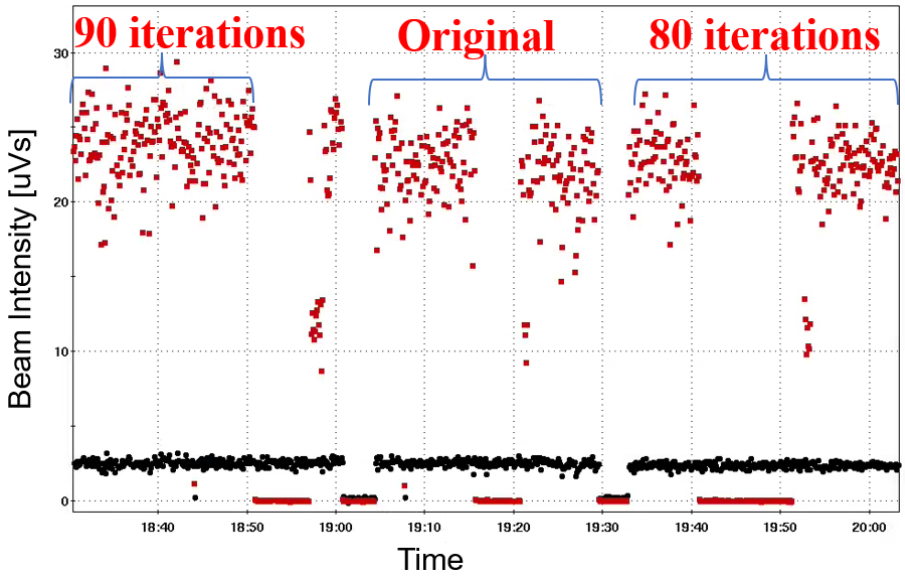

Fig. 9: Optimization Result with both injection and extraction line

4 XGBoost Offline Optimization

4.1 XGBoost Modelling

Following online GPTune optimization, XGBoost was employed for offline data analysis to determine the beam intensity as a function of individual control parameters.

Regression models for the EBIS injection and extraction beam lines were constructed offline using the XGBoost Python package. Figure 10 depicts the constructed XGBoost model predictions alongside their respective test datasets for the injection beam line (left plot) and extraction beam line (right plot). The models achieved scores of 79% and 80% for the injection and extraction beam lines, respectively, demonstrating a strong agreement between model predictions and test data.

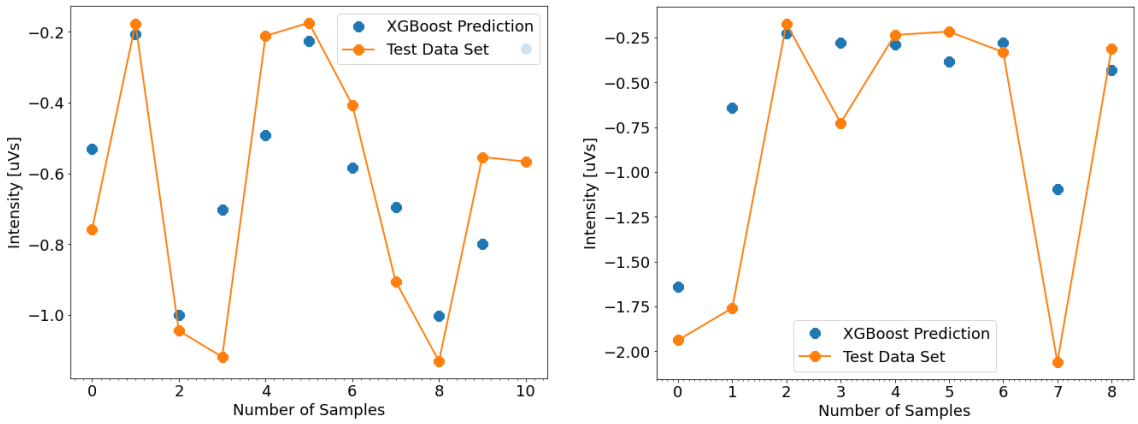

Fig. 10: XGBoost model predictions for the EBIS injection beam line (left plot) and extraction beam line (right plot)

To interpret the constructed regression models, partial dependence plots (PDPs) were employed. PDPs visualize the marginal effects of one or two input parameters on the model's predictions [7, 8].

To address potential correlation issues that could arise in PDPs, Shapley values (SHAP, Shapley Additive exPlanation) were also utilized to interpret the effects of individual input parameters on the XGBoost model predictions [9]. Shapley values calculate the marginal contribution of each input parameter, similar to PDPs, but offer a more robust approach to understanding feature importance in the presence of correlated features.

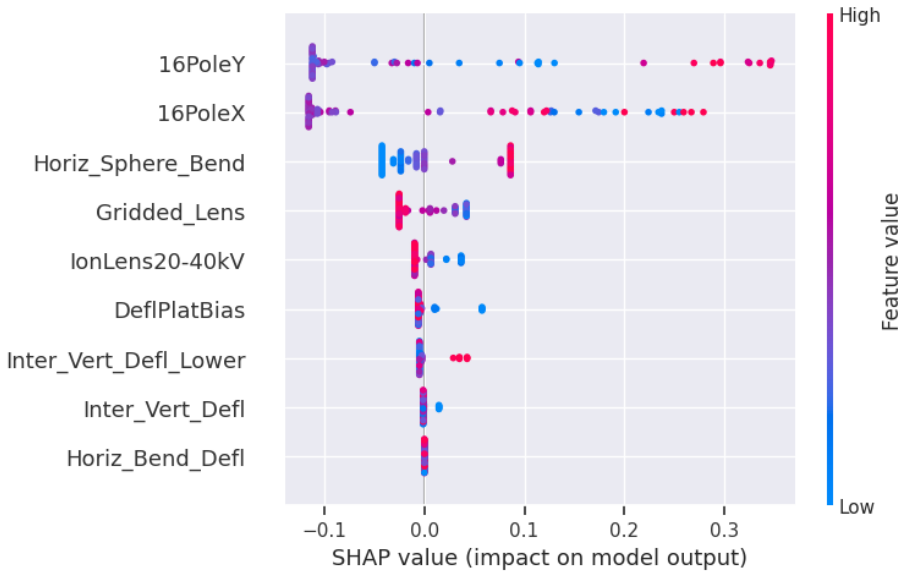

Fig. 11: The importance of individual feature/parameter for the injection beam line.

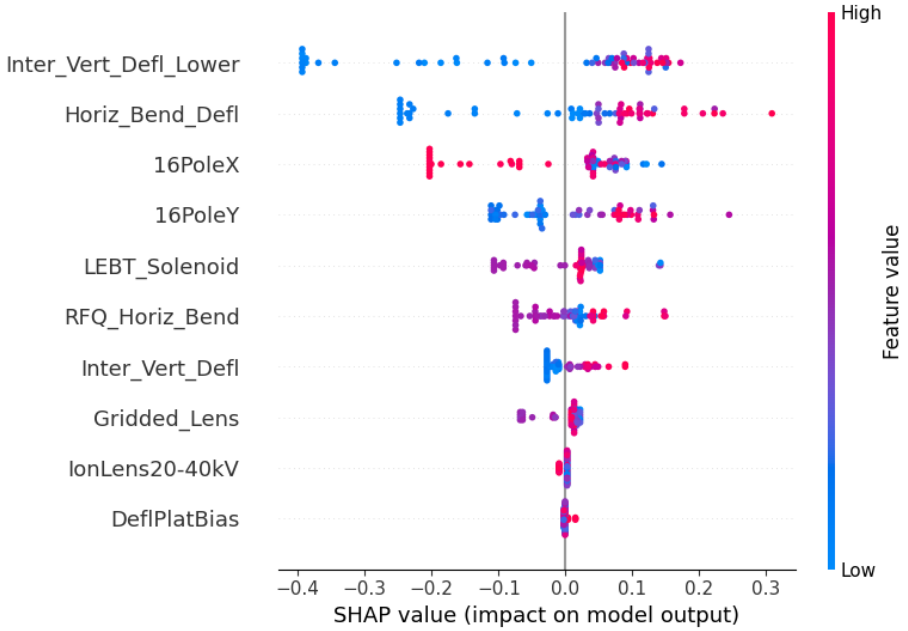

Fig. 12: The importance of individual feature/parameter for the extraction beam line.

4.2 XGBoost Model Feature Importance

To provide a visual summary of the feature importance values calculated by SHAP for a given set of instances, we present Figures 11 and 12. These plots visualize the mean magnitude and direction of the SHAP values for each feature across all instances. Features are sorted based on their mean absolute SHAP values, or importance, providing an overview of their overall impact on the model output.

Positive values indicate a feature's contribution towards increasing the model output, while negative values indicate a contribution towards decreasing it. The length of the dots represents the magnitude of the SHAP values.

From Figure 11, we can observe that 16PoleX and 16PoleY are the two most critical power supplies for beam intensity optimization in the EBIS injection beam line. In contrast, Figure 12 reveals that Inter-vert-Defl-Lower and Horiz-Bend-Defl are the two most important power supplies in the EBIS extraction beam line. For both sets of power supplies in the extraction beam line, higher voltage values tend to correspond with higher beam intensity, as indicated by the right color bar.

Therefore, these plots serve as valuable tools for understanding the relative importance of different features in a model's predictions. They help identify the features that consistently contribute the most to the model output across various instances.

After identifying the top features/parameters that contribute most to the XGBoost model's performance, we can delve deeper into their individual and interactive impacts on predictions using various visualization techniques. These include prediction plots, dependence plots, and interaction plots.

4.3 The Prediction Plot of XGBoost Model

The prediction plot in the pdpbox Python package [10] visualizes the partial dependence of a model's predictions on specific features. It can handle both single and multiple features, providing a comprehensive view of their effects. The function allows you to plot the partial dependence lines, data point distributions, and other pertinent information.

Figure 13 depicts prediction plots for features 16PoleX and 16PoleY in the injection line. The horizontal axes represent the values of 16PoleX and 16PoleY, respectively, each divided into 10 bins. The green bars indicate the number of data points within each bin range, with values (7 or 8) displayed atop the bars. The sum of these values, 70, corresponds to the total iterations of GPTune optimization. The blue curves with error bars represent the predicted beam intensity values for 16PoleX (top plot) and 16PoleY (bottom plot). The plots suggest that optimal results are achieved when 16PoleX falls within the range of [-104.15, -74.78] volts and 16PoleY falls within the range of [-254.61, -245.77] volts. These ranges exhibit minimal error bars and yield the highest beam intensity.

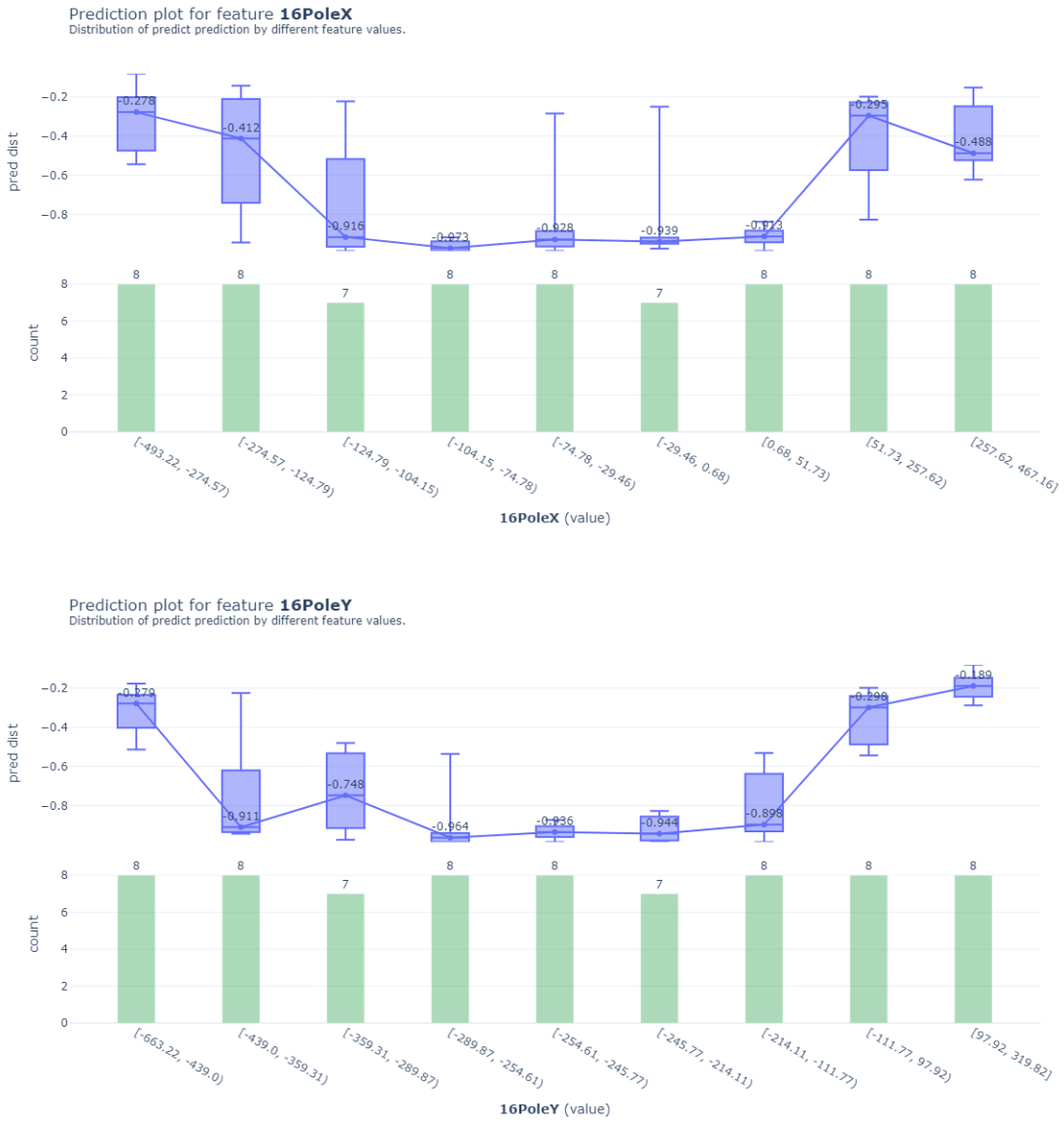

Fig. 13: The prediction plots for 16PoleX (top) and 16PoleY (bottom) in the injection line.

Fig. 14: The prediction plot for Inter-Vert-Defl-Lower (top) and Horiz-Bend-Defl (bottom) in the extraction line.

Figure 14 presents analogous plots for features Inter-Vert-Defl-Lower and Horiz-Bend-Defl. These plots indicate that high beam intensity is achieved when Inter-Vert-Defl-Lower falls within the range of [-373.42, -343.59] volts and Horiz-Bend-Defl falls within the range of [-292.96, -277.26] volts.

4.4 The Partial Dependence Plot of XGBoost Model

In addition to the prediction plot in the pdpbox Python library, the partial dependence function within the same library offers another method for visualizing the partial dependence of a model's predictions on a specific feature. It enables the visualization of the marginal effect of a feature on the predicted outcome while holding other features constant. In other words, partial dependence plots illustrate how the predicted outcome changes in response to variations in a specific feature, while maintaining other features at their average values. The plot

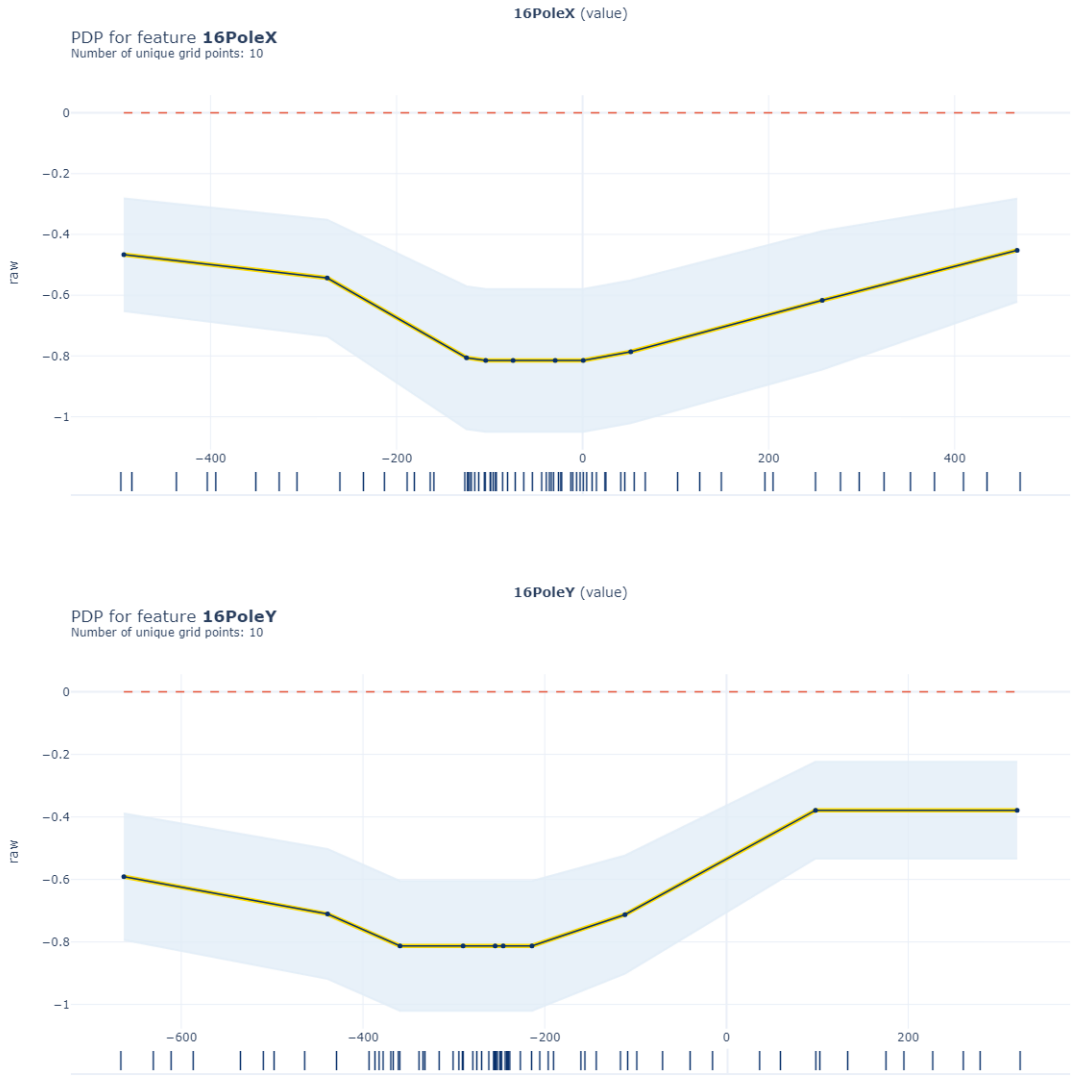

Fig. 15: The partial dependence plot for 16PoleX (top) and 16PoleY (bottom) in the injection line.

demonstrates the average effect of the chosen feature on the model's predictions, aiding in the identification of relationships and patterns. This function can effectively handle both numerical and categorical features.

The partial dependence plots for the features 16PoleX and 16PoleY in the injection line are presented in Fig. 15. The horizontal axis represents the 70 sample values (iterations) for 16PoleX and 16PoleY. The yellow curve, accompanied by a gray-blue shaded area representing one sigma deviation, depicts the predicted beam intensity values for 16PoleX (top plot) and 16PoleY (bottom plot), respectively. These plots lead to conclusions that align with those drawn from Fig. 13.

Similarly, the partial dependence plots for the features Inter-Vert-Defl-Lower (top) and Horiz-Bend-Defl (bottom) in the extraction line are displayed in Fig. 16.

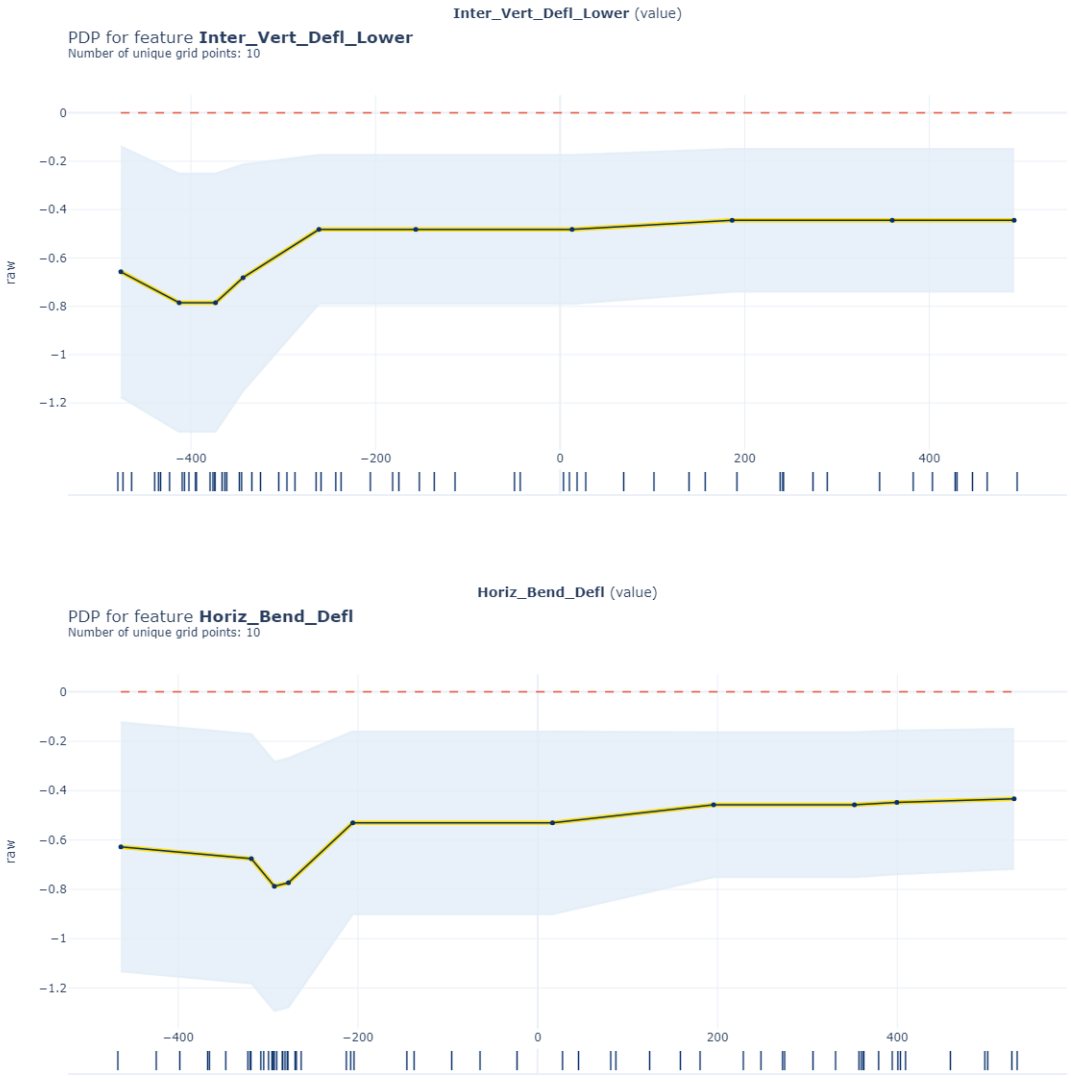

Fig. 16: The partial dependence plot for Inter-Vert-Defl-Lower (top) and Horiz-Bend-Defl (bottom) in the extraction line.

4.5 The Interact Target Plot of XGBoost Model

The interact target plot function visualizes the interaction effects between two features on the model's predictions. By dynamically exploring different combinations of the chosen features, it reveals how the predicted outcome changes based on their joint impact. This interactive plot can be rotated and examined to gain deeper insights into complex relationships between features.

In simpler terms, interaction plots unveil how the interplay between two features affects the model's predictions. They showcase regions of higher or lower predicted outcomes based on various feature combinations. Visualized as grid plots with 3D representations, these plots offer a valuable tool for understanding the joint influence of features on the model's behavior. They highlight interaction patterns that might remain hidden when examining individual features in isolation.

Figures 17 and 18 illustrate the interact target plots for the two most influential features of the injection and extraction lines, respectively. Each axis represents value bins of the corresponding feature, while the size of each circle indicates the number of experimental data points within that two-dimensional range. The sum of these values should be 70 (iterations) for the injection line and 60 (iterations) for the extraction line. Lighter colors represent lower/smaller values, which correspond to better optimization results in the context of GPTune optimization.

Analyzing Figure 17, we observe that for the injection beam line, a promising optimization outcome (though not necessarily the best) is obtainable when 16PoleX falls within the range [-104.15, -74.78] and 16PoleY lies within [-289.87, -254.61]. This aligns with the findings from the prediction plot in Figure 13.

Similarly, from Figure 18, we can identify favorable results with Inter-Vert-Defl-Lower values within the range [-343.59, -261.36] volts and Horiz-Bend-Defl values within [-318.57, -292.96] volts. This closely matches the predictions revealed in Figure 14, where Inter-Vert-Defl-Lower values within [-373.42, -343.59] volts and Horiz-Bend-Defl values within [-292.96, -277.26] volts also lead to advantageous outcomes.

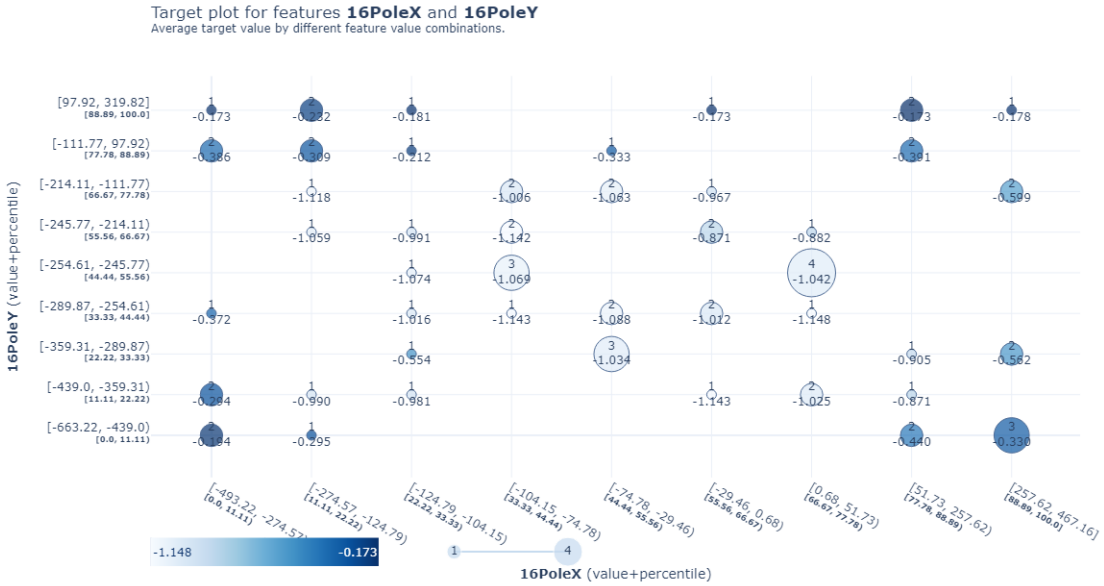

Fig. 17: The interact target plot for the two most importance features in the injection beam line.

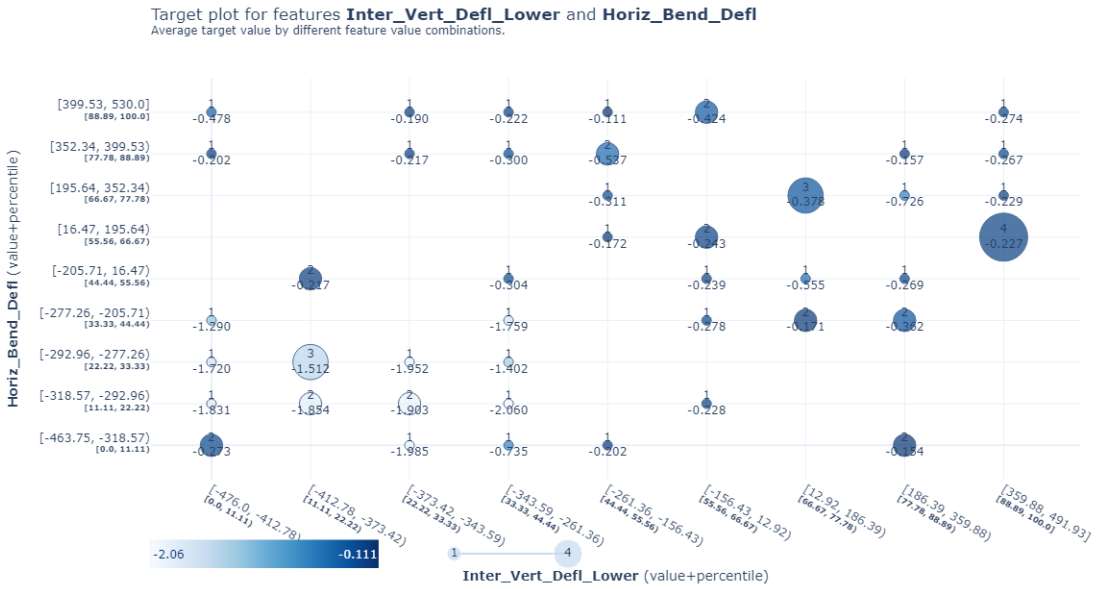

Fig. 18: The interact target plot for the most important features of the extraction beam line.

5 Summary and Discussion

In this paper, we demonstrate the power of GPTune as an optimization tool. We applied it to EBIS intensity optimization and achieved a 22 30% intensity improvement at fc96 (reaching 70% with xf14 CT).

This was achieved despite the presence of noisy signals with 10% standard deviation and involving 19 variables. We plan to expand GPTune's use to other beam lines in the RHIC complex.

Meanwhile, XGBoost exhibits excellent model construction capabilities. Combined with model interpretation algorithms like SHAP, it can provide valuable insights into setting operation parameter ranges.

Xf14 and fc96 beam intensity signals are determined by integrating the waveform signal. Variations in EBIS beam intensity measurements originate from intrinsic beam fluctuations. While noise is present even with no beam, it is negligible compared to the EBIS beam fluctuations.

These fluctuations have two main sources: (1) EBIS itself: The Si beam from EBIS operates satisfactorily for NSRL but is not fully optimized for stability. This inherent fluctuation is beyond the control of the GPTune script. (2) Beam striking electrostatic devices: Suboptimal settings applied by the GPTune script can exacerbate beam instability through these devices.

References

[1] Tianqi Chen, Carlos Guestrin, XGBoost A Scalable Tree Boosting System, arXiv preprint arXiv:1603.02754

[2] https://github.com/dmlc/xgboost

[3] https://home.cern/news/news/computing/flavours-physics-join-lhcb-machine-learning-contest

[4] Claire Adam-Bourdarios, Glen Cowan, Cécile Germain, Isabelle Guyon, Balázs Kégl, et al., The Higgs Machine Learning Challenge, doi:10.1088/1742-6596/664/7/072015

[5] https://cds.cern.ch/record/2017394?ln=en

[6] https://www.sciencedirect.com/topics/computer-science/ensemble-learning

[7] Ray Wright, Interpreting Black-Box Machine Learning Models Using Partial Dependence and Individual Conditional Expectation Plots, Paper SAS1950-2018, https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/1950-2018.pdf

[8] Christoph Molnar, Interpretable Machine Learning - A Guide for Making Black Box Models Explainable, https://christophm.github.io/interpretable-ml-book/pdp.html

[9] Scott Lundberg, Su-In Lee.

[10] https://pdpbox.readthedocs.io/en/latest/

6 Appendix: More PDP Plots and SHAP plots

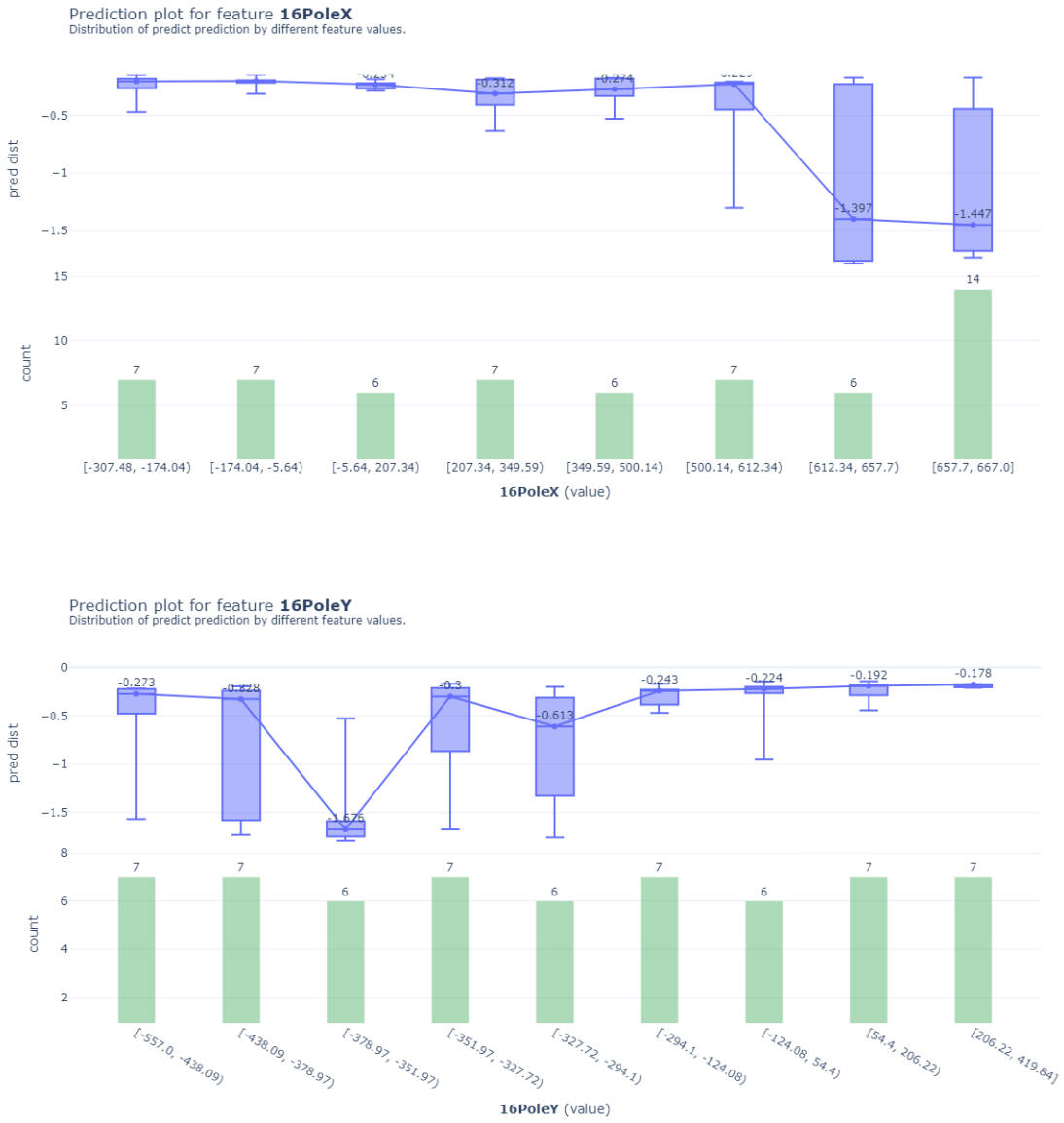

Fig. 19: the prediction plot for feature 16PoleX (top) and 16PoleY (bottom) n the extraction line.

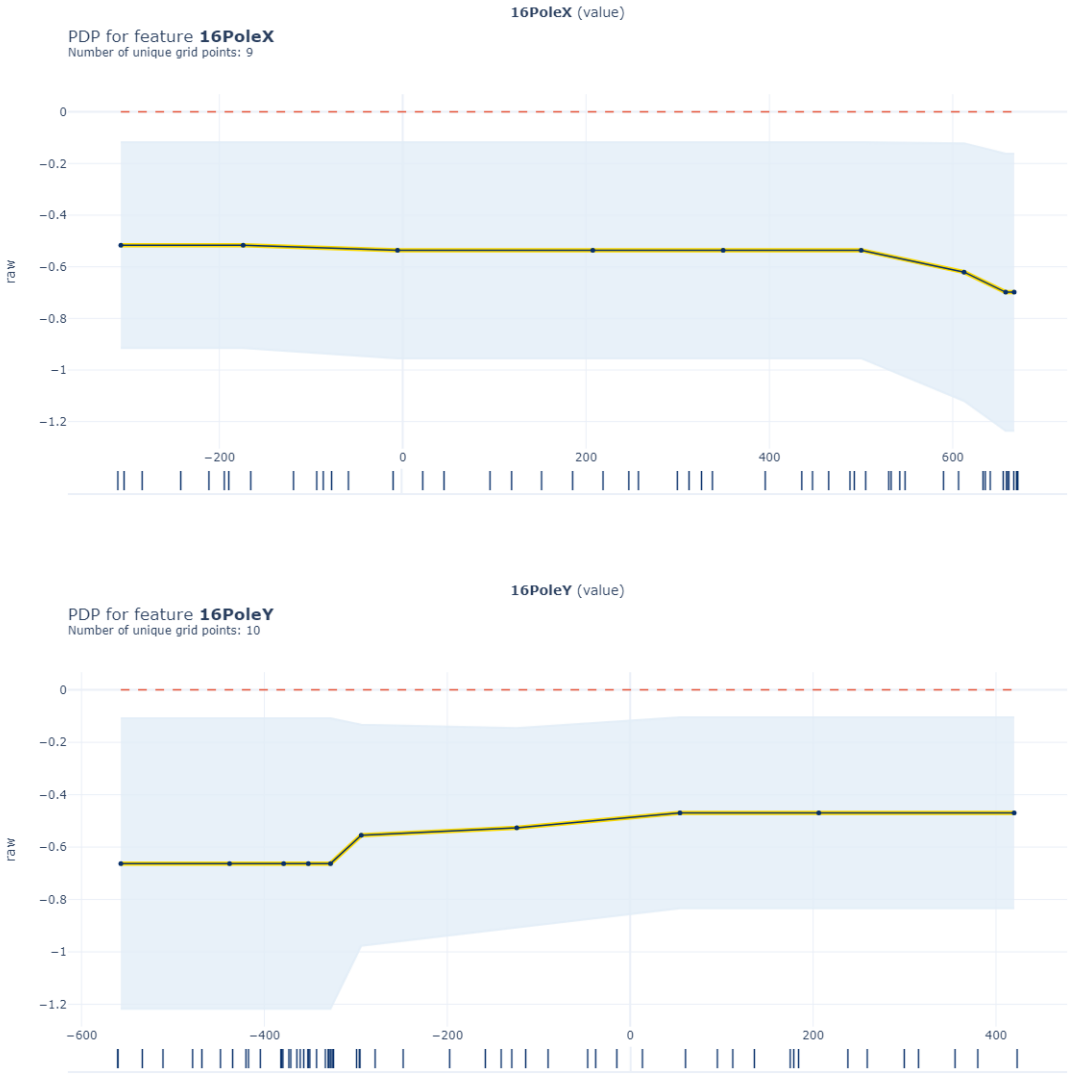

Fig. 20: the partical dependence plot for feature 16PoleX (top) and 16PoleY (bottom) in the extraction line.

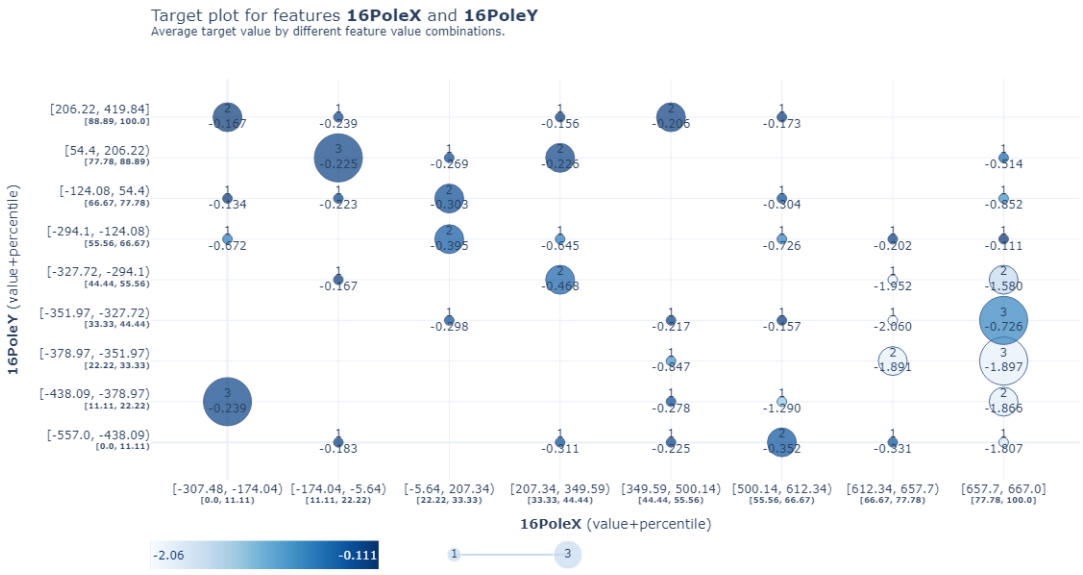

Fig. 21: The interact target plot of 16PoleX and 16PoleY for the extraction beam line.